

Dialogue State Tracking with Pretrained Encoder for Multi-domain Task-oriented Dialogue Systems

Dingmin Wang¹, Chenghua Lin², Li Zhong¹, Kam-Fai Wong³

¹Tencent, China

{dimmywang, reggiezhong}@tencent.com

²University of Sheffield, UK

c.lin@sheffield.ac.uk

³The Chinese University of Hong Kong, Hong Kong SAR

kfwong@se.cuhk.edu.hk

Abstract

In task-oriented dialogue systems, Dialogue State Tracking (DST) is a core component, responsible for tracking users’ goals over the whole course of conversation, which then are utilized for deciding the next action to take. Recently proposed approaches either treat DST as a classification task by scoring all enumerated slot value pairs, or adopt encoder-decoder models to generate states, which fall short in tracking unknown slot values or hold a high computational complexity. In this work, we present a novel architecture, which decomposes the DST task into three sub-tasks to jointly extract dialogue states. Furthermore, we enhance our model with a pre-trained language model and introduce domain-guided information to avoid predicting slots not belonging to the current domain. Experimental results on a multi-turn multi-domain dataset (MultiWoz) demonstrate the effectiveness of our proposed model, which outperforms previously reported results.

1 Introduction

In task-oriented dialogue systems, more especially in modular dialogue systems (Young et al., 2013), dialogue state tracking (DST) is a core component, which extracts users goals/intentions over the whole course of conversation. In order to complete a specific task, such as ticket booking or restaurant reservation, a dialogue process usually involves multiple turns between the system and the user. In a single domain dialogue, the dialogue states usually comprise a list of key-value pairs, e.g. in the “train” domain, the key-value pairs may include (departure, cambridge), (destination, oxford), etc. However, in a multi-domain dialogue, apart from extracting slots and their values, DST needs to predict the domain the current conversation content belongs to, thus the dialogue state can be represented as a triple tuple, e.g., (hotel,

people, 3) and (attraction, location, center) (Ramadan et al., 2018). In particular, as the example shown in Figure 1, some slot values can be found in the utterance, like *cambridge* and *london liverpool street*. However, some slot values are obtained based on a binary classification, as the red part involving *parking* and *internet*.

While it is possible to predefine all domains and their corresponding slot categories due to their limited range, it is infeasible to obtain all possible slot values in real-world applications. Therefore, one of the challenges in DST is to predict slot values whose range could potentially be very wide and which usually change dynamically. Advanced DST models extract slot values mostly based on generative approaches (Xu and Hu, 2018; Wu et al., 2019). However, there are some noticeable limitations of the aforementioned works. One one hand, the computational complexity of generative approaches is not constant (Ren et al., 2019) since the number of predictions is equal to the combination of domains and slots, e.g., more than 30 in MultiWoZ as shown in (Wu et al., 2019). On the other hand, previously proposed models usually directly concatenate the history content and the current utterance as input, which is difficult to scale in the multi-turn scenarios, especially when the turn of the dialogue is large.

Language model pretraining, like BERT (Devlin et al., 2019), GPT (Radford et al., 2018), has advanced the state of the art in different NLP tasks. In this paper, we explore the potential of pretrained models for DST. Especially, we decompose DST into two classification modules and one sequence labeling module, all of which are fine-tuned on top of the pretrained encoder. In particular, in the sequence labeling module, we introduce the domain constrained contextual information to avoid the selection of slots that are out of the current domain context. We apply our proposed model on Multi-

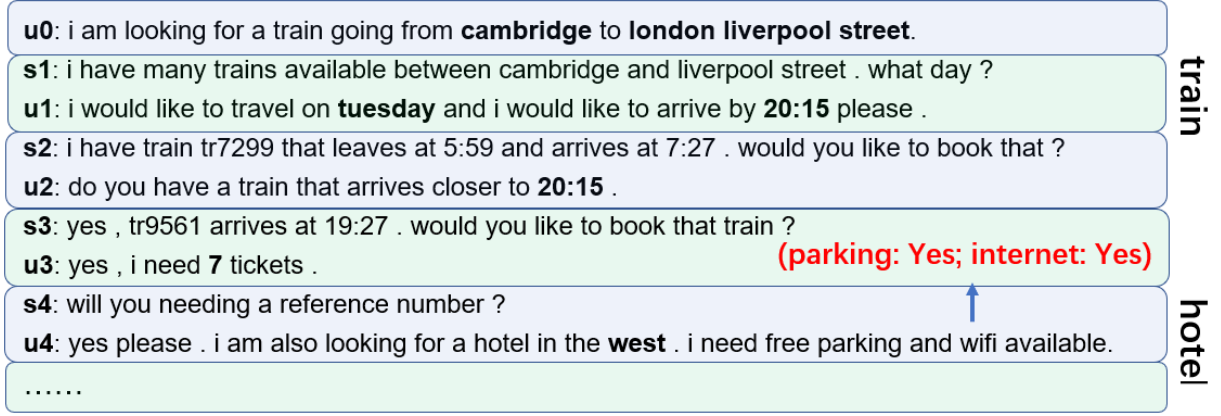


Figure 1: A multi-domain dialogue extracted from MultiWoz. The **S-type** slot values are marked in bold and the blue arrow points to two pairs of **C-type** slots and their corresponding values . The domain discussed changes from “train” to “hotel” at the fourth turn.

Woz dataset and compare it with recent state-of-the-art models (Xu and Hu, 2018; Wu et al., 2019; Ren et al., 2019). Experimental results validate the effectiveness and superiority of our proposed model, pushing the joint goal accuracy to 50.71%.

2 Our Proposed Model

First, we divide the slots into two categories - one is **S-type** slots¹, whose value can be tagged directly from the given input, such as *hotel-area* and *train-departure*; we denote another type of slots as **C-type**, whose values can not be found in the utterance and need to be answered with “Yes” or “No”, e.g. *hotel-parking* and *hotel-internet*.

Figure 2 shows the overall architecture consisting of three modules: Domain Prediction Module (DPM), **C-type** Slots Classification Module (CSCM) and **S-type** Slots Tagging Module (SSLM), which will be further elaborated.

2.1 Encoder

Given a multi-turn dialogue, previous works (Xu and Hu, 2018; Ren et al., 2018; Wu et al., 2019; Li et al., 2019) directly encode both history content and current utterances as input, and then output all the states, which casts a great challenge to the model. In our model, we output belief estimates of the dialogue at each turn, and the global dialogue state is a collection of accumulated states of different turns. When users change their goals as the dialogue proceeds, the global dialogue states will be updated with the latest detected states.

¹**S-type** means these slots are predicted in the sequence labelling module. Likewise, the following **C-type** denotes that these slot values are obtained by our classification module.

We represent a multi-turn dialogue as $D = \{(s_1, u_1, d_0), (s_2, u_2, d_1), \dots, (s_n, u_n, d_{n-1})\}$, in which s_i and u_i correspond to the system utterance and the user utterance at turn i , respectively. d_i is the domain result of the previous turn. At turn i , the input to our model is the concatenation of s_i and u_i consisting of a sequence of words $\{w_1, w_2, \dots, w_n\}$ and the previous domain d_{i-1} .

For encoder, we use a model similar to the one in (Devlin et al., 2019), in which we try two different initialization methods: one using the BERT-Large² and the another initializing our encoder with MT-DNN (Liu et al., 2019), which has the same architecture as BERT, but is trained on multiple GLUE tasks (Wang et al., 2018). The output from the encoder is represented as $\{H_{[CLS]}, H_1, H_2, \dots, H_n\}$, where n is the length of the concatenation of system and user utterances.

2.2 Domain Prediction Module

In a multi-domain dialogue, the target domain may change as the dialogue goes, as shown in Figure 1, where the dialogue involves two domains (*hotel* and *train*), and the domain discussed changes from *hotel* to *train* at 4th turn. Different from some previous works (Chen et al., 2019; Castellucci et al., 2019), which directly use the first hidden state ($H_{[CLS]}$), in our model, apart from $H_{[CLS]}$, we additionally incorporate the domain result of the last turn, denoted as D_l into the our domain prediction module. The logic behind is that when the domain of current utterances is not so obvious, D_l can provide reference information. The domain is

²www.github.com/google-research/bert

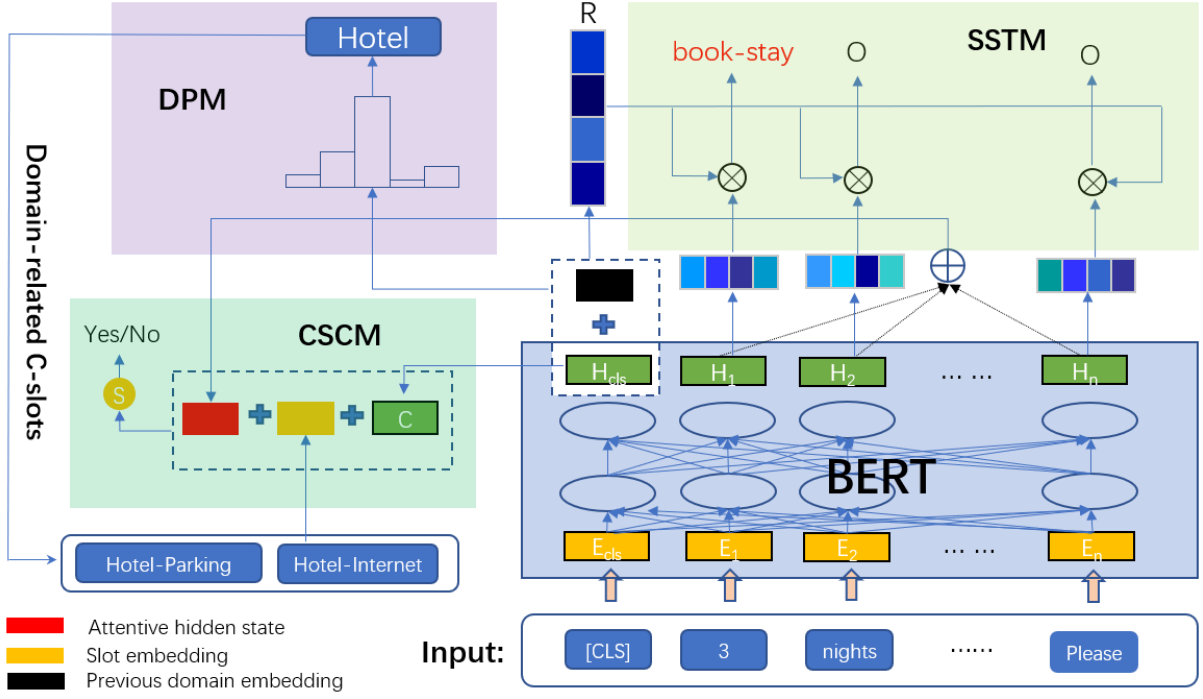


Figure 2: Our neural model architecture, which includes DPM for the domain prediction, CSCM for the binary classification of the domain-associated **C-type** slots and SSTM for tagging **S-type** slots in the given input.

predicted as:

$$y^d = \text{softmax}(W^d[H_{[CLS]}; E(D_l)]) \quad (1)$$

where $;$ denotes the concatenation operation and $E(\cdot)$ embeds a word into a distributed representation. At the first turn, D_l is a special token $[None]$ randomly initialized.

Domain constrained contextual information R Furthermore, a domain constrained contextual record $R \in \mathbb{R}^{1 \times (s+1)}$, where s is the number of **S-type** slots of all domains³, is generated to prevent our model from predicting some slots not belonging to the current domain. R is a distribution over all **G-slot** and $[EMPTY]$ using

$$R = \text{softmax}(W^R[H_{[CLS]}; E(D_l)]) \quad (2)$$

In particular, L_R , the loss for R is defined as the Kullback-Leibler (KL) divergence between $D(R^{real}||R)$, where distribution R^{real} from the ground truth is computed as follows:

- If there is no slot required to be predicted, $R_{[EMPTY]}^{real}$ receives probability mass 1 for the special slot $[EMPTY]$.

³In particular, we add a $[EMPTY]$, the value of which is expected to be 1 when there is no slot needed to be predicted.

- If the number of slots needed to be predicted is $k(\geq 1)$, then corresponding k slot positions receive probability mass of $\frac{1}{n}$, respectively.

2.3 C-type Slots Classification Module

Given the currently predicted domain result, D_c , we build a set C_{D_c} which contains all **C-type** slots from D_c . If C_{D_c} is empty, it indicates that there is no **C-type** slot needed to be predicted in the current domain. Otherwise, inspired by the works of (Ren et al., 2019; Wu et al., 2019), we execute a *For Loop Operation* to classify each slot in C_{D_c} into “Yes” or “No” as a binary classification task. The classification function is given below

$$y^c = \text{sigmoid}(W^c[E(slot_i); h_{attn}; H_{[CLS]}]), \quad (3)$$

where $E(slot_i)$ output the embedding representation for i^{th} slot in C_{D_c} , and h_{attn} is computed by

$$h_{attn} = \sum_{j=1}^n \alpha_{ij} H_j \quad (4)$$

$$\alpha_{ij} = \text{softmax}(W^{attn}[E(slot_i); H_j]) \quad (5)$$

2.4 S-type Slots Tagging Module

To tag **S-type** slots for the given input, we feed the final hidden states of H_1, H_2, \dots, H_n into a

softmax layer to classify over the all **S-type** slots,

$$y_i^s = \text{softmax}(W^s H_i), i \in 1, 2, \dots, N \quad (6)$$

where H_i is the hidden state of the word w_i .

Instead of directly predicting the **S-type** slot result based on y_i^s , we introduce a domain constrained contextual information, R (described in 2.2), aiming at avoid generating **S-type** slots that do not belong to the predicted domain. To this end, we execute an multiplication operation by

$$\hat{y}_i^s = R \odot y_i^s \quad (7)$$

where \odot is the element-wise multiplication.

During training, we use cross entropy loss for y^d , y^c and y^s , which are represented as L_{y^d} , L_{y^c} and L_{y^s} , respectively. The loss for R is defined as Kullback-Leibler (KL) divergence as described aforementioned, denoted as L_R . Lastly, all the parameters are jointly trained by minimizing the weighted-sum of three losses ($\alpha, \beta, \gamma, \theta$ are hyper-parameters):

$$\text{Loss} = \alpha L_{y^d} + \beta L_{y^c} + \gamma L_{y^s} + \theta L_R \quad (8)$$

3 Experiments

We use the default train/dev/test split of the MultiWoZ (Budzianowski et al., 2018) dataset. We adopt the joint goal accuracy (JGA) as the metric to evaluate the model performance. Besides, we initialize the encoder with Large-BERT and MT-DNN, denoted as **Ours**_{BERT} and **Ours**_{MT-DNN}, and then continue to learn all parameters in Section 2. Specifically, we train 30 epochs and use the dev set to pick the best model based on the JGC.

Overall comparison. We compare our models against three strong baselines on the multi-domain dataset MultiWoz test set. Results are reported in Table 1 based on joint goal accuracy. Experimental results show that MT-DNN achieves the best performance of 50.71%, which slightly outperforms our BERT-based model. When comparing to the baselines, both of our models outperform all the baseline models (with 2.1% to 18% performance gain), demonstrating the superiority of our proposed model.

Ablation study. We conduct two ablation experiments to investigate the impacts of D_l and R . In particular, we introduce a metric, called outlier slot ratio (OSR), denoting the proportion of slots predicted by our model that do not belong to the current domain. From Table 2, we conclude:

Model	JGA
PtrNet (Xu and Hu, 2018)	32.13%
COMER (Ren et al., 2019)	45.72%
TRADE (Wu et al., 2019)	48.62%
Ours _{BERT}	50.22%
Ours _{MT-DNN}	50.71%

Table 1: Experimental results.

- Incorporating D_l into DPM improves the domain accuracy. One possible reason is that there exist some utterances from the middle of the dialogue that do not have a clear domain attribute, thus the incorporated previous domain is believed to provide useful guiding information in the domain prediction.
- By comparing OSR from SSTM with and without using R we can observe that using R reduces the proportion of generating slots that do not align to the predicted domain, further improving the model performance.

Model	DA	OSR	JGA
Ours	95.23%	44.62%	50.71%
- D_l	91.03%	45.62%	47.62%
- R	92.13%	54.83%	45.62%

Table 2: Ablation study on the MultiWoz dataset with MT-DNN. DA refers to the domain accuracy.

4 Conclusion

In DST, we observe that the greatest challenge mainly lies in extracting slot values since the number of possible slot values could be large and variable, while the slot names are relatively limited and fixed. In this paper, propose to decompose DST into three subtasks, which jointly complete the state extraction task. In particular, the novelty of our proposed model lies in two aspects. First, we utilize the power of pretrained language models to help improve the representation. Second, we adopt Kullback-Leibler (KL) divergence as a loss for the learning of a domain-related contextual information, which is incorporated into the tagging module to avoid predicting some domain-irrelevant slots. All of these help our model to obtain a better performance on DST of the multi-domain task-oriented dialogue systems.

References

- Pawel Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica 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, Belgium, October 31 - November 4, 2018*, pages 5016–5026.
- Giuseppe Castellucci, Valentina Bellomaria, Andrea Favalli, and Raniero Romagnoli. 2019. [Multi-lingual intent detection and slot filling in a joint bert-based model](#). *CoRR*, abs/1907.02884.
- Qian Chen, Zhu Zhuo, and Wen Wang. 2019. [BERT for joint intent classification and slot filling](#). *CoRR*, abs/1902.10909.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186.
- Ruizhe Li, Chenghua Lin, Matthew Collinson, Xiao Li, and Guanyi Chen. 2019. A dual-attention hierarchical recurrent neural network for dialogue act classification. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 383–392.
- Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. [Multi-task deep neural networks for natural language understanding](#). In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4487–4496.
- Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. [URL https://s3-us-west-2.amazonaws.com/openai-assets/researchcovers/languageunsupervised/languageunderstandingpaper.pdf](https://s3-us-west-2.amazonaws.com/openai-assets/researchcovers/languageunsupervised/languageunderstandingpaper.pdf).
- Osman Ramadan, Pawel Budzianowski, and Milica Gasic. 2018. [Large-scale multi-domain belief tracking with knowledge sharing](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pages 432–437.
- Liliang Ren, Jianmo Ni, and Julian McAuley. 2019. Scalable and accurate dialogue state tracking via hierarchical sequence generation. 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)*, pages 1876–1885.
- Liliang Ren, Kaige Xie, Lu Chen, and Kai Yu. 2018. [Towards universal dialogue state tracking](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2780–2786.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. [GLUE: A multi-task benchmark and analysis platform for natural language understanding](#). In *Proceedings of the Workshop: Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@EMNLP 2018, Brussels, Belgium, November 1, 2018*, pages 353–355.
- Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. [Transferable multi-domain state generator for task-oriented dialogue systems](#). In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 808–819.
- Puyang Xu and Qi 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, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1448–1457.
- Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. 2013. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.