Toward Scalable Neural Dialogue State Tracking Model

Elnaz Nouri
Microsoft Research and AI
elnouri@microsoft.com

Ehsan Hosseini-Asl
Salesforce Research
ehosseiniasl@salesforce.com

Abstract

The latency in the current neural based dialogue state tracking models prohibits them from being used efficiently for deployment in production systems, albeit their highly accurate performance. This paper proposes a new scalable and accurate neural dialogue state tracking model, based on the recently proposed Global-Local Self-Attention encoder (GLAD) model by Zhong et al. (2018). By using only one recurrent networks with global conditioning, compared to (1 + # slots) recurrent networks with global and local conditioning used in the GLAD model, our proposed model reduces the latency in training and inference times by 35% on average, while preserving performance of belief state tracking, by 97.38% on turn request and 88.51% on joint goal and accuracy. Evaluation on Multi-domain dataset (Multi-WoZ) also demonstrates that our model outperforms GLAD on turn inform and joint goal accuracy.

1 Introduction

Dialog State Tracking (DST) is an important component of task-oriented dialogue systems. DST keeps track of the interaction goal and what has happened in the dialog history. Majority of the deployed dialogue systems in commercial settings use dialogue state defined based on using user utterance at each turn, context from previous turns, and other external information as well as the output of the system at each turn. These include customer supports and intelligent assistants, such as Amazon Alexa, Apple Siri and Google Assistant. Decision made by the dialogue state tracker, is then used to determine what action should be taken by the system in next steps. Therefore, Dialogue state tracking (DST) is a key component of a task oriented dialogue systems.

DST estimates user's goal at each dialogue turn given the interaction so far. State of the art approaches for dialogue state tracking rely on deep learning models, which represent the dialogue state as a distribution over all candidate slot values that are defined in the ontology. Recently, several neural-based DST systems have been proposed. Mrksic et al. (2017) proposed a Neural Belief Tracker (NBT) model based on binary decision making of each state-values, where representation of user utterance, system action, and candidate pairs are computed based on deep distribiutional representation of word vectors. In their model, they used deep network (DNN) and convolutional network (CNN) to compute such representation vectors. Wen et al. (2017) proposed a sequence-to-sequence model for estimating the next dialogue state. In their work, the encoded hidden vector of user utterance is used to determine the current dialogue state, followed by a policy network to query over knowledge databse. Then, the retrieved information is used as a conditionining input to the decoder, to generate the system response.

Recently, Zhong et al. (2018) proposed a model based on training a binary classifier for each slot-value, Global-Locally Self Attentive encoder (GLAD, by employing recurrent and self attention for each utterance and previous system actions, and measuring similaity of these computed representation to

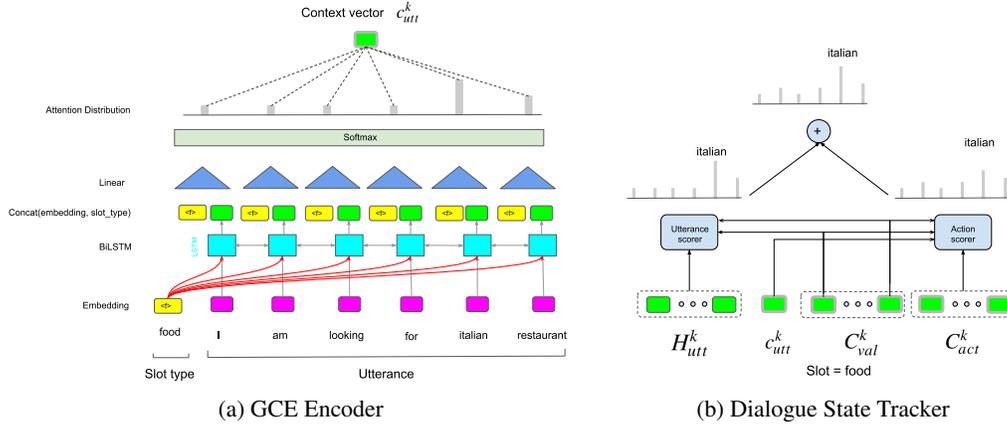| (a) GCE Encoder | (b) Dialogue State Tracker |

Figure 1: Proposed Dialogue State Tracking model with (a) Globally-Conditioned Encoder (GCE), and (b) overall state tracking model.

each slot-value, which achieve state of the art results on WoZ (Wen et al., 2017) and DSTC2 (Williams et al., 2013) datasets.

Although the proposed neural based models achieves state of the art results on several benchmark, they are still inefficient for deployment in production system, due to their latency which stems from using recurrent networks. In this paper, we propose a new encoder, by improving GLAD architecture (Zhong et al., 2018). The proposed encoder is based on removing slot-dependent recurrent network for utterance and system action encoder, and employing a global conditioning of aforementioned encoder on the slot type embedding vector. By removing the slot-dependent recurrent network, the proposed model is able to preserve the performance in predicting correct belief state, while improving computational complexity. The detailed description of encoder is explained in the section 2.

1.1 Related Works

A similar scalable dialogue state tracking model is also proposed by Rastogi et al. (2017), which is based on conditioning the encoder input. They used a similar conditioning of user utterance representation on slot values (candidate sets) and slot type. However, our proposed model is based on conditioning only on slot type. Therefor, our proposed model is simpler since it contains only one conditioned encoder for user utterance, whereas Rastogi et al. (2017) model requires two independet conditioned encoder.

Recently, Xu and Hu (2018) proposed a model for unknown slot type by using a pointer network, based on conditioning to slot type embedding. Our proposed model is also relaxing the current GLAD architecture for unknown slot types during inference.

2 Proposed model

In this section, we describe the proposed model. First, section 2.1 explains the recently proposed GLAD encoder (Zhong et al., 2018) architecture, followed by our proposed encoder in section 2.2.

2.1 Global-Locally Self-Attentive Model

GLAD model is based on learning multiple binary classifier for each slot-value pair. In this architecture, separate encoders are considered for utterance, previous system action, and all slot values. The output of these encoders are then used by two scores model, i.e. previous system action and utterance, to predict the probability distribution on slot-value pairs. This means, each scores model compute the similarity of each slot-value to the utterance representation or previous system action. All encoders used similar architecture, i.e. global-local self attention (GLAD). To compute the hidden representation of its input sequence and its summary (context), GLAD a combination of bidirectional

LSTM (Hochreiter and Schmidhuber, 1997), to compute the temporal representation, followed by self-attention layer to extract the context vector. To incorporate information regarding each slot, there is a dedicated recurrent and self-attention network for each slot. Therefor, to estimate the probability distribution over values of each slot, GLAD encoder has to learn a different hidden and context vector for utterance and previous system action.

2.2 Globally-Conditioned Encoder (GCE)

In this section, we describe the proposed globally-conditioned encoder (GCE) model. Here, we employ the similar approach of learning slot-specific temporal and context representation of user utterance and system actions, as proposed in GLAD (Zhong et al., 2018). However, we emphasize the limitation of GLAD encoder in using slot-specific recurrent and self-attention layers in their encoders. Our proposed encoder is based on improving the latency and speed of inference by remving the inefficient recurrent layers and self-attention layers, without degrading the performance.

The proposed model is based on removing slot-specific recurrent and self-attention layers, and using only slot embedding vector (i.e. s_k for k-th slot), as a conditioning vector to the temporal and context extraction layers, as shown in Figure 1.

$$H^k = \text{biLSTM}(f(X, s_k)) \in \mathbb{R}^{n \times d_{rnn}} \tag{1}$$

$$a_i^k = W f(H_i^k, s_k) + b \in \mathbb{R} \tag{2}$$

$$p^k = \text{softmax}(a^k) \in \mathbb{R}^n \tag{3}$$

$$c^k = \sum_i p_i^k H_i^k \in \mathbb{R}^{d_{rnn}} \tag{4}$$

To compute k-th slot-based representation H^k, the slot embedding s_k is concatenated with sequence tokens X, i.e. user utterance or previous system actions, as input to the recurrent layer, where concatentation denoted as $f(X, s_k)$. Then a slot-based attention score a_i^k is computed for each token hidden representation H_i^k, by concatenating them to slot embedding s_k and passing to a linear layer. In this way, the computed attention is conditioned on the slot embedding, to pay attention to the slot-only information in the input sequence X.

Therefor, the GCE encoder function can be represented as,

$$\text{encode} : X, s_k \rightarrow H^k, c^k \tag{5}$$

Encoding Modules: Based on the definition of the proposed GCE encoder, the representation of user utterance, previous system action and current slot-value pair is computed as below,

$$H_{utt}^k, c_{utt}^k = \text{encode}(U, s_k) \tag{6}$$

$$H_{act_j}^k, C_{act_j}^k = \text{encode}(A_j, s_k) \tag{7}$$

$$H_{val}^k, c_{val}^k = \text{encode}(V, s_k) \tag{8}$$

where U denotes the user utterance word embeddings, A_j is the j-th previous system action, and V is the current slot value pair to be evaluated (e.g *food=italian*).

Scoring Model: We follow the proposed architecture in GLAD (Zhong et al., 2018) for computing score of each slot-value pair, in the user utterance and previous system actions.

To determine whether the user has mentioned a specific value of slot k, we compute the slot-kth conditioned scores for its values.

$$a_{utt_i}^k = (H_{utt_i}^k)^\intercal c_{val}^k \in \mathbb{R} \tag{9}$$

$$p_{utt}^k = \text{softmax}(a_{utt}^k) \in \mathbb{R}^m \tag{10}$$

$$q_{utt}^k = \sum_i p_{utt_i}^k H_{utt_i}^k \in \mathbb{R}^{d_{rnn}} \tag{11}$$

$$y_{utt}^k = W q_{utt}^k + b \in \mathbb{R} \tag{12}$$

3

Similarly, to determine whether any slot-value is mentioned in previous system actions, that the user is referring to in the current utterance, we compute slot-conditioned scores of previous j system actions.

$$a_{act_j}^k = (C_{act_j}^k)^\intercal c_{utt}^k \in \mathbb{R} \tag{13}$$

$$p_{act}^k = \text{softmax}(a_{act}^k) \in \mathbb{R}^{l+1} \tag{14}$$

$$q_{act}^k = \sum_j p_{act_j}^k C_{act_j}^k \in \mathbb{R}^{d_{rnn}} \tag{15}$$

$$y_{act}^k = (q_{act}^k)^\intercal c_{val}^k \in \mathbb{R} \tag{16}$$

The final scores of slot k is the weighted sum of user-based and action-based scores, i.e. y_{utt}^k and y_{act}^k, which are normalized by sigmoid function σ.

$$y = \sigma(y_{utt}^k + \omega y_{act}^k) \in \mathbb{R} \tag{17}$$

where ω is a learned parameter.

3 Experiment

In this section, we evaluate the proposed encoder for the task of dialogue state tracking om single and multi-domain dialogue state tracking. Wizard of oz (WoZ) restaurant reservation dataset Wen et al. (2017) is chosen for single-domain, and the performance is compared with the recent neural belief tracking models. Moreover, we also evaluate on recen;t proposed multi-domain dataset, Multi-WoZ (Budzianowski et al., 2018), which consists of seven domains, i.e. restaurant, hotel, train, attraction, taxi, and police.

The evaluation metric is based on joint goal and turn-level request and joint goal tracking accuracy. The joint goal is the accumulation of turn goals as described in Zhong et al. (2018). The fixed pretrained GLoVe embedding (Pennington et al., 2014) with character-n gram embedding (Hashimoto et al., 2017) are used in embedding layer. The implementation details and code of the GCE model can be found at https://github.com/elnaaz/GCE-Model.

Single-Domain: Table 1 shows the evaluation performance on WoZ dataset. It is indicated that our proposed GCE model performance is on par with GLAD model. To further compare the latency of GCE and GLAD during training and testing, computation time for a batch of turn and the overall epoch time during training is measured. We further evaluate the complete test time, which contains 400 dialogue and 1646 turns (WoZ test set), as shown in Table 2. The computation time is measured in second, and it is indicated that GCE improves latency in both training and testing by 35% on average.

Multi-Domain: Table 3 shows the evauation on Multi-Woz (Budzianowski et al., 2018) dataset which consists of 10k dialogues. In this setting, we completely ignore the domain information and use slot names only. The results indicate that GCE model outperforms GLAD on turn inform and join goal accuracy.

Table 1: Test accuracy on WoZ restaurant reservation dataset.

Model	WoZ	
	Joint goal	Turn request
Delex. Model (Mrksic et al., 2017)	70.8%	87.1%
Delex. + Semantic Dictionary (Mrksic et al., 2017)	83.7%	87.6%
Neural Belief Tracker-DNN (Mrksic et al., 2017)	84.4%	91.2%
Neural Belief Tracker-CNN (Wen et al., 2017)	84.2%	91.6%
GLAD (Zhong et al., 2018)	88.1±0.4%	97.1±0.2%
GCE (Ours)	**88.51%**	**97.38%**

Table 2: Time complexity for each batch of turn, and train and test epoch on WoZ dataset. Each batch contains 50 turns. All numbers are in second.

Model	Train (sec.)		Test (sec.)	
	Turn	Total	Turn	Total
GLAD (Zhong et al., 2018)	1.78	89	2.32	76
GCE (Ours)	**1.16**	**60**	**1.92**	**63**

Table 3: Performance on Multi-Domain dataset, Multi-WoZ (Budzianowski et al., 2018).

Model	split	Multi-WoZ	
		Turn inform	Joint goal
GLAD (Zhong et al., 2018)	Dev	66.91%	34.83%
	Test	66.89%	35.57%
GCE (Ours)	Dev	**67.78%**	**37.42%**
	Test	**67.88%**	**35.58%**

4 Conclusion

In this paper, we proposed a neural model for dialogue state traking. Based on globally conditioning the encoder model on slot types (GCE), slot-conditioned representations are computed for user utterance and previous system actions, which are used to compute the mentioned slot value. By relaxing GLAD model from slot-specific recurrent networks and self-attentions, our model achieved lower computational complexity with better accuracy. We also showed that GCE model is generalizable to multi-domain dialogue state tracking, by evaluation on Multi-WoZ dataset.

References

P. Budzianowski, T.-H. Wen, B.-H. Tseng, I. Casanueva, S. Ultes, O. Ramadan, and M. Gavsi'c. Multiwoz - a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. 2018.

K. Hashimoto, C. Xiong, Y. Tsuruoka, and R. Socher. A joint many-task model: Growing a neural network for multiple nlp tasks. In *EMNLP*, 2017.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9:1735–1780, 1997.

N. Mrksic, D. Ó. Séaghdha, T.-H. Wen, B. Thomson, and S. J. Young. Neural belief tracker: Data-driven dialogue state tracking. In *ACL*, 2017.

J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

A. Rastogi, D. Z. Hakkani-Tür, and L. P. Heck. Scalable multi-domain dialogue state tracking. *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 561–568, 2017.

T.-H. Wen, L. M. Rojas-Barahona, M. Gasic, N. Mrksic, P. hao Su, S. Ultes, , S. J. Young, and D. Vandyke. A network-based end-to-end trainable task-oriented dialogue system. In *EACL*, 2017.

J. D. Williams, A. Raux, D. Ramachandran, and A. W. Black. The dialog state tracking challenge. In *SIGDIAL Conference*, 2013.

P. Xu and Q. Hu. An end-to-end approach for handling unknown slot values in dialogue state tracking. In *ACL*, 2018.

V. Zhong, C. Xiong, and R. Socher. Global-locally self-attentive dialogue state tracker. In *ACL*, 2018.