ASR techniques

Jianbo Ma

December 2021

1 Introduction

As the proliferation of deep learning, development of Automatic Speech Recognition (ASR) and research in this area, the field of ASR experienced dramatic progresses during the last two decades. Techniques to attack problems encountered in ASR are continuously emerge, and resources including data, evaluation metrics, tool-kits become ubiquitous for researchers and developers. All of these factors are advancing the research and application of ASR, making it one of the hottest area of artificial intelligence (AI). In this document, we are trying to have a thorough review of the current techniques of ASR. But we are not going to put much effort on the historical development of ASR. We are going to mainly focus on different techniques and what are the strength and weakness of different branches and what are the best usage situation.

2 Different branches of ASR system

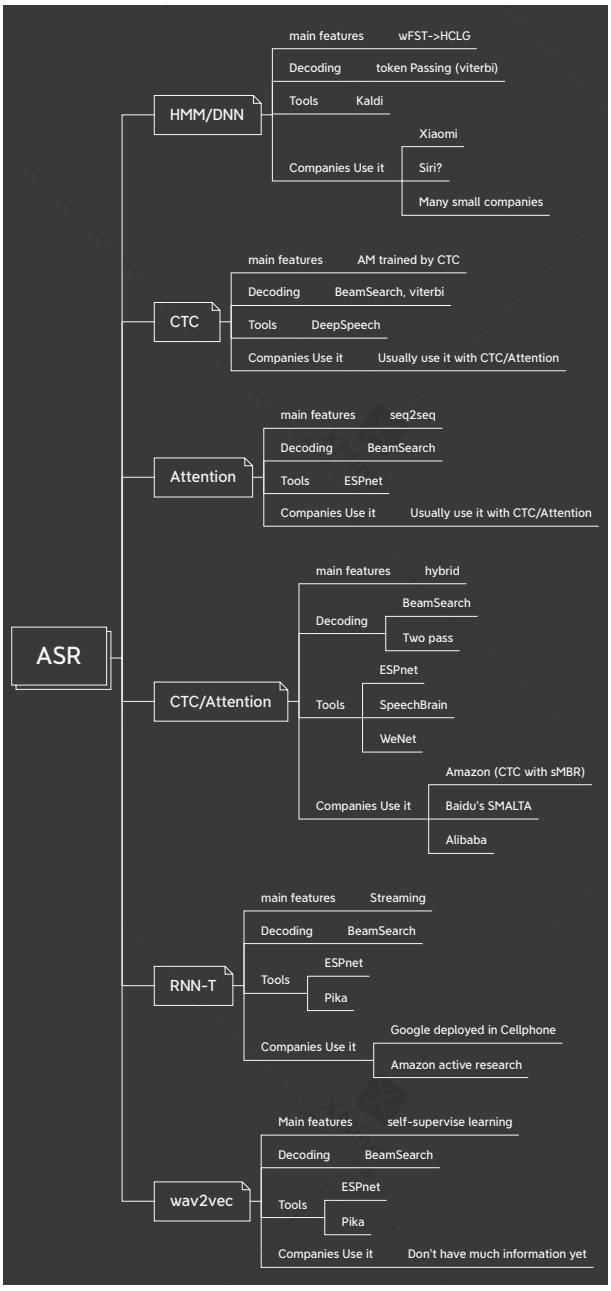

Figure 1: ASR techniques.

There are six branches of ASR system if we broadly split it. They are listed as follow. Below we will go through them one by one.

- HMM/DNN
- CTC
- Attention-based Encoder-Decoder
- CTC/Attention-based Encoder-Decoder
- RNN-T
- Self-supervised Representation Learning

2.1 HMM/DNN

Conventional speech recognition system usually use Gaussian mixture model (GMM) as the acoustic model, with Hidden Markov models to model the sequence [1]. With the popularity of deep learning, DNN based acoustic model then replaced the GMM to model the emission probability in HMM/GMM based model [2], which is the well-know HMM/DNN model. To build a HMM/DNN speech recognition system, there are several steps. The GMM based acoustic models (mono phone model, triphone model) will be firstly trained to align different frames to modeling unit (e.g. phone). Then frame alignments are extracted by the pre-trained GMM aligner by using force alignment of viterbi decoding. Based on those frame alignments, a DNN model can be trained by using cross-entropy loss.

Fig 2 is a simple diagram of HMM/DNN in kaldi [3]. The essence of speech recognition in kaldi can be represented as HCLG, in which H stands for HMM transition-id to context dependent phone, C represents the context-dependency, L is the lexicon and G encoders the grammar or language model. This decoder will create a search graph and viterbi algorithm with token passing implementation is used for decoding.

This system is still wildly used in many companies with good implementation support from kaldi toolkit.

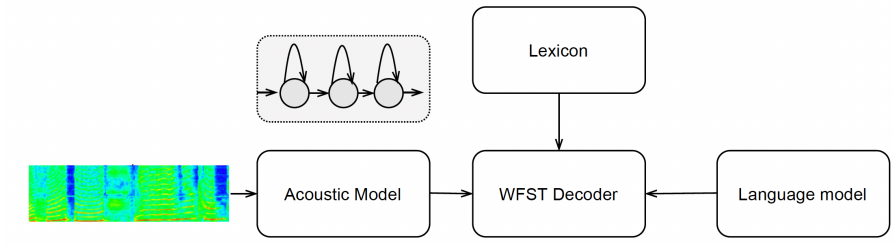

Figure 2: Architecture of HMM/DNN model.

2.2 CTC

Connectionist temporal classification (CTC) was first proposed by Graves [4] in 2006. Conventionally in ASR, the HMMs were used to model sequence and map input audio frame sequence to label sequences. HMMs are generative models and usually Expectation-Maximization algorithms are used to update parameters. The CTC is also starting from the independent assumption of each frame by observing a particular label (this is the so called conditional independent assumption), so that the probability of observing a particular label at a specific time frame will can be collected in product manner to form the probability of observing a label sequence. By introducing a special symbol blank, CTC then can present the case that there is no label output for the given time frame. The objective function of CTC is then derived from maximum likelihood and gradient-based optimization can be used to update parameters of neural networks. Specifically, dynamic programming algorithm can be used to calculate the conditional probabilities of different valid label sequence, similar with HMMs, which uses forward-backward algorithm [1].

Fig 3 is to show how the forward operation is done to summarize the conditional probabilities of different 'path'. We define the 'path' as a particular label sequence in the trellis like fig 3. The idea is similar with forward-backward algorithm of HMMs. For example, in the forward operation, all the probabilities of possible valid path are summarized before entering a new time frame. This is guaranteed by summarizing all possible valid step from time t to $t+1$. As the monotonic property of speech or handwriting, this is possible to be exhausted. For example, in fig 3 from state 'C' of time $t = 2$, there are only two possible advancing step from $t = 1$, one from blank and another one from state 'C' at time $t = 1$. This advancing recursion is then formulated as equation (6) in [4], in which essential means the transition probability from one state to another is modeled as the same.

Finally, the summation of all the probabilities that the corresponding path passed a particular state K at time t will be able to modeled by forward and backward operators (α and β in [4]). This makes calculating the gradient of the loss against this particular state at time t possible and essentially formulated as equation (16) in [4].

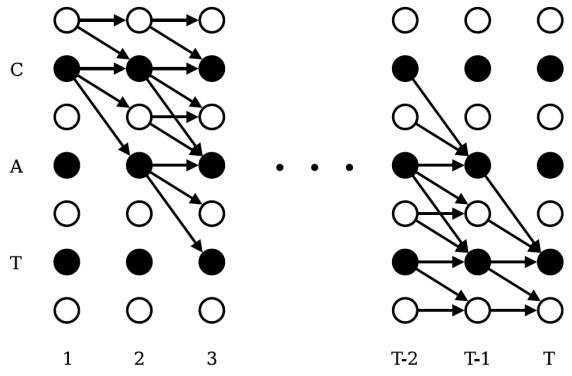

Figure 3: Illustration of forward operation in CTC (from Graves etc. CTC paper).

The CTC then can be used to model directly the grapheme output of audio speech. This means we do not have to use pronunciation unit as the intermediate state and then map the pronunciation unit into a word sequence, which is the case in conventional HMM/DNN or HMM/GMM ASR system. Of course there are some other streams to model ASR as end-to-end fashion, such as the Attention model [5] [6]. The work of deepspeech [7] shows that it can have state-of-the-art accuracy by using CTC.

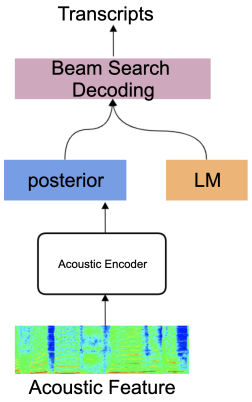

Figure 4: Basic system structure of CTC end-to-end ASR.

The whole ASR is then can be presented by fig 4. There are mainly three components and they are acoustic model (AM), language model (LM), and beam search decoder. The beam search decoder is not a model and is the prefix beam search algorithm that will combine the posterior probability from AM and text sequence probability from LM together to give the top-K candidate

label sequences for a given audio segments. Usually we will choose the top-1 label sequence as the final transcript of the given audio segment. But the top-K candidates may be used for other modules such as correction module for further analysis.

2.3 RNN-T

The Recurrent Neural Network (RNN) Transducer (RNN-T) is proposed by Graves [8] as a subsequent work of connectionist temporal classfier (CTC) [4]

In fig 3, we can see that for each state at time t, there are two or three options for the next time step $t+1$ depending on if it is on the blank state or not. They are staying at the same state (self-loop), emitting to the blank or emitting to the next state (e.g. state 'C' to 'A') that is defined in the transcript. The probabilities for those transitions are equal and does not depend on the previous state. The emitting probability of state 'C' at time t is only determined by the acoustic feature and model parameters. This is also corresponding to the conditional independent property of CTC. In other words, the CTC models the input-output dependencies. While RNN-T is remedy the assumption by modeling both input-output and output-output dependencies [8].

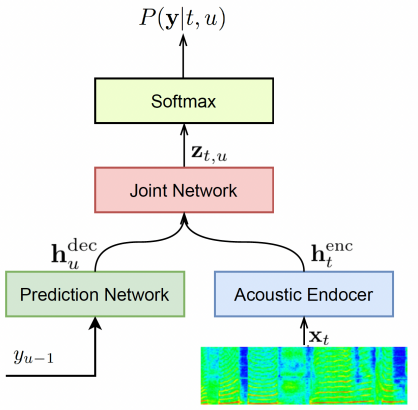

Figure 5: RNN-T Structure (inspired from Kanishka etc. [9] paper.)

The RNN-T accomplishes the joint modeling by introducing a network structure depicted in fig 5. Apart from the AM, there are two other components. The prediction network served analogue to a LM to generate the output-output distribution that is the next state conditioned on the previous states (using internal states in the RNN). The joint network is then fed by both acoustic and textual information and generate a combined vector $\mathbf{z}_{t,u}$ before the softmax layer. This will generate a different lattice compared with CTC (fig 3) as shown in fig 6.

Before illustrating fig 6, we need to define a special symbol used in RNN-T. The *null* output $\emptyset$. This intuitive meaning of $\emptyset$ is 'output nothing' [8]. This

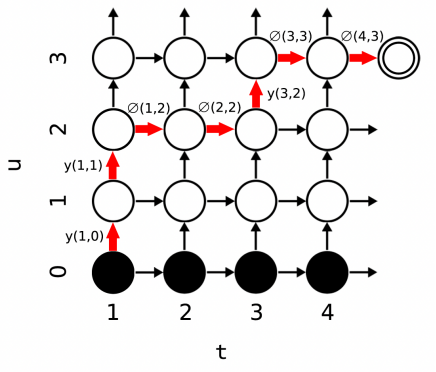

Figure 6: RNN-T probability lattice (from Graves etc. [8] paper.)

symbol is needed to determine whether entering the next time step. In fig 6, it starts from *null* state and them emits to the next state in time $t = 1$ until it reached the *null* state again. When it reached the *null* state, advancing to $t = 2$ is accomplished by feeding the next frame. Then the loop continues until reach the last frame. Similar with CTC, the objective of RNN-T is to maximize likelihood of all valid path. This is again can be accomplished by forward-backward algorithm.

The research topic of RNN-T attracts extensive attention as the natural streaming property [10], good performance and low latency [11]. There are many literature to deal with the training of RNN-T. For example, the paper of google [9] is very useful when training a RNN-T model. A two pass decoding structure has proposed in [12] to combine the output of RNN-T decoder and LAS decoder to utilize both RNN-T and LAS. The work in [13] also use decoder of AED and CTC to have a second decoding. It can be used as a keyword spotting system as well [14]. Google has implemented the RNN-T model on-device (available on Google AI blog).

2.4 Attention-based Encoder-Decoder

Attention-based encoder-decoder (AED) end-to-end speech recognition is similar with RNN-T in the sense that the decoded output (labels) will also be used to predict the next label. It was first proposed in [15] [5] and it was first. evaluated in the Phoneme recognition. The LAS system [6] is another a well-known attention-based encoder-decoder speech recognition system that is similar with the sequence-to-sequence model [16].

The architecture of LAS is depicted as fig 7. The component of Listener served as an encoder that maps the acoustic feature into a encoder memory in hidden space. Then there is an attend and speller to use the encoder memory and pre-obtained the output label to predict the next label. This behaviours are mathematically modeled by equation (6)-(8) in [6]. The entire model is trained

by maximize the summed log probability represented by equation (12) in [6]. The idea of the objective function is that we will use the previous step group truth label and acoustic feature to model the distribution of the next label.

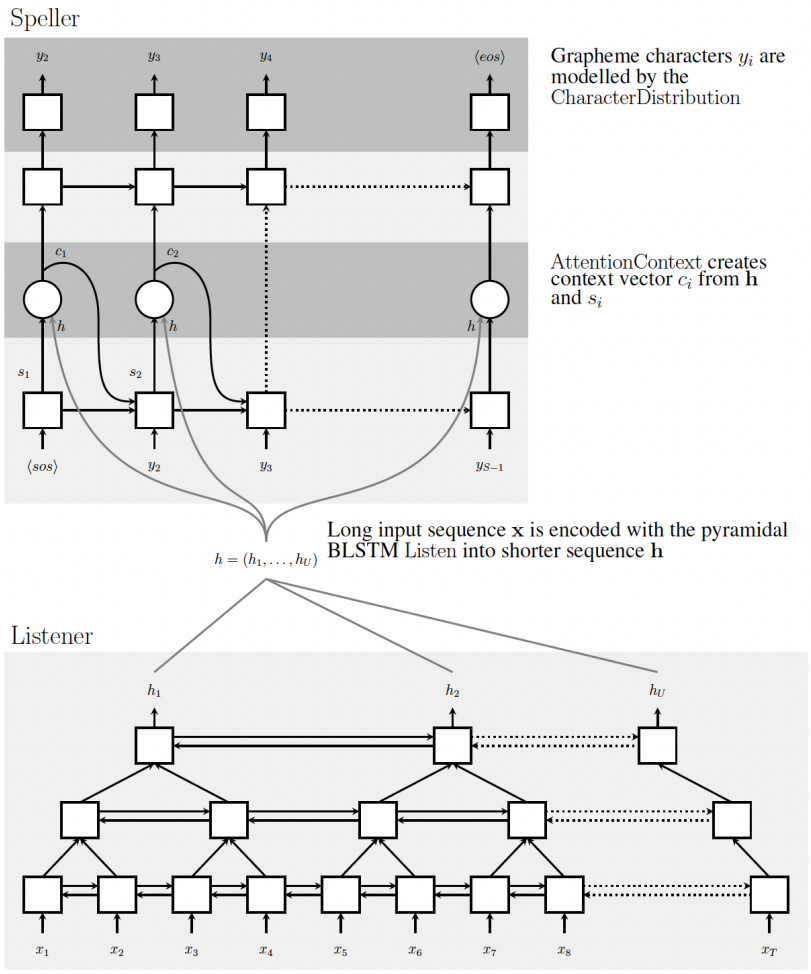

Figure 7: LAS system architecture (copied from [6]

Fig 8 is simpler version of the attention-based encoder-decoder speech recognition system. It zooms in the decoder and we can see that the previous label will be firstly transformed to a embedding using a lookup table which is trainable. The text embedding will be used in the attention mechanism to attend the memory and model the distribution of the next label. It should be noted that, in the context of LAS and attention-based model, all the acoustic features are expected to be seen when decoding performed, which means it is a offline model. That's why there are a lot of efforts to make the AED streaming, like

hard monotonic attention [17], Monotonic Chunkwise Attention (MoCHA) [18], trigger attention [19] or adding a scout net to predict the region for attention [20]. All those effort to make it streaming are applicable for the CTC/AED system in the next section.

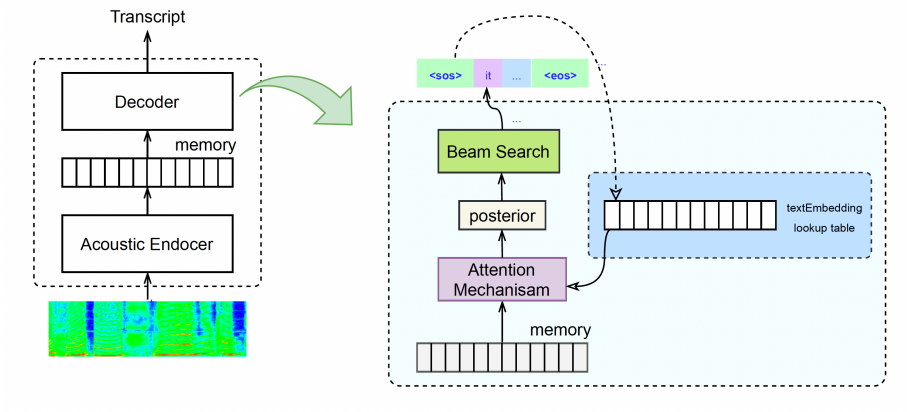

Figure 8: Attention ASR structure.

2.5 CTC/Attention-based Encoder-Decoder

In previous section, it can be seen that the AED does not have frame alignment, while CTC has the ability to align frames to different units. The idea that combine both CTC-based architecture and AED is then explored in [21]. The architecture of this work is depicted in fig 9, in which two different branches are used one for CTC and another one for Attention decoder. This is a multi-task learning. In decoding phase, both posterior probabilities from CTC and attention-decoder will be used. The work for making the attention model streaming in previous section can be also applied here.

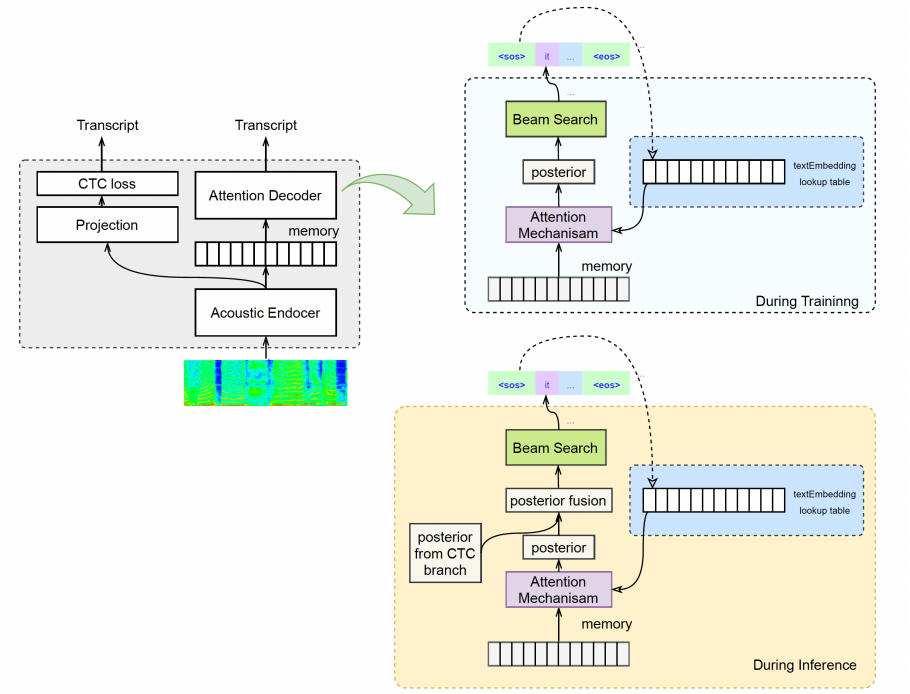

Figure 9: CTC/Attention ASR structure.

2.6 Self-supervised Representation Learning

Self-supervised learning (SSL) has achieved great success in field such as natural language processing (NLP), e.g. BERT [22]. In computer vision, there are also a lot of literature to conduct research on this topic. This trend is also true in speech processing task. The self-supervised learning is of great desire as it have the ability to boost performance for different down-streaming tasks using small amount labeled data and effort. This has been shown in many literature, like wav2vewav2vec [23] and its successors wave2vec2 [24], wav2vec-U [25] and HuBERT [26]. These works extracted a lot of attention in the speech processing community. The work of SUPERB [27] is a great effort for the benchmark of the self-supervised learning in speech processing.

We need to give enough attention to this trend. The ability of SRL will be beneficial for many task in speech processing.

References

[1] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[2] 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.

[3] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. CONF. IEEE Signal Processing Society, 2011.

[4] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 369–376.

[5] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," *arXiv preprint arXiv:1506.07503*, 2015.

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

[7] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates *et al.*, "Deep speech: Scaling up end-to-end speech recognition," *arXiv preprint arXiv:1412.5567*, 2014.

[8] A. Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.

[9] K. Rao, H. Sak, and R. Prabhavalkar, "Exploring architectures, data and units for streaming end-to-end speech recognition with rnn-transducer," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2017, pp. 193–199.

[10] Q. Zhang, H. Lu, H. Sak, A. Tripathi, E. McDermott, S. Koo, and S. Kumar, "Transformer transducer: A streamable speech recognition model with transformer encoders and rnn-t loss," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7829–7833.

[11] X. Chen, Y. Wu, Z. Wang, S. Liu, and J. Li, "Developing real-time streaming transformer transducer for speech recognition on large-scale dataset," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5904–5908.

[12] T. N. Sainath, R. Pang, D. Rybach, Y. He, R. Prabhavalkar, W. Li, M. Visontai, Q. Liang, T. Strohman, Y. Wu *et al.*, "Two-pass end-to-end speech recognition," *arXiv preprint arXiv:1908.10992*, 2019.

[13] B. Zhang, D. Wu, Z. Yao, X. Wang, F. Yu, C. Yang, L. Guo, Y. Hu, L. Xie, and X. Lei, "Unified streaming and non-streaming two-pass end-to-end model for speech recognition," *arXiv preprint arXiv:2012.05481*, 2020.

[14] Y. Tian, H. Yao, M. Cai, Y. Liu, and Z. Ma, "Improving rnn transducer modeling for small-footprint keyword spotting," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5624–5628.

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

[16] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[17] S. Wu and R. Cotterell, "Exact hard monotonic attention for character-level transduction," *arXiv preprint arXiv:1905.06319*, 2019.

[18] C.-C. Chiu and C. Raffel, "Monotonic chunkwise attention," *arXiv preprint arXiv:1712.05382*, 2017.

[19] N. Moritz, T. Hori, and J. Le Roux, "Triggered attention for end-to-end speech recognition," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5666–5670.

[20] C. Wang, Y. Wu, S. Liu, J. Li, L. Lu, G. Ye, and M. Zhou, "Low latency end-to-end streaming speech recognition with a scout network," *arXiv preprint arXiv:2003.10369*, 2020.

[21] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, "Hybrid ctc/attention architecture for end-to-end speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240–1253, 2017.

[22] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[23] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," *arXiv preprint arXiv:1904.05862*, 2019.

[24] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *arXiv preprint arXiv:2006.11477*, 2020.

[25] A. Baevski, W.-N. Hsu, A. Conneau, and M. Auli, "Unsupervised speech recognition," *arXiv preprint arXiv:2105.11084*, 2021.

[26] 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," *arXiv preprint arXiv:2106.07447*, 2021.

[27] S.-w. Yang, P.-H. Chi, Y.-S. Chuang, C.-I. J. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G.-T. Lin *et al.*, "Superb: Speech processing universal performance benchmark," *arXiv preprint arXiv:2105.01051*, 2021.