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


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 pre-trained 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.
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 $X = (x_1, x_2, ..., x_T)$ to an acoustic representation $h^{enc} = (h_1^{enc}, ..., h_n^{enc}, h_1^{enc})$ where $h^{enc} \in \mathbb{R}^T$ is the number of acoustic frames, and $h^{dec} \in \mathbb{R}$ where $n$ is the number of encoder output.

\begin{align*}
    h^{enc} &= Encoder_{asr}(X) \\
    h^{dec} &= AttentionDecoder_{asr}(h^{enc}, y^{asr}_{1:t-1})
\end{align*}

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

\begin{align*}
    P(y^{asr}_t|X, y^{asr}_{1:t-1}) = \text{Softmax}(\text{Linear}(h^{dec}_u))
\end{align*}

**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 $h^{asr}$ from the ASR encoder and the text representation $h^{asr}$ from the ASR decoder using a multi-head attention (MHA) module. The MFA module uses acoustic representation $h^{asr}$ as a query and text representation $h^{asr}$ as key and value. The output of the MFA module is fed to a linear layer, resulting in a joint representation $h^{joint}$. This joint representation is used in the SLU decoder along with previous SLU labels.

\begin{align*}
    h^{attn} &= h^{enc} + MHA(h^{enc}, h^{dec}) \\
    h^{joint} &= \text{Linear}(h^{attn})
\end{align*}

**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 $h^{joint}$ and previous SLU labels, which can be written as

\begin{align*}
    h^{slu}_{\text{attention}} &= \text{AttentionDecoder}_{slu}(h^{joint}, y^{slu}_{1:v-1}) \\
    P(y^{slu}_v|h^{joint}, y^{slu}_{1:v-1}) &= \text{Softmax}(\text{Linear}(h^{slu}_u))
\end{align*}

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{align*}
    L_{\text{MTL}} &= -(1 - \alpha)\sum_u \log P(y^{asr}_u|x, y^{asr}_{1:t-1}) \\
    &- \alpha\sum_v \log P(y^{slu}_v|x, y^{slu}_{1:v-1})
\end{align*}

where $\alpha$ is a scaling factor for balancing the ASR loss and the SLU loss. In order to enable 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.
Table 1: Statistics of FSC, SLURP, and Samsung in-house datasets

<table>
<thead>
<tr>
<th></th>
<th>FSC</th>
<th>SLURP</th>
<th>SLURP-synth</th>
<th>Samsung</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duration [hrs]</td>
<td>19</td>
<td>58</td>
<td>4:53</td>
<td>90</td>
</tr>
<tr>
<td>Audio files</td>
<td>30,043</td>
<td>727,277</td>
<td>69,253</td>
<td>107,564</td>
</tr>
<tr>
<td>Scenarios</td>
<td>0</td>
<td>18</td>
<td>18</td>
<td>17</td>
</tr>
<tr>
<td>Actions</td>
<td>0</td>
<td>46</td>
<td>14</td>
<td>73</td>
</tr>
<tr>
<td>Entities</td>
<td>0</td>
<td>56</td>
<td>56</td>
<td>162</td>
</tr>
<tr>
<td>Intents</td>
<td>31</td>
<td>69</td>
<td>49</td>
<td>109</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Method</th>
<th>IntAcc</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>B0</td>
<td>98.10</td>
<td>-</td>
</tr>
<tr>
<td>B0</td>
<td>99.09</td>
<td>-</td>
</tr>
<tr>
<td>B7</td>
<td>99.70</td>
<td>-</td>
</tr>
<tr>
<td>M1</td>
<td>78.45</td>
<td>8.50</td>
</tr>
<tr>
<td>M2</td>
<td>99.71</td>
<td>2.52</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Method</th>
<th>SLURP</th>
<th>Samsung</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IntAcc</td>
<td>SLU-F1</td>
</tr>
<tr>
<td>B1  Branchformer [27]</td>
<td>88.1</td>
<td>77.7</td>
</tr>
<tr>
<td>B2  NL-Mask-CTC [6]</td>
<td>89.1</td>
<td>77.9</td>
</tr>
<tr>
<td>B3  NEMO-SLU [35]</td>
<td>90.3</td>
<td>81.3</td>
</tr>
<tr>
<td>B4  SLU Adapt (from scratch)</td>
<td>74.3</td>
<td>55.6</td>
</tr>
<tr>
<td>F1  ASR PT + ASR F1 + SLU Adapt (SLU decoder)</td>
<td>89.3</td>
<td>85.5</td>
</tr>
<tr>
<td>F2  ASR PT + ASR F1 + SLU Adapt (SLU Decoder + Adapter)</td>
<td>90.6</td>
<td>83.7</td>
</tr>
<tr>
<td>F3  ASR PT + ASR F1 + SLU Adapt (All)</td>
<td>90.4</td>
<td>84.4</td>
</tr>
<tr>
<td>M1  ASR PT + SLU Adapt (SLU Decoder)</td>
<td>82.9</td>
<td>77.6</td>
</tr>
<tr>
<td>M2  ASR PT + SLU Adapt (SLU Decoder + Adapter)</td>
<td>89.0</td>
<td>81.7</td>
</tr>
<tr>
<td>M3  ASR PT + SLU Adapt (All)</td>
<td>89.4</td>
<td>83.5</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Method</th>
<th>Trainable Params</th>
<th>Training Speed #uttrs/seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1  ASR PT + SLU Adapt (SLU Decoder)</td>
<td>9.6</td>
<td>1015</td>
</tr>
<tr>
<td>M2  ASR PT + SLU Adapt (SLU Decoder + Adapter)</td>
<td>11.9</td>
<td>614</td>
</tr>
<tr>
<td>M3  ASR PT + SLU Adapt (All)</td>
<td>135</td>
<td>399</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Method</th>
<th>IntAcc</th>
<th>SLU-F1</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1  Fusion module</td>
<td>90.4</td>
<td>84.4</td>
<td>9.0</td>
</tr>
<tr>
<td>- ASR PT + SU F1</td>
<td>89.5</td>
<td>84.0</td>
<td>9.5</td>
</tr>
<tr>
<td>- ASR PT + SU F1 + SLU (All)</td>
<td>89.0</td>
<td>81.8</td>
<td>15.5</td>
</tr>
<tr>
<td>- ASR PT + SU F1 + SLU (All)</td>
<td>74.5</td>
<td>55.6</td>
<td>20.1</td>
</tr>
</tbody>
</table>

3.4.1. Ablation Study

In order to better understand the influence of different methods 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, outperforms 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 showed competitive performance using an adapter without additional task-specific fine-tuning.
5. References


