On joint training with interfaces for spoken language understanding

Anirudh Raju, Milind Rao*, Gautam Tiwari, Pranav Dheram, Bryan Anderson, Zhe Zhang, Chul Lee, Bach Bui, Ariya Rastrow*

Amazon Alexa AI, USA

ranirudh@amazon.com, milinrao@amazon.com, tgautam@amazon.com

Abstract

Spoken language understanding (SLU) systems extract both text transcripts and semantics associated with intents and slots from input speech utterances. SLU systems usually consist of (1) an automatic speech recognition (ASR) module, (2) an interface module that exposes relevant outputs from ASR, and (3) a natural language understanding (NLU) module. Interfaces in SLU systems carry information on text transcriptions or richer information like neural embeddings from ASR to NLU. In this paper, we study how interfaces affect joint-training for spoken language understanding. Most notably, we obtain the state-of-the-art results on the publicly available 50-hr SLURP [1] dataset. We first leverage large-size pretrained ASR and NLU models that are connected by a text interface, and then jointly train both models via a sequence loss function. For scenarios where pretrained models are not utilized, the best results are obtained through a joint sequence loss training using richer neural interfaces. Finally, we show the overall diminishing impact of leveraging pretrained models with increased training data size.

Index Terms: speech recognition, spoken language understanding, neural interfaces, multitask training

1. Introduction

Spoken dialog systems enable voice-based human-machine interactions. A key component of any spoken dialog system is its spoken language understanding (SLU) system that extracts semantic information associated with intents and named-entities from the user's speech utterances. The task of SLU systems is modeled through two separate subtasks, namely ASR and NLU. ASR generates text transcripts from input speech while NLU extracts semantic information from text transcripts. NLU, in turn, performs two subtasks, namely intent determination and slot filling. In the conventional SLU setting, ASR and NLU components are built and deployed independent of each other. They are sequentially executed with NLU consuming text transcript outputs from ASR. An overall trend towards end-to-end neural architectures is allowing on-device deployment with model sizes tuned to comply with different hardware constraints in addition to a tighter coupling between ASR and NLU.

End-to-end speech recognition models like attention-based listen-attend-spell (LAS) [2], transformers [3] and RNN-Transducer (RNNT) [4] have been shown to outperform traditional RNN-HMM hybrid ASR systems, especially when trained on large speech corpora [5]. For real-time ASR systems, streaming compatible RNNT is the most suitable model choice [6]. Recurrent neural network [7], recursive neural network [8], and transformer [9] based NLU models have been shown to be effective for multi-task intent classification and named entity recognition.

The easiest way of designing a full SLU sytem is to run ASR and NLU modules in sequence. Even though such a design approach is simple to implement and practical, such pipelined SLU systems are prone to a potential downstream propagation of ASR errors or an overall SLU performance degradation as each task is trained completely independent of its upstream or downstream model. This motivated the emergence of end-to-end SLU models that directly extract serialized semantic information from speech inputs [10–15]. While ASR transcripts have been typically considered as the standard interface choice in such end-to-end SLU systems, other interfaces like word confusion networks [16], lattices [17], or the n-best hypotheses [18] have been recently proposed.

Since the inception of end-to-end SLU models, several attempts have been made to improve the overall SLU performance or overcome the limitations of existing end-to-end SLU systems. For those models that do not produce ASR transcripts, pretraining with an ASR task or masked language model has shown to be beneficial [12, 13]. Integrating NLU intent signals into an ASR system was studied in [19]. Joint or multi-task training of ASR and NLU has shown to improve the overall SLU performance in [10, 11]. Neural interfaces (e.g. decoder hidden layer interfaces) to better facilitate the joint training of ASR and NLU were introduced in [11]. To design more streaming friendly ASR, an RNNT based SLU system to directly predict serialized semantics from audio was proposed in [20].

A variety of loss functions have been used for the training of different ASR, NLU and SLU models. For instance, cross-entropy or RNNT loss functions have been used in these models even though their metrics of interest are word error rate (WER) or semantic error rate (SemER). The REINFORCE framework in [21] enables model training in a way that the probabilities of hypotheses are boosted if they perform well on arbitrary chosen metrics. Sequence-discriminative criteria such as minimum word, phone error or minimum Bayes risk were used for ASR training in [22]. Motivated by mWER-based ASR training, using non-differentiable semantic criteria to directly optimize SLU metrics for training was shown to be effective in [14, 23–25].

1.1. Contributions

While several joint training methods for SLU have been proposed in the past, it is not always straightforward to decide which interfaces are suitable for different types of ASR and NLU models. Our work is the first of its kind to fill in this research gap in this field since we study the effectiveness of interfaces, as well as the impact of pre-training, during SLU joint training. Our main contributions in this paper are:

Joint Training with Interfaces: We observe in various experiments that when joint training for SLU is performed with well designed interfaces, ASR and NLU metrics can be significantly improved, ranging from 5% to 15%. Notably, we obtain the state-of-the-art results on the SLURP dataset by using pretrained ASR and NLU models and then jointly training via a text interface.

Novel Neural Interfaces: Most prior SLU models are based on simple text interfaces, with the exception of [11,14,26]

*Equal Contribution

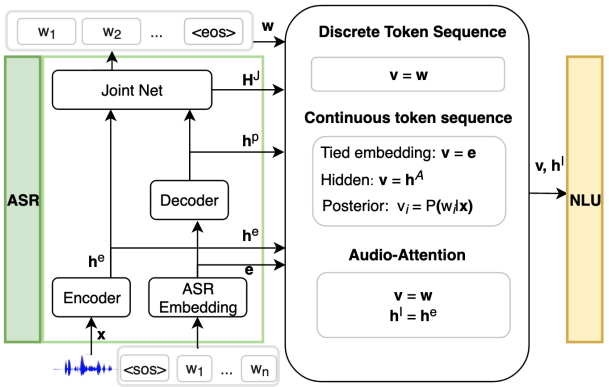

Figure 1: *A SLU pipeline comprising of ASR (e.g. RNNT), blackbox NLU and different interface types to connect ASR and NLU for joint training.*

that use neural interfaces, but only limited to encoder-decoder ASR models. In this paper, we propose a broader set of new neural interfaces that are particularly suitable for transducer ASR models like RNNT and Transformer-Transducer [3].

Effectiveness of Pretraining: For small datasets like SLURP, we observe that pretrained NLU models have a strong impact on SLU performance. In contrast, for large datasets, we observe that pretrained NLU models do not show as much significant impact.

2. Methods

Notation: The input audio feature sequence is $\mathbf{x} = (x_1, ..., x_T)$, $x_i \in \mathbb{R}^{|X|\times 1}$. The ground truth transcript of N words is $\mathbf{y} = (y_1, ..., y_N)$ where $y_i \in \mathcal{Y}$, the discrete set of words in the vocabulary. The ground truth NLU labels include a slot tag for each ground truth word, i.e. the discrete slot label sequence $\mathbf{y}^{slot} = (y_1^{slot}, ..., y_N^{slot})$, and a label for the utterance's intent, i.e. y^{int}.

2.1. Formulation of ASR/NLU Models and their Interfaces

ASR: E2E ASR models decode audio sequence $\mathbf{x}$ and produce a U length output token sequence $\mathbf{w} = (w_1, ..., w_U)$ that corresponds to the best decoding hypothesis, where $w_u \in \mathcal{W}$, the set of discrete output labels of subword units. We consider two types of models that have been widely used for E2E ASR.

RNN-Transducer: RNNT [4] is a streaming compatible model that consists of three networks. An RNN encoder, analogous to an acoustic model, maps input audio feature sequence $\mathbf{x}$ to hidden representations $\mathbf{h}^e = (h_1^e, ..., h_T^e)$, $h_t^e \in \mathbb{R}^{|E|\times 1}$. The prediction network takes as input the previous output label prediction w_{u-1}, maps to a token embedding e_i, and corresponding hidden representations $h_u^p \in \mathbb{R}^{|P|\times 1}$. The joint network, typically a feed-forward neural network [27], takes the encoder representation h_t^e and the prediction network representation h_u^p as input, produces intermediate joint hidden layer representations for each t, u pair $h_{t,u}^J \in \mathbb{R}^{|J|\times 1}$, along with corresponding logits and softmax normalized probabilities. The intermediate joint hidden layer outputs for all audio and token inputs can be represented as a rank-3 tensor $\mathbf{H}^J \equiv h_{t,u}^J$, $\mathbf{H}^J \in \mathbb{R}^{T\times U\times |J|}$.

Listen-Attend-Spell (LAS): LAS [2] is a non-streaming model that consists of a recurrent encoder, and a recurrent decoder using an attention mechanism. The encoder maps input audio feature sequence $\mathbf{x}$ to hidden representations $\mathbf{h}^e = (h_1^e, ..., h_T^e)$. The decoder uses an attention mechanism [28] to attend to the encoder representations, producing decoder hidden representations $\mathbf{h}^d = (h_1^d, ..., h_U^d)$, and output token probabilities $P(w_i|\mathbf{x})$.

Table 1: *Summary of various ASR - NLU interfaces. All interfaces can leverage pretrained ASR models*

Interface	Joint Training	Pretrained NLU	Interface Type
Text	Seq-loss	Y	Discrete token seq
Tied embeddings	MLE + Seq-loss	N	Continuous token seq
Posterior	MLE + Seq-loss	Y	
Hidden	MLE + Seq-loss	N	
Audio-Attention	MLE + Seq-loss	N	Discrete token seq + Audio representations

ASR-NLU Interface for SLU: The role of an interface for SLU is to process ASR model outputs along with intermediate representations, and then subsequently produce inputs that are compatible with the NLU model. Formally, the interface output described in Eqn 1 comprises the pair of (1) discrete or continuous feature, $\mathbf{v} = (v_1, ..., v_U)$, of length U (2) additional hidden representations, $\mathbf{h}^I$, as an option.

$$\mathbf{v}, \mathbf{h}^I = Interface(\mathbf{H}_{asr}, \mathbf{w}) \quad (1)$$

where $\mathbf{H}_{asr}$ refers to exposed hidden representations from ASR. For RNNT, this comprises the set of encoder, prediction and joint network hidden representations: $\mathbf{H}_{asr} = \{\mathbf{h}^e, \mathbf{h}^p, \mathbf{H}^J\}$. For LAS, this includes the encoder and decoder final hidden layer representations: $\mathbf{H}_{asr} = \{\mathbf{h}^e, \mathbf{h}^d\}$. Interfaces are further elaborated in Sec 2.2.

NLU: A neural NLU model predicts the true slot label sequence of $\mathbf{y}^{slot}$ and intent label y^{int}. The model takes as input $\mathbf{v}, \mathbf{h}^I$ from the given interface. The discrete or continuous sequence of features $\mathbf{v}$ is utilized as inputs to transformer NLU (TNLU) model with multiple layers of self-attention. The layers can be initialized using pretrained models like BERT [29]. When an interface produces additional representations $\mathbf{h}^I$ in addition to $\mathbf{v}$, NLU can attend to these representations. A transformer-decoder [30] is suitable for this purpose, and consists of stacked layers, each with a multi-head attention block to perform self-attention over $\mathbf{v}$ (i.e. query, key, value is $\mathbf{v}$), and a multi-head attention block that cross-attends $\mathbf{h}^I$.

2.2. Interfaces and Joint Training

We introduce various interfaces below, and summarize them in Table 1. Fig 1 depicts these interfaces in detail. All interfaces except the text interface in Sec 2.2.1 will be labeled as "neural" for the rest of the paper since these are all neural model based.

2.2.1. Discrete Token Sequence: Text Interface

This is mainly suitable for conventional SLU systems, in which ASR produces one-best text transcripts consumed by NLU. That is, ASR's one-best label sequence $\mathbf{w}$ of discrete tokens is an input to NLU, namely $\mathbf{v} = \mathbf{w}$. $\mathbf{h}^I = null$, i.e. optional additional representations are not emitted by these interfaces. The two models can be decoupled, have potentially different vocabularies, and be trained independent of each other.

2.2.2. Continuous Token Sequence Interface

Continuous token sequence interfaces produce an output $\mathbf{v}$ of length U, where each output v_i is a continuous representation corresponding to each token in ASR's one-best hypothesis. $\mathbf{h}^I = null$ i.e. optional additional representations are not emitted by these interfaces.

Tied Embedding Interface: The input token embedding matrix from ASR, Emb_{asr}, that is available from the LAS decoder or RNNT prediction network processes the discrete ASR's one-best token sequence $\mathbf{w}$ to produce token embeddings $\mathbf{e} = (e_1, ..., e_U)$. Effectively, this ties the input token embedding params across ASR and NLU. $\mathbf{v} = \mathbf{e} = Emb_{asr}(\mathbf{w})$.

Posterior Interface: Posterior probabilities corresponding to the one-best token sequence $\mathbf{w}$ of ASR are passed as inputs to NLU. Each $v_i \in \mathbb{R}^{|V|\times 1}$ is a continuous representation corresponding to the posterior probability of the token over vocabulary $|V|$, i.e. $v_i = P(w_i|\mathbf{x})$.
Hidden Interface: It produces a hidden layer representation corresponding to each token output from ASR, $\mathbf{v} = \mathbf{h}^A$ where $\mathbf{h}^A = (h_1^A, ..., h_U^A)$. Strategies to obtain the token-aligned hidden representation sequence $\mathbf{h}^A$ of length U from ASR depend on the E2E ASR architecture.
For LAS-based ASR, a hidden decoder representation is available for each step of decoding producing a token output. Similar to that of [11], the decoder hidden sequence $\mathbf{h}^d$ of length U can be directly utilized, i.e. $\mathbf{h}^A = \mathbf{h}^d$. For RNNT-based ASR, designing a hidden interface is not as straightforward since the prior work [11] on hidden interfaces for LAS cannot be extended to RNNT as easily. Note that the hidden output representations from the joint network corresponding to a decoding hypothesis in RNNT are rank-3 tensors $\mathbf{H}^J$ (i.e. $h_{t,u}^J \in \mathbb{R}^{|J|\times 1}$). In our paper, we propose a novel way of processing these joint network hidden representations and then providing e a U length input to NLU. This hidden interface maps intermediate hidden outputs $h_{t,u}^J$ corresponding to the final joint network layer to $\mathbf{h}^A = (h_1^A, ..., h_U^A)$. For each subword output w_u, we pick a joint hidden representation h_u^A at the input frame index i_u that has the maximum label transition probability from the previous subword w_{u-1} corresponding to state $(i_u, u-1)$. We now have $h_u^A = h_{t=i_u,u}^J$ where $i_u = \arg\max_t P(w_u|t, u-1)$ as shown in Fig 2. Although this hidden interface does not explicitly enforce monotonicity in time (i.e. frame indices) for the chosen hidden representations, we observe in various experiments that training results in chosen hidden representations which are monotonic in time. The intuition behind this interface is in Fig 2, illustrating how the joint hidden outputs corresponding to each label are selected.

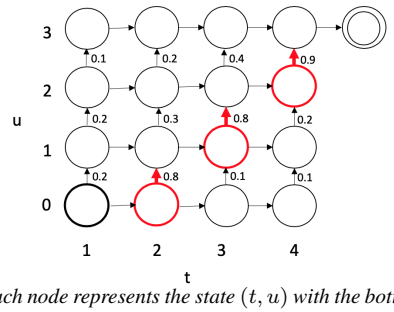

Figure 2: *Each node represents the state (t, u) with the bottom left node $(1, 0)$ being the start state. Vertical transitions correspond to transitions with probability $P(w_u|t, u-1)$. Horizontal transitions correspond to blank transitions [4] with probability $P(\phi|t, u-1)$. Each label w_u has a node with the maximum probability vertical transition (red) at row $u-1$, marked in red. The hidden states from red nodes form $\mathbf{h}^A$, which is input to neural NLU.*

2.2.3. Token Sequence + Audio Representations Interface

Audio-Attention Interface: This novel interface exposes additional audio representations from ASR besides $\mathbf{v}$. TNLU can cross-attend [30] to these, to have an access to richer information, i.e. $\mathbf{v} = \mathbf{w}; \mathbf{h}^I = \mathbf{h}^e$. Unlike other prior interfaces described in our paper, this interface adds trainable weights to NLU due to its cross-attention.

2.3. Performance Metrics

Word Error Rate (WER): ASR metric that is defined as the normalized minimum word edit distance. Sentence ER (SER) is the percentage of utterances with word errors.
Intent Classification Error Rate (ICER) and Intent Accuracy (IntAcc): An SLU metric, ICER refers to the percentage (%) of utterances whose intent predictions by the model are incorrect. IntAcc is obtained as 100%-ICER.
Semantic Error Rate (SemER): Introduced in [14], for this SLU metric, all slots (including intents) in the reference are compared to the given hypothesis and marked as (1) Substitution: if the slot name is correct but not the value (2) Insertion: if the slot is added in hypothesis or (3) Deletion: if the slot is missing in hypothesis. SemER refers to the number of slot errors normalized by the number of reference slots.
SLU-F1: Introduced in [1], this SLU metric 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.

2.4. Loss Functions for Joint Training

2.4.1. MLE Loss

ASR Loss: Given an input audio sequence, the negative log posterior of ASR's output label sequence is the ASR loss, defined as $L_{asr} = -lnP(\mathbf{w}|\mathbf{x})$. For RNNT, the loss function and its gradient are calculated using a forward-backward algorithm [4].
NLU Loss: The cross-entropy loss function is employed in both the output slot distribution of each token and the utterance intent. These are summed to obtain L_{nlu}.
Multi-Task Loss: Neural interface based SLUs can be jointly trained using both ASR and NLU parameters via a multi-task loss function as $L_{mt} = L_{asr} + L_{nlu}$.

2.4.2. Sequence Loss

This loss function directly optimizes the expected SLU performance metrics (e.g. WER or SLU-F1) over the output candidate distribution produced. For the non-differentiable SLU error metric $M(C, c^\star)$ of a SLU output candidate C and its corresponding ground truth annotation $c^\star$, its expected metric cost can be approximated by assuming that the probability mass is concentrated in the top n-hypotheses $\bar{\mathcal{C}}$, similar to previous sequence discriminative training approaches in [14, 22–24]. The gradient of the model weights θ is computed with respect to the candidate probabilities $\bar{p}(c;\theta)$ normalized over the n-best hypotheses as shown in Eqn 2.

$$\nabla_\theta L_{seq} = \nabla\mathbb{E}[M(C, c^\star)] \approx \sum_{c\in\bar{\mathcal{C}}} M(c, c^\star)\nabla_\theta \bar{p}(c;\theta) \quad (2)$$

3. Experimental Setup

3.1. Datasets

We perform our experiments on the following datasets that include parallel speech transcriptions as well as NLU annotations.
SLURP Dataset: A public dataset [1] that consists of 58-hrs of speech data with 72k utterances representing 18 domains (scenarios), 46 intents (actions), and 56 slots (entities). We do not use any additional provided synthetic data
Voice Assistant Dataset: A dataset that consists of 19K-hrs of de-identified in-house far-field speech data. These are English utterances that are directed to voice assistants, representing 26 domains, 176 intents and 141 slots. The evaluation set has 68-hrs of data. As a baseline ASR dataset for comparison, we use

Table 2: *Performance numbers on the public SLURP speech dataset. Our results are shown below prior work as well as NLU results.*

Model	Interface	WER	Intent Acc	SLU-F1
NLU models with ground-truth text input				
RoBERTa [26]	-	-	87.73	84.34
TNLU	-	-	85.80	75.25
BertNLU	-	-	88.14	85.97
SLU models with speech input				
Baseline [1]	text	16.3	78.33	70.84
Transformer ASR - RoBERTa [26]	posterior	16.7	82.93	71.20
RNNT → TNLU (independently trained)	text	16.9	78.00	67.56
RNNT → TNLU (jointly trained)	text	16.7	78.71	67.89
	hidden	16.9	78.79	68.37
	attention	16.9	79.50	68.67
RNNT → BertNLU (independently trained)	text	16.9	82.00	71.28
RNNT → BertNLU (jointly trained)	text	**15.2**	82.45	**72.35**
LAS → TNLU (jointly trained)	text	17.5	78.57	67.93
	hidden	17.4	79.50	69.35
LAS → BertNLU (jointly trained)	text	16.4	82.30	70.65

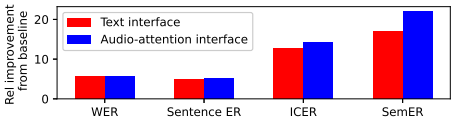

Figure 3: *Impact of joint training with text and neural interfaces vs independently trained RNNT, NLU baseline.*

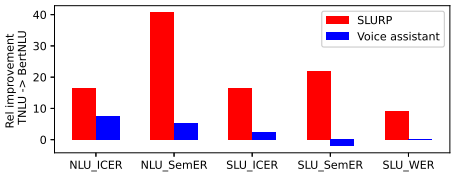

Figure 4: *Measuring the impact of pretraining with dataset size. NLU_ metrics refer to ground-truth text-input, while SLU_ metrics refer to speech input*

another 23k-hrs corpus with speech transcriptions only to pretrain our LAS and RNNT.

3.2. Model details

Features: Audio features are 64-dim log-mel filterbank energies computed over a 25ms window with 10ms shifts; stacked and downsampled to a 30ms frame rate. Ground truth text is tokenized into subword tokens using a unigram language model of vocabulary of 2500 (RNNT), 4500 (LAS), and 30000 (pretrained-BERT) [31] .

ASR/NLU Models: RNNT has 68M params with a 5x1024 encoder, 2x1024 prediction network, output projection to 512, a 1x512 feedforward joint network with tanh activation [27]. LAS has 77M params with a 5x512 BiLSTM encoder, 2x1024 decoder, and 4-head 768-unit attention. ASR decoding uses beam width of 4. TNLU has 20M params with 8-head 3x1024 self-attention units. The per-token representations are passed through a dense layer to get the per-token slot logits, and are maxpooled and passed through a 2-layer feed-forward network with 512 units for the intent and domain detection. The transformer-decoder is 8-head 2x512 units for cross-attention adding 8M params. The BertNLU models make use of the pretrained BERT model [29, 32] with 111M params with 12-head 12x768 self-attention layers, and extra layers as described in TNLU for the slot, intent, and domain detection.

Training: Independently trained ASR, NLU models use respective MLE loss as in Sec 2.4.1. Jointly trained models use sequence loss or multi-task MLE + seq-loss, as in Table 1.

4. Results and Discussion

We reports our experimental results for *SLURP* in Table 2 and those for *Voice Assistant* in Figures 3 and 4.

4.1. Joint Training with Interfaces Improves ASR and NLU

First, in Table 2, we observe that the SLU metrics like SLU-F1 are degraded for all the SLU models with speech input, when compared to the NLU models with ground-truth text input. This is not surprising since it is well known that the performance of downstream NLU depends on the performance of upstream ASR in pipelined SLU systems. As a consequence, the robustness of NLU to ASR errors is important in the design of any SLU system. A common observation through almost all joint training results in Table 2 and Figure 3 is that regardless of either text interface or other interfaces, joint training via sequence loss or MLE loss significantly improves the performance of ASR and NLU by 5-15%, compared to the baseline SLU model with independently trained ASR and NLU. More importantly, in Table 2 row "RNNT→ BertNLU (jointly trained)", we obtain the state-of-the-art results for SLURP by using pre-trained ASR/NLU and then jointly training using the text interface. Moreover, the joint training of ASR with pretrained NLU produces a significant ASR improvement for SLURP.

4.2. Neural Interfaces Improve NLU

Our proposed continuous token sequence and attention interfaces let NLU have an easier access to acoustic context and ASR confusion information. We validate this by observing a modest yet statistically significant performance improvement of SLU metrics in Table 2 for the jointly trained RNNT→ TNLU and LAS→ TNLU when these use the attention interface and the hidden interface respectively. We further validate this with the larger *Voice Assistant* dataset. In Figure 3, we observe that joint training with the attention interface yields a substantial SemER improvement of 22%, while with the text interface we obtain a 16% SemER improvement, compared to the independently trained baseline.

4.3. Impact of pretraining

The SLURP dataset is only 58 hrs and the training data does not fully capture acoustic or semantic variations. We observe an outsized impact of pretraining NLU models based on a BERT language model. In Figure 4, on the SLURP dataset, we see that switching from a TNLU model trained from scratch to the BERT based pretrained model leads to improvements of 15-40% in NLU metrics for the NLU model and 10-20% improvements in SLU metrics. However, on the much larger *Voice Assistant* dataset, the NLU models display smaller differences with and without pretraining and this leads to near-identical performance of the SLU model.

5. Conclusions and Future Work

In this work, we studied the impact of jointly training an ASR model and an NLU model that are connected with well-designed interfaces. We obtained state-of-the-art results on the public SLURP dataset by leveraging pretrained ASR, NLU models that are connected by a text interface and jointly trained via a sequence loss function. Moreover, we proposed richer neural interfaces that show best performance when pretrained models are not utilized, and studied the diminishing impact of pretraining based on training data size. As future work, we plan to extend our neural interfaces to work for pretrained NLUs.

Acknowledgments: We thank Thejaswi, Kai, Jing, Kanthashree, Samridhi, Ross, Nathan, Andreas, Jasha and Zhiqi for helpful discussions.

6. References

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

[2] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 4960–4964.

[3] Q. Zhang, H. Lu, H. Sak, A. Tripathi, E. McDermott, S. Koo, and S. Kumar, "Transformer transducer: A streamable speech recognition model with transformer encoders and rnn-t loss," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7829–7833.

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

[5] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina *et al.*, "State-of-the-art speech recognition with sequence-to-sequence models," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4774–4778.

[6] T. N. Sainath, Y. He, B. Li, A. Narayanan, R. Pang, A. Bruguier, S.-y. Chang, W. Li, R. Alvarez, Z. Chen *et al.*, "A streaming on-device end-to-end model surpassing server-side conventional model quality and latency," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6059–6063.

[7] 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.

[8] D. Guo, G. Tur, W.-t. Yih, and G. Zweig, "Joint semantic utterance classification and slot filling with recursive neural networks," in *2014 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2014, pp. 554–559.

[9] Q. Chen, Z. Zhuo, and W. Wang, "Bert for joint intent classification and slot filling," *arXiv preprint arXiv:1902.10909*, 2019.

[10] 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 *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 720–726.

[11] M. Rao, A. Raju, P. Dheram, B. Bui, and A. Rastrow, "Speech to Semantics: Improve ASR and NLU Jointly via All-Neural Interfaces," in *Proc. Interspeech 2020*, 2020, pp. 876–880. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2020-2976

[12] L. Lugosch, M. Ravanelli, P. Ignoto, V. S. Tomar, and Y. Bengio, "Speech Model Pre-Training for End-to-End Spoken Language Understanding," in *Proc. Interspeech 2019*, 2019, pp. 814–818. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-2396

[13] Y. Qian, X. Bianv, Y. Shi, N. Kanda, L. Shen, Z. Xiao, and M. Zeng, "Speech-language pre-training for end-to-end spoken language understanding," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 7458–7462.

[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 *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 7473–7477.

[15] S. Rongali, B. Liu, L. Cai, K. Arkoudas, C. Su, and W. Hamza, "Exploring transfer learning for end-to-end spoken language understanding," in *AAAI*. AAAI Press, 2021, pp. 13 754–13 761.

[16] D. Hakkani-Tür, F. Béchet, G. Riccardi, and G. Tur, "Beyond asr 1-best: Using word confusion networks in spoken language understanding," *Computer Speech & Language*, vol. 20, no. 4, pp. 495–514, 2006.

[17] C.-W. Huang and Y.-N. Chen, "Adapting pretrained transformer to lattices for spoken language understanding," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 845–852.

[18] M. Li, X. Liu, W. Ruan, L. Soldaini, W. Hamza, and C. Su, "Multi-task learning of spoken language understanding by integrating n-best hypotheses with hierarchical attention," in *Proceedings of the 28th International Conference on Computational Linguistics: Industry Track*, 2020, pp. 113–123.

[19] S. N. Ray, M. Wu, A. Raju, P. Ghahremani, R. Bilgi, M. Rao, H. Arsikere, A. Rastrow, A. Stolcke, and J. Droppo, "Listen with intent: Improving speech recognition with audio-to-intent front-end," in *Proc. Interspeech 2021*, 2021.

[20] 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 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 7493–7497.

[21] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine learning*, vol. 8, no. 3-4, pp. 229–256, 1992.

[22] K. Veselỳ, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks." in *Interspeech*, vol. 2013, 2013, pp. 2345–2349.

[23] R. Prabhavalkar, T. N. Sainath, Y. Wu, P. Nguyen, Z. Chen, C.-C. Chiu, and A. Kannan, "Minimum word error rate training for attention-based sequence-to-sequence models," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4839–4843.

[24] J. Guo, G. Tiwari, J. Droppo, M. V. Segbroeck, C.-W. Huang, A. Stolcke, and R. Maas, "Efficient Minimum Word Error Rate Training of RNN-Transducer for End-to-End Speech Recognition," in *Proc. Interspeech*, 2020, pp. 2807–2811.

[25] 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, pp. 120–130.

[26] S. Seo, D. Kwak, and B. Lee, "Integration of pre-trained networks with continuous token interface for end-to-end spoken language understanding," *arXiv preprint arXiv:2104.07253*, 2021.

[27] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 6645–6649.

[28] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 3rd International Conference on Learning Representations, ICLR 2015.

[29] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 4171–4186.

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[31] T. Kudo, "Subword regularization: Improving neural network translation models with multiple subword candidates," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 66–75.

[32] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz *et al.*, "Huggingface's transformers: State-of-the-art natural language processing," *arXiv preprint arXiv:1910.03771*, 2019.