Adaptive Global-Local Context Fusion for Multi-Turn Spoken Language Understanding

Thanh Tran*, Kai Wei†, Weitong Ruan, Ross McGowan, Nathan Susanj, Grant P. Strimel

Amazon Alexa
{tdt, kaiwe, weitong, rosmcgow, nsusanj, gsstrime}@amazon.com

Abstract
Recent years have seen significant advances in multi-turn Spoken Language Understanding (SLU), where dialogue contexts are used to guide intent classification and slot filling. However, how to selectively incorporate dialogue contexts, such as previous utterances and dialogue acts, into multi-turn SLU still remains a substantial challenge. In this work, we propose a novel contextual SLU model for multi-turn intent classification and slot filling tasks. We introduce an adaptive global-local context fusion mechanism to selectively integrate dialogue contexts into our model. The local context fusion aligns each dialogue context using multi-head attention, while the global context fusion measures overall context contribution to intent classification and slot filling tasks. Experiments show that on two benchmark datasets, our model achieves absolute F1 score improvements of 2.73% and 2.57% for the slot filling task on Sim-R and Sim-M datasets, respectively. Additional experiments on a large-scale, de-identified, in-house dataset further verify the measurable accuracy gains of our proposed model.

Introduction
The last few years have seen an increasing application of Spoken Language Understanding (SLU) systems, such as Google Assistant, Amazon Alexa, etc. One of the fundamental tasks of these systems is to map the meaning of spoken utterances expressed in natural language to machine comprehensible language (Allen 1995; Tur and De Mori 2011). For example, the machine learns to map find a restaurant in Richmond to an intent for finding restaurants (intent classification) and to slots such as Richmond: Location (slot filling).

One important topic in the SLU research is effectively interpreting a user’s intents in multi-turn dialogues, where the user and the system have multiple turns of back-and-forth conversations to achieve the user’s goal. Historically, this line of work has focused on using traditional machine learning methods (Miller et al. 1996; Bhargava et al. 2013). For example, Bhargava et. al. (2013) found that using previous utterances as contexts in an SVM-HMM SLU system could help resolve ambiguities. Recently, deep learning approaches have become increasingly popular to incorporate contextual information (Qin et al. 2021; Su, Yuan, and Chen 2019; Abro et al. 2019; Chen et al. 2019; Su, Yuan, and Chen 2018; Gupta, Rastogi, and Hakkani-Tur 2018; Chen et al. 2016; Wei et al. 2021). Chen et. al. (2016) proposed to use end-to-end memory networks to model previous utterance transcripts in multi-turn dialogues. Gupta et al. (2018) proposed an efficient method to encode dialogue acts with a feedforward network from prior dialogue history with limited degradation in accuracy compared to the end-to-end memory network approach (Gupta, Rastogi, and Hakkani-Tur 2018). Gupta et al. (2019) fuses signals like previous intents via a self-attention mechanism with a variable context window. Wang et. al. (2019) encodes historical utterances using the Bidirectional Long Short Term Memory (Bi-LSTM) networks and ConceptNet to encode external knowledge, and construct knowledge attention over these contexts. Qin et. al. (2021) proposed a context-aware graph convolutional network for contextual SLU. Yet, how to selectively incorporate both dialogue acts and previous utterance context to multi-turn intent detection and slot filling still remains under-explored.

In this paper, we propose a contextual SLU model for intent classification and slot filling in multi-turn dialogues, where both dialogue acts and previous utterance contexts are exploited. To selectively incorporate dialogue contexts into the model, we propose an adaptive global-local context fusion mechanism. The local context fusion aligns each contextual source information with the utterance transcript signals using the multi-head attention (Vaswani et al. 2017), while the global context measures contribution of all contextual information. The closest work to ours are (Gupta, Rastogi, and Hakkani-Tur 2018) and (Qin et al. 2021). However, these works use BiLSTM to encode previous utterances, whereas our work uses BERT to enrich their contextually semantic representations. Moreover, Qin et. al. (2021) focuses on graph-based methods to filter out irrelevant information only for slot filling, whereas our work uses the global-local multi-head attention for both slot filling and intent detection. Our model achieves the SOTA results on intent classification and outperforms previous methods for slot filling by a large margin on two benchmark datasets. We further experiment with our model on a large scale, in-house, de-identified dataset. In addition, we study the effects of contexts by conducting ablation studies and visualizing the
Problem Statement

Our contextual SLU model takes a current utterance \( u_t \), and a list of previous dialogue acts \( D^*=\{(a_1, s_1), ..., (a_{|D|}, s_{|D|})\} \), and previous utterance transcripts \( U^t=\{u_1, u_2, ..., u_{t-1}\} \). Each \( (a_i, s_i) \) pair indicates a dialogue action \( a_i \) and a dialogue slot \( s_i \). Given the ground truth intent \( y^{tgt}_t \) of \( u_t \) and ground truth slot \( y^{slot}_t \) per each word token \( p_{t,i} \in u_t \), our contextual SLU model aims to maximize the intent probability \( P(y^{tgt}_t | u_t, U^t, D^t) \) for \( u_t \), and the slot probability \( P(y^{slot}_t | u_t, U^t, D^t) \) for each \( p_{t,i} \).

Proposed Model

Figure 1 shows our proposed contextual SLU model architecture. At a high level, we input (i) the wordpiece embeddings of current utterance transcripts and (ii) the context encoder of dialogue acts and previous utterance transcripts into our adaptive global-local context fusion mechanism.

The local context fusion aligns each contextual source information with the utterance transcript signals using the multi-head attention (Vaswani et al. 2017), while the global context measures the contribution of all contextual information. The local context fusion considers each context encoding type as a separated key and value and the wordpiece embeddings as the query. Then, it assigns attention scores to the context encoding. Intuitively, the global attention serves as a gating layer to produce how much all contexts can contribute to the input query. Without the global attention, the local attention always give an accumulated attention score of 1.0, as a result of performing a softmax function. This is not optimal. In many cases, the contexts contribute insignificantly to the SLU tasks. For example, a user asks a voice assistant system to call uncle sam, and the system confirms back to see if the user wants to call a nearby uncle sam’s sandwich bar. The user then says in a second turn call my uncle who’s first name is sam. In this case, dialogue contexts coming from the first turn are not helpful for the interpretation of the second turn. Therefore, we propose this global-local fusion mechanism to allow the model to selectively pay attention to previous dialogue contexts in multi-turn dialogues.

After selectively fusing contextual information with the wordpiece embeddings, we use a BiLSTM encoder to learn context-aware embeddings, the output of which are used for the intent classification and slot filling tasks, simultaneously. We detail our architecture as below.

**Embedding Layer**

We pre-train a SentencePiece (SP)\(^1\) model on the training data with 4,500 wordpieces to avoid the explosion of vocabulary size and the out-of-vocabulary problems when using a word-level representation. We denote \( E \in \mathbb{R}^{4500 \times d} \) as an embedding layer with \( d \) as the embedding size. Given the input utterance transcript \( u_t \), we use the pretrained SP tokenizer to tokenize the input transcript and project the tokenized wordpieces into \( E \) to obtain the wordpiece embeddings \( P^t = \{p_1, p_2, ..., p_n\} \), where \( n \) is the number of tokenized wordpieces.

**Context Encoder**

Figure 1 shows our methods for encoding dialogue acts (in light orange color) and previous utterance transcripts (in dark orange color). We describe the details of our context encoder methods below:

**Encoding Dialogue Acts:** Its input contains a list of dialogue action and slot pairs \( D^*=\{(a_1, s_1), ..., (a_{|D|}, s_{|D|})\} \). Given that \( \ell_D \) is the maximum number of dialogue action-slot pairs in all training input data instances, if \( |D^t| < \ell_D \), we pad \( D^t \) with default action-slot pairs until reaching \( \ell_D \). During inference, if an utterance in the test set has more than \( \ell_D \) action-slot pairs, we only take the latest \( \ell_D \) pairs. We use zero-embeddings for the padding actions and slots so that they have no effect on our model.

**Embedding Layer:** This layer maintains two embedding matrices: a dialogue action embedding matrix \( \mathcal{A} \in \mathbb{R}^{|A| \times d} \) and a dialogue slot embedding matrix \( \mathcal{S} \in \mathbb{R}^{|S| \times d} \), where \( |A| \) and \( |S| \) refer to the number of dialogue actions and slots in the model, respectively. By projecting each action \( a_i \) and slot \( s_i \) in the action-slot pair \( (a_i, s_i) \in D^t \) via \( \mathcal{A} \) and \( \mathcal{S} \), we obtain their corresponding embeddings \( a_i \) and \( s_i \).

**Processing Layer:** With each \( (a_i, s_i) \in D^t \), we output action embedding \( a_i \) and slot embedding \( s_i \) from the embedding layer. We then perform an element-wise addition to fuse \( a_i \) and \( s_i \). Next, we transform the fused embedding of \( a_i \) and \( s_i \) by a linear transformation with a ReLu activation to obtain \( g_i \) as follows:

\[
g_i = ReLU(W_a(a_i + s_i) + b_g) \quad \text{ (1)}
\]

For all \( |D^t| \) action-slot pairs in \( D^t \), we obtain the corresponding fused embeddings \( \{g_1, g_2, ..., g_{|D^t|}\} \) by following the same process that produces \( g_i \) in Eq.(1). To obtain the output embeddings, we perform a row-wise concatenation across the fused embeddings.

\[
G^t = g_1 \oplus g_2 \oplus ... \oplus g_{|D^t|} \quad \text{ (2)}
\]

**Encoding Previous Utterance Transcripts:** Its input is a list of previous utterance transcripts \( U^t=\{u_1, u_2, ..., u_{t-1}\} \). To learn the contextual embeddings of an utterance transcript \( u_j \in U^t \), we use the pre-trained uncased BERT-based language model (Devlin et al. 2019). Specifically, we first tokenize each \( u_j \) with the BERT-based tokenizer. Next, we prepend a [CLS] token and append a [SEP] token to the tokenized transcript. Since utterances at different turns have a different number of previous utterance transcripts, we use \( \ell_U \) as the maximum number of turns in all the training examples. At turn \( t \)-th \((t < \ell_U)\), we pad \( \ell_U - t \) empty transcripts to obtain a length of \( \ell_U \). During inference, if an utterance has more than \( \ell_U \) turns, we only take its latest \( \ell_U \) previous utterance transcripts.

**Processing Layer:** We input each \( u_j \in U^t \) into the pre-trained BERT-based model and extract the embeddings from

\[^{1}\text{https://github.com/google/sentencepiece}\]
Figure 1: Architecture of our proposed contextual SLU model.

Figure 2: Adaptive attention via global-local context fusion.

The [CLS] token as the summarized embeddings for u_j. For all previous utterance transcripts in Ut, we obtain the corresponding output embeddings \{u_1, u_2, ..., u_Ut\}. We mask the padded empty transcripts as zero embeddings so that they have no effect on our model performance.

**Output:** We perform a row-wise concatenation for all previous utterance transcripts’ embeddings \{u_1, u_2, ..., u_Ut\} as follows:

\[ U^t = u_1 \oplus u_2 \oplus ... \oplus u_Ut \]  

(3)

**Adaptive Global-Local Context Fusion**

Figure 2 shows our proposed adaptive global-local context fusion mechanism. We use the multi-head attention to compute local attention scores and design a global attention mechanism to measure the contribution of all contexts. Then, we fuse the global and local attention scores into one. Details of this architecture are described below:

**Global-Local Multi-Head Attention Layer:** Recall G^t as the query, we apply the scaled dot attention (Vaswani et al. 2017) to measure the local attention scores \( \alpha_G \) between G^t and P^t, and the local attention scores \( \alpha_U \) between U^t and P^t as follows:

\[ \alpha_G = \text{softmax} \left( \frac{Q_G K^T_G}{\sqrt{d}} \right) \quad \alpha_U = \text{softmax} \left( \frac{Q_U K^T_U}{\sqrt{d}} \right) \]  

(4)

where \( Q_G, K_G \) and \( V_G \) are learned by linearly transforming the corresponding P^t and G^t. \( Q_U, K_U \) and \( V_U \) are learned by linearly transforming the corresponding P^t and U^t.

\[ Q_G = W^{(g)}_G P^t + b^{(g)}_G \quad K_G = W^{(k)}_G G^t + b^{(k)}_G \quad V_G = W^{(v)}_G G^t + b^{(v)}_G \quad Q_U = W^{(u)}_U P^t + b^{(u)}_U \quad K_U = W^{(k)}_U U^t + b^{(k)}_U \quad V_U = W^{(v)}_U U^t + b^{(v)}_U \]  

(5)

To measure global attention scores, we first perform a column-wise concatenation between G^t and U^t, resulting in a long context vector C^t \( \in \mathbb{R}^{1 \times (t_d \times d + t_U \times 768)} \) (d is the dialogue act embedding size and 768 is the BERT-based embedding size). Then, we measure the global attention scores as following:

\[ \beta = \text{sigmoid} \left( Q_\beta K^T_\beta \right) \]  

(6)

where \( Q_\beta, K_\beta \) are learned by linearly transforming P^t and C^t as follows:

\[ Q_\beta = W^{(g)}_\beta P^t + b^{(g)}_\beta \quad K_\beta = W^{(k)}_\beta C^t + b^{(k)}_\beta \]

Note that \( \beta \) is an n x 1 matrix, where each entry \( \beta_i \in \beta \) shows how much all the contextual information contributes to each subquery \( p_i \in P^t \). Thus, we replicate \( \beta \) to have a similar dimension size with \( \alpha_G \) and \( \alpha_U \), resulting in \( \beta_G \) and \( \beta_U \) respectively. Then, we perform an element-wise product between \( \alpha_G \) and \( \beta_G \), as well as \( \alpha_U \) versus \( \beta_U \):

\[ \gamma_G = \alpha_G \odot \beta_G \quad \gamma_U = \alpha_U \odot \beta_U \]

Lastly, we perform matrix multiplication between \( V_G \) and \( \gamma_G \) to obtain adaptive dialogue act embeddings C_{G,att} \( \in \)
\( R^{n \times d} \), and between \( V_U \) and \( \gamma_U \) to obtain adaptive previous utterance transcript embeddings \( C_U^t, att \) \( \in R^{n \times 768} \). Finally, we column-wise concatenate \( C^{t, att}_G \) and \( C^{t, att}_U \) with word-piece embeddings \( P^t \):
\[
\begin{align*}
C^{t, att}_G & = \gamma_U V_U; \quad C^{t, att}_U = \gamma_U V_U \\
P^t_{context} & = [P^t, \ C^{t, att}_G, C^{t, att}_U]
\end{align*}
\]

**Processing Layer:** With \( P^t_{context} \) established, we pass \( P^t_{context} \) through a \( m \)-layer Bi-LSTM encoder to produce a series of context-aware hidden states \( H^{(m)}_{slot} = \{h^{(slot)}_1, h^{(slot)}_2, ..., h^{(slot)}_n\} \) and a summarized bidirectional embedding vector \( h^{(int)}_i \). Here, we use the BiLSTM encoder to have a fair comparison against previous works such as (Qin et al. 2021; Gupta, Rastogi, and Hakkani-Tur 2018). In addition, it has also been shown that the BiLSTM encoder outperforms the transformer-based models on the public benchmark datasets used in this study (Qin et al. 2021). Of note, our adaptive global-local context fusion design can also be integrated with the transformer-based models.

\[
\begin{align*}
\bar{h}^{(k)}_{i} &= LSTM(h^{(k-1)}_{i}, \bar{h}_{i-1}) \\
\hat{h}^{(k)}_{i} &= LSTM(h^{(k-1)}_{i}, \hat{h}_{i+1}) \\
\text{with } i \in [1, n], k \in [1, m], \text{ } h^{(0)}_{i} &= P^t_{context} \\
h^{(slot)}_{i} &= [\bar{h}^{(m)}_{i}, \hat{h}^{(m)}_{i}], \text{ } h^{(int)}_{i} &= [\bar{h}^{(m)}_{n}, \hat{h}^{(m)}_{1}]
\end{align*}
\]

**Intent Classification and Slot Filling**

**Intent Classification:** It is a multi-class classification problem. We use \( h^{(int)}_i \) in Eq. (8) to produce an intent distribution over all \( |I| \) intents at each input utterance \( u_i \). We define the cross entropy loss for \( u_i \) as follows:
\[
\begin{align*}
\hat{y}^{(int)}_{i, j} &= \text{softmax}(W^{(int)} h^{(int)}_i + b^{(int)}) \\
L_{int} &= \sum_{j=1}^{|I|} y^{(int)}_{i,j} \log(\hat{y}^{(int)}_{i,j})
\end{align*}
\]

**Slot Filling:** Similar to the intent classification, we use \( H^{(slot)}_{slot} \) for the slot filling task for \( u_i \) with \( |S| \) slots over each of \( n \) tokens at each input utterance using the following cross entropy loss:
\[
\begin{align*}
\hat{y}^{(slot)}_{i, j, k} &= \text{softmax}(W^{(slot)} h^{(slot)}_i + b^{(slot)}) \\
L_{slot} &= -\sum_{i=1}^n \sum_{k=1}^{|S|} y^{(slot)}_{i, j, k} \log(\hat{y}^{(slot)}_{i, j, k})
\end{align*}
\]

**Multi-Task Learning:** We use a multi-task learning strategy to train our model. The joint cost function is defined as:
\[
L = L_{int} + \lambda L_{slot}
\]

where \( \lambda \) are hyper-parameters to control loss contribution.

**Experimental Settings**

**Dataset:** We conduct experiments on the benchmark Simulated Dialogue dataset (Sim) (Liu et al. 2018; Shah et al. 2018), which consists of two datasets: (i) Simulated Restaurant Dialogue (Sim-R) and (ii) Simulated Movie Dialogue (Sim-M). Sim-R contains dialogues for booking a restaurant table, whereas Sim-M contains dialogues for buying a movie ticket. Specifically, Sim-R has 11k turns in 1,116 training dialogues, 349 development dialogues, and 775 testing dialogues. Sim-M has nearly 4k turns in 384 training dialogues, 120 development dialogues, and 264 testing dialogues. In total, the Sim dataset has 3 intents, 12 slot types, and 21 user dialogue act types. A key challenge of this dataset is the presence of unseen entities in testing sets. For example, only 13% of movie names in the validation and test sets are in the training set.

In addition, we conduct experiments on our large-scale, de-identified, multi-domain in-house dataset. This dataset has both single turn utterances and multi-turn utterances.

**Baseline Models:** We compare our proposed model with the following baseline models:

- **NoContext:** It is a two-layer stacked Bidirectional RNN using GRU and LSTM cells respectively, where no context information is incorporated. We report its best architecture’s results.
- **PrevTurn:** It is similar to the NoContext baseline, but encodes only the utterances in previous turns.
- **MemNet (Chen et al. 2016):** An end-to-end memory network that dynamically exploits the contextual knowledge.
- **SDEN (Bapna et al. 2017):** It uses a sequential dialogue encoder to encode contexts from the dialogue history in chronological order with recurrent neural networks.
- **EfficientNet (Gupta, Rastogi, and Hakkani-Tur 2018):** It is a hierarchical recurrent neural network that efficiently encodes dialogue act context.
- **GraphNet (Qin et al. 2021):** It uses a Graph Convolutional Network for integrating dialogue act contexts.

Note that we do not compare with BERT because Qin at. al., (2021) showed that GraphNet outperformed BERT on the Sim datasets.

**Implementation Details:** Our experiments are implemented in Tensorflow 2.3 (Abadi et al. 2016). The hyper-parameters are selected based on the best performance on the validation set. During training, we minimize the sum of intent and slot losses using Adam optimizer (Kingma and Ba 2015) for 100 training steps with a batch size of 32. We use two BiLSTM layers with each having a wordpiece embedding size of 256. The decoders for slot filling and intent classification both are two-layer dense networks with 256 and 512 units, respectively. For the context encoder, the dialogue act embedding size is set to 256; and the embedding size for BERT encoding of previous utterances is set to 768. The attention size for global-local multi-head attention is set to 256 with 1 head. We set \( \lambda = 1 \) in Eq. (11) to give equal contribution for \( L_{int} \) and \( L_{slot} \) losses.

**Evaluation Metrics:** To benchmark against SOTA approaches, we report the performance of the intent classification task using intent accuracy and the performance of
Table 1: Overall Performance on Sim-R and Sim-M datasets.

<table>
<thead>
<tr>
<th>Model</th>
<th>Sim-R Results</th>
<th>Sim-M Results</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Intent Acc.</td>
<td>Slot F1</td>
</tr>
<tr>
<td>NoContext</td>
<td>83.61%</td>
<td>94.24%</td>
</tr>
<tr>
<td>PrevTurn</td>
<td>99.37%</td>
<td>94.96%</td>
</tr>
<tr>
<td>MemNet-6</td>
<td>99.75%</td>
<td>94.42%</td>
</tr>
<tr>
<td>MemNet-20</td>
<td>99.67%</td>
<td>94.28%</td>
</tr>
<tr>
<td>SDEN-20</td>
<td>99.84%</td>
<td>94.81%</td>
</tr>
<tr>
<td>EfficientNet</td>
<td>99.65%</td>
<td>94.70%</td>
</tr>
<tr>
<td>GraphNet</td>
<td>99.97%</td>
<td>95.37%</td>
</tr>
<tr>
<td>Our model</td>
<td><strong>99.97%</strong></td>
<td><strong>98.10%</strong></td>
</tr>
</tbody>
</table>

Table 2: Effectiveness of our global-local context fusion. w/o GL-CF means that we remove the global and local context fusion from the proposed model; w/o G-CF means that we remove the global context fusion from the proposed model.

Table 2: Effectiveness of different contexts. w/o DialogAct means that we remove the dialogue act encoder from the proposed model. w/o PrevUtt means that we remove the previous utterance encoder from the proposed model.

The slot filling task using the slot chunk F1 score (Tjong Kim Sang and Buchholz 2000) for the Sim-R and Sim-M dataset. For the in-house dataset, we report model performance on intent classification error rate (ICER), and semantic error rate (SemER). ICER measures the proportion of utterances with a misclassified intent, i.e. \( ICER = 1.0 - \text{intent accuracy} \). IRER and SemER measures the utterance-level error rate that considers both intent and slot errors. SemER (Makhoul et al. 1999) combines intent and slot accuracy into a single metric, i.e. \( \text{SemER} = \frac{\text{# (slot errors + intent errors)}}{\text{# (slots + intents in reference)}} \). For the in-house dataset, we report relative improvements with respect to the baseline.

## Results

### Overall Performance

Table 1 shows the performance of our proposed model and SOTA baselines for intent accuracy and slot F1 on Sim-R and Sim-M datasets. Overall, our proposed model achieves the best performance for intent accuracy and slot F1 on the two benchmarking datasets. On the Sim-R dataset, our model achieves 99.97% intent accuracy and 98.10% slot F1. Compared to the previously best performing model, our model achieves the same intent accuracy and an absolute slot F1 improvement of 2.73% (\( p\text{-value} < 0.001 \)) under the non-directional Mann-Whitney U test). On Sim-M dataset, our model achieves 100% intent accuracy and 98.10% slot F1. Compared to the previously best performed model, with an absolute intent accuracy improvement of 0.3% in the absolute intent accuracy and 0.3% in the absolute slot F1. To further verify the effectiveness of the global attention, we plot the histogram of global attentive scores in Sim-R and Sim-M datasets. Figure 3 shows global attention scores are mostly distributed in ranges of [0.9, 1.0] and [0.0, 0.1]. This suggests that our global-local fusion can reduce the context contributions when all global attention scores are close to [0.9, 1.0] and [0.0, 0.1]. The global attention scores (blue bars) are mostly distributed in ranges of [0.9, 1.0] and [0.0, 0.1].

Table 3: Effectiveness of different contexts. w/o DialogAct means that we remove the dialogue act encoder from the proposed model. w/o PrevUtt means that we remove the previous utterance encoder from the proposed model.

Next, we present ablation studies to understand the effectiveness of our global-local context fusion design and different contexts. As obtaining a high intent accuracy is trivial on Sim-R and Sim-M datasets, we report only Slot F1 results.

### Effectiveness of Global-Local Fusion

Table 2 shows that removing global-local fusion has a negative impact on our model performance, with an average drop of 0.08% in the absolute intent accuracy and 0.3% in the absolute slot F1. To further verify the effectiveness of the global attention, we plot the histogram of global attentive scores in Sim-R and Sim-M datasets. Figure 3 shows global attention scores are mostly distributed in ranges of [0.9, 1.0] and [0.0, 0.1]. This suggests that our global-local fusion can reduce the context contributions when all global attention scores are close to [0.0, 0.1], which is an enhancement over the traditional multi-head attention approach (Vaswani et al. 2017).

### Effectiveness of Different Contexts

Table 3 shows the ablation study results when removing different contexts and attention mechanisms. A first observation is that removing dialogue acts leads to the lowest performance on the Sim-M dataset, whereas removing previous utterances leads to the lowest performance on the Sim-R dataset. This indicates that dialogue history contexts play a crucial role in improving SLU task in the multi-turn dialogue setting.
We propose a novel E2E SLU model designed for multi-turn dialogues where dialogue acts and previous utterance transcripts are utilized as contexts to improve the performance of intent prediction and slot filling tasks. We introduce a global-local multi-head attention mechanism to effectively incorporate contextual signals into our model. We demonstrate that our proposed approach improves intent accuracy and slot F1 – two well known SLU metrics over six state-of-the-art baselines on two publicly available datasets. Extensive experiments on an in-house dataset further verify the effectiveness of our proposed model.

**Conclusion**
References


