A Dialogue State Tracking Model with Slot Embedding

Boyu Qiu*, Yingfei Sun
University of Chinese Academy of Sciences, Beijing, 100000, China
*Corresponding author’s e-mail: qiuboyu18@mails.ucas.ac.cn

Abstract. Dialogue State Tracking (DST) is the core component of the task-oriented dialogue systems. Although recent neural DST models have made great progress, they often ignore the phenomenon that the current dialogue state is closely related to the earlier dialogue states. In this paper, we try to introduce the slot embedding into the transformer to focus on those special tokens, which ever appear in earlier dialogue states. On the basis, we leverage the copy mechanism to predict the state over the dialogue utterances. Our model also imitates the architecture of reading comprehension model to make full use of the current utterances. The experimental results verify the benefits of the slot embedding, and our model achieves significant improvements than baselines on MultiWOZ 2.0 and MultiWOZ 2.1 datasets.

1. Introduction
The task-oriented dialogue system is responsible for understanding the user’s requests and completing some tasks on their behalf. Usually, the system includes five components: automatic speech recognition (ASR), natural language understanding (NLU), dialogue state tracking (DST), natural language generation (NLG), text to speech (TTS) [1]. Among them, DST is a connecting module, responsible for evaluating the current state of the dialogue in each turn. A dialogue state consists of a set of (domain, slot, value) triples. For example, the utterance “I am looking for a place to stay that has cheap range and it should be in a type of hotel.” contains two slots, which are respectively (hotel, price-range, cheap) and (hotel, type, hotel).

Traditional DST models tend to utilize hand-crafted features to track the dialogue states [2]. Usually, such models define the ontology in advance, and then transform the DST task into the multi-class classification task. However, the predefined ontology means that all the possible values are given, which is difficult to achieve in practical applications. In response to this situation, the generative DST models break the assumption of predefined ontology [3]-[4]. These models analyse the dialogue history between the user and the agent, and generate the values.

Although the generative DST models have achieved great improvements, they are often trapped in those tokens which occur frequently in the dialogue history. To alleviate this phenomenon, we propose the slot embedding to record whether each token ever appears in earlier dialogue states, which helps the model pay more attention to those special tokens. Besides, the current utterances should also play a different role with the earlier utterances. We make them interaction by imitating the architecture from the reading comprehension model, which is responsible for evaluating how well a machine understands the human language.

To summarize our contributions:
- We design a novel neural network model to solve the DST task. The transformer is used as the feature extractor. Different from previous work, we introduce the slot embedding into model, to emphasize those special words. Some other embeddings are also optional in our model.
- We borrow the architecture from the reading comprehension model. The current utterances are viewed as the question while the history utterances are treated as the context. The interaction between them helps the model comprehend the utterances better.
- The experimental results on public dialogue datasets MultiWOZ 2.0 [5] and MultiWOZ 2.1 [6] shows that our model performs better than the baselines.

2. Related Work
Dialogue state tracking models can be roughly divided into two categories: ontology-free models and ontology-based models. In particular, the latter approach predicts the values without predefined ontology, which is suitable for multi-domain dialogue scenarios.

Ref. [3] first proposes to extract the slot values from the dialogue history. They introduce the pointer network into the DST models, where the attention mechanism is utilized to predict the start and end position for the values. Further, Ref. [4] utilizes the bi-directional GRU to obtain the word distribution from both the history word and the vocabulary word, and then generate the slot value. Some work focus on the significance of the history dialogue states. For example, Ref. [7] designs the slot connecting module, which can directly use the previous dialogue states. Ref. [8] improves the attention mechanism to highlight the history dialogue states.

Besides, the pre-trained models (especially BERT) also achieve great performance [9]. Ref. [10] transforms the DST task into the MRC task and leverages BERT-style reading comprehension model to predict the value. Ref. [11] introduces dual strategy into BERT-style model, which jointly handle both categorical and non-categorical slots. Similar with BERT, our work chooses the transformer to extract the feature.

3. Model
3.1. Task Definition
Let us define $A = \{(U_{1}, S_{1}), \cdots,(U_{T}, S_{T})\}$ as interaction utterances between the user and the system in $T$ turns, and $B = \{B_{1}, \cdots, B_{T}\}$ as the corresponding dialogue states. Each dialogue state consists of $N$ (domain, slot, value) triplets. The DST model should predict corresponding values according to the utterances after given a set of (domain, slot) pairs. The generation process will repeat $N$ times for an entire dialogue state. Figure 1 shows the architecture of our model.

3.2. Encoding Module
In this module, we utilize transformer to embed the dialogue history [12]. Given target (domain, slot) pair and $A = \{(U_{1}, S_{1}), \cdots,(U_{T}, S_{T})\}$. We concatenate the history utterances $\{U_{1};S_{1};\cdots;U_{T-1};S_{T-1}\}$ as context and concatenate the current utterances $\{U_{T};S_{T}\}$ as question.

Then, we construct slot embedding to make full use of the earlier dialogue states. That is to say, each token in context will be record whether it ever appears in the state values. Besides, we also record which (domain, slot) pair the token belongs to. There are some optional embeddings to do additional processing for each token. For example, the frequency embedding can help the model focus on low-frequency words.

The input to the encoder is the connection of all the embeddings. After the processing with multi-head attention, the encoding module can provide a compact representation of the utterances $H$ for the subsequent module.

$$Attention(Q, K, V) = \text{softmax} \left( \frac{QK^{T}}{\sqrt{d_{k}}} \right)$$

$(1)$
Figure 1. the architecture of our proposed DST model

\[ H = \text{TransformerEncoder}(\text{word}_\text{embed}; \text{pos}_\text{embed}; \text{slot}_\text{embed}; \text{other}_\text{embed}) \]  

(2)

3.3. Decoding Module

This module is responsible for generating the values. The first input to the decoder is the embedding of target (domain, slot) pair. The decoding process is consistent with the transformer and the output can be viewed as the hidden state \( h \).

\[ h = \text{TransformerDecoder}(\text{word}_\text{embed}; \text{pos}_\text{embed}) \]  

(3)

We follow previous work to generate the word distribution from both the history word and the vocabulary word. On one hand, the embedding of target slot is sent to interact with \( H \). Then, by introducing the copy mechanism, the decoder maps the vector into the history word space. On the other hand, the decoder directly maps the hidden state \( h \) into the vocabulary space. The calculation process is shown in (4) and (5), where \( L_1, L_2 \) and \( L_3 \) are the linear mappings. The final word distribution is the weighted-sum of two distributions by a soft-gate \( g \). Specially, \( g \) is calculated by (7), where \( W_g \) is a trainable matrix and \( y \) is the input word in this decoding step.

Based on the final distribution, we use beam search to select \( k \) words with highest probability and make the selected words as the input of the next step.

\[ P^{\text{copy}} = \text{Softmax}(L_1 \cdot (L_2 \cdot (H; \text{slot}))) \in \mathbb{R}^{\text{copy}} \]  

(4)

\[ P^{\text{vocab}} = \text{Softmax}(L_3 \cdot h^T) \in \mathbb{R}^{\text{vocab}} \]  

(5)

\[ P = g \cdot P^{\text{vocab}} + (1-g) \cdot P^{\text{copy}} \]  

(6)

\[ g = \text{sigmoid}(W_g[H; h; \phi^{\text{vocab}}(y)]) \in \mathbb{R}^1 \]  

(7)
In some dialogue scenarios, *None* and *Dontcare* are also possible states. *None* means that the target slot isn’t mentioned in the dialogues and *Dontcare* indicates that the user doesn’t care the value of target slot. To tackle the issue, we introduce a slot gate to judge whether to trigger the generation process by (8), where $H$ is the output of the encoder and $W_s$ is another trainable matrix.

$$G = \text{softmax}(W_s \cdot H^\top) \in \mathbb{R}^3$$

(8)

4. Experiment

4.1. Dataset

We conduct experiments on MultiWOZ 2.0, which is the largest available multi-domain dialogue dataset (MultiWOZ 2.1 is a revised version with less noise). The dataset contains seven distinct domains across 10K dialogues. In particular, the seven domains are *Attraction*, *Hotel*, *Restaurant*, *Taxi*, *Train*, *Hospital* and *Police*. During training, we exclude the *Hospital* domain and *Police* domain because they only appear in the training set. There are 30 (domain, slot) pairs within the remaining 5 domains.

4.2. Setting

In our experiments, we use 512 as the dimension of embeddings. Both the encoder and the decoder are composed of 6 identical layers. We choose Adam optimizer with a learning rate of 0.005 to train our model. Besides, the batch size is set to 32. The beam search strategy is used in the decoding module, where $k$ is set to 2 and the maximum length is set to 10. We also introduce the teacher forcing technology when decoding.

4.3. Results

The joint goal accuracy is used to evaluate the performance of DST models. For each dialogue state, only when all possible (domain, slot, value) triples are equal to the ground truth label, it’s viewed as a positive case, otherwise negative. We compare our model with several classic ontology-free DST models: *SpanPtr* [2], *FJST* [6], *Hyst* [13], *TRADE* [3] and *DST-SC* [7].

The experimental results are shown in Table 1. Our model performs better than most baselines, but weaker than the DST-SC model, which directly utilizes the last dialogue states. It can be seen that our model is very competitive.

4.4. Ablation Study

We also conduct the ablation study to analyse each component’s influences. There are three designed architectures. (1) The model doesn’t use the slot embeddings. (2) The model doesn’t distinguish between the current utterances with the earlier utterances. (3) The model doesn’t use the copy mechanism when decoding. Table 2 shows the results. Obviously, the copy mechanism plays the most important role. To some extent, the slot embedding can be regarded as an optimization of the copy mechanism.

5. Conclusion

This paper tackles the issue that the copy mechanism in multi-domain dialogue state tracking model tends to select those tokens with high frequency. By introducing the slot embedding into transformer, our model can focus on those special tokens, which ever appear in past dialogue state values. Besides, we borrow the architecture from machine reading comprehension models to distinguish between the current utterances and the history utterances. Our model achieves significant improvements on two public dialogue datasets.

| Table 1. The joint goal accuracy on two datasets |
|-----------------------------------------------|
| **Model** | MultiWOZ 2.0 | MultiWOZ 2.1 |
| Spanptr   | 30.28%       | 29.09%       |
| FJST      | 40.20%       | 38.00%       |
| Model  | MultiWOZ 2.0 | MultiWOZ 2.1 |
|--------|--------------|--------------|
| Slot embedding | 48.73% | 46.21% |
| MRC structure | 49.23% | 47.38% |
| Copy mechanism | 43.40% | 40.32% |
| Full | 51.55% | 49.44% |

Table 2. The ablation experiments on two datasets

References

[1] Young, Steve and Gasic, Milica and Thomson, Blaise and Williams, Jason. (2012) POMDP-based Statistical Spoken Dialogue Systems: a Review. In: Proceedings of the IEEE. pp.1-20.

[2] M.Henderson, B.Thomson, and S.J.Young. (2014) Word-based dialog state tracking with recurrent neural networks. In: Proceedings of the 15th Annual SIGdial Meeting on Discourse and Dialogue. Philadelphia. pp. 292–299.

[3] P.Xu and Q.Hu. (2018) An end-to-end approach for handling unknown slot values in dialogue state tracking. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics. Melbourne. pp. 1448–1457.

[4] C.Wu, A.Madotto, E.Hosseini-Asl, C.Xiong, R.Socher, and P.Fung. (2019) Transferable multi-domain state generator for task-oriented dialogue systems. In: Proceedings of the 57th Conference of the Association for Computational Linguistics. Florence. pp. 808–819.

[5] P. Budzianowski, T. Wen, B. Tseng, I. Casanueva, S. Ultes, O. Ramadan, and M. Gasic. (2018) Multiwoz - A large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Brussels. pp. 5016–5026.

[6] Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, and Dilek HakkaniTur. (2019) Multiwoz 2.1: Multi-domain dialogue state corrections and state tracking baselines. http://arxiv.org/abs/1907.01669.

[7] Y.Ouyang, M.Chen, X.Dai, Y.Zhao, S.Huang, and J.Chen. (2020) Dialogue state tracking with explicit slot connection modeling. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Online. pp. 34–40.

[8] A.Kumar, P.Ku, A.K. Goyal, A.Metallinou, and D.Hakkani-Tur. (2020) MA-DST: multi-attention-based scalable dialog state tracking. In: The 34th AAAI Conference on Artificial Intelligence. New York. pp. 8107–8114.

[9] J.Devlin, M.Chang, K.Lee, and K.Toutanova. (2019) 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. Minneapolis. pp. 4171–4186.

[10] S.Gao, S.Agarwal, T.Chung, D.Jin, and D.Hakkani-Tur. (2020) From machine reading comprehension to dialogue state tracking: Bridging the gap. https://arxiv.org/abs/2004.05827.

[11] J.Zhang, K.Hashimoto, C.Wu, Y.Wan, P.S.Yu, R.Socher, and C.Xiong. (2019) Find or classify? dual strategy for slot-value predictions on multi-domain dialog state tracking. http://arxiv.org/abs/1910.03544.
[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. (2017) Attention is all you need. In: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017. Long Beach, CA. pp. 5998–6008.

[13] Rahul Goel, Shachi Paul, and Dilek Hakkani-Tur. (2019) Hyst: A hybrid approach for flexible and accurate dialogue state tracking. http://arxiv.org/abs/1907.00883.