COMBINING UNSUPERVISED AND TEXT AUGMENTED SEMI-SUPERVISED LEARNING FOR LOW RESOURCED AUTOREGRESSIVE SPEECH RECOGNITION

Chak-Fai Li, Francis Keith, William Hartmann, Matthew Snover

Raytheon BBN Technologies, Cambridge MA, USA

{chak.fai.li, francis.keith, william.hartmann, matt.snover}@raytheon.com

ABSTRACT

Recent advances in unsupervised representation learning have demonstrated the impact of pretraining on large amounts of read speech. We adapt these techniques for domain adaptation in low-resource—both in terms of data and compute—conversational and broadcast domains. Moving beyond CTC, we pretrain state-of-the-art Conformer models in an unsupervised manner. While the unsupervised approach outperforms traditional semi-supervised training, the techniques are complementary. Combining the techniques is a 5% absolute improvement in WER, averaged over all conditions, compared to semi-supervised training alone. Additional text data is incorporated through external language models. By using CTC-based decoding, we are better able to take advantage of the additional text data. When used as a transcription model, it allows the Conformer model to better incorporate the knowledge from the language model through semi-supervised training than shallow fusion. Final performance is an additional 2% better absolute when using CTC-based decoding for semi-supervised training compared to shallow fusion.

Index Terms— seq2seq, unsupervised learning, semi-supervised training, domain adaptation

1. INTRODUCTION

Epistemic uncertainty [1], due to limited training data, is a universal problem for automatic speech recognition (ASR), and machine learning in general. Transcribed data is difficult to obtain because it is either expensive or restricted due to privacy or proprietary reasons. Some companies do have access to tremendous amounts of transcribed data, but it cannot be shared beyond the company. It is also possible to obtain large amounts of transcribed data in certain domains like read speech [2], but that data is unlikely to be useful in building models for more difficult domains like far-field and conversational speech. While gathering large quantities of labeled data is difficult, unlabeled data is easier to find. Assuming the model's purpose is to process large amounts of data, then this same data can serve as untranscribed training data. Even if a human is not allowed to inspect the data for privacy or security reasons, an automated system may still use the data for training.

A common approach to incorporating untranscibed data in ASR training is through semi-supervised training (SST)—the use of pseudotranscripts from an initial model for supervision. A strong lexicon and language model—provided externally to a hybrid model or implicit in a sequence-to-sequence model—restricts the output to sequences of likely words in the language. Because of the extra information added by the language model, SST works remarkably well for ASR. When trying to improve a well-trained supervised model, it can take an order of magnitude more unsupervised data to provide an improvement [3, 4]. In the case of domain adaptation, SST can produce large gains with just a few hours of data [5].

While SST performs well for ASR, it does have drawbacks. The output from the transcription model is a discrete word sequence that has removed uncertainty in the output. Alternatives that utilize the lattice have been proposed, but additional gains are minimal [6]. Most work uses academic datasets where the untranscribed data has already been filtered. We can assume the data is of high quality and matched to the domain of interest. With wild data, the data itself is questionable. Portions of the data may be from the wrong domain, wrong language, or even contain no discernible speech at all. Using wild data indiscriminately can harm the final model. Even in cases where the data is curated, large errors in the hypothesized transcripts, especially in the form of deletions, can harm the final model [5].

An alternative, and potentially complementary approach is unsupervised learning. Unsupervised representation learning can address some of the concerns with SST. While SST requires curation of the untranscribed data, unsupervised training should be more robust to low quality, or out-of-domain data. Since we are not assigning real labels, poor hypothesized transcripts cannot corrupt the model. The model might still waste capacity modeling irrelevant data, but that is less detrimental than hallucinating words for out-of-domain data.

Unsupervised learning approaches work by defining a *pretext* task. We can largely separate the approaches into two categories, contrastive and reconstruction-based. The fundamental difference between the two approaches is that the contrastive approaches require negative samples. The model generates a candidate output and compares it with the true target and a number of negative samples. The objective function is based on the candidate's similarity with the target and distance from the negative samples. In [7], future states are predicted from the current state. The more recent *wav2vec 2.0* approach predicts a state given surrounding context [8], similar to the way BERT [9] is trained for NLP tasks.

In reconstruction-based approaches have no negative samples; the model directly attempts to reconstruct the input. Autoencoder-style approaches are typical reconstruction-based approaches [10, 11, 12]. A variant of wav2vec 2.0, DeCoAR 2.0 [13] replaces the contrastive loss with a reconstruction loss. The recent HuBERT [14] approach can also be considered a reconstruction-based approach. We discuss the HuBERT approach in more detail in the next section.

Reconstruction has the potential downside that the model has no prior knowledge of what parts of the representation are more important than others, so all errors are weighted equally. When dealing with high-dimensional signals (e.g. speech or images), modeling the underlying relationships between the dimensions in the original feature space is difficult and costly. Reconstructing every detail is likely unnecessary as there are many nuisance dimensions that are irrelevant for the eventual task. We know the number of relevant classes/categories are orders of magnitude smaller than the number

of unique signals. We also know the objective function cannot distinguish between irrelevant differences in the reconstructed signal. Approaches like CPC [7], wav2vec 2.0 [8], DeCoAR 2.0 [13] use quantized or latent representations representations to allow the model to focus on the more relevant dimensions during reconstruction.

In this work we use the HuBERT approach to unsupervised learning to pretrain the encoder of a Conformer model for the purposes of domain adaptation. The model is then fine-tuned on out-of-domain supervised data. We compare unsupervised learning with semi-supervised learning, both separately and in combination. We also have access to large amounts of text data in the new domain. Despite the advance of techniques like shallow fusion, deep fusion [15], cold fusion [16], and internal language model estimation [17], there is a limit to how much an autoregressive model can gain from additional language model data. We experiment with utilizing the encoder from the Conformer model as a CTC model in a non-autoregressive framework in order to take full advantage of the additional text data. Our contributions include:

- Application of unsupervised learning in a low-resource domain adaptation task with an autoregressive model
- Integration of external language model information through semi-supervised training
- Confirmation that unsupervised pretraining is both more powerful and complementary with semi-supervised learning.

In Section 2 we describe our approaches to unsupervised and semi-supervised training. Section 3 details the data and experimental setup. Our results are presented and discussed in Section 4, and conclusions are presented in Section 5.

2. LEARNING FROM UNTRANSCRIBED DATA

2.1. Unsupervised Pretraining Approach

The structure of the network in the HuBERT approach [14] is similar to wav2vec 2.0 [8]. Both use the same convolutional waveform encoder followed by a large transformer network, however, HuBERT does not use a quantization component. The first step in the training process requires a k-means clustering of the features to generate unsupervised targets. The initial features are MFCC features, but they are used to bootstrap the initial model and later iterations use an internal representation from the transformer network. Given the clusters as targets, the model is trained using frame-level cross-entropy. Learning the cluster targets would be trivial for a sophisticated model. To combat a degenerative solution, some of the inputs to the transformer network are masked and only updates related to the masked frames are used; instead of directly predicting the target from the input, the model predicts the target based on context.

While we are inspired by the HuBERT approach [14], our implementation and use differs in several aspects from the original work. Our design decisions stem from a desire to reduce computational and data requirements. Instead of building the initial clusters from MFCC features, we first train a Conformer [18] model on a small amount of supervised out-of-domain data. Once the model has been trained, the decoder portion of the network is discarded and only the encoder is kept. We use the final layer of the encoder to generate embeddings for the untranscribed data. After embedding the acoustic signal with the encoder, we cluster the embedded features using k-means. As in [14], the clusters serve as labels for the unsupervised training. We remove the convolutional waveform encoder and train a Conformer encoder from filterbank features instead. During this step, there is no decoder involved. The model is trained using frame-level cross-entropy using the clusters as targets. After the encoder has been pretrained using the untranscribed data, a randomly initialized decoder is appended and the entire model is fine-tuned using the original transcribed audio.

Our work also differs from HuBERT in terms of the task. Our data consists of a mix of CTS and broadcast news, more difficult domains than read speech. The HuBERT paper used a minimum of 960 hours and as much as 60k hours, much more data than we have available. We consider the task of domain adaptation with unsupervised learning. The models trained in the HuBERT paper were CTC models with external LMs. We focused on encoder-decoder style models, both with and without external LMs. The HuBERT paper mentions that the number of GPUs used during unsupervised pretraining is critical because of batch size. A minimum of 16 GPUs was required to achieve good results. We found performance to be stable with respect to the number of GPUs used, with a single GPU being sufficient. The original HuBERT paper focuses only on masked frames during unsupervised learning. We perform the same update on the model regardless of whether the original frame was masked.

We do not report these results, but our model failed to improve over the baseline when the clusters were generated using MFCC features. While best performance was seen by clustering features extracted from an encoder, the original HuBERT paper was able to learn using the MFCC features for clustering. The reason for our failure to learn using MFCC-based clusters could be due to a variety of factors including more difficult, limited data, and limited GPUs.

2.2. Semi-Supervised Learning

The standard approach to semi-supervised training uses an initial model to transcribe the unlabeled data and then treats the hypothesized transcripts as truth. We start with the supervised model as our transcription model. During SST we use both the supervised data and the untranscribed data, making no distinction between the two during the training process. For hybrid models keeping only a subset of the hypothesized transcripts can be important. The common approach is to simply filter by confidence, but more sophisticated methods are sometimes used [5, 19]. For autoregressive models, we found selection to be less of an issue. The models are data hungry and tend to perform better with more data, even if it is lower quality.

The recent *noisy student* approach [20, 21] is also related. That approach extends the classical approach to SST by applying more data augmentation and a more careful filtering of the pseudotranscripts. Typically the noisy student work also increases the model size as the amount of data increases. Since the untranscribed data in our sets is significantly less than the amount used in the noisy student work, we do not apply these additional improvements.

3. EXPERIMENTAL SETUP

3.1. MATERIAL Data

We use three languages from the IARPA MATERIAL[1] program: Bulgarian, Swahili, and Tagalog. The languages are a representative sample of the nine languages from the program in terms of difficulty. All transcribed training data in the program consists of conversational telephone speech (CS). The test data contains a small amount of CS data, but mostly consists of news broadcast (NB) and topical broadcast (TB) data. The same is true of the additional untranscribed data used for training. For the remainder of the paper, we combine

[1] https://www.iarpa.gov/index.php/research-programs/material

Table 1. *Amount of transcribed CS and untranscribed CS and BN data for each language. Most of the untrascribed data is BN and no transcribed BN data is available.*

Language	Transcribed		Untranscribed	
	CS	BN	CS	BN
Bulgarian	41.1	0	33.3	149.9
Swahili	68.3	0	57.6	149.0
Tagalog	127.9	0	48.4	153.8

the two broadcast sets into a single set labeled broadcast (BN). This represents a large domain shift between the training and test data. In addition to the domain shift, the out-of-vocabulary (OOV) rate is high for all languages. To address the domain shift from the text side we augment that text data with source-side parallel data provided by the MATERIAL program, ParaCrawl [22] data (where available), and automatically collected web data [23]. The total amount of text data ranges from 80 to 120 million words per language. Note that the text data is still out-of-domain, though likely a closer match than the original acoustic transcripts. The distribution of domain for the acoustic data is shown in Table 1. For more details about the data, see [24]. Note that the transcribed CS data for Swahili and Tagalog is identical to the data from the IARPA BABEL program, available through the LDC (LDC2017S05, LDC2016S13).

3.2. Conformer Model Training

Our sequence-to-sequence models are conformer-based [18] encoder-decoder models trained in EspNet [25]. The configuration is similar to the ones described in [26]. The encoder has 12 layers with four attention heads, an embedding dimension of 256, and a FFN dimension of 2048. The decoder uses 6 layers with identical parameters. In addition to the standard cross-entropy objective function, we also use the CTC objective function [27]. Our output units are characters.

During unsupervised pretraining, we use frame-level cross entropy to train the Conformer encoder with the k-means clusters as targets. The encoder is trained for a maximum of 40 epochs. The minibatch size is 128 utterances, corresponding to approximately 500 seconds of audio. We use SpecAugment [28] during both supervised and unsupervised training.

3.3. Combining the Encoder with N-Gram Language Models

Since our Conformer models are jointly trained with a cross-entropy and CTC objective function, the encoder can be used separately from the decoder for ASR. Using the encoder alone makes the model non-autoregressive. One benefit of a non-autoregressive model is it will generate posteriors for each output unit at each frame, making it easy to combine the model with an external word-level LM and lexicon.

We combine a weighted finite state transducer (WFST)-based decoder with the Conformer encoder in order to utilize an expanded lexicon and n-gram language model. For our experiments, we use trigram LMs. This has been done before in [29][30], and in particular our approach is similar to WFST decoding in [30] using a decoder implemented in Kaldi [31]. The WFST is a simple composition of three transducers: a token transducer, which removes blank symbols output by CTC and collapses repeated characters; a lexicon transducer, which maps sequences of collapsed characters into words; and a grammar transducer, which contains the n-gram LM.

Table 2. *Performance (WER) of unsupervised clusters from K-Means compared to using position-dependent phone labels.*

Language	Cluster Type	CS	BN
Bulgarian	Position Dependent Phones	40.1	38.7
Bulgarian	K-Means (5000)	35.0	40.2
Swahili	Position Dependent Phones	38.1	57.9
Swahili	K-Means (5000)	33.7	49.0
Tagalog	Position Dependent Phones	44.8	52.8
Tagalog	K-Means (5000)	38.8	53.3

Table 3. *Comparing WER performance using unsupervised and semi-supervised training without external language model data.*

Model Description	Bulgarian		Swahili		Tagalog	
	CS	BN	CS	BN	CS	BN
Supervised	39.1	50.7	37.5	58.1	43.5	65.0
SST iter. 1	35.5	41.5	35.7	49.7	42.4	57.6
SST iter. 2	34.8	36.9	36.3	47.0	42.6	53.7
Unsup iter. 1	35.0	40.2	33.7	49.0	38.8	53.3
Unsup iter. 2	32.1	35.0	33.6	44.2	**38.6**	49.4
SST from Unsup iter. 1	**31.9**	**33.4**	**33.4**	**36.9**	39.2	**47.8**

4. RESULTS

4.1. Impact of Cluster Type in Unsupervised Training

For all three languages we compared the unsupervised targets with using position-dependent phones as targets. Results are in Table 2. Results are similar to using the unsupervised clusters, except for Swahili where the unsupervised clusters are significantly better. Note that we do not have ground truth transcripts for the unsupervised data, so the phone targets come from forced alignment with the hypothesized transcripts from a hybrid model. We did tune the number of clusters on Bulgarian, cluster size has little impact on overall performance. In all cases we are using the output from a supervised encoder for clustering. We were unable to improve over a baseline model when using MFCC features for clustering.

4.2. Performance without Additional LM Data

Results without an external language model can be seen in Table 3. All models are Conformer encoder-decoder models; the external LM is a two-layer LSTM model. The supervised model, trained only on CS, sets the baseline for all subsequent models. While WER for BN is typically lower than CS in most languages, that is not that case in our results. The supervised model has never seen broadcast data and the BN data has a high OOV rate, explaining the discrepancy.

We directly compare the WER reduction from semi-supervised and unsupervised training. The semi-supervised model shows a clear improvement across all conditions, but the unsupervised model consistently outperforms the semi-supervised model (41.7% average WER vs. 43.7%). When considering multiple iterations, two rounds of unsupervised training is better than two rounds of SST. In fact, the gap between the unsupervised and semi-supervised approach grows from 2.0% absolute to 3.1%. We also consider combining the two approaches by applying SST using the initial unsupervised model

for transcription. Combining the two approaches gives the best overall performance—4.8% absolute improvement compared to multiple iterations of SST alone—demonstrating the complementarity of the unsupervised and semi-supervised approaches.

The overall improvement on BN data is not surprising given the lack of BN data in the supervised model, but the gain in CS is also significant. Unsupervised pretraining improves the CS result of Tagalog by almost five points absolute. Tagalog had the largest amount of supervised CS data ($\approx$ 120 hours) and there was only an additional 50 hours in the untranscribed set. The unsupervised pretraining clearly provides benefits for both in and out-of-domain data.

4.3. Performance with additional LM data

Table 4. *Comparing performance using supervised, unsupervised, and semi-supervised training with external language data. For each model comparisons are given using shallow fusion and CTC decoding with an N-Gram LM*

Supervision	Transcription Model	Bulgarian		Swahili		Tagalog	
		CS	BN	CS	BN	CS	BN
Supervised	**none**						
Fusion		37.6	41.8	38.6	52.6	46.0	62.5
CTC		39.2	36.4	38.6	48.1	46.5	56.8
Unsup.	**none**						
Fusion		35.1	32.9	35.6	44.2	41.4	51.1
CTC		37.5	29.1	36.4	40.0	42.0	48.7
SST	**Sup+Fusion**						
Fusion		33.7	32.2	37.3	43.0	43.4	55.4
CTC		38.4	30.9	39.0	39.3	45.3	49.6
SST	**Unsup+Fusion**						
Fusion		31.6	26.4	35.3	38.3	38.6	44.9
CTC		37.5	26.9	36.3	35.6	41.2	41.8
SST	**Unsup+CTC**						
Fusion		**30.8**	**24.5**	**33.4**	**33.7**	**38.2**	**41.0**
CTC		36.1	27.3	35.1	34.9	41.8	44.1

Table 4 contains the full set of results comparing unsupervised and SST with the use of external LMs. Initially, we consider the use of shallow fusion and ignore the CTC results. Overall, we can see a dramatic improvement in the performance on BN for all languages and models. The external LM has a much lower OOV rate and the additional language data is more similar to BN data than CS data. Once the external LM has been incorporated, the superiority of the unsupervised approach over the SST approach is no longer seen; they each outperform the other half the time. However, best performance is still achieved by combining the two approaches. Average performance of the supervised model is 46.5%. A single round of SST improves the WER to 40.8%, while SST on top of the unsupervised pretraining further improves the WER to 35.9%.

4.4. CTC-based Decoding with N-Gram LMs

In addition to shallow fusion, Table 4 contains results using the CTC encoder with an external lexicon and LM. Entries marked with *CTC* use a trigram LM trained on the same data as the model used for shallow fusion. CTC only decoding is significantly worse than using the full Conformer model. Due to space, full results are not shown, but CTC decoding is about 5 to 8 percent absolute worse than the Conformer model when no external LM is used for either. However, CTC decoding is able to recover the gap and improve upon the Conformer model through the use of an external lexicon and LM. For the supervised, unsupervised, and semi-supervised models (using a transcription model with shallow fusion), the CTC decoding outperforms the full Conformer model with shallow fusion on BN data. For example, the supervised model is improved by 4.5% to 5.7% absolute on BN data when using CTC decoding. Performance on CS data is worse though, likely due the external lexicon and LM providing less additional information for the CS data.

The conclusion from the final semi-supervised model, where the transcription comes from a CTC decode, shows a different pattern. Even for the BN domain, CTC decoding no longer provides an improvement over shallow fusion. When the transcription model uses CTC decoding, the resulting semi-supervised model will have largely incorporated that information into the model. The additional information in the lexicon and LM is no longer beneficial. The CTC decoding is helpful, but only for the initial transcription. Using CTC decoding for transcription in semi-supervised training yields an average WER of 33.6% across all conditions when the full Conformer model is used for decoding with shallow fusion, a 2.3% average improvement over using shallow fusion for pseudotranscription.

5. CONCLUSIONS

Unsupervised representation learning is a powerful approach to using untranscribed data in an ASR system. Prior work has focused on large amounts of read speech. We demonstrate the applicability to smaller amounts of more difficult data—CTS and BN. The approach works on a single GPU and does not require multiple GPUs to increase the minibatch size. Previous work has focused on non-autoregressive models trained with CTC, but we show the approach can improve an autoregressive Conformer model. In a direct comparison with SST approaches, the unsupervised approach proves more effective; though we achieve our best performance by combining the techniques—5% absolute better than SST alone.

We demonstrated that external language models can significantly improve performance on out-of-domain test sets. While an autoregressive model like the Conformer model is generally superior to a non-autoregressive CTC approach, we show that CTC decoding can make more effective use of external language model data. By using CTC decoding with an external LM as a transcription model for semi-supervised training, we are able to incorporate the information into the Conformer model, leading to an overall additional improvement of 2% absolute. Once we have used CTC decoding in semi-supervised transcription, it becomes better to use the full autoregressive semi-supervised model with shallow fusion for decoding compared to decoding with CTC.

We plan to continue to improve our unsupervised learning approach. While semi-supervised training is mature, unsupervised learning is a less explored space. We expect further optimization will lead to even larger gains.

6. ACKNOWLEDGEMENTS

This work was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense US Air Force Research Laboratory contract number FA8650-17-C-9118. This document does not contain technology or Technical Data controlled under either the U.S. International Traffic in Arms Regulations or the U.S. Export Administration Regulations.

7. REFERENCES

[1] Eyke Hüllermeier and Willem Waegeman, "Aleatoric and epistemic uncertainty in machine learning: A tutorial introduction," *arXiv preprint arXiv:1910.09457*, 2019.

[2] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *IEEE ICASSP*, 2015, pp. 5206–5210.

[3] Yan Huang, Dong Yu, Yifan Gong, and Chaojun Liu, "Semi-supervised gmm and dnn acoustic model training with multi-system combination and confidence re-calibration.," in *Interspeech*, 2013, pp. 2360–2364.

[4] Sree Hari Krishnan Parthasarathi and Nikko Strom, "Lessons from building acoustic models with a million hours of speech," in *IEEE ICASSP*, 2019, pp. 6670–6674.

[5] Shannon Wotherspoon, William Hartmann, Matthew Snover, and Owen Kimball, "Improved data selection for domain adaptation in asr," in *IEEE ICASSP*, 2021.

[6] Vimal Manohar, Hossein Hadian, Daniel Povey, and Sanjeev Khudanpur, "Semi-supervised training of acoustic models using lattice-free mmi," in *IEEE ICASSP*, 2018, pp. 4844–4848.

[7] Aäron van den Oord, Yazhe Li, and Oriol Vinyals, "Representation learning with contrastive predictive coding," 2018.

[8] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," 2020.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[10] Michael Neumann and Ngoc Thang Vu, "Improving speech emotion recognition with unsupervised representation learning on unlabeled speech," in *IEEE ICASSP*, 2019, pp. 7390–7394.

[11] Andy T Liu, Shang-Wen Li, and Hung-yi Lee, "Tera: Self-supervised learning of transformer encoder representation for speech," *IEEE/ACM TASLP*, vol. 29, pp. 2351–2366, 2021.

[12] Yu-An Chung, Wei-Ning Hsu, Hao Tang, and James Glass, "An unsupervised autoregressive model for speech representation learning," *arXiv preprint arXiv:1904.03240*, 2019.

[13] Shaoshi Ling and Yuzong Liu, "Decoar 2.0: Deep contextualized acoustic representations with vector quantization," *arXiv preprint arXiv:2012.06659*, 2020.

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

[15] Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loic Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, "On using monolingual corpora in neural machine translation," *arXiv preprint arXiv:1503.03535*, 2015.

[16] Anuroop Sriram, Heewoo Jun, Sanjeev Satheesh, and Adam Coates, "Cold fusion: Training seq2seq models together with language models," *arXiv preprint arXiv:1708.06426*, 2017.

[17] Zhong Meng, Sarangarajan Parthasarathy, Eric Sun, Yashesh Gaur, Naoyuki Kanda, Liang Lu, Xie Chen, Rui Zhao, Jinyu Li, and Yifan Gong, "Internal language model estimation for domain-adaptive end-to-end speech recognition," in *IEEE SLT*, 2021, pp. 243–250.

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

[19] Félix de Chaumont Quitry, Asa Oines, Pedro Moreno, and Eugene Weinstein, "High quality agreement-based semi-supervised training data for acoustic modeling," in *IEEE SLT*, 2016, pp. 592–596.

[20] Daniel S. Park, Yu Zhang, Ye Jia, Wei Han, Chung-Cheng Chiu, Bo Li, Yonghui Wu, and Quoc V. Le, "Improved noisy student training for automatic speech recognition," 2020.

[21] Yu Zhang, James Qin, Daniel S Park, Wei Han, Chung-Cheng Chiu, Ruoming Pang, Quoc V Le, and Yonghui Wu, "Pushing the limits of semi-supervised learning for automatic speech recognition," *arXiv preprint arXiv:2010.10504*, 2020.

[22] Miquel Esplà-Gomis, Mikel L Forcada, Gema Ramírez-Sánchez, and Hieu Hoang, "Paracrawl: Web-scale parallel corpora for the languages of the eu," in *Proceedings of Machine Translation Summit XVII*, 2019, pp. 118–119.

[23] Le Zhang, Damianos Karakos, William Hartmann, Roger Hsiao, Richard Schwartz, and Stavros Tsakalidis, "Enhancing low resource keyword spotting with automatically retrieved web documents," in *Interspeech*, 2015.

[24] Chak-Fai Li, Francis Keith, William Hartmann, Matthew Snover, and Owen Kimball, "Overcoming domain mismatch in low resource sequence-to-sequence asr models using hybrid generated pseudotranscripts," *arXiv preprint arXiv:2106.07716*, 2021.

[25] Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, et al., "Espnet: End-to-end speech processing toolkit," *arXiv preprint arXiv:1804.00015*, 2018.

[26] Pengcheng Guo, Florian Boyer, Xuankai Chang, Tomoki Hayashi, Yosuke Higuchi, Hirofumi Inaguma, Naoyuki Kamo, Chenda Li, Daniel Garcia-Romero, Jiatong Shi, et al., "Recent developments on espnet toolkit boosted by conformer," in *IEEE ICASSP*, 2021, pp. 5874–5878.

[27] Suyoun Kim, Takaaki Hori, and Shinji Watanabe, "Joint ctc-attention based end-to-end speech recognition using multi-task learning," in *IEEE ICASSP*, 2017, pp. 4835–4839.

[28] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.

[29] Yajie Miao, Mohammad Gowayyed, and Florian Metze, "Eesen: End-to-end speech recognition using deep rnn models and wfst-based decoding," in *IEEE ASRU*, 2015, pp. 167–174.

[30] Thomas Zenkel, Ramon Sanabria, Florian Metze, Jan Niehues, Matthias Sperber, Sebastian Stüker, and Alex Waibel, "Comparison of decoding strategies for ctc acoustic models," *arXiv preprint arXiv:1708.04469*, 2017.

[31] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., "The kaldi speech recognition toolkit," in *IEEE ASRU*, 2011.