

# MULTI-TASK RNN-T WITH SEMANTIC DECODER FOR STREAMABLE SPOKEN LANGUAGE UNDERSTANDING

Xuandi Fu<sup>†</sup>, Feng-Ju Chang, Martin Radfar, Kai Wei,  
Jing Liu, Grant P. Strimel, Kanthashree Mysore Sathyendra

Carnegie Mellon University<sup>†</sup>, Amazon Alexa

xuandifu@cmu.edu<sup>†</sup>, {fengjc, radfarmr, kaiwe, jlmk, gsstrime, ksathyen}@amazon.com

## ABSTRACT

End-to-end Spoken Language Understanding (E2E SLU) has attracted increasing interest due to its advantages of joint optimization and low latency when compared to traditionally cascaded pipelines. Existing E2E SLU models usually follow a two-stage configuration where an Automatic Speech Recognition (ASR) network first predicts a transcript which is then passed to a Natural Language Understanding (NLU) module through an interface to infer semantic labels, such as intent and slot tags. This design, however, does not consider the NLU posterior while making transcript predictions, nor correct the NLU prediction error immediately by considering the previously predicted word-pieces. In addition, the NLU model in the two-stage system is not streamable, as it must wait for the audio segments to complete processing, which ultimately impacts the latency of the SLU system. In this work, we propose a streamable multi-task semantic transducer model to address these considerations. Our proposed architecture predicts ASR and NLU labels auto-regressively and uses a *semantic decoder* to ingest both previously predicted word-pieces and slot tags while aggregating them through a fusion network. Using an industry scale SLU and a public FSC dataset, we show the proposed model outperforms the two-stage E2E SLU model for both ASR and NLU metrics.

**Index Terms**— Spoken Language Understanding, End-to-End, Multi-task Learning, RNN-Transducer, Semantic Beam Search

## 1. INTRODUCTION

With the widespread application of intelligent voice assistants, e.g. Alexa, Siri, and Google Home, SLU systems have generated increased interest in the recent years. An SLU system predicts semantic information implied by an audio signal. This semantic content is commonly represented as intent, slot tags, named entities and/or part-of-speech taggings. Today’s SLU technology typically accomplishes this task in two separate stages, which we refer to as ASR-NLU approaches for SLU [1]: an ASR system first transcribes the audio signals [2, 3], and then the transcripts are passed to an NLU system to extract corresponding intent and slot tags [4, 5, 6]; an example is presented in Table 1. Given the extracted semantic labels, downstream applications of the voice assistant can produce an appropriate response to the user.

Recently, complete E2E-SLU based approaches have attracted attention due to their efficiency and reduced model complexity compared with an ASR-NLU pipeline, making them suitable candidates for deployment on low-resource devices [7, 8, 9, 10]. Furthermore, performance improvements in both tasks, driven by joint training, has been observed in several studies [7, 11].

**Table 1.** An example of a transcription, slot tags, and intent.

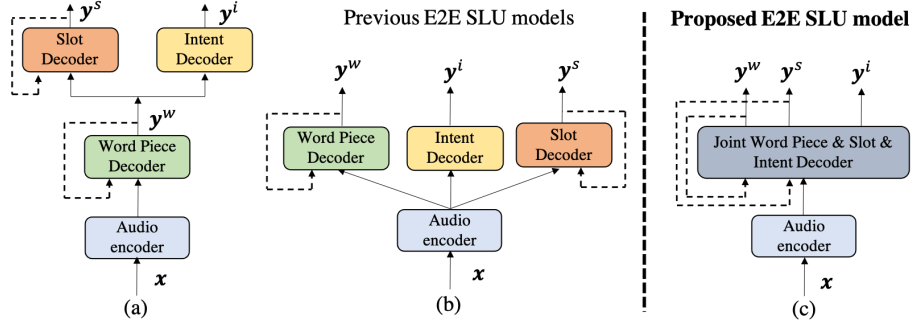
|                      |   |
|----------------------|---|
| <b>transcription</b> | turn on the kitchen light   |
| <b>slot tags</b>     | [DeviceLocation]: kitchen<br>[ApplianceType]: light, [Other]: turn, on, the |
| <b>intent</b>        | TurnOnApplianceIntent   |

Most of existing E2E-SLU models still adopt a two-stage setup as shown in Fig. 1(a), where the NLU subsystem waits for the transcripts of the whole utterance produced by the ASR subsystem to generate semantic labels [12, 7, 11, 13]. Meanwhile, the NLU subsystem is typically non-streamable. One-stage approaches to E2E-SLU have been proposed as well [7, 10]; however, again the NLU subsystem remains non-streamable. For example, in [10], slot tag prediction only occurs after the intent is extracted at the end of an utterance. Moreover, in all the above approaches, the ASR and NLU label generations do not interact with one another during the forward pass of inference (Fig. 1(b)). As a result, this design can lead to three main limitations. First, the NLU posterior or hypothesis does not provide any feedback upon word-piece generation, while its feedback could be helpful to narrow down potential word-piece candidates generated in the next time step. Secondly, the decoding of the NLU label predictions is not streamable [7], given that the model is an encoder-decoder framework augmented by attention. Finally, the inference speed of an SLU system may be affected by the nature of the cascaded setup and non-streamable NLU subsystem, all the while low latency is crucial for a responsive virtual assistant.

To address these limitations, we propose a streamable E2E-SLU model based on RNN-T [14, 15] with a novel semantic beam search decoder which predicts word-pieces and NLU labels jointly, as illustrated in Fig. 1(c). Specifically, we introduce a semantic decoder to aggregate not only the word-pieces but also slot candidates during the beam search, which we call *semantic beam search*. Furthermore, we propose different multi-task loss functions to learn the alignment between word-pieces and slot tags along with the intent prediction.

## 2. RELATED WORK

A multi-task learning framework for E2E SLU was first introduced in [7]; the authors investigated several encoder-decoder structures for joint training of ASR and NLU tasks, in which the multi-task structure achieves the best performance. This work was followed by proposed pre-training based approaches to improve the performance [11, 17, 18]; all of these models are designed specifically for intent classification. Another category of work attempts a combined parameter transfer from well trained end-to-end ASR systems



**Fig. 1.** A high-level diagram of comparing the proposed E2E SLU model to the previous two-stage [16, 4, 5, 6, 1] (a) and one-stage [7, 10] (b) E2E SLU models. Dotted lines represent the conditioning of output label on its history, as a part of decoders.

and end-to-end NLU models such as pretrained BERT [19] through teacher-student learning [13, 20, 21]. Note that both of these categories rely on at least a two-stage training process and all operate with non-streamable inference.

Leveraging RNN-T, in [16], a two-stage E2E SLU structure was proposed where the RNN-T based ASR subsystem interacts with an NLU subsystem through an interface, which is not streamable. In the most recent work, [22] proposed a CTC-based streamable E2E SLU framework which employs a unidirectional RNN to make multiple intent predictions. The NLU output is generated either directly from the audio signal or based on an intermediate ASR output. Namely, the advantage of semantic posterior that we employ, was not considered in making word-piece predictions. To the best of our knowledge, there is no existing E2E SLU model, either streamable or non-streamable, which takes both word-piece and slot tag in beam search decoding on joint multi-task sequence prediction for word-piece, slot tag, and intent label.

### 3. METHODOLOGY

The inputs of the multi-task RNN-T are  $D$ -dimensional audio features of length  $T$ ,  $\mathcal{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T)$ ,  $\mathbf{x}_k \in \mathbb{R}^D$ . The outputs are transcript tokens of length  $U$ ,  $\mathbf{y}^w = (y_1^w, y_2^w, \dots, y_U^w)$ ,  $y_u^w \in \mathcal{W}$ , its corresponding slot tags (also of length  $U$ ),  $\mathbf{y}^s = (y_1^s, y_2^s, \dots, y_U^s)$ ,  $y_u^s \in \mathcal{S}$ , and the intent  $y^i \in \mathcal{I}$ ; here  $\mathcal{W}$ ,  $\mathcal{S}$ , and  $\mathcal{I}$  are the predefined set of token labels (or token vocabulary), slot tags, and intents. Both transcript tokens and slot tags are encoded as one-hot vectors.

The model defines a conditional distribution of  $p(\mathcal{W}, \mathcal{S}, \mathcal{I}|\mathcal{X})$ , and we factorize it as follows (Fig. 2),

$$p(\hat{\mathbf{y}}^w, \hat{\mathbf{y}}^s, y^i | \mathcal{X}) = \prod_{k=1}^{T+U} p(\hat{y}_k^w | \mathcal{X}, t_k, y_0^w, \dots, y_{u_{k-1}}^w, y_0^s, \dots, y_{u_{k-1}}^s) p(y^i | \hat{y}_k^w) \prod_{j=1}^{T+U} p(\hat{y}_j^s | \mathcal{X}, t_j, y_0^s, \dots, y_{u_{j-1}}^s, y_0^w, \dots, y_{u_{j-1}}^w) \quad (1)$$

where  $\hat{\mathbf{y}}^w = (\hat{y}_1^w, \dots, \hat{y}_{T+U}^w) \subset \{\mathcal{W} \cup \langle b^w \rangle\}^{T+U}$ ,  $\hat{\mathbf{y}}^s = (\hat{y}_1^s, \dots, \hat{y}_{T+U}^s) \subset \{\mathcal{S} \cup \langle b^s \rangle\}^{T+U}$  correspond to any possible alignment path with  $T$  blank symbols and  $U$  token/slot labels such that after removing all blank symbols,  $b^w$  and  $b^s$ , in  $\hat{\mathbf{y}}^w$  and  $\hat{\mathbf{y}}^s$  correspondingly, it yields  $\mathbf{y}^w$  and  $\mathbf{y}^s$ .  $y_0^w$  and  $y_0^s$  are the start of sentence and slot symbol respectively.

#### 3.1. Multi-task Semantic RNN-T

The multi-task Semantic RNN-T architecture consists of three components, an audio encoder, a semantic decoder and a multiple-output joint network as shown in Fig. 2. The audio encoder is a unidirectional RNN [23] that takes the audio features  $\mathcal{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T)$  as inputs and generates the hidden representations,  $\mathcal{H} = (\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_T)$  auto-regressively. The semantic decoder takes in the word-pieces along with slot tags and outputs hidden label embeddings,  $\mathcal{G} = (\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_U)$ . The encoder and semantic decoder output,  $\mathbf{h}_t$  and  $\mathbf{g}_u$ , respectively, are then fed into a joint network to predict next word-piece,  $y_{u+1}^w$ , and slot tag,  $y_{u+1}^s$ .

The semantic decoder has two separate prediction networks for encoding word-pieces and slot tags, correspondingly, and a fusion layer is employed to aggregate the prediction network outputs. Each of the prediction networks is a recurrent neural network consisting of an embedding layer, an output layer, and a recurrent hidden layer. The outputs of the two prediction networks,  $\mathbf{g}_u^w$  and  $\mathbf{g}_u^s$ , are then fused together producing the semantic decoder output,  $\mathbf{g}_u$ . While we investigated both the addition,  $\mathbf{g}_u = \mathbf{g}_u^w + \mathbf{g}_u^s$ , and the concatenation with a projection,  $\mathbf{g}_u = W([\mathbf{g}_u^w; \mathbf{g}_u^s])$ , as fusion methods, we did not observe a significant performance difference between them, so we report all results with the addition fusion method.

Given the audio feature vector  $\mathbf{h}_t$  and semantic decoder output  $\mathbf{g}_u$ , the multi-output joint network yields distributions for the word-piece and slot tag at the next time step  $u + 1$ . The joint network is composed of a feed-forward neural network and two separate classification layers to produce joint logits, also called lattice, for transcript tokens and slot tags,  $Z^w \in \mathbb{R}^{T \times U \times V^w}$  and  $Z^s \in \mathbb{R}^{T \times U \times V^s}$ , where  $V^w$  and  $V^s$  stand for the word piece size and slot value size, respectively. Each element of  $Z^w$  and  $Z^s$  represents the probability of the next word piece  $p(y_{u+1}^w | t, u)$ , and the next slot tag  $p(y_{u+1}^s | t, y_u^s)$ , correspondingly. The intent classification layer is appended upon the prediction network for the word-piece. The reason for separating intent prediction from the slot tag hypotheses is to reduce the effect of [Other] slot hypotheses (see Table 1) in prior time steps on the intent prediction for the final state.

#### 3.2. Semantic Beam Search

To jointly decode the word piece and slot tag sequences at inference time, we propose a semantic beam search algorithm based on the search algorithm in [14] (only applied to the word-pieces) to find the top-n best output pairs of word pieces and slot tags. The motivation is to provide the decoder with the most possible candidate pairs of the fixed beam size in every time step and select the best aligned

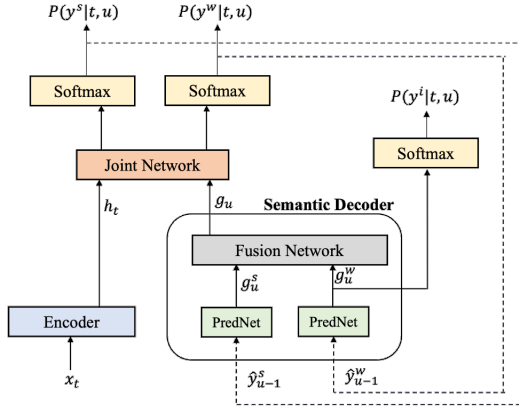


Fig. 2. The proposed Multi-task Semantic RNN-T SLU model.

sequences through the decoding path. The semantic beam search parameters of each decoding are local word beam size ( $B_{wp}$ ), local slot beam size ( $B_{slot}$ ), local candidate pair beam size ( $B_{local}$ ), and global candidate pair beam size ( $B_{beam}$ ).  $B_{wp}$  and  $B_{slot}$  define the number of top possible word piece and slot candidates selected by the top log probability respectively. Given the  $B_{wp} \times B_{slot}$  candidate pairs (combining the top possible word pieces and slot tags),  $B_{local}$  of candidate pairs with the highest addition of log probabilities are then selected. Finally, among  $B_{local} \times B_{beam}$  candidate pairs, we preserve  $B_{beam}$  ones for the next decoding step.

### 3.3. Loss Functions

#### 3.3.1. Word-Piece Loss

The word-piece prediction is optimized with the RNN-T loss [14], denoted as  $L_{rnnT}(wp)$ , which computes the alignment probability summation,  $p(\hat{y}^w|\mathcal{X})$  with a forward-backward algorithm.

#### 3.3.2. Intent Classification Loss

The intent classification is optimized by minimizing the cross-entropy loss between the intent logits and the ground truth intent label, summed over a batch of utterances.

$$L_{ce}(intent) = - \sum y^i \times \log(p(\hat{y}^i|t, u, \hat{y}^w)) \quad (2)$$

#### 3.3.3. Slot Tagging Loss

Given the generated transcript tokens and slot tags from prediction networks (Fig. 2), this loss is designed to learn the alignment between the two sequences. In particular, we investigate two losses:

- **Cross Entropy Loss:** The cross-entropy loss is computed at each state of the slot lattice  $Z^s$  and averaged over  $T$  time steps for each decoder state.

$$L_{ce}(slot) = - \sum_{u=1}^U \frac{1}{T} \sum_{t=0}^T y^s \log(p(\hat{y}^s|t, u, \hat{y}^w)) \quad (3)$$

- **Aligned RNN-T Loss:** This loss consists of two terms as follows,

$$L_{rnnT,align}(slot) = L_{rnnT}(slot) + L_{align}(slot) \quad (4)$$

Table 2. Relative Improvements of ASR and NLU metrics (%) for Multi-task Semantic RNN-T over the two-stage SLU [16].

| Model                     | WERR       | SemERR     | IRERR       | ICERR      |
|---------------------------|------------|------------|-------------|------------|
| Two-stage SLU [16]        | 0          | 0          | 0           | 0          |
| One-stage version of [16] | 0.6        | 1.2        | 0.1         | -5.8       |
| Multi-task Semantic RNN-T | <b>1.4</b> | <b>9.5</b> | <b>14.4</b> | <b>5.1</b> |

Similar to the word-piece loss, the first term,  $L_{rnnT}(slot)$  is used to learn the alignment between the audio inputs and the slot tags with the standard RNN-T loss [14]. The second term  $L_{align}(slot)$ , is responsible for learning the alignment between the word pieces and their corresponding slot tags at each state,  $L_{align}(slot) = -\log(p(\mathbf{y}^w, \mathbf{y}^s|\mathcal{X}))$ . Based on a conditional independence assumption, we use  $p(y_{u+1}^w, y_{u+1}^s|t, y_u^s, y_u^w) = p(y_{u+1}^w|t, y_u^w) \cdot p(y_{u+1}^s|t, y_u^s)$  at each state  $(t, u+1)$  and are able to simply reapply the same transducer forward-backward algorithm on this combined lattice to efficiently compute  $L_{align}(slot)$ .

## 4. EXPERIMENTAL RESULTS

### 4.1. Dataset

To evaluate the multi-task semantic RNN-T model, we use 1,300 hours of speech utterances from our in-house de-identified far-field SLU dataset, containing not only transcriptions but also slot tags and intents. This is broken into training and test sets of 910 hours and 195 hours, respectively. The device-directed far-field, speech data is captured using a smart speaker across multiple English locales (e.g. en-US, en-IN, etc.). The input audio features fed into the network consist of 64-dimensional LFBE features, which are extracted every 10 ms with a window size of 25 ms from audio samples. The features of each frame are then stacked with the left two frames, followed by a downsampling of factor 3 to achieve a low frame rate, with 192 feature dimensions. The subword tokenizer [24, 25] is used to create tokens from the transcriptions; we use 4000 word-pieces in total. There are 63 intent classes and 183 slot tags annotated in this dataset. We also conducted experiments on the public SLU corpus, Fluent Speech Commands (FSC) dataset [11], which contains 15 hrs (23k utterances), 11 intents and 3 slots, including the “Other” class.

### 4.2. Baselines and Model Configurations

We compare the proposed method with the state-of-the-art RNN-T based two-stage SLU model [16]. We also compare to another baseline by extending [16] to the one-stage version (as in Fig. 1) introduced in [7]. The proposed model has the following configurations. The audio encoder is a 5-layer LSTM with 736 neurons and output size of 512-dimension. The word piece prediction network is of a 512-dim embedding layer followed by a 2-layer LSTM with 736 neurons and output size of 512-dim. The slot tag prediction network is of a 128-dim embedding layer and a 2-layer LSTM with 256 neurons and output size of 512-dim. The intent decoder consists of two 128-dim dense layers with *ReLU* as an activation function. The joint network is a fully-connected feed-forward component with one hidden layer followed by a *tanh* activation function. Overall, the size of both the proposed model and the baselines sum to approximately 40 million parameters. For the FSC data set, given the small amount of transcribed data to train the ASR module of million parameters well, we followed [16] by first pre-training the audio en-

**Table 3.** Comparisons of different proposed multi-task losses

| Model                     | Loss Type  | WERR        | SemERR      | IRERR        | ICERR       |
|---------------------------|--|-------------|-------------|--------------|-------------|
| Two-stage SLU [16]        | -  | 0           | 0           | 0            | 0           |
| Multi-task Semantic RNN-T | $L_{rnnl}(wp) + L_{ce}(slot) + L_{ce}(intent)$         | <b>1.41</b> | <b>9.49</b> | <b>14.38</b> | <b>5.13</b> |
|                           | $L_{rnnl}(wp) + L_{rnnl,align}(slot) + L_{ce}(intent)$ | -0.99       | <b>7.43</b> | <b>12.04</b> | -1.26       |

**Table 4.** Comparisons of different slot beam sizes,  $B_{slot}$ , in semantic beam search configuration,  $(B_{wp}, B_{slot}, B_{local}, B_{beam})$ , for Multi-task Sem-RNN-T

| Semantic Beam Search    | WERR       | SemERR      | IRERR      | ICERR      |
|-------------------------|------------|-------------|------------|------------|
| (1,1,1,1)-Greedy Search | 0          | 0           | 0          | 0          |
| (10,1,10,8)             | 8.5        | 0.6         | 0.6        | 1.1        |
| (10,2,10,8)             | <b>8.6</b> | <b>11.9</b> | <b>9.5</b> | <b>2.9</b> |
| (10,4,10,8)             | <b>8.6</b> | <b>11.9</b> | <b>9.5</b> | 2.8        |

**Table 5.** ASR and NLU performances (%) on the public Fluent Speech Commands dataset

| Model                     | Streaming | SemDec.  | WER         | IRER        |
|---------------------------|-----------|----------|-------------|-------------|
| Two-stage SLU [16]        | N         | N        | 0.61        | 0.85        |
| One-stage version of [16] | Y         | N        | 0.54        | 0.91        |
| MT Semantic RNN-T (Ours)  | <b>Y</b>  | <b>Y</b> | <b>0.55</b> | <b>0.84</b> |

coder, the word piece prediction network, and the joint network on 910 hrs Alexa data, before finetuning on the FSC dataset.

### 4.3. Metrics

We use four metrics to evaluate the performance of an E2E SLU system. (i) **Word Error Rate (WER)**: WER is a word-level metric used for evaluating the word-piece recognition performance. It calculates Levenshtein distance or edit distance that is the shortest distance required for transforming word-piece hypothesis to the ground truth by using insertion, deletion and substitution. (ii) **Semantic Error Rate (SemER)**: The SemER metric jointly evaluates the performance of intent classification and slot filling or say NLU performance. By comparing a word sequence reference and their accompanying slot tags, performance is calculated as:

$$SemER = \frac{\#Deletion + \#Insertion + \#Substitution}{\#Correct + \#Deletion + \#Substitution}, \quad (5)$$

where *Correct* is when slot tag and slot value (words) are correctly identified, *Deletion* is when a slot tag present in the reference is not the hypothesis, *Insertion* is an extraneous slot tag included by hypothesis, and *Substitution* is when a slot tag from hypothesis is included but with the incorrect slot value. Intent classification errors are counted as substitution errors. (iii) **Interpretation Error Rate (IRER)**: The IRER metric is an utterance-level metric for evaluating the joint intent classification and slot filling performance without partial credit. Namely, it is the fraction of utterances where either the intent or any of the slots are predicted incorrectly. (iv) **Intent Classification Error Rate (ICER)**: The ICER metric measures the error rate of intent classification, which is an utterance-level evaluation metric. The results of all experiments are presented as the relative error rate reductions (WERR/SemERR/IRERR/ICERR). For example, given model A’s WER ( $WER_A$ ) and a baseline B’s WER ( $WER_B$ ), the WERR of A over B is computed as  $WERR = (WER_B - WER_A) / WER_B$ .

### 4.4. Results

The results of multi-task semantic RNN-T over the baselines are shown in Table 2. Jointly training ASR and NLU tasks with a semantic decoder shows consistent improvements across tasks and metrics by providing more contextual information. Of note is that the NLU metrics such as SemER and IRER are significantly improved while the improvement of WER is comparably small. We attribute this to the annotation bias of the slot tag distribution of the dataset: Around 60% of the utterances have their slot tags mapped to the [Other] label. Therefore, the semantic information provided by slot tags for the word-piece generation may be limited.

In Table 3, we compare different loss combinations as introduced in Sec 3.3. Using cross-entropy loss for slot tagging has demonstrated the best overall performances, and greatly improves the two-stage SLU model in terms of both NLU and ASR metrics. Imposing aligned RNN-T loss also significantly improves the NLU metrics such as SemER and IRER, but slightly degrades the WER and ICER. We believe this is because the loss is more sensitive to the misalignments between the word-piece and slot tags produced by the separate prediction networks.

Finally, we validate the effectiveness of the semantic beam search algorithm. We fixed the best-performing parameters of  $B_{wp}$ ,  $B_{local}$ ,  $B_{beam}$ , and varied  $B_{slot}$  to 1, 2, 4 and showed the results in Table 4. As it can be seen, changing the parameters from a greedy search to the beam search, from (1,1,1,1) to (10,1,10,8), improves all metrics, while mainly improving WER. When further increasing the slot beam size from 1 to 2 or 4, the improvements over NLU metrics become significant, 11.9% and 9.5% in terms of SemERR and IRERR. Again, the improvement of WER from increasing slot beam search size is limited and may be attributed to the inherent bias of the slot tag annotations.

Table 5 presents the results on FSC dataset [11], where the streaming capability and the use of semantic information during decoding (Yes/No: Y/N) are also shown in the table. In our model (MT Semantic RNN-T), we used the additive fusion to obtain the semantic decoder output, with the semantic beam search size set by  $B_{wp}=10$ ,  $B_{slot}=2$ ,  $B_{local}=10$ , and  $B_{beam}=16$ . Note that due to the lower complexity and limited number of intents and slots in FSC, all the models in our experiments lead to  $< 1\%$  WER and IRER values. The proposed model improved the WER of 2-stage SLU by 9.8% while improved IRER of 1-stage SLU by 7.7% relatively.

## 5. CONCLUSION

In this paper, we proposed a multi-task semantic RNN-T architecture for streamable end-to-end spoken language understanding. The proposed semantic decoder and semantic beam search empower the model to consider more contextual information, both from the predicted word piece and slot tags in the history, and produce higher quality word-piece and slot tag hypothesis in the next time step. Moreover, the model with the proposed losses, both cross-entropy loss and the aligned RNN-T loss for slot tagging, outperformed the two-stage and one-stage E2E SLU models.

## 6. REFERENCES

- [1] Martha Larson and Gareth JF Jones, “Spoken content retrieval: A survey of techniques and technologies,” *Foundations and Trends in Information Retrieval*, vol. 5, no. 4-5, pp. 235–422, 2012.
- [2] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Sathesh, Shubho Sengupta, Adam Coates, et al., “Deep speech: Scaling up end-to-end speech recognition,” *arXiv preprint arXiv:1412.5567*, 2014.
- [3] William Chan, Navdeep Jaitly, Quoc Le, and Oriol 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)*. IEEE, 2016, pp. 4960–4964.
- [4] Bing Liu and Ian R. Lane, “Attention-based recurrent neural network models for joint intent detection and slot filling,” in *INTERSPEECH*, 2016.
- [5] Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen, “Slot-gated modeling for joint slot filling and intent prediction,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 2018, pp. 753–757.
- [6] Qian Chen, Zhu Zhuo, and Wen Wang, “Bert for joint intent classification and slot filling,” *arXiv preprint arXiv:1902.10909*, 2019.
- [7] Parisa Haghani, Arun Narayanan, Michiel Bacchiani, et al., “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.
- [8] Cheng-I Lai, Yung-Sung Chuang, Hung-Yi Lee, Shang-Wen Li, and James Glass, “Semi-supervised spoken language understanding via self-supervised speech and language model pretraining,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 7468–7472.
- [9] Yung-Sung Chuang, Chi-Liang Liu, Hung-Yi Lee, and Linshan Lee, “Speechbert: An audio-and-text jointly learned language model for end-to-end spoken question answering,” *arXiv preprint arXiv:1910.11559*, 2019.
- [10] Martin Radfar, Athanasios Mouchtaris, Siegfried Kunzmann, and Ariya Rastrow, “Fans: Fusing asr and nlu for on-device slu,” *Proc. Interspeech 2021*, pp. 1224–1228, 2021.
- [11] Loren Lugosch, Mirco Ravanelli, Patrick Ignoto, Vikrant Singh Tomar, and Yoshua Bengio, “Speech model pre-training for end-to-end spoken language understanding,” *Proc. Interspeech 2019*, pp. 814–818, 2019.
- [12] Dmitriy Serdyuk, Yongqiang Wang, Christian Fuegen, Anuj Kumar, Baiyang Liu, and Yoshua Bengio, “Towards end-to-end spoken language understanding,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5754–5758.
- [13] Pavel Denisov and Ngoc Thang Vu, “Pretrained semantic speech embeddings for end-to-end spoken language understanding via cross-modal teacher-student learning,” *Proc. Interspeech 2020*, pp. 881–885, 2020.
- [14] Alex Graves, “Sequence transduction with recurrent neural networks,” *arXiv e-prints*, pp. arXiv–1211, 2012.
- [15] Takaki Makino, Hank Liao, Yannis Assael, Brendan Shillingford, Basilio Garcia, Otavio Braga, and Olivier Siohan, “Recurrent neural network transducer for audio-visual speech recognition,” in *2019 IEEE automatic speech recognition and understanding workshop (ASRU)*. IEEE, 2019, pp. 905–912.
- [16] Anirudh Raju, Gautam Tiwari, Milind Rao, Pranav Dheram, Bryan Anderson, Zhe Zhang, Bach Bui, and Ariya Rastrow, “End-to-end spoken language understanding using rnn-transducer asr,” *arXiv preprint arXiv:2106.15919*, 2021.
- [17] Yuan-Ping Chen, Ryan Price, and Srinivas Bangalore, “Spoken language understanding without speech recognition,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 6189–6193.
- [18] Sahar Ghannay, Antoine Caubrière, Yannick Estève, Nathalie Camelin, Edwin Simonnet, Antoine Laurent, and Emmanuel Morin, “End-to-end named entity and semantic concept extraction from speech,” in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 692–699.
- [19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.
- [20] Minjeong Kim, Gyuwan Kim, Sang-Woo Lee, and Jung-Woo Ha, “St-bert: Cross-modal language model pre-training for end-to-end spoken language understanding,” in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 7478–7482.
- [21] Yao Qian, Ximo Bianv, Yu Shi, Naoyuki Kanda, Leo Shen, Zhen Xiao, and Michael 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.
- [22] Nihal Potdar, Anderson R. Avila, Chao Xing, Dong Wang, Yiran Cao, and Xiao Chen, “A streaming end-to-end framework for spoken language understanding,” in *IJCAI*, 2021.
- [23] Sepp Hochreiter and Jürgen Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [24] Rico Sennrich, Barry Haddow, and Alexandra Birch, “Neural machine translation of rare words with subword units,” in *ACL*, 2016.
- [25] Taku Kudo, “Subword regularization: Improving neural network translation models with multiple subword candidates,” *arXiv preprint arXiv:1804.10959*, 2018.