END-TO-END ANCHORED SPEECH RECOGNITION

Yiming Wang[1*], *Xing Fan*[2], *I-Fan Chen*[2], *Yuzong Liu*[2], *Tongfei Chen*[1*], *Björn Hoffmeister*[2]

[1] Center for Language and Speech Processing, Johns Hopkins University, Baltimore, MD, USA
[2] Amazon.com, Inc., USA
{yiming.wang,tongfei}@jhu.edu,
{fanxing,ifanchen,liuyuzon,bjornh}@amazon.com

ABSTRACT

Voice-controlled house-hold devices, like Amazon Echo or Google Home, face the problem of performing speech recognition of device-directed speech in the presence of interfering background speech, i.e., background noise and interfering speech from another person or media device in proximity need to be ignored. We propose two end-to-end models to tackle this problem with information extracted from the *anchored segment*. The anchored segment refers to the wake-up word part of an audio stream, which contains valuable speaker information that can be used to suppress interfering speech and background noise. The first method is called *Multi-source Attention* where the attention mechanism takes both the speaker information and decoder state into consideration. The second method directly learns a frame-level mask on top of the encoder output. We also explore a multi-task learning setup where we use the ground truth of the mask to guide the learner. Given that audio data with interfering speech is rare in our training data, we also propose a way to synthesize "noisy" speech from "clean" speech to mitigate the mismatch between training and test data. Our proposed methods show up to 15% relative reduction in WER for Amazon Alexa live data with interfering background speech without significantly degrading on clean speech.

Index Terms— End-to-End ASR, anchored speech recognition, attention-based encoder-decoder network, robust speech recognition

1. INTRODUCTION

We tackle the ASR problem in the scenario where a foreground speaker first wakes up a voice-controlled device with an "anchor word", and the speech after the anchor word is possibly interfered with background speech from other people or media. Consider the following example:

SPEAKER 1: *Alexa, play rock music.*

SPEAKER 2: *I need to go grocery shopping.*

Here the wake-up word "Alexa" is the anchor word, and thus the utterance by speaker 1 is considered as device-directed speech, while the utterance by speaker 2 is the interfering speech. Our goal is to extract information from the anchor word in order to recognize the device-directed speech and ignore the interfering speech. We name this task *anchored speech recognition*. The challenge of this task is to learn a speaker representation from a short segment corresponding to the anchor word. A couple of techniques have been proposed for learning speaker representations, e.g., i-vector [1, 2], mean-variance normalization [3], maximum likelihood linear regression (MLLR) [4]. With the recent progress in deep learning, neural networks are used to learn speaker embeddings for speaker verification/recognition [5, 6, 7, 8]. More relevant to our task, two methods—anchor mean subtraction (AMS) and an encoder-decoder network—are proposed to detect desired speech by extracting speaker characteristics from the anchor word [9]. This work is further extended for acoustic modeling in hybrid ASR systems [10]. Speaker-dependent mask estimation is also explored for target speaker extraction [11, 12].

Recently, much work has been done towards end-to-end approaches for speech recognition [13, 14, 15, 16, 17, 18]. These approaches typically have a single neural network model to replace previous independently-trained components, namely, acoustic, language, and pronunciation models from hybrid HMM systems. End-to-end models greatly alleviate the complexity of building an ASR system. Compared with the others, the attention-based encoder-decoder models [19, 16] do not assume conditional independence for output labels as CTC based models [20, 14] do.

We propose two end-to-end models for anchored speech recognition, focusing on the case where each frame is either completely from device-directed speech or completely from interfering speech, but not a mixture of both [11, 21]. They are both based on the attention-based encoder-decoder models. The attention mechanism provides an explicit way of aligning each output symbol with different input frames, enabling *selective decoding* from an audio, i.e., only decode desired speech that is taking place in part of the entire audio stream. In the first method, we incorporate the speaker information when calculating the attention energy, which leads to an anchor-aware soft alignment between the decoder state and encoder output. The second method learns a frame-level mask on top of the encoder, where the mask can optionally be learned in the multi-task framework if the gold mask is given. This method will pre-select the encoder output before the attention energy is calculated. Furthermore, since the training data is relatively clean, in the sense that it contains device-directed speech only, we propose a method to synthesize "noisy" speech from "clean" speech, mitigating the mismatch between training and test data.

We conduct experiments on a training corpus consisting of 1200 hours live data in English from Amazon Echo. The results demonstrate a significant WER relative gain of 12-15% in test sets with interfering background speech. For a test set that contains only device-directed speech, we see a small relative WER degradation from the proposed method, ranging from 1.5% to 3%.

This paper is organized as follows. Section 2 first gives an overview of the attention-based encoder-decoder model and then presents our two end-to-end anchored ASR models. Section 3 describes how we synthesize our training data and train our second proposed model in a multi-task fashion. Section 4 shows our experiments and results. Section 5 includes conclusions and future work.

[*] This work was done while the authors were interns at Amazon.

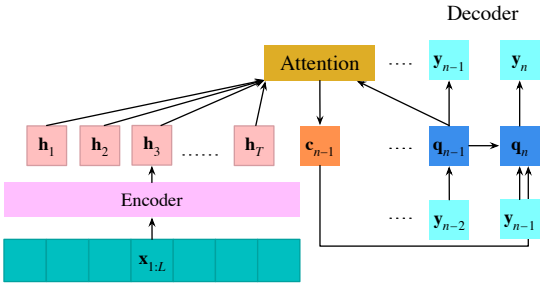

Fig. 1. Attention-based Encoder-Decoder Model. It is an illustration in the case of a one-layer decoder. If there are more layers, as in our experiments, an updated context vector $\mathbf{c}_n$ will also be fed into each of the upper layers in the decoder at the time step n.

2. MODEL OVERVIEW

2.1. Attention-based Encoder-Decoder Model

The basic attention-based encoder-decoder model typically consists of 3 modules as depicted in Fig 1: 1) an encoder transforming a sequence of input features $\mathbf{x}_{1:L}$ into a high-level representation of the features $\mathbf{h}_{1:T}$ through a stack of convolution/recurrent layers, where $T \leq L$ due to possible frame down-sampling; 2) an attention module summarizing the output of the encoder $\mathbf{h}_{1:T}$ into a fixed length context vector $\mathbf{c}_n$ at each output step for $n \in [1, \ldots, N]$, which determines parts of the sequence $\mathbf{h}_{1:T}$ to be attended in order to predict the output symbol y_n; 3) a decoder module taking the context vector $\mathbf{c}_n$ as input and predicting the next symbol y_n given the history of previous symbols $y_{1:n-1}$. The entire model can be formulated as follows:

$$\mathbf{h}_{1:T} = \text{Encoder}(\mathbf{x}_{1:L}) \quad (1)$$
$$\alpha_{n,t} = \text{Attention}(\mathbf{q}_n, \mathbf{h}_t) \quad (2)$$
$$\mathbf{c}_n = \sum_t \alpha_{n,t}\mathbf{h}_t \quad (3)$$
$$\mathbf{q}_n = \text{Decoder}(\mathbf{q}_{n-1}, [y_{n-1}; \mathbf{c}_{n-1}]) \quad (4)$$
$$y_n = \arg\max_v(\mathbf{W}^f\mathbf{q}_n + \mathbf{b}^f) \quad (5)$$

Although our proposed methods do not limit itself in any particular attention mechanism, we choose the *Bahdanau Attention* [19] as the attention function for our experiments. So Eqn. (2) takes the form of:

$$\omega_{n,t} = \mathbf{v}^\top \tanh(\mathbf{W}^q\mathbf{q}_n + \mathbf{W}^h\mathbf{h}_t + \mathbf{b}) \quad (6)$$
$$\alpha_{n,t} = \text{softmax}(\omega_{n,t}) \quad (7)$$

2.2. Multi-source Attention Model

Our first approach is based on the intuition that the attention mechanism should consider both the speaker information and the decoder state – when computing the attention weights, in addition to conditioning on the decoder state, the speaker information extracted from the input frames is also utilized. In our scenario, the device-directed speech and the anchor word are uttered by the same speaker, while the interfering background speech is from a different speaker. Therefore, the attention mechanism can be augmented by placing more attention probability mass on frames that are more similar to the anchor word in terms of speaker characteristics.

Formally speaking, besides our previous notations, the anchor word segment is denoted as $\mathbf{w}_{1:L'}$. We add another encoder S-Encoder to be applied on both $\mathbf{w}_{1:L'}$ and $\mathbf{x}_{1:L}$ to generate a fixed-length vector $\tilde{\mathbf{w}}$ and a variable length sequence $\mathbf{u}_{1:T}$ respectively:

$$\tilde{\mathbf{w}} = \text{Pooling}(\text{S-Encoder}(\mathbf{w}_{1:L'})) \quad (8)$$
$$\mathbf{u}_{1:T} = \text{S-Encoder}(\mathbf{x}_{1:L}) \quad (9)$$

As shown above, S-Encoder extracts speaker characteristics from the acoustic features. In our experiments, the pooling function is implemented as *Max-pooling* across all output frames if S-Encoder is a convolutional network, or picking the hidden state of the last frame if S-Encoder is a recurrent network. Rather than being appended to acoustic feature vector and fed into the decoder[1] as proposed in [10], $\tilde{\mathbf{w}}$ is directly involved in computing the attention weights. Specifically, Eqn. (7) and Eqn. (3) are replaced by:

$$\phi_t = \text{Similarity}(\mathbf{u}_t, \tilde{\mathbf{w}}) \quad (10)$$
$$\alpha_{n,t}^{\text{anchor\_aware}} = \text{softmax}(\omega_{n,t} + g \cdot \phi_t) \quad (11)$$
$$\mathbf{c}_n = \sum_t \alpha_{n,t}^{\text{anchor\_aware}}\mathbf{h}_t \quad (12)$$

where g is a trainable scalar used to automatically adjust the relative contribution from the speaker acoustic information. $\text{Similarity}(\cdot,\cdot)$ is implemented as dot-product in our experiments. As a result, the attention weights are essentially computed from two different sources: the ASR decoding state, and the confidence of decision on whether each frame belongs to the device-directed speech. We call this model *Multi-source Attention* to reflect the way the attention weights are computed.

2.3. Mask-based Model

The Multi-source Attention model jointly considers speaker characteristic and ASR decoder state when calculating the attention weights. However, since the attention weights are normalized with a softmax function, whether each frame needs to be ignored is not independently decided, which reduces the modeling flexibility in frame selection.

As the second approach we propose the *Mask-based* model, where a frame-wise mask on top of the encoder[2] is estimated by leveraging the speaker acoustic information contained in the anchor word and the actual recognition utterance. The attention mechanism is then performed on the masked feature representation. Compared with the Multi-source Attention model, attention in the Mask-based model only focuses on remaining frames after masking, and for each frame it is independently decided whether to be masked out based on their acoustic similarity. Formally, Eqn. (6) and Eqn. (3) are modified as:

$$\phi_t = \text{sigmoid}(g \cdot \text{Similarity}(\mathbf{u}_t, \tilde{\mathbf{w}})) \quad (13)$$
$$\mathbf{h}_t^{\text{masked}} = \phi_t\mathbf{h}_t \quad (14)$$
$$\omega_{n,t} = \mathbf{v}^\top \tanh(\mathbf{W}^q\mathbf{q}_n + \mathbf{W}^h\mathbf{h}_t^{\text{masked}} + \mathbf{b}) \quad (15)$$
$$\mathbf{c}_n = \sum_t \alpha_{n,t}\mathbf{h}_t^{\text{masked}} \quad (16)$$

where $\text{Similarity}(\cdot,\cdot)$ in Eqn. (13) is dot-product as well.

[1] We tried that in our preliminary experiments but it did not perform well.

[2] Here "frame-wise" actually means *frame-wise after down-sampling*, in accordance with the frame down-sampling in the encoder network (see Section 4.1 for details).

3. SYNTHETIC DATA AND MULTI-TASK TRAINING

3.1. Synthetic Data

A problem we encountered in our task is: there is very little training data that has the same condition as the test case. Some utterances in the test set contain speech from two or more speakers (denoted as the "speaker change" case), and some of the other utterances only contain background speech (denoted as the "no desired speaker" case). In contrast, most of the training data does not have interfering or background speech, making the model unable to learn to ignore.

In order to simulate the condition of the test case, we generate two types of synthetic data for training:

- Synthetic Method 1: for an utterance, a random segment[3] from another utterance in the dataset is inserted at a random position after the wake-up word part within this utterance, while its transcript is unchanged.
- Synthetic Method 2: the entire utterance, excluding the wake-up word part, is replaced by another utterance, and its transcript is considered as empty.

Fig. 2 illustrates the synthesizing process. These two types of synthetic data simulate the "speaker change" case and the "no desired speaker" case respectively. The synthetic and device-directed data are mixed together to form our training data. The mixing proportion is determined from experiments.

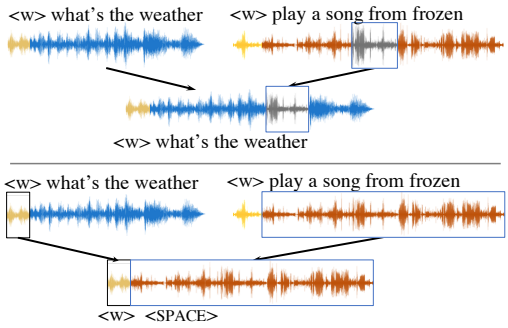

Fig. 2. Two types of synthetic data: Synthetic Method 1 (top) and 2 (bottom). The symbol ⟨W⟩ represents the wake-up word, and ⟨SPACE⟩ represents empty transcripts.

3.2. Multi-task Training for Mask-based Model

For the generated synthetic data, we know which frames come from the original utterance and which are not, i.e., we have the gold mask for each synthetic utterance, where the frames from the original utterance are labeled with "1", and the other frames are labeled with "0". Using this gold mask as an auxiliary target, we train the Mask-based model in a multi-task way, where the overall loss is defined as a linear interpolation of the normal ASR cross-entropy loss and the cross-entropy-based mask loss: $(1-\lambda)\mathcal{L}_{\mathbf{ASR}} + \lambda\mathcal{L}_{\mathbf{mask}}$.

The gold mask provides a supervision signal to explicitly guide S-Encoder to extract acoustic features that can better distinguish the inserted frames from those in the original utterance. As will be shown in our experiments, with the multi-task training the predicted mask is more accurate in selecting desired frames for the decoder.

[3]The frame length of a segment is uniformly sampled within the range [50,150] in our experiments. It is possible that the randomly selected segment is purely non-speech or even silence.

4. EXPERIMENTS

4.1. Experimental Settings

We conduct our experiments on training data of 1200-hour live data in English collected from the Amazon Echo. Each utterance is hand-transcribed and begins with the same wake-up word whose alignment with time is provided by end-point detection [22, 23, 24, 25]. As we have mentioned, while the training data is relatively clean and usually only contains device-directed speech, the test data is more challenging and under mismatched conditions with training data: it may be noisy, may contain background speech[4], or may even contain no device-directed speech at all. In order to evaluate the performance on both the matched and mismatched cases, two test sets are formed: a "normal set" (25k words in transcripts) where utterances have a similar condition as those in the training set, and a "hard set" (5.4k words in transcripts) containing the challenging utterances with interfering background speech. Note that both of the two test sets are real data without any synthesis. We also prepare a development set ("normal"+"hard") with a similar size as the test sets for hyper-parameter tuning. For all the experiments, 64-dimensional log filterbank energy (LFBE) features are extracted every 10ms with a window size of 25ms. The end-to-end systems are grapheme-based and the vocabulary size is 36, which is determined by thresholding on the minimum number of character counts from the training transcripts. Our implementation is based on the open-sourced toolkit OPENSEQ2SEQ [26].

Our baseline end-to-end model does not consider anchor words. Its encoder consists of three convolution layers resulting in 2x frame down-sampling and 8x frequency down-sampling, followed by 3 Bi-directional LSTM [27] layers with 320 hidden units. Its decoder consists of 3 unidirectional-LSTM layers with 320 hidden units. The attention function is *Bahdanau Attention* [19]. The cross-entropy loss on characters is optimized using Adam [28], with an initial learning rate 0.0008 which is then adjusted by exponential decay. A beam search with beam size 15 is adopted for decoding. The above setting is also used in our proposed models.

4.2. Multi-source Attention Model vs. Baseline

S-Encoder consists of three convolution layers with the same architecture as that in the baseline's encoder.

First of all, we compare Multi-source Attention Model and the baseline trained on the device-directed-only data, i.e., without any synthetic data. The results are shown in Table 1.

Table 1. Multi-source Attention Model vs. Baseline with device-directed-only training data. The WER, substitution, insertion and deletion values are all normalized by the baseline WER on the "normal" set[6]. The normalization applies to all the tables throughout this paper in the same way.

Model	Training Set	Test Set	WER	sub	ins	del	WERR(%)
Baseline	Device-directed-only	normal	1.000	0.715	0.108	0.177	—
		hard	3.354	1.762	1.123	0.469	—
Mul-src. Attn.	Device-directed-only	normal	1.015	0.731	0.115	0.169	-1.5
		hard	3.262	1.746	1.062	0.454	+2.8

[4]background speech includes: 1) interfering speech from an actual non-device-directed speaker; and 2) multi-media speech, meaning that a television, radio, or other media device is playing back speech in the background.

[6]For example, if WER for the baseline is 5.0% for the "normal" set, and

The relative WER reduction (WERR) of Multi-source Attention on the "hard" set is 2.8% and it is mostly due to a reduction in insertion errors. We also observe a slight WER degradation of 1.5% relative on the "normal" set. It implies that the proposed model is more robust to interfering background speech.

Next, we further validate the effectiveness of the Multi-source Attention model by showing how synthetic training data has different impact on it and the baseline model respectively. Synthetic training data is prepared such that 50% of the utterances in the training set are kept unchanged, 44% are processed with Synthetic Method 1, and 6% are processed with Synthetic Method 2. The ratio is tuned on the development set. This new training data is referred as "augmented" in all result tables. Table 2 exhibits the results. For the baseline model, the performance degrades drastically when trained on augmented data: the deletion errors on both of the "normal" and "hard" test sets get much higher. This is expected since without the anchor word the model has no extra acoustic information of which part of the utterance is desired, so that it tends to ignore frames regardless of whether they are actually from device-directed speech. On the contrary, for the Multi-source Attention model the WERR (augmented vs. device-directly-only) on the "hard" set is 12.5%, and WER on the "normal" set does not get worse. Moreover, the insertion errors on both test sets get reduced while the deletion errors increase much less than those in the case of the baseline model, indicating that by incorporating the anchor word information the proposed model effectively improves the ability of focusing on device-directed speech and ignoring others. This series of experiments also reveals significant benefits from using the synthetic data with the proposed model. In total, the combination of the Multi-source Attention model and augmented training data achieves 14.9% WERR on the "hard" set, with only 1.5% degradation on the "normal" set.

Table 2. Augmented vs. Device-directed-only training data. Results with "Device-directed-only" training set are from Table 1 for clearer comparisons.

Model	Training Set	Test Set	WER	sub	ins	del	WERR(%)
Baseline	Device-directed-only	normal	1.000	0.715	0.108	0.177	—
		hard	3.354	1.762	1.123	0.469	—
	Augmented	normal	3.215	1.223	0.038	1.954	-221.5
		hard	4.208	1.777	0.246	2.185	-30.9
Mul-src. Attn.	Device-directed-only	normal	1.015	0.731	0.115	0.169	**-1.5**
		hard	3.262	1.746	1.062	0.454	+2.8
	Augmented	normal	1.015	0.700	0.108	0.207	**-1.5**
		hard	2.854	1.569	0.723	0.562	**+14.9**

4.3. Mask-based Model

In the Mask-based model experiments, 3 convolution and 1 Bi-directional LSTM layers are used as S-Encoder, as we observed that it empirically performs better than convolution-only layers. Due to the importance of using the augmented data for training our previous model, the same synthetic approach is directly applied to train the Mask-based model. Also, as we mentioned in Sec 3.2, multi-task training can be conducted since we know the gold mask for each synthesized utterance. Given the imbalanced mask labels, i.e., frames with label "1" (corresponding to those from the original utterance) constitute the majority compared with frames with label "0" (corresponding to those from another random utterance), we use weighted cross entropy loss for the auxiliary mask learning task, where the weight on frames with label "1" is 0.6 and on those with label "0" is 1.0, to counteract the label imbalance.

We first set the multi-task loss weighting factor $\lambda = 1.0$ so that only the mask learning is performed. It turns out that around 70% of frames with label "0" and 98% with label "1" are recalled on a held-out set synthesized the same way as the training data, which demonstrates the effectiveness of estimating masks from the synthetic data.

Then we perform ASR using the Mask-based model with and without mask supervision respectively, and the results are presented in Table 3. WERRs are all relative to the baseline model trained on device-directed-only data. For the Mask-based model without mask supervision, it achieves 3.9% WERR on the "hard" set while has a degradation of 34.8% on the "normal" set. On the other hand, with mask supervision ($\lambda = 0.1$) corresponding to the multi-task training, it yields 12.6% WERR on the "hard" set while only 3.0% worse on the "normal" set. The performance gap between them can be attributed to the ability of mask prediction: while with mask supervision the recall is still around 70% (for frames labeled as "0") and 98% (for frames labeled as "1") on the held-out set, it is only 48% and 50% respectively without mask supervision.

Note that even with multi-task training, the WER performance of the Mask-based model is still slightly behind the Multi-source Attention model, mainly due to the insertion error. Our conjecture is, the mask prediction is only done within the encoder, which may lose semantic information from the decoder that is potentially useful for discriminating device-directed speech from others.

Table 3. Mask-based Model: with and without mask supervision.

Model	Training Set	Test Set	WER	sub	ins	del	WERR(%)
w/o Supervision	Augmented	normal	1.348	0.725	0.096	0.527	-34.8
		hard	3.223	1.508	0.628	1.087	+3.9
w/ Supervision	Augmented	normal	1.030	0.715	0.115	0.200	**-3.0**
		hard	2.931	1.586	0.809	0.536	**+12.6**

5. CONCLUSIONS AND FUTURE WORK

In this paper we propose two approaches for end-to-end anchored speech recognition, namely *Multi-source Attention* and the *Mask-based* model. We also propose two ways to generate synthetic data for end-to-end model training to improve the performance. Given the synthetic training data, a multi-task training scheme for the Mask-based model is also proposed. With the information extracted from the anchor word, both of these methods show their ability in picking up device-directed part of speech and ignore other parts. This results in large WER improvement of 15% relative on the test set with interfering background speech, with only a minor degradation of 1.5% on clean speech. Obviously the mismatch still exists between the training and test data. Future work would include finding a better way to generate synthetic data with more similar condition to the "hard" test set, and taking decoder state into consideration when estimating the mask. The other direction is to utilize anchor word information in contextual speech recognition [29].

6. ACKNOWLEDGEMENTS

The authors would like to thank Hainan Xu for proofreading.

25.0% for the "hard" set, then the normalized values would be 1.000 and 5.000 respectively.

7. REFERENCES

[1] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[2] George Saon, Hagen Soltau, David Nahamoo, and Michael Picheny, "Speaker adaptation of neural network acoustic models using i-vectors.," in *ASRU*, 2013, pp. 55–59.

[3] Fu-Hua Liu, Richard M Stern, Xuedong Huang, and Alejandro Acero, "Efficient cepstral normalization for robust speech recognition," in *Proceedings of the workshop on Human Language Technology*. Association for Computational Linguistics, 1993, pp. 69–74.

[4] Christopher J Leggetter and Philip C Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models," *Computer speech & language*, vol. 9, no. 2, pp. 171–185, 1995.

[5] Ehsan Variani, Xin Lei, Erik McDermott, and Javier Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification.," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 4080–4084.

[6] Mitchell McLaren, Yun Lei, and Luciana Ferrer, "Advances in deep neural network approaches to speaker recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4814–4818.

[7] David Snyder, Pegah Ghahremani, Daniel Povey, Daniel Garcia-Romero, Yishay Carmiel, and Sanjeev Khudanpur, "Deep neural network-based speaker embeddings for end-to-end speaker verification," in *Spoken Language Technology Workshop (SLT), 2016 IEEE*. IEEE, 2016, pp. 165–170.

[8] Georg Heigold, Ignacio Moreno, Samy Bengio, and Noam Shazeer, "End-to-end text-dependent speaker verification," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 5115–5119.

[9] Roland Maas, Sree Hari Krishnan Parthasarathi, Brian King, Ruitong Huang, and Björn Hoffmeister, "Anchored speech detection," in *INTERSPEECH*, 2016, pp. 2963–2967.

[10] Brian King, I-Fan Chen, Yonatan Vaizman, Yuzong Liu, Roland Maas, Sree Hari Krishnan Parthasarathi, and Björn Hoffmeister, "Robust speech recognition via anchor word representations," in *INTERSPEECH*, 2017, pp. 2471–2475.

[11] Marc Delcroix, Katerina Zmolikova, Keisuke Kinoshita, Atsunori Ogawa, and Tomohiro Nakatani, "Single channel target speaker extraction and recognition with speaker beam," in *Acoustics, Speech and Signal Processing (ICASSP), 2018 IEEE International Conference on*. IEEE, 2018, pp. 5554–5558.

[12] Jun Wang, Jie Chen, Dan Su, Lianwu Chen, Meng Yu, Yanmin Qian, and Dong Yu, "Deep extractor network for target speaker recovery from single channel speech mixtures," in *INTERSPEECH*, 2018, pp. 307–311.

[13] Alex Graves, "Sequence transduction with recurrent neural networks," *CoRR*, vol. abs/1211.3711, 2012.

[14] Alex Graves and Navdeep Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *International Conference on Machine Learning*, 2014, pp. 1764–1772.

[15] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio, "Attention-based models for speech recognition," in *Advances in neural information processing systems*, 2015, pp. 577–585.

[16] William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 4960–4964.

[17] Chung-Cheng Chiu, Tara N Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J Weiss, Kanishka Rao, Ekaterina Gonina, et al., "State-of-the-art speech recognition with sequence-to-sequence models," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4774–4778.

[18] Zhehuai Chen, Qi Liu, Hao Li, and Kai Yu, "On modular training of neural acoustics-to-word model for lvcsr," in *Acoustics, Speech and Signal Processing (ICASSP), 2018 IEEE International Conference on*. IEEE, 2018, pp. 4754–4758.

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

[20] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 369–376.

[21] Zhehuai Chen, Jasha Droppo, Jinyu Li, and Wayne Xiong, "Progressive joint modeling in unsupervised single-channel overlapped speech recognition," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 26, no. 1, pp. 184–196, 2018.

[22] Won-Ho Shin, Byoung-Soo Lee, Yun-Keun Lee, and Jong-Seok Lee, "Speech/non-speech classification using multiple features for robust endpoint detection," in *Acoustics, Speech, and Signal Processing (ICASSP), 2000 IEEE International Conference on*. IEEE, 2000, vol. 3, pp. 1399–1402.

[23] Xin Li, Huaping Liu, Yu Zheng, and Bolin Xu, "Robust speech endpoint detection based on improved adaptive band-partitioning spectral entropy," in *International Conference on Life System Modeling and Simulation*. Springer, 2007, pp. 36–45.

[24] Matt Shannon, Gabor Simko, and Carolina Parada, "Improved end-of-query detection for streaming speech recognition," in *INTERSPEECH*, 2017, pp. 1909–1913.

[25] Roland Maas, Ariya Rastrow, Chengyuan Ma, Guitang Lan, Kyle Goehner, Gautam Tiwari, Shaun Joseph, and Björn Hoffmeister, "Combining acoustic embeddings and decoding features for end-of-utterance detection in real-time far-field speech recognition systems," in *Acoustics, Speech and Signal Processing (ICASSP), 2018 IEEE International Conference on*. IEEE, 2018, pp. 5544–5548.

[26] Oleksii Kuchaiev, Boris Ginsburg, Igor Gitman, Vitaly Lavrukhin, Jason Li, Huyen Nguyen, Carl Case, and Paulius Micikevicius, "Mixed-precision training for nlp and speech recognition with openseq2seq," 2018.

[27] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

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

[29] Zhehuai Chen, Mahaveer Jain, Yongqiang Wang, Michael Seltzer, and Christian Fuegen, "End-to-end contextual spebech recognition using class language models and a token passing decoder," in *Acoustics, Speech and Signal Processing (ICASSP), 2019 IEEE International Conference on*. IEEE, 2019 (to appear).