Toward Real-life Dialogue State Tracking Involving Negative Feedback Utterances

Puhai Yang
School of Computer Science and Technology,
Beijing Institute of Technology
Beijing, China
Beijing Engineering Research Center of High Volume Language Information Processing and Cloud Computing Applications
Beijing, China
phyang@bit.edu.cn

Heyan Huang*
School of Computer Science and Technology,
Beijing Institute of Technology
Beijing, China
Beijing Engineering Research Center of High Volume Language Information Processing and Cloud Computing Applications
Beijing, China
hhy63@bit.edu.cn

Wei Wei
School of Computer Science and Technology,
Huazhong University of Science and Technology
Wuhan, China
weiw@hust.edu.cn

Xian-Ling Mao
School of Computer Science and Technology,
Beijing Institute of Technology
Beijing, China
maoxl@bit.edu.cn

ABSTRACT

Recently, the research of dialogue systems has been widely concerned, especially task-oriented dialogue systems, which have received increased attention due to their wide application prospect. As a core component, dialogue state tracking (DST) plays a key role in task-oriented dialogue systems, and its function is to parse natural language dialogues into dialogue state formed by slot-value pairs. It is well known that dialogue state tracking has been well studied and explored on current benchmark datasets such as the MultiWOZ. However, almost all current research completely ignores the user negative feedback utterances that exist in real-life conversations when a system error occurs, which often contains user-provided corrective information for the system error. Obviously, user negative feedback utterances can be used to correct the inevitable errors in automatic speech recognition and model generalization. Thus, in this paper, we will explore the role of negative feedback utterances in dialogue state tracking in detail through simulated negative feedback utterances. Specifically, due to the lack of dataset involving negative feedback utterances, first, we have to define the schema of user negative feedback utterances and propose a joint modeling method for feedback utterance generation and filtering. Then, we explore three aspects of interaction mechanism that should be considered in real-life conversations involving negative feedback utterances and propose evaluation metrics related to negative feedback utterances. Finally, on WOZ2.0 and MultiWOZ2.1 datasets, by constructing simulated negative feedback utterances in training and testing, we not only verify the important role of negative feedback utterances in dialogue state tracking, but also analyze the advantages and disadvantages of different interaction mechanisms involving negative feedback utterances, lighting future research on negative feedback utterances.

CCS CONCEPTS

• **Computing methodologies** → Discourse, dialogue and pragmatics.

KEYWORDS

Real-life dialogue, Dialogue state tracking, Negative feedback

ACM Reference Format:
Puhai Yang, Heyan Huang, Wei Wei, and Xian-Ling Mao. 2022. Toward Real-life Dialogue State Tracking Involving Negative Feedback Utterances. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22), August 14–18, 2022, Washington, DC, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3534678.3539385

1 INTRODUCTION

Currently, task-oriented dialogue systems [14] are receiving increasing attention, which can help users complete specific tasks, such as ordering food, booking hotel, etc. Dialogue state tracking (DST) [2, 8, 21] is the most core component of task-oriented dialogue system, its purpose is to parse the natural language multi-turn dialogues into the dialogue state in the form of slot-value pairs, which can be used in the subsequent dialogue policy learning module to predict the next system action.

It is well known that dialogue state tracking has been well studied on current benchmark datasets. In recent years, with the release of large multi-domain task-oriented dialogue datasets [1, 16, 27], especially since MultiWOZ series [4, 5, 23] was proposed, the research and exploration of dialogue state tracking have developed rapidly. For example, the end-to-end architecture overcomes the error propagation in traditional pipeline modeling [2, 10, 22], the

*Corresponding author

KDD '22, August 14–18, 2022, Washington, DC, USA.

ACM ISBN 978-1-4503-9385-0/22/08…$15.00
https://doi.org/10.1145/3534678.3539385

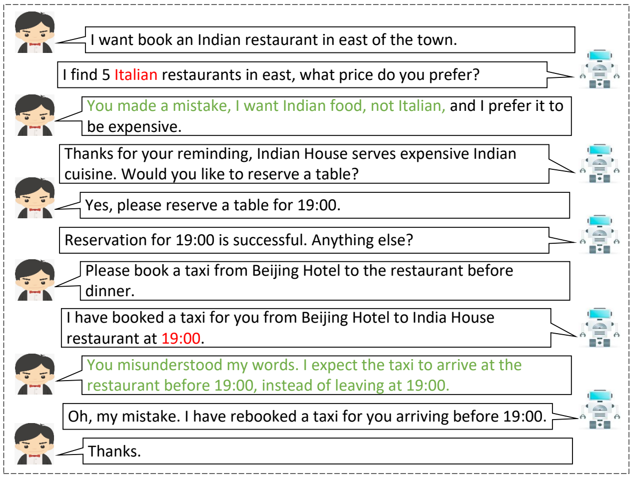


Figure 1: An example of a real-life conversation involving negative feedback utterances. System errors and user negative feedback utterances have been color-coded.

generative and extractive models free dialogue state tracking from the classification model's constraints on ontology [6, 8, 21], and the use of BERT, GPT and T5 language models improves the natural language understanding of dialogue state tracking model [11, 12].

However, almost all current research completely ignores the user negative feedback utterances that exist in real-life conversations when a system error occurs, which often contains user-provided corrective information for the system error. In a real-life conversation scenario, when a user observes a system error, the user will usually remind the system and provide the system with some words to correct the error. For example, as shown in Figure 1, at the beginning of the dialogue, due to the error of automatic speech recognition, the system misunderstands Indian food as Italian food. When the user observes the error of the food mentioned by the system, the user instinctively reminds and corrects the error. At the end of the dialogue, the system again misinterprets the time of arrival as the time of departure, and again the user corrects it.

Obviously, user negative feedback utterances can be used to correct the errors in the dialogue state. Due to the limitation of automatic speech recognition and model generalization, it is inevitable for a dialogue system to make mistakes in real-life conversation [7, 11, 20], and thus the ability of understanding negative feedback utterances for a dialogue state tracking model should be considered. In real-life conversations, when a dialogue system detects a user negative feedback utterance, it can use the negative feedback utterance to correct the errors in the dialogue state. Meanwhile, the system response can also be adjusted in time to improve the friendliness of system according to the negative feedback utterance. For example, as shown in Figure 1, the system uses the user negative feedback utterances to correct the food and arrival time in the dialogue state, and inserts thanking phrases such as "thanks for your reminding" and "Oh, my mistake" in subsequent replies.

Thus, in this paper, we will explore the role of negative feedback utterances in dialogue state tracking in detail through simulated negative feedback utterances. Specifically, due to the lack of dataset involving negative feedback utterances, first, we define user negative feedback utterance as the combination of negative emotional phrase and feedback utterance, and propose a T5-based feedback utterance generation and filtering joint model. Then, we explore the feedback acquisition strategy, the feedback acquisition content, and the feedback acquisition timing in the interaction mechanism involving negative feedback utterances, and propose two evaluation metrics related to negative feedback utterances: feedback cost and feedback accuracy. Finally, on WOZ2.0 [20] and MultiWOZ2.1 [4] datasets, by constructing simulated negative feedback utterances in training and testing, we analyze and verify the importance of negative feedback utterances in dialogue state tracking and study the performance of dialogue state tracking model under different interaction mechanisms. The code and data are publicly released[1].

In summary, the contributions of our paper are:

- To the best of our knowledge, our paper is the first work to study user negative feedback utterance in task-oriented dialogue system, which shows a new research direction for dialogue systems.
- We define the schema of negative feedback utterances and propose models and algorithms for creating and using simulated negative feedback utterances on current benchmark datasets.
- We explore human-computer interaction mechanisms involving negative feedback utterances in real-life conversations and propose two evaluation metrics related to negative feedback utterances.

The rest of this paper is organized as follows: The definition and modeling of negative feedback utterances are introduced in section 2. Section 3 lists three important aspects of interaction mechanisms involving negative feedback utterances. Section 4 is about experiment settings and evaluation metrics. Experimental results and analysis are elaborated in Section 5, after which we conclude in Section 6. Besides, limited by space, related work is discussed in Appendix A.

2 DEFINITION AND MODELING OF NEGATIVE FEEDBACK UTTERANCES

In real-life dialogues, the negative feedback utterance from users is very important to a dialogue system, which is related to whether the system errors can be corrected and the friendliness of the dialogue system can be improved. In this section, we will first define the schema of negative feedback utterances, and then obtain negative feedback utterances through joint modeling of feedback utterance generation and filtering.

2.1 Definition of Negative Feedback Utterances

Generally, in real-life conversations, when users observe errors in a system, they will warn the system first, and then explain the specific content of the errors. Therefore, in this paper, we divide a negative feedback utterance into two parts: a negative emotional phrase used to warn errors and a feedback utterance used to describe the specific content of errors, as shown in Figure 2.

[1]https://github.com/yangpuhai/Feedback-involved-DST

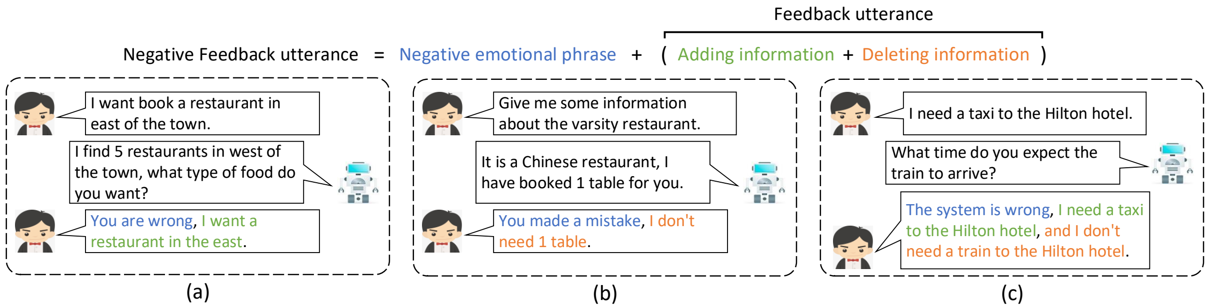


Figure 2: Structure of negative feedback utterances defined in the paper and three different examples. In examples, blue, green, and yellow texts represent negative emotional phrases, adding information, and deleting information, respectively.

Table 1: Templates of negative emotional phrases.

train	you made a mistake, you are wrong, the system is wrong, you misunderstand what i mean, there are bugs in the system
dev	you made a mistake, you are wrong, the system is wrong, you misunderstand, there's something wrong with the system
test	you made a mistake, you got it wrong, the system is wrong, something went wrong, i am afraid you are wrong

2.1.1 Negative Emotional Phrase. As the beginning mark of a negative feedback utterance, a negative emotional phrase plays a guiding role. Since none of the current dialogue datasets contain data related to negative emotional phrases, we use templates to construct negative emotional phrases. We design 10 negative emotional phrase templates and use them for training, development, and testing. as shown in Table 1.

2.1.2 Feedback Utterance. Feedback utterance is the main part of a user negative feedback utterance, which contains user's detailed feedback on system errors. In this paper, we further divide a feedback utterance into two parts: adding information and deleting information, as follows:

(1) Adding information: When a slot value that a user has provided to the system is misunderstood by the system, the user's feedback utterance will repeat the information about the slot value again, which is called adding information in this paper. As shown in Figure 2 (a), the system misunderstands the restaurant area provided by the user, so the user repeats the area.
(2) Deleting information: For a slot that a user has not mentioned in the dialogue, when its value is generated redundantly by the system, the user's feedback utterance will negate the information of the slot value, which is called deleting information in this paper. As shown in Figure 2 (b), the system mistakenly book a table for the user, and the user points out the error in subsequent feedback.

A typical example about user negative feedback utterances is shown in Figure 2 (c). The system misunderstands the taxi destination provided by the user as the train destination, which not only loses the information of taxi destination but also predicts the information of train destination redundantly. Therefore, the user repeats the taxi destination and denies the train destination in the feedback utterance.

2.2 Joint Modeling of Feedback Utterance Generation and Filtering

In this paper, except for one more negative auxiliary verb, the sentence structure of deleting information is defined as exactly the same as that of adding information, as shown in Figure 2, Therefore, for simplification, we only modeling the feedback utterance corresponding to adding information. Meanwhile, considering the diversity of natural language generation, the modeling of feedback utterance turns to how to generate user utterance candidates from slot value information and how to further filter these candidates.

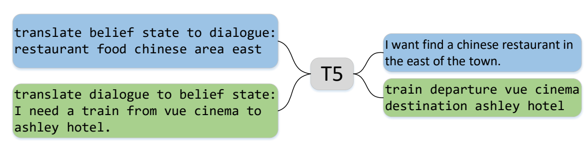


Figure 3: High-level description for joint modeling of feedback utterance generation (blue) and filtering (green).

2.2.1 Design of the Joint Model. In this paper, the generation and filtering method of user utterances is similar to CoCo-DST [11] which uses system response and dialogue state to generate a set of user utterance candidates and then uses a heuristic method and a BERT-based classifier to filter the generated candidates. More efficiently and adaptively, we use a single T5 model to jointly model the generation and filtering of user utterances, as shown in Figure 3. Given a dialogue $X = \{(S_1, U_1), (S_2, U_2), ..., (S_t, U_t)\}$ containing t turns, where S_i and U_i $(1 \leq i \leq t)$ denote system and user utterance at the i-th turn, respectively. The corresponding dialogue state of this dialogue is $Y = \{(L_1, B_1), (L_2, B_2), ..., (L_t, B_t)\}$, where L_i and B_i represent the dialogue state of turn-level and dialogue-level respectively, that is, L_i is the slot value information of the i-th turn and B_i is the accumulated state from the first turn to the i-th turn. In our model, the input sequence of the user utterance generator is $Prompt_1 \oplus L_i$, where $Prompt_1$ is "translate belief state to dialogue:", then, the sequence is used as the input to the encoder of T5, and the decoder generates the corresponding user utterance U_i:

$$U_i = T5(Prompt_1 \oplus L_i) \tag{1}$$

In contrast, the input sequence of the user utterance filter is $Prompt_2 \oplus U_i$, where $Prompt_2$ is "translate dialogue to belief state:", the input sequence then passes through T5 to generate the corresponding dialogue state L_i:

$$L_i = T5(Prompt_2 \oplus U_i) \tag{2}$$

The objective function of the model is to minimize the sum of the negative log-likelihood of U_i for given L_i and the negative log-likelihood of L_i for given U_i:

$$Loss = -\sum_{i=1}^{t}(log\ P(U_i|Prompt_1, L_i) + log\ P(L_i|Prompt_2, U_i)) \tag{3}$$

2.2.2 Application of the Joint Model in Inference. For a well-trained dialogue state tracking model, its ability of parsing user negative feedback utterances reflects its adaptability to real-life dialogue scenarios, here, we describe how to test the dialogue state tracking model with our feedback utterance generation and filtering joint model, as shown in Algorithm 1 in Appendix B. First, according to the ground truth of the dialogue state, the adding information and deleting information are analyzed from the dialogue state predicted by the dialogue state tracking model. Then, with the ability of our model, feedback utterances are generated and filtered, and combined with negative emotional phrases to obtain negative feedback utterances. Finally, the dialogue history with negative feedback utterances is inputted into the dialogue state tracking model to obtain the new dialogue state.

2.2.3 Utilization of the Joint Model in Training. All current dialogue state tracking datasets do not involve user negative feedback utterances, which makes it difficult for dialogue state tracking models to parse negative feedback utterances. Therefore, how to add negative feedback utterances to the train set is a very important problem. In this paragraph, we will elaborate on the process of the construction of the train set involving negative feedback utterances on the basis of existing dataset. The detailed steps are shown in Algorithm 2 in Appendix B, first, the adding information and the deleting information are randomly selected from the dialogue state, and the slot value in the adding information is randomly replaced with the value in the ontology. Then, feedback utterances are constructed with the help of our joint modeling model of feedback utterance generation and filtering. Finally, new user utterance and new dialogue state are generated by adding information, deleting information, and their corresponding feedback utterances.

3 INTERACTION MECHANISM INVOLVING NEGATIVE FEEDBACK UTTERANCES

In a real-life scenario, negative feedback utterances will greatly affect the interaction mechanism between the user and the system. In this section, we explore three important aspects that should be considered in the interaction mechanism involving negative feedback utterances: (1) How to obtain user negative feedback utterances; (2) What information to obtain from user negative feedback utterances; (3) When to obtain user negative feedback utterances. These three aspects can be taken in short as feedback acquisition strategy, feedback acquisition content, and feedback acquisition timing respectively, which are introduced in detail as follows:

3.1 Feedback Acquisition Strategy

The key to solving the problem of "how to obtain user negative feedback utterances?" lies in how to present the dialogue state information to users. In this paper, we divide the presentation of the dialogue state information into implicit and explicit strategies, as shown in Figure 4. The implicit strategy is to include the dialogue state information in the system utterance, while the explicit strategy is to display the dialogue state information by means of an external display device.

3.1.1 Implicit Strategy. Without the need for an external display device, the dialogue in implicit strategy appears more natural and anthropomorphic. As shown in Figure 4 (a), the system presents the hotel area information in the system utterance to users. If negative feedback is obtained from users, the dialogue state can be modified.

3.1.2 Explicit Strategy. With the support of an external display device, the system can present richer information to users and improve interaction efficiency. As shown in Figure 4 (b), after the system shows the dialogue state or state description to the user, the user will give feedback immediately, and this process can be iterative. Before the next system utterance, the dialogue state can be fully verified by using the user's negative feedback.

3.2 Feedback Acquisition Content

Obviously, what information the system wants to obtain from the user negative feedback depends on what information the system presents to the user. Since the dialogue state can be divided into turn-level and dialogue-level, the feedback acquisition content is also divided into turn-level content and dialogue-level content. As shown in Figure 5, when the turn-level content needs to be acquired, the system only presents the dialogue state predicted from the user's last utterance; meanwhile, when the dialogue-level content is acquired, the dialogue-level dialogue state is presented.

3.2.1 Turn-level Content. When obtaining the turn-level content, the less slot value information in the turn-level dialogue state can be very succinctly integrated into the system utterance. As shown in Figure 5 (a), the system includes the food and area information predicted in the previous turn in the system utterance, and the negative feedback utterance from the user can conveniently complete the correction of the errors in the previous turn-level dialogue state.

3.2.2 Dialogue-level Content. In the process of obtaining dialogue-level content, the presentation of the dialogue-level dialogue state brings some redundant information but enables users to have a more detailed understanding of the entire conversation. As shown in Figure 5 (b), in addition to correcting the area information based on the user's negative feedback, the correctly predicted food information will still be displayed in the subsequent system utterance.

3.3 Feedback Acquisition Timing

The time when the system gets negative feedback from the user depends on the time when the system presents the information to

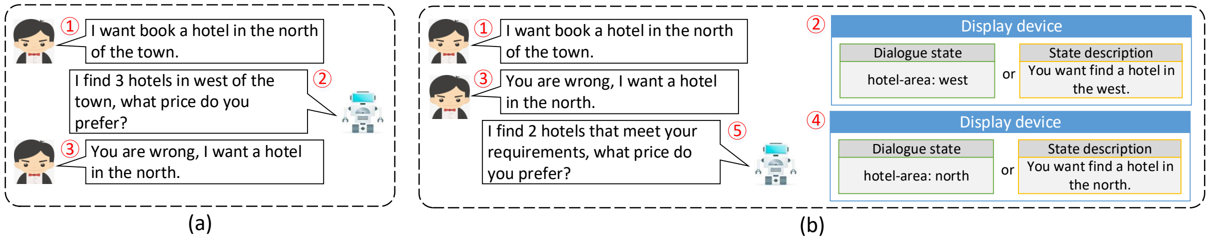


Figure 4: Feedback acquisition strategy in interaction mechanism involving negative feedback utterances. (a) is the implicit strategy, and (b) is the explicit strategy. Numbers with red circles indicate the order of conversations.

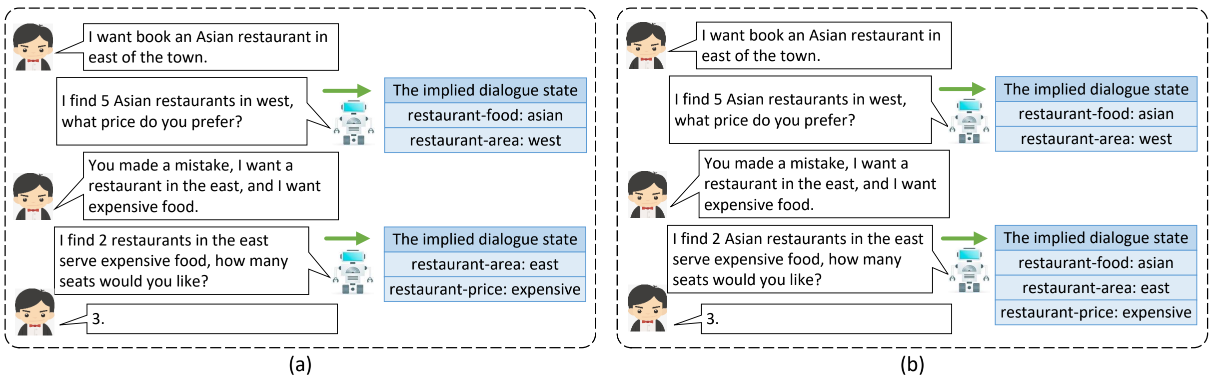


Figure 5: Feedback acquisition content in interaction mechanism involving negative feedback utterances. (a) indicates that the system shows the turn-level dialogue state, while (b) indicates that the system shows the dialogue-level dialogue state.

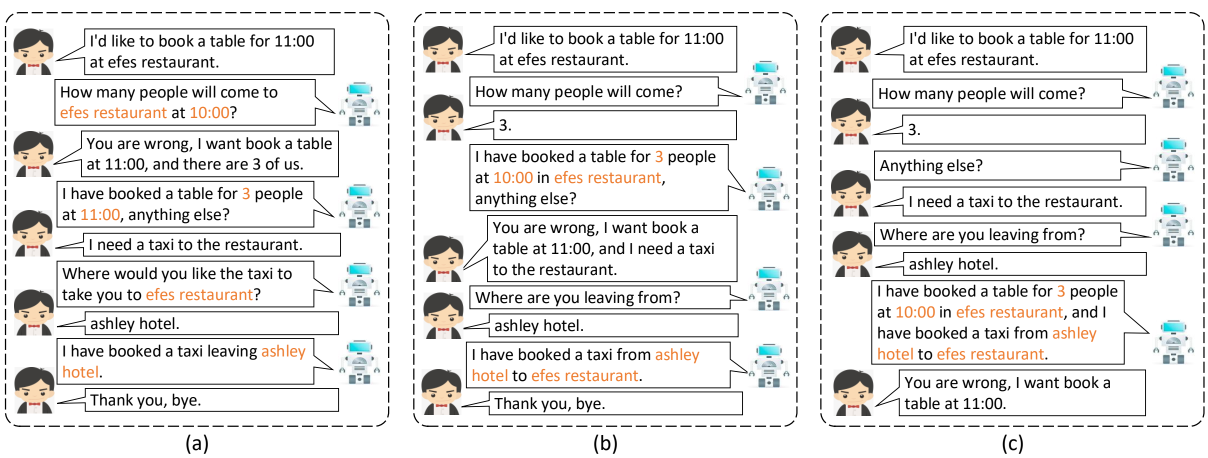


Figure 6: Feedback acquisition timing in interaction mechanism involving negative feedback utterances. (a), (b), and (c) mean that the system displays the dialogue state at the end of each turn, each task, and the entire session respectively. The yellow text indicates the slot value in the dialogue state displayed by the system.

the user. According to the real-life situation of multi-domain multi-turn dialogue state tracking, we divide the feedback acquisition timing into three levels, as shown in Figure 6, which are as follows:

3.3.1 End of Turn. In this case, the system displays the state information to the user after each turn of dialogue, and the system can obtain timely correction information due to frequent feedback from the user, as shown in Figure 6 (a).

3.3.2 End of Task. In this case, the system only presents the dialogue state information to the user at the end of each task, protecting the user from the enormous effort of frequent feedback, as shown in Figure 6 (b).

Table 2: Statistics on the characteristics of the 4 baselines. In state update, scratch-based means that a new slot value is directly predicted for each slot in each turn, and previous-based means that updates are made based on the previous dialogue state.

Models	backbone	Decoder	Input	State Update
TRADE	RNN	Generative + Classifier	all dialogue	scratch-based
BERTDST	BERT	Extractive + Classifier	turn dialogue	previous-based
SOMDST	BERT	Generative + Classifier	turn dialogue + previous state	previous-based
T5DST	T5	Generative	all dialogue + slot information	scratch-based

3.3.3 End of Session. In this case, the system presents the dialogue state information of the entire conversation to the user in the form of a dialogue summary, which can further reduce the feedback pressure of the user, as shown in Figure 6 (c).

3.4 Negative Feedback Evaluation

After the introduction of user negative feedback utterances, it is particularly important to evaluate the ability of dealing with negative feedback utterances for dialogue state tracking models. In this paper, we propose two evaluation criteria of negative feedback: feedback cost and feedback accuracy. Feedback cost is used to measure the total effort of users when they give negative feedback in conversation, while feedback accuracy directly reflects the ability of parsing negative feedback for dialogue state tracking models.

3.4.1 Feedback Cost. We define the feedback cost as the ratio of slots in the user's negative feedback utterance to slots in the dialogue-level dialogue state. On the one hand, the stronger the ability of predicting slot values in traditional user utterance for dialogue state tracking model, the fewer slot values in negative feedback, and the lower the feedback cost. On the other hand, the stronger the ability of dealing with the negative feedback utterance for dialogue state tracking model, the fewer negative feedback in the subsequent dialogue, and the lower the feedback cost.

3.4.2 Feedback Accuracy. In this paper, feedback accuracy refers to the proportion of slots correctly predicted from the negative feedback utterance to all slots in the negative feedback utterance. This evaluation criterion directly reflects the ability of parsing negative feedback utterances for dialogue state tracking models.

4 EXPERIMENTAL SETTINGS

4.1 Datasets

To fully consider the different situations of single-domain and multi-domain datasets, we conduct experiments on two benchmark datasets: WOZ2.0 [20] and MultiWOZ2.1 [4].

The WOZ2.0 dataset is a multi-turn dialogue dataset for the restaurant domain that contains three slots: food, location, and price range. Its train, development, and test sets contain 600, 200, and 400 dialogues respectively, and it provides automatic speech recognition (ASR) hypotheses of user utterances that can be used to assess the robustness of dialogue state tracking against ASR errors.

The MultiWOZ 2.1 dataset is the most widely known modified version of the MultiWOZ dataset and has been used as a benchmark for many previous studies. It is a multi-turn dialogue dataset with over 30 slots covering 7 domains, which uses 8,438, 1000, and 999 dialogues for training, development, and testing, respectively. Following previous works [6, 8, 21], our experiments are carried out in five domains: train, attraction, restaurant, hotel, and taxi.

4.2 Baselines

We use 4 different types of dialogue state tracking models as baselines, and their characteristics are shown in Table 2.

TRADE [21] is the first multi-domain dialogue state tracking model evaluated on the MultiWOZ dataset. It uses bidirectional RNN to encode the entire dialogue history and generates the slot value of each slot in a pointer-generator network [17].

BERTDST [2] predicts the turn-level dialogue state and then updates the previous dialogue state using a rules-based mechanism. It uses BERT to encode the current turn dialogue and then extracts slot values in a predictive span after sorting slot value types.

SOMDST [8] treats the dialogue state as memory and overwrites the slot values that need to be updated in each turn. It inputs the previous dialogue state and the current turn dialogue into BERT and generates slot values using the pointer-generator network after predicting whether slot values need to be updated.

T5DST [12] models dialogue state tracking as a text-to-text task. For each slot, it concatenates the entire dialogue history and slot information into the T5 model, and the output is the slot value.

4.3 Configurations

Our deployment uses the official source code for SOMDST[2] and T5DST[3], while TRADE and BERTDST are reproduced by ourselves in this paper. In all experiments, for the backbone, RNN uses GRU [3] with a hidden size of 400, BERT uses pre-trained BERT-base [19], and T5 uses pre-trained T5-small [15]. In the training phase, all models are customized with the maximum epoch and early stop on the development set, and the same model configuration is used for comparison in the same dialogue state tracking system. Detailed experimental parameters are given in Appendix C.

4.4 Metrics

For the evaluation of dialogue state tracking, in addition to using joint accuracy (JA) following previous works [6, 8, 21], we also consider task joint accuracy (TJA) and session joint accuracy (SJA). In this paper, joint accuracy, task joint accuracy, and session joint accuracy are used to check whether all predicted slot values are exactly the same as the ground truth slot values at the end of each turn, each task, and each session respectively.

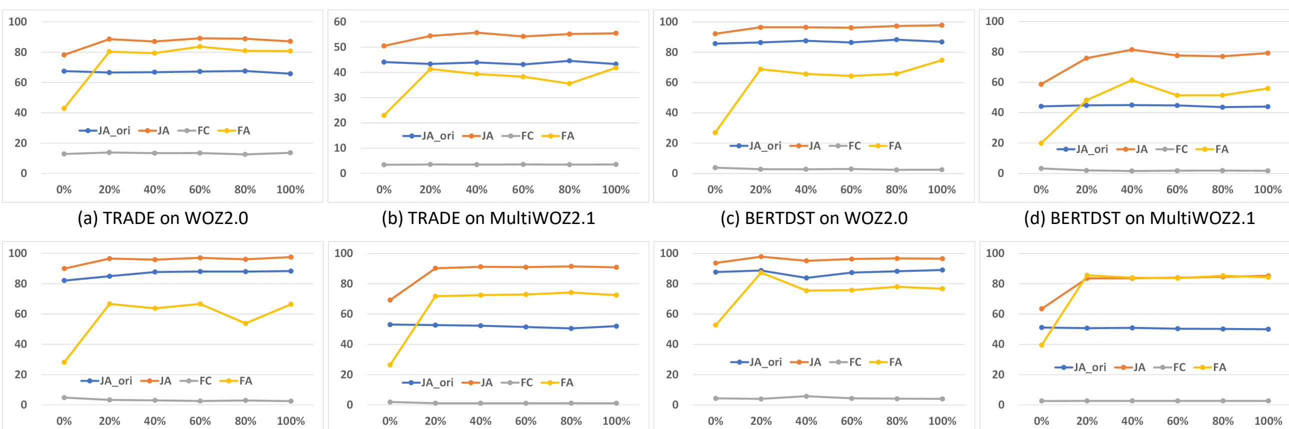
To evaluate the ability of dealing with negative feedback utterances for dialogue state tracking models, we use the feedback cost

[2]https://github.com/clovaai/som-dst

[3]https://github.com/facebookresearch/Zero-Shot-DST

Table 3: Performance of all baselines when negative feedback utterance is considered during training or testing on WOZ2.0 and MultiWOZ2.1. FC-add and FC-del refer to the feedback cost of slots in adding information and deleting information respectively, while FA-add and FA-del refer to the feedback accuracy of slots in adding information and deleting information respectively.

Models	Negetive Feedback		WOZ2.0					MultiWOZ2.1				
	train	test	JA	FC-add	FC-del	FA-add	FA-del	JA	FC-add	FC-del	FA-add	FA-del
TRADE	×	×	67.50	-	-	-	-	44.10	-	-	-	-
	×	✓	78.25	10.04	2.79	54.03	2.90	50.49	2.30	1.10	29.78	8.76
	✓	×	65.80	-	-	-	-	43.30	-	-	-	-
	✓	✓	87.18	11.44	2.17	82.83	70.09	55.52	2.35	1.20	43.42	38.68
BERTDST	×	×	85.72	-	-	-	-	44.14	-	-	-	-
	×	✓	92.22	3.18	0.65	31.21	6.25	58.73	2.68	0.58	22.10	9.66
	✓	×	86.88	-	-	-	-	43.96	-	-	-	-
	✓	✓	97.87	2.11	0.38	80.77	42.11	79.22	1.23	0.48	61.08	42.65
SOMDST	×	×	82.08	-	-	-	-	53.07	-	-	-	-
	×	✓	89.98	3.75	1.07	35.14	3.77	69.27	1.01	0.95	47.73	4.08
	✓	×	88.34	-	-	-	-	52.00	-	-	-	-
	✓	✓	97.57	1.98	0.49	69.39	54.17	90.85	0.77	0.36	65.18	88.11
T5DST	×	×	87.73	-	-	-	-	51.13	-	-	-	-
	×	✓	93.74	3.26	1.11	70.19	1.82	63.45	1.54	1.08	64.91	3.51
	✓	×	89.13	-	-	-	-	49.97	-	-	-	-
	✓	✓	96.60	2.96	1.05	82.19	61.54	85.19	1.57	1.11	81.92	87.60



(a) TRADE on WOZ2.0 (b) TRADE on MultiWOZ2.1 (c) BERTDST on WOZ2.0 (d) BERTDST on MultiWOZ2.1

(e) SOMDST on WOZ2.0 (f) SOMDST on MultiWOZ2.1 (g) T5DST on WOZ2.0 (h) T5DST on MultiWOZ2.1

Figure 7: Performance of all baselines with different proportions of negative feedback training data. JA_ori and JA respectively represent the joint accuracy without and with negative feedback utterance in inference.

(FC) and feedback accuracy (FA) proposed in this paper. In order to make a more detailed comparison, we calculate the feedback cost and feedback accuracy of the slot values in the adding information and deleting information respectively.

5 EXPERIMENTAL RESULTS AND ANALYSIS

5.1 Main Results

Table 3 shows the performance of all baselines when negative feedback utterances are considered during training or testing on WOZ2.0 and MultiWOZ2.1, it can be observed that:

- In the testing phase, the performance of the dialogue state tracking model will be greatly improved if it can make use of user negative feedback utterances. For example, the joint accuracy of SOMDST on WOZ2.0 and MultiWOZ2.1 datasets is improved to 89.98% and 69.27%, respectively.
- Most of the slots mentioned in negative feedback utterances belong to adding information, and the model trained without negative feedback samples can parse the adding information to some extent, while it is almost impossible to predict the slot value in deleting information. For example, without negative feedback training samples, the FA-add of T5DST on

Table 4: Performance of all baselines when different interaction mechanisms are adopted. FAS, FAC and FAT represent feedback acquisition strategy, feedback acquisition content and feedback acquisition timing respectively.

Models	FAS	FAC	FAT	WOZ2.0					MultiWOZ2.1				
				JA	TJA	SJA	FC	FA	JA	TJA	SJA	FC	FA
TRADE	explicit	dialogue-level	turn	87.18	88.00	88.00	13.61	80.80	55.52	48.07	38.14	3.55	41.82
	implicit	dialogue-level	turn	71.26	70.00	70.00	8.49	74.46	44.41	35.30	27.03	2.83	38.35
	explicit	turn-level	turn	78.74	77.25	77.25	8.16	88.34	49.69	40.96	31.33	1.41	55.72
	explicit	dialogue-level	task	72.96	87.75	87.75	4.11	82.27	47.26	48.52	38.34	1.15	42.41
	explicit	dialogue-level	session	72.96	87.75	87.75	4.11	82.27	45.20	40.91	38.84	0.71	41.92
BERTDST	explicit	dialogue-level	turn	97.87	98.00	98.00	2.49	74.80	79.22	77.52	74.87	1.71	55.93
	implicit	dialogue-level	turn	92.83	95.50	95.50	2.25	65.77	57.19	55.53	54.65	1.88	37.21
	explicit	turn-level	turn	92.22	90.00	90.00	0.99	87.76	54.72	46.82	36.94	0.42	88.64
	explicit	dialogue-level	task	89.73	96.25	96.25	1.40	76.81	56.27	63.65	56.66	0.94	56.85
	explicit	dialogue-level	session	89.73	96.25	96.25	1.40	76.81	47.24	47.92	49.75	0.75	55.56
SOMDST	explicit	dialogue-level	turn	97.57	98.25	98.25	2.47	66.39	90.85	90.09	89.29	1.13	72.51
	implicit	dialogue-level	turn	93.56	96.00	96.00	2.07	60.78	73.83	75.81	83.28	1.11	62.95
	explicit	turn-level	turn	91.25	88.75	88.75	0.71	74.29	65.68	59.64	51.65	0.39	92.19
	explicit	dialogue-level	task	91.19	97.50	97.50	1.19	83.05	71.51	86.58	85.79	0.64	78.85
	explicit	dialogue-level	session	91.19	97.50	97.50	1.19	83.05	58.33	67.75	82.28	0.53	81.30
T5DST	explicit	dialogue-level	turn	96.60	95.75	95.75	4.01	76.77	85.19	83.27	80.08	2.68	84.28
	implicit	dialogue-level	turn	90.04	92.25	92.25	2.88	66.20	58.37	54.78	51.75	1.66	75.26
	explicit	turn-level	turn	93.92	92.25	92.25	2.33	87.83	69.10	63.55	57.06	1.39	93.64
	explicit	dialogue-level	task	90.89	96.00	96.00	1.05	73.08	61.25	83.93	80.88	0.85	83.75
	explicit	dialogue-level	session	90.89	96.00	96.00	1.05	73.08	56.32	65.75	80.48	0.52	83.70

WOZ2.0 and MultiWOZ2.1 reaches 70.19% and 64.91%, respectively, while FA-del is only 1.82% and 3.51%, respectively.

- When negative feedback samples are added into the train set, the model's ability of understanding negative feedback utterances will be significantly improved, especially for deleting information, the model's performance has a revolutionary breakthrough. For example, for deleting information, the feedback accuracy of TRADE on WOZ2.0 is improved from 2.90% to 70.09%, while the feedback accuracy of SOMDST on MultiWOZ2.1 is improved from 4.08% to 88.11%.
- For different types of models, scratch-based models can obtain higher feedback accuracy for adding information without negative feedback training samples, such as TRADE and T5DST. In contrast, the feedback cost of the previous-based model is lower, and the feedback cost of such model will be further reduced after the training of negative feedback utterances, such as BERTDST and SOMDST.

The above results fully reflect that the modeling of negative feedback utterances is particularly important in the dialogue state tracking task. When the possible negative feedbacks in user utterances are effectively utilized, the performance of the dialogue state tracking model will be greatly improved. In addition, the negative feedback training samples can make the current dialogue state tracking model obtain excellent ability of understanding negative feedback utterances without reducing the original performance.

To further explore the influence of different numbers of negative feedback training samples, we sample negative feedback data of different proportions and add them into the original training set, and the performance is shown in Figure 7. It can be found that:

- The number of negative feedback training samples does not affect the original performance of the dialogue state tracking model, which also reflects that the accuracy of the simulated negative feedback training samples is high, and the practicability can be guaranteed.
- Only 20% of the negative feedback training samples are needed for the model to learn the ability of understanding the negative feedback utterance, while more negative feedback training samples do not bring significant improvement.

The above results show that, due to the high quality of the simulated negative feedback samples, only a small number of negative feedback training samples are needed to enable the model to acquire the ability of understanding the negative feedback utterances.

5.2 Impact of Different Interaction Mechanisms

In a dialogue system involving negative feedback utterances, the interaction mechanism between users and the system will play an important role. Here, according to the three aspects of the interaction mechanism involving negative feedback utterances proposed in this paper, we will study the performance of the dialogue state tracking model under different interaction mechanisms.

Table 4 shows the performance of all baselines when different interaction mechanisms are adopted, it can be seen that:

- For different feedback acquisition strategies, explicit strategy can make the model better understand negative feedback utterances, and correct the dialogue state in time. However, for implicit strategy, the system can only obtain the negative feedback of the current turn in the next turn of the dialogue, which makes the negative feedback lag behind, and the current turn dialogue state has not been corrected.

- For different feedback acquisition contents, dialogue-level content requires users to give feedback on all slot values. In this case, although the cost of negative feedback is high, the model can use sufficient negative feedback information to achieve better results. In contrast, turn-level content can only obtain less negative feedback information, but its feedback cost is low and it has excellent feedback accuracy.
- For feedback acquisition timing, when negative feedback is obtained at the end of each turn, the performance of the model is particularly outstanding, but the feedback cost is also extremely high. When only the negative feedback at the end of the task is obtained, the task joint accuracy and session joint accuracy can be guaranteed with minimal feedback cost. When only the negative feedback at the end of the whole session is obtained, the user only needs to pay a little feedback cost to ensure the session joint accuracy.

The above analysis shows that different interaction mechanisms have their own advantages, and the interaction mechanism should be chosen carefully according to the actual situation.

6 CONCLUSION

In this paper, we study the user negative feedback utterance, which is ignored in previous works but is inevitable in real-life conversations. Specifically, first, we define the schema of negative feedback utterances and propose a model to create simulated negative feedback utterances on the current datasets. Then, we explore the interaction mechanism involving negative feedback utterances and propose evaluation metrics related to negative feedback utterances. Finally, through extensive experiments on WOZ2.0 and MultiWOZ2.1 datasets, it can be found that user negative feedback utterance does play a very important role in dialogue state tracking, and different interaction mechanisms involving negative feedback utterances have their own advantages and disadvantages. Our future work will focus on building manually annotated negative feedback datasets.

ACKNOWLEDGMENTS

The work is supported by National Key R&D Plan (No. 2020AAA010 6600), National Natural Science Foundation of China (No.U19B2020, 62172039).

REFERENCES

[1] Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Inigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ–A Large-Scale Multi-Domain Wizard-of-Oz Dataset for Task-Oriented Dialogue Modelling. *arXiv preprint arXiv:1810.00278* (2018).

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

[3] Kyunghyun Cho, B van Merrienboer, Caglar Gulcehre, F Bougares, H Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*.

[4] 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* (2019).

[5] Ting Han, Ximing Liu, Ryuichi Takanobu, Yixin Lian, Chongxuan Huang, Dazhen Wan, Wei Peng, and Minlie Huang. 2020. MultiWOZ 2.3: A multi-domain task-oriented dialogue dataset enhanced with annotation corrections and co-reference annotation. *arXiv preprint arXiv:2010.05594* (2020).

[6] 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*. 35–44.

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

[8] 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*. 567–582.

[9] Takyoung Kim, Yukyung Lee, Hoonsang Yoon, Pilsung Kang, and Misuk Kim. 2021. Oh My Mistake!: Toward Realistic Dialogue State Tracking including Turnback Utterances. *arXiv preprint arXiv:2108.12637* (2021).

[10] Hwaran Lee, Jinsik Lee, and Tae-Yoon Kim. 2019. SUMBT: Slot-Utterance Matching for Universal and Scalable Belief Tracking. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 5478–5483.

[11] Shiyang Li, Semih Yavuz, Kazuma Hashimoto, Jia Li, Tong Niu, Nazneen Rajani, Xifeng Yan, Yingbo Zhou, and Caiming Xiong. 2020. Coco: Controllable counterfactuals for evaluating dialogue state trackers. *arXiv preprint arXiv:2010.12850* (2020).

[12] Zhaojiang Lin, Bing Liu, Seungwhan Moon, Paul Crook, Zhenpeng Zhou, Zhiguang Wang, Zhou Yu, Andrea Madotto, Eunjoon Cho, and Rajen Subba. 2021. Leveraging Slot Descriptions for Zero-Shot Cross-Domain Dialogue State Tracking. *arXiv preprint arXiv:2105.04222* (2021).

[13] Nikola Mrkšić, Diarmuid O Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. 2016. Neural belief tracker: Data-driven dialogue state tracking. *arXiv preprint arXiv:1606.03777* (2016).

[14] Jinjie Ni, Tom Young, Vlad Pandelea, Fuzhao Xue, Vinay Adiga, and Erik Cambria. 2021. Recent advances in deep learning based dialogue systems: A systematic survey. *arXiv preprint arXiv:2105.04387* (2021).

[15] 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 (2020), 1–67.

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

[17] Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368* (2017).

[18] Pararth Shah, Dilek Hakkani-Tür, Gokhan Tür, Abhinav Rastogi, Ankur Bapna, Neha Nayak, and Larry Heck. 2018. Building a conversational agent overnight with dialogue self-play. *arXiv preprint arXiv:1801.04871* (2018).

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

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

[21] 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*. 808–819.

[22] 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 (Volume 1: Long Papers)*. 1448–1457.

[23] 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. *arXiv preprint arXiv:2007.12720* (2020).

[24] Linhao Zhang, Dehong Ma, Xiaodong Zhang, Xiaohui Yan, and Houfeng Wang. 2020. Graph lstm with context-gated mechanism for spoken language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 9539–9546.

[25] Lin Zhao and Zhe Feng. 2018. Improving slot filling in spoken language understanding with joint pointer and attention. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 426–431.

[26] 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)*. 1458–1467.

[27] Qi Zhu, Kaili Huang, Zheng Zhang, Xiaoyan Zhu, and Minlie Huang. 2020. Crosswoz: A large-scale chinese cross-domain task-oriented dialogue dataset. *Transactions of the Association for Computational Linguistics* 8 (2020), 281–295.

A RELATED WORK

A.1 DST Benchmark Datasets

In recent years, dialogue state tracking has developed rapidly from single domain to multi-domain, and has formed a universally recognized benchmark dataset system. The single domain dialogue state tracking datasets most evaluated by previous work mainly include M2M [18], WOZ2.0 [20] and DSTC2 [7]. Among them, M2M is a machine-to-machine simulated conversation with manual post-processing, and it contains two completely independent single-domain datasets (movie and restaurant). DSTC2 and WOZ2.0 are standard benchmarks for human-to-machine and human-to-human single-domain dialogue state tracking respectively, both of which are in the restaurant domain. In particular, they provide the automatic speech recognition (ASR) hypotheses of user utterances to evaluate the robustness of the dialogue state tracking model against ASR errors.

Since the MultiWOZ [1] dataset was proposed, multi-domain dialogue state tracking has received the most extensive attention. Meanwhile, MultiWOZ and its modified versions [4, 5, 23] have gradually become the standard benchmarks for multi-domain dialogue state tracking. Built on the Wizard-of-Oz setup, MultiWOZ collects multi-domain dialogues from human-to-human conversations and builds a dialogue dataset covering more than 5 domains (restaurant, hotel, attraction, taxi, train). More recently, more multi-domain dialogue datasets have been proposed, such as SGD [16] and CrossWOZ [27], which focus on schema-guided continuous growth service dialogues and Chinese multi-domain dialogues, respectively.

A.2 Previous DST Models

Traditionally, dialogue state tracking is a downstream task of language understanding [24, 25], which uses semantics extracted upstream to estimate dialogue state. Due to the error propagation in this pipeline structure, current systems mostly use end-to-end dialogue state tracking to extract the dialogue state directly from the dialogue utterances [2, 6, 8, 10, 21, 22].

In general, end-to-end dialogue state tracking can be divided into two categories: ontology-based methods and open vocabulary-based methods. Ontology-based methods [10, 13, 26] usually assume that all slot values are known in advance and stored in the ontology, while classification models are usually used to predict the probability of each slot pair in the dialogue state. However, in practice, it is difficult to obtain all slot values in advance, so ontology-based methods are greatly limited.

Under the open vocabulary setting, the dialogue state tracking model has two different development directions. One is extraction-based model [2, 6, 22], which extracts a span from the dialogue utterances as a slot value for each extractable slot. The other is generation-based model [8, 12, 21] that directly uses dialogue content to generate appropriate slot values for each slot. Recent work shows that extraction-based models are more robust to unknown slot values where generation-based models are better for language diversity generalization.

A.3 Real-life DST

There is little research on real-life dialogue state tracking. The work most relevant to our research is on the modeling of turnback utterances [9], which suggests that users may change their minds during conversations in real-life. It is limited by the fact that it only uses a few simple templates to generate turnback utterances and does not take into account the negative feedback problem explored in this paper.

B ALGORITHMS

B.1 Algorithm 1: Application of our joint model in inference

Algorithm 1 Application of our feedback utterance generation and filtering joint model in inference

Input: our model $\mathcal{T}$, dialogue history H, ground turth of dialogue state S^{truth}, a dialogue state tracking model $\mathcal{D}$, negative emotional phrase template set P
Output: the original predicted dialogue state S^{ori}, dialogue state with negative feedback utterance S^{feed}

adding information $\alpha = \varnothing$, deleting information $\beta = \varnothing$
predict dialogue state: $S^{ori} = \mathcal{D}(H)$
for (slot, value) in S^{truth} **do**
  if (slot, value) not in S^{ori} **then**
    add (slot, value) to α
  end if
end for
for (slot, value) in S^{ori} **do**
  if (slot, value) not in S^{truth} **then**
    add (slot, value) to β
  end if
end for
utterance candidate set $U_\alpha = \mathcal{T}_{beam\_search}(Prompt_1 \oplus \alpha)$
utterance candidate set $U_\beta = \mathcal{T}_{beam\_search}(Prompt_1 \oplus \beta)$
adding information utterance $U_{add} = pop(U_\alpha)$
deleting information utterance $U_{delete} = pop(U_\beta)$
for u in U_α **do**
  filter result $r = \mathcal{T}(Prompt_2 \oplus u)$
  if r equal to α **then**
    $U_{add} = u$
  end if
end for
for u in U_β **do**
  filter result $r = \mathcal{T}(Prompt_2 \oplus u)$
  if r equal to β **then**
    $U_{delete} = u$
  end if
end for
add negative auxiliary verb to U_{delete}
negative emotion phrase $E = random\_choice(P)$
negative feedback utterance $U_{feed} = E \oplus U_{add} \oplus U_{delete}$
predict dialogue state: $S^{feed} = \mathcal{D}(H \oplus U_{feed})$
return S^{ori}, S^{feed}

B.2 Algorithm 2: Utilization of our joint model in train set

Algorithm 2 Utilization of our feedback utterance generation and filtering joint model in train set

Input: our model $\mathcal{T}$, turn user utterance U^{ori}, ground turth of dialogue state S^{truth}, slot value ontology O, negative emotional phrase template set P, max number of add M_{add}, max number of delete M_{delete},
Output: sign R, new user utterance U^{feed}, new dialogue state S^{feed}

1: $R = False$, adding information $\alpha = \emptyset$, deleting information $\beta = \emptyset$, $S^{feed} = S^{truth}$
2: select 1 to M_{add} (slot,value) pairs randomly from S^{truth} and add them to α
3: $S^{truth} = S^{truth} - \alpha$
4: select 1 to M_{delete} (slot,value) pairs randomly from S^{truth} and add them to β
5: select a new value randomly from O for each (slot, value) pair in α
6: utterance candidate set $U_\alpha = \mathcal{T}_{beam\_search}(Prompt_1 \oplus \alpha)$
7: utterance candidate set $U_\beta = \mathcal{T}_{beam\_search}(Prompt_1 \oplus \beta)$
8: adding information utterance $U_{add} = empty\ string$
9: deleting information utterance $U_{delete} = empty\ string$
10: **for** u in U_α **do**
11: filter result $r = \mathcal{T}(Prompt_2 \oplus u)$
12: **if** r equal to α **then**
13: $R = True$
14: $U_{add} = u$
15: **end if**
16: **end for**
17: **for** u in U_β **do**
18: filter result $r = \mathcal{T}(Prompt_2 \oplus u)$
19: **if** r equal to β **then**
20: $R = True$
21: $U_{delete} = u$
22: **end if**
23: **end for**
24: **if** U_{delete} is not a empty string **then**
25: add negative auxiliary verb to U_{delete}
26: **for** (slot, value) in β **do**
27: delete (slot, value) in S^{feed}
28: **end for**
29: **end if**
30: **if** U_{add} is not a empty string **then**
31: **for** (slot2, value2) in α **do**
32: change the value of slot2 in S^{feed} to value2
33: **end for**
34: **end if**
35: negative emotion phrase $E = random\_choice(P)$
36: $U_{feed} = U^{ori} \oplus E \oplus U_{add} \oplus U_{delete}$
37: **return** R, U^{feed}, S^{feed}

The operation of "add negative auxiliary verb" in line 29 of Algorithm 1 and Line 25 of Algorithm 2 is to directly replace the auxiliary verbs in the utterance with their antonyms, as shown in Table 5.

Table 5: The substitution of auxiliary verbs

Before	After
need	don't need
want	don't want
i'm	i'm not
i am	i am not
i would	i would not
i'd	i'd not
i'll	i'll not
i don't care	i am not don't care

C MODEL DETAILS

C.1 Configuration of all baselines

The model configuration for all baseline models in this paper is shown in Table 6.

Table 6: The detailed setting of hyperparameters. MWOZ2.1 refers to the MultiWOZ2.1 dataset

Hyperparameters	TRADE	BERTDST	SOMDST	T5DST
Training epochs	200	200	30	10 on WOZ2.0 5 on MWOZ2.1
Early stop evaluation	Joint acc	Loss	Joint acc	Loss
Early stop patient epochs	6	6	6	2
Decoder teacher forcing	0.5	-	0.5	-
Dropout	0.1	0.1	0.1	0.1
Word dropout	0.1	0.1	0.1	-
RNN hidden size	400	-	768	-
Batch size	32	16	16 on WOZ2.0 32 on MWOZ2.1	64 on WOZ2.0 128 on MWOZ2.1
Input max length	-	100 on WOZ2.0 150 on MWOZ2.1	150 on WOZ2.0 256 on MWOZ2.1	512
Learning rate	1e-3	Enc: 4e-5 Dec: 1e-4	Enc: 4e-5 Dec: 1e-4	5e-4 on WOZ2.0 1e-4 on MWOZ2.1
Warmup proportion	-	Enc: 0.1 Dec: 0.1	Enc: 0.1 Dec: 0.1	-

C.2 Details of our feedback utterance generation and filtering joint model

We take T5-small as the backbone of our joint model and use WOZ2.0 and MultiWOZ2.2 as its training data respectively. In the training, the training epochs is set to 20 and the early stop with the patient epoch as 5 is adopted on the development set. The learning rate is set to 5e-4 and the maximum sequence length is set to 50. On WOZ2.0, when the batch size is set to 30, the final accuracy of the best checkpoint can reach 87.3%. On MultiWOZ2.1, the batch size is set to 240, and the final accuracy of the best checkpoint is 87.8%. The final accuracy is the accuracy to check whether the dialogue state obtained after the generation and filtering of our model is exactly the same as the original dialogue state.