INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Efficient Adaptation of Spoken Language Understanding based on End-to-End Automatic Speech Recognition

Eesung Kim, Aditya Jajodia[*], *Cindy Tseng*[*], *Divya Neelagiri*[*], *Taeyeon Ki*[*], *Vijendra Raj Apsingekar*

Samsung Research America, USA

{eesung.kim, aditya.2, c.tseng, d.neelagiri, taeyeon.ki, v.akar}@samsung.com

Abstract

In production scenarios that require frequent change, it is inefficient to repeatedly train and update the entire End-to-end (E2E) model for spoken language understanding (SLU). In this paper, we present a study on efficiently adapting E2E SLU models based on pre-trained ASR model. Specifically, we propose the ASR-based E2E SLU model integrating an additional decoder for SLU and a fusion module that incorporates acoustic representation from the shared encoder and text transcript representation from ASR decoder. Furthermore, we investigate the effectiveness of an adapter module that fine-tunes only a small number of parameters for semantic and transcript predictions. The experimental results show that the proposed model outperforms other competitive baselines in intent accuracy, SLU F1 score and word error rate (WER) on FSC, SLURP, and Samsung in-house SLU datasets.

Index Terms: Spoken Language Understanding, SLU Adaptation, Adapter, Multi-Task Learning

1. Introduction

The task of spoken language understanding (SLU) involves extracting semantic information related to domain, intent, and named entities from spoken commands made by users. SLU is becoming increasingly important for front-end devices such as smartphones, IoT home devices, and virtual assistants. Traditionally, SLU systems [1, 2] employ cascaded approach, consisting of automatic speech recognition (ASR) and natural language understanding (NLU) sequentially. ASR generates text transcripts of user's spoken commands, while NLU extracts semantic information from the text transcripts. However, the cascaded approach has limitations in errors propagating from ASR to NLU and ignoring acoustic information in speech.

Recently, an end-to-end (E2E) SLU-based approach that can directly predict semantic information from speech commands that can directly estimate semantic information from speech commands has been actively investigated [3, 4, 5, 6]. Several works have shown the effectiveness of applying transfer learning when performing E2E SLU tasks [7, 8, 9, 10, 11, 12, 13]. One approach [13] jointly fine-tuned the E2E ASR encoder and the BERT model by applying unsupervised learning and transfer learning. In [7, 10], authors proposed a model trained progressively to transcribe the utterances then extract its semantics. In [9], authors investigated the impact of domain-specific adaptation based on domain independent pre-trained recurrent neural network transducer (RNNT) ASR model by integrating extra output nodes to the pre-trained ASR model as semantic target.

There has also been an efforts to better facilitate the joint training of ASR and NLU with interface [11, 14, 15, 16, 17, 18, 19]. In [11], authors present jointly trainable E2E SLU model, consisting of ASR and NLU subsystems which are connected through an interface related to output of ASR module. [15] has shown an effective approach to jointly train large-size pretrained ASR and NLU models via a sequence loss with well designed variant neural interfaces. [16] also used output of ASR as an interface for Bert-based NLU model with same vocabulary. [20] utilized contrastive learning to learn better multi-modal representation from audio and text encoders for intent classification. In [19], authors suggested a two-pass SLU pipeline that incorporates semantic and acoustic details to enhance intent prediction accuracy and decrease SLU system latency.

However, there are several drawbacks with these approaches. Some approaches directly predict semantic results without taking into account ASR performance, which is important when ASR result is required in an end to end speech recognition system. Another issue is that many of these methods consume significant computational resources or have long training times in order to achieve competitive performance. These issues are especially transparent in two-stage frameworks, approaches which requires updating the entire pre-trained model, or models requiring large amounts of transcribed audio data. Furthermore, to ensure that the ASR model performs well in a real-world voice assistant application, it needs to be frequently updated to reflect real word events such as new popular phrases or global events. It can be very inefficient to train and update the entire E2E SLU model, including the ASR component, every time there is a minor change.

To alleviate these issues, we explore various techniques in transfer learning on E2E SLU and E2E ASR model. Specifically, we propose an attention encoder decoder (AED) architecture-based E2E SLU model with a fusion module which incorporates both acoustic representation and text transcript representation. Furthermore, we investigate effectiveness of adapters [21] for semantic and transcript predictions. The experimental results show that the proposed method achieves better performance on ASR and SLU tasks than existing competitive baselines on SLURP [22] and Fluent Speech Command (FSC) dataset [7]. Our approaches lead to a significant reduction in total number of trainable parameters by a factor of 11 and reduction in total training time by a factor of 1.5, all while maintaining competitive performance. Notably, we achieve state-of-the-art results on the SLURP dataset by jointly fine-tuning the entire SLU models based on task-specific pre-trained ASR model. We also attain strong performance using adapters without fine-tuning the entire pre-trained model on domain specific tasks. Lastly, we conducted an ablation study to evaluate the effect of using a pre-training ASR model and fine-tuning using

[*] Equal Contribution.

10.21437/Interspeech.2023-1878

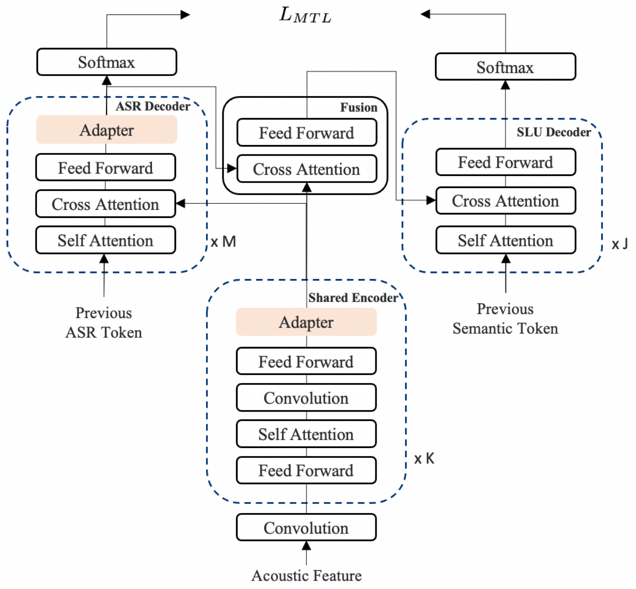

Figure 1: *Overview of proposed architecture with adapter injected in shared encoder and ASR decoder.*

variations adaptation methods.

2. Method

2.1. Model Architecture

An overview of the model architecture is shown in Figure 1. Our model includes a conformer encoder, an attention-based ASR decoder, and an attention-based SLU decoder. A multi head attention module is used to incorporate acoustic representations from the ASR encoder and text representations from the ASR decoder. We also explored using adapter modules in the ASR encoder and ASR decoder to further improve training efficiency.

ASR Encoder and Decoder. The conformer encoder maps input filter bank feature sequence $\mathbf{X} = (x_1, x_i, .., x_T)$ to an acoustic representation $\mathbf{h^{enc}} = (h_1^{enc}, ..., h_i^{enc}, h_n^{enc})$ where $\mathbf{h}^{enc} \in \mathbb{R}$, $\mathbf{T}$ is the number of acoustic frames, and $\mathbf{h}^{dec} \in \mathbb{R}$ where $\mathbf{n}$ is the number of encoder output.

$$\mathbf{h}^{enc} = Encoder_{asr}(\mathbf{X}) \quad (1)$$

$$\mathbf{h}_u^{dec} = AttentionDecoder_{asr}(\mathbf{h}^{enc}, \mathbf{y}_{1:u-1}^{asr}) \quad (2)$$

Given $\mathbf{h}^{enc}$ the transformer-based ASR decoder autoregressively predicts symbols one at time as follows:

$$P(y_u^{asr}|\mathbf{X}, \mathbf{y}_{1:u-1}^{asr}) = Softmax(Linear(\mathbf{h}_u^{dec})) \quad (3)$$

Fusion Module. We keep ASR transcriptions while predicting semantics in order to reuse components and make the entire end to end speech pipeline more efficient. Inspired by [15, 17], we combine the acoustic representation $\mathbf{h^{enc}}$ from the ASR encoder and the text representation $\mathbf{h^{dec}}$ from the ASR decoder using a multi-head attention (MHA) module. The MFA module uses acoustic representation $\mathbf{h^{enc}}$ as a query and text representation $\mathbf{h^{dec}}$ as key and value. The output of the MFA module is fed to a linear layer, resulting in a joint representation $\mathbf{h^{joint}}$. This joint representation is used in the SLU decoder along with previous SLU labels.

$$\mathbf{h}^{attn} = \mathbf{h}^{enc} + MHA(\mathbf{h}^{enc}, \mathbf{h}^{dec}) \quad (4)$$

$$\mathbf{h}^{joint} = Linear(\mathbf{h}^{attn}) \quad (5)$$

SLU decoder. Previous work used large pretrained NLU models [16] to predict the intent and slot labels, we chose a smaller attention based decoder for SLU. The SLU decoder predicts intents and slots conditioned on joint representation $\mathbf{h}^{joint}$ and previous SLU labels, which can be written as

$$\mathbf{h}_v^{slu} = AttentionDecoder_{slu}(\mathbf{h}^{joint}, \mathbf{y}_{1:v-1}^{slu}) \quad (6)$$

$$P(y_v^{slu}|\mathbf{h}^{joint}, \mathbf{y}_{1:v-1}^{slu}) = Softmax(Linear(\mathbf{h}_v^{slu})) \quad (7)$$

where $v \in V$ for all possible intents and slot values.

The multi-task loss (MTL) of the model is a weighted sum of the negative log likelihoods from the ASR and SLU tasks. This MTL approach enables accurate estimation of both transcript and semantic information by jointly optimizing the ASR decoder, the SLU decoder, and the shared encoder.

$$\begin{aligned} L_{MTL} = &-(1-\alpha)\Sigma_u lnP(y_u^{asr}|x, y_{1:u-1}^{asr}) \\ &-\alpha\Sigma_v lnP(y_v^{slu}|x, y_{1:v-1}^{slu}) \end{aligned} \quad (8)$$

where α is a scaling factor for balancing the ASR loss and the SLU loss. In order to enabe the model to estimate transcript and semantic simultaneously, we use MTL to jointly train the ASR decoder, SLU decoder and shared encoder.

Adapter. Recently, the adapters [21, 23, 24, 25] have been extensively studied for fine-tuning a model on a specific task or domain by updating or adding a small number of parameters, rather than fine-tuning all the parameters of the pre-trained model. In this paper, we investigate its effectiveness for semantic and transcript predictions by incorporating adapters in the pre-trained ASR encoder and ASR decoder modules. We assume that adapter based transfer learning on pretrained ASR model is beneficial for domain-specific ASR adaptation as well as corresponding semantic prediction. To apply the adapter modules, it is inserted into each layer of pre-trained transformer layers; specifically, after the self-attention and position-wise feed-forward networks. The adapter module generally uses a down-projection linear layer to project the input to a lower-dimensional space, followed by a nonlinear activation function, and an up-projection linear layer. These adapters are surrounded by a residual connection.

2.2. Training Strategy

2.2.1. Pre-trained ASR Model

We firstly train E2E ASR model which include ASR encoder and ASR decoder on public Librispeech [26], as in [12, 7, 11, 9, 18]. The E2E ASR model is then optionally fine-tuned using SLU dataset.

2.2.2. Adaptation for ASR and SLU with multi-task learning

To efficiently adapt E2E SLU, we present three approaches detailed below.

	FSC	SLURP	SLURP-synth	Samsung
Duration [hrs]	19	58	43.5	90
Audio files	30,043	727,277	69,253	107,564
Scenarios	0	18	18	10
Actions	0	46	54	73
Entities	0	56	56	162
Intents	31	69	69	109

Table 1: *Statistics of FSC, SLURP, and Samsung in-house datasets*

Full model. The approach involves updating all three components which are ASR encoder, decoder, and SLU decoder using multi-task learning.

SLU Decoder. The ASR encoder and decoder are kept frozen, while the fusion module and the SLU decoder are updated.

Adapter. The approach entails training the fusion module, SLU decoder, and adapters while keeping the ASR encoder and decoder frozen using multi-task learning.

3. EXPERIMENT

3.1. Datasets

Experiments are conducted on three SLU datasets: the FSC [7], SLURP [22] and Samsung in-house SLU dataset. FSC is a dataset that comprises spoken commands for a virtual assistant, with around 30,000 utterances split into training, validation, and testing subsets with 23, 132, 3, 118, and 3793 utterances respectively. The SLURP dataset, which has been used in various recent studies [16, 19, 27, 18, 6, 15] for an in-home personal robot assistant. It comprises three levels of semantics: Scenario, Action and Entities. We define intent as scenarios combined with their respective actions as in [19]. A statistics of the dataset is depicted in Table 1. The Samsung in-house SLU dataset consists of 90 hour English of de-identified Samsung in-house SLU data. All samples are recorded in 16Khz sampling rate. The Samsung in-house SLU dataset terminology and expression are described in Figure 2. In terms of semantics, our terminology and SLURP are interchangeable. In other words, we can exchange Capsule with Scenario, Goal with Action, and Slot with Entity. We use SLURP terms in this paper unless otherwise noted. The data includes 10 scenarios, 73 actions, 162 entities and 109 intents as shown in Table 1. The dataset is split into 70%, 15% and 15% for train, dev, and test respectively. Sampling rate for all audios is 16kHz.

3.2. Performance Metrics

Intent Accuracy (IntAcc). IntAcc is defined as the percentage (%) of utterances that the model accurately predicted the intent.

SLU-F1. The performance of slot filling is evaluated using the SLU-F1 metric [22]. It combines span based F1 score in named entity recognition with a text based distance measure to accommodate ASR errors. For each reference slot: (1) True Positive (TP) if the slot name and the value match, (2) False Negative (FN) if it is a slot deletion error, and (3) False Positive (FP) if it is a slot insertion error. A slot substitution error counts as TP with penalty that equals to word and character error rates of the slot value, adding to FN and FP values. SLU-F1 refers to the F1 score of these. We use Word Error Rate (WER) as ASR metric which is defined as the normalized minimum word edit distance.

The aim of this paper is to propose an approach that achieve a high IntAcc, SLU-F1, and a low WER with a small number of trainable parameters.

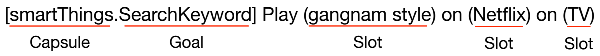
[smartThings.SearchKeyword] Play (gangnam style) on (Netflix) on (TV)
Capsule
Goal
Slot
Slot
Slot

Figure 2: *An example of Samsung SLU data*

3.3. Model configurations

We use an 80-dimensional log-mel filterbank feature computed over a 25 ms sliding window with stride of 10 ms. We optimized an E2E ASR AED [28, 29] with a hybrid of CTC loss and attention loss [29] using the Librispeech 960-hour dataset [26]. Our E2E SLU consists of Conformer [30] (K=17) encoder layers with dimension of 512 and (M=6) ASR transformer decoder layers with 1024 dimension and (J=3) SLU transformer decoder layers with 1024. The fusion model use 4-head 512 attention. Transcripts and semantics are tokenized into subword tokens using sentencepiece (BPE) [31] with vocabulary of 1024 and 64 for ASR and SLU, respectively. We use data augmentation using SpecAugment [32] and used 0.1 label smoothing [33] only for the ASR decoder. The decoding beam size was set to 10 for all experiments. For multi task learning we choose an α value of 0.5. For adapter, we use linear layers with 32 dimension and Swish activation function [34].

3.4. Results

As we described in section 2.2.2, there are three different experiments to adapt the SLU model. The first method involves updating the shared encoder, the ASR decoder, the SLU decoder, and the fusion MHA model. We refer to this experiment as ***ALL***. The second method updates the fusion module and the SLU decoder, called as ***SLU decoder***. The final method, called ***SLU decoder + adapter***, involves updating the fusion module, SLU decoder, and adapters inserted in the shared encoder and ASR decoder. All experiments are based on pre-trained ASR model and multi task loss. Our ***ALL*** model and ***SLU decoder*** module outperform the baseline models for SLURP corpus in terms of IntACC and WER, shown in Table 3 M1 and M3. Our ***SLU decoder*** model and ***SLU decoder + adapter*** model outperform existing competitive baselines [19, 20, 16] in both intACC and WER for the FSC corpus Table 2. Since FSC is a relatively simple dataset we use the SLURP and Samsung dataset to showcase the effectiveness of ASR fine-tuning [P1-P3].

Table 3 shows a comparison of the metric scores of our proposed model with that of other state of the art models such as branchformer [27], RNNT-BERT [19], SC-Mask-CTC [6], and Nemo-SLU [35]. All of these architectures use the same SLURP dataset and the same metrics that we used to train and test our model. We present various adaptation techniques conducted using SLURP dataset. We present an experiment in which all parameters of the randomly initialized shared encoder,

	Method	IntAcc	WER
B5	Two pass SLU [19]	98.10	-
B6	CMCL [20]	99.69	6.50
B7	CTI [16]	99.70	-
M1	ASR PT→SLU Adapt (SLU Decoder)	75.45	8.50
M2	ASR PT→SLU Adapt (SLU Decoder+Adapter)	**99.71**	**2.52**
M3	ASR PT→SLU Adapt (All)	99.66	3.37

Table 2: *Comparison of performances in terms of IntAcc and WER on FSC dataset*

	Method	SLURP			Samsung		
		IntAcc	SLU-F1	WER	IntAcc	SLU-F1	WER
B1	Branchformer [27]	88.1	77.7	-	-	-	-
B2	SC-Mask-CTC [6]	89.1	77.5	16.8	-	-	-
B3	NEMO-SLU [35]	90.3	81.3	-	-	-	-
B4	SLU Adapt (from scratch)	74.5	55.6	20.1	90.0	76.6	13.2
P1	ASR PT→ASR FT→SLU Adapt (SLU decoder)	90.3	83.5	**8.1**	92.8	84.1	1.7
P2	ASR PT→ASR FT→SLU Adapt (SLU Decoder + Adapter)	90.0	83.7	8.4	**93.0**	**85.1**	**1.1**
P3	ASR PT→ASR FT→SLU Adapt (All)	**90.4**	**84.4**	9.0	91.3	82.2	1.8
M1	ASR PT→SLU Adapt (SLU Decoder)	84.9	77.6	28.6	90.7	81.1	23.8
M2	ASR PT→SLU Adapt (SLU Decoder + Adapter)	89.0	81.7	11.9	91.9	82.0	12.9
M3	ASR PT→SLU Adapt (All)	89.4	83.5	9.5	92.2	83.7	2.0

Table 3: *Comparison of performances in terms of IntAcc, SLU-F1, and WER on SLURP and Samsung In-house dataset*

	Method	Trainable Params	Training Speed #uttrs/seconds
M1	ASR PT→SLU Adapt (SLU Decoder)	9.6	1015
M2	ASR PT→SLU Adapt (SLU Decoder+Adapter)	11.9	614
M3	ASR PT→SLU Adapt (All)	135	399

Table 4: *Efficiency comparisons in terms of trainable params and tranining speed*

Method	IntAcc	SLU-F1	WER
P3	90.4	84.4	9.0
-Fusion module	89.5	84.0	9.5
- ASR FT	89.0	81.8	15.5
- ASR PT	74.5	55.6	20.1

Table 5: *Ablation study about the effect of fusion model and pre-training and fine-tuning ASR*

ASR decoder, SLU decoder, and fusion module are trained with MTL loss as a baseline [B4]. We can see that except for [M1], the results of [B4] experiment perform worse than all other experiments based on pre-trained ASR. This means that the pre-training ASR model is crucial for E2E SLU task. We conducted experiments [P1-P3], where the ASR model was pre-trained with domain-independent data, fine-tuned with target domain data, and then fine-tuned for SLU. Additionally, we conducted experiments [M1-M3] without fine-tuning the ASR model with target domain data. Firstly, we observe that domain specific fine-tune process is important. All three SLU adaptation methods [P1-P3] perform better than a model [M1-M3] without domain specific fine-tuning for ASR. Furthermore, based on the [P1] experiment, it can be observed that if the pre-trained ASR model is adapted to the target domain, it can achieve good SLU intent accuracy and SLU F1 performance by solely updating the SLU decoder. However, if the SLU decoder is trained using a model that is not fine-tuned with target domain data [M1], poor SLU performance is observed, and WER is significantly degraded. Among the experiments [M1-M3] where domain-specific ASR fine-tuning was not performed, it was observed that the [M3] model, which updates the entire parameters using both ASR and SLU losses, showed the best performance in terms of intACC, SLU F1, and WER. It is worth mentioning that using the method [M2], which trains only the SLU decoder and adapter without updating all parameters, still achieves competitive performance over the baseline. Also as we can see in Table 4, the [M2] approach can efficiently train the model 1.5 times faster with 11 times fewer parameters than the [P2] model, which requires updating 135M parameters, maintaining 89.4% intAcc and 83.5% SLU F1 score, which are competitive performance compared to the baselines.

Our experiments consisted of pre-trained ASR model which was just adapted or fine-tuned and adapted with SLU data. The adaptation step had 3 experiments: (1) Shared encoder, ASR decoder, SLU decoder adaptation, (2) SLU decoder adaptation (3) SLU decoder, Adapter module adaptation. The results indicated that the best IntAcc and SLU-F1 were achieved when a pre-trained ASR encoder and decoder model were fine-tuned using a small amount of target SLU dataset and then that ASR encoder and decoder are adapted along with the initialized SLU decoder. Although the WER is optimal when only SLU decoder is adapted after fine-tuning pre-trained ASR model with small sample of SLU dataset.

Figure 3 shows the [M1] model achieving similar accuracy as [M2] model despite retraining 10 times less number of parameters as seen in Table 4. The final performance for the [M2] model is only slightly less compared to the [M1] model and is still competitive against state of the art.

3.4.1. Ablation Study

In order to better understand the influence of different method in the proposed model on the overall performance, we conducted an ablation study over the fusion module and training strategy. We use the the proposed model (section 2.1) and report IntAcc, SLU-F1 and WER for each of the methods. Results are presented in Table 5. The results indicate that incorporating a fusion component and fine-tuning ASR enhance the overall performance, with ASR pre-training yielding the most significant performance improvement.

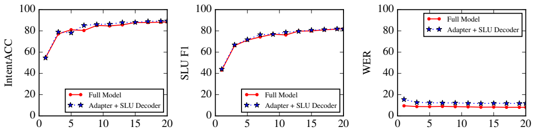

Figure 3: *IntAcc, SLU F1 and WER of Full model and Adapter + SLU Decoder (X-axis: epoch)*

4. CONCLUSION

In this paper, we explored adaptation of an E2E SLU model based on an ASR model through different approaches. The proposed E2E SLU model, which integrates a fusion module and an additional SLU decoder, outperform existing competitive baselines on ASR and SLU tasks when it is updated based on pre-trained ASR models. Furthermore, our approaches achieved state-of-the-art results on the FSC and SLURP dataset by jointly fine-tuning the entire SLU model based on a task-specific pre-trained ASR model, and shown competitive performance using an adapter without additional task-specific fine-tuning.

5. References

[1] G. Tur and R. De Mori, *Spoken language understanding: Systems for extracting semantic information from speech*. John Wiley & Sons, 2011.

[2] G. Mesnil, Y. Dauphin, K. Yao, Y. Bengio, L. Deng, D. Hakkani-Tur, X. He, L. Heck, G. Tur, D. Yu *et al.*, "Using recurrent neural networks for slot filling in spoken language understanding," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 3, pp. 530–539, 2014.

[3] D. Serdyuk, Y. Wang, C. Fuegen, A. Kumar, B. Liu, and Y. Bengio, "Towards end-to-end spoken language understanding," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[4] P. Haghani, A. Narayanan, M. Bacchiani, G. Chuang, N. Gaur, P. Moreno, R. Prabhavalkar, Z. Qu, and A. Waters, "From audio to semantics: Approaches to end-to-end spoken language understanding," in *IEEE Spoken Language Technology Workshop (SLT)*, 2018.

[5] M. Radfar, A. Mouchtaris, S. Kunzmann, and A. Rastrow, "Fans: Fusing asr and nlu for on-device slu," *INTERSPEECH 2021*.

[6] M. Li and R. Doddipatla, "Non-autoregressive end-to-end approaches for joint automatic speech recognition and spoken language understanding," in *IEEE Spoken Language Technology Workshop (SLT)*, 2023.

[7] L. Lugosch, M. Ravanelli, P. Ignoto, V. S. Tomar, and Y. Bengio, "Speech model pre-training for end-to-end spoken language understanding," *INTERSPEECH 2019*.

[8] S. Thomas, H.-K. J. Kuo, B. Kingsbury, and G. Saon, "Towards reducing the need for speech training data to build spoken language understanding systems," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

[9] 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 *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.

[10] M. Dinarelli, N. Kapoor, B. Jabaian, and L. Besacier, "A data efficient end-to-end spoken language understanding architecture," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[11] M. Rao, A. Raju, P. Dheram, B. Bui, and A. Rastrow, "Speech to semantics: Improve asr and nlu jointly via all-neural interfaces," *INTERSPEECH 2020*.

[12] N. Tomashenko, A. Caubrière, Y. Estève, A. Laurent, and E. Morin, "Recent advances in end-to-end spoken language understanding," in *Statistical Language and Speech Processing (SLSP)*. Springer, 2019, pp. 44–55.

[13] C.-I. Lai, Y.-S. Chuang, H.-Y. Lee, S.-W. Li, and J. Glass, "Semi-supervised spoken language understanding via self-supervised speech and language model pretraining," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.

[14] M. Rao, P. Dheram, G. Tiwari, A. Raju, J. Droppo, A. Rastrow, and A. Stolcke, "Do as i mean, not as i say: Sequence loss training for spoken language understanding," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 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," in *INTERSPEECH 2022*.

[16] S. Seo, D. Kwak, and B. Lee, "Integration of pre-trained networks with continuous token interface for end-to-end spoken language understanding," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

[17] D. Le, A. Shrivastava, P. Tomasello, S. Kim, A. Livshits, O. Kalinli, and M. L. Seltzer, "Deliberation model for on-device spoken language understanding," *INTERSPEECH 2022*.

[18] Z. Huang, M. Rao, A. Raju, Z. Zhang, B. Bui, and C. Lee, "Mtl-slt: multi-task learning for spoken language tasks," in *Proceedings of the 4th Workshop on NLP for Conversational AI*, 2022.

[19] S. Arora, S. Dalmia, X. Chang, B. Yan, A. W. Black, and S. Watanabe, "Two-Pass Low Latency End-to-End Spoken Language Understanding," in *INTERSPEECH 2022*.

[20] J. Dong, J. Fu, P. Zhou, H. Li, and X. Wang, "Improving spoken language understanding with cross-modal contrastive learning," *INTERSPEECH 2022*.

[21] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for nlp," in *International Conference on Machine Learning*, 2019.

[22] E. Bastianelli, A. Vanzo, P. Swietojanski, and V. Rieser, "SLURP: A spoken language understanding resource package," in *Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2020.

[23] W. Hou, H. Zhu, Y. Wang, J. Wang, T. Qin, R. Xu, and T. Shinozaki, "Exploiting adapters for cross-lingual low-resource speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 317–329, 2021.

[24] J. Peng, T. Stafylakis, R. Gu, O. Plchot, L. Mošner, L. Burget, and J. Černockỳ, "Parameter-efficient transfer learning of pre-trained transformer models for speaker verification using adapters," *arXiv preprint arXiv:2210.16032*, 2022.

[25] B. Thomas, S. Kessler, and S. Karout, "Efficient adapter transfer of self-supervised speech models for automatic speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

[26] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *IEEE International conference on acoustics, speech and signal processing (ICASSP)*, 2015.

[27] Y. Peng, S. Dalmia, I. Lane, and S. Watanabe, "Branchformer: Parallel mlp-attention architectures to capture local and global context for speech recognition and understanding," in *International Conference on Machine Learning*, 2022.

[28] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *IEEE International conference on acoustics, speech and signal processing (ICASSP)*, 2016.

[29] 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.

[30] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *INTERSPEECH 2020*.

[31] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Association for Computational Linguistics*, 2016.

[32] D. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. Cubuk, and Q. Le, "Deliberation model for on-device spoken language understanding," *INTERSPEECH 2019*.

[33] R. Müller, S. Kornblith, and G. E. Hinton, "When does label smoothing help?" *Advances in Neural Information Processing Systems*, 2019.

[34] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," *arXiv preprint arXiv:1710.05941*, 2017.

[35] O. Kuchaiev, J. Li, H. Nguyen, O. Hrinchuk, R. Leary, B. Ginsburg, S. Kriman, S. Beliaev, V. Lavrukhin, J. Cook *et al.*, "Nemo: a toolkit for building ai applications using neural modules," *arXiv preprint arXiv:1909.09577*, 2019.