Knowledge-Based Systems 286 (2024) 111383

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Multi-domain gate and interactive dual attention for multi-domain dialogue state tracking

Xu Jia [a], Ruochen Zhang [b], Min Peng [a,*]

[a] School of Computer Science, Wuhan University, Wuhan, China
[b] School of Engineering Science, Lappeenranta University of Technology, Lahti, Finland

ARTICLE INFO

Keywords:
Multi-domain dialogue state tracking
Multi-domain gate
Interactive dual attention

ABSTRACT

Multi-domain dialogue state tracking (MDST) is a crucial component of task-oriented dialogue systems. In the context of multi-turn dialogues between the user and the system, MDST necessitates the continuous keeping track of dialogue states based on the information present in the current dialogue utterance and the dialogue states from the preceding turn. Recent work achieves the successful execution of multi-domain dialogue tasks by adopting an approach that treats each state as an individual label, while regrettably neglecting the potential benefits of incorporating domain-specific information associated with these states. Simultaneous, existing models exhibit a deficiency in effectively modelling the explicit correlations between dialogue contextual semantics and dialogue states. In this paper, we introduce the modules of multi-domain gate and interactive dual attention as novel solutions to address the aforementioned concerns. For the efficient exploitation of domain-specific information within states, we leverage the multi-domain gate as indices to amplify the states pertinent to the current utterance domain while filtering out irrelevant states. Interactive dual attention comprises utterance attention and slot attention, effectively modelling the correlation between dialogue utterances and slots. Additionally, interactive dual attention ensures that each dialogue utterance interacts with the slots once to derive all state updates, thereby ensuring computational efficiency. Specifically, slot attention models the associations between slots by incorporating semantic features to forecast updates in slot values. Meanwhile, utterance attention captures the semantics of dialogue context and integrates it with slot name features to generate dialogue states. All the aforementioned modules are designed based on a slot-independent framework, enabling efficient scalability of slots and circumventing issues related to model input limitations. The experimental results on the multi-domain dialogues dataset MultiWOZ 2.4 demonstrate the superior performance of our model compared to the baselines. Additionally, we conduct a comprehensive analysis of the effectiveness of the multi-domain gate and interactive dual attention modules, elucidating their contribution to the performance of the model through visualization and case studies.

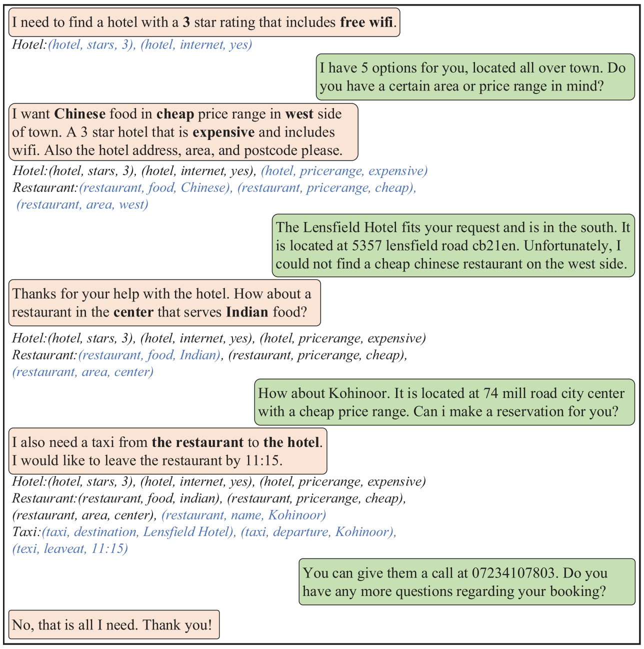
1. Introduction

Task-oriented dialogue assumes a crucial role within the dialogue system, encompassing functionalities that enable users to accomplish diverse tasks such as book tickets, control smart homes, and others [1–3]. Dialogue state tracking (DST) can extract essential features from the dialogue history in the pipeline of task-oriented dialogue to obtain the current dialogue states [4,5]. DST treats the dialogue states as a set of *(slot, value)* pairs. For example, *(stars, 3)* can be used to represent a 3-star rating hotel in the hotel domain. In contrast to the single-domain dialogue state tracking task, real-life dialogues often encompass multiple inter-connected domains. As illustrated in Fig. 1, the dialogue scenario encompasses various domains such as hotel, restaurant, and taxi. Consequently, multi-domain dialogue state tracking (MDST) has garnered increased attention within the research community, surpassing the focus on single-domain tracking [6–9]. The prevailing methodologies adopt *(domain, slot, value)* triples to supplant the former *(slot, value)* pairs in handling the MDST task. In line with prior work [10,11], we adopt the convention of considering domain and slot name pairs as slots in the context of multi-domain tasks.

Recent research has witnessed notable advancements in both single-domain and multi-domain dialogue state tracking (DST) tasks. In scenarios with a limited number of domains and slots, numerous studies adopt the approach of treating the DST task as a classification task by predefining a fixed ontology [12–14]. Despite the introduction of the

[*] Corresponding author.
E-mail addresses: jia_xu@whu.edu.cn (X. Jia), Ruochen.zhang@student.lut.fi (R. Zhang), pengm@whu.edu.cn (M. Peng).

https://doi.org/10.1016/j.knosys.2024.111383
Received 1 August 2023; Received in revised form 8 October 2023; Accepted 7 January 2024
Available online 9 January 2024



I need to find a hotel with a **3** star rating that includes **free wifi**.

Hotel:(hotel, stars, 3), (hotel, internet, yes)

I have 5 options for you, located all over town. Do you have a certain area or price range in mind?

I want **Chinese** food in **cheap** price range in **west** side of town. A 3 star hotel that is **expensive** and includes wifi. Also the hotel address, area, and postcode please.

Hotel:(hotel, stars, 3), (hotel, internet, yes), (hotel, pricerange, expensive)
Restaurant:(restaurant, food, Chinese), (restaurant, pricerange, cheap), (restaurant, area, west)

The Lensfield Hotel fits your request and is in the south. It is located at 5357 lensfield road cb21en. Unfortunately, I could not find a cheap chinese restaurant on the west side.

Thanks for your help with the hotel. How about a restaurant in the **center** that serves **Indian** food?

Hotel:(hotel, stars, 3), (hotel, internet, yes), (hotel, pricerange, expensive)
Restaurant:(restaurant, food, Indian), (restaurant, pricerange, cheap), (restaurant, area, center)

How about Kohinoor. It is located at 74 mill road city center with a cheap price range. Can i make a reservation for you?

I also need a taxi from **the restaurant** to **the hotel**.
I would like to leave the restaurant by 11:15.

Hotel:(hotel, stars, 3), (hotel, internet, yes), (hotel, pricerange, expensive)
Restaurant:(restaurant, food, indian), (restaurant, pricerange, cheap), (restaurant, area, center), (restaurant, name, Kohinoor)
Taxi:(taxi, destination, Lensfield Hotel), (taxi, departure, Kohinoor), (texi, leaveat, 11:15)

You can give them a call at 07234107803. Do you have any more questions regarding your booking?

No, that is all I need. Thank you!

Fig. 1. An illustration of a multi-domain dialogue state tracking task. The user utterances on the left and the corresponding system agent utterances on the right are presented. The dialogue states for each domain are denoted as a collection of triples in the format *(domain, slot, value)*. The updated states for the current turn are highlighted in blue.

new dataset [6,7] with an increased number of domains and slots, some methods continue to achieve state-of-the-art results by leveraging similarity calculations with predefined *(slot, value)* pairs [10,14]. However, in real scenarios, many slot values cannot be obtained by predefinition, and even some are infinite (such as *hotel names*). The introduction of additional domains would render these issues unsolvable. Consequently, some work endeavours adopt open vocabulary models that rely on generating [15,16] or extracting [17,18] slot values from the dialogue history and existing dialogue states [19]. With the progress in large language models, recent work has leveraged the generative capacity of language models combined with prompt learning to achieve remarkable results across various datasets [20–24]. However, the aforementioned methods all treat multi-domain dialogue states as distinct labels when confronted with MDST tasks, failing to fully harness the domain-specific features within the states. Furthermore, the current models lack the explicit construction of a bidirectional correlation between the dialogue history and dialogue states. In the MDST task, prevalent open vocabulary models generally adopt techniques like concatenating the dialogue history and dialogue states within the encoder or iteratively computing the relationship between the dialogue history and each individual dialogue state. Both of the aforementioned methods not only fail to establish an explicit bidirectional correlation between them but also pose input limitation and computational inefficiency challenges as the number of slots increases, respectively. In this paper, we propose the adoption of Input Length Constraints (ILC) as a metric for evaluating the input length of the model, while also introducing Inference Time Complexity (ITC) [16,22] as a measure of computational efficiency. A model that concatenates dialogue history and dialogue states with the ILC set to $L(J)$, where J represents the number of slots, and achieves the best ITC of $O(1)$. This indicates that while these models demonstrate efficient inference, the input length is influenced by the number of slots. As for models that iteratively calculate both the dialogue history and each individual dialogue state, they exhibit the ILC of $L(1)$ and the best ITC of $O(J)$. These models remain unaffected by input constraints, yet their computational efficiency diminishes with the increasing number of slots. Hence, existing methods still encounter unresolved challenges when confronted with MDST tasks.

To tackle these challenges, we propose Multi-Domain Gate and Interactive dual Attention (MGIA) for the multi-domain dialogue state tracking (MDST) task. By employing the multi-domain gate, the model becomes capable of capturing a broader spectrum of domain-specific features. During inference, the domain can be regarded as an index, thereby facilitating the enhancement of domain-relevant information while filtering out domain-irrelevant states. Moreover, the utilization of the multi-domain gate presents a viable solution for scenarios in which multiple domains are present within a single turn, as exemplified in Fig. 1 where the hotel and restaurant domains co-occur during the 2nd turn of dialogue. Subsequently, leveraging the foundation of the vanilla transformer mechanism, we introduce interactive dual attention to effectively capture the bidirectional association between the dialogue history and slots. Specifically, slot attention facilitates the reinforcement of inter-slot relationships within the slot matrix through a multi-layer self-attention mechanism, following the acquisition of semantic features. Hence, the prediction obtains the state operator for each slot, which is subsequently employed to ascertain the necessity of updating the current state. By integrating slot name features into the semantic context of the dialogue, utterance attention can leverage the dialogue history during the decoding phase to produce slot values for the respective slots. To address the input limitation and computational inefficiency challenges posed by existing methods, our model adopts a slot-independent framework. We vertically concatenate all slots to obtain an encoder-independent slot matrix. The independent slot matrix exhibits excellent scalability without input limitations resulting from the inclusion of domains and slots, thereby resulting in the ILC of $L(1)$. In the interactive dual attention module, a single interaction between the dialogue history and the slot matrix is performed to acquire the updates for all state operators. Consequently, our model achieves the best ITC of $O(1)$. We conduct a validation of our model using the most recent multi-domain dialogue dataset, MultiWOZ 2.4 [7]. To comprehensively assess the efficacy of our model and baselines, we have expanded upon the original evaluation metrics by introducing extensions to Slot(Recall), Operator(Acc), and Dialogue(Acc) metrics. Subsequent experiments investigate the efficacy of the multi-domain gate and interactive dual attention modules proposed in this paper. Furthermore, the functions of each module are illustrated through visualization and case studies. In summary, our work makes the following main contributions:

- We introduce a novel slot-independent framework for multi-domain dialogue state tracking tasks, incorporating the multi-domain gate and interactive dual attention modules. In contrast to prior methods based on open vocabulary, our model effectively tackles the challenges of input limitation and computational inefficiency in the context of multi-domain tasks.
- The multi-domain gate effectively leverages domain information among states as an indexing mechanism to strengthen domain-specific features and filter out states irrelevant to the given domain. Simultaneously, it adeptly tackles the challenges associated with multi-domain settings encountered in dialogues. The interactive dual attention module explicitly captures and models the correlation between the semantic representations of the dialogue context and the dialogue states.
- We conduct a comprehensive evaluation of our proposed model on the latest publicly available multi-domain dialogue dataset, MultiWOZ 2.4. To provide a comprehensive assessment, we introduce novel evaluation metrics, namely Slot(Recall), Operator(Acc), and Dialogue(Acc). These three metrics assess the accuracy of slot value generation, operator accuracy, and dialogue level accuracy, respectively. Furthermore, our experiments also assess the effectiveness of the multi-domain gate and interactive dual attention from various perspectives.

The rest sections of this paper are structured as follows: Section 2 composes the related work for the dialogue state tracking task. Section 3 describes our model in detail. Section 4 demonstrates the dataset and the experimental setup. Section 5 discuss the performance of the model and measure model validity in several dimensions. Finally, Section 6 concludes the paper.

2. Related work

Dialogue state tracking (DST) is a component of dialogue management in task-oriented dialogues that serves as a bridge between natural language understanding (NLU) and dialogue policy learning. Recent works have argued that feeding NLU results into DST in the pipeline leads to the accumulation of errors, ultimately leading to a decline in the overall performance of the model [12]. Hence, to address this issue, recent studies have proposed end-to-end models that demonstrate state-of-the-art performance. These approaches can be broadly categorized into two main groups: the predefined ontology-based models and the open vocabulary-based models.

2.1. Predefined ontology-based models

The predefined ontology-based model treats the DST task as a classification task [11,25,26]. These methods require the definition of all *(slot, value)* pairs. The models aim to identify the appropriate pairs by analysing the utterances of the appropriate *(slot, value)* pairs. To capture the features of dialogue context and dialogue states, the models commonly employ LSTM modules [12] to encode the user and system utterances and the states, respectively. The introduction of pre-trained models and graph structures enables the models to effectively capture the features that exist between dialogue states and utterances [27,28]. The most recent research employs noise-enhanced training to rectify previously incorrect state predictions in subsequent turns [29]. Nevertheless, in real-world multi-domain dialogue scenarios, it poses a challenge to predefine all the ontology in advance. On the one hand, the practical collection of numerous ontologies poses a challenge. On the other hand, certain slots exhibit an unbounded set of possible slot values, such as hotel names. Therefore, recent works have primarily emphasized the adoption of open vocabulary-based models [16].

2.2. Open vocabulary-based models

To tackle the challenges presented by predefined ontology-based models, open vocabulary-based models approach the task of multi-domain dialogue state tracking (MDST) as either an extraction [17,18,30] or a generation [31] task. Therefore, these methods typically span or generate slot values from the dialogue context. Extraction-based models commonly transform an MDST task into a reading comprehension task [17]. These models consider the dialogue history as articles and the slots as questions, resembling a reading comprehension setup. They can infer slot values by identifying the starting and ending positions of spans within a dialogue. The generation-based models employ a soft-copy mechanism to retrieve and incorporate relevant words from the dialogue history and vocabulary as slot values [15,16]. With the advancements in large language models, recent work has leveraged the generative capabilities of pre-trained models and augmented them with prompt learning techniques to attain state-of-the-art performance [23,24,32–34]. The recent work [35] utilizes Beam Search to replace the original model's greedy search, leading to a remarkable enhancement in the performance of the generated model. Certain models concatenate the dialogue history and dialogue states and input them into the pre-trained model, obtaining comprehensive updates of the states in a single step [16,19,36]. Nevertheless, the issue of input limitation arises as the number of domains and slots increases since the pre-trained model imposes a constraint on the maximum input length. Some models avoid the problem by iteratively processing each dialogue state and dialogue history to obtain updates for individual states [17,18,37]. However, this methodology comes at the cost of computational efficiency, leading to slower computational efficiency. Furthermore, the aforementioned methods disregard the domain-specific features between state labels while addressing the challenges of multi-domain dialogue state tasks. Moreover, these methods do not possess the capability to explicitly model the correlation between the dialogue history and the dialogue state.

To address these challenges, we introduce novel components, namely the multi-domain gate and interactive dual attention (MGIA), designed specifically for the multi-domain dialogue state tracking task. In contrast to the aforementioned approach, our model adopts a slot-independent framework. By vertically concatenating all slots, we construct an encoder-independent slot matrix that exhibits excellent scalability. The multi-domain gate enhances the representation of multi-domain features in utterances and serves as an index to filter out domain-irrelevant states. The interactive dual attention module explicitly captures the correlations between the semantics of the dialogue context and the dialogue states. It facilitates comprehensive state updates through a single interaction.

3. Model

In this section, we provide an overview of the Multi-domain Gate and Interaction dual Attention (MGIA) model. Subsequently, we present a detailed description of the encoder, interactive dual attention, and multi-domain gate components. Lastly, the state generator is introduced.

3.1. An overview of MGIA

The architecture of MGIA is demonstrated in Fig. 2. The bottom-left portion illustrates the utilization of a pre-trained model as the encoder in our framework. The input to the encoder includes the user utterance, dialogue history, and the dialogue states from the previous turn. The right-hand side presents the encoder-independent slot matrix that encompasses all $(domain, slot)$ pairs. We employ the identical embedding weights from the pre-trained language model of the encoder to map all slot names to the dialogue semantic space, resulting in the generation of the slot matrix. The interactive dual attention module comprises two components: slot attention, responsible for predicting the updates for each slot, and utterance attention, which aids in generating the corresponding slot value. To fully leverage the domain-specific features among states, we introduce the multi-domain gate. The multi-domain gate can be regarded as an indexing mechanism that amplifies the relevant features of the current domain while filtering out domain-irrelevant states. It facilitates the slot attention mechanism in accurately capturing the desired operators. Finally, the state generator produces the corresponding values using the operator and the output of utterance attention as the basis.

3.2. Encoder and slot matrix

3.2.1. Encoder

We employ the pre-trained language model Bert [38] as the encoder to extract vector representations of the dialogue context. We denote the representation of the dialogue utterances at the turn t as $U_t = Sys_t \oplus ; \oplus User_t \oplus [SEP]$, where Sys_t and $User_t$ are the utterances of system and user at the turn t, respectively. $[SEP]$ is used to mark the end of a dialogue turn. We concatenate the utterances before the turn t in reverse order as the dialogue history, denoted as $C_t = U_{t-1} \oplus \cdots \oplus U_1$. Because the closer the utterance is to the tth turn, the more relevant it is. We define the multi-domain dialogue states at the turn t, $B_t = \{b_1 \oplus \cdots \oplus b_J | value_j \neq NULL\}$, where $b_j = \{domain_j \oplus slot_j \oplus - \oplus value_j \oplus ;\}$, $domain_j$, $slot_j$ and $value_j$ denote the state, domain name, slot name and slot value corresponding to the jth slot, respectively. The input

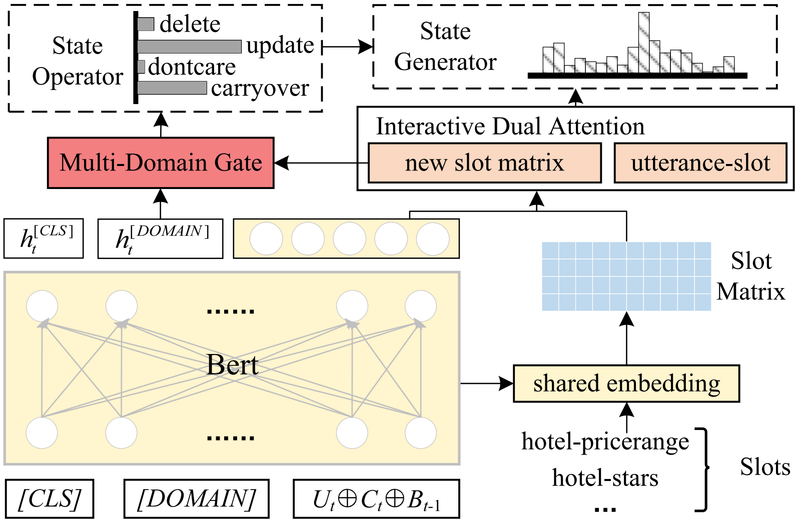


Fig. 2. The illustration of the multi-domain gate and interactive dual attention (MGIA) model. We use the fine-tune pre-trained model to obtain the slot matrix and the representation of dialogue context. The multi-domain gate module effectively exploits the domain-specific features between states. The interactive dual attention explicitly captures the correlations between the semantic information of the dialogue context and the dialogue states.

tokens for Bert consist of the concatenation of the dialogue utterances for the current turn, the dialogue history, and the dialogue states from the previous turn.

$$X_t = [CLS] \oplus [DOMAIN] \oplus U_t \oplus C_t \oplus B_{t-1}, \tag{1}$$

$$H_t^X = Bert(X_t), \tag{2}$$

where $[DOMAIN]$ is a special token inserted before U_t to consolidate the domain information of the utterance at the tth turn into a vector representation. In contrast to other models that rely on the $[CLS]$ token for domain prediction, our approach utilizes a dedicated token, $[DOMAIN]$, for predicting the domain of the current utterance. This is motivated by the fact that the $[CLS]$ representation encompasses features from the current utterance U_t, the dialogue history C_t, and the previous turn's dialogue states B_{t-1}. To ensure that the dialogue history and dialogue states do not influence the prediction, we employ the representation of $[DOMAIN]$ specifically for domain prediction.

The input to the Bert model is composed of the sum of the embeddings of X_t, segment id embeddings, and position embeddings. For the segment id, we assign the value of 1 to tokens corresponding to U_t, and the value of 0 to tokens corresponding to C_t and B_{t-1}. The position embeddings follow the standard choice of Bert. We obtain a d-dimensional vector representation for each token in X_t. The output representation of the encoder is $H_t \in \mathbb{R}^{|X_t| \times d}$. $h_t^{[CLS]}, h_t^{[DOMAIN]} \in \mathbb{R}^d$ are the outputs that correspond to $[CLS]$ and $[DOMAIN]$, respectively.

3.2.2. Encoder-independent slot matrix

The current models exhibit limited consideration for the complexities posed by multi-domain scenarios, thereby overlooking the potential challenges arising from the growing number of domains and slots in MDST. As the number of domains and slots increases, these models encounter challenges related to input limitation and computational inefficiency. Therefore, we introduce a novel approach to represent an encoder-independent slot matrix. We vertically concatenate all domain and slot pairs into a J-dimensional matrix $M^S \in \mathbb{R}^{J \times max(|slot|)}$, where $|\cdot|$ denotes the number of tokens, as depicted in the bottom right of Fig. 2. Following previous works [10], we employ the identical embedding utilized in the pre-trained model to encode all slots. Taking into account the varying lengths of slot names, we utilize the mean embedding values as the representations for the slots. Thus, we have:

$$E^S = Embedding(M^S), \tag{3}$$

$$H^S = MeanPooling(E^S), \tag{4}$$

where $H^S \in \mathbb{R}^{J \times d}$ and d is the hidden layer size.

As a growing number of domains and slots are encountered, current methods either concatenate or iteratively process the dialogue states, resulting in challenges related to input constraint or computational inefficiency. Instead, we independently input all $(domain, slot)$ pairs into the model, utilizing the shared embedding module to map them to the same semantic space as the encoder, thus obtaining the slot matrix. The encoder is only required to take as input the dialogue history and the dialogue states from the previous turn. The model's Input Length Constraint (ILC) is defined as $L(1)$ due to the fixed lengths of the preceding turns in both dialogue history and dialogue states, with an imposed upper limit. This implies that the model's inputs are not susceptible to exceeding the pre-defined input length of the pre-trained model as the number of slots increases. Simultaneously, the interactive dual attention module computes the context semantics and slot matrix, both mapped to the same semantic space, in a single computation step to derive the operators for all states. This suggests that the model achieves the best Inference Time Complexity (ITC) of $O(1)$, and it does not undergo additional traversals with an increase in the number of slots. In summary, the utilization of an encoder-independent slot matrix by the model mitigates the challenges posed by input limitation and computational inefficiency encountered in existing models as the number of slots increases. The slot matrix of the independent encoder effectively scales by vertically expanding the matrix length whenever new slots are introduced.

3.3. Interactive dual attention

Prior models [39] solely rely on slot attention, which fails to sufficiently capture the correlation between dialogue context semantics and the corresponding slots. We introduce an interactive dual attention module, built upon the vanilla transformer mechanism [40], that explicitly captures the correlation between the semantics of the dialogue context and the associated slots. This module enables comprehensive integration of these two crucial features, as illustrated in Fig. 3. The utterance-slot and slot-utterance representations are derived by swapping the encoder and slot matrix outputs that correspond to the keys and values in the self-attention mechanism. We utilize the utterance-slot representation to encapsulate the semantic features of the dialogue

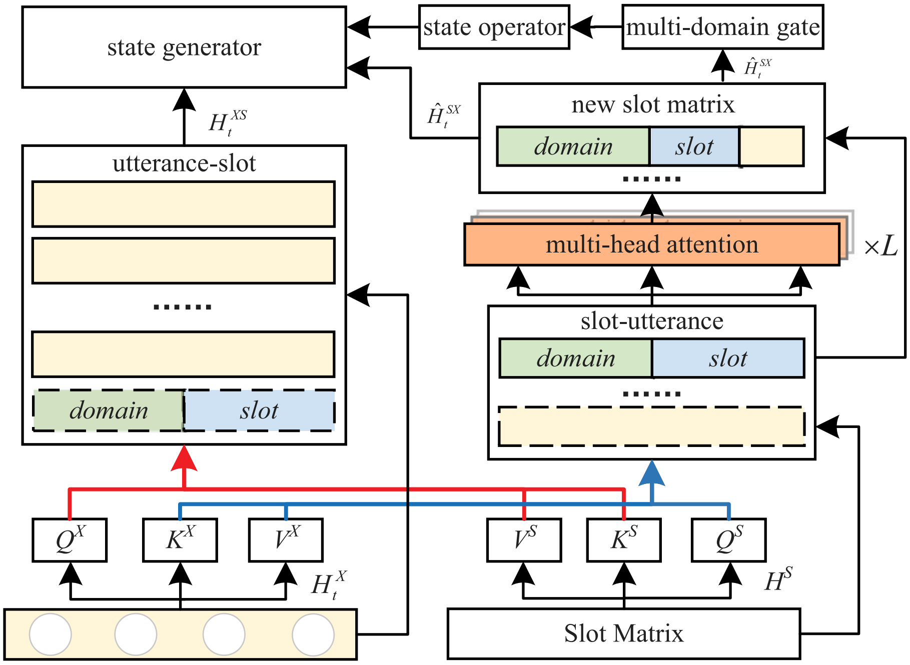


Fig. 3. The illustration of interactive dual attention. The encoder and slot matrix outputs are mapped to the corresponding Q, K and V, respectively. The utterance-slot and slot-utterance are obtained by swapping K and V. The former is depicted on the left side of the figure as a semantic representation of the dialogue context for the input of the state generator. The latter, following multiple layers of multi-head attention, generates the new slot matrix that is utilized for extracting the state operators and values associated with the respective slots.

context, which are then employed for the generation of dialogue states in subsequent steps, as illustrated in the left side of Fig. 3. To enhance the accuracy of slot updates, the slot-utterance undergoes multiple layers of multi-head self-attention, resulting in a new slot matrix that integrates the contextual semantics of the dialogue, as depicted in the right side of Fig. 3. The newly generated slot matrix is employed to generate state operators and values corresponding to the respective slots.

3.3.1. Multi-head attention

We first briefly introduce the vanilla multi-head attention mechanism [40]. We are provided with three matrices $Q \in \mathbb{R}^{|Q|\times d}$ as queries, $K \in \mathbb{R}^{|K|\times d}$ as keys and $V \in \mathbb{R}^{|V|\times d}$ as values. The attention A with N heads is calculated as follows:

$$A^n = softmax(\frac{Q^n K^{n\mathrm{T}}}{\sqrt{d/N}})V^n, \tag{5}$$

$$A = concat(A^1, \dots, A^N), \tag{6}$$

where $1 \le n \le N$. We formulate the entire process as:

$$A = MultiHead(Q, K, V). \tag{7}$$

3.3.2. Utterance attention and slot attention

To explicitly capture the correlations between dialogue utterances and all slots, we employ a multi-head attention mechanism to build an interactive dual attention module, which encompasses both utterance attention and slot attention. Specifically, to derive utterance representations that incorporate slot features, we utilize the encoder output as the query and the slot matrix as both the key and value. Consequently, the attention between utterance representation H_t^X and slot matrix representation H^S is summarized as:

$$A_t^{XS} = MultiHead(H_t^X, H^S, H^S), \tag{8}$$

where $A_t^{XS} \in \mathbb{R}^{|X_t|\times d}$. We concatenate the utterances representations H_t^X and utterance attention A_t^{XS}. The utterance-slot representation is derived by employing feed-forward neural networks to obtain:

$$H_t^{XS} = W_2^{XS} LeakyReLU(W_1^{XS}(H_t^X \oplus A_t^{XS})), \tag{9}$$

where W_1^{XS}, W_2^{XS} are learnable parameters. We use the same approach to obtain the slot attention A_t^{SX} and the slot-utterance representation H_t^{SX} incorporating features of utterances:

$$A_t^{SX} = MultiHead(H^S, H_t^X, H_t^X), \tag{10}$$

$$H_t^{SX} = W_2^{SX} LeakyReLU(W_1^{SX}(H^S \oplus A_t^{SX})). \tag{11}$$

To strengthen the relationship and interaction between slots and utterances, we additionally apply self-attention to the slot-utterance representation. We utilize the identical multi-head attention mechanism and subsequently apply layer normalization to normalize the results. We stack this process with L layers as follows:

$$G^l = F^l + MultiHead(F^l, F^l, F^l), \tag{12}$$

$$F^{l+1} = G^l + W_2 \max(0, W_1 G^l), \tag{13}$$

where $1 \le l \le L, F^1 = H_t^{SX}, \hat{H}_t^{SX} = F^L$. $\hat{H}_t^{SX}$ represents the updated slot matrix that integrates the utterances and slot features from the tth turn.

3.4. Multi-domain gate

The existing work fails to acknowledge the correlation between domains and slots, resulting in inadequate exploitation of domain information for effectively filtering out irrelevant slots. Many slots correspond to specific domains, such as the slot *food* belongs to the domain *restaurant*. Several existing methods [16,19,41] aim to enhance the performance of the model by incorporating domain features. However, these methods either treat the domain information as an auxiliary task or solely focus on the single-domain scenario. In the multi-domain dialogue state tracking task, it is common for a single dialogue turn to encompass multiple domains. For example, in the second and third turns of the user dialogue in Fig. 1, the domains are both *hotel* and *restaurant*. Therefore, we propose the multi-domain gate and treat the domain classification task as a multi-label classification task. The gate mechanism efficiently preserves the information pertaining to the

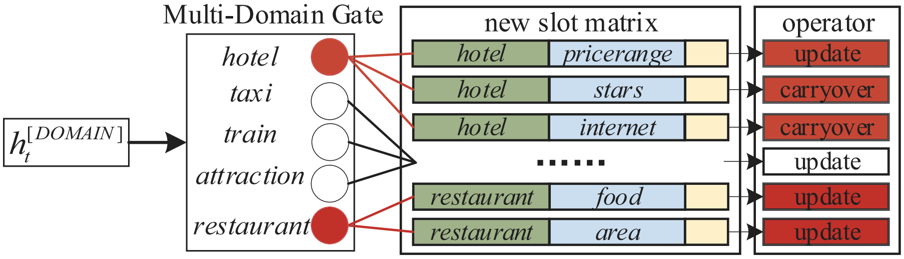


Fig. 4. The illustration of the multi-domain gate. We categorize the current utterance features into multiple domains and employ the gate mechanism to extract the corresponding features from the new slot matrix. The utilization of the multi-domain gate enhances domain-specific features and effectively filters out irrelevant slots.

respective slots while filtering out irrelevant slots based on the current utterance domain, as shown in Fig. 4.

In this paper, we refrain from utilizing $[CLS]$ for the prediction of the multi-domain of the current utterance. This decision is motivated by the fact that $[CLS]$ encompasses contextual information from the dialogue history and the state of the previous turn, which has the potential to introduce interference with the accurate representation of the domain information in the current utterance. Therefore, we incorporate a specialized token $[DOMAIN]$ into the input sequence to enable the prediction of multi-domain labels. We employ a multi-label classification approach to estimate the probabilities of multiple domains:

$$P_{mdom,t} = \sigma(W_{mdom} h_t^{[DOMAIN]}), \tag{14}$$

where W_{mdom} denotes a trainable parameter, $P_{mdom,t}$ represents the multi-domain probability at turn t, and $\sigma(\cdot)$ denotes the sigmoid activation function. The probability $P_{mdom,t}$ can be interpreted as a weighted combination of multi-domain features. A higher value of $P^i_{mdom,t}$ suggests that the ith domain is more prominent in the current utterance of the dialogue. Hence, it is essential for the dialogue state operator to allocate greater attention to the modifications occurring within these domain-specific states. We utilize $P_{mdom,t}$ as the weighting factor for $\hat{H}_t^{SX}$ to derive the operator corresponding to each state:

$$P_{opr,t} = softmax(W_{opr}(P_{mdom,t}\hat{H}_t^{SX})), \tag{15}$$

where W_{opr} represents a learnable parameter, and $P_{opr,t} \in \mathbb{R}^{J\times|O|}$ denotes the probability distribution over operators for all slots at the turn t. The state operators are defined as $O = \{DELETE, UPDATE, DONTCARE, CARRYOVER\}$. $DELETE$ represents the action of removing the previous slot value, $UPDATE$ indicates generating a new slot value, $DONTCARE$ signifies that the slot value is considered as *dontcare*, and $CARRYOVER$ implies that the slot value remains the same as in the previous turn. Based on the probabilities $P_{opr,t}$, we can determine the state operators to be applied to each slot. The precise state operators facilitate the model in making more accurate predictions of the dialogue states.

3.5. State generator

For the decoding phase, we employ a GRU-based decoder, similar to prior works [15,16]. The GRU is initialized with $g_t^{j,0} = h_t^{[CLS]} + h_t^{[DOMAIN]}$ and $e_t^{j,0} = \hat{H}_t^{SX,j}$, and $e_t^{j,k}$ represents the word embedding generated in every step:

$$g_t^{j,k} = GRU(e_t^{j,k}, g_t^{j,k-1}). \tag{16}$$

The generation of the final dialogue states is determined by the probability distribution of vocabulary $P^{j,k}_{vcb,t}$ and the dialogue context $P^{j,k}_{ctx,t}$. The probability distributions of the vocabulary and dialogue context are obtained by utilizing all word embeddings and encoder outputs:

$$P^{j,k}_{vcb,t} = softmax(Eg_t^{j,k}), \tag{17}$$

$$P^{j,k}_{ctx,t} = softmax(H_t^{XS} g_t^{j,k}), \tag{18}$$

where E denotes the matrix of all word embeddings. The generation probability of the final state is obtained by summing these probabilities:

$$P^{j,k}_{w,t} = \lambda P^{j,k}_{vcb,t} + (1-\lambda) P^{j,k}_{ctx,t}, \tag{19}$$

where the scalar λ is computed by:

$$\lambda = sigmoid(W_\lambda [g_t^{j,k}; e_t^{j,k}; c_t^{j,k}]), \tag{20}$$

where $c_t^{j,k} = P^{j,k}_{ctx,t} H_t^{XS}$.

4. Experiment setup

4.1. Dataset

We perform an evaluation of our model on the most recent version of the task-oriented dialogue dataset, MultiWOZ 2.4[1] [7], which has been updated with corrected annotations. MultiWOZ 2.4 represents an enhanced variant of MultiWOZ 2.1 [42] that includes refined annotations for the validation and test sets, improving the overall quality of the dataset. The MultiWOZ 2.4 dataset comprises a collection of over 10,000 multi-turn dialogues, encompassing seven distinct domains, namely $\{attraction, hotel, restaurant, taxi, train, hospital, police\}$. As the $hospital$ and $police$ domains are excluded from the validation and test sets, we exclusively focus on the remaining five domains in our experimental setup, which aligns with previous works. The resultant dataset comprises 17 distinct slots and 30 pairs of *(domain, slot)*, as depicted in Table 1, showcasing all the domains and their respective slots. Since the MultiWOZ 2.4 dataset only includes single-domain labels for each utterance, we augment the dataset by incorporating additional domain labels derived from updates in the dialogue states. We designate the domains of the user's 2nd and 3rd turn conversation in Fig. 1 as $\{0,1,1,0,0\}$, with the respective domain labels being $\{attraction, hotel, restaurant, taxi, train\}$. This signifies that when the label is 1, it indicates the presence of the corresponding domain in the current turn of the dialogue, whereas a label of 0 signifies the absence of the domain. Therefore, the label $\{0,1,1,0,0\}$ signifies that the ongoing conversation encompasses two domains: hotel and restaurant. We employ a multi-label method to signify the domains within the user's 2nd and 3rd turns of the conversation, encompassing both hotels and restaurants, in contrast to the prior single-label representation solely denoting the hotel domain. The analysis of the re-labelled dataset

[1] https://github.com/smartyfh/MultiWOZ2.4

Table 1
The dataset information of MultiWOZ 2.4.

Domain	Slot	Train		Dev		Test	
		Dialogue	Turn	Dialogue	Turn	Dialogue	Turn
Hotel	area, pricerange, stars, type, parking, name,internet, book day, book stay, book people	3381	14 926	416	1793	394	1760
Taxi	leaveat, arriveby, departure, destination	1854	4622	207	695	195	664
Train	leaveat, arriveby, departure, destination,day, book people	3103	12 308	484	2001	494	2008
Attraction	area, name, type	2717	8271	401	1242	395	1280
Restaurant	area, pricerange, food, name, book day,book time, book people	3813	15 400	438	1697	437	1725
All	-	8420	54 984	1000	7370	999	7368

reveals that 8.9% of the dialogues contain at least one turn that involves multiple domains. We adopt similar data preprocessing procedures to [15,16] for preprocessing the MultiWOZ 2.4 dataset.

4.2. Evaluation metrics

We primarily assess the performance of our models using the official evaluation metrics employed in the works of [7,16]. In this paper, we introduce Slot(Recall), Operator(Acc), and Dialogue(Acc) as evaluation metrics to assess the performance of our model in the multi-domain dialogue state tracking task. The following are the descriptions of these metrics:

- *Slot(Acc):* This metric represents the average accuracy in predicting the correct slot value.
- *Slot(Recall):* In the task of multi-domain dialogue state tracking, the majority of slots remain unpredicted, resulting in a high slot accuracy. Hence, we utilize Slot(Recall) as a metric to assess the average accuracy of the predicted slot values for each dialogue turn. The Slot(Recall) metric serves as an indicator of the quality of the generated results.
- *Slot(F1):* The metric assesses the combined accuracy and recall of the slots.
- *Operator(Acc):* The metric quantifies the classification accuracy of the model in predicting state operators, which play a crucial role in incorporating domain features. Hence, the metric also serves as an indicator of the effectiveness of the multi-domain gate.
- *Turn(Acc):* The metric evaluates the model's performance in accurately predicting all slots within a given turn. In previous works, this metric commonly referred to as *Joint Accuracy* is denoted as *Turn(Acc)* in this paper to maintain consistency with the terminology used for other metrics. *Turn(Acc)* serves as the primary evaluation metric for the multi-domain state tracking task.
- *Dialogue(Acc):* It provides an indication of the model's performance in accurately predicting all slots throughout the entire dialogue. *Dialogue(Acc)* imposes stricter requirements compared to *Turn(Acc)*, as it necessitates accurate predictions for every turn within the dialogue.

4.3. Implementation

We use the PyTorch library to implement the model. To ensure equitable comparisons, we adopt the configuration settings described in [16]. We utilize the pre-trained Bert [38] as the encoder component in our model, and we derive the slot matrix through the embedding layer of the same Bert model. We utilize the AdamW optimizer to optimize the parameters of our model. For the encoder, we set the peak learning rate and warmup proportion to 4e−5 and 0.1, respectively. As for the decoder, we set the peak learning rate and warmup proportion to 1e−4 and 0.1, respectively. The dropout rate is set to 0.1. During the experiments, we select the model that achieves the highest Turn(Acc) on the validation set and subsequently evaluate its performance on the test set.

4.4. Baseline

In the field of Multi-Domain Dialogue State Tracking (MDST), predefined ontology-based models suffer from limitations in acquiring ontology and computational inefficiency. Hence, in this paper, we evaluate the performance of our model in comparison to open vocabulary-based models. Our baselines mainly include:

- **TRADE.** The TRADE [15] encodes the entire dialogue context using a bidirectional GRU and decodes the slot values using a copy-augmented GRU decoder.
- **PIN.** PIN [8] incorporates an interactive encoder to simultaneously capture the dependencies within each turn and across turns, while also introducing slot-level context to extract richer features.
- **SOM-DST.** SOM-DST [16] treats the dialogue states as an explicit fixed-sized memory and introduces a selective overwriting mechanism to update this memory at each turn.
- **SAVN.** SAVN [39] introduces a novel architecture comprising Slot Attention (SA) and Value Normalization (VN) modules to accurately predict the supporting span.
- **TripPy.** TripPy [18] utilizes three copy mechanisms to retrieve slot values from user utterances, system inform memory and previous dialogue states.
- **Seq2Seq.** Seq2Seq [20] addresses the dialogue state tracking task by employing a sequence-to-sequence model.
- **SimpleTOD.** SimpleTOD [32] leverages the GPT-2 model for fine-tuning through multi-task, optimizing all tasks in an end-to-end manner.
- **IC-DST Codex.** IC-DST Codex [33] employs an in-context learning framework to convert DST into text-to-SQL problems, enabling improved utilization of language model prompts.
- **RefPyDST.** RefPyDST [34] employs the in-context learning framework and transforms DST into a Python programming task, thereby leveraging relevant examples and re-weighting methods to improve the model's performance.
- **ChatGPT.** The model [23] leverages the generative capacity of ChatGPT to address the dialogue state tracking task by employing a prompt learning approach.

5. Evaluation and results

5.1. Basic results

To thoroughly evaluate the effectiveness of our model, we conduct two sets of fundamental experiments. The first set involves comparing our model against all baselines using traditional evaluation metrics, while the second set compares our model with SOM-DST using novel evaluation metrics. In accordance with the evaluation metrics specified in MultiWOZ 2.4, we conduct a comprehensive comparison of our model against all baseline models. The performance of the models on the dataset is presented in Table 2, where Slot(Acc) and Turn(Acc) metrics are utilized to assess the state tracking performance and turn-level joint accuracy, respectively. The results reported in Table 2 are derived from either the MultiWOZ 2.4 [7] dataset or the corresponding paper. For metrics that are not reported in their paper, we employ the

Table 2
Main Results on MultiWOZ 2.4 Dataset.

Model	Slot(Acc)	Turn(Acc)
TRADE	97.62	55.05
PIN	98.02	58.92
SOM-DST	98.38	66.78
SAVN	98.05	60.55
TripPy	97.94	59.62
SimpleTOD	–	66.78
Seq2Seq	–	67.1
IC-DST Codex	–	62.4
RefPyDST	–	65.2
ChatGPT	98.12	64.23
Ours	**98.46**	**68.62**

Table 3
Detail comparison with SOM-DST on MultiWOZ 2.4 Dataset.

	SOM-DST	MGIA
Slot(Acc)	98.46	98.46
Slot(Recall)	93.67	94.53
Slot(F1)	94.78	94.77
Operator(Acc)	88.59	88.79
Turn(Acc)	67.24	68.62
Dialogue(Acc)	48.05	50.35

default value "-" to signify this absence. As Slot(Acc) and Turn(Acc) do not provide a comprehensive assessment of model performance, we conduct a detailed comparison of the SOM-DST model on the dataset, and the corresponding results are presented in Table 3.

Based on the results presented in Table 2, we have made the following observations: (1) Our model demonstrates a substantial improvement in performance when compared to the baseline models. Specifically, our model exhibits improvements of 0.08% and 1.52% in Slot(Acc) and Turn(Acc) metrics, respectively, compared to the previously reported results. Despite the utilization of more powerful pre-trained models by the latest methods, our model demonstrates superior performance compared to them. The improved utilization of domain features between states is attributed to our multi-domain gate module. By achieving a domain prediction accuracy of over 97%, our multi-domain gate module ensures more precise forecasting of state operators. (2) Our model exhibits a significant improvement of 8.07% in Turn(Acc) compared to SAVN, which shares a similar framework based on independent-encoder slots. In contrast to SAVN, which solely relies on slot attention (SA), our approach introduces interactive dual attention. This novel approach entails integrating utterance context semantics into slot features as well as incorporating slot features into the context semantics. Subsequently, following the slot attention mechanism, we integrate a multi-domain gate module to enhance the extraction of domain-specific slot features. (3) TRADE, PIN, and TripPy all necessitate iterating through each slot to obtain its corresponding value. The results of these models not only exhibit the lowest performance in terms of Turn(Acc) but also yield the poorest result in Slot(Acc). This is likely because these models focus on individual slots in isolation, thereby disregarding the interdependencies among them. Similarly, the absence of considering slot correlations in SAVN contributes to its relatively lower performance compared to SOM-DST and our model in terms of Slot(Acc).

Our paper extends the comparison between MGIA and SOM-DST by evaluating them on six distinct metrics to demonstrate their effectiveness in addressing the MDST task. The results of this comprehensive analysis are presented in Table 3. Based on the data presented in Table 3, our model demonstrates a performance improvement of 1.38% and 2.3% over SOM-DST in terms of Turn(Acc) and Dialogue(Acc) metrics, respectively. The model achieves comparable performance to SOM-DST in terms of Slot (Acc) and Slot (F1). By leveraging the multi-domain gate, our model has the capability to filter out irrelevant slots in the current utterance, resulting in an overall performance improvement. The model demonstrates a 0.2% improvement in Operator (Acc) compared to SOM-DST, resulting in MGIA surpassing SOM-DST by 0.87% on Slot (Recall). It indicates that our model generates slot values with greater accuracy.

The preliminary findings indicate that our model outperforms the best model on the MultiWOZ 2.4 dataset. Subsequently, to demonstrate the effectiveness of the multi-domain gate and interactive dual attention mechanisms in handling multi-domain tasks, we undertake a series of experiments to thoroughly analyse our model from diverse perspectives.

5.2. Ablation experiments

In order to demonstrate the effectiveness of MGIA, we perform ablation experiments specifically targeting the multi-domain gate and interactive dual attention mechanisms proposed in this paper. The interactive dual attention module consists of two components: utterance attention and slot attention.

To demonstrate the efficacy of the multi-domain gate, we conduct an experiment where we exclude the multi-domain gate module, and the corresponding results are presented in Table 4 under the row *w/o MG*. The incorporation of the multi-domain gate allows the model to selectively focus on updates to slot values within the current domain, utilizing the domain features extracted from the dialogue utterances. Thus, the removal of the multi-domain gate module leads to a decrease of 2.31% and 1.19% in Slot (Recall) and Operator (Acc) respectively in our model. It indicates that the utilization of the multi-domain gate enables the model to effectively filter out irrelevant slots based on the current utterance, leading to an improvement in model performance. The model exhibits a decrease of 4.82% and 4.7% in Turn(Acc) and Dialogue(Acc), respectively. However, our model surpasses SAVN with slot attention in performance because of the inclusion of interactive dual attention, which effectively captures the relationship between dialogue utterances and slots.

Subsequently, we evaluate the effectiveness of the interactive dual attention module. When the utterance attention and slot attention are removed individually, we observe a significant decrease in the performance of model, as demonstrated in Table 4 under *w/o utt att* and *w/o slot att* conditions. The removal of utterance attention results in a 2.89% decrease in Dialogue(Acc) compared to MGIA. Due to the absence of utterance attention, the model *w/o utt att* fails to capture the contextual semantics within the dialogue history, resulting in a decrease in the overall dialogue accuracy performance. The utterance attention module plays a crucial role in modelling the correlation between dialogue history, enabling a better understanding of the dialogue context. By removing the slot attention module, the model loses the ability to capture the interdependencies among the slots, which leads to a decline in the accuracy of the slot operators. The slot attention module plays a crucial role in capturing the informative features shared between different slots, enabling a more accurate prediction of slot values. In contrast to MGIA, the removal of the slot attention component (*w/o slot att*) leads to a reduction of 1.63% in Operator(Acc), thereby resulting in a 4.61% decrease in Turn(Acc). When the entire interactive dual attention module is removed, the model experiences a substantial decrease in performance. Specifically, when excluding the interactive dual attention module, the Operator (Acc), Turn (Acc), and Dialogue (Acc) metrics for the *w/o IA* decrease by 2.88%, 7.04%, and 6.51% respectively compared to the MGIA model. This indicates that the incorporation of interactive dual attention enables the model to capture semantic features between utterances and slots more effectively, thereby enhancing the accuracy of slot operators and dialogues.

Table 4
Ablation experiments.

Model	Slot(Acc)	Slot(Recall)	Slot(F1)	Operator(Acc)	Turn(Acc)	Dialogue(Acc)
w/o IA	97.99	91.75	93.36	85.91	61.58	43.84
w/o utt att	98.33	93.94	94.38	87.32	65.73	46.35
w/o slot att	98.18	92.88	94.07	87.16	64.01	44.24
w/o MG	98.22	91.75	94.19	87.6	63.8	45.65
MGIA	98.46	94.53	94.77	88.79	68.62	50.35

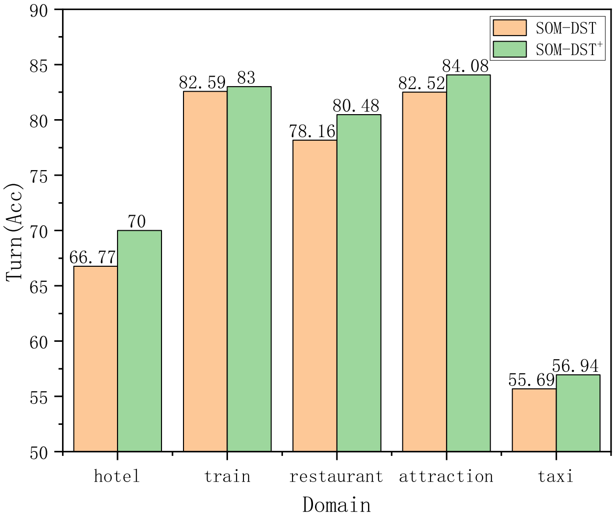


Fig. 5. Turn(Acc) performance of SOM-DST and SOM-DST$^+$ with the introduction of the multi-domain gate on 5 domains.

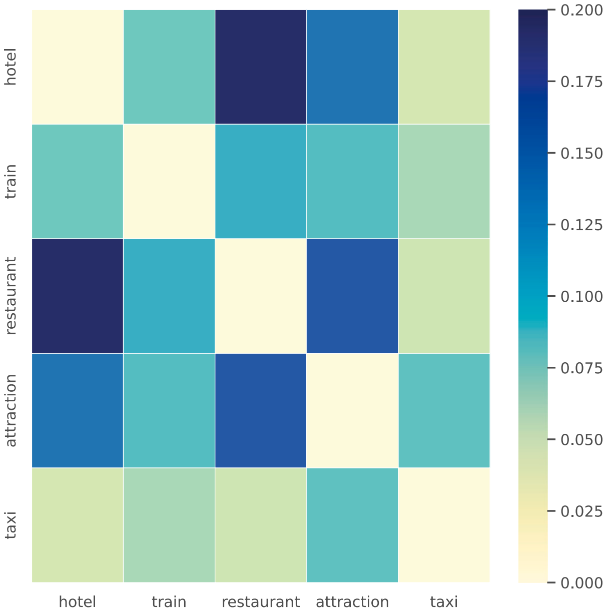


Fig. 6. Co-occurrence probability of 5 domains appearing in the same dialogue utterance on the MultiWOZ 2.4 dataset.

5.3. Effectiveness and scalability of multi-domain gate

To validate the efficacy and scalability of the multi-domain gate, we incorporate the multi-domain gate module into the SOM-DST model, resulting in a variant called SOM-DST$^+$. As the current study places a greater emphasis on turn-level accuracy, we conduct a comparison of the Turn(Acc) metric between SOM-DST and SOM-DST$^+$ across five domains, and the findings are presented in Fig. 5. The figure clearly demonstrates that the introduction of the multi-domain gate module in SOM-DST$^+$ has resulted in significant performance improvements across all five domains, surpassing the performance of SOM-DST. The notable improvements are observed particularly in the hotel, restaurant, and attraction domains, with performance increases of 3.23%, 2.32%, and 1.56%, respectively. On one hand, Table 1 shows a larger dataset sample available for these three domains. On the other hand, these three domains frequently co-occur within the same dialogue utterances, forming multi-domain dialogues. Consequently, the utilization of the multi-domain gate module leads to a notable enhancement in model performance for these specific domains. To visually depict the co-occurrence relationship between domains, we present a graphical representation of the probabilities of domains appearing together in the same dialogue utterance. It is illustrated in Fig. 6. As depicted in the figure, the co-occurrence between hotel, restaurant, and attraction domains exhibits the highest probability. The results indicate that the multi-domain gate module demonstrates a more substantial performance improvement for multi-domain dialogues. It effectively enhances the incorporation of domain-related information while effectively filtering out irrelevant domain states. As a result, the overall model performance is significantly enhanced. The findings from SOM-DST$^+$ provide additional evidence that the multi-domain gate module is not only effective for our model but also holds the potential to enhance the performance of existing models in the context of MDST tasks. This demonstrates the scalability and generalizability of the multi-domain gate module.

5.4. Effectiveness of interactive dual attention

To validate the efficacy of interactive dual attention in MGIA, we evaluate the model's performance on Turn(Acc) and Dialogue(Acc) by incrementally incorporating the number of turns in the dialogue history. The results are illustrated in Fig. 7. We conduct a comparative analysis of the performance between the MGIA and *w/o IA* models as the number of turns in the dialogue history increases. As depicted in Fig. 7 left, the performance of MGIA on Turn(Acc) exhibits a consistent upward trend with an increase in the number of turns in the dialogue history. The incorporation of interactive dual attention facilitates the modelling of bidirectional associations between semantics and slots in dialogue utterances. Consequently, as the dialogue history expands, the model exhibits enhanced capability in identifying the slot values that correspond to the respective slots. Conversely, the *w/o IA* model demonstrates an inverse pattern, displaying an overall decreasing trend as the number of turns in the dialogue history increases. This phenomenon could be attributed to the increased dialogue history introducing more noise to the *w/o IA* model, consequently leading to a decline in model performance. Likewise, in Fig. 7 right, MGIA exhibits an upward trend in overall model performance as the number of turns in the dialogue history increases. This suggests that providing the model with more dialogue history as input facilitates better completion of the entire conversation. The model *w/o IA* maintains a relatively low level of performance as the number of turns in the dialogue history increases.

5.5. Performance on multiwoz 2.1

To elucidate our model's performance across different datasets, we conducted additional comparisons between MGIA and SOM-DST

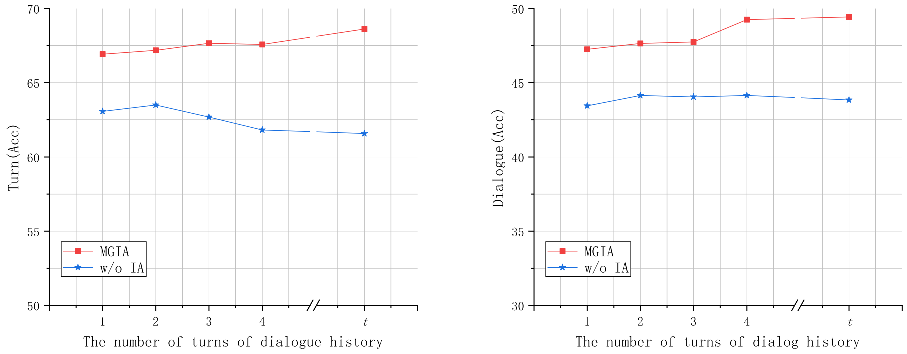


Fig. 7. The performance of MGIA and the *w/o IA* models on Turn(Acc) and Dialogue(Acc) varies with the number of turns in the dialogue history.

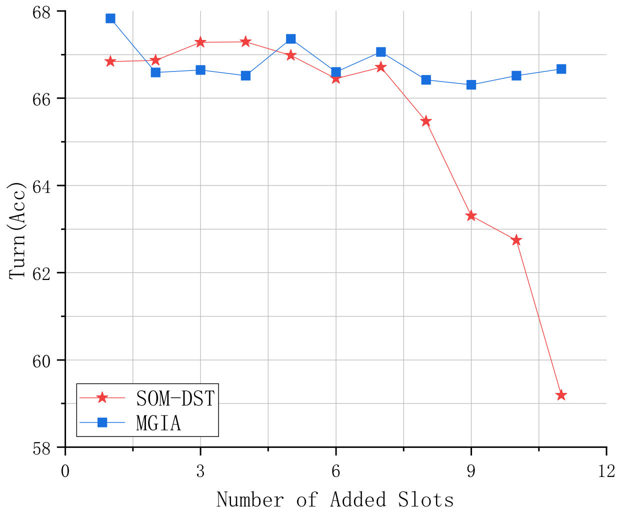


Fig. 8. The impact of adding new slots on Turn(Acc) is compared between SOM-DST and our model.

using the original dataset, MultiWOZ 2.1, evaluating them on six distinct metrics. The outcomes of these comparisons are presented in Table 5. From Table 5, it is evident that MGIA achieves significant performance, surpassing the comparative model SOM-DST in all six metrics. Our model exhibits marginal enhancements over SOM-DST in Slot(Acc) and Slot(F1). However, due to a 0.61% improvement in Operator(Acc), the model achieves a notable 0.9% enhancement in Slot(Recall) performance. Our model achieves notable enhancements of 1.19% in Turn (Acc) and 2.2% in Dialogue (Acc) by utilizing multi-domain gate and interactive dual attention. These components enable our model to capture domain-specific information within conversations and effectively model the relationships between semantics and slots. Simultaneously, we note a significant degradation in the model's performance when evaluated on MultiWOZ 2.1 compared to its performance on MultiWOZ 2.4. This can be attributed to the fact that the most recent dataset has rectified labelling errors within the validation and test sets of MultiWOZ 2.1. Consequently, despite the model's proficiency in learning conversation features and generating accurate predictions, its performance might still suffer from the presence of labelling errors. In this paper, to mitigate this issue and thereby enhance the robustness of our model validation, our comparisons are centred on the most recent and rectified MultiWOZ 2.4 dataset.

5.6. Challenges of multi-domain tasks

In the task of multi-domain dialogue state tracking, the number of slots exhibits an upward trend corresponding to the increase in the number of domains. Models like SOM-DST face the challenge of exceeding pre-trained input limits as the number of slots increases, as they require the concatenation of all dialogue states and history. The addition of each new slot introduces a minimum of 5 additional words to the input, leading to an increase in the input length. We compare the experimental results of MGIA and SOM-DST as the number of slots increases. Fig. 8 presents the Turn(Acc) results for our model and SOM-DST. Based on the observations from Fig. 8, it can be noted that our model exhibits consistent performance across the addition of new slots, with a marginal variation of within 2% in Turn(Acc) results. SOM-DST similarly demonstrates a minimal variation in performance until the addition of 7 slots. However, starting from the inclusion of 8 slots, the performance experiences a rapid decline, exhibiting a decrease of 8.1% by the time 11 slots are added. Due to the concatenation of all states and dialogue history into the encoder in SOM-DST, the increasing length of states has an adverse effect on the length of dialogue history, resulting in a decrease in model performance. In contrast, our model employs an encoder-independent slot matrix, where the addition of new slots does not impact the original input length. Thus, our model achieves a Turn (Acc) of 66.67% even after incorporating 11 new slots, whereas SOM-DST exhibits a significant decline, reaching only 59.19%.

Table 5
Detail comparison with SOM-DST on MultiWOZ 2.1 Dataset.

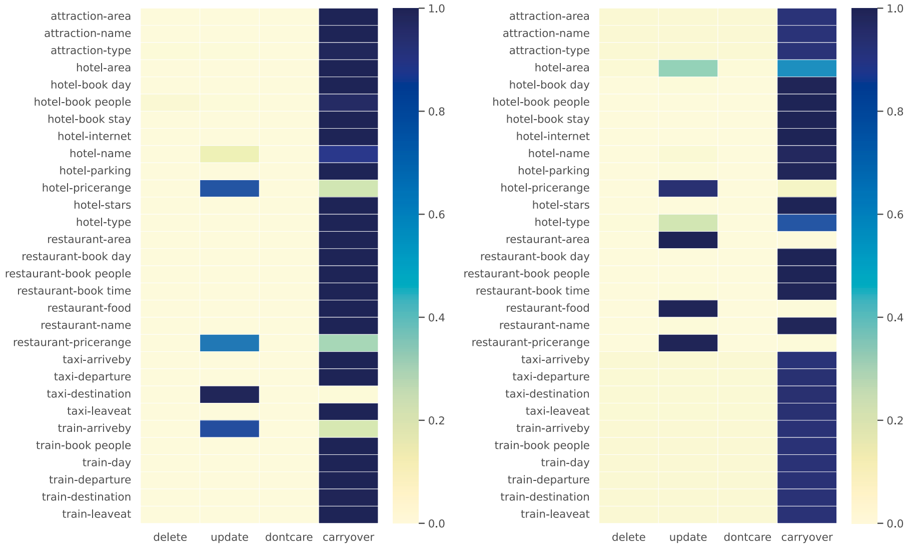
	SOM-DST	MGIA
Slot(Acc)	97.25	97.27
Slot(Recall)	90.37	91.27
Slot(F1)	91.36	91.63
Operator(Acc)	79.36	79.97
Turn(Acc)	51.76	52.95
Dialogue(Acc)	26.13	28.33

5.7. Case study

For qualitative analysis, we conduct a case study by selecting a sample of multi-domain dialogues from MultiWOZ 2.4, comparing the performance of MGIA and SOM-DST. Table 6 presents the dialogue context along with the predicted states by SOM-DST and MGIA for the 2 turns. The dialogue context for the first turn is limited to the hotel domain, enabling us to validate the model's predictions in the context of a traditional single-domain dialogue. The second turn of the dialogue encompasses two domains, namely hotel and restaurant, thus serving as a demonstration of the model's capability to track the state of multi-domain dialogues. Based on the findings presented in Table 6, both SOM-DST and MGIA accurately predict the first turn of single-domain dialogue states. In the case of the 2nd turn of multi-domain dialogue context, MGIA continues to demonstrate precise prediction of the dialogue states. However, SOM-DST exhibits a series of prediction errors. Primarily, there exist inconsistencies in mapping the slots

Table 6
In the case study of multi-domain dialogue state tracking, we examine a two-turn dialogue. The dialogue states are presented below, with **bold** indicating incorrectly predicted states. The multi-domain gate value denotes the domain probability predicted by the model for the turn, while the probability in *italics* represents the domain finally predicted by the model.

Turn	Dialogue context	SOM-DST	MGIA	Multi-domain gate
1	[System]: [User]: I need to find a hotel with a 3 star rating that includes free wifi.	hotel-internet: yes hotel-stars: 3	hotel-internet: yes hotel-stars: 3	hotel: *0.9999* train: 0.0 restaurant: 0.0003 attraction: 0.0 taxi: 0.0
2	[System]: I have 5 options for you, located all over town. Do you have a certain area or price range in mind? [User]: I want Chinese food in cheap price range in west side of town. A 3 star hotel that is expensive and includes wifi. Also the hotel address, area, and postcode please.	hotel-internet: yes hotel-stars: 3 **hotel-pricerange: cheap** **restaurant-pricerange: west** **taxi-destination: Chinese** **train-arriveby: west**	hotel-internet: yes hotel-stars: 3 hotel-pricerange: expensive restaurant-area: west restaurant-food: Chinese restaurant-pricerange: cheap	*hotel: 0.8582* train: 0 *restaurant: 0.9470* attraction: 0 taxi: 0.0003



(a) SOM-DST.

(b) MGIA.

Fig. 9. Visualization results of all slot operators of SOM-DST and MGIA in multi-domain dialogue state tracking.

and corresponding values. Specifically, the correct value for *"hotel-pricerange"* should be *"expensive"*, while for *"restaurant-pricerange"*, it should be *"cheap"*. The issue arises due to the model's inadequate ability to capture semantic relationships between the dialogue context and slots. Secondly, the dialogue states predicted by SOM-DST fail to include the predictions for the slots *"restaurant-area"* and *"restaurant-food"*. Due to the insufficient capture of correlations between the dialogue context and the slots *"restaurant-area"* and *"restaurant-food"*, the model fails to accurately predict the slot operators, thereby neglecting the necessary updates for these two slots. The MGIA model proposed in this paper utilizes an interactive dual attention module to address the aforementioned issues effectively. By capturing the semantic relationships between dialogue utterances and slots, the model establishes crucial correlations between them, thereby enhancing its capability to handle these challenges. Finally, the SOM-DST model exhibits erroneous predictions for two unrelated slots, namely *"taxi-destination"* and *"train-arriveby"*. This issue arises due to the erroneous slot operator predictions, falsely indicating that updates are required for these two slots. By utilizing the multi-domain gate, our model accurately identifies the presence of two domains, namely "restaurant" and "hotel", in the current multi-domain dialogue. Consequently, the slot operator is constrained to predict slots exclusively within these two domains.

5.8. Visualization

In order to visualize the impact of the MGIA model on multi-domain dialogues, we generate visualizations of the operators associated with each slot in the second turn of the dialogues from the case study. The resulting visualizations are presented in Fig. 9. In Fig. 9(a), we can observe the outcomes of the slot operator utilized in SOM-DST. Notably, the model exhibits significant attention towards updating the slots *"taxi-destination"* and *"train-arriveby"*. Consequently, the model is compelled to predict the slot values corresponding to these two slots, resulting in the generation of incorrect dialogue states. Fig. 9(b) displays the operator results for our model. The model exclusively concentrates on updating slots within the two correct domains, leading to accurate predictions for the updates of four slots. To demonstrate the effectiveness of the multi-domain gate proposed in this paper, we present in Table 6 the domain probabilities predicted by the model for two turns of dialogues. Based on the data presented in Table 6, it can be observed that during the first turn, the MGIA model significantly

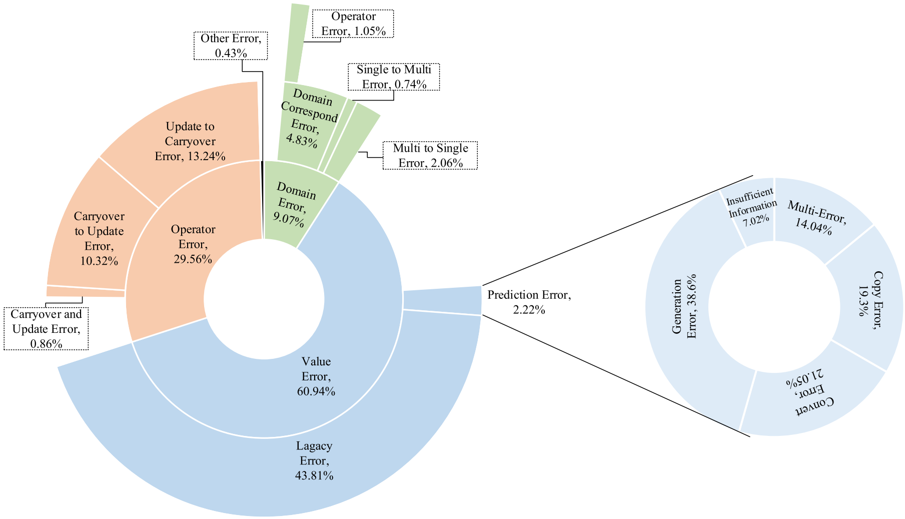


Fig. 10. Various error types and percentages for our model on the MultiWOZ 2.4 dataset.

Table 7
Sample of various error types.

Type	Dialogue context	Prediction state	Ground trueth state
Multi-error	[User]: I would like to eat not **too pricy or cheap** in centre of town .	restaurant-area: centre restaurant-pricerange: **moderate**	restaurant-area: centre restaurant-pricerange: **cheap|moderate**
Copy error	[User]: I am trying to find information about a particular restaurant called **ian hong house**.	restaurant-name: **lan hong house**	restaurant-name: **ian hong house**
Convert Error	[User]: I am looking for a restaurant in the centre of town with a **modest price range**. Can you recommend 1?	restaurant-area: centre restaurant-pricerange: **cheap**	restaurant-area: centre restaurant-pricerange: **moderate**
Generation error	[User]: I would like a taxi from **saint johns college** to pizza hut fen ditton.	taxi-departure: **saint hut college** taxi-destination: pizza hut fenditton	taxi-departure: **saint johns college** taxi-destination: pizza hut fenditton
Insufficient information	[System]: There are 5 options . what area would you like? [User]: I would like an area **near town**.	attraction-type: park attraction-area: **south**	attraction-type: park attraction-area: **centre**
Other error	[User]: I need a taxi to arrive by 17:30 at the **cambridge punte**.	taxi-arriveby: 17:30 taxi-destination: **cambridge punte**	taxi-arriveby: 17:30 taxi-destination: **cambridge punter**

favours predicting the hotel domain. Furthermore, in the second turn of the multi-domain dialogue, the MGIA model predicts two domains with a high probability: 85.82% for the hotel domain and 94.7% for the restaurant domain. As the probabilities of the remaining domains are nearly 0, the multi-domain gate filtering process eliminates all slots in these domains from prediction, leading to a substantial improvement in the final performance of the model.

5.9. Error analysis

In this paper, we conduct an analysis of error cases in our model using the MultiWOZ 2.4 dataset, aiming to gain insights into the limitations of the current model. The results of this analysis are presented in Fig. 10. We categorize the errors into four main types, specifically domain errors, value errors, operator errors, and other errors. Based on the left side of Fig. 10, it can be observed that domain errors constitute 9.07% of all the errors. Our further analysis indicates that 4.83% of the domains are associated with incorrect predictions, leading to 1.05% of the slot operator prediction errors. Due to the imbalanced number of samples between single and multiple domains in the dataset, 2.8% of predictions incorrectly identified the number of domains. Subsequently, operator errors account for 29.56%, with a comparable percentage of errors attributed to excessive updates (10.32%) and missing updates (13.24%), respectively. The results indicate the potential for enhancing the precision of the slot operator, which could contribute to an improved overall prediction accuracy of the model. Finally, value errors constitute the largest portion, comprising 60.94% of the total. Legacy errors comprise 43.81% of the total, signifying that the model consistently incorporates inaccurate dialogue states into subsequent dialogue turns due to various prediction errors. Therefore, recent studies have begun exploring the rectification of erroneous dialogue states in subsequent turns, leveraging the dialogue history as a basis for correction [29,43]. We further investigate the 2.22% state prediction error and present the corresponding plot on the right side of Fig. 10. For each type of error, we provide specific examples in Table 7. Generation errors, copying errors, and conversion errors constitute 38.6%, 19.3%, and 21.05% of the total errors, respectively. As observed in Table 7, these error types occur during the decoding phase, leading to the generation of states that are inconsistent with the ground truth labels. In this paper, we employed the conventional GRU as a decoder for state generation. However, the generation capability can be further enhanced by subsequent integration with generative large language models. Insufficient information and conversion errors may necessitate the incorporation of external knowledge to facilitate the model in gaining a deeper understanding of potential knowledge representations present in user utterances.

6. Conclusion

In this paper, we present the Multi-Domain Gate and Interactive dual Attention (MGIA) model designed for multi-domain dialogue state tracking tasks. The incorporation of the multi-domain gate allows the model to effectively capture domain-specific features by treating the domain as an index, thereby reinforcing domain-relevant information and filtering out domain-irrelevant states. Simultaneously, the multi-domain gate effectively addresses the challenge of handling multiple domains present in a single turn of the dialogue. Interactive dual attention is a variant of the vanilla transformer mechanism, designed to model the bidirectional correlations between dialogue history and slots. To overcome the limitations of input constraint and computational inefficiency in existing methods for addressing multi-domain tasks, our model adopts an independent-slot framework. The slot matrix is obtained by vertically concatenating all slots and performing a single interaction with the dialogue history to obtain updates for all slots, resulting in excellent scalability. We conduct a validation of our model using the most recent multi-domain dialogue dataset, MultiWOZ 2.4. To comprehensively assess the efficacy of our model and baselines, we have expanded upon the original evaluation metrics by introducing extensions to Slot (Recall), Operator (Acc), and Dialogue (Acc) metrics. Subsequent experiments investigate the efficacy of the multi-domain gate and interactive dual attention modules. Furthermore, the functions of each module are illustrated through visualization and case studies.

CRediT authorship contribution statement

Xu Jia: Writing – review & editing, Writing – original draft, Methodology, Conceptualization. **Ruochen Zhang:** Visualization, Data curation. **Min Peng:** Writing – review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

This work was supported by the National Key Research and Development Program of China (No. 2021ZD0113304), the General Program of Natural Science Foundation of China (NSFC) (Grant No. 62072346), Founded by Joint&Laboratory on Credit Technology.

References

[1] J.D. Williams, A. Raux, M. Henderson, The dialog state tracking challenge series: A review, Dialogue Discourse 7 (3) (2016) 4–33.

[2] Q. Liu, G. Bai, S. He, C. Liu, K. Liu, J. Zhao, Heterogeneous relational graph neural networks with adaptive objective for end-to-end task-oriented dialogue, Knowl.-Based Syst. 227 (2021) 107186.

[3] M. Zhao, L. Wang, Z. Jiang, R. Li, X. Lu, Z. Hu, Multi-task learning with graph attention networks for multi-domain task-oriented dialogue systems, Knowl.-Based Syst. 259 (2023) 110069.

[4] V. Balaraman, S. Sheikhalishahi, B. Magnini, Recent neural methods on dialogue state tracking for task-oriented dialogue systems: A survey, in: Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue, 2021, pp. 239–251.

[5] L. Xiang, Y. Zhao, J. Zhu, Y. Zhou, C. Zong, Zero-shot language extension for dialogue state tracking via pre-trained models and multi-auxiliary-tasks fine-tuning, Knowl.-Based Syst. 259 (2023) 110015.

[6] P. Budzianowski, T.-H. Wen, B.-H. Tseng, I. Casanueva, S. Ultes, O. Ramadan, M. Gasic, 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, 2018, pp. 5016–5026.

[7] F. Ye, J. Manotumruksa, E. Yilmaz, MultiWOZ 2.4: A multi-domain task-oriented dialogue dataset with essential annotation corrections to improve state tracking evaluation, in: Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue, 2022, pp. 351–360.

[8] J. Chen, R. Zhang, Y. Mao, J. Xu, Parallel interactive networks for multi-domain dialogue state generation, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, 2020, pp. 1921–1931.

[9] T. Hong, J. Cho, H. Yu, Y. Ko, J. Seo, Knowledge-grounded dialogue modelling with dialogue-state tracking, domain tracking, and entity extraction, Comput. Speech Lang. 78 (2023) 101460.

[10] F. Ye, J. Manotumruksa, Q. Zhang, S. Li, E. Yilmaz, Slot self-attentive dialogue state tracking, in: Proceedings of the Web Conference 2021, 2021, pp. 1598–1608.

[11] F. Ye, X. Wang, J. Huang, S. Li, S. Stern, E. Yilmaz, Metaassist: Robust dialogue state tracking with meta learning, in: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, 2022, pp. 1157–1169.

[12] N. Mrkšić, D.Ó. Séaghdha, T.-H. Wen, B. Thomson, S. Young, 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), 2017, pp. 1777–1788.

[13] V. Zhong, C. Xiong, R. Socher, Global-locally self-attentive encoder for dialogue state tracking, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, 2018, pp. 1458–1467.

[14] J. Xu, D. Song, C. Liu, S.C. Hui, F. Li, Q. Ju, X. He, J. Xie, Dialogue state distillation network with inter-slot contrastive learning for dialogue state tracking, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2023, pp. 13834–13842.

[15] C.-S. Wu, A. Madotto, E. Hosseini-Asl, C. Xiong, R. Socher, P. Fung, Transferable multi-domain state generator for task-oriented dialogue systems, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 808–819.

[16] S. Kim, S. Yang, G. Kim, S.-W. Lee, Efficient dialogue state tracking by selectively overwriting memory, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 567–582.

[17] S. Gao, A. Sethi, S. Agarwal, T. Chung, D. Hakkani-Tur, Dialog state tracking: A neural reading comprehension approach, in: Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue, 2019, pp. 264–273.

[18] M. Heck, C. van Niekerk, N. Lubis, C. Geishauser, H.-C. Lin, M. Moresi, M. Gasic, 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, 2020, pp. 35–44.

[19] X. Li, Q. Li, W. Wu, Q. Yin, Generation and extraction combined dialogue state tracking with hierarchical ontology integration, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, 2021, pp. 2241–2249.

[20] J. Zhao, M. Mahdieh, Y. Zhang, Y. Cao, Y. Wu, Effective sequence-to-sequence dialogue state tracking, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, 2021, pp. 7486–7493.

[21] H. Jeon, G.G. Lee, Schema encoding for transferable dialogue state tracking, in: Proceedings of the 29th International Conference on Computational Linguistics, 2022, pp. 355–366.

[22] P. Lesci, Y. Fujinuma, M. Hardalov, C. Shang, L. Marquez, Diable: Efficient dialogue state tracking as operations on tables, in: Findings of the Association for Computational Linguistics, 2023, pp. 9697–9719.

[23] W. Pan, Q. Chen, X. Xu, W. Che, L. Qin, A preliminary evaluation of chatgpt for zero-shot dialogue understanding, 2023, arXiv preprint arXiv:2304.04256.

[24] M. Heck, N. Lubis, B. Ruppik, R. Vukovic, S. Feng, C. Geishauser, H.-C. Lin, C. van Niekerk, M. Gašić, Chatgpt for zero-shot dialogue state tracking: A solution or an opportunity? in: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, 2023, pp. 936–950.

[25] F. Ye, Y. Feng, E. Yilmaz, ASSIST: Towards label noise-robust dialogue state tracking, in: Findings of the Association for Computational Linguistics, 2022, pp. 2719–2731.

[26] L. Yang, J. Li, S. Li, T. Shinozaki, Multi-domain dialogue state tracking with disentangled domain-slot attention, in: Findings of the Association for Computational Linguistics, 2023, pp. 4928–4938.

[27] L. Yang, J. Li, S. Li, T. Shinozaki, Multi-domain dialogue state tracking with top-k slot self attention, in: Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue, 2022, pp. 231–236.

[28] Y. Feng, A. Lipani, F. Ye, Q. Zhang, E. Yilmaz, Dynamic schema graph fusion network for multi-domain dialogue state tracking, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, 2022, pp. 115–126.

[29] H. Zhang, J. Bao, H. Sun, Y. Wu, W. Li, S. Cui, X. He, MoNET: Tackle state momentum via noise-enhanced training for dialogue state tracking, in: Findings of the Association for Computational Linguistics, 2023, pp. 520–534.

[30] B. Bebensee, H. Lee, Span-selective linear attention transformers for effective and robust schema-guided dialogue state tracking, in: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, 2023, pp. 78–91.

[31] J. Qiu, Z. Lin, H. Zhang, Y. Yang, Hierarchical temporal slot interactions for dialogue state tracking, Neural Comput. Appl. 35 (8) (2023) 5791–5805.

[32] E. Hosseini-Asl, B. McCann, C.-S. Wu, S. Yavuz, R. Socher, A simple language model for task-oriented dialogue, Adv. Neural Inf. Process. Syst. 33 (2020) 20179–20191.

[33] Y. Hu, C.-H. Lee, T. Xie, T. Yu, N.A. Smith, M. Ostendorf, In-context learning for few-shot dialogue state tracking, in: Findings of the Association for Computational Linguistics, 2022, pp. 2627–2643.

[34] B. King, J. Flanigan, Diverse retrieval-augmented in-context learning for dialogue state tracking, in: Findings of the Association for Computational Linguistics, 2023, pp. 5570–5585.

[35] S. Won, H. Kwak, J. Shin, J. Han, K. Jung, BREAK: Breaking the dialogue state tracking barrier with beam search and re-ranking, in: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, 2023, pp. 2832–2846.

[36] J. Guo, K. Shuang, J. Li, Z. Wang, Dual slot selector via local reliability verification for 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 Processin, 2021, pp. 139–151.

[37] H. Yu, Y. Ko, Enriching the dialogue state tracking model with a asyntactic discourse graph, Pattern Recognit. Lett. 169 (2023) 81–86.

[38] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2019, pp. 4171–4186.

[39] Y. Wang, Y. Guo, S. Zhu, Slot attention with value normalization for multi-domain dialogue state tracking, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, 2020, pp. 3019–3028.

[40] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017).

[41] J. An, S. Cho, J. Bang, M. Kim, Domain-slot relationship modeling using a pre-trained language encoder for multi-domain dialogue state tracking, IEEE/ACM Trans. Audio Speech Lang. Process. 30 (2022) 2091–2102.

[42] M. Eric, R. Goel, S. Paul, A. Sethi, S. Agarwal, S. Gao, A. Kumar, A. Goyal, P. Ku, D. Hakkani-Tur, MultiWOZ 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines, in: Proceedings of the 12th Language Resources and Evaluation Conference, 2020, pp. 422–428.

[43] H. Xie, H. Su, S. Song, H. Huang, B. Zou, K. Deng, J. Lin, Z. Zhang, X. He, Correctable-DST: Mitigating historical context mismatch between training and inference for improved dialogue state tracking, in: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, 2022, pp. 876–889.