

# DO AS I MEAN, NOT AS I SAY: SEQUENCE LOSS TRAINING FOR SPOKEN LANGUAGE UNDERSTANDING

Milind Rao, Pranav Dheram, Gautam Tiwari, Anirudh Raju,  
Jasha Droppo, Ariya Rastrow, Andreas Stolcke

Amazon Alexa, USA,

{milinrao, pddheram, tgautam, ranirudh, drojasha, arastrow, stolcke}@amazon.com

## ABSTRACT

Spoken language understanding (SLU) systems extract transcriptions, as well as semantics of intent or named entities from speech, and are essential components of voice activated systems. SLU models, which either directly extract semantics from audio or are composed of pipelined automatic speech recognition (ASR) and natural language understanding (NLU) models, are typically trained via differentiable cross-entropy losses, even when the relevant performance metrics of interest are word or semantic error rates. In this work, we propose non-differentiable sequence losses based on SLU metrics as a proxy for semantic error and use the REINFORCE trick to train ASR and SLU models with this loss. We show that custom sequence loss training is the state-of-the-art on open SLU datasets and leads to 6% relative improvement in both ASR and NLU performance metrics on large proprietary datasets. We also demonstrate how the semantic sequence loss training paradigm can be used to update ASR and SLU models without transcripts, using semantic feedback alone.

**Index Terms**— speech recognition, spoken language understanding, REINFORCE, multitask training, neural interfaces

## 1. INTRODUCTION

Spoken language understanding systems that aim to understand user commands are an integral part of voice interfaces or spoken dialogue systems. Our focus is on developing compact models that can be deployed on edge devices allowing low-latency processing without transmitting audio and/or transcripts to cloud servers and enabling offline use in remote, medical, vehicular, or emergency environments. Table 1 shows an example of the transcript and semantics of an utterance. A conventional deployment for SLU comprises two distinct pipelined stages: (1) ASR to transcribe utterances (2) an NLU system that consumes the transcription and produces utterance intent and named entities or slots.

### 1.1. Prior Work

A pipelined or compositional deployment would make use of end-to-end (E2E) ASR architectures such as RNN-T [1], CTC [2], Transformer-transducers [3], LAS [4], or conventional RNN-HMM hybrid ASR systems [5]. Extracting intent and slots from transcripts is a long running problem in NLU [6, 7, 8] that uses LSTMs or Transformers [9, 10]. The interface between ASR and NLU systems has traditionally been the single best hypothesis generated by ASR, although richer interfaces such as lattices and word confusion networks have also been proposed [11, 12, 13, 14].

With the compositional approach listed above, ASR errors cascade down to the NLU system, ASR is not trained aware of downstream NLU use, and NLU is not trained to compensate for ASR ambiguity or error. [15] first introduced multi-stage, multi-task and joint models for E2E SLU. Most prior work in this space [16, 17,

**Table 1:** An example of intent, slots for an utterance.

Transcript	set an alarm for six a.m
Intent	SetNotificationIntent
Slots	NotificationType - alarm, Time - six a.m.

18, 19, 20] directly computes a serialization of the semantics without intermediate text output. Another common approach uses transfer learning of pretrained ASR models to SLU tasks by replacing the final layer. In contrast, [21] used pretrained ASR models and NLU architectures and replaced the one-best ASR hypothesis interface with a neural network interface allowing joint training of ASR and NLU.

ASR systems are typically first trained with differentiable losses such as cross-entropy (CE), CTC or RNN-T. NLU systems are trained using CE losses for classification problems like intent, domain, or named entity tags. E2E SLU systems make use of cross-entropy on either transcripts, intents, slots, or some serialization of semantics. The CE metric is simply a proxy for and does not directly minimize SLU metrics of interest. REINFORCE [22] can be used to train with arbitrary non-differentiable loss functions. This was extended to mWER training for ASR [23], LAS [24], and RNN-T [25]. REINFORCE corresponds to the policy gradient approach among other reinforcement learning methods for seq2seq networks [26].

### 1.2. Contributions

We consider the class of SLU models composed of multistage ASR and NLU subsystems, connected via text, subword tokens, or neural interfaces that can be *jointly* trained. In these systems, ASR is trained with backpropagation of semantic feedback from NLU, and NLU is trained to be aware of ASR ambiguity and errors preventing a downward cascade of ASR errors as seen in *compositional* systems. We consider ASR systems based on LAS[4] and LSTM- or Transformer-encoder-based NLU systems.

We first develop custom sequence loss training approaches to make use of non-differentiable arbitrary risk values or losses on the entire sequence of outputs. Similar to minimum-word-error-rate (mWER) training, we develop minimum-semantic-error-rate (mSemER) training that directly minimizes intent and slot errors. We introduce alternatives which additionally factor in interpretation (concept) and word errors. Using datasets with complete ASR transcriptions and NLU annotations, we first show significant gains in both ASR and NLU metrics using sequence loss training across datasets ranging from 15 to 15,000 hours for limited to general use cases. We beat all known external benchmarks in the open Fluent speech dataset [20].

As a further application of sequence loss training, we show how a dataset with audio and semantic annotations without human transcriptions can still be used to drive ASR and SLU model improvements.

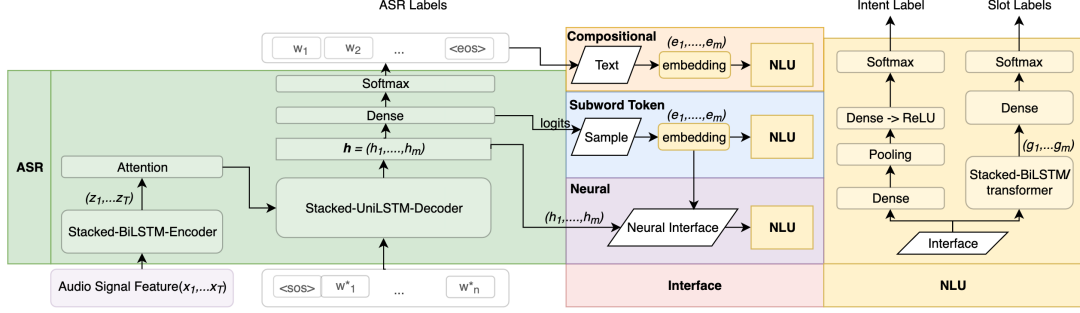


Fig. 1: E2E SLU architectures including ASR subsystem, neural NLU subsystem and 3 interfaces - token, text and neural

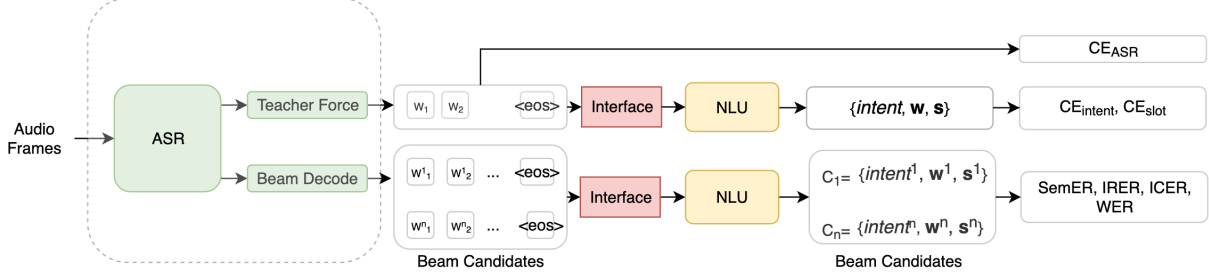


Fig. 2: Training SLU models with non-differentiable sequence losses. Dotted box encompasses ASR model including decoder

## 2. TECHNICAL APPROACH

### 2.1. ASR-Interface-NLU Models

We consider SLU models that comprise an ASR subsystem and an NLU subsystem connected by an interface that passes the 1-best or sampled ASR hypotheses, or via a neural network hidden layer.

**The ASR subsystem** is an attention-based Listen Attend and Spell (LAS) model as shown in the green box of Fig. 1. The LAS used here primarily comprises two components - a stacked RNN encoder that encodes audio frames  $\mathbf{x}$  to generate representations, and an auto-regressive RNN decoder that sequentially generates logits or subword probability distribution  $p_{w,i}(w_i) = \mathbf{P}(w_i | \{w\}_1^{i-1}, \mathbf{x})$  at each decoding step by using multiple attention heads to attend to the audio encoding.

In this work, we focus on the LAS ASR subsystem, but the results can be extended to other architectures, such as streaming-compatible RNN-T systems or Transformer-based ASR architectures.

**The Neural NLU subsystem** as shown in the yellow box of Fig. 1 accepts a sequence of embeddings or features of tokens decoded by ASR and passes it through multiple BLSTM layers. The outputs of the final layer are used to generate probability distribution  $p_{s,i}(s_i) = \mathbf{P}(s_i | \mathbf{w}, \mathbf{x})$  of the slots  $\mathbf{s}$  of each subword-token. The slot of a word is the slot of the last token of the word. The outputs or features of the final layer are also max-pooled and passed through feed-forward networks to perform the sequence classification task of obtaining logits over the utterance intent  $p_{\text{intent}}(\text{intent}) = \mathbf{P}(\text{intent} | \mathbf{w}, \mathbf{x})$ .

We also present results using a Transformer-encoder NLU architecture. The Transformer-encoder replaces the BLSTM and applies multiple layers of self-attention to embeddings/features of the transcript tokens to produce latent representations that are used to obtain slot and intent logits.

**ASR-NLU Interfaces.** We design the SLU system which comprises multistage ASR and NLU systems with a choice of interfaces:

- **Text** from the ASR hypothesis is the interface between ASR and NLU, and embeddings of these tokenized text are the inputs to the NLU system. This allows use of pre-trained ASR and NLU models, using transcribed audio datasets for the former and text-only NLU

datasets for the latter. We also term this the *compositional* baseline model that chains pretrained ASR and NLU.

- **Subword tokens** sampled from the posteriors produced by the ASR decoder form the interface with NLU. ASR and NLU can be jointly trained with NLU trained aware of ASR errors. The Gumbel-softmax sampling approach [27] allows backpropagation of semantic feedback to ASR through the categorical subword token interface.
- **Neural network interface** computes the feature for the token at decoder step  $i$  using the token embedding concatenated with the hidden output layer from the LAS decoder LSTM. This interface allows NLU to be trained aware of ASR error, local ASR decoding ambiguity as well as the audio context and allows for ASR to be trained with semantic backpropagation. A pretrained ASR model can be used using transcribed audio. These models are also termed *joint* models in this work.

### 2.2. Loss Functions and Performance Metrics

Traditionally, differentiable cross-entropy loss functions are used in ASR or SLU model training. The ASR system is *teacher-forced* with the ground truth transcript subword sequence  $\mathbf{w}$  and the cross-entropy loss  $CE_{\text{asr}} = -\sum_i \log p_{w,i}(w_i)$  is calculated using the one-step ahead decoded subword probability sequence. NLU consumes the features from ASR and is trained with an intent loss  $CE_{\text{intent}} = -\log p_{\text{intent}}(\text{intent})$  and a slot-loss  $CE_{\text{slot}} = -\sum_i \log p_{s,i}(s_i)$  using ground-truth intent and slot sequence  $\mathbf{s}$ .

During joint ASR-NLU model multi-task training, a linear combination of these loss functions is used:

$$CE_{\text{total}} = CE_{\text{asr}} + CE_{\text{intent}} + CE_{\text{slot}} \quad (1)$$

While the versatile cross-entropy metric allows for end-to-end model training, it serves merely as a differentiable proxy and does not directly optimize for the final SLU metrics of interest, such as:

**Speech recognition:** Word error rate (WER) computed as the ratio of word edit distance (the length of the shortest sequence of insert, delete, and substitute operations over words to transform the hypothesis to

the reference) to sequence length. A slot-WER metric that upweights critical words and not carrier phrases may also be used.

**Intent classification:** Intent classification error rate (ICER) is the primary metric for evaluating intent. This is a recall-based metric.

**Slot filling:** Semantic error rate (SemER) metric is used to evaluate jointly the intent and slot-filling performance or NLU performance. Comparing a reference of words and their accompanying tags, performance is classified as: (1) Correct slots - slot name and slot value correctly identified, (2) Deletion errors - slot name present in reference but not hypothesis, (3) Insertion errors - extraneous slot names included by hypothesis, (4) Substitution errors - correct slot name in hypothesis but with incorrect slot value. Intent classification errors are substitution errors.

$$\text{SemER} = \frac{\#Deletion + \#Insertion + \#Substitution}{\underbrace{\#Correct + \#Deletion + \#Substitution}_{\#Slots\ in\ Reference}} \quad (2)$$

The **interpretation error rate (IRER)** metric, also known as concept error rate, or simply SLU accuracy is related and is the fraction of utterances for which a semantic error has been made.

For internal datasets, we report % relative improvements in these metrics. For example, IRERR is IRER-relative reduction.

### 2.3. Sequence Loss Training

We make use of the REINFORCE framework [22, 24] to directly optimize for a non-differentiable semantic metric  $M(C)$  of interest on random candidate  $C = \{w, s, \text{intent}\} \in \mathcal{C}$  that the SLU model with weight  $\theta$  produces with probability  $\Pr(C = c|\mathbf{x}) = p(c; \theta) = p_{\text{intent}}(\text{intent}) \prod_i p_{w,i}(w_i) p_{s,i}(s_i)$ . To train the SLU model, we minimize the expected value of metric  $M$  for each utterance coupled with the cross-entropy loss  $CE$  weighted by parameter  $\lambda$ :

$$\theta^* = \text{argmin}_{\theta} \mathbb{E}[M(C)] + \lambda CE \quad (3)$$

Sub-gradient descent solvers require access to  $\nabla_{\theta} \mathbb{E}[M(C)]$ . In the *sampling approximation* to the term, we use an empirical average of an equivalent quantity,

$$\begin{aligned} \nabla_{\theta} \mathbb{E}[M(C)] &= \mathbb{E}[(M(C) - \bar{M}) \nabla_{\theta} \log p(C; \theta)] \\ &\approx \frac{1}{n} \sum_{c_i \sim p(c; \theta)} (M(c_i) - \bar{M}) \nabla_{\theta} \log p(c_i; \theta), \end{aligned} \quad (4)$$

where constant  $\bar{M}$  is used to reduce the variance of the estimate.

In the *n-best approximation*,

$$\nabla_{\theta} \mathbb{E}[M(C)] \approx \sum_{c \in \bar{\mathcal{C}}} M(c) \nabla_{\theta} \bar{p}(c; \theta), \quad (5)$$

$$\bar{p}(c; \theta) = \frac{p(c; \theta)}{\sum_{c \in \bar{\mathcal{C}}} p(c; \theta)} \quad \forall c \in \bar{\mathcal{C}} \quad (6)$$

where  $\bar{\mathcal{C}}$  is a subset of candidates, here the n-best candidates produced by performing *beam-decoding* on the ASR subsystem followed by applying the NLU model to obtain intent, slots for each candidate, is used to obtain a finite-sample approximation of the expectation. Probabilities  $\bar{p}(c; \theta)$  are obtained by zeroing out probabilities of candidates not in  $\bar{\mathcal{C}}$  and normalizing.

In either approximation, backpropagation using the non-differentiable metric  $M$  is enabled as solvers have access to  $\nabla_{\theta} p(c; \theta)$ , as  $p(c; \theta)$  is a differentiable function of weights  $\theta$ . We make use of the n-best approximation in the results section.

Thus we run both teacher-forcing to obtain  $CE_{\text{total}}$  as well as beam-decoding to obtain candidates  $\bar{\mathcal{C}}$  as demonstrated in Fig. 2. As noted in prior work [24], the cross-entropy lends stability to sequence loss training. In Table 2, we describe the choice of semantic metrics, candidate probability calculations, and regularizing cross-entropy functions for the custom sequence loss training (mSemER, mSLU, mNLU) we propose for joint ASR-NLU model training. We also

**Table 2:** By varying metric  $M$  of interest, candidate probability  $p(c; \theta)$ , and regularizing  $CE$ , different sequence loss training methods can be realized for SLU or ASR models.

Training	Metric $M$	hyp-prob $p(c; \theta)$	$CE$
mWER mSLU-ASR	WER WER + SemER	$\bar{p}(w; \theta)$ (ASR)	$CE_{\text{asr}}$
mSemER mNLU	SemER SemER + IRER + $CE_{\text{intent}}$	$\bar{p}(c; \theta)$ (ASR, NLU) Eq. (6)	$CE_{\text{total}}$ as Eq. (1)
mSLU	WER + SemER + IRER + $CE_{\text{intent}}$		
Transcript-free	SemER + IRER + $CE_{\text{intent}}$	$\bar{p}(c; \theta)$ , Eq. (6)	$\tilde{CE}_{\text{total}}$ , Eq. (7)

recover standard mWER training with WER metric, ASR candidate probability and cross-entropy. mSLU-ASR is an example of using semantic sequence losses from an external NLU model for ASR model training.

### 2.4. Application: Transcript-Free Training of ASR models

For ASR model training, ground-truth transcripts are normally required, primarily for the computation of  $CE_{\text{asr}}$ . We now show how a dataset with audio and only semantic or NLU annotations (intents, slots) and no transcript can be used to update ASR models. This weak label learning problem is motivated by deployments where human transcriptions are not available, but where an inferred semantic feedback from downstream dialogue management systems, applications or user interactions can be used to drive ASR model improvements. We focus on the case where semantic labels are available. In the absence of a reference transcript, the 1-best ASR hypothesis tokens, slots as well as the reference intent  $\tilde{c} = \{\tilde{w}, \tilde{s}, \text{intent}\}$  are treated as the reference in order to prevent catastrophic forgetting of the ASR task. The ASR subsystem is *teacher-forced* [28] with the sequence  $\tilde{w}$ , and NLU obtains the intent, slots for the resulting sequence. The cross-entropy can be computed as

$$\tilde{CE}_{\text{total}} = CE_{\text{intent}} - \sum_i \log p_{w,i}(\tilde{w}_i) + \log p_{s,i}(\tilde{s}_i), \quad (7)$$

without requiring access to a reference transcript. The NLU metrics of ICER, SemER, IRER can be computed from the available labels. The sequence loss training procedure minimizes NLU errors that also results in better ASR performance.

Note that this is not the only approach to obtaining the cross-entropy regularizer. Teacher ASR or NLU labels or mixing with dataset with transcribed audio are some alternatives.

## 3. DATA AND EXPERIMENTAL SETUP

We use datasets that include parallel speech transcripts and NLU annotations of intent and slots:

- Fluent speech dataset: Public dataset [20] of 23k utterances (15 hours) that has been processed to fit the intent, named-entity framework with 10 intents and 2 slots<sup>1</sup>
- 18 intent: Dataset of approximately 5.6M utterances (3.3k hours) with utterances from 18 intents in home automation, global, and notifications and 40 slots
- More Intent: 22M utterances (16k hours) spanning across 64 intents accounting for 90% of the data and 122 slots accounting for 99% of the slots in the data
- ASR-only 23k-hour dataset for pretraining the ASR model

<sup>1</sup>Actions are treated as intents. In addition, (inc/dec)raise\_(volume/heat) and (de)activate\_music are added to form 10 intents and 2 slots of object and location.

**Table 3:** Performance results on open and proprietary SLU datasets

(a) ASR-interface-NLU or Joint modeling approach with mSLU sequence loss training beats all baselines on the test and dev splits of the open Fluent speech dataset on IRER or accuracy

Model	Test IRER%	Dev IRER%
Transformer audio-intent [29]	2.5	-
Baseline [20]	1.2	-
AT-AT (SOTA) [30]	0.5	-
Oracle neural NLU	0.00	0.00
Compositional ASR→NLU	0.42	2.15
ASR-Gumbel-NLU	0.40	2.05
Joint SLU - no seq training	<b>0.39</b>	2.05
Joint mSLU	<b>0.39</b>	<b>1.89</b>

(c) Comparison of compositional baselines and sequence loss approaches on the MoreIntent eval set of 500k utterances. Performance figures are relative % improvement to row M1 shown as 0%

	Model	WERR%	SemERR%	ICERR%
M1	Compositional	0	0	0
M2	Comp mWER	<b>9.40</b>	1.59	0.71
M2a	Comp mSLU-ASR	6.95	3.97	0.95
M3	Joint mSLU	6.53	<b>6.73</b>	<b>1.98</b>

**Training details:** The audio feature is composed of 3 stacked 25 ms LFBE frames with 10 ms shift. This LAS model has 77M parameters: 5x512 BLSTM encoder, 2x1024 LSTM decoder with 4 attention heads of depth 256, projection 728, 4500 subword vocabulary. The NLU model has 4 (text interface)-11 (neural network interface) million parameters with a 2x512 BLSTM encoder, a dense layer for slots, and 2x512 relu feed-forward layers for intent. We also experimented with a 3M parameter Transformer-encoder (2 layers, 8 attention heads, 256 units) NLU model. The LAS model is first pretrained on the 23k hour dataset and finetuned on the specific dataset. With ASR now frozen, NLU is first trained in joint systems followed by joint ASR-NLU fine-tuning using sequence losses. In the 18 intent dataset, NLU is trained in the joint system for 6 epochs followed by sequence loss training for 2, taking 1 day on 8 Nvidia Tesla V100 GPUs.

#### 4. RESULTS AND DISCUSSION

##### Sequence loss training beats baselines

On the open Fluent speech dataset in Table 3a, we see all ASR-interface-NLU models beat external baselines that directly extract semantics from audio without intermediate transcript showing utility of ASR pretraining. Both the neural and Gumbel-softmax interface joint models outperform compositional text baselines. The joint model with mSLU sequence loss training is the best-performing model as seen by results on both dev and test splits. This can be categorized as a small dataset of lower semantic complexity as the oracle NLU model perfectly recovers semantics from ground truth transcripts.

In the 18-intent dataset results of Table 3b, the NLU metrics degrade substantially from row 1 (NLU consuming ground-truth transcript) to row 2 (NLU consuming ASR hypothesis), showing impact of ASR errors. In row 2a, the LAS model is further trained with mWER sequence loss, leading to gains in WER as well as NLU metrics. The joint model with mSLU sequence loss training results in best ASR and NLU metrics. From rows 2a and 4b, we see worse WER for the joint model, but better NLU metrics showing that joint training improves ASR performance relevant to downstream NLU. In row T1, mSemER training was used with jointly trained LAS ASR and Transformer-encoder NLU system; this has 1M fewer parameters than joint models with LSTM-based NLU, but shows better performance.

##### Sequence loss training optimizes a metric of interest

Table 3b shows the impact of the non-differentiable metric  $M$  to optimize on ASR and NLU performance. mWER training optimizes for WER but this may not reflect its optimal NLU metrics (row 2a vs 4). In rows 4a-c, we use metrics rooted in different definitions of semantic error. The mNLU metric optimizing SemER, IRER, ICER

(b) Comparison of compositional models & joint models with various sequence loss approaches on the 18-intent eval set of 700k utterances. Performance figures are relative % improvement from row 2 shown as 0%

	Model	WERR%	SemERR%	IRER%	ICER%
1	Oracle NNLU	-	41.17	42.93	56.34
2	Compositional: LAS →NLU	0	0	0	0
2a	Comp. mWER LAS → NLU	6.23	1.07	0.96	2.82
3a	LAS-Gumbel-NLU	2.04	1.50	0.12	3.87
4a	Joint mSemER	6.87	5.66	3.12	7.68
4b	Joint mNLU	5.45	<b>5.92</b>	<b>3.26</b>	8.91
4c	Joint mSLU	<b>7.67</b>	<b>5.91</b>	3.08	<b>11.58</b>
T1	Transformer-NLU Joint mSemER	7.46	<b>6.29</b>	<b>4.16</b>	<b>11.62</b>

(d) Relative % improvement from a baseline Joint ASR-NLU model with transcript-free training on the 18 intent and Moreintent datasets

Dataset	WERR%	SemERR%	ICERR%
18-intent	2.19	5.87	10.49
Moreintent	1.01	5.88	1.77

leads to better ICER and IRER than mSemER metric training that optimizes only mSemER. mSLU training (adds WER to mNLU) shows the best ASR performance, reflecting the importance of semantic feedback even for ASR training. Thus we can customize any sequence loss to optimize model performance metric(s).

##### Results on a general dataset

The conclusions from 15 and 3k hours datasets carry over to the large 16k hour *Moreintent* dataset seen in Table 3c. Row M2 primarily shows improvements in WER from mWER training of ASR resulting in fewer SLU errors. M2a is an example of semantic sequence loss training of ASR. However, the joint model of M3 trained to optimize SLU metrics shows the best NLU performance. We thus have a recipe to improve ASR and NLU model performance: train an ASR model with mWER sequence loss. The ASR subsystem in the joint model is initialized with these weights and the entire system is trained minimizing SLU sequence losses.

##### Both ASR and NLU improve with transcript-free training

In Table 3d, we update models from a common starting point using weak-feedback training with only NLU labels. A 5% relative improvement in SemER is seen for both the 18-intent and *Moreintent* datasets, and modest ASR improvements suggesting that semantic feedback alone can be used to improve both ASR and SLU.

#### 5. CONCLUSION

Edge deployments of ASR and SLU systems for voice activated assistants require the development of low-footprint performant models. Prior approaches involving either pipelined ASR and NLU models or end-to-end SLU models use the differentiable cross-entropy loss to train, but these do not map to metrics of interest such as word and semantic error rates. In this work, we propose non-differentiable semantic sequence losses and use the REINFORCE framework to train ASR and SLU models. Joint training with custom sequence losses lets ASR be trained with semantic feedback from NLU, and NLU be trained aware of ASR errors. We show that both ASR and NLU performance metrics of SLU systems improve across a range of open and proprietary datasets and beat state-of-the-art models. We also improve and update ASR systems without access to transcripts using weak-feedback via NLU labels alone.

**Acknowledgement:** We thank Bach, Ehry, Chul, Shehzad, and reviewers for helpful technical comments. Abhinav Khattar assisted with Transformers, Jinxi Guo with mWER discussions, and Zhe Zhang with data preparation.

## 6. REFERENCES

- [1] Alex Graves, “Sequence transduction with recurrent neural networks,” *arXiv preprint arXiv:1211.3711*, 2012.
- [2] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 369–376.
- [3] Shigeki Karita, Nanxin Chen, Tomoki Hayashi, Takaaki Hori, Hirofumi Inaguma, Ziyang Jiang, Masao Someki, Nelson Enrique Yalta Soplín, Ryuichi Yamamoto, Xiaofei Wang, et al., “A comparative study on Transformer vs RNN in speech applications,” in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 449–456.
- [4] William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals, “Listen, Attend and Spell: A neural network for large vocabulary conversational speech recognition,” in *ICASSP*, 2016.
- [5] Chung-Cheng Chiu, Tara N Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjali Kannan, Ron J Weiss, Kanishka Rao, Ekaterina 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] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer, “Neural architectures for named entity recognition,” *arXiv preprint arXiv:1603.01360*, 2016.
- [7] Young-Bum Kim, Sungjin Lee, and Karl Stratos, “Onenet: Joint domain, intent, slot prediction for spoken language understanding,” in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2017, pp. 547–553.
- [8] Yonghui Wu, Min Jiang, Jun Xu, Degui Zhi, and Hua Xu, “Clinical named entity recognition using deep learning models,” in *AMIA Annual Symposium Proceedings*. American Medical Informatics Association, 2017, vol. 2017, p. 1812.
- [9] Qian Chen, Zhu Zhuo, and Wen Wang, “Bert for joint intent classification and slot filling,” *arXiv preprint arXiv:1902.10909*, 2019.
- [10] Tanja Bunk, Daksh Varshneya, Vladimir Vlasov, and Alan Nichol, “Diet: Lightweight language understanding for dialogue systems,” *arXiv preprint arXiv:2004.09936*, 2020.
- [11] Dilek Hakkani-Tür, Frédéric Béchet, Giuseppe Riccardi, and Gokhan Tur, “Beyond ASR 1-best: Using word confusion networks in spoken language understanding,” *Computer Speech & Language*, vol. 20, no. 4, pp. 495–514, 2006.
- [12] Matthew Henderson, Milica Gašić, Blaise Thomson, Pirros Tsiakoulis, Kai Yu, and Steve Young, “Discriminative spoken language understanding using word confusion networks,” in *2012 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2012, pp. 176–181.
- [13] Gokhan Tur, Jerry Wright, Allen Gorin, Giuseppe Riccardi, and Dilek Hakkani-Tür, “Improving spoken language understanding using word confusion networks,” in *Seventh International Conference on Spoken Language Processing*, 2002.
- [14] Chao-Wei Huang and Yun-Nung 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.
- [15] Parisa Haghani, Arun Narayanan, Michiel Bacchiani, Galen Chuang, Neeraj Gaur, Pedro Moreno, Rohit Prabhavalkar, Zhongdi Qu, and Austin 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.
- [16] 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.
- [17] 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.
- [18] Yao Qian, Rutuja Ubale, Vikram Ramanaryanan, Patrick Lange, David Suendermann-Oeft, Keelan Evanini, and Eugene Tsuprun, “Exploring ASR-free end-to-end modeling to improve spoken language understanding in a cloud-based dialog system,” in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2017, pp. 569–576.
- [19] Natalia Tomashenko, Christian Raymond, Antoine Caubrière, Renato De Mori, and Yannick Estève, “Dialogue history integration into end-to-end signal-to-concept spoken language understanding systems,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 8509–8513.
- [20] Loren Lugosch, Mirco Ravanelli, Patrick Ignoto, Vikrant Singh Tomar, and Yoshua Bengio, “Speech model pre-training for end-to-end spoken language understanding,” *arXiv preprint arXiv:1904.03670*, 2019.
- [21] Milind Rao, Anirudh Raju, Pranav Dheram, Bach Bui, and Ariya Rashtrow, “Speech to Semantics: Improve ASR and NLU Jointly via All-Neural Interfaces,” in *Proc. Interspeech*, 2020, pp. 876–880.
- [22] Ronald J Williams, “Simple statistical gradient-following algorithms for connectionist reinforcement learning,” *Machine learning*, vol. 8, no. 3-4, pp. 229–256, 1992.
- [23] Biing-Hwang Juang, Wu Hou, and Chin-Hui Lee, “Minimum classification error rate methods for speech recognition,” *IEEE Transactions on Speech and Audio processing*, vol. 5, no. 3, pp. 257–265, 1997.
- [24] Rohit Prabhavalkar, Tara N Sainath, Yonghui Wu, Patrick Nguyen, Zhifeng Chen, Chung-Cheng Chiu, and Anjali 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.
- [25] Jinxi Guo, Gautam Tiwari, Jasha Droppo, Maarten Van Segbroeck, Che-Wei Huang, Andreas Stolcke, and Roland Maas, “Efficient Minimum Word Error Rate Training of RNN-Transducer for End-to-End Speech Recognition,” in *Proc. Interspeech*, 2020, pp. 2807–2811.
- [26] Yaser Keneshloo, Tian Shi, Naren Ramakrishnan, and Chandan K Reddy, “Deep reinforcement learning for sequence-to-sequence models,” *IEEE transactions on neural networks and learning systems*, vol. 31, no. 7, pp. 2469–2489, 2019.
- [27] Eric Jang, Shixiang Gu, and Ben Poole, “Categorical reparameterization with Gumbel-softmax,” *arXiv preprint arXiv:1611.01144*, 2016.
- [28] Ronald J Williams and David Zipser, “A learning algorithm for continually running fully recurrent neural networks,” *Neural computation*, vol. 1, no. 2, pp. 270–280, 1989.
- [29] Martin Radfar, Athanasios Mouchtaris, and Siegfried Kunzmann, “End-to-End Neural Transformer Based Spoken Language Understanding,” in *Proc. Interspeech*, 2020, pp. 866–870.
- [30] Subendhu Rongali, Beiye Liu, Liwei Cai, Konstantine Arkoudas, Chengwei Su, and Wael Hamza, “Exploring transfer learning for end-to-end spoken language understanding,” *arXiv preprint arXiv:2012.08549*, 2020.