CACHE: MODELING CONTRIBUTION-AWARE CONTEXT HIERARCHICALLY FOR LONG-RANGE DIALOGUE STATE TRACKING

Jianshu Qi[1], *Yuke Si*[1,*], *Longbiao Wang*[1,*], *Jianwu Dang*[1,2]

[1]Tianjin Key Laboratory of Cognitive Computing and Application,
College of Intelligence and Computing, Tianjin University, Tianjin, China
[2]Japan Advanced Institute of Science and Technology, Ishikawa, Japan
{jeanshoe,siyuke,longbiao_wang}@tju.edu.cn, jdang@jaist.ac.jp

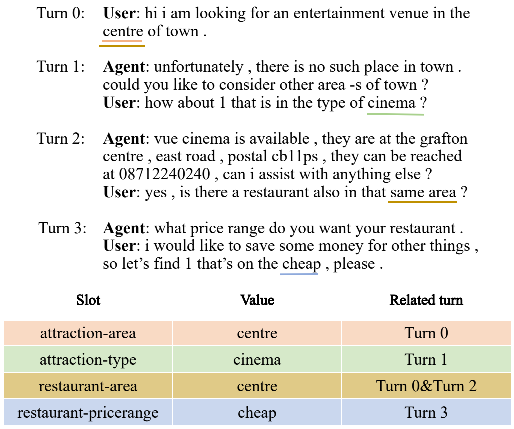
ABSTRACT

Recently, many studies on dialogue state tracking (DST) based on the copy-augmented encoder-decoder framework have been proposed and have achieved encouraging performance. However, these studies commonly lose earlier information during encoding the long dialogues with RNNs, and have difficulty for the decoder to focus on specific dialogue turns from lengthy context, which causes decreased performance as the dialogue gets longer. In this work, we propose a novel method to model **C**ontribution-**A**ware **C**ontext **HiE**rarchically (**CACHE**) with a hierarchical encoder and a slot-turn attention module. The hierarchical encoder is designed to prevent information loss by reducing the length of the sequence sent to each encoder. The slot-turn attention module is explored to help the decoder focus on the slot-related dialogue turn information. To evaluate models more appropriately, we introduce a new metric *continued joint accuracy* considering the prediction accuracy of both current and historical dialogue turns. Experiments on MultiWOZ 2.0 show that CACHE is an effective model for tracking states especially in long context.

Index Terms— dialogue state tracking, contribution-aware context modeling, continued joint accuracy

1. INTRODUCTION

Dialogue state tracking (DST) is an essential module in task-oriented dialogue systems [1]. It aims to predict user's goal with a set of slot-value pairs, also named dialogue state [2]. Since every action of a dialogue system is decided by dialogue states, it is necessary to build an accurate DST model [3]. Current studies on DST can be mainly categorized into two types: predefined ontology-based methods and open vocabulary-based methods [4]. Most traditional DST models [1, 5, 6] use the first type and have been successfully applied on single-domain datasets [7, 8]. However, they are limited by a predefined ontology, which is hard to obtain for industrial application due to the variability of data and the access right to data. To tackle this problem, the open vocabulary-based methods are proposed [2, 9, 10], which break the assumption of predefined ontology, and turn to generate values with only target slots [11]. Following this path, Xu and Hu [12] proposed a pointer network to predict a span in the dialogue history as the value of a slot. Ren et al. proposed COMER [13] with a hierarchical stacked decoder to generate the domains, slots and values in the belief state in turn. Wu et al. proposed a copy-augmented encoder-decoder framework (TRADE) to generate the slot value, which has become a paradigm commonly used in many studies.



Turn 0: **User**: hi i am looking for an entertainment venue in the centre of town .

Turn 1: **Agent**: unfortunately , there is no such place in town . could you like to consider other area -s of town ?
User: how about 1 that is in the type of cinema ?

Turn 2: **Agent**: vue cinema is available , they are at the grafton centre , east road , postal cb11ps , they can be reached at 08712240240 , can i assist with anything else ?
User: yes , is there a restaurant also in that same area ?

Turn 3: **Agent**: what price range do you want your restaurant .
User: i would like to save some money for other things , so let's find 1 that's on the cheap , please .

Slot	Value	Related turn
attraction-area	centre	Turn 0
attraction-type	cinema	Turn 1
restaurant-area	centre	Turn 0&Turn 2
restaurant-pricerange	cheap	Turn 3

Fig. 1. An example of MultiWOZ 2.0. The upper part is the dialogue context, and the lower part is the dialogue state annotation of Turn 3.

Although many progress has been made, previous work usually concatenates the utterances as a long context, and encodes it with a sequential model, which may lose earlier information. Even worse, they ignore the importance of different turns, and regard them equally important in the context representation, which may bring interference from redundant turns to the decoder. As shown in Fig. 1, there are 5 turns in the long context, but only turn 3 is useful for predicting the

*Corresponding author.

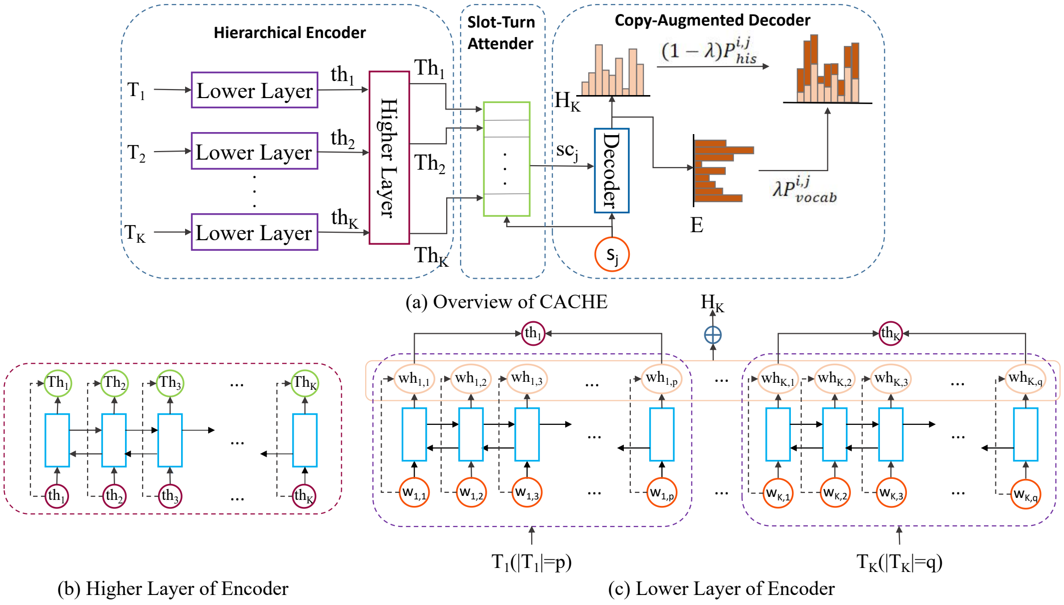


Fig. 2. The architecture of CACHE, where (a) is the general framework of the model including a hierarchical encoder, a slot-turn attention module and a copy-augmented decoder; (b) and (c) are the higher and lower layers of the encoder, respectively. (c) encodes each dialogue turn T_k to get the word vectors and turn-level representation, where $|T_k|$ indicates the length of sequence T_k; (b) encodes all turn-level representations to get the context-level representations for all turns.

value of the slot "restaurant-pricerange". Furthermore, common metrics only focus on the prediction accuracy of current dialogue turn. However, the current turn state prediction is based on the historical predicted results in real applications.

To solve the aforementioned problems, we novelly model **C**ontribution-**A**ware **C**ontext **Hi**Erarchically (**CACHE**) based on TRADE. Specifically, a hierarchical encoder is designed to model each dialogue turn on turn- and context- level. In this way, the information loss will be alleviated by reducing the length of sequence sent to each encoder. Then, the contribution-aware context is calculated by a slot-turn attention module, helping the decoder exploit the slot-related information in the context. Furthermore, we introduce a new metric called *continued joint accuracy* to measure the state tracking ability of a DST model with considering the accuracy in the entire dialogue turns of a conversation.

2. PROPOSED CACHE MODEL

The proposed CACHE model (shown in Fig. 2(a)) contains three modules: a hierarchical encoder, a slot-turn attention module, and a copy-augmented decoder. The model takes the entire dialogue history as the input. In the hierarchical encoder, the dialogue history is first split by turns, and then we send each turn into the lower layer of the encoder separately to obtain turn-level representations. Next, these representations are concatenated and processed by the higher layer to get the context-level representations. In the slot-turn attention module, all the context-level turn are used for scoring to indicate the contribution of each turn for current slot value generation, by which we can get a slot-specific contribution-aware context. The copy-augmented decoder generates the value for each slot from either vocabulary or dialogue history according to the slot-specific context representation.

2.1. Hierarchical Encoder

A two-layer hierarchical encoder is designed to process the dialogue history. The lower layer is used for turn-level encoding while the higher layer is applied to get the context-level representation for each turn. First, we split the dialogue history by turns, which are denoted as $X_K = \{T_1, T_2, ..., T_K\}$, where $T_k = \{w_{k,1}, w_{k,2}, ..., w_{k,|T_k|}\} \in R^{|T_k|*d_{emb}}$ represents the k-th dialogue turn, $|T_k|$ is the number of tokens in the k-th turn, and d_{emb} represents the embedding size.

In the lower layer (Fig. 2(c)), a Bi-GRU [14] is used to encode each dialogue turn T_k. First, each word in T_k is mapped to a distributed embedding $w_{k,i}$. Then, the word embeddings are encoded into hidden states, which can be represented as $T_k^{'} = \{h_{k,1}, h_{k,2}, ..., h_{k,|T_k|}\} \in R^{|T_k|*d_{hdd}}$, where d_{hdd} is the hidden size of the encoder. To optimize the network, we use a residual connection [15] around the lower layer to obtain the final vector $wh_{k,i}$ for each word in the utterance. Then, the word vectors in all turns are concatenated as the dialogue history $H_K = \{wh_{1,1}, wh_{1,2}, ..., wh_{K,|T_K|}\} \in R^{|H_K|*d_{hdd}}$. Moreover, we add the forward hidden state of the last word and the backward hidden state of the first word together as the turn-level representation th_k for the k-th dialogue turn:

$$h_{k,i} = GRU_{lower}\left(w_{k,i}, h_{k,i-1}\right) \tag{1}$$

$$wh_{k,i} = w_{k,i} + h_{k,i} \tag{2}$$

$$th_k = \underset{h_{k,1}}{\leftarrow} + \underset{h_{k,|T_k|}}{\rightarrow} \tag{3}$$

In the higher layer (Fig. 2(b)), we concatenate the turn-level representations as a dialogue sequence $C_K = \{th_1, th_2, ..., th_K\}$ and use another Bi-GRU [14] to encode them so that they can obtain context information. We denote these representations as $C_K^{'} = \left\{th_1^{'}, th_2^{'}, ..., th_K^{'}\right\}$. Similar to the lower layer, we use a residual connection structure [15] around the higher layer to optimize the network and obtain the context-level representation Th_k for k-th turn:

$$th_k^{'} = GRU_{higher}\left(th_k, th_{k-1}^{'}\right) \tag{4}$$

$$Th_k = th_k + th_k^{'} \tag{5}$$

2.2. Slot-turn Attention Module

There are many turns in most multi-domain dialogues. However, different turns are not helpful equally to the slot value prediction. To model the context effectively, the slot-turn attention is introduced to score the dialogue turns with an attention mechanism, which indicates the contribution each turn will make to the value prediction for target slot. Then, the slot-specific contribution-aware context is obtained by the weighted summation of all turns with corresponding scores:

$$score_{i,j} = s_j Th_i \tag{6}$$

$$w_{i,j} = \frac{exp(score_{i,j})}{\sum_{n=1}^{K} exp(score_{n,j})} \tag{7}$$

$$sc_j = \sum_{i=1}^{K} w_{i,j} Th_i \tag{8}$$

where s_j is the embedding of j-th slot, $w_{i,j}$ is the result of the softmax for $score_{i,j}$ and sc_j indicates the j-th slot-specific contribution-aware context.

2.3. Copy-Augmented Decoder

The copy-augmented decoder is designed to predict the slot value. Different from most previous approaches feeding the dialogue with all turns equally into a decoder, we employ slot-specific contribution-aware context to help the decoder focus more on the current slot-related dialogue turn information.

We use a unidirectional GRU [14] as the decoder, which is initialized with the contribution-aware context as the initial hidden state and the slot embedding as the first input. The decoder updates the hidden state recurrently until an [EOS] token is generated. At the i-th decoding step for j-th slot, we map the hidden state $dh_{i,j}$ into the space of vocabulary and dialogue history separately and obtain two distributions $P_{vocab}^{i,j}$ and $P_{his}^{i,j}$, which indicate the probability of selecting a word from the vocabulary and dialogue history, respectively.

Table 1. Statistics of MultiWOZ 2.0 and its training, dev, and test sets. Max length indicates the number of tokens in the longest dialogue. The last five lines are the numbers of dialogue turns in the corresponding ranges.

Corpus	All	Training	Dev	Test
Max Length	879	879	659	615
0-99	30797	24954	2936	2907
100-199	22956	18005	2469	2482
200-299	13330	10293	1535	1502
300-399	3616	2834	382	400
400-	711	582	52	77

$$P_{vocab}^{i,j} = Softmax\left(E \cdot dh_{i,j}\right) \in R^{|V|} \tag{9}$$

$$P_{his}^{i,j} = Softmax\left(H_K \cdot dh_{i,j}\right) \in R^{|H_K|} \tag{10}$$

where E is the embedding matrix of the vocabulary and $|V|$ is the size of it, $|H_K|$ is the number of words in the dialogue history and H_K represents corresponding vector sequence.

The final distribution $P_{final}^{i,j}$ is the weighted sum of above two distributions:

$$P_{final}^{i,j} = (1-\lambda)P_{his}^{i,j} + \lambda P_{vocab}^{i,j} \tag{11}$$

The λ, as the weight, is determined by three factors:

$$\lambda = Sigmoid(W[dh_{i,j} \oplus dw_{i,j} \oplus wd_{i,j}]) \tag{12}$$

$$wd_{i,j} = P_{his}^{i,j} H_t \tag{13}$$

where $dw_{i,j}$ is the input to the decoder, and $wd_{i,j}$ is the weighted dialogue context.

3. EXPERIMENTS

We compare CACHE with baselines TRADE [9] and COMER [13]. Previous studies show that TRADE can be improved by employing extra components such as auxiliary language model learning (MLCSG [2]) and slot information sharing (SAS [16]). To verify the flexibility and extensibility of CACHE, we take MLCSG as an example, extend CACHE with the auxiliary language model introduced by MLCSG represented by CACHE+LM, and compare them.

3.1. Dataset

Experiments are conducted on MultiWOZ 2.0 [17]. Table 1 presents the statistics of dialogues according to length. The proportion of long dialogues with over 100 tokens is over 50%, where about 1% dialogues are longer than 400 tokens.

Table 2. Results on MultiWOZ 2.0. For TRADE and MLCSG, we reproduced them by source codes, and for COMER, we used the results reported in the original papers.

Model	Slot Accuracy	Joint Accuracy	Continued Joint Accuracy
TRADE	96.94%	48.53%	35.74%
COMER	-	48.79%	-
CACHE	96.99%	49.54%	35.91%
MLCSG	**97.18%**	50.72%	38.37%
CACHE+LM	97.15%	**50.96%**	**38.38%**

Table 3. Joint accuracy of the models on the test set of MultiWOZ 2.0 split by different length ranges.

	0-99	100-199	200-299	300-399	400-
TRADE	71.86%	41.82%	25.50%	14.25%	11.69%
CACHE	71.24%	42.95%	29.09%	**16.25%**	**14.29%**
MLCSG	**73.07%**	45.37%	27.90%	15.50%	7.79%
CACHE+LM	72.21%	**46.17%**	**29.29%**	15.75%	9.09%

3.2. Evaluation Metrics and Implementation Details

Besides commonly used slot accuracy and joint accuracy [18], we introduce a new metric *continued joint accuracy*. Since a fluent dialogue needs accurate dialogue states in each turn, the proposed metric considers a dialogue state to be correct only if the dialogue states of the current turn and all previous turns in the dialogue history are all predicted correctly.

We concatenate GloVe embedding [19] and character-wise embedding [20] as the word embedding with a dimension of 400. For the residual connection, we also set the hidden size of the GRU [14] to 400. In the training process, we employ the Adam optimizer to update the parameters of the model with a batch size of 32. The training will be early stopped according to joint accuracy with a patience of six.

3.3. Results and Discussion

The experimental results are shown in Table 2. As for the slot accuracy, the results of all the models are very close and pretty good. The reason is that in each turn, the values of most slots are none, which is easier for models to predict. Therefore, we focus on the performance of models in joint accuracy. To verify the effectiveness of CACHE, we compare it with TRADE and COMER. The results show CACHE gained 1.01% and 0.75% absolute improvement respectively, which illustrates CACHE is effective. MLCSG is a TRADE-based method, improving TRADE by adding an auxiliary language model. To test the flexibility and extensibility of CACHE, we compare CACHE+LM with MLCSG. The results show CACHE+LM achieved an absolute improvement of 0.24%, indicating that our method is still effective when transplanted to TRADE-based methods. As for continued joint accuracy, the results of CACHE, MLCSG, and CACHE+LM were better than TRADE's. It indicates tracking states accurately in a long context is beneficial for accomplishing the DST task.

To further evaluate the ability of models on tracking states in long contexts, we classified the dialogues according to their lengths (0-99, 100-199, 200-299, 300-399, and 400-), and did more experiments on these five groups. The results show that our CACHE obtained excellent performance on dialogues longer than 100. Notably, in the range of 200-299, we obtained an absolute improvement of 3.59% compared with TRADE. Moreover, CACHE+LM also obtained better results compared with MLCSG when the length of dialogues is longer than 100. Consequently, our proposed model performs better on tracking dialogue states in long dialogues. However, the CACHE-based methods perform a little worse than TRADE-based methods in short dialogues (0-99). It may be caused by slight information loss during information passing from word-embedding level to higher turn-encoding level. According to the results, we speculate that the complete dialogue history and the weighted importance of different turns are more useful for long-range DST.

Furthermore, the number of dialogues decreased dramatically as the length increases (Table 1), which is thought to be the reason why our model achieves a relatively small improvement on the overall performance (1.01%), even though it is proved to be much effective in the long context. Similarly, for dialogues 300-399 and 400- groups, all the methods had poor performance. Noticeably, CACHE achieved the best performance and even surpass the CACHE+LM. This may be caused by the limited data, and extra LM usually needs more data for training. Besides, for dialogues 400- group, MLCSG have sharply decreasing results and even obtained the worst result. Instead, CACHE and CACHE+LM decreased slower, and CACHE still maintained a high performance, which indicates that CACHE is less insensitive to the data amount.

4. CONCLUSION AND FUTURE WORK

In this work, we introduced an effective method CACHE to model contribution-aware context for long-range DST. We designed a hierarchical encoder to prevent information loss while modeling long context; we also designed a slot-turn attention to help the decoder to capture slot-related dialogue turn information. Although CACHE only achieved a 1.01% improvement on the MultiWOZ 2.0, it showed encouraging performance on DST in longer conversations. The improvement of 3.59% on joint accuracy in 200-299 group demonstrated CACHE is more suitable for complex real-life situations. Additionally, we proposed a new metric named continued joint accuracy to measure the DST models more reasonably by considering the whole dialogue. The future work is to adapt CACHE to the short dialogues for further improvement.

5. ACKNOWLEDGEMENTS

This work was supported by the National Key R&D Program of China (Grant No. 2018YFB1305200) and the National Natural Science Foundation of China (Grant No. 62176182).

6. REFERENCES

[1] Victor Zhong, Caiming Xiong, and Richard Socher, "Global-locally self-attentive dialogue state tracker," *arXiv preprint arXiv:1805.09655*, 2018.

[2] Jun Quan and Deyi Xiong, "Modeling long context for task-oriented dialogue state generation," *arXiv preprint arXiv:2004.14080*, 2020.

[3] Sungdong Kim, Sohee Yang, Gyuwan Kim, and Sang-Woo Lee, "Efficient dialogue state tracking by selectively overwriting memory," *arXiv preprint arXiv:1911.03906*, 2019.

[4] Mihail Eric, Rahul Goel, Shachi Paul, Adarsh Kumar, Abhishek Sethi, Peter Ku, Anuj Kumar Goyal, Sanchit Agarwal, Shuyang Gao, and Dilek Hakkani-Tur, "Multiwoz 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines," *arXiv preprint arXiv:1907.01669*, 2019.

[5] Hwaran Lee, Jinsik Lee, and Tae-Yoon Kim, "Sumbt: Slot-utterance matching for universal and scalable belief tracking," *arXiv preprint arXiv:1907.07421*, 2019.

[6] Yong Shan, Zekang Li, Jinchao Zhang, Fandong Meng, Yang Feng, Cheng Niu, and Jie Zhou, "A contextual hierarchical attention network with adaptive objective for dialogue state tracking," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 6322–6333.

[7] Matthew Henderson, Blaise Thomson, and Jason D Williams, "The second dialog state tracking challenge," in *Proceedings of the 15th annual meeting of the special interest group on discourse and dialogue (SIGDIAL)*, 2014, pp. 263–272.

[8] Tsung-Hsien Wen, David Vandyke, Nikola Mrksic, Milica Gasic, Lina M Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young, "A network-based end-to-end trainable task-oriented dialogue system," *arXiv preprint arXiv:1604.04562*, 2016.

[9] Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung, "Transferable multi-domain state generator for task-oriented dialogue systems," *arXiv preprint arXiv:1905.08743*, 2019.

[10] Jieyu Li, Su Zhu, and Kai Yu, "A hierarchical tracker for multi-domain dialogue state tracking," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 8014–8018.

[11] Yawen Ouyang, Moxin Chen, Xinyu Dai, Yinggong Zhao, Shujian Huang, and Jiajun Chen, "Dialogue state tracking with explicit slot connection modeling," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 34–40.

[12] Puyang Xu and Qi Hu, "An end-to-end approach for handling unknown slot values in dialogue state tracking," *arXiv preprint arXiv:1805.01555*, 2018.

[13] Liliang Ren, *Scalable and accurate dialogue state tracking via hierarchical sequence generation*, University of California, San Diego, 2020.

[14] 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.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[16] Jiaying Hu, Yan Yang, Chencai Chen, Liang He, and Zhou Yu, "Sas: Dialogue state tracking via slot attention and slot information sharing," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 6366–6375.

[17] Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Inigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić, "Multiwoz–a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling," *arXiv preprint arXiv:1810.00278*, 2018.

[18] Elnaz Nouri and Ehsan Hosseini-Asl, "Toward scalable neural dialogue state tracking model," *arXiv preprint arXiv:1812.00899*, 2018.

[19] Jeffrey Pennington, Richard Socher, and Christopher D Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[20] Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher, "A joint many-task model: Growing a neural network for multiple nlp tasks," *arXiv preprint arXiv:1611.01587*, 2016.