ATTENTIVE CONTEXTUAL CARRYOVER FOR MULTI-TURN END-TO-END SPOKEN LANGUAGE UNDERSTANDING

Kai Wei[*]*, Thanh Tran*[*]*, Feng-Ju Chang, Kanthashree Mysore Sathyendra, Thejaswi Muniyappa, Jing Liu, Anirudh Raju, Ross McGowan, Nathan Susanj, Ariya Rastrow, Grant P. Strimel*

Alexa Speech, Amazon

ABSTRACT

Recent years have seen significant advances in end-to-end (E2E) spoken language understanding (SLU) systems, which directly predict intents and slots from spoken audio. While dialogue history has been exploited to improve conventional text-based natural language understanding systems, current E2E SLU approaches have not yet incorporated such critical contextual signals in multi-turn and task-oriented dialogues. In this work, we propose a contextual E2E SLU model architecture that uses a multi-head attention mechanism over encoded previous utterances and dialogue acts (actions taken by the voice assistant) of a multi-turn dialogue. We detail alternative methods to integrate these contexts into the state-of-the-art recurrent and transformer-based models. When applied to a large de-identified dataset of utterances collected by a voice assistant, our method reduces average word and semantic error rates by 10.8% and 12.6%, respectively. We also present results on a publicly available dataset and show that our method significantly improves performance over a non-contextual baseline.

Index Terms— Spoken language understanding, multi-turn, attention, contextual, RNN/Transformer-Transducer

1. INTRODUCTION

End-to-end (E2E) spoken language understanding (SLU) aims to infer intents and slots from spoken audio via a single neural network. For example, when a user says *order some apples*, the model maps this spoken utterance (in the form of audio) to the intent *Shopping* and slots such as *Apple: Item*. Recent research has made significant advances in E2E SLU [1–6]. Notably, [6] develops a jointly trained E2E model, consisting of automatic speech recognition (ASR) and natural language understanding (NLU) models connected by a differentiable neural interface, that outperforms the compositional SLU where ASR and NLU models are trained separately. Yet, how to incorporate contexts into E2E SLU remains unexplored.

Contexts have been shown to significantly improve performance separately for ASR [7–15] and NLU [11, 16–21].

[*] Equal contribution

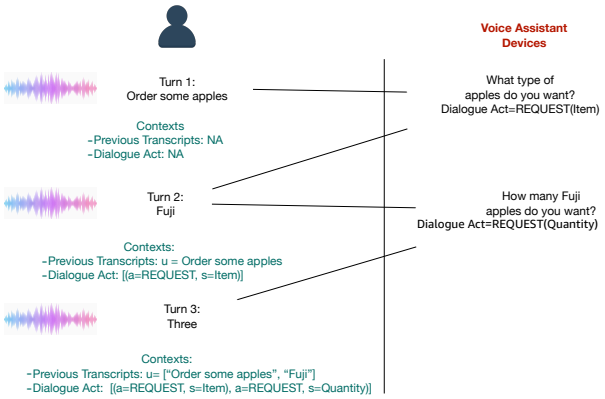

Fig. 1. A multi-turn dialogue example.

For example, [14] proposes a multi-hot encoding to incorporate contextual information into a RNN transducer network (RNN-T) via the speech encoder sub-network and found that contexts such as dialogue state could improve accuracy of ASR. [8] uses cross-attention mechanism in an E2E speech recognizer decoder for two-party conversations. [20] has shown that encoding dialogue acts using a feedforward network from dialogue history resulted in a faster and more generalizable model without any accuracy degradation compared to [21]. [22] encodes historical utterances with the BiLSTM and external knowledge with ConceptNet.

In this work, we propose a novel approach to encode dialogue history in a multi-turn E2E SLU system. Figure 1 illustrates a task-oriented turn-by-turn dialogue between a user and a voice assistant (VA). In this figure, the first turn is *order some apples*. To clarify the apple type, the VA asks *What type of apples do you want?* at the second turn; and the user's answer is *Fuji*. To clarify the quantity, the VA asks *How many Fuji apples do you want?* at the third turn; and the user's answer is *three*. If *three* is treated as a single-turn utterance, it is ambiguous since it can mean three apples or three o'clock. However, this utterance can correctly be interpreted as *three apples* when presented with previous dialogue contexts (e.g., *order some apples* and *Fuji*). Prior E2E SLU research has focused on single-turn interactions where the VA receives the user's speech signals from just the current turn. They ignore the relevant contexts from previous turns that can enhance

the VA's ability to correctly disambiguate user's intent.

In contrast to prior works, where dialogue acts are encoded singularly for ASR (e.g., [14]) or NLU(e.g., [20]), we encode both dialogue acts and previous utterances to improve an E2E SLU architecture. Specifically, we propose a multi-head gated attention mechanism to encode dialogue contexts. The attention-based context can be integrated at different layers of a neural E2E SLU model. We explore variants where either the audio frames, the neural interface layer (from ASR to NLU), or both are supplemented by the attention-based context vectors. Furthermore, the learnable gating mechanism in our proposed multi-head gated attention can downscale the contribution of the context when needed. Our proposed approach improves the performance of the state-of-the-art E2E SLU models – namely recurrent neural network transducer SLU and transformer transducer SLU on both internal industrial voice assistant datasets and publicly available ones.

2. PROBLEM DEFINITION

We formulate the problem of a multi-turn E2E SLU as follows: In a multi-turn setting, a *dialogue* between a user and the voice assistant system has T turns. Each turn $t \in [1, T]$ extends a growing list of dialogue acts $\mathcal{F}^t$=$\{(a_1, s_1), \ldots, (a_{t-1}, s_{t-1})\}$ corresponding to the preceding system responses and a list of the user's previous utterance transcripts $\mathcal{U}^t$=$\{u_1, u_2, \ldots, u_{t-1}\}$. Each dialogue act (a_j, s_j) in $\mathcal{F}^t$ comprises of a dialogue action a_j from an action set $\mathcal{A}$ and a dialogue slot s_j from a slot set $\mathcal{S}$. Take the second turn in Figure 1 as an example: the previous utterance u_2 = *Fuji*, the dialogue action a_2 = *REQUEST* and the dialogue slot s_2 = *Item*.

Inputs and Outputs: The inputs of each turn t include acoustic input and dialogue contexts. The acoustic input X^t comprises of a sequence of n frame-based acoustic frames, X^t=$\{\boldsymbol{x_1^t}, \boldsymbol{x_2^t}, \ldots, \boldsymbol{x_n^t}\}$. Dialogue contexts include preceding dialogue acts $\mathcal{F}^t$, and the previous utterance transcripts $\mathcal{U}^t$. Our goal is to build a contextual neural E2E SLU architecture that correctly generates transcription and semantic outputs for each spoken turn, namely intent $y^{\text{int.}}$, transcript token sequence $\{y^{\text{tok.}}\}$, and slot sequence (one per token) $\{y^{\text{slot}}\}$.

3. PROPOSED CONTEXTUAL E2E SLU

The proposed contextual E2E SLU architecture consists of a context encoder component, a context combiner, and a base E2E SLU model. The base model consists of ASR and neural NLU modules jointly trained via a differentiable neural interface [6, 23], which has been shown to achieve state-of-the-art SLU performance.

Figure 2 shows the contextual E2E SLU model architecture using speech encoder context ingestion. The context encoder, described in Section 3.1, converts dialogue acts and utterance transcriptions of previous turns into contextual embeddings. The contextual embeddings are then combined with input audio features X^t=$\{\boldsymbol{x_1^t}, \boldsymbol{x_2^t}, \ldots, \boldsymbol{x_n^t}\}$ (described in Section 3.2) and then processed by the ASR module to obtain the output sequence $\boldsymbol{y} = \{y_1^{\text{tok.}}, \ldots, y_m^{\text{tok.}}\}$, where the outputs $y_i^{\text{tok.}}$ are transcription graphemes, word or subword units [24]. Context encoder embeddings are trained along with the rest of the E2E SLU architecture. The hidden interface (or ASR-NLU interface) [25] is connected to the speech encoder via the joint network, which is a feedforward neural network that combines the outputs from the encoder and prediction network. This interface passes the intermediate hidden representation sequence H^t=$\{\boldsymbol{h_1^t}, \boldsymbol{h_2^t}, \ldots, \boldsymbol{h_m^t}\}$ to a neural NLU module that predicts intents $y^{\text{int.}}$ and a sequence of predicted slots, one per token, $\{y^{\text{slot}}\}$. Our objective is to minimize the E2E SLU loss: $\mathcal{L}_{\text{total}} = \lambda_1 \mathcal{L}_{\text{tok.}} + \lambda_2 \mathcal{L}_{\text{slot}} + \lambda_3 \mathcal{L}_{\text{int.}}$, where $\mathcal{L}_{\text{tok.}}$ is the loss for word prediction, $\mathcal{L}_{\text{slot}}$ is the loss for slot prediction, and $\mathcal{L}_{\text{int.}}$ is the loss for intent prediction. The following section describes context encoder in detail.

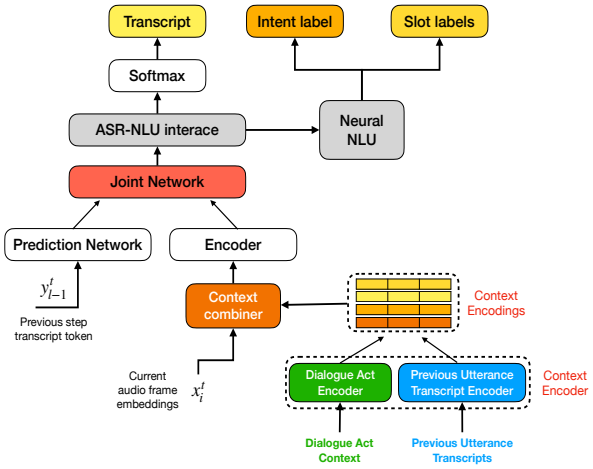

Fig. 2. Contextual joint SLU model architecture using speech encoder context ingestion.

3.1. Context Encoder

In this section, we describe approaches to encode dialogue acts and previous utterance transcripts. We first describe the *Dialogue Act Encoder* that encodes the dialogue acts. Then, we describe the *Previous Utterance Transcript Encoder* that encodes transcripts from previous utterances.

3.1.1. Dialogue Act Encoder

Input: For the t-th turn, a list of dialogue acts for all previous turns denoted by $\mathcal{F}^t$=$\{(a_1, s_1), \ldots, (a_{t-1}, s_{t-1})\}$ is provided as the input. We set the maximum number of dialogue action-slot pairs to l_a. If $\mathcal{F}^t$ has less than l_a dialogue action-slot pairs, we pad to length l_a with a default action and slot.

Embedding layer: The embedding layer maintains two embedding matrices - a dialogue action embedding matrix $M^{\mathcal{A}} \in$

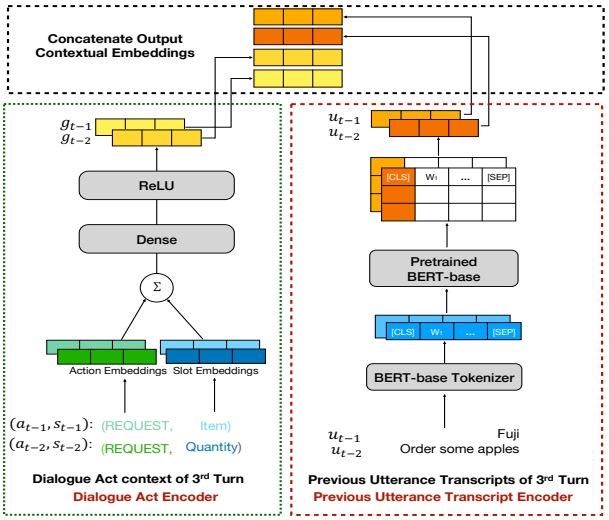

Fig. 3. Encoding previous utterance transcripts and dialogue contexts in a multi-turn dialog between a user and a voice assistant system.

$\mathbb{R}^{|\mathcal{A}|\times d}$, and a dialogue slot embedding matrix $M^{\mathcal{S}} \in \mathbb{R}^{|\mathcal{S}|\times d}$, with $|\mathcal{A}|$ and $|\mathcal{S}|$ referring to the total number of unique dialogue actions and slot types in the system, respectively. By passing each dialogue action a_j and dialogue slot s_j through their respective embedding matrices, we obtain their corresponding embeddings $\boldsymbol{a_j}$ and $\boldsymbol{s_j}$.

Encoding layer: Given the dialogue action and slot embeddings, $\boldsymbol{a_j}$ and $\boldsymbol{s_j}$, we fuse both embeddings via an element-wise addition followed by a nonlinear transformation with a $ReLU$ activation [20] as summarized below.

$$\boldsymbol{g_j} = ReLU\big(W^g(\boldsymbol{a_j} + \boldsymbol{s_j})\big) \tag{1}$$

Output: We produce the output G^t as a stack of dialogue act embeddings by aggregation of the list of $\boldsymbol{g_{t-l_a}}, \ldots, \boldsymbol{g_{t-1}}$.

3.1.2. Previous Utterance Transcript Encoder

Input: A list of previous utterance transcripts in the dialogue denoted by $\mathcal{U}^t$=$\{u_1, u_2, \ldots, u_{t-1}\}$. For each previous utterance transcript u_k, we first tokenize it using the pre-trained BERT-base tokenizer. Next, we prepend a [CLS] token and append a [SEP] token to the tokenized transcript. We set the maximum number of previous utterance transcripts to l_b. We pad empty sequences for $\mathcal{U}^t$ if its length is less than l_b, and take the l_b latest sequences in $\mathcal{U}^t$ if its length is greater than l_b.

Encoding layer: From the tokenized transcripts, we apply the pre-trained BERT-base [26] model to obtain an utterance transcript embedding $\boldsymbol{u_k}$ for each previous utterance u_k where we use the [CLS] token embedding as the summarized embedding for a full utterance transcript.

Output: Similar to G^t, we output U^t by stacking the list of utterance embeddings $\boldsymbol{u_{t-l_b}}, \ldots, \boldsymbol{u_{t-1}}$ from previous turns.

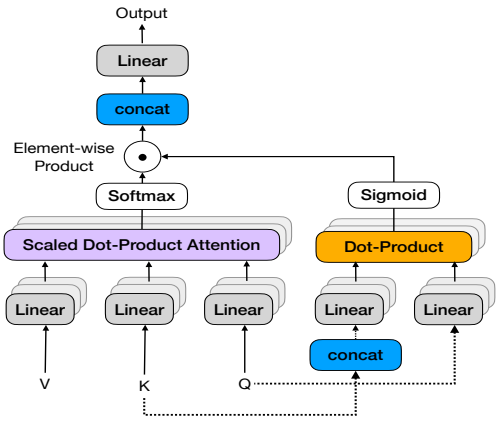

Fig. 4. Architecture of Gated Multi-head attentions.

3.2. Context Combiner

The context combiner combines the context encodings G^t and U^t to create the final context vectors that are fed into the model. We explore different ways to combine the context encodings into the model: (i) averaged contextual carryover, (ii) attentive contextual carryover, and (iii) gated attentive contextual carryover.

To illustrate, we detail our approaches with an example that combines dialogue act encodings G^t and the previous utterance transcript encodings U^t with the acoustic embeddings X^t=$\{\boldsymbol{x_1^t}, \boldsymbol{x_2^t}, \ldots, \boldsymbol{x_n^t}\}$ of the t-th turn. Note that the same process can be applied to combine context encodings at different ingestion points in the model (see Section 3.2.3). We describe our context combiner methods below.

3.2.1. Averaged Contextual Carryover

Recall that G^t is the stack of dialogue act contextual embeddings and U^t is previous utterance transcript embeddings at turn t. In this method, we first compute the average embeddings [14] of all dialogue act contextual embeddings $\boldsymbol{g_j} \in G^t$ and average encodings of all previous utterance transcript embeddings $\boldsymbol{u_k} \in U^t$. Then, we combine the averaged contextual embeddings with the input by concatenating them with the acoustic embeddings, $\{\boldsymbol{x_1^t}, \boldsymbol{x_2^t}, \ldots, \boldsymbol{x_n^t}\}$, for each acoustic time step, as follows:

$$\begin{gathered} \bar{\boldsymbol{g}}^{\boldsymbol{t}} = \frac{1}{l_a} \sum_{\boldsymbol{g_j} \in G^t} \boldsymbol{g_j} \quad \bar{\boldsymbol{u}}^{\boldsymbol{t}} = \frac{1}{l_b} \sum_{\boldsymbol{u_k} \in U^t} \boldsymbol{u_k} \\ \boldsymbol{c^t} = \big[\bar{\boldsymbol{g}}^{\boldsymbol{t}}; \bar{\boldsymbol{u}}^{\boldsymbol{t}}\big] \\ \{\boldsymbol{x_1^t}\prime, ..., \boldsymbol{x_n^t}\prime\} = \big\{\big[\boldsymbol{x_1^t}; \boldsymbol{c^t}\big], \ldots, \big[\boldsymbol{x_n^t}; \boldsymbol{c^t}\big]\big\} \end{gathered} \tag{2}$$

3.2.2. Attentive Contextual Carryover

Averaging contextual embeddings of the previous turns can hamper the ability of the model to access fine-grained contextual information for a specific turn and time step. Therefore, we utilize the multi-head attention mechanism [27], which uses acoustic embeddings, of each time step, to attend to relevant dialogue contexts and create the final contextual embeddings.

Specifically, we create the queries, keys, and values, $Q_i, K_i, V_i, i \in \{g, u\}$ via linear projections as follows:

$$Q_g = W_g^{(q)} X^t;\ K_g = W_g^{(k)} G^t;\ V_g = W_g^{(v)} G^t$$
$$Q_u = W_u^{(q)} X^t;\ K_u = W_u^{(k)} U^t;\ V_u = W_u^{(v)} U^t \quad (3)$$

Here, X^t, G^t, U^t are acoustic, dialogue act, and previous utterance embeddings for the t-th turn, respectively. Matrices $W_g^{(\cdot)}, W_u^{(\cdot)}$ are learnt linear projections. A scaled dot-product attention is then used to calculate the final dialogue and utterance context vectors through the weighted sum of projected contextual embeddings of the previous turns. This process is formulated as:

$$\boldsymbol{\alpha_g} = \text{Softmax}\left(\frac{Q_g K_g^\top}{\sqrt{d}}\right), \quad C_g^t = \boldsymbol{\alpha_g} V_g$$
$$\boldsymbol{\alpha_u} = \text{Softmax}\left(\frac{Q_u K_u^\top}{\sqrt{d}}\right), \quad C_u^t = \boldsymbol{\alpha_u} V_u \quad (4)$$

where d is the hidden size of the attention layer applied for numerical stability [27]. The attention outputs C_g^t and C_u^t are then concatenated with the acoustic embeddings X^t provided as input.

3.2.3. Gated Attentive Contextual Carryover

One limitation of the attention mechanism is that it cannot downscale the contribution of a context when needed [28]. Take a two-turn dialogue as an example:

> A user asks a voice assistant to *call uncle sam* in the first turn, and the system confirms back to see if the user wants to call *Uncle Sam's Sandwich Bar* (associated dialogue act is REQUEST(restaurant)). Then, in the second turn, the user corrects that she wants to "call my uncle sam".

In this case, simply applying multi-head attention as described in Eq.(3)-(4) on the previous turn utterance *call uncle sam*, U^t, and dialogue act *REQUEST(restaurant)*, G^t, can lead to a wrong interpretation for the second turn. This is because the results of the Softmax function in Eq.(4) assigns dialogue act context to positive scores, misleadingly associating *uncle sam* with a *restaurant name* rather than a *person name*.

Inspired by the gating mechanism to control information flow or integrate different types of information [8, 29–33], we introduce a learnable gating mechanism on top of the multi-head attentive contextual carryover to further reduce a context's influence when it does not help the interpretation. Specifically, we concatenate all the contextual embeddings in G^t and U^t to obtain C_c^t. Then, we obtain the gating scores by computing the similarity between the linearly projected X^t and C_c^t, as follows:

$$\beta_c = \text{sigmoid}\left(Q_c K_c^\top\right), \quad (5)$$

$$Q_c = W_c^{(q)} X^t;\ K_c = W_c^{(k)} C_c^t$$

where $W_c^{(q)}$ and $W_c^{(k)}$ are learnable parameters. $\beta_c \in \mathbb{R}^{n \times 1}$ and n is the number of frames. Each entry β_c shows how much contexts contribute to the acoustic embedding $\boldsymbol{x_i^t}$ at i-th frame, $i \in [1, n]$). We replicate β_c to make it have the same dimension as $\boldsymbol{\alpha}_g$ and $\boldsymbol{\alpha}_u$. The gated attention scores $\boldsymbol{\gamma}$ are then computed by the element-wise product between $\boldsymbol{\alpha}$ scores and $\boldsymbol{\beta_c}$:

$$\boldsymbol{\gamma_g} = \boldsymbol{\alpha_g} \odot \boldsymbol{\beta_c} \quad \boldsymbol{\gamma_u} = \boldsymbol{\alpha_u} \odot \boldsymbol{\beta_c} \quad (6)$$

We compute the gated attentive contextual embeddings across each attention head, as follows:

$$C_{g,\text{gated}}^t = \boldsymbol{\gamma_g} V_g;\ C_{u,\text{gated}}^t = \boldsymbol{\gamma_u} V_u. \quad (7)$$

Finally, $C_{g,\text{gated}}^t$ and $C_{u,\text{gated}}^t$ are row-wise concatenated with the acoustic embeddings X^t as input.

3.3. Context Ingestion Scenarios

We consider the integration of the context encoder using three schemes: ingestion by the speech encoder network, ingestion with the hidden ASR-NLU interface, and finally at both insertion points.

Speech encoder ingestion: In this method, we incorporate the outputted context embeddings only into the acoustic embeddings for ASR pre-training/training task. This approach is motivated by prior research showing that context benefits the speech encoder more than the prediction network of ASR transducer models [15] . To combine context with acoustic embeddings, we input the acoustic embeddings X^t=$\{\boldsymbol{x_1^t}, \boldsymbol{x_2^t}, \ldots, \boldsymbol{x_n^t}\}$ as the query, and the context encodings G^t, U^t serve as the keys and values in the context combiner. The output $\{\boldsymbol{x_1^t}\prime, ..., \boldsymbol{x_n^t}\prime\}$ with ingested context (Equation (2)) are then used to perform the ASR task.

ASR-NLU interface ingestion: In this approach, we ingest the output context embeddings only into the ASR-NLU interface embeddings for the SLU training task. As such, we now use the ASR-NLU interface embeddings H^t=$\{\boldsymbol{h_1^t}, \boldsymbol{h_2^t}, \ldots, \boldsymbol{h_m^t}\}$ as queries for context combiner instead of the acoustics.

Shared context ingestion: In this method, we integrate context into both acoustic embeddings and ASR-NLU interface embeddings. We maintain a shared context encoder between the ASR and NLU submodule, resulting in a shared G^t, U^t between them. For fusion, we maintain two separated context combiners to increase the context ingestion flexibility. Specifically, we establish a gated multi-head attentive context combiner for the ASR submodule with X^t as queries, while having another gated multi-head attentive context combiner for the NLU submodule with H^t as queries.

In the following sections, we perform experiments on incorporating multi-turn context into two SLU architectures: a Transformer-based Joint SLU model and an RNN-T based Joint SLU model.

4. EXPERIMENTAL SETTINGS

4.1. Datasets

The internal industrial voice assistant (IVA) dataset is a far-field dataset with more than 10k hours of audio data and their corresponding intent and slot annotations. It is a multi-domain dataset with both single-turn and multi-turn utterances. In total, there are 55 intents, 183 slot types, and 49 dialogue acts. In addition, we built a synthetic and publicly available multi-turn E2E SLU (Syn-Multi) dataset based on [34]. [34] contains two datasets with a text-only format from Restaurant (11,234 turns in 1,116 training dialogues) and Movie (3,562 turns in 384 training dialogues) domains. To obtain audio signals, we used a Transformer text-to-speech model [1] to synthesize the audio and combine the two datasets into one dataset for model training and evaluation. Finally, we used SpecAugment [35] to augment audio feature inputs. In total, Syn-Multi has 3 intents, 12 slot types, and 21 user dialogue act types[2].

4.2. Implementation setup

Audio features: The input audio features are 64-dimensional LFBE features extracted every 10 ms with a window size of 25 ms from audio samples. The features of each audio frame are stacked with the features of two previous audio frames, followed by a downsampling factor of 3 to achieve a low frame rate, resulting in 192 feature dimensions per audio frame. We use a token set with 4,000 wordpieces trained by the sentence-piece tokenization model [36].

Model setup: Table 1 shows our model setup details. We built contextual E2E SLU models based on the Recurrent Neural Network Transducer (RNN-T) [37] and the Transformer Transducer (T-T) [38], respectively. E2E SLU models share an audio encoder network that encodes LFBE features, a prediction network that encodes a sequence of predicted wordpieces, a joint network that combines the encoder and the prediction network, and an NLU tagger that predicts intents and slots. The intent tagger contains two feedforward layers before projecting into the number of intents, and the slot tagger directly takes the output embeddings from the NLU tagger and projects them into the slot size. The audio encoder in the E2E T-T SLU and E2E RNN-T SLU are Transformer layers (with 4 attention heads) and LSTM layers, respectively. The NLU tagger in E2E T-T SLU and E2E RNN-T SLU are transformer layers (with 8 attention heads) and BiLSTM layers, respectively. For l_a (the maximum number of dialog action-slot pairs) and l_b (the maximum number of previous utterance transcripts), we set $l_a = l_b = 5$ in the IVA dataset. We set $l_a = l_b = 20$ in the Syn-Multi dataset.

Training setup: We adopt a stage-wise joint training strategy for the proposed contextual models and baseline non-contextual models. We first pre-trained an ASR model to minimize the RNN-T loss [24]. We then freeze the ASR module to train the NLU module to minimize the cross entropy losses for the intent and slot predictions. During training, all constituent subwords of a word are tagged with its slot. During inference, the constituent subwords are combined to form the word, and the slot tag for the last constituent subwords is taken as the slot tag for the word. Last, we jointly tuned ASR and NLU modules to minimize all three losses. We used the teacher forcing technique [39] that uses the human-annotated transcripts of previous turns for training, and the automatic transcripts of previous turns from our model for inference. We applied the Adam optimizer [40] for all model training. For E2E RNN-T SLU, the learning rate is warmed linearly from 0 to 5×10^{-4} during the first 3K steps, held constant until 150K steps, and then decays exponentially to 10^{-5} until 620K steps. For E2E T-T SLU, the learning rate is warmed from 0 to 5×10^{-4} in the first 16K steps, then is decayed to 10^{-5} in the following 604K steps exponentially. We used 24 NVIDIA® V100 Tensor Core GPUs and a batch size of 32 for training the model.

	IVA Dataset		Syn-Multi Dataset	
Statistic	RNN-T SLU	T-T SLU	RNN-T SLU	T-T SLU
Audio encoder network				
# Layers	5	6	4	2
Layer embed-size	736	256	640	256
# Attention heads	–	4	–	4
#FeedForward layer	1	1	1	1
FeedForward embed-size	512	2048	256	512
Prediction network				
# Layers	2	2	2	2
Layer embed-size	736	736	640	640
#FeedForward layer	1	1	1	1
FeedForward embed-size	512	512	256	512
Joint network				
Vocab embed-size	512	512	512	512
#FeedForward layer	1	1	1	1
FeedForward embed-size	512	512	512	512
Activation	tanh	tanh	tanh	tanh
NLU decoder network				
# Layers	2	2	2	2
Layer embed-size	256	256	256	256
#FeedForward layer	1	1	1	1
Feedforward size	256	256	256	256
Intent Predictor Network				
#FeedForward layer	2	2	2	2
Feedforward size	512	512	512	512
Activation	relu	relu	relu	relu
#FeedForward layer	1	1	1	1
Feedforward size	#intent	#intent	#intent	#intent
Slot Tagger Network				
#FeedForward layer	1	1	1	1
Feedforward size	#slots	#slots	#slots	#slots
# Attention heads	–	8	–	8

Table 1. Model setup for E2E SLU.

4.3. Evaluation Metrics and Baselines

We evaluate the model performance on word error rate (WER), intent classification error rate (ICER), and semantic error rate (SemER). WER measures the proportion of words that are misrecognized (deleted, inserted, or substituted) in the hypothesis relative to the reference. ICER measures the proportion of utterances with a misclassified intent. SemER combines intent and slot accuracy into a single metric, i.e., SemER =

[1] https://github.com/as-ideas/TransformerTTS

[2] https://github.com/google-research-datasets/simulated-dialogue

(slot errors + intent errors) / # (slots + intents in reference). We only show relative error rate reduction results on the IVA dataset. Take WER for example, given a method A's WER (WER_A) and a baseline B's WER (WER_B), the relative word error rate reduction (WERR) of A over B can be computed by $(\text{WER}_B - \text{WER}_A)/\text{WER}_B$; the higher the value, the greater the improvement. We denote relative errors for WER, ICER and SemER as WERR, ICERR and SemERR.

5. RESULTS

Improving E2E SLU with contexts: Table 2 shows overall model performance and the total number of parameters (in millions) of the baseline and our proposed models on the IVA dataset. We observe that contexts play a crucial role in improving E2E SLU across speech recognition and semantic interpretations. Particularly, our contextual E2E RNN-T SLU model relatively reduces 7.75% of WER, 10.96% of ICER, and 14.56% of SemER. Our contextual E2E T-T SLU model relatively reduces 13.83% of WER, 11.06% of ICER, and 10.60% of SemER. Interestingly, encoding contexts with gated attentive contextual carryover performed better than the traditional multi-head attention [27]. It gave the best performance with a relative improvement for SemER of 14.56% and 10.6% respectively across RNN-T and T-T based models.

For all subsequent discussion, we focus on SemER, as it summarizes the performance across all tasks.

		Relative Error Reduction		
Model	Config. (# params)	WERR	ICERR	SemERR
	No Context (35.12M)	Baseline	Baseline	Baseline
	w/ DA (35.31M)	5.86%	8.94%	8.06%
E2E	PrevUtt (37.38M)	7.44%	6.62%	12.23%
RNN-T	DA+ PrevUtt + AvC (37.57M)	7.38%	8.74%	12.66%
SLU	DA + PrevUtt + AttC (37.72M)	7.88%	6.92%	13.14%
	DA + PrevUtt + GAttC (37.94M)	7.75%	10.96%	14.56%
	No Context (28.58M)	Baseline	Baseline	Baseline
	w/ DA (28.61M)	5.39%	4.59%	1.48%
E2E	PrevUtt (28.78M)	12.37%	8.87%	6.32%
T-T	DA+ PrevUtt + AvC (28.80M)	11.50%	8.46%	7.85%
SLU	DA + PrevUtt + AttC (30.67M)	12.63%	9.50%	9.27%
	DA + PrevUtt + GAttC (30.89M)	13.83%	11.06%	10.60%

Table 2. Overall results on the IVA dataset. NoContext: E2E without contexts. DA and PrevUtt: dialogue act and previous utterance context. AvC: average contextual carryover. AttC: attentive contextual carryover. GAttC: AttC with gating layers.

Table 3 summarizes the results for utterances with two turns, three turns, and at least four turns. We observe that encoding contexts can lead to an average relative improvement of 40.73% and 37.09% across RNN-T and T-T E2E SLU.

		SemERR		
Model	Config.	2-turn	3-turn	4-turn +
E2E RNN-T SLU	No Context	Baseline	Baseline	Baseline
	DA + PrevUtt + GAttC	30.35%	37.80%	54.04%
E2E T-T SLU	NoContext	Baseline	Baseline	Baseline
	DA + PrevUtt + GAttC	37.07%	37.91%	36.30%

Table 3. Results on the IVA multi-turn utterances.

The effect of context ingestion: Table 4 and Table 5 show the effects of context ingestion on the E2E SLU performance. We observe that the context encoder improves E2E SLU for all scenarios, giving an average relative improvement of 13.69% and 12.07%, respectively, across RNN-T and T-T E2E SLU. Compared to the speech encoder and hidden interface ingestion, the shared context ingestion gave the biggest improvement on T-T E2E SLU with a relative improvement of 19.4%.

		Relative Error Reduction		
Model	Config	WERR	ICERR	SemERR
	No Context	Baseline	Baseline	Baseline
E2E	Speech Encoder	7.75%	10.96%	14.56%
RNN-T	ASR-NLU Interface	8.83%	8.23%	13.56%
SLU	Shared Context	9.14%	7.13%	12.95%
	No Context	Baseline	Baseline	Baseline
E2E	Speech Encoder	13.83%	11.06%	10.6%
T-T	ASR-NLU Interface	1.26%	6.89%	6.21%
SLU	Shared Context	15.16%	20.81%	19.4%

Table 4. The effect of context ingestion: IVA datasets.

		Absolute Error Rate		
Model	Config	WER	ICER	SemER
	No Context	16.02%	32.49%	40.76%
E2E	Speech Encoder	19.76%	3.62%	29.24%
RNN-T	ASR-NLU Interface	10.59%	0.36%	18.99%
SLU	Shared Context	12.14%	0.25%	18.55%
	No Context	13.06%	30.4%	36.83%
E2E	Speech Encoder	14.1%	2.38%	26.56%
T-T	ASR-NLU Interface	12.81%	0.25%	18.49%
SLU	Shared Context	13.68%	0.21%	18.62%

Table 5. The effect of context ingestion: Syn-Multi datasets.

We also qualitatively examined the effect of contexts. Contextual models recognized *cancel* correctly with the *Select(Time)* dialogue act context, whereas non-context model recognized the word as *cascal*. Further, contextual models can better handle ambiguous utterances. For example, contextual models correctly predict utterance *next Monday for inferno* as *BuyMovieTickets* intent as its previous utterance is *i want to buy movie tickets*, whereas non-context models confuse this utterance with *ReserveRestaurant* intent.

6. CONCLUSION

We propose a novel E2E SLU approach where a multi-head gated attention mechanism is introduced to effectively incorporate the dialogue history from the spoken audio. Our proposed approach significantly improves E2E SLU accuracy on the internal industrial voice assistant and publicly available datasets compared to the non-contextual E2E SLU models. In the future, we will apply our proposed approach on other datasets and further improve our contextual model architecture.

7. REFERENCES

[1] Loren Lugosch, Mirco Ravanelli, Patrick Ignoto, Vikrant Singh Tomar, and Yoshua Bengio, "Speech model pre-training for end-to-end spoken language understanding," in *Interspeech 2019*, 2019, pp. 814–818.

[2] Swapnil Bhosale, Imran Sheikh, Sri Harsha Dumpala, and Sunil Kumar Kopparapu, "End-to-end spoken language understanding: Bootstrapping in low resource scenarios.," in *Interspeech*, 2019, pp. 1188–1192.

[3] Martin Radfar, Athanasios Mouchtaris, and Siegfried Kunzmann, "End-to-end neural transformer based spoken language understanding," in *Interspeech 2020*, 2020, pp. 866–870.

[4] Loren Lugosch, Brett H Meyer, Derek Nowrouzezahrai, and Mirco Ravanelli, "Using speech synthesis to train end-to-end spoken language understanding models," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 8499–8503.

[5] Subendhu Rongali, Beiye Liu, Liwei Cai, Konstantine Arkoudas, Chengwei Su, and Wael Hamza, "Exploring transfer learning for end-to-end spoken language understanding," *AAAI*, 2021.

[6] Milind Rao, Anirudh Raju, Pranav Dheram, Bach Bui, and Ariya Rastrow, "Speech to semantics: Improve asr and nlu jointly via all-neural interfaces.," in *Interspeech 2020*, 2020, pp. 876–880.

[7] Suyoun Kim and Florian Metze, "Dialog-context aware end-to-end speech recognition," in *2018 IEEE Spoken Language Technology Workshop (SLT)*, 2018, pp. 434–440.

[8] Suyoun Kim, Siddharth Dalmia, and Florian Metze, "Gated embeddings in end-to-end speech recognition for conversational-context fusion.," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 1131–1141.

[9] Suyoun Kim, Siddharth Dalmia, and Florian Metze, "Cross-attention end-to-end asr for two-party conversations.," in *Interspeech 2019*, 2019, pp. 4380–4384.

[10] Suyoun Kim, "End-to-end speech recognition on conversations," 2020.

[11] Arshit Gupta, Peng Zhang, Garima Lalwani, and Mona T. Diab, "Casa-nlu: Context-aware self-attentive natural language understanding for task-oriented chatbots," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 1285–1290.

[12] Anirudh Raju, Behnam Hedayatnia, Linda Liu, Ankur Gandhe, Chandra Khatri, Angeliki Metallinou, Anushree Venkatesh, and Ariya Rastrow, "Contextual language model adaptation for conversational agents," in *Interspeech 2018*, 2018, pp. 3333–3337.

[13] Golan Pundak, Tara N Sainath, Rohit Prabhavalkar, Anjuli Kannan, and Ding Zhao, "Deep context: end-to-end contextual speech recognition," in *2018 IEEE spoken language technology workshop (SLT)*. IEEE, 2018, pp. 418–425.

[14] Zelin Wu, Bo Li, Yu Zhang, Petar S Aleksic, and Tara N Sainath, "Multistate encoding with end-to-end speech rnn transducer network," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7819–7823.

[15] Swayambhu Nath Ray, Minhua Wu, Anirudh Raju, Pegah Ghahremani, Raghavendra Bilgi, Milind Rao, Harish Arsikere, Ariya Rastrow, Andreas Stolcke, and Jasha Droppo, "Listen with intent: Improving speech recognition with audio-to-intent front-end.," *Interspeech*, 2021.

[16] Libo Qin, Wanxiang Che, Minheng Ni, Yangming Li, and Ting Liu, "Knowing where to leverage: Context-aware graph convolutional network with an adaptive fusion layer for contextual spoken language understanding," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 1280–1289, 2021.

[17] Shang-Yu Su, Pei-Chieh Yuan, and Yun-Nung Chen, "Dynamically context-sensitive time-decay attention for dialogue modeling," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 7200–7204.

[18] Waheed Ahmed Abro, Guilin Qi, Huan Gao, Muhammad Asif Khan, and Zafar Ali, "Multi-turn intent determination for goal-oriented dialogue systems," in *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–8.

[19] Qian Chen, Zhu Zhuo, Wen Wang, and Qiuyun Xu, "Transfer learning for context-aware spoken language understanding," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 779–786.

[20] Raghav Gupta, Abhinav Rastogi, and Dilek Hakkani-Tür, "An efficient approach to encoding context for spoken language understanding," in *Interspeech 2018*, 2018, pp. 3469–3473.

[21] Yun-Nung Chen, Dilek Hakkani-Tür, Gökhan Tür, Jianfeng Gao, and Li Deng, "End-to-end memory networks with knowledge carryover for multi-turn spoken language understanding.," in *Interspeech*, 2016, pp. 3245–3249.

[22] Yufan Wang, Tingting He, Rui Fan, Wenji Zhou, and Xinhui Tu, "Effective utilization of external knowledge and history context in multi-turn spoken language understanding model," in *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 2019, pp. 960–967.

[23] Milind Rao, Pranav Dheram, Gautam Tiwari, Anirudh Raju, Jasha Droppo, Ariya Rastrow, and Andreas 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)*, 2021.

[24] Alex Graves, "Sequence transduction with recurrent neural networks," *CoRR, vol. abs/1211.3711, 2*, 2012.

[25] 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.

[26] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina N. 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)*, 2018, pp. 4171–4186.

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, vol. 30, pp. 5998–6008.

[28] Lanqing Xue, Xiaopeng Li, and Nevin L Zhang, "Not all attention is needed: Gated attention network for sequence data," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, vol. 34, pp. 6550–6557.

[29] Suyoun Kim, *End-to-End Speech Recognition on Conversations*, Ph.D. thesis, Carnegie Mellon University, 2019.

[30] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[31] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[32] Suyoun Kim and Michael L Seltzer, "Towards language-universal end-to-end speech recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4914–4918.

[33] Jamie Kiros, William Chan, and Geoffrey Hinton, "Illustrative language understanding: Large-scale visual grounding with image search," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 922–933.

[34] Pararth Shah, Dilek Hakkani-Tur, Bing Liu, and Gokhan Tur, "Bootstrapping a neural conversational agent with dialogue self-play, crowdsourcing and on-line reinforcement learning," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, 2018, vol. 3, pp. 41–51.

[35] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin Dogus Cubuk, and Quoc V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition.," in *Interspeech 2019*, 2019, pp. 2613–2617.

[36] Taku 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, vol. 1, pp. 66–75.

[37] Yanzhang He, Tara N Sainath, Rohit Prabhavalkar, Ian McGraw, Raziel Alvarez, Ding Zhao, David Rybach, Anjuli Kannan, Yonghui Wu, Ruoming Pang, et al., "Streaming end-to-end speech recognition for mobile devices," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6381–6385.

[38] Qian Zhang, Han Lu, Hasim Sak, Anshuman Tripathi, Erik McDermott, Stephen Koo, and Shankar 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.

[39] 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.

[40] Diederik P. Kingma and Jimmy Lei Ba, "Adam: A method for stochastic optimization," in *ICLR 2015 : International Conference on Learning Representations 2015*, 2015.