Amendable Generation for Dialogue State Tracking

Xin Tian, Liankai Huang, Yingzhan Lin, Siqi Bao, Huang He,
Yunyi Yang, Hua Wu, Fan Wang, Shuqi Sun
Baidu Inc., China
{tianxin06,huangliankai,linyingzhan01,baosiqi,hehuang,
yangyunyi01,wu_hua,wang.fan,sunshuqi01}@baidu.com

Abstract

In task-oriented dialogue systems, recent dialogue state tracking methods tend to perform one-pass generation of the dialogue state based on the previous dialogue state. The mistakes of these models made at the current turn are prone to be carried over to the next turn, causing error propagation. In this paper, we propose a novel **A**mendable **G**eneration for **D**ialogue **S**tate **T**racking (**AG-DST**), which contains a two-pass generation process: (1) generating a primitive dialogue state based on the dialogue of the current turn and the previous dialogue state, and (2) amending the primitive dialogue state from the first pass. With the additional amending generation pass, our model is tasked to learn more robust dialogue state tracking by amending the errors that still exist in the primitive dialogue state, which plays the role of reviser in the double-checking process and alleviates unnecessary error propagation. Experimental results show that AG-DST significantly outperforms previous works in two active DST datasets (MultiWOZ 2.2 and WOZ 2.0), achieving new state-of-the-art performances.

1 Introduction

Dialogue state tracking (DST) is a crucial task in task-oriented dialogue systems, as it affects database query results as well as the subsequent policy prediction (Chen et al., 2017). It extracts users' goals at each turn of the conversation and represents them in the form of a set of (*slot*, *value*) pairs, i.e., dialogue state.

Traditional methods of DST mainly rely on a predefined ontology which includes all possible slots and corresponding values. These models predict the value for each slot as a classification problem (Mrkšić et al., 2017; Zhong et al., 2018; Ramadan et al., 2018). However, in practical applications, some slot values appearing in the conversations cannot be predefined, and it is infeasible to acquire a fully predefined ontology or transfer to other domains with fixed predefined ontology. To address such challenges, open-vocabulary DST has been proposed, where the value of each slot is directly generated or extracted from the dialogue history (Chao and Lane, 2019; Hosseini-Asl et al., 2020; Ham et al., 2020; Heck et al., 2020). Although this approach offers scalability and is capable of handling unseen slot values, many of the previous models are not efficient enough as they need to predict the dialogue state from scratch based on the dialogue history.

On the merits of utilizing the previous dialogue state as a compact representation of the previous dialogue history, some recent methods choose to take the previous dialogue state into consideration when generating the slot values. One direction is to decompose DST into two explicit sub-tasks: State Operation Prediction and Value Generation (Kim et al., 2020; Zeng and Nie, 2020). At each turn, whether or how to modify the value in the previous dialogue state is determined by the discrete operations from the state operation prediction, so the accuracy of state operation prediction holds back the overall DST performance (Kim et al., 2020). Another direction of recent works recasts dialogue state tracking into a single causal language model by using the dialogue of the current turn and the previous dialogue state as input sequence (Lin et al., 2020; Yang et al., 2021), where the current dialogue state is generated by jointly modeling the state operation prediction and value generation in a implicit fashion. While it is more effective and reasonable to use the previous dialogue state under the Markov assumption, the mistakes of these models made during the prediction of the current turn are prone to be carried over to the next turn, causing error propagation. These carried-over mistakes are unlikely to be fixed in the next turn. Essentially, these models perform a one-pass generation process and lack a double-checking process to amend the mistakes of

Proceedings of the Third Workshop on Natural Language Processing for Conversational AI, pages 80–92

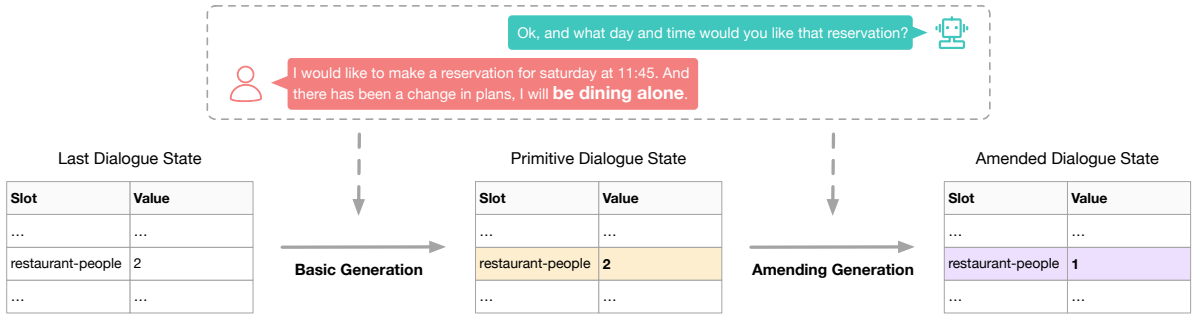


Figure 1: An example of AG-DST with two-pass generation process. In this example, the user wants to change *restaurant-people* from 2 to 1 by "... *a change in plans, I will be dining alone*". In the two-pass generation process, the amending generation obtains corresponding information to correct mistake in the basic generation.

the current turn. Missing such amending process would result in some potential mistakes being left unfixed.

To nip the mistakes in the bud and alleviate the error propagation problem, we propose **A**mendable **G**eneration for **D**ialogue **S**tate **T**racking (**AG-DST**), a pretrained language model that generates the dialogue state based on the dialogue of the current turn and previous dialogue state. In contrast to previous one-pass generation process (Kim et al., 2020; Zeng and Nie, 2020; Lin et al., 2020; Yang et al., 2021), AG-DST employs a two-pass generation process consisting of a basic generation and an amending generation, where the first pass uses the dialogue of the current turn and the previous dialogue state to generate a primitive dialogue state, and second pass utilizes the dialogue of the current turn to amend the primitive dialogue state. With the additional amending generation pass, our model is tasked to learn more robust dialogue state tracking by amending the errors that still exist in the primitive dialogue state. These errors are more challenging to fix and relatively scarce during training. Therefore, we further design a negative sampling mechanism to exploit more challenging errors and facilitate the effective learning of the amending generation pass. With such two-pass generation process, AG-DST is less likely to generate false dialogue state for the next turn, and thus reduces error propagation. Figure 1 illustrates a complete dialogue state generation process of AG-DST.

Experimental results show that AG-DST consistently outperforms all prior works on MultiWOZ 2.2 and WOZ 2.0. Especially on MultiWOZ 2.2, AG-DST achieves 57.26% joint goal accuracy, 2.86% higher than the previous state-of-the-art performance. Besides, we provide ablation study and the attention visualization to demonstrate the effectiveness of the amending generation, and analyze the types of mistakes that can be corrected by the amending generation. Our models and code will be released for further research.[1]

2 Methodology

In this section, we introduce AG-DST in the following aspects: the basic generation, the amending generation and the training objective.

2.1 Basic Generation

A dialogue with T turns can be represented as $D = \{D_1, D_2, \ldots, D_T\}$, where D_t is the dialogue at turn t consisting of system response R_t and user utterance U_t. We denote the dialogue states at every turn as $\mathcal{B} = \{\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_T\}$. For multi-domain DST, the dialogue state at turn t is denoted as $\mathcal{B}_t = \{(S^i, V_t^i) | 1 \leq i \leq I\}$, in which S^i is the slot and V_t^i is its corresponding value (I is the number of all slots in different domains). Particularly, S^i is represented as a special token concatenated by domain and slot (i.e. `<domain-slot>`) following most previous works. We use special tokens `<nm>` and `<dc>` to indicate *not mentioned* and *don't care* in slot value respectively.

We leverage the dialogue of the current turn and the previous dialogue state to generate the current dialogue state. The dialogue of the current turn D_t is used in the input tokens under the Markov assumption. To a certain extent the previous dialogue state B_{t-1} could be viewed as a compact representation of the previous dialogue history (Kim et al., 2020). Specifically, we denote the dialogue at turn t as:

$$D_t = [R_t; U_t] \quad (1)$$

[1] https://github.com/PaddlePaddle/Knover/tree/develop/projects/AG-DST

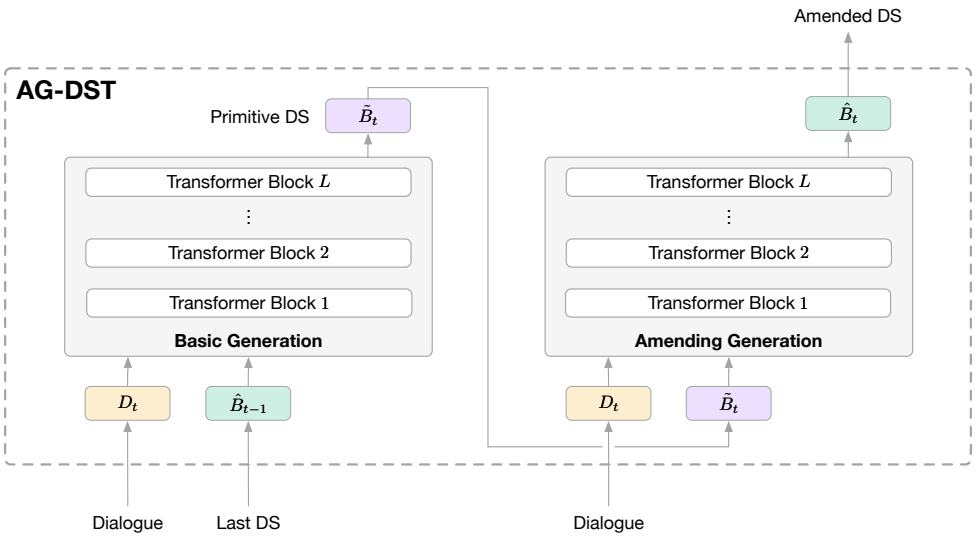


Figure 2: The overview of the proposed AG-DST. In the basic generation, AG-DST takes the dialogue of the current turn D_t and the previous dialogue state $\hat{B}_{t-1}$ as input and generates the primitive dialogue state $\tilde{B}_t$. In the amending generation, AG-DST takes the dialogue of the current turn D_t and the primitive dialogue state $\tilde{B}_t$ as input and outputs the amended dialogue state $\hat{B}_t$.

where R_t and U_t are system response and user utterance accordingly. Special tokens <con/> and </con> are added around D_t for marking the boundary of the dialogue context, and special tokens <sys> and <usr> are added before R_t and U_t respectively to indicate the role. The dialogue state at turn t is denoted as

$$B_t = [B_t^1; B_t^2; \ldots; B_t^I] \tag{2}$$

where $B_t^i = [S^i; V_t^i]$ is the concatenation of i-th slot and its value. Similar to the dialogue context, two special tokens <ds/> and </ds> are added around the whole dialogue state.

The overview of AG-DST is illustrated in Figure 2. AG-DST is a generation model based on transformer (Vaswani et al., 2017; Dong et al., 2019). In the basic generation of AG-DST, the input sequence is composed of the current turn dialogue and the previous dialogue state, and the primitive dialogue state is predicted by:

$$\tilde{B}_t = Transformer(D_t, B_{t-1}) \tag{3}$$

where <gen/> and </gen> are added around the whole input sequence to indicate the first pass generation process.

As shown in Figure 3, the input embedding of each token is the sum of token embedding, position embedding, role embedding and segment embedding. Among them, position embedding is added to discriminate input token positions; role embedding is employed to distinguish the characters of the speaker in the dialogue; segment embedding is used for different types of sequence.

2.2 Amending Generation

To amend the potential errors in the primitive dialogue state, we propose a novel amending generation that takes the dialogue of the current turn and the primitive dialogue state predicted by the basic generation as input, and generates the amended dialogue state:

$$\hat{B}_t = Transformer(D_t, \tilde{B}_t) \tag{4}$$

where the new input sequence of amending generation is consisted of the current turn dialogue and the primitive dialogue state, $\hat{B}_t$ is the amended dialogue state. The amending generation shares the same parameters with the basic generation model. To differentiate this two-pass generation process, the special tokens <amend/> and </amend> are added around the new input sequence in Equation 4 as opposed to the <gen/> and </gen> in Equation 3.

Negative Sampling To facilitate effective learning of the amending generation process, we propose a negative sampling strategy that actively mines the examples on which the model is prone to make mistakes (i.e., generating the wrong slot values, or failing to update some slots). Specifically, we performs negative sampling on the dialogue state with changed slot values between turns

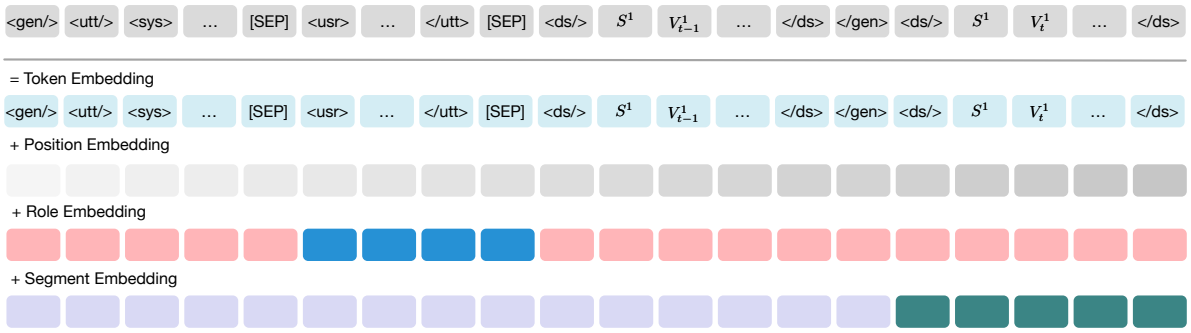


Figure 3: An example of input representation. The input embedding of each token is the sum of token embedding, position embedding, role embedding and segment embedding.

$\Delta\mathcal{B}_t = \mathcal{B}_t - \mathcal{B}_t \cap \mathcal{B}_{t-1}$, where these slot values are randomly replaced by <nm>, <dc> or some wrong values. During training, the dialogue state after negative sampling is used as the primitive dialogue state $\tilde{B}_t$ in the amending generation to encourage the model to lay more emphasis on error correction.

2.3 Training Objective

The training objective of the basic generation is the negative log-likelihood loss given the dialogue of the current turn and the previous dialogue state:

$$\mathcal{L}_{basic} = -\log P(B_t|D_t, B_{t-1}) \quad (5)$$

Similar to the basic generation, the loss of the amending generation is also the negative log-likelihood loss:

$$\mathcal{L}_{amending} = -\log P(B_t|D_t, \tilde{B}_t) \quad (6)$$

The total objective of AG-DST is to minimize the sum of the above two losses:

$$\mathcal{L} = \mathcal{L}_{basic} + \mathcal{L}_{amending} \quad (7)$$

3 Experiments

3.1 Datasets

We conduct our experiments on both multi-domain dataset MultiWOZ 2.2 (Zang et al., 2020) and single-domain dataset WOZ 2.0 (Wen et al., 2017). MultiWOZ (Budzianowski et al., 2018) is a large-scale multi-domain dialogue dataset with human-human conversations including over 10,000 dialogues. It is a widely used benchmark for dialogue state tracking. MultiWOZ 2.2 (Zang et al., 2020) is a new version of MultiWOZ 2.1 (Eric et al., 2019), in which a number of dialogue state annotation errors across 17.3% of the utterances have been fixed.

Statistics	MultiWOZ 2.2	WOZ 2.0
# domains	5	1
# slots	30	3
# dialogues	10424	1200
# train dialogues	8426	600
# valid dialogues	998	200
# test dialogues	1000	400
Avg. turns per dialogue	13.71	8.35
Avg. tokens per turn	16.86	13.18

Table 1: Statistics of the datasets in the experiments.

Following Wu et al. (2019), due to the absence of *hospital* and *police* domains in the validation and test datasets, there are only 5 domains (*attraction*, *hotel*, *restaurant*, *taxi* and *train*) and 30 corresponding domain-slot pairs in our experiments. WOZ 2.0 (Wen et al., 2017) is a well-known single-domain DST dataset, where 3 slots (*area*, *food* and *price range*) are involved in the *restaurant* domain. Table 1 summarizes the statistics of the above two datasets.

3.2 Implementation Details

Our AG-DST approach can be easily deployed with many pre-trained generation models (such as GPT-2 (Radford et al., 2019), T5 (Raffel et al., 2020), PLATO-2 (Bao et al., 2021)). In this paper, we employ the GPT-2 and PLATO-2 to initialize our model parameters. GPT-2 is a large causal language model, and PLATO-2 is a large-scale open-domain dialogue model trained on the Reddit comments. Specifically, GPT-2 has 117M parameters containing 12 transformer blocks, 12 attention heads and 768 hidden units, and PLATO-2 has 310M parameters containing 24 transformer

blocks, 16 attention heads and 1024 hidden units.[2] Adam optimizer (Kingma and Ba, 2015) is employed for optimization in all experiments. For the hyper-parameters we used in the best model, the PLATO-2 is fine-tuned with a dynamic batch size of 8192 tokens, a learning rate of $1e-5$, warmup steps of 1000 and a learning rate decay rate of 0.01. For GPT-2, we set the batch size of 6, a learning rate of $5e-5$ with no warmup and learning rate decay. All the models are trained on 4 Nvidia Telsa V100 32G GPU cards for 40 epochs and early stop according to the performance on the validation set. All the reported results of AG-DST are averages over five runs.

3.3 Baselines

We compare our approach with the following methods:

Neural Belief Tracker (Mrkšić et al., 2017) learns the distributed representation of system responses and user utterances from pre-trained word vectors, and decides which slot-value pairs are required by the user.

Belief Tracking (Ramadan et al., 2018) proposes a model that utilizes semantic similarity between ontology terms and utterances and shares the information across domains.

GLAD (Zhong et al., 2018) uses global modules to share parameters across slots and local modules to learn the features which take into account the specific slot.

StateNet (Ren et al., 2018) proposes a model that composes a representation of the dialogue history and measures the distances between the representation and the value vectors.

TRADE (Wu et al., 2019) uses a state operation with an utterance encoder and a dialogue state generator to handle the cross domain phenomenon.

DS-DST (Zhang et al., 2020) proposes a dual strategy: picklist-based and span-based. The span-based strategy includes a slot-gate classifier and span-based slot-value prediction.

BERT-DST (Chao and Lane, 2019) adopts BERT to encode dialogue context and extracts slot values by the state operation prediction and the span prediction.

SOM-DST (Kim et al., 2020) proposes an efficient decoder by updating dialogue state as a memory to reduce the burden of the decoder.

COMER (Ren et al., 2019) adopts two sequential decoders to generate dialogue state. The first decoder is used to generate state sketch (i.e. domains and slots), and the second one is used to generate slot values by conditioning on the dialogue history and state sketch.

SGD-baseline (Rastogi et al., 2020) adapts BERT to obtain the schema embeddings (including intents, slots and possible values of categorical slots) and utterance embeddings and uses different strategies for non-categorical and categorical slots.

SimpleTOD (Hosseini-Asl et al., 2020) changes sub-tasks of task-oriented dialogue into a single causal language model which generates dialogue state, system action and system response successively.

Seq2Seq-DU (Feng et al., 2021) applies schema descriptions to deal with unseen domains with a sequence-to-sequence framework.

3.4 Experimental Results

We use joint goal accuracy (Joint Acc) as our main evaluation metric for dialogue state tracking. Joint goal accuracy measures the percentage of *correct* in all dialogue turns, where a turn is considered as *correct* if and only if all the slot values are correctly predicted. We also show the slot accuracy for each domain which measures the accuracy of all the slot values in that specific domain.

Table 2 shows the performance of the AG-DST in comparison to the baselines. We can observe that the AG-DST consistently outperforms all baselines on both MultiWOZ 2.2 and WOZ 2.0 datasets, achieving a new state-of-the-art performance. On MultiWOZ 2.2, AG-DST achieve 57.26% joint goal accuracy, by 2.86% significant improvement on the top of the Seq2Seq-DU (Feng et al., 2021), the latest sequence-to-sequence generation model. In addition, the domain-specific results of our approach are also provided in Table 3. On the single-domain dataset, WOZ 2.0, we obtain joint goal accuracy of 91.37%, which indicates that our model is also effective in a relatively simple scenario. We

[2]Unless otherwise specified in subsequent experiments, the pre-trained backbone we used is PLATO-2 as it carries out better results.

Model	MultiWOZ 2.2	WOZ 2.0
Neural Belief Tracker	-	84.20
Belief Tracking	-	85.50
GLAD	-	88.10
StateNet	-	88.90
TRADE	45.40†	-
DS-DST	51.70†	-
BERT-DST	-	87.70
SOM-DST	53.81‡	-
COMER	-	88.60
SGD-baseline	42.00†	-
SimpleTOD	54.02‡	-
Seq2Seq-DU	54.40	91.20
AG-DST	**57.26**	**91.37**

Table 2: Joint goal accuracy of AG-DST and baselines on MultiWOZ 2.2 and WOZ 2.0 datasets. AG-DST consistently outperforms all baselines. †: the results borrowed from Zang et al. (2020). ‡: our reproduction results by official code.

Domain	Joint Acc	Slot Acc
multi-domain	57.26	97.48
attraction	89.91	96.26
hotel	82.31	97.16
restaurant	89.00	97.97
taxi	96.09	98.31
train	89.15	97.51

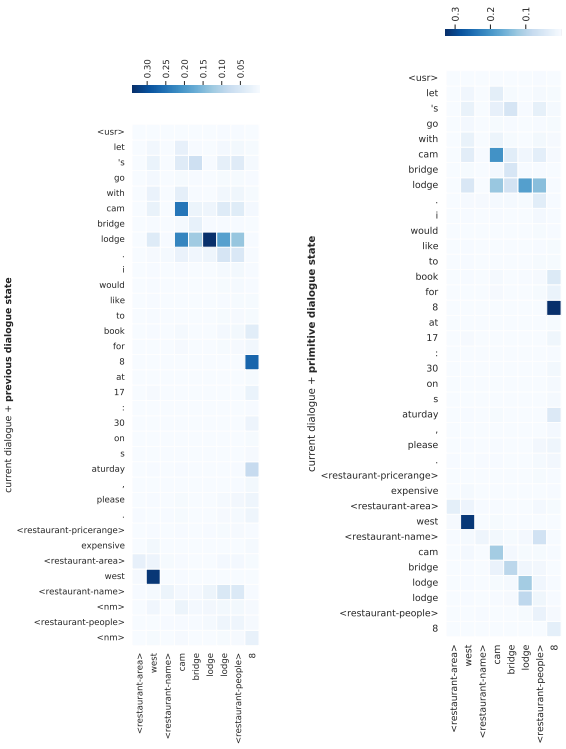
Table 3: Joint goal accuracy and slot accuracy of AG-DST on MultiWOZ 2.2 by domains.

will analyse the strengths of AG-DST in the subsequent section.

4 Analysis

4.1 Amending Generation

As shown in Table 4, we conduct ablation study to investigate the effectiveness of the proposed amending generation. The results show that the amending generation has a significant improvement on the basic generation, where the model gets extra joint goal accuracy of 0.87%. When we perform negative sampling to facilitate effective learning of the amending generation process, it will lead to an increase by 1.06%. Furthermore, we implement heuristic negative sampling which exchanges some correlated slot values (such as *leave at* and *arrive by*, *departure* and *destination*, *area* in different domains) on MultiWOZ 2.2. While using this further



(a) Basic Generation (b) Amending Generation

Figure 4: The attention visualization of (a) basic generation and (b) amending generation on an example from MultiWOZ 2.2. Note that in both figures, the abscissa and ordinate are a fraction of whole input and output sequence respectively because of limited space.

heuristic negative sampling, the model achieves the best joint goal accuracy of 57.35%. Essentially, the amending generation process amends the challenging mistakes that still exists after the basic generation and provides a more accurate prediction of the dialogue state of the current turn, which effectively alleviate the error propagation problem. To the best of our knowledge, AG-DST is the first DST method that involves a two-pass generation process to amend the primitive dialogue state.

Visualization Figure 4 shows the attention visualization of the basic generation and the amending

Model	Joint Acc
basic generation	56.20
+ amending generation	57.07 (+0.87)
+ amending generation w/ NS	57.26 (+1.06)
+ amending generation w/ NS$^+$	57.35 (+1.15)

Table 4: The ablation study of the amending generation on MultiWOZ 2.2 with joint goal accuracy. NS means the negative sampling mentioned in Section 2.2, as well as NS$^+$ performs a heuristic negative sampling.

generation on an example of MultiWOZ 2.2. In the basic generation, the value of slot *restaurant-area* is copied from previous dialogue state and the value of slot *restaurant-people* is predicted from user utterance accurately. However, for the *restaurant-name*, although the model pays attention to the corresponding tokens in the dialogue of the current turn, it generates a wrong slot value. In the amending generation, the *restaurant-name* and its value attend to both the corresponding tokens in the user utterance and slot value in the primitive dialogue state with high weight, which indicates that the amending generation can utilize both the dialogue of the current turn and the primitive dialogue state for reference to correct the mistakes in the primitive dialogue state.

Error Analysis In our analysis, we found three frequent errors in a one-pass generation model of DST. (1) The slot value is not updated, as the model fails to obtain the key information in the dialogue, an example can be seen in Figure 1. (2) Some common mistakes occur in generation model. For example, *cambridge lodge* is predicted to *cambridge lodge lodge*, and *camboats* is predicted to *cambots*. (3) There is confusion between correlated slots, such as *leave at* and *arrive by*, *departure* and *destination*. As reported in Figure 5, we make the statistics of error types by random sampling on MultiWOZ 2.2, which indicates that 51% of the error comes from the first type, 4% comes from the second type, 3% comes from the third type, 36% is due to the inconsistent annotation, and 6% is due to others. Furthermore, we show that the proposed amendable generation can greatly correct these errors and improve the overall performance of DST with concrete examples of the amending generation are shown in Table 9 of Appendix B.

4.2 Effect of Previous Dialogue State

As shown in Table 5, we compare three types of input sequence formations: only dialogue history, dialogue history and previous dialogue state, current turn dialogue and previous dialogue state. We find that using the previous dialogue state (i.e. dialogue state memory) performs better than using only dialogue history, which confirms that the previous dialogue state can be served as a compact representation of the dialogue history and utilizing the previous dialogue state is more effective than performing DST from scratch. The results also show that using the current turn dialogue instead of the whole dialogue history barely hurts the performance, yet it greatly improves the efficiency, which justifies our use of the current turn dialogue and previous dialogue state as the input sequence.

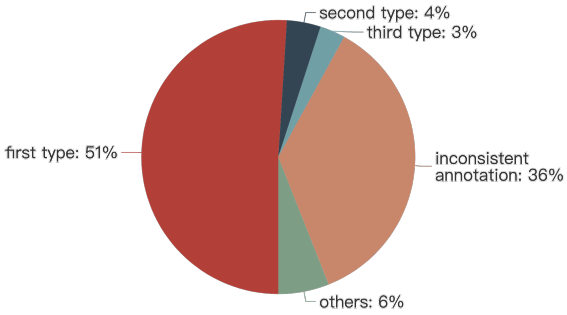


Figure 5: Error type statistics of the basic generation. first type: slot value not updated; second type: common mistakes of generation; third type: confusion between correlated slots.

Structure	Joint Acc
$(B_t \mid D)$ †	54.02
$(B_t \mid D, B_{t-1})$	**56.27**
$(B_t \mid D_t, B_{t-1})$	56.20

Table 5: Joint goal accuracy of different structures of generation model on MultiWOZ 2.2. The structures with dialogue state memory perform better than the rest which uses only dialogue context. †: we reuse the result of SimpleTOD in Table 2.

4.3 Effect of Pre-training

In prior works, GPT-2 is often the pre-trained backbone in generation model (Hosseini-Asl et al., 2020; Ham et al., 2020; Yang et al., 2021). We also analyze the ability of GPT-2 on MultiWOZ 2.2 (see Table 6), where only token embedding and position embedding are added as input embedding. Results show that our approach initialized with GPT-2 surpasses the SimpleTOD (a generation model initialized by GPT-2) and Seq2Seq-DU (the prior state-of-the-art), and obtains similar results

Model	Joint Acc
SimpleTOD	54.02
Seq2Seq-DU	54.40
GPT-2 w/ basic generation	55.24
GPT-2 w/ amending generation	55.85
GPT-2 w/ amending generation + NS	56.09
PLATO-2 w/ basic generation	56.20
PLATO-2 w/ amending generation	57.07
PLATO-2 w/ amending generation + NS	**57.26**

Table 6: The ablation study of the pre-trained model on MultiWOZ 2.2 with joint goal accuracy.

Model	Joint Acc
basic generation	**56.20**
- name special token	55.34 (-0.86)
- utterance special token	55.70 (-0.50)
- DS special token	54.50 (-1.70)
- above three	54.41 (-1.79)

Table 7: The ablation study of the absence of different special tokens on MultiWOZ 2.2 with joint goal accuracy.

with the PLATO-2 initialization. This indicates our approach's ability to universally improve the performances of other pretrained models.

4.4 Effect of Special Tokens

Special tokens are important for identifying different input components (Hosseini-Asl et al., 2020). In our experiments, similar to prior works, we use the utterance special token and dialogue state special token to differentiate each part. Besides, to boost the extraction of entity names on MultiWOZ 2.2, we add special tokens `<name/>` and `</name>` around the candidate entity names. Table 7 shows the joint goal accuracy reduction caused by the absence of various special tokens, which further confirms the neccesity of using special tokens.

5 Related Work

Traditional methods regard DST as a classification task, in which an encoder is employed for obtaining a representation of utterances and a classifier is utilized for predicting a slot value from a pre-defined ontology (Mrkšić et al., 2017; Zhong et al., 2018; Ramadan et al., 2018). However, it is difficult to obtain a full ontology in a real scenario. Some open-vocabulary DST models extract or generate the dialogue state from the dialogue history at each turn (Chao and Lane, 2019; Hosseini-Asl et al., 2020; Ham et al., 2020; Heck et al., 2020). These methods predict the dialogue state from scratch at each turn, which undoubtedly puts a great burden on the model.

To address this problem, some recent methods consider to utilize the previous dialogue state and recent dialogue history as input information. Specifically, the DST is separated into two components, where a component named State Operation Prediction is used to encode utterance and previous dialogue state, as well as predict the operation of state at each turn, and the other component named Value Generation predicts the value for each slot (Kim et al., 2020; Zeng and Nie, 2020). Kim et al. (2020) encodes the dialogue history of the last two turns and the previous dialogue state by BERT and predicts four kinds of state operations: *carryover*, *delete*, *dontcare* and *update*. For *update* operation, a GRU-based value generation is used to decode the slot value. Zeng and Nie (2020) reuses a fraction of the hidden states of the encoder in the decoder to build a flat model for effective parameter updating. Besides, some methods treat dialogue state tracking as a causal language model by using the dialogue of the current turn and previous dialogue state as input sequence (Lin et al., 2020; Yang et al., 2021). Lin et al. (2020) utilizes an encoder-decoder framework to generate dialogue state and system response sequentially, where minimal slot value pairs are generated for efficiently tracking. Yang et al. (2021) models task-oriented dialogs on a dialog session level, which generates dialogue state, dialogue action and system response sequentially based on the whole previous dialogue context, including the generated dialogue states and dialogue actions. Moreover, some schema-guided DST methods leverage schema descriptions to deal with unseen schemas with new domains and slots (Zhu et al., 2020; Feng et al., 2021; Noroozi et al., 2020).

6 Conclusion

In this paper, we propose a novel amendable generative approach for dialogue state tracking, which learns implicit state operation prediction and value generation jointly in a single model to reduce the error propagation. Meanwhile, our model offers an opportunity to amend the primitive dialogue state in the amending generation. Experimental results show that our model outperforms previous works on both MultiWOZ 2.2 and WOZ 2.0 datasets, achieving state-of-the-art performance. The ablation study and attention visualization demonstrate that the proposed amending generation is significantly effective. Moreover, we analyze the types of mistakes that can be resolved by the amending generation and provide some examples to illustrate them in the Appendix. In the future, we will try to integrate schema descriptions into our architecture and explore a generative approach to support end-to-end task-oriented dialogue system.

References

Siqi Bao, Huang He, Fan Wang, Hua Wu, Haifeng Wang, Wenquan Wu, Zhen Guo, Zhibin Liu, and Xinchao Xu. 2021. PLATO-2: Towards building an open-domain chatbot via curriculum learning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2513–2525, Online. Association for Computational Linguistics.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 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*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.

Guan-Lin Chao and Ian Lane. 2019. BERT-DST: Scalable End-to-End Dialogue State Tracking with Bidirectional Encoder Representations from Transformer. In *Proc. Interspeech 2019*, pages 1468–1472.

Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. 2017. A survey on dialogue systems: Recent advances and new frontiers. *Acm Sigkdd Explorations Newsletter*, 19(2):25–35.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13042–13054.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyag Gao, and Dilek Hakkani-Tur. 2019. Multiwoz 2.1: Multi-domain dialogue state corrections and state tracking baselines. *arXiv preprint arXiv:1907.01669*.

Yue Feng, Yang Wang, and Hang Li. 2021. A sequence-to-sequence approach to dialogue state tracking. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1714–1725, Online. Association for Computational Linguistics.

Donghoon Ham, Jeong-Gwan Lee, Youngsoo Jang, and Kee-Eung Kim. 2020. End-to-end neural pipeline for goal-oriented dialogue systems using GPT-2. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 583–592, Online. Association for Computational Linguistics.

Michael Heck, Carel van Niekerk, Nurul Lubis, Christian Geishauser, Hsien-Chin Lin, Marco Moresi, and Milica Gasic. 2020. TripPy: A triple copy strategy for value independent neural dialog state tracking. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 35–44, 1st virtual meeting. Association for Computational Linguistics.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Sungdong Kim, Sohee Yang, Gyuwan Kim, and Sang-Woo Lee. 2020. Efficient dialogue state tracking by selectively overwriting memory. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 567–582, Online. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Zhaojiang Lin, Andrea Madotto, Genta Indra Winata, and Pascale Fung. 2020. MinTL: Minimalist transfer learning for task-oriented dialogue systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3391–3405, Online. Association for Computational Linguistics.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. 2017. Neural belief tracker: Data-driven dialogue state tracking. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1777–1788, Vancouver, Canada. Association for Computational Linguistics.

Vahid Noroozi, Yang Zhang, Evelina Bakhturina, and Tomasz Kornuta. 2020. A fast and robust bert-based dialogue state tracker for schema-guided dialogue dataset. *arXiv preprint arXiv:2008.12335*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Osman Ramadan, Paweł Budzianowski, and Milica Gašić. 2018. Large-scale multi-domain belief tracking with knowledge sharing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*,

pages 432–437, Melbourne, Australia. Association for Computational Linguistics.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8689–8696.

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, Hong Kong, China. Association for Computational Linguistics.

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*, pages 2780–2786, Brussels, Belgium. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 438–449, Valencia, Spain. Association for Computational Linguistics.

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 Annual Meeting of the Association for Computational Linguistics*, pages 808–819, Florence, Italy. Association for Computational Linguistics.

Yunyi Yang, Yunhao Li, and Xiaojun Quan. 2021. Ubar: Towards fully end-to-end task-oriented dialog system with gpt-2. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14230–14238.

Xiaoxue Zang, Abhinav Rastogi, Srinivas Sunkara, Raghav Gupta, Jianguo Zhang, and Jindong Chen. 2020. MultiWOZ 2.2 : A dialogue dataset with additional annotation corrections and state tracking baselines. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 109–117, Online. Association for Computational Linguistics.

Yan Zeng and Jian-Yun Nie. 2020. Multi-domain dialogue state tracking–a purely transformer-based generative approach. *arXiv preprint arXiv:2010.14061*.

Jianguo Zhang, Kazuma Hashimoto, Chien-Sheng Wu, Yao Wang, Philip Yu, Richard Socher, and Caiming Xiong. 2020. Find or classify? dual strategy for slot-value predictions on multi-domain dialog state tracking. In *Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics*, pages 154–167, Barcelona, Spain (Online). Association for Computational Linguistics.

Victor Zhong, Caiming Xiong, and Richard Socher. 2018. Global-locally self-attentive encoder for dialogue state tracking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1458–1467, Melbourne, Australia. Association for Computational Linguistics.

Su Zhu, Jieyu Li, Lu Chen, and Kai Yu. 2020. Efficient context and schema fusion networks for multi-domain dialogue state tracking. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 766–781, Online. Association for Computational Linguistics.

A Error Propagation Analysis

Our model learns simultaneously the implicit state operation prediction and value generation by a single generation model with dialogue state memory. Indeed, such joint learning can boost the performance of the *update* gate. Because the traditional method needs to predict the state operation firstly, and then use another component to predict the corresponding value, which leads to error propagation. The *update* gate accuracy and the value generation F1 under the *update* gate are given in Table 8.[3] Compared with the SOM-DST, our AG-DST attains better results of both gate accuracy and value F1. This implies that the coupling structure we used is beneficial to learning *update* gate and corresponding value prediction.

B Case Study

Table 9 indicates the the amendable ability of the AG-DST. On the test set of MultiWOZ 2.2, the total number of improved examples for the three error types is 313. AG-DST was able to amend the slot value during the amending generation, when the basic generation decoded nothing or a wrong value.

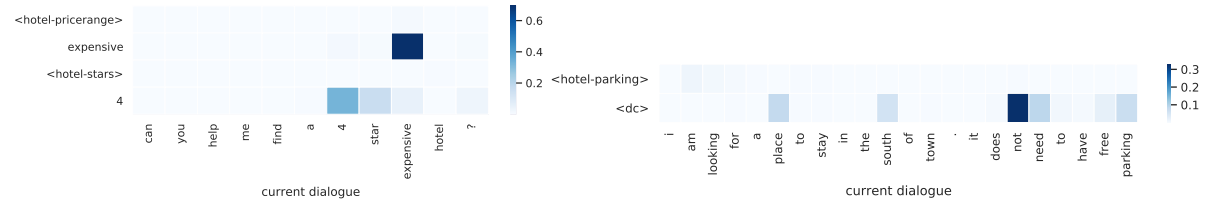
[3]Note that for our model, the state operation can be concluded from the difference between last and current dialogue states.

Model	Acc	F1
SOM-DST	70.50	78.58
AG-DST	**72.77**	**79.84**

Table 8: The comparison of *update* gate accuracy and corresponding value generation F1 between SOM-DST (Kim et al., 2020) and our basic generation on MultiWOZ 2.2.

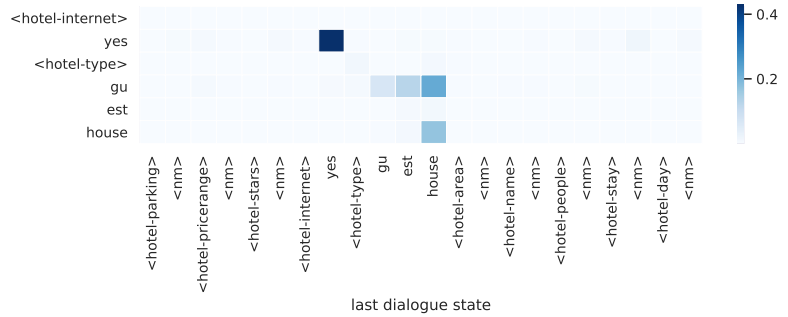
C Attention Visualization of State Operation Prediction

Figure 6 shows the attention visualization of different state operations on test set of MultiWOZ 2.2. From the *update* gate and *dontcare* gate attention visualizations in Figure 6a and 6b, we can easily find that the slot values attend to the corresponding tokens in the utterance with a highest weight. Figure 6c indicates that an implicit copy mechanism is used for the *carryover* gate. Figure 7 shows the attention visualization of *delete* gate and *coreference* phenomenon. For *delete* gate in Figure 7a, it is easily seen that the model can grasp the information in the user utterance to generate appropriate <nm> slot value. As shown in Figure 7b, when the user only said "from the restaurant to the hotel", the model can extract the corresponding restaurant and hotel names from DS memory. This indicates that our model is able to handle the *coreference* operation. Most examples of this *coreference* phenomenon appear in *taxi* domain.



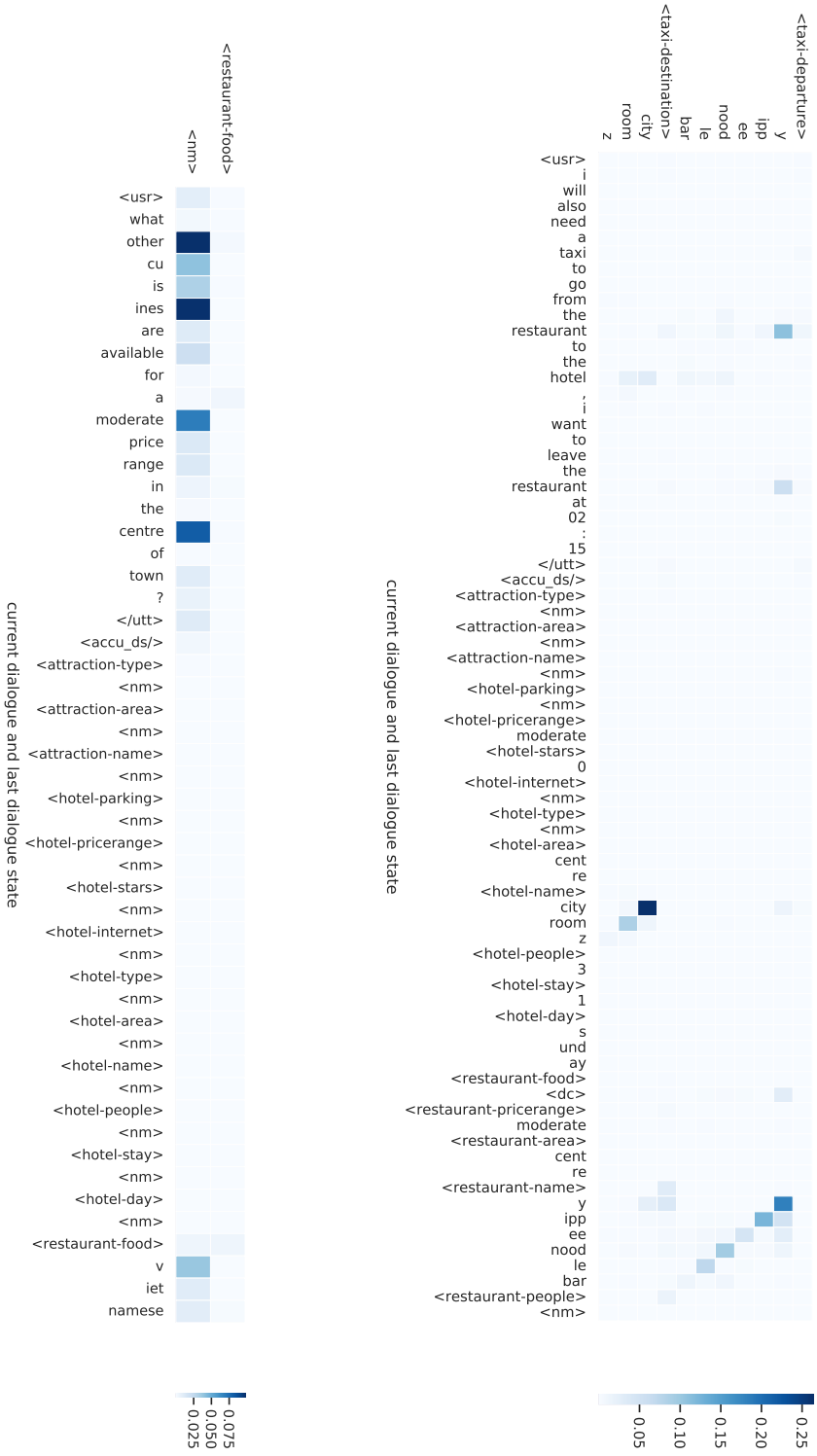
(a) *update* gate

(b) *dontcare* gate



(c) *carryover* gate

Figure 6: The attention visualization of state operation on MultiWOZ 2.2.



(a) *delete* gate (b) *coreference* phenomenon

Figure 7: The attention visualization of *delete* gate and *coreference* phenomenon on MultiWOZ 2.2.

Current Turn Dialogue	**Previous Dialogue State** $\hat{B}_{t-1}$	**Primitive Dialogue State** $\tilde{B}_t$	**Amended Dialogue State** $\hat{B}_t$
R_t: no , i am sorry . i am searching for a 4 star hotel in the centre for 2 nights on saturday . is that correct ? U_t: i need the hotel for *3 nights* starting on saturday please .	hotel-stay-2	hotel-stay-2	hotel-stay-3
R_t: ok , and what day and time would you like that reservation ? U_t: i would like to make a reservation for saturday at 11:45 . and there has been *a change in plans , i will be dining alone* .	restaurant-people-2	restaurant-people-2	restaurant-people-1
R_t: hello . U_t: i am looking for info on expensive *south indian* restaurant -s in cambridge .	restaurant-food-<nm>	restaurant-food-indian	restaurant-food-south indian
R_t: ok , so you would like a taxi from the restaurant to the park ? could you please let me know your desired departure and arrival times ? U_t: i am sorry , i would like a taxi from *wandlebury country park* to taj tandoori . i would like the taxi to pick me up at 10:15 .	taxi-departure-byard art	taxi-departure-byard art	taxi-departure-wandlebury country park
R_t: yes , <name/> *wandlebury country park* </name> is in the south . in order to help you book a taxi between the park and your hotel , i need to know what hotel you are at . U_t: i want a taxi from the restaurant that i am at	taxi-destination-kohinoor	taxi-destination-kohinoor	taxi-destination-wandlebury country park
R_t: i am sorry , i am experiencing a system error . could you please restate your request ? U_t: i want a place to go in the south , *a swimmingpool . or another type of entertainment* , if there is no pool ?	attraction-type-swimmingpool	attraction-type-entertainment	attraction-type-swimmingpool
R_t: <name/> caffe uno </name> is a very nice , expensive italian restaurant in the center of town . would you like a table there ? U_t: actually , *i change my mind* . i think i want to stick with british food after all . can you suggest any 1 thats in the centre of town ?	restaurant-name-clowns	restaurant-name-clowns	restaurant-name-<nm>
R_t: <name/> primavera </name> is a museum in the center of town . their address is 10 king s parade . what else can i help with ? U_t: oh , i made a mistake . i really need that table *for 16:15 , not 17:45* . can you change it , do you thing ?	restaurant-time-17:45	restaurant-time-17:45	restaurant-time-16:15
R_t: there are 5 museums in the west . i recommend *kettle's yard* . would you like the address and phone ? U_t: yes , i would love the address . thank you so much !	attraction-name-cambridge punter	attraction-name-cambridge punter	attraction-name-kettles yard
R_t: you would like a second hotel , for which night and location ? U_t: i am sorry , i do not want a second hotel . i need a taxi between *the bed and breakfast* and the restaurant that arrives to the restaurant by the booked time .	taxi-departure-<nm>	taxi-departure-<nm>	taxi-departure-alexander bed and breakfast
R_t: to clarify you wanted me to book a table on friday ? U_t: i am sorry . my train was supposed to be for *tuesday* as well as my restaurant reservation .	train-day-<nm>	train-day-<nm>	train-day-tuesday
R_t: there are 5 choices . to narrow down you need to choose the side of town you prefer U_t: *area does not matter* , but it should be a guest house with a 4 star rating .	hotel-area-<nm>	hotel-area-<nm>	hotel-area-<dc>
R_t: the entrance fee is unknown . can i help you with anything else today ? U_t: i would also like a taxi to *arrive at the restaurant* on time . can i have the taxis phone number and vehicle type ?	taxi-destination-<nm>	taxi-destination-tajione restaurant and coffee bar	taxi-destination-stazione restaurant and coffee bar
R_t: okay ! they are at king's parade , cb21rl . their phone number is 01223338300 . what time would you like to depart in the taxi ? U_t: i need a taxi to *commute between the 2 place -s . i need to leave the restaurant* by 18:15 and need the contact # and car type	taxi-destination-<nm>	taxi-destination-saint catharine's college	taxi-destination-saint catharines college
R_t: i have got you a reservation for 6 at *hobson's house* for 2 nights . your reference number is 4wngilmf . U_t: thank you so much ! that should be all i need .	hotel-name-<nm>	hotel-name-hobson's house	hotel-name-hobsons house
R_t: is there anything i can do for you ? U_t: i would like a taxi to the *cafe jello gallery* please .	taxi-destination-<nm>	taxi-destination-caffe jello gallery	taxi-destination-cafe jello gallery
R_t: <name/> *camboats* </name> is located at the plough , green end , fen ditton , in the east side of town . U_t: ok ! thank you ! can you get me a taxi to cambots ?	taxi-destination-<nm>	taxi-destination-cambots	taxi-destination-camboats
R_t: the postcode is cb28rj for <name/> *bridge guest house* </name> . what time do you want to be *picked up at the guest house* or to arrive at eraina ? U_t: okay i need a taxi too .	taxi-destination-eraina	taxi-destination-eraina	taxi-destination-bridge guest house

Table 9: Examples of the amending generation. The key information in the dialogue is shown in blue.