ACOUSTICALLY GROUNDED WORD EMBEDDINGS FOR IMPROVED ACOUSTICS-TO-WORD SPEECH RECOGNITION

Shane Settle[⋆] *Kartik Audhkhasi*[†] *Karen Livescu*[⋆] *Michael Picheny*[†]

[⋆] TTI-Chicago [†]IBM Research AI

ABSTRACT

Direct acoustics-to-word (A2W) systems for end-to-end automatic speech recognition are simpler to train, and more efficient to decode with, than sub-word systems. However, A2W systems can have difficulties at training time when data is limited, and at decoding time when recognizing words outside the training vocabulary. To address these shortcomings, we investigate the use of recently proposed acoustic and acoustically grounded word embedding techniques in A2W systems. The idea is based on treating the final pre-softmax weight matrix of an AWE recognizer as a matrix of word embedding vectors, and using an externally trained set of word embeddings to improve the quality of this matrix. In particular we introduce two ideas: (1) Enforcing similarity at training time between the external embeddings and the recognizer weights, and (2) using the word embeddings at test time for predicting out-of-vocabulary words. Our word embedding model is *acoustically grounded*, that is it is learned jointly with acoustic embeddings so as to encode the words' acoustic-phonetic content; and it is *parametric*, so that it can embed any arbitrary (potentially out-of-vocabulary) sequence of characters. We find that both techniques improve the performance of an A2W recognizer on conversational telephone speech.

Index Terms— automatic speech recognition, direct acoustics-to-word models, connectionist temporal classification, acoustic word embeddings, triplet contrastive loss

1. INTRODUCTION

End-to-end automatic speech recognition (ASR) focuses on replacing the modular training approaches of traditional ASR systems with conceptually simpler methods. Instead of requiring separately trained acoustic, pronunciation, and language models, neural network-based connectionist temporal classification (CTC) and encoder-decoder approaches allow for joint optimization of a single objective. In principle such models can map acoustics directly to words. However, to achieve performance comparable with traditional methods, these systems are still typically trained to predict sub-word units such as characters or "wordpieces" [1–4], thereby relying on additional decoders and externally trained language models.

Acoustics-to-word (A2W) systems [5–9] jointly model the acoustic, pronunciation, and language models at the word level under a unified framework. Word-level modeling avoids the need for additional decoding, but introduces new challenges. By modeling acoustics at the word level, the system needs to deal with significant variability in word duration, and it is challenging to learn to recognize infrequent words. A2W systems perform well given access to very large amounts of training data. For example, [5] trained a 100K-word vocabulary A2W model and matched the performance of a state-of-the-art sub-word CTC model, but required 125K hours of training speech. Other recent work [6, 7, 9] has explored techniques for training on much less data (e.g., 300 hours), but performance gaps still remain between A2W and sub-word models.

In this work we develop techniques for addressing the challenge of infrequent words in A2W recognition. In an end-to-end neural A2W model, the final weight layer consists of one vector per word in the vocabulary, which can be seen as a word embedding matrix. Most weights in this large matrix are associated with very few training examples since most words are rare. In addition, out-of-vocabulary words cannot be predicted at all (unlike in sub-word models).

Our approach is to first learn a word embedding model in a data-efficient way, and then to use it in two ways: (1) By training the recognizer so as to retain similar weights to the pre-trained embeddings, and (2) for predicting words that are unseen in training. The embedding model learns shared structure between words, and therefore generalizes well to rare or unseen words. Our pre-trained embedding model is learned using techniques from recent work on acoustic word embeddings, which has found that high-quality discriminative embeddings can be learned from very little data (e.g., $\sim$100 minutes) [10–13]. In particular, our embedding approach closely follows that of [12], which jointly learns an acoustic embedding function—mapping an acoustic signal to a fixed-dimensional vector—and an *acoustically grounded* textual embedding function—mapping a character sequence to a vector.

We explore a number of variants of these ideas, and find that A2W recognition performance is consistently improved by either initializing the recognizer's word embeddings with acoustically grounded embeddings or by regularizing toward them. We also introduce a simple method for predicting words outside of the training vocabulary, which improves performance when the training vocabulary is limited.

2. APPROACH

2.1. Acoustics-to-Word Model

The acoustics-to-word (A2W) model uses a single recurrent neural network, typically a bidirectional long short-term memory network (BLSTM), trained with the connectionist temporal classification (CTC) loss to recognize words from input acoustic sequences. Prior work has found that A2W models either require very large amounts of training data [5] or careful training when using limited amounts of training data [6, 7]. In particular, [7] showed that presenting utterances in increasing order of length, initializing with a phone CTC BLSTM, and dropout contributed to significant improvements in the word error rate (WER). This recipe, when applied to the (intermediate sized) 2000-hour Switchboard-Fisher training set, produced a WER on par with several state-of-the-art sub-word based models at the time.

Despite some success training competitive A2W models with large amounts of data, they lag behind conventional models when given more limited training data. Furthermore, an A2W model is trained with a fixed vocabulary and cannot recognize out-of-

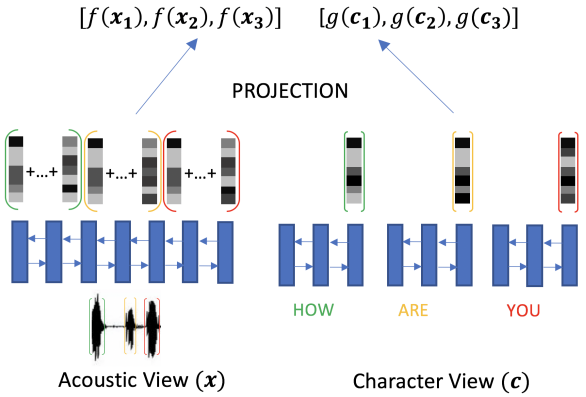

Fig. 1: Acoustically grounded word embeddings from an utterance-transcript pair. The utterance x is fed through the Acoustic View BLSTM, and the hidden state outputs are averaged over each word's alignment region. The character sequence c is fed through the Character View BLSTM and the final output hidden state is retained. Finally, the outputs from each view are passed through a projection layer.

vocabulary (OOV) words. Several prior approaches have been developed to improve the OOV recognition performance of an A2W model. This includes the spell-and-recognize model [7] that is trained to predict the character sequence of a word followed by the the word itself. This enables the model to backoff from an unknown word token to its sequence of characters. Another approach is to train a multi-task CTC network to predict both word and character sequences [14].

2.2. Acoustically Grounded Word Embeddings

Word embeddings, or continuous vector representations of words, are a common tool in natural language processing, and are typically used to represent the meaning (semantics) of words [15–19]. The final (pre-softmax) layer of weights in an A2W model consists of one vector per word in the vocabulary, and can therefore be viewed as embeddings of those words. In fact, in earlier work on A2W-based speech recognition [6], the final layer was initialized with GloVe word embeddings [19]. In this work, we investigate the effect of learning a final weight layer which is encouraged to match externally trained word embeddings. Rather than using semantic word embeddings, we consider whether *acoustically grounded* embeddings—that is, embeddings that encode acoustic-phonetic similarity rather than semantic similarity—may be helpful.

Recent work has explored a number of acoustic and acoustically grounded word embedding approaches. Several approaches have been developed for learning *acoustic word embedding* models—functions mapping arbitrary-duration spoken word signals to fixed-dimensional vectors—so as to encode either phonetic [10, 11, 20–24] or semantic [25] information, or both [26].

Other work has considered *acoustically grounded word embeddings*, that is embeddings of written words that encode their acoustic/phonetic content [12, 24, 27, 28].[1] Our approach, sketched in Figure 1, is based on that of [12], where two embedding functions are learned jointly, one for acoustic signals (spoken words) and one for character sequences (written words).

[1]The term *acoustic word embedding* is sometimes used to refer to embeddings of either spoken or written words. To clarify the distinction, we use *acoustically grounded word embedding* for embeddings of written words.

We learn two embedding models, f and g, which map acoustic sequences $\mathbf{x}$ and character sequences $\mathbf{c}$, respectively, to fixed-dimensional vectors. The acoustic embedding model consists of a stacked BLSTM followed by a sum over the output layer hidden states and a projection to a lower-dimensional vector in $\Re^d$, which is the acoustic embedding $f(\mathbf{x})$. The character sequence embedding model consists of a learned character embedding layer and a single-layer BLSTM; the final hidden state is projected to $\Re^d$ and the result is used as the (acoustically grounded) textual embedding vector $g(\mathbf{c})$. Here we use a shared final projection layer.

We learn the embedding functions f and g jointly so that same-word pairs are mapped to similar vectors while different-word pairs are mapped far apart. Let d denote the cosine distance, $d(x, y) = 1 - \frac{x \cdot y}{\|x\| \|y\|}$, m a margin hyperparameter, and $char(\mathbf{x})$ the character sequence corresponding to the word label of acoustic sequence $\mathbf{x}$. We learn using a sum of two multi-view objectives (namely objectives 0 and 2 of [12]):

$$\min_{f,g} \sum_{i=1}^{N} \left[m + d(f(\mathbf{x}_i), g(\mathbf{c}_i)) - \min_{\mathbf{c} \neq \mathbf{c}_i} d(f(\mathbf{x}_i), g(\mathbf{c})) \right]_+ + \sum_{i=1}^{N} \left[m + d(g(\mathbf{c}_i), f(\mathbf{x}_i)) - \min_{char(\mathbf{x}) \neq \mathbf{c}_i} d(g(\mathbf{c}_i), f(\mathbf{x})) \right]_+ \quad (1)$$

where N is the number of training pairs $(\mathbf{x}_i, \mathbf{c}_i)$. In practice, we do not minimize over all $\mathbf{c} \neq \mathbf{c}_i$ and all $char(\mathbf{x}) \neq \mathbf{c}_i$, but rather select the k most offending examples within each mini-batch and use the mean of their cosine distances [29].

Finally, we use the acoustic segment embedding function f and the character sequence embedding function g in pretraining the A2W speech recognizer. Notice that, since g can be applied to arbitrary character sequences, it is in principle applicable to words that have been seen very few times, or not at all, in training.

2.3. Acoustically Grounded Word Embeddings for Recognition

In prior work on A2W modeling [6, 7, 9], careful model initialization and regularization techniques are cited as essential for effective training on limited data. Two techniques explored in prior work were phone CTC pretraining and GloVe embedding initialization at the word-level prediction layer. Without such initializations, early training methods in this area failed to converge at all [6]. Since GloVe embeddings are trained such that proximity in the embedding space implies similarity in semantics, we may be able to improve performance by instead utilizing an embedding space optimized for acoustic-phonetic similarity. In particular we propose using the acoustically grounded embeddings trained with the contrastive loss of Eq. 1.

We consider using our embeddings in several ways for improved training of the prediction layer weights: (1) **initializing** the weights with the word embeddings, and then training as usual; (2) **regularizing** the weights to remain similar to the word embeddings, after initializing as in (1); and (3) **freezing** the weights at the initialized values given by the word embeddings.

For the regularization approach, we train the recognizer with a training objective that is a weighted average of the baseline recognizer loss and an embedding regularization L2 loss:

$$\mathcal{L}(\mathbf{X}, \mathbf{Y}) = (1 - \lambda) \sum_{i=1}^{N} \mathcal{L}_{CTC}(\mathbf{x}_i, \mathbf{y}_i) + \lambda \sum_{y \in \cup_{i=1}^{N} \mathbf{y}_i} \|g(char(y)) - w(y)\|^2 \quad (2)$$

where $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$ and $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^N$ are a batch of N utterance-transcript pairs, y is a word and char(y) is its character sequence, g is the character sequence embedding function, $w(y)$ is the row of the CTC prediction layer for word y, and $\mathcal{L}_{CTC}$ is the usual CTC loss used for the baseline recognizer.

The last approach, of freezing the prediction layer after initialization, is the extreme case of optimizing for CTC while strictly adhering to the embedding initialization. One helpful feature of this approach is that it can naturally apply to an extended vocabulary at decoding time, some of which was unseen at training time. When using the model to decode, we first predict outputs from the training vocabulary; however, if we predict an <UNK> token, then we rescore by removing <UNK> and predicting from the entire extended vocabulary. Early work on learning acoustic word embeddings [20, 27] used word-level acoustic representations precisely for rescoring in large-vocabulary speech recognition systems. In this work, we show the ability for our learned embedding function g to accurately extend to arbitrary unseen words without any additional training.

3. EXPERIMENTAL SETUP

3.1. Data

We use the standard 300-hour Switchboard corpus of conversational English speech. Speaker-independent 40-dimensional log-Mel features are computed with the addition of Δs+$\Delta\Delta$s and stacking+frame skipping with a rate of 2, resulting in 240-dimensional input acoustic features. Vocabularies of approximately 4K, 10K, and 20K words are used, corresponding to minimum occurrence thresholds set at 25, 6, and 2, respectively. Experiments on out-of-vocabulary extension use a 34K vocabulary for rescoring.

For training the acoustic word embeddings, character sequences are composed of symbols from a 35-character vocabulary: 26 English letters, 6 punctuation symbols ([]<>-'), and 3 sounds ([NOISE], [VOCALIZED-NOISE],[LAUGHTER]). Words containing digits are spelled out using this vocabulary (e.g. "7-11" is spelled "SEVEN-ELEVEN"). Unspoken parts of partial words are in [] with partial starts and ends denoted by < and >, respectively (e.g. "<[YO]UR" and "YO[UR]>"). Acoustic word segment boundaries are obtained from word-level forced alignments produced by a competitive ASR system for Switchboard. When conducting word embedding training, word segments shorter than 6 frames and words outside the training vocabulary are omitted from the embedding loss computation. During pre-processing, if these restrictions omit all words from an utterance, then the utterance itself is removed. The same set of utterances is then used for training both the word embeddings and the recognizer.

3.2. Experiments

During training of both the word embedding and recognition models, a batch size of 64 utterances is used, split across 4 GPUs. The same learning rate reduction scheme is used for both: If the held-out performance (see Section 3.2.1 and 3.2.2) fails to improve after 4 epochs, the learning rate is decayed by a factor of 10 and the model is reset to the previous best. The acoustically grounded word embedding model is trained using the Adam optimizer [30] with an initial learning rate of 0.0005 and parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. The A2W recognition model is trained using stochastic gradient descent (SGD) with Nesterov momentum [31] with an initial learning rate of 0.02 and momentum of 0.9. Training is stopped when the held-out performance stops improving and the learning rate is $< 10^{-8}$. All experiments were conducted using the PyTorch toolkit [32].

3.2.1. Acoustically grounded word embeddings

The acoustic view model is composed of a 6-layer BLSTM with 512 hidden units (per direction per layer), which is initialized with a BLSTM trained as a phone CTC recognizer. The acoustic BLSTM takes full utterances as input, but to produce a word embedding for a given word segment within an utterance, the hidden state outputs corresponding to frames in that segment are averaged to produce a single 1024-dimensional vector. The character view model includes an initial character embedding layer that maps each of the 35 characters to a 64-dimensional vector, followed by a 1-layer BLSTM with 512 hidden units (per direction). A projection layer is used to transform the 1024-dimensional vectors output by the acoustic and character view models to 256-dimensional embeddings. Unlike CTC, the loss is not calculated for each utterance in isolation. Since the multi-view objective (Equation 1) involves sampling the k most offending examples with respect to each acoustic segment and each character sequence, distributing across 4 GPUs limits samples to those present in only 16 utterances. However, we find that distributing speeds up training without degradation in performance. We start with $k = 15$ and reduce to $k = 5$ over the first 300 mini-batches.

As in [12], we evaluate the quality of our embeddings using a cross-view word discrimination task applied to the held-out set. Given a word and an acoustic segment, the goal is to determine if the acoustic segment is a spoken instance of the word. If the cosine distance between their embeddings is below a threshold, we consider them a match. We evaluate performance via the average precision (AP) over all thresholds. This AP is our held-out performance measure, used to tune the embedding model before using it in the A2W recognizer. For M acoustic word segments and a vocabulary of N words, we compute the average precision over $M \times N$ pairs.

3.2.2. Acoustics-to-word recognition model

The A2W model is composed of a 6-layer BLSTM with 512 hidden units (per direction per layer), a 256-dimensional linear projection layer, and a prediction layer with dimension determined by the vocabulary size. Dropout ($p = 0.25$) is used between BLSTM layers. The baseline A2W system uses a phone CTC initialization for the BLSTM, while feed-forward layers are randomly initialized as in [7]. During A2W training, the held-out WER is used to measure performance.

Initialization	SWB Dev WER (%)
Phone CTC init	17.5%
Phone CTC init + AWEs	17.6%
Acoustic view init + AWEs	16.7%

Table 1: SWB development (held-out) word error rates for different methods of initializing the A2W model, using a 10k-word vocabulary.

For word embedding integration experiments, the A2W model is initialized with the acoustic view BLSTM and the 256-dimensional projection layer from multi-view training. The prediction layer is initialized with unit-normalized word embeddings output by the character view model for each word in the vocabulary (with the exception of <BLANK> and <UNK>, which are randomly initialized). We find that multi-view pre-training of the BLSTM and projection layer is essential to improvement over the baseline system (see Table 1).

Regularization experiments penalize deviation (L2 distance) of the prediction layer weights from the word embedding initialization. This penalty is only applied with respect to those words present in a given mini-batch. The hyperparameter $\lambda \in \{0.1, 0.25, 0.5, 0.75, 0.9, 0.99\}$ is tuned to manage the trade-off between the CTC objective and this penalty (Equation 2).

At the extreme end of regularization, we experiment with freezing the prediction layer after initialization, including the randomly initialized <BLANK> and <UNK> tokens. By retaining the consistency of the learned embedding space throughout training, we can acquire new embeddings for any OOV words by running their character sequences through the character view model. We conduct experiments with OOV prediction by concatenating these new word embeddings to the prediction layer, and rescoring with the extended vocabulary whenever an <UNK> is predicted.

Vocab	AGWE	CTC		
		Baseline	Initialized	Regularized
4K	0.894	0.489	0.719	0.762
10K	0.879	0.279	0.644	0.734
20K	0.858	0.160	0.633	0.596

Table 2: Cross-view word discrimination performance, measured via average precision (AP), of acoustically grounded word embeddings (AGWE) and CTC-based embeddings (prediction layer weights).

4. RESULTS

In Table 2, we compare the quality of acoustically grounded word embeddings (AGWE) trained explicitly using the multi-view objective against CTC-based embeddings given by the prediction-layer weights after CTC training. The significantly better AP of AGWE shows that they capture discriminative information that is not discovered implicitly by CTC training. By using these pre-trained embeddings to initialize our model or regularize CTC training, the prediction layer is better able to retain this word discrimination ability.

System	Vocab		
	4K	10K	20K
Baseline	16.4/25.7	14.8/24.9	14.7/24.3
Initialized	15.6/**25.3**	14.2/**24.2**	13.8/24.0
Regularized	15.5/25.4	**14.0**/24.5	**13.7/23.8**
Frozen	15.6/25.6	14.6/24.7	14.2/24.7
+OOV rescoring	**15.0/25.3**	14.4/24.5	14.2/24.7
Curriculum [9]	-	-	13.4/24.2
Curriculum+Joint CTC/CE [9]	-	-	13.0/23.4

Table 3: Results (% WER) on the SWB/CH evaluation sets. The best result for each data set and each vocabulary size is boldfaced.

Table 3 shows evaluation results on the Switchboard (SWB) and CallHome (CH) test sets. We find that both embedding initialization and regularization improve over the baseline WERs. For the regularized model, the values of λ (tuned on held-out data) are 0.25, 0.25, and 0.5 for the 4K, 10K, and 20K vocabularies. However, all values of λ yield improvements on the held-out set over the baseline.

Although outperformed by the embedding-initialized and regularized models, the model trained with a frozen prediction layer ("Frozen") also consistently outperforms the baseline, with the exception of the 20K CH evaluation. An added feature of the Frozen model, as discussed in Section 2.3, is that it allows for straightforward OOV extension via rescoring. Table 3 shows that vocabulary extension and rescoring using the Frozen model improves performance considerably when using the smallest vocabulary. For the 4K vocabulary recognizer, this approach results in an overall absolute WER reduction of 1.4% over the baseline model. We also note that the relative improvement seen by adding OOV rescoring to the 10K vocabulary Frozen model is similar to that offered by the spell-and-recognize system in [7] without the need for additional training.

Recent work [9] shows strong results using a multi-stage A2W approach including curriculum learning from the 10K to the 20K vocabulary, joint CTC/cross entropy (CE) training, and data augmentation. In Table 3, we report results from their most comparable setups, curriculum and joint CTC/CE [9]. Future work may improve further upon these results by combining embedding regularization with the techniques from [9].

Inspecting outputs from the Frozen+OOV rescoring model, we find that when the A2W system produces an <UNK> prediction in place of a single word, we often accurately recover the correct word within the top hypotheses, as seen in the first two rows of Table 4. The majority of remaining mistakes correspond to the first-pass model predicting <UNK> in place of multiple words or part of a word. In such cases we cannot recover the correct word, but we find that many predictions are reasonable phonetic matches for the ground truth. For example, in the third example in Table 4, the first-pass model combines two words "LOANS ARE" into a single <UNK>, and the rescoring model produces a close phonetic match, "LOANER". Other typical examples include splitting up compound words such as "CAREGIVER" and words that are outside the extended 34K vocabulary such as "CANTEENS".

REF: some REMINDERS for me as we are talking **HYP (1st pass):** some <UNK> for me as we are talking **HYP (rescoring):** some **REMINDERS** for me as we are talking
REF: fair and speedy TRIAL **HYP (1st pass):** fair and speedy <UNK> **HYP (rescoring):** fair and speedy **TRIAL**
REF: but those LOANS ARE so much cheaper **HYP (1st pass):** but those <UNK> so much cheaper **HYP (rescoring):** but those **LOANER** so much cheaper
REF: one particular CAREGIVER AND then that one **HYP (1st pass):** one particular CARE <UNK> then that one **HYP (rescoring):** one particular CARE **GIVER** then that one
REF: bring two CANTEENS just to make sure **HYP (1st pass):** bring two <UNK> just to make sure **HYP (rescoring):** bring two **CAMPING'S** just to make sure

Table 4: Successes and failures in OOV prediction with the Frozen+OOV rescoring model.

5. CONCLUSION

We have introduced techniques for using pre-trained acoustically grounded word embeddings for improving acoustics-to-word CTC speech recognition models. We have found that consistent performance improvements can be obtained by incorporating embeddings through initialization, regularization, and out-of-vocabulary prediction. For example, by regularizing the recognizer prediction layer toward the embeddings, we obtain $0.8 - 1\%/0.4 - 0.5\%$ absolute WER improvements for the Switchboard/CallHome data sets. If we also rescore with an expanded vocabulary to resolve OOVs, then in the small-vocabulary (4k-word) case we can improve the WER by a total of 1.4% absolute on Switchboard.

Future directions include exploring additional kinds of embedding models and training criteria, as well as tighter integration of the embedding and recognizer training. Another promising direction, considering the encouraging results with small vocabulary sizes, is to apply these ideas to the recognition of low-resource languages.

6. REFERENCES

[1] K. Rao, H. Sak, and R. Prabhavalkar, "Exploring architectures, data and units for streaming end-to-end speech recognition with RNN-transducer," in *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2017.

[2] C.-C. Chiu, T. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. Weiss, K. Rao, E. Gonina, et al., "State-of-the-art speech recognition with sequence-to-sequence models," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[3] R. Sanabria and F. Metze, "Hierarchical Multi Task Learning With CTC," in *Proc. IEEE Workshop on Spoken Language Technology (SLT)*, 2018.

[4] K. Krishna, S. Toshniwal, and K. Livescu, "Hierarchical Multitask Learning for CTC-based Speech Recognition," *arXiv preprint arXiv:1807.06234*, 2018.

[5] H. Soltau, H. Liao, and H. Sak, "Neural speech recognizer: Acoustic-to-word LSTM model for large vocabulary speech recognition," in *Proc. Interspeech*, 2017.

[6] K. Audhkhasi, B. Ramabhadran, G. Saon, M. Picheny, and D. Nahamoo, "Direct acoustics-to-word models for English conversational speech recognition," in *Proc. Interspeech*, 2017.

[7] K. Audhkhasi, B. Kingsbury, B. Ramabhadran, G. Saon, and M. Picheny, "Building competitive direct acoustics-to-word models for English conversational speech recognition," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[8] J. Li, G. Ye, A. Das, R. Zhao, and Y. Gong, "Advancing acoustic-to-word CTC model," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[9] C. Yu, C. Zhang, C. Weng, J. Cui, and D. Yu, "A multistge training framework for acoustic-to-word model," in *Proc. Interspeech*, 2018.

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

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

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

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

[14] J. Li, G. Ye, R. Zhao, J. Droppo, and Y. Gong, "Acoustic-to-word model without OOV," in *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2017.

[15] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American society for information science*, vol. 41, no. 6, pp. 391, 1990.

[16] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of Machine Learing Research*, vol. 3, no. Feb, pp. 1137–1155, 2003.

[17] A. Mnih and G. Hinton, "Three new graphical models for statistical language modelling," in *Int. Conf. on Machine Learning (ICML)*, 2007.

[18] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems (NIPS)*, 2013.

[19] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proc. Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[20] A. Maas, S. Miller, T. O'neil, A. Ng, and P. Nguyen, "Word-level acoustic modeling with convolutional vector regression," in *Proc. ICML Workshop on Representation Learning*, 2012.

[21] K. Levin, K. Henry, A. Jansen, and K. 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.

[22] G. Chen, C. Parada, and T. Sainath, "Query-by-example keyword spotting using long short-term memory networks," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2015.

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

[24] K. Audhkhasi, A. Rosenberg, A. Sethy, B. Ramabhadran, and B. Kingsbury, "End-to-end asr-free keyword search from speech," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1351–1359, 2017.

[25] Y.-A. Chung and J. Glass, "Speech2vec: A sequence-to-sequence framework for learning word embeddings from speech," in *Proc. Interspeech*, 2018.

[26] Y.-C. Chen, S.-F. Huang, C.-H. Shen, H.-Y. Lee, and L.-S. Lee, "Phonetic-and-semantic embedding of spoken words with applications in spoken content retrieval," *Proc. IEEE Workshop on Spoken Language Technology (SLT)*, 2018.

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

[28] S. Ghannay, Y. Esteve, N. Camelin, and P. Deleglise, "Evaluation of acoustic word embeddings," in *Proc. ACL Workshop on Evaluating Vector-Space Representations for NLP*, 2016.

[29] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2015.

[30] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Proc. Int. Conf. on Learning Representations (ICLR)*, 2015.

[31] Y. Nesterov, "A method for solving the convex programming problem with convergence rate o (1/kˆ 2)," in *Dokl. Akad. Nauk SSSR*, 1983, vol. 269, pp. 543–547.

[32] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," 2017.