ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-6327-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICASSP49357.2023.10095838

A COMPARISON OF SEMI-SUPERVISED LEARNING TECHNIQUES FOR STREAMING ASR AT SCALE

Cal Peyser[1,2]*, Michael Picheny*[1]*, Kyunghyun Cho*[1]*,*
Rohit Prabhavalkar[2]*, W. Ronny Huang*[2]*, Tara N. Sainath*[2]

[1]New York University, Center for Data Science, [2]Google Inc.
cpeyser@google.com

ABSTRACT

Unpaired text and audio injection have emerged as dominant methods for improving ASR performance in the absence of a large labeled corpus. However, little guidance exists on deploying these methods to improve production ASR systems that are trained on very large supervised corpora and with realistic requirements like a constrained model size and CPU budget, streaming capability, and a rich lattice for rescoring and for downstream NLU tasks. In this work, we compare three state-of-the-art semi-supervised methods encompassing both unpaired text and audio as well as several of their combinations in a controlled setting using joint training. We find that in our setting these methods offer many improvements beyond raw WER, including substantial gains in tail-word WER, decoder computation during inference, and lattice density.

1. INTRODUCTION

Methods for learning from large-scale supervised datasets have been the primary driver of progress in speech processing from the HMM/GMM era [1] well into the era of deep learning [2, 3, 4]. However, as the scope of ASR research has expanded into challenging settings such as low-resource languages and difficult acoustic conditions, it has become difficult to gather large-scale in-domain supervised datasets [5]. In recent years, much of the speech community's attention has moved to alternatives to purely supervised learning.

Semi-supervised learning has emerged as a powerful paradigm for addressing contemporary problems in speech recognition [6]. In a semi-supervised training scheme, unpaired speech and/or text examples supplement a supervised dataset to provide greater acoustic/language coverage. A broad literature has emerged exploring various mechanisms for incorporating speech-only and text-only data into ASR training (see Section 2).

Semi-supervised learning with both audio and text has yielded very strong results on benchmark ASR tasks, motivating interest in these methods for large-scale, production applications. However, published results generally report on datasets smaller than industrial-scale, usually emphasizing the low-resource case in which very little supervised data is available. Furthermore, they use large full-context architectures that do not meet realistic requirements for modern production ASR systems such as being small enough to fit on a mobile phone and capable of streaming predictions. Finally, the literature reports almost entirely on WER improvements, with little study of measures like CPU load and lattice richness that are applicable when an ASR system acts as an individual component of an on-device system.

In this work, we provide a comparison of several leading semi-supervised methods in a controlled setting geared towards production implementation. Unlike previous work, we apply these methods to a state-of-the-art, 160M-parameter streaming Conformer [7] model that is already trained on a very large supervised corpus. We further depart from previous work by training supervised and unsupervised tasks jointly, which is being increasingly shown to be preferable to the conventional fine-tuning approach on very large datasets [8]. We find that under these conditions, none of the studied methods improve general WER at all. However, we report improvements in the decoder's computational load and in lattice density, as well as in several targeted WER measurements assessing performance on known categories of particularly difficult utterances. Through this comparison and analysis, we hope to offer a more nuanced and comprehensive view of the usefulness of unpaired audio and text in industrial ASR.

The rest of our paper is structured as follows. Section 2 summarizes the literature on the three methods under study. Section 3 presents our architecture for a streaming ASR system that supports these three methods and their combinations. Section 4 details our datasets, experiments and evaluation criteria. Section 5 presents our results, and Section 6 concludes.

2. RELATED WORK

In this section, we summarize the literature surrounding the three semi-supervised learning methods that we study in this work.

2.1. Text Injection

Unsupervised text injection in ASR is traditionally done with language model "fusion", either at inference time [9] or training time [10, 11]. These methods involve the explicit separation of the model parameters into an acoustic model trained on paired data and a language model trained on unpaired text. The improvements yielded by these methods come at the cost of the additional language model parameters at inference time.

A simultaneous line of work has sought an alternative to fusion in which unsupervised text is used to train an acoustic model directly. One major line of work focuses on creating pseudolabels for unpaired text through synthesized audio. This has been studied by generating a raw audio signal [12] or higher level lexical features [13]. Work adapting cycle consistency losses from machine translation have trained ASR and TTS together with a fully end-to-end objective [14, 15]. We choose TTS-based augmentation as the first method to study in this work (see Section 3.2.1).

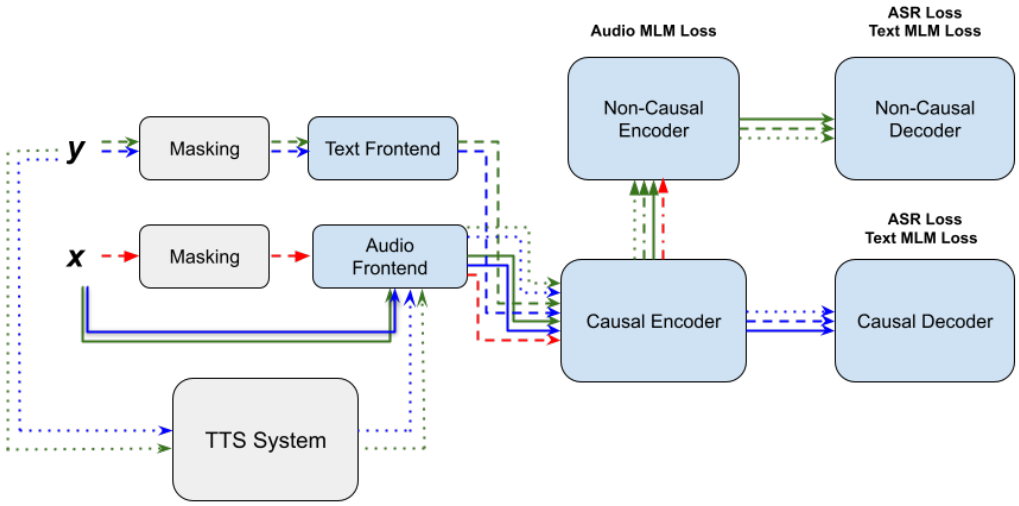

Fig. 1. High-level model architecture.

Finally, a third class of methods for unpaired text injection makes use of auxiliary, text only objectives to train an ASR encoder without generating TTS pseduolabels. Most such works have sought to train an ASR encoder to agnostically represent either audio or text, such that unpaired text is processed similarly to audio [16, 17, 18]. JOIST [19] is a recent method which does this using a masked language modeling task in the spirit of BERT [20]. We study JOIST in this work since it is one of the few methods that has been shown to work well together with very large supervised datasets and with on-device sized streaming models (see Section 3.2.2).

2.2. Audio Injection

Unsupervised audio injection is very well studied and has yielded a large literature [21]. Recent work has largely built of the success of the Wav2Vec series of models [22, 23], which work by modeling masked segments of audio using a contrastive loss. One line of further work investigated audio clustering to generate targets for the contrastive loss [24, 25] while another investigated methods for computing that signal by quantizing the audio inputs [26]. BEST-RQ [27] in particular finds that fixed random projection to a pre-initialized codebook works effectively as a quantizer. We choose BEST-RQ as the third method to study in this work (see Section 3.2.3).

3. METHODS

In this section, we frame the problem of semi-supervised speech recognition, develop our model architecture, and specify the multi-task optimization problem that it is trained for.

3.1. Architecture

We are interested in the setting in which unsupervised data in both the speech and text domains is available alongside a large supervised corpus. We denote as $(x, y) \in \mathcal{S}$ the supervised pair of a speech utterance x and text label y in the supervised dataset $\mathcal{S}$. We similarly denote unsupervised speech examples as $x \in \mathcal{U}^S$ and unsupervised text examples $y \in \mathcal{U}^T$.

We extend the cascading conformer proposed in [28] to support semi-supervised multitask training. To this end, we define four neural modules:

1. E_C, the "causal" encoder, which consumes streamed audio features with no right-context.
2. E_{NC}, the "non-causal" encoder, which consumes the outputs of E_C with 900ms of right-context.
3. D_C, a decoder for the causal encoder. During inference, this decoder may be used to generate immediate predictions as the user speaks.
4. D_{NC}, a decoder for the non-causal encoder. During inference, this decoder may be used to revise the predictions of the causal decoder with short latency.

Unlike [28], we would like our model to consume representations of either audio or text. For this we follow JOIST, seeking mechanisms to cause the E_C to be agnostic to the input domain. We choose to include two neural "frontends", one for audio features and one for text. As in JOIST, we upsample text frontend outputs by repetition so that audio and text representations will be of approximately the same length.

3.2. Tasks

In this framework, causal and non-causal ASR are trained as they are in [28]. In particular, for causal ASR, x is processed by the audio frontend, encoded by E_C, and decoded by D_C, while non-causal ASR is processed analogously with the non-causal modules E_{NC} and D_{NC}. The model is trained end-to-end with an RNN-T [2] loss. This is represented by the solid blue (causal) and solid green (non-causal) paths in Figure 1. For semi-supervised tasks we require different formulations.

3.2.1. TTS Augmentation

Using a pre-trained TTS system with frozen parameters, we generate an audio clip $\hat{x}$ corresponding to each unsupervised text segment $y \in \mathcal{U}^T$. We then treat $(\hat{x}, y)$ as a supervised audio-text pair and train the causal and non-causal ASR tasks. This is represented by the dotted blue (causal) and dotted green (causal) paths in Figure 1.

We found that in order to achieve reasonable training speed it is important that the TTS system convert input word-pieces not into raw audio but instead into the (much shorter) sequence of acoustic features that is consumed by the audio frontend. This is due to the fact that since the decoder of our TTS system which produces audio features is autoregressive, audio sequence length has critical implications for training speed and quickly becomes a bottleneck.

3.2.2. JOIST

Following the design of JOIST in [19], we pass masked unpaired text examples through a text frontend, which consists simply of a learned projection. The results are treated identically to audio features; that is, they are passed in turn to the causal (E_C and D_C) and non-causal (E_{NC} and D_{NC}) and compared to original text sequence via an RNN-T loss. This is represented by the dashed blue (causal) and dashed green (causal) paths in Figure 1.

We find that it is critical for WER that JOIST consume phonemic representations of y, as opposed to text tokens, corroborating the findings of [19]. We include a text-to-phonemes lookup in the model which processes text before masking. The JOIST loss still operates with respect to the standard word-piece representation - that is, the JOIST loss learns to generate word pieces from a masked phoneme sequence.

3.2.3. BEST-RQ

We model our audio injection after BEST-RQ as implemented in [27]. Audio features are masked and processed by the frontend. They are then encoded by the casual and non-causal encoders of the ASR stack. Additionally, audio features are processed by a randomly initialized projection with frozen weights and then discretized by rounding to the nearest entry in a fixed codebook. The encoder is then trained to predict the quantized targets inside the masked region. This is represented by the dashed red path in Figure 1.

3.3. Training Scheme

There are many approaches to multi-task semi-supervised learning, mostly focused on pretrain-finetune paradigm [22, 23, 26, 25]. While this methodology has achieved state of the art results on datasets such as Librispeech, we found that on our large dataset it is prone to forgetting representations learned in pretraining during finetuning, which is consistent with the findings in [8] for very large training sets. We therefore restrict our study to joint training of ASR together with the unsupervised tasks. Note that even though joint training includes ASR, we find that it is still beneficial and convenient to initialize from a strong ASR baseline.

At each iteration during training we sample a separate batch from each dataset, $b_{\mathcal{S}} \in \mathcal{S}$, $b_{\mathcal{U}^S} \in \mathcal{U}^S$, and $b_{\mathcal{U}^T} \in \mathcal{U}^T$. We then propagate each batch through the model, performing the preprocessing specified for TTS augmentation and JOIST on $b_{\mathcal{U}^T}$ and that specified for BEST-RQ on $b_{\mathcal{U}^S}$. We apply the relevant losses to each task and sum them according to specified weights.

4. EXPERIMENTS

This section details the implementation, training, and evaluation of the architecture described above.

4.1. Model

Following the components in Figure 1 the architecture of our model is as follows.

The causal audio encoder E_C consists of six conformer [7] layers with model dimension 2048 and eight attention heads. The non-causal audio encoder E_{NC} adds a further nine such conformer layers. The decoders D_C and D_{NC} are each HAT [11] decoders with prediction and joint networks with model dimension 640. These four components and the audio frontend, which together make up the inference-time model, contain about 164M parameters.

The TTS system is based on Tacotron 2 [29]. The encoder consists of three convolutions followed by a single RNN layer, while the decoder consists of a single RNN layer with attention to the encoder outputs followed by a post-net consisting of five convolutional layers.

4.2. Training

We train our model with a supervised dataset $\mathcal{S}$ consisting of about 4M utterances, totalling about 200k hours of speech. We also use an unsupervised audio set $\mathcal{U}^S$ of about 600M utterances and an unsupervised text set $\mathcal{U}^T$ of about 230B examples.

At timestep t, the audio head of our model consumes 512-dimensional features consisting of four 128-dimensional log-mel features representing the range $[t-2, t+1]$. The log-mel features are computed at 10ms intervals and on 32ms frames. We subsample stacked features by a factor of 3, so that each feature represents 30ms in the input. During BEST-RQ, we a mask single span consisting of 15% of the input features. Text inputs are represented by a wordpiece model of size 4096.

Our baseline model is trained for 800k steps with a batch size of 2048 for each of $\mathcal{S}$, $\mathcal{U}^S$, and $\mathcal{U}^T$. Our semi-supervised experiments are trained for a further 35k steps, using task splits detailed in Section 5.

4.3. Evaluation

We evaluate our models on several test sets, seeking to measure performance under the acoustic and language conditions which are typically targeted using unsupervised data. Our voice search test set (**VS**) is sampled from anonymized traffic to Google production services. The **NOISY** set consists of anonymized traffic with artificial noise added. Our remaining test sets are synthesized using a TTS system from anonymized text traffic to Google services, and are selected according to a criterion meant to target difficult language conditions. The rare proper nouns set (**RPN**) consists of examples that contain a proper noun (as determined by a neural proper noun tagger) that occurs fewer than five times in $\mathcal{S}$. The Rare-LM set (**R_LM**) consists of examples containing a unigram that occurs fewer than five times in both $\mathcal{S}$ and $\mathcal{U}^T$, while the (**C_LM**) consists of examples containing a unigram that occurs fewer than five times in $\mathcal{S}$ but at least 150 times in $\mathcal{U}^T$. **RPN** and **C_LM** are measure tail performance, while **C_LM** is intended to measure the degree to which information from $\mathcal{U}^T$ has been incorporated into the model.

Model	VS	Noisy	RPN	R_LM	C_LM
E-0	162	187	297	357	325
E-A	-7.2%	-5.3%	**-11.1%**	**-10.1%**	**-8.9%**
E-B	-7.2%	-5.3%	-9.8%	-8.4%	-7.4%
E-C	**-9.9%**	**-6.9%**	-6.3%	-5.8%	-4.9%
E-AB	-7.2%	-4.8%	-6.1%	-5.0%	-4.0%
E-AC	**-9.9%**	-6.4%	-10.8%	-9.2%	-8.3%
E-ABC	-8.5%	-5.9%	-9.8%	-8.7%	-7.7%

Table 1. Average Decoding States

Model	VS	Noisy	RPN	R_LM	C_LM
E-0	3.2	3.3	6.2	8.1	9.7
E-A	**+12.5%**	**+15.2%**	**+3.2%**	+3.7%	+3.1%
E-B	**+12.5%**	+12.1%	**+3.2%**	+3.7%	+3.1%
E-C	**+12.5%**	+12.1%	**+3.2%**	+3.7%	+3.1%
E-AB	**+12.5%**	+12.1%	+1.6%	+3.7%	+3.1%
E-AC	**+12.5%**	+12.1%	**+3.2%**	**+4.9%**	**+4.1%**
E-ABC	**+12.5%**	+12.1%	**+3.2%**	**+4.9%**	+3.1%

Table 2. Lattice Density

5. RESULTS

We denote JOIST with the letter **A**, TTS augmentation with **B**, and BEST-RQ with **C**. We find the best results when each of these experiments are trained with 40% task weighting each on causual and non-causal ASR, with the remaining 20% split across unsupervised tasks. The weightings of the unsupervised tasks are given in Table 3.

Model	C-JOIST	NC-JOIST	TTS	BEST-RQ
E-A	1/2	1/2	0	0
E-B	0	0	1	0
E-C	0	0	0	1
E-AB	1/4	1/4	1/2	0
E-AC	1/4	1/4	0	1/2
E-ABC	1/6	1/6	1/3	1/3

Table 3. Task Weights. "C-JOIST" and "NC-JOIST" refer to the causal and non-causal variants.

We denote our baseline experiment **E-0**, which splits its weight equally between causal and non-causal supervised ASR.

We give our WER results in Table 4. We are unsurprised to find that given a very large supervised corpus and limited model capacity, none of our methods improve performance on the unspecialized voice search test set. We find considerable improvement, however, under tail conditions. JOIST consistently performs best on the acoustically clean but linguistically difficult TTS tail-word test sets, which agrees with the intuition that JOIST acts to improve the encoder's text representation. However, JOIST in fact degrades performance on the acoustically challenging Noisy test set. BEST-RQ seems beneficial only when combined with JOIST, where it appears to recover lost performance on noisy data while retain some of the improvements on the tail-word sets.

Model	VS	Noisy	RPN	R_LM	C_LM
E-0	6.0	8.2	21.2	38.3	55.8
E-A	**-0.0%**	+1.2%	**-4.7%**	**-5.0%**	**-2.3%**
E-B	**-0.0%**	**-1.2%**	-0.5%	-2.1%	-0.7%
E-C	**-0.0%**	+1.2%	+0.1%	-0.0%	-0.4%
E-AB	**-0.0%**	+1.2%	-3.8%	-4.2%	-2.0%
E-AC	**-0.0%**	-0.0%	-2.8%	-2.9%	-1.2%
E-ABC	**-0.0%**	+2.4%	-3.3%	-3.7%	-1.4%

Table 4. Word Error Rate

In production systems, model performance goes beyond raw WER, since it is often not a 1-best hypothesis but rather the produced lattice that is used to generate predictions or fed directly to a downstream NLU task. In Table 2, we measure the richness of the lattice by computing "lattice density", which we define as the number of arcs in the lattice divided by the number of wordpieces in the ground truth. On this measure, we find that all three methods offer considerable improvement in voice search. For difficult utterances, we find that combinations of methods largely outperform single methods. This agrees with the intuition that many training criteria lead to a greater diversity of plausible predictions, and invites investigation into the combination of these methods for applications like biasing or intent classification which can benefit from a rich lattice.

Finally, since an autoregressive decoder is often a computational bottleneck in on-device systems, we seek to determine the impact of our methods on the work the decoder has to do. In Table 1, we measure the average number of states expanded by the decoder during beam search. We find that all three methods provide meaningful improvements over the baseline on this metric, with the best results coming from JOIST. This suggests, unsurprisingly, that the decoder explores the fewest states when the encoder has a strong language representation.

6. CONCLUSIONS

In this work we apply several contemporary semi-supervised training methods to a realistic, state-of-the-art production ASR system. We find that unlike in the conventional setting, with a large full-context model and only a small amount of supervised data, these methods do not offer improvement on unspecialized WER. We demonstrate, however, that these techniques nevertheless offer meaningful utility for tail-condition performance, lattice density, and decoder computational load. We believe that these results motivate a broader perspective on semi-supervised training in its application to industrial ASR.

7. REFERENCES

[1] Geoffrey Hinton, Li Deng, Dong Yu, George E. Dahl, Abdelrahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N. Sainath, and Brian Kingsbury, "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.

[2] Alex Graves, "Sequence transduction with recurrent neural networks," in *International Conference on Machine Learning (ICML)*, 2012.

[3] William Chan, Navdeep Jaitly, Quoc Le, and Oriol 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)*, 2016, pp. 4960–4964.

[4] Jing Pan, Joshua Shapiro, Jeremy Wohlwend, Kyu Han, Tao Lei, and Tao Ma, "Asapp-asr: Multistream cnn and self-attentive sru for sota speech recognition," 10 2020, pp. 16–20.

[5] Samuel Thomas, Michael L. Seltzer, Kenneth Church, and Hynek Hermansky, "Deep neural network features and semi-supervised training for low resource speech recognition," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 6704–6708.

[6] Jennifer Drexler, *Deep unsupervised learning from speech*, Ph.D. thesis, 01 2016.

[7] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang, "Conformer: Convolution-augmented transformer for speech recognition," in *INTERSPEECH*, 2020.

[8] Junwen Bai, Bo Li, Yu Zhang, Ankur Bapna, Nikhil Siddhartha, Khe Chai Sim, and Tara N. Sainath, "Joint unsupervised and supervised training for multilingual ASR," in *International Conference on Acoustics, Speech, Signal Processing (ICASSP)*, 2021.

[9] Çaglar Gülçehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loïc Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, "On using monolingual corpora in neural machine translation," *CoRR*, vol. abs/1503.03535, 2015.

[10] Anuroop Sriram, Heewoo Jun, Sanjeev Satheesh, and Adam Coates, "Cold fusion: Training seq2seq models together with language models," *CoRR*, vol. abs/1708.06426, 2017.

[11] Ehsan Variani, David Rybach, Cyril Allauzen, and Michael Riley, "Hybrid autoregressive transducer (hat)," in *International Conference on Acoustics, Speech, Signal Processing (ICASSP)*, 2020.

[12] Jin Xu, Xu Tan, Yi Ren, Tao Qin, Jian Li, Sheng Zhao, and Tie-Yan Liu, "Lrspeech: Extremely low-resource speech synthesis and recognition," in *ACM SIGKDD International Conference on Knowledge Discovery Data Mining*, 2020.

[13] Zhehuai Chen, Yu Zhang, Andrew Rosenberg, Bhuvana Ramabhadran, Gary Wang, and Pedro J. Moreno, "Injecting text in self-supervised speech pretraining," in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*.

[14] Takaaki Hori, Ramón Fernandez Astudillo, Tomoki Hayashi, Yu Zhang, Shinji Watanabe, and Jonathan Le Roux, "Cycle-consistency training for end-to-end speech recognition," in *International Conference on Acoustics, Speech, Signal Processing (ICASSP)*, 2019.

[15] Murali Karthick Baskar, Shinji Watanabe, Ramon Astudillo, Takaaki Hori, Lukáš Burget, and Jan Černocký, "Semi-supervised sequence-to-sequence asr using unpaired speech and text," in *INTERSPEECH*, 2019.

[16] Bolaji Yusuf, Ankur Gandhe, and Alex Sokolov, "Usted: Improving asr with a unified speech and text encoder-decoder," in *International Conference on Acoustics, Speech, Signal Processing (ICASSP)*, 2022.

[17] Ankur Bapna, Yu-An Chung, Nan Wu, Anmol Gulati, Ye Jia, Jonathan H. Clark, Melvin Johnson, Jason Riesa, Alexis Conneau, and Yu Zhang, "SLAM: A unified encoder for speech and language modeling via speech-text joint pre-training," in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2021.

[18] Yun Tang, Hongyu Gong, Ning Dong, Changhan Wang, Wei-Ning Hsu, Jiatao Gu, Alexei Baevski, Xian Li, Abdelrahman Mohamed, Michael Auli, and Juan Pino, "Unified speech-text pre-training for speech translation and recognition," in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2022.

[19] Tara N. Sainath, Rohit Prabhavalkar, A. Bapna, Y. Zu, Z. Huo, Z. Chen, B. Li, W. Wang, and T. Strohman, "Joist: A joint speech and text streaming model for asr," in *IEEE Spoken Language Technology Workshop (SLT)*, 2022.

[20] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018.

[21] Abdelrahman Mohamed, Hung-yi Lee, Lasse Borgholt, Jakob D. Havtorn, Joakim Edin, Christian Igel, Katrin Kirchhoff, Shang-Wen Li, Karen Livescu, Lars Maaløe, Tara N. Sainath, and Shinji Watanabe, "Self-supervised speech representation learning: A review," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1179–1210, 2022.

[22] Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli, "wav2vec: Unsupervised pre-training for speech recognition," *CoRR*, vol. abs/1904.05862, 2019.

[23] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *CoRR*, vol. abs/2006.11477, 2020.

[24] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *CoRR*, vol. abs/2106.07447, 2021.

[25] Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, Jian Wu, Long Zhou, Shuo Ren, Yanmin Qian, Yao Qian, Jian Wu, Michael Zeng, and Furu Wei, "Wavlm: Large-scale self-supervised pre-training for full stack speech processing," *CoRR*, vol. abs/2110.13900, 2021.

[26] Alexei Baevski, Steffen Schneider, and Michael Auli, "vq-wav2vec: Self-supervised learning of discrete speech representations," in *International Conference on Learning Representations (ICLR)*, 2019.

[27] Chung-Cheng Chiu, James Qin, Yu Zhang, Jiahui Yu, and Yonghui Wu, "Self-supervised learning with random-projection quantizer for speech recognition," in *International Conference on Machine Learning (ICML)*, 2022.

[28] Arun Narayanan, Tara N. Sainath, Ruoming Pang, Jiahui Yu, Chung-Cheng Chiu, Rohit Prabhavalkar, Ehsan Variani, and Trevor Strohman, "Cascaded encoders for unifying streaming and non-streaming asr," in *International Conference on Acoustics, Speech, Signal Processing (ICASSP)*, 2021.

[29] Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, R. J. Skerry-Ryan, Rif A. Saurous, Yannis Agiomyrgiannakis, and Yonghui Wu, "Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions," in *International Conference on Acoustics, Speech, Signal Processing (ICASSP)*, 2018.