FINE-GRAINED TEXTUAL KNOWLEDGE TRANSFER TO IMPROVE RNN TRANSDUCERS FOR SPEECH RECOGNITION AND UNDERSTANDING

Vishal Sunder[*1], *Samuel Thomas*[2], *Hong-Kwang J. Kuo*[2], *Brian Kingsbury*[2], *Eric Fosler-Lussier*[1]

[1] The Ohio State University, [2] IBM Research AI

ABSTRACT

RNN Tranducer (RNN-T) technology is very popular for building deployable models for end-to-end (E2E) automatic speech recognition (ASR) and spoken language understanding (SLU). Since these are E2E models operating on speech directly, there remains a potential to improve their performance using purely text based models like BERT, which have strong language understanding capabilities. In this paper, we propose a new training criteria for RNN-T based E2E ASR and SLU to transfer BERT's knowledge into these systems. In the first stage of our proposed mechanism, we improve ASR performance by using a fine-grained, tokenwise knowledge transfer from BERT. In the second stage, we fine-tune the ASR model for SLU such that the above knowledge is explicitly utilized by the RNN-T model for improved performance. Our techniques improve ASR performance on the Switchboard and CallHome test sets of the NIST Hub5 2000 evaluation and on the recently released SLURP dataset on which we achieve a new state-of-the-art performance. For SLU, we show significant improvements on the SLURP slot filling task, outperforming HuBERT-base and reaching a performance close to HuBERT-large. Compared to large transformer based speech models like HuBERT, our model is significantly more compact and uses only 300 hours of speech pretraining data.

Index Terms— automatic speech recognition, spoken language understanding, knowledge transfer

1. INTRODUCTION

Transformer based language models like BERT [1] have a good semantic understanding of language as evidenced by their performance on various language understanding tasks. On the other hand, many modern end-to-end (E2E) automatic speech recognition (ASR) systems are trained without any explicit criterion for understanding language semantics like BERT. The lack of such knowledge may lead them to underperform in downstream E2E spoken language understanding (SLU) tasks. Transformer based speech encoders like HuBERT are indeed trained using the masked language modelling criterion like BERT and have performed well in ASR and SLU tasks. Yet, their use in real-world settings, which may involve on-device deployment, is constrained by their large size and audio processing latency. It is thus imperative to devise methods for efficient knowledge transfer from existing models like BERT into conventional speech models like RNN transducers (RNN-T) [2], which are compact and deployment friendly. In this work, we propose techniques to perform this knowledge transfer such that both ASR and SLU performances are improved.

There has been an increased interest in exploring techniques to distill semantic knowledge from BERT into speech processing models. One class of approaches proposes distillation techniques so that ASR performance is improved by transferring knowledge into the text generation module of the ASR model, where it is trivial to do a token by token comparison with BERT's output [3, 4, 5, 6]. While these techniques improve ASR accuracy, it is not clear how they can be useful for SLU, where the decoding targets are a sequence of slots and values and not natural language text. Thus, knowledge gained from natural language sentences during ASR training cannot transfer to a sequence of slots and values which occupy a completely different semantic space. In this paper, we propose to distill the knowledge from BERT embeddings into the transcription network of the RNN-T model, which is the speech encoder, rather than the prediction network, which is the text encoder. This ensures that the knowledge from BERT is always retained in the speech embeddings irrespective of the final task. To overcome the challenge of sequence length mismatch between a natural language sequence and a speech sequence, we propose to integrate a tokenwise alignment criterion in the RNN-T ASR training [7].

Another class of approaches focuses on utilizing BERT's knowledge for E2E speech-to-intent (S2I) tasks. This is done by learning an embedding level alignment between the text representation from BERT and the speech representation from the speech encoder [8, 9, 10, 11, 12]. For S2I tasks, speech embeddings (aligned with BERT embeddings) are fed forward into a classifier which predicts an intent. However, it is not clear how these techniques can help with slot filling which require a full decoding of a sequence of slots and values.

One can argue that improving an ASR model will be enough to improve the downstream slot filling task. While this has been the approach adopted by many researchers [13, 14, 15], in this paper, we adopt a novel technique to further improve RNN-T's slot filling performance by explicitly incor-

[*] Work done during an internship at IBM

porating the knowledge from BERT acquired during the ASR pretraining stage into the SLU fine-tuning stage. This is done by utilizing a self attention layer for SLU training which acts as a proxy for BERT and is seeded through the pretraining stage. Techniques that align speech and text on a token-by-token basis to improve ASR and SLU in CTC based models ([16]) are restricted to speech-text paired data and generic text encoders for alignment. This reduces the potential of utilizing large models like BERT trained on large scale text-only data.

Our proposed knowledge transfer techniques improve ASR and SLU performance over strong baselines. We report ASR improvements on the Switchboard and CallHome test sets of the NIST Hub5 2000 evaluation and on the recently released SLURP dataset, setting a new state-of-the-art (SOTA) on the latter. We also show significant improvements in SLU performance on the SLURP dataset, outperforming HuBERT-base [14] on the slot filling task and reaching a performance close to HuBERT-large [14] with five times fewer parameters and only 300 hours of speech pretraining data.

2. RNN TRANSDUCERS

We describe the traditional RNN-T based ASR modelling, borrowing some notation from [2, 17]. For a speech sequence $\mathbf{x} = (x_1, ..., x_T)$ of length T, the RNN-T models the conditional distribution, $p(\mathbf{y}|\mathbf{x})$ of output sequence $\mathbf{y} = (y_1, ..., y_U)$ of length U. This distribution is learnt as a marginalization of all possible alignments of length $T + U$ between $\mathbf{x}$ and $\mathbf{y}$ such that the model is allowed to output T BLANK symbols.

The input speech sequence is encoded by a transcription network which we implement as a bidirectional LSTM whose output is a sequence of speech embeddings, $\mathbf{H} = [\mathbf{h}_1\mathbf{h}_2...\mathbf{h}_T]^{\mathbb{T}}$. Similarly, the output grapheme sequence is encoded by a prediction network implemented as a unidirectional LSTM and whose output is denoted as $\mathbf{G} = [\mathbf{g}_1\mathbf{g}_2...\mathbf{g}_U]^{\mathbb{T}}$. Now, a joint network is used to model the probability distribution over the set of output symbols given a combination of $\mathbf{h}_t$ and $\mathbf{g}_u$ as,

$$p^{ASR}(.|\mathbf{h}_t, \mathbf{g}_u) = \text{softmax}[\mathbf{W}^{out}\text{tanh}(\mathbf{W}^{enc}\mathbf{h}_t + \mathbf{W}^{pred}\mathbf{g}_u + \mathbf{b})]$$

The probability of an alignment is computed using the above distribution, and a marginalization over all alignments gives $p(\mathbf{y}|\mathbf{x})$. Here, $\mathbf{W}^{out}$, $\mathbf{W}^{enc}$, $\mathbf{W}^{pred}$ and $\mathbf{b}$ are learnable parameters. The model is trained by minimizing the negative log likelihood, $L^{ASR} = -\log p(\mathbf{y}|\mathbf{x})$ over the training set.

2.1. Stage I: Knowledge transfer for ASR

We transfer BERT's knowledge into the transcription network of the RNN-T during ASR training. To do this, we utilize the tokenwise contrastive learning criterion [7]. Figure 1 gives an overview of the ASR training setup.

A sequence of non-contextual (NC) WordPiece embeddings (with absolute position encodings), $\mathbf{E} \in \mathbb{R}^{n\times 768}$ of an utterance is converted to a sequence of contextual embeddings,

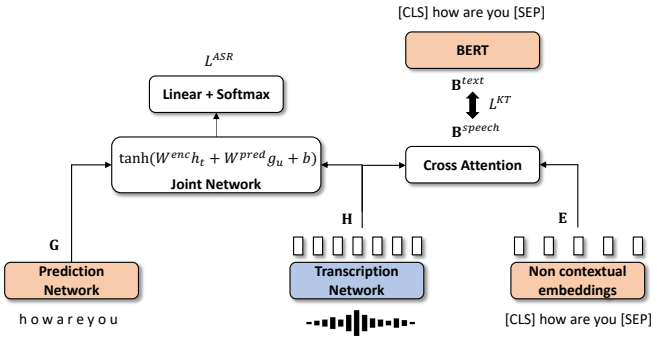

Fig. 1. ASR training with knowledge transfer from BERT.

$\mathbf{B}^{speech} \in \mathbb{R}^{n\times 768}$ using cross-attention between the output of the transcription network[1], $\mathbf{H} \in \mathbb{R}^{T\times 768}$ and $\mathbf{E}$. Then, $\mathbf{B}^{speech}$ is aligned, token by token, with the output the BERT model, $\mathbf{B}^{text} \in \mathbb{R}^{n\times 768}$. $\mathbf{E}$ is initialized with BERT's WordPiece embedding layer.

The cross-attention is query-key-value based and has a set of learnable weights $\mathbf{W}^q$, $\mathbf{W}^k$ and $\mathbf{W}^v \in \mathbb{R}^{768\times 768}$, then queries, keys and values are computed as,

$$\mathbf{Q} = \mathbf{E}\mathbf{W}_q$$
$$\mathbf{K} = \mathbf{H}\mathbf{W}_k$$
$$\mathbf{V} = \mathbf{H}\mathbf{W}_v$$

Now, the contextual embeddings, $\mathbf{B}^{speech}$ are computed as,

$$\mathbf{B}^{speech} = \text{softmax}(\mathbf{Q}\mathbf{K}^{\mathbb{T}})\mathbf{V}$$

$\mathbf{B}^{speech}$ can now be aligned with $\mathbf{B}^{text}$ easily as both have the same sequence length. It is important to use NC embeddings as queries [7]. Otherwise, the cross-attention mechanism ignores the context from speech, failing to learn a meaningful alignment between speech and BERT embeddings.

The alignment between $\mathbf{B}^{text}$ and $\mathbf{B}^{speech}$ is computed as a contrastive loss. To do this, the output sequences in a batch are row-wise concatenated such that $\mathbf{B}^{text}$ and $\mathbf{B}^{speech}$ are $\in \mathbb{R}^{b\times 768}$ where b is the sum of all sequence lengths in a batch. Now the contrastive loss is computed as,

$$L^{KT} = -\frac{\tau}{2b}\sum_{i=1}^{b}(\log\frac{\exp(s_{ii}/\tau)}{\sum_{j=1}^{b}\exp(s_{ij}/\tau)} + \log\frac{\exp(s_{ii}/\tau)}{\sum_{j=1}^{b}\exp(s_{ji}/\tau)})$$

Here, s_{ij} is the cosine similarity between the i^{th} row of $\mathbf{B}^{text}$ and the j^{th} row of $\mathbf{B}^{speech}$ and τ is the temperature.

The final loss function to train the ASR model is,

$$L^{KT} + \lambda L^{ASR}$$

Here, λ scales the ASR loss. We found that $\lambda = 0.20$ and $\tau = 0.07$ works best in most cases. During training, BERT is kept frozen.

[1]If the output of the transcription network is not 768 dimensional, it can be converted using a linear layer. Here, we use 768 for brevity.

2.2. Stage II: Knowledge transfer for SLU

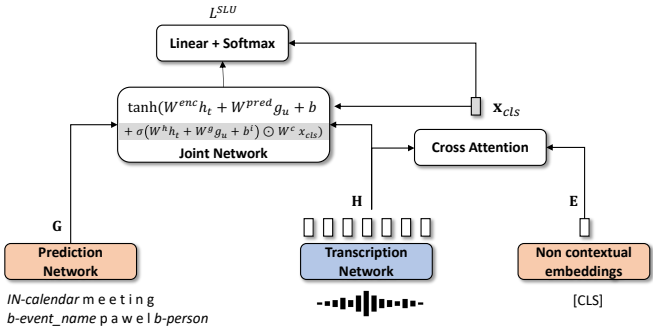

Fig. 2. SLU training with knowledge transfer. $\mathbf{x}_{cls}$ acts as a proxy for the contextual [CLS] embedding from BERT. All modules are initialized with the ASR training in Figure 1. The input utterance is "Put meeting with Pawel".

An RNN-T can be adapted for SLU by fine-tuning it to produce a sequence of intent, slots and values. This has been the approach of most E2E SLU systems [18, 19]. In this work, we use an approach so that the knowledge gained from stage I can be utilized more effectively for SLU as shown in Figure 2.

In particular, we use the cross attention mechanism from stage I and the NC embedding of [CLS] to get the embedding $\mathbf{x}_{cls}$ from the transcription network. This embedding is an approximation of the [CLS] embedding from BERT for the input utterance and can be used in the RNN-T for better SLU. To do this, we modify the joint network equation of the RNN-T to allow for information flow from $\mathbf{x}_{cls}$. This information flow is controlled by a gating mechanism which is a function of $\mathbf{h}_t$ and $\mathbf{g}_u$. Also, $\mathbf{x}_{cls}$ is concatenated in the input to the final classification layer. The joint network is modified as,

$$\begin{aligned} \mathbf{i} &= \sigma(\mathbf{W}^h\mathbf{h}_t + \mathbf{W}^g\mathbf{g}_u + \mathbf{b}^i) \odot \mathbf{W}^c\mathbf{x}_{cls} \\ \mathbf{j} &= \tanh(\mathbf{W}^{enc}\mathbf{h}_t + \mathbf{W}^{pred}\mathbf{g}_u + \mathbf{b} + \mathbf{i}) \\ p^{SLU}(.|\mathbf{h}_t, \mathbf{g}_u, \mathbf{x}_{cls}) &= \text{softmax}[\mathbf{W}^{out}\text{concat}(\mathbf{j}, \mathbf{x}_{cls})] \end{aligned} \quad (1)$$

Here, $\mathbf{i}$ is information from $\mathbf{x}_{cls}$, controlled through a sigmoid gate $\sigma(.)$, $\odot$ is elementwise multiplication. The joint network output, $\mathbf{j}$, is concatenated with $\mathbf{x}_{cls}$ and fed into the classifier.

The SLU loss, L^{SLU} is computed in the same way as L^{ASR}, but over sequences of intent, slots and values instead of the utterance sequence. The weights $\mathbf{W}^h, \mathbf{W}^g, \mathbf{W}^c$ and $\mathbf{b}^i$ are newly initialized for SLU. $\mathbf{W}^{out}$ is reused from the pretraining stage but we also add the extra columns of new weights in it.

3. EXPERIMENTAL SETUP

3.1. Datasets

ASR pretraining: We use 300 hours of the Switchboard dataset for pretraining the ASR models. This dataset is a corpus of dyadic English telephone conversations on open ended topics. We evaluate the pretrained models on the commonly used Hub5 2000 Switchboard and CallHome test sets.

ASR adaptation: We adapt the pretrained models on the recently released SLURP dataset [20]. This dataset comprises around 80 hours of training audio out of which 40 hours is synthetic. The 10 hour test set of SLURP, being far-field audio, is challenging for ASR.

SLU adaptation: For SLU, we again use the SLURP dataset which is annotated for slots, values and intents. The ASR adapted model is further adapted for SLU.

3.2. Data Augmentation

We use the following data augmentation techniques to train all our models for optimal performance.

Speed and tempo augmentation [21]: By changing the rate of spoken utterances by 1.1 and 0.9, the original dataset is augmented with additional copies.

SpecAugment [22]: We mask continuous frequency and time intervals of the log-mel spectrogram input according to the SM policy in Park et al. [22].

Sequence noise injection [23]: The downscaled spectra of a random utterance is added to the input log-mel.

Reverberation (SLURP only)[2]: To the synthetic part of the SLURP dataset, we add reverberation to make it far-field such that the training set matches the test conditions.

3.3. Training details

We model the transcription network of the RNN-T as a 6-layer BiLSTM with 1280 hidden units and the prediction network as a single layer LSTM with 1024 hidden units.

The models are trained using 40-dimensional, global mean and variance normalized log-mel filterbank features, extracted every 10 ms using a 25 ms window. These features are augmented with Δ and Δ^2 coefficients. We stack consecutive frames and skip every other frame resulting in a 240 dimensional sequence of speech features. The input to the prediction network is a sequence of graphemes. Models are trained with a batch size of 32 on a A100 GPU.

We use the AdamW [24] optimizer and a OneCycleLR policy [25] with these schedules:

ASR pretraining: 60 epochs with a peak learning rate of 5e-4

ASR adaptation: 20 epochs with a peak learning rate of 2e-4

SLU adaptation: 20 epochs with a peak learning rate of 2e-5

4. RESULTS

The results for ASR are shown in Table 1. We report the results on Switchboard and SLURP datasets. We pretrain the ASR model on 300 hours of Switchboard data and then adapt this model for the SLURP dataset. For the SLURP dataset, we run ASR experiments with (SLP+) and without synthetic

[2] https://github.com/mravanelli/pySpeechRev

	Pretraining		→	Adaptation	
Model	SWB	CH		SLP	SLP+
ASR	7.3	15.7		19.6	15.8
ASR w/ KT	**7.2**	**14.8**		**18.9**	**14.8**

Table 1. ASR performance in WER(↓) using the baseline RNN-T and with the proposed knowledge transfer (KT) in section 2.1. Here, SWB and CH are the Switchboard and CallHome test sets. Results on the SLURP test set are with models trained without (SLP) and with (SLP+) synthetic data.

data (SLP). From Table 1, we note that using knowledge transfer (KT) with ASR lowers the word error rate (WER) on all datasets. Although the improvement on SWB is small, we see more improvements on other test sets. Particularly, when trained with SLP+, we are able to achieve SOTA performance with our proposed approach on the SLURP test set, the previous SOTA being a WER of 15.2% by Raju et al. [15].

The SLU results are reported in Table 2. We report the slot filling F1 (SF), intent classification accuracy (IC) and the number of parameters used (# Params). The first six rows are the baseline models. Rows (7) to (11) are implementations of our E2E models. All our E2E models are significantly more compact compared to all the baselines. This is a very important attribute for building real-world, deployable models.

Row (1) represents the oracle BERT model run on ground truth transcripts. We train the BERT model to emit IOB tags for corresponding entities in the input and the intent tag at the [CLS] token. Row (2) represents the traditional ASR NLU cascaded setup where we use our best performing ASR model (WER of 14.8) to transcribe speech and then use the BERT model to tag the ASR transcript. Compared to (1), we see that the performance drops dramatically. Rows (3) to (6) represent a subset of previous work with best performance on SLURP.

Row (7) is our baseline E2E SLU model where we train an ASR model and adapt it for SLU without any of our proposed techniques. Rows (8) to (11) are variants of our proposed KT techniques. In row (8), we show results for a model pretrained for ASR with KT (section 2.1) and then adapted for SLU without KT. We already see an improvement in the SF and IC performance. This shows that the proposed KT methodology for ASR also helps with SLU. But this improvement might just be due to ASR improvement. Hence, to further boost SLU performance, we implement KT for SLU proposed in section 2.2. Rows (9) to (11) show variants of this approach.

We run three different variants of KT for SLU. First, we set $\mathbf{i} = \mathbf{0}$ in Equation 1. This reduces the SLU model to an RNN-T which concatenates $\mathbf{x}_{cls}$ before the final classification layer. Using this, we see an improvement in row (9) compared to row (8). This shows that $\mathbf{x}_{cls}$ is like BERT's [CLS] embedding and useful not just for IC but also SF.

Next, in row (10), we incorporate the entire information from $\mathbf{x}_{cls}$ into the joint network of the RNN-T, i.e. without the gating mechanism in equation 1. This gives some improvement over row (9). However, when we use a gating mechanism as shown in Equation 1, we achieve the best performance in terms of both SF and IC. This may be because now the information from $\mathbf{x}_{cls}$ is a function of $\mathbf{h}_t$ and $\mathbf{g}_u$. This allows for flexibility in how the knowledge gained from BERT is actually used by the RNN-T for SLU instead of just static knowledge integration.

Model	SF	IC	# Params
Baselines			
(1) Oracle BERT	88.54	94.00	110M
(2) Cascaded ASR → BERT	74.83	86.49	172M
(3) wav2vec2.0 [13]	74.62	85.34	94M
(4) CTI [26]	74.66	86.92	313M
(5) HuBERT base [14]	75.32	87.51	94M
(6) HuBERT large [14]	78.92	89.38	315M
Our E2E models			
(7) ASR → SLU	74.35	83.84	62M
(8) ASR w/ KT → SLU	75.90	86.43	62M
(9) ASR w/ KT → SLU w/ KT ($\mathbf{i} = \mathbf{0}$)	76.31	87.39	65M
(10) ASR w/ KT → SLU w/ KT ($\mathbf{i} = \mathbf{W}^c\mathbf{x}_{cls}$)	76.51	87.77	65M
(11) ASR w/ KT → SLU w/ KT ($\mathbf{i} = \sigma(.)\mathbf{W}^c\mathbf{x}_{cls}$)	**76.96**	**87.95**	66M

Table 2. SLU performance on SLURP dataset. Slot filling F1 (↑) (SF), intent classification accuracy (↑) (IC) and number of model parameters (in million) reported.

Rows (8) to (11) show how each of our proposed techniques contributes to reaching the final best performance. Note that our best performing model in row (11) falls short of the SOTA HuBERT model in row (6); however, our model is five times smaller than HuBERT large. This is a significant advantage of our model in terms of real-world utility. Also, all our models are pretrained only on 300 hours of speech from Switchboard, whereas HuBERT large is trained on 60,000 hours of Librilight data. Thus, in terms of training setup, our model is more accessible. Apart from HuBERT large, we outperform all other baselines, and use fewer parameters.

5. CONCLUSION

In this paper, we propose knowledge transfer techniques for both E2E ASR and SLU. Using BERT as a teacher, we perform a fine grained knowledge transfer on a token by token basis from BERT into the transcription network on an RNN-T model. Furthermore, we extend our model such that the knowledge gained during the ASR stage can be explicitly utilized in the SLU stage. Our methods improve ASR performance on Switchboard and SLURP dataset and do better on the slot filling task on SLURP compared to strong baselines.

6. ACKNOWLEDGEMENTS

We would like to thank George Saon for providing us the initial code for the RNN-T model.

7. REFERENCES

[1] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *NAACL*, 2019.

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

[3] Y. Kubo, S. Karita, and M. Bacchiani, "Knowledge transfer from large-scale pretrained language models to end-to-end speech recognizers," in *ICASSP*, 2022.

[4] H. Futami, H. Inaguma, S. Ueno, M. Mimura, S. Sakai, and T. Kawahara, "Distilling the knowledge of BERT for sequence-to-sequence ASR," *Interspeech*, 2020.

[5] K. Choi and H. Park, "Distilling a pretrained language model to a multilingual ASR model," *Interspeech*, 2022.

[6] Yosuke Higuchi, Brian Yan, Siddhant Arora, Tetsuji Ogawa, Tetsunori Kobayashi, and Shinji Watanabe, "BERT meets CTC: New formulation of end-to-end speech recognition with pre-trained masked language model," *arXiv preprint arXiv:2210.16663*, 2022.

[7] V. Sunder, E. Fosler-Lussier, S. Thomas, H.K.J. Kuo, and B. Kingsbury, "Tokenwise contrastive pretraining for finer speech-to-BERT alignment in end-to-end speech-to-intent systems," *Interspeech*, 2022.

[8] P. Denisov and N.T. Vu, "Pretrained semantic speech embeddings for end-to-end spoken language understanding via cross-modal teacher-student learning," *Interspeech*, 2020.

[9] Y. Huang, H.K.J. Kuo, S. Thomas, Z. Kons, K. Audhkhasi, B. Kingsbury, R. Hoory, and M. Picheny, "Leveraging unpaired text data for training end-to-end speech-to-intent systems," in *ICASSP*, 2020.

[10] B. Agrawal, M. Müller, S. Choudhary, M. Radfar, A. Mouchtaris, R. McGowan, N. Susanj, and S. Kunzmann, "Tie your embeddings down: Cross-modal latent spaces for end-to-end spoken language understanding," in *ICASSP*, 2022.

[11] V. Sunder, S. Thomas, H.K.J. Kuo, J. Ganhotra, B. Kingsbury, and E. Fosler-Lussier, "Towards end-to-end integration of dialog history for improved spoken language understanding," in *ICASSP*, 2022.

[12] Y. Chung, C. Zhu, and M. Zeng, "SPLAT: Speech-language joint pre-training for spoken language understanding," *NAACL*, 2020.

[13] M. Ravanelli, T. Parcollet, P. Plantinga, A. Rouhe, S. Cornell, L. Lugosch, C. Subakan, N. Dawalatabad, A. Heba, J. Zhong, et al., "SpeechBrain: A general-purpose speech toolkit," *arXiv preprint arXiv:2106.04624*, 2021.

[14] Y. Wang, A. Boumadane, and A. Heba, "A fine-tuned wav2vec 2.0/HuBERT benchmark for speech emotion recognition, speaker verification and spoken language understanding," *arXiv preprint arXiv:2111.02735*, 2021.

[15] A. Raju, M. Rao, G. Tiwari, P. Dheram, B. Anderson, Z. Zhang, C. Lee, B. Bui, and A. Rastrow, "On joint training with interfaces for spoken language understanding," *Interspeech*, 2022.

[16] W. Wang, Y. Ren, S.and Qian, S. Liu, Y. Shi, Y. Qian, and M. Zeng, "Optimizing alignment of speech and language latent spaces for end-to-end speech recognition and understanding," in *ICASSP*, 2022.

[17] G. Saon, Z. Tüske, D. Bolanos, and B. Kingsbury, "Advancing RNN transducer technology for speech recognition," in *ICASSP*, 2021.

[18] H.K.J. Kuo, Z. Tüske, S. Thomas, Y. Huang, K. Audhkhasi, B. Kingsbury, G. Kurata, Z. Kons, R. Hoory, and L. Lastras, "End-to-End spoken language understanding without full transcripts," in *Interspeech*, 2020.

[19] S. Thomas, H.K.J. Kuo, G. Saon, Z. Tüske, B. Kingsbury, G. Kurata, Z. Kons, and R. Hoory, "RNN transducer models for spoken language understanding," in *ICASSP*, 2021.

[20] E. Bastianelli, A. Vanzo, P. Swietojanski, and V. Rieser, "SLURP: A spoken language understanding resource package," *arXiv preprint arXiv:2011.13205*, 2020.

[21] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *Interspeech*, 2015.

[22] D.s. Park, W. Chan, Y. Zhang, C. Chiu, B. Zoph, E.D. Cubuk, and Q.V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," *Interspeech*, 2019.

[23] G. Saon, Z. Tüske, K. Audhkhasi, and B. Kingsbury, "Sequence noise injected training for end-to-end speech recognition," in *ICASSP*, 2019.

[24] I. Loschilov and F. Hutter, "Decoupled weight decay regularization," in *ICLR*, 2019.

[25] L.N. Smith and N. Topin, "Super-convergence: Very fast training of neural networks using large learning rates," *arXiv preprint arXiv:1708.07120*, 2017.

[26] S. Seo, D. Kwak, and B. Lee, "Integration of pre-trained networks with continuous token interface for end-to-end spoken language understanding," in *ICASSP*, 2022.