Multi-Domain Dialogue State Tracking via Dual Dynamic Graph with Hierarchical Slot Selector

Anonymous ACL submission

Abstract

Dialogue state tracking aims to maintain user intent as a consistent state across multi-domains to accomplish natural dialogue systems. However, previous researches often fall short in capturing the multiple type of slots and fail to adequately consider the selection of discerning information. The increase in unnecessary information correlates with a decrease in predictive performance. Therefore, the careful selection of high-quality information is imperative. Moreover, considering that the types of essential and available information vary for each slot, the process of selecting appropriate information may also differ. To address these issues, we propose HS2DG-DST, a Hierarchical Slot Selector and Dual Dynamic Graph-based DST. Our model is meticulously designed to differentiate slots and provide maximal information for optimal value prediction. We hierarchically classify slot types based on the multiple properties. The two dynamic graphs in our model supply highly relevant information to each slot. Experimental results on MultiWOZ datasets demonstrate that our model outperforms state-of-the-art models.

1 Introduction

Task-oriented dialogue (TOD) systems are designed to accomplish specific goals, such as providing weather forecasts or making restaurant reservations (Zhang et al., 2020c). Dialogue state tracking (DST) within TOD systems aims to track user intents across various domains consistently.

Previous researches employ ontology-based lexicons to assign relevant values in DST models (Lee et al., 2019; Zhang et al., 2020a). On the other hand, some approaches focus on extracting values based on span labels (Gao et al., 2019; Heck et al., 2020; Chao and Lane, 2019; Lei et al., 2018) or generating values (Wu et al., 2019; Kim et al., 2020; Kumar et al., 2020; Ren et al., 2019).

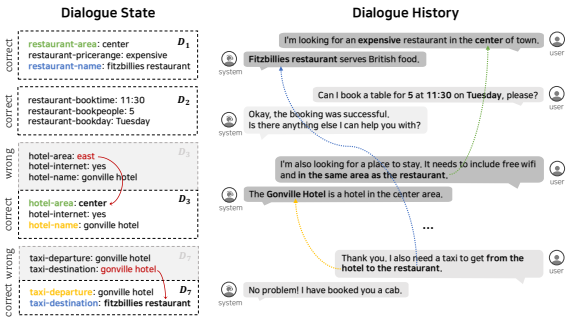

Figure 1: An example of multi-domain conversation (right) and dialogue state tracking process (left) at each turn. The green, blue, and yellow represent co-referential slots in the context, and red emphasizes incorrectly predicted states.

Using a single tracker to predict slots with diverse types overlooks the opportunity to leverage supplementary information, such as possible values. Therefore, previous studies (Zhang et al., 2020a; Zhou and Small, 2019) attempt to classify slot types into categorical and span. Other approaches (Kim et al., 2020; Guo et al., 2021) differentiate slots based on whether they are updated in the current turn or inherit from the previous state. However, these studies only consider two types of slots, which results in the neglect of the possibility that a slot can belong to multiple types, such as both "*update*" and "*span*" types. Consequently, a hierarchical approach is needed to handle slots that can belong to multiple types simultaneously.

Effective co-reference resolution is crucial for contextual understanding in DST. In Figure 1, co-reference resolution is paramount when updating the "*taxi-destination*" slot. The system must adeptly discern the user's intent in referencing the previously mentioned "*restaurant-name*" in statements like "*taxi from the hotel to the restaurant*." Accomplishing this task requires the ability to identify the most relevant information within the dialogue history, particularly the mention of the "*Fitzbillies restaurant*."

However, previous graph-based approaches (Feng et al., 2022; Guo et al., 2022; Zhang et al., 2022; Zeng and Nie, 2020; Zhou and Small, 2019; Lin et al., 2021) neglect the integration of relevant dialogue information into the model. Previous models primarily focus on learning new relationships within the state, taking the dialogue context into consideration. For example, they may suggest correlations between *taxi-destination* and *restaurant-name* slots. Nonetheless, as noted by (Zhang et al., 2022), the emergence of state momentum, indicative of models preserving predicted slot values, may lead to inaccuracies in the previous dialogue state. In such scenarios, depending solely on state relations may result in inaccurate values. In contrast, in our approach, the retrieval of dialogue turns enables precise value prediction, as the dialogue turns themselves contain the correct information. Consequently, the adept retrieval of the most relevant dialogue information is essential for accurately tracking values associated with co-referential slots.

To address these challenges, we propose a novel approach called HS2DG-DST (Hierarchical Slot Selector and Dual Dynamic Graph-based DST). We emphasize that a slot can have both "*update*" and "*span*" types simultaneously. Thus, we introduce a hierarchical slot selector to provide a more detailed classification of slots. Furthermore, we utilizes two dynamic graphs, a value graph and a dialogue graph, to effectively manage semantic dialogue information and provide relevant knowledge to the target slots. These graphs operate akin to information retrieval, tailored to deliver the essential information for the selected slot. Finally, we utilize a fine-grained value-generation method for each target slot, enabling the model to generate values more precisely and accurately. Our contributions can be summarized as follows:

- We introduce a novel framework called HS2DG-DST, designed to predict slot values hierarchically and provide maximal information for fine-grained value prediction.
- We design a dual dynamic graph to assist in information management and enhance the accurate prediction of co-referential slots.
- We conduct experiments on two variations of MultiWOZ datasets. Results show that our proposed model significantly outperforms state-of-the-art models.

2 Related Work

We categorize existing research in DST from two perspectives and introduce a knowledge selection model that inspired the design of our graph model.

2.1 Dialogue State Tracking

In the early stage of DST, researches can be classified into two principal categories: ontology-based DST (Henderson et al., 2014; Nouri and Hosseini-Asl, 2018; Lee et al., 2019; Zhang et al., 2020a) and open-vocabulary-based DST (Zhang et al., 2020b; Gao et al., 2020; Chen et al., 2020; Feng et al., 2021; Kim et al., 2020). For instance, Kim et al. (2020) treat dialogue state as a fixed-size memory to efficiently update slot values. Guo et al. (2021) propose dual slot selection to identify updated slots effectively. In contrast, Zhang et al. (2020a) and Zhou and Small (2019) distinguish slot types based on the existence of a possible value set. Moreover, Zeng and Nie (2020) introduce the state graph representing the dialogue state, and Feng et al. (2022) focus on learning new relationships within the slot by considering the dialogue context. Additionally, Guo et al. (2022) propose a top-k dialogue selection model that leverages updated slot selection and establishes relationships between slots and dialogues. However, previous studies does not adequately consider semantic dialogue information. In our approach, we construct a graph that captures sophisticated relationships between dialogue turns. Moreover, we develop an elaborate approach for handling multiple slot types, resulting in fine-grained value prediction in DST.

2.2 Semantic Document Graphs

In open-domain dialogue systems, incorporating relevant background knowledge is crucial for improving the quality of conversations. Li et al. (2022) argue that previous approaches overlook the inherent semantic connections between sentences in real-world documents. To overcome this limitation, they propose a semantic document graph to capture the implicit connectivity between sentences, enabling the selection of the most relevant knowledge based on the dialogue context. We extend the idea of a semantic graph to DST by treating the entire dialogue history as a document. Rather than representing sentences as concepts, we utilize selected slots to capture the relationships between dialogue turns, facilitating accurate dialogue state prediction through relevant knowledge acquisition.

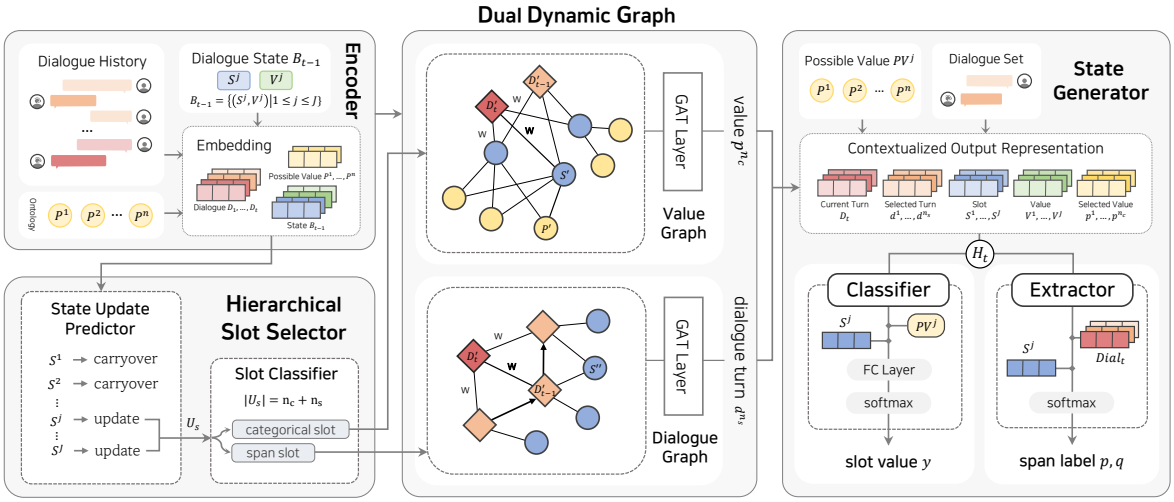

Figure 2: The overview of Dual Dynamic Graph-based DST with a Hierarchical Slot Selector, HS2DG-DST

3 Our Approach: HS2DG-DST

Figure 2 illustrates an overview of our proposed model, HS2DG-DST, comprising four main components: encoder, hierarchical slot selector, dual dynamic graph, and state generator. In this work, we define the problem setting as predicting the dialogue state at each turn t ($t \leq T$). The dialogue state is denoted as $B_t = \{(S^j, V^j) \mid 1 \leq j \leq J\}$, where S^j is the slot name and V^j is the corresponding slot value. Here, J denotes the total number of slots. Similar to Guo et al. (2021), we refer to the concatenation of a domain name and a slot name as a "*slot*" (e.g., *restaurant-area*).

3.1 Encoder

We construct the input by concatenating each dialogue turn D_t and the previous dialogue state B_{t-1} as follows:

$$X_t = [CLS]_t \oplus D_t \oplus B_{t-1} \quad (1)$$

where $[CLS]_t$ is a special token aggregating the input information.

The representation of each dialogue at turn t is denoted as $D_t = R_t \oplus ; \oplus U_t \oplus [SEP]$, where R_t represents the system response, U_t represents the user utterance, and ";" is a special token indicating the boundary between R_t and U_t. $[SEP]$ is used to mark the end of the dialogue turn.

The representation of the state at turn t is $B_t = \mathrm{B}_t^1 \oplus \ldots \oplus \mathrm{B}_t^J$, where $\mathrm{B}_t^j = [SLOT]^j \oplus S^j \oplus - \oplus V_t^j$ represents the j-th slot-value pair. "$-$" is a special token indicating the boundary between a slot and its corresponding value, and $[SLOT]^j$ represents the aggregated information of the j-th slot-value pair.

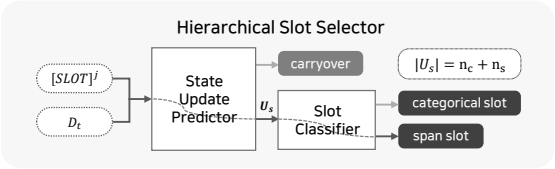

Figure 3: The process of hierarchical slot selection

3.2 Hierarchical Slot Selector

The hierarchical slot selector comprises a state update predictor and a slot classifier. We first determine if a slot needs an update in the current turn and then classify the relevant slots into categorical or span slots. The overall process is illustrated in Figure 3. This hierarchical approach enables us to accurately identify slots with both "*update*" and "*span*" types, facilitating fine-grained value prediction in the subsequent module.

3.2.1 State Update Predictor

This module predicts the slots that need to be updated while other slots inherit their values from the previous dialogue state. We follow the training mechanism described by Guo et al. (2021). We define the set of updated slot indices as:

$$U_s = \{\, j \mid \mathbf{SUP}(S^j) = update \} \quad (2)$$

This module serves two purposes. Firstly, it helps alleviate computational costs by focusing on predicting only the updated slots. Secondly, identifying whether a slot needs updating serves as an indicator of the current dialogue's relevance to that specific slot. This information is essential for constructing the dialogue graph, as discussed in Section 3.3.2 Dialogue Graph, where we provide how the updated slots are utilized.

3.2.2 Slot Classifier

The updated slots can be classified into categorical or span slots based on the number of possible values they can have. Categorical slots, such as "*area*," have a limited number of values {*east, west, south, north, center*}, which helps avoid out-of-vocabulary issues. In contrast, span slots like "*name*" or "*time*" cannot have predetermined values, necessitating the use of a span prediction. Further details can be found in Appendix A.1 Classification of Slots.

We can express the number of slots in each type as follows:

$$|U_s| = n_c + n_s \tag{3}$$

,where n_c is the number of categorical slots, and n_s is the number of span slots.

3.3 Dual Dynamic Graph

The dual dynamic graph consists of a value graph and a dialogue graph, both utilizing graph attention networks (GATs) (Veličković et al., 2018; Li et al., 2021). These graphs are responsible for updating co-referential slots and enhancing value prediction with additional information. The value graph is employed for categorical slots to select the most suitable value from a predefined ontology. Conversely, the dialogue graph is used for span slots to identify the most relevant dialogue turn, leveraging an understanding of the semantic structure within the dialogue context.

3.3.1 Value Graph

The value graph comprises dialogue turn nodes D'_t, slot nodes S', and possible value nodes P'. These nodes allow for bidirectional feature exchange among them. Specifically, possible value nodes are connected to slot nodes when a value is available for a given slot. However, if no value is presented, the slot nodes remain disconnected. Moreover, each dialogue turn node is connected to all slot nodes. The graph structure is visually represented in Figure 2.

At each dialogue turn t, a weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is defined, where the set $\mathcal{V}$ represents the dialogue turn, slot, and possible value nodes, and the set $\mathcal{E}$ represents the connections between these nodes. The graph is represented by a binary symmetric adjacency matrix $\mathcal{M}$ of size $N \times N$, where N denotes the total number of nodes. Each node v_i is associated with a feature vector x_i, and these feature vectors are stored in the matrix $\mathcal{X}$ of size $N \times F$, where F represents the input feature dimension.

We utilize the graph attention mechanism introduced by Lin et al. (2021) to perform graph operations. The initial node features $\mathcal{X}_t^{(0)}$ for the graph attention networks are obtained by concatenating the dialogue turn embedding, slot embedding, and possible value embedding, which are derived from the encoder output. The dialogue turn embedding is obtained from the $[CLS]_t$ token, capturing the dialogue context for each turn, while the slot embedding is obtained from the $[SLOT]^j$ token, representing the slot context. The possible value embedding is initialized by tokenizing the candidate value representations.

After conducting the graph operations, we extract an attention embedding from the final tensor $\mathcal{X}_t^{(L)}$. We utilize this attention embedding to capture the relevance score between nodes. And then, the index of the highest attention score is used to determine the most appropriate possible value for the updated slot, represented by p^{n_c}.

3.3.2 Dialogue Graph

In our dialogue graph design, we are inspired by the work of Li et al. (2022), who proposed a semantic document graph for selecting relevant knowledge from documents. They represent sentence nodes by multiple concepts, and the connections between these concepts reflect the semantic relationships within the sentences. We adapt this approach by introducing a semantic dialogue graph, where we incorporate updated slots (S^j where $j \in U_s$) as similar to concepts within dialogue turns. Updating a slot in a dialogue turn indicates the presence of relevant information in that turn. Therefore, the updated slots and dialogue turns are strongly correlated. By leveraging these updated slots, we construct a graph representing each dialogue's meaning and enhancing the semantic connections between dialogue turns.

The dialogue graph comprises dialogue turn nodes D'_t and updated slot nodes S''. The graph connectivity is established through three types of edges: 1) Edges between previous dialogue turn nodes D'_{t-1}: These edges are sequentially connected, making the graph aware of contextual turn information. 2) Edges between dialogue turn nodes and updated slot nodes: These edges connect each dialogue turn node only to its corresponding updated slot nodes, facilitating the effective representation of semantic information. 3) Edges between

the current dialogue turn node D'_t and all other dialogue turn nodes D'_{t-1}: These edges enable the current turn node to assess the correlations with the previous turn nodes.

The graph attention mechanism is the same in the value graph. And the initial node features $\mathcal{X}_t^{(0)}$ are obtained by concatenating the dialogue turn embedding and the slot embedding. By learning connections between each node, the dialogue graph captures semantic relationships between dialogue turns and provides relevant information to the target slots. The output of the dialogue graph is the most pertinent dialogue turn d^{n_s} to the target slot.

3.4 State Generator

The selected possible values p^{n_c} and dialogue turns d^{n_s} are combined with the current turn D_t and the previous dialogue state B_{t-1} to update the state jointly. This is achieved by concatenating them to form a new input sequence, denoted as $X = [CLS] \oplus D_t \oplus [SEP] \oplus d^{n_s} \oplus [SEP] \oplus B_{t-1} \oplus [SEP] \oplus p^{n_c}$.

Subsequently, this sequence is fed into a frozen pre-trained language model, specifically ALBERT (Lan et al., 2019) to obtain the contextualized output representation H_t.

3.4.1 Extractor

To predict the values of span slots, we utilize a span-based extraction method. We employ two different linear layers W_s and W_e to predict the start and end labels. The attention-based representation of the j-th slot at turn t, denoted as $H_t([SLOT]_t^j)$, is used in this process. From this, we obtain the representations p and q as follows:

$$p = \text{softmax}(W_s H_t([SLOT]_t^j)^\top) \quad (4)$$

$$q = \text{softmax}(W_e H_t([SLOT]_t^j)^\top) \quad (5)$$

The position of the maximum value in each p and q corresponds to the predicted start and end positions of the slot value. Furthermore, we define Dial_t as the concatenation of D_t and d^{n_s} from the input sequence X. The span slot value is then extracted from the dialogue sentence using $\text{Dial}_t[p:q]$.

3.4.2 Classifier

For categorical slots, we employ a classification-based method to select an appropriate value. Let PV^j denote the possible value set of the j-th slot. Similar to the extractor, we use the slot representation as attention to the output representation H_t, resulting in $H_t([SLOT]_t^j)^\top$. We then pass this representation through a linear layer W_c to obtain the distribution over PV^j:

$$y = \text{softmax}(W_c H_t([SLOT]_t^j)^\top) \quad (6)$$

We select the slot value corresponding to the maximum value in the distribution. By finding the index using $\text{argmax}(y)$, we can obtain the categorical slot value from the value set PV^j.

3.4.3 Optimization

We utilize cross-entropy loss as the training objectives for the extractor and the classifier during the training process.

$$loss_E = -\frac{1}{|U_s|}\sum_{j}^{|U_s|}(p \log \hat{p} + q \log \hat{q}) \quad (7)$$

$$loss_C = -\frac{1}{|U_s|}\sum_{j}^{|U_s|} y \log \hat{y} \quad (8)$$

Here, $\hat{p}$ and $\hat{q}$ are the target values representing the proportion of all possible start and end positions. And $\hat{y}$ is the target indicating the probability of candidate values.

4 Experiments

4.1 Datasets and Metrics

4.1.1 Datasets

MultiWOZ (Budzianowski et al., 2018) is a multi-domain human-human written dialogue dataset that contains over 10K dialogues across 8 domains. It is one of the most popular benchmarks in the DST literature. We conducted experiments on two variants of the datasets: MultiWOZ 2.1 (Eric et al., 2020) and MultiWOZ 2.2 (Zang et al., 2020). The labels and utterances have been refined in subsequent versions. In particular, MultiWOZ 2.2 redefined the datasets by dividing all slots into two types: non-categorical and categorical.

4.1.2 Metrics

Joint Goal Accuracy (JGA) refers to the accuracy of the dialogue state in each turn. It compares the predicted dialogue state to the ground truth at every turn, and it is correct only if all the predicted slot values exactly match the ground truth.

Slot Accuracy (SlotAcc) considers individual slot-level accuracy. It measures the ratio of successful slot value predictions among all the slots of each dialogue in the ground truth.

Model	MultiWOZ 2.1		MultiWOZ 2.2			
	JGA	Slot Acc	JGA	Slot Acc	Cate-joint	Span-joint
TRADE	45.60	96.55	45.40	-	62.80	66.60
DSTQA	51.17	97.21	-	-	-	-
DS-DST	51.21	97.35	51.70	-	70.60	70.10
SOM-DST	53.68	97.15	-	-	-	-
TripPy	55.30	97.48	50.71	-	-	-
DST-as-Prompting	56.66	-	57.60	-	-	-
DSS-DST*	60.73	98.05	58.04	97.66	76.32	73.39
DiCoS-DST*	61.02	98.05	61.13	98.06	-	-
Our Model (HS2DG-DST)*	**65.91**	**98.31**	**66.01**	**98.43**	**80.76**	**80.27**

Table 1: Performance comparison of the baseline models. * indicates a result in same experimental setting.

Model	MultiWOZ 2.2
Our Model (HS2DG-DST)	**66.01**
w/o B_{t-1}	63.50 (-2.51)
w/o state update predictor	63.36 (-2.65)
w/o dual dynamic graph	61.52 (-4.49)

Table 2: Ablation study of main components. "*w/o dual dynamic graph*" indicates that the model could not access selected information from both graphs. "*w/o* B_{t-1}" refers to the exclusion of the previous dialogue state as input. And "*w/o state update predictor*" indicates that all slots were updated at every turn.

4.2 Baseline Models

TRADE (Wu et al., 2019) utilizes a copy mechanism, enabling knowledge transfer across domains. **DSTQA** (Zhou and Small, 2019) employs a GAT to learn inter-slot relationships and the questions allowing the model to handle unseen domains. **DS-DST** (Zhang et al., 2020a) proposes a dual strategy that combines categorical and non-categorical slots using a reading comprehension model. **SOM-DST** (Kim et al., 2020) treats the dialogue state as a fixed-size memory and dynamically overwrites it. **TripPy** (Heck et al., 2020) employs three copy mechanisms to extract span values from the dialogue context. **DST-as-Prompting** (Lee et al., 2021) introduces a language modeling approach that utilizes schema-driven prompting to incorporate task-aware history encoding. **DSS-DST** (Guo et al., 2021) proposes a dual slot selector that determines whether each slot needs to be updated. **DiCoS-DST** (Guo et al., 2022) dynamically selects relevant dialogue contents corresponding to each slot.

4.3 Main Results

Table 1 provides a performance comparison between our HS2DG-DST model and other baselines on the MultiWOZ datasets. The best result is highlighted in bold. Our model achieved state-of-the-art performance on both MultiWOZ 2.1 and 2.2 test sets, with JGAs of 65.91% and 66.01%, respectively. Specifically, our model outperformed the previous state-of-the-art by approximately 4.89%p for MultiWOZ 2.1 and 4.88%p for MultiWOZ 2.2 in terms of JGA. In addition, we conducted experiments on slot type classification to MultiWOZ 2.2. In these experiments, "Cate-joint" refers to the JGA specifically for categorical slots, while "Span-joint" represents the JGA for span slots. Our model outperformed existing public models by 80.76% and 80.27%, achieving a lead of 4.44%p and 6.88%p in the cate-joint and span-joint, respectively. We excluded DiCoS-DST (Guo et al., 2022) results from the Table 1 cate-joint and span-joint as there is no available information on its strengths in slot classification. The consistent performance of our model across different slot types can be attributed to our elaborate slot type classification and effective utilization of optimal information (i.e., selected possible values and dialogue turns). For further analysis, please refer to Section 4.5.1 Slot Type Classification.

4.4 Ablation Study

4.4.1 Effect of Main Components

To investigate the effectiveness of the main components, we conducted an ablation study on the MultiWOZ 2.2 with JGA presented in Table 2. The results showed that removing each module led to a decrease in JGA to varying degrees. Without

Graph	MultiWOZ 2.2
both graph in all slot connection	64.99
w/o value graph	64.12
w/o dialogue graph	64.12
dialogue graph in updated slot only	**66.01**
w/o value graph	65.05
value graph in updated slot only	65.83
w/o dialogue graph	62.86

Table 3: Ablation study of graph node connections. we conducted two comparative experiments: *all slot connection* and *updated slot only*. In the former experiment, each dialogue turn node was connected to all slot nodes, while in the latter, it was connected only to the updated slot nodes.

Model	MultiWOZ 2.2
Baseline Retriever	54.05
Our Model (HS2DG-DST)	**66.01**

Table 4: Performance comparison with a baseline retriever model.

the previous dialogue state, the JGA decreased by 2.51%p due to the inability to refer to previously predicted values and address co-referential slots. Additionally, eliminating the state update predictor resulted in a 2.65%p decrease, along with the absence of updated slot information for constructing the dialogue graph and increased memory usage. Removing the dual dynamic graph component, essential for managing co-referential information, led to a significant decrease of 4.49%p in JGA. This finding highlights the critical role of the dual dynamic graph in providing semantic information. Each graph component is responsible for selecting relevant dialogue turns or possible values associated with the target slot. Without these information, the model lacks the necessary context to make precise predictions.

4.4.2 Effect of Graph Node Connections

To evaluate the effectiveness of our proposed graph structure in capturing semantic relationships between each node, we conducted experiments for the impact of different node connections in the dual dynamic graph. Table 3 demonstrated the superior performance of the *all slot connection* graph compared to the baseline models, achieving a 64.99% of JGA. This result proves the effectiveness of including dialogue turn nodes in the graph. Furthermore, in the *update slot only* condition, the dialogue graph achieved the highest performance at 66.01%, while the value graph achieved 65.83%. Connecting only the updated slot nodes led to more precise graph structures for capturing semantic connections between dialogue turns. Moreover, we conducted an ablation study on the separate graph components, specifically the dialogue graph and the value graph. The results showed that even without the value graph, the dialogue graph in the *update slot only* condition performed better than the *all slot connection* condition. Additionally, the JGA of the separate value graph in the *update slot only* condition decreased by 2.97%p. These findings consistently demonstrates the superiority of the dialogue graph over the value graph. The dialgoue graph, which selects the dialog turns, is relatively more informative than the value graph, so it performs better in our experiments.

4.4.3 Effect of Dialogue Graph

To empirically validate the effectiveness of our semantic graph structure, we conducted an experiment comparing it to a baseline retriever. We adopted dense passage retrieval (DPR) (Karpukhin et al., 2020), a renowned retrieval system in the open-domain dialogue systems, as a point of comparison. The dialogue graph can be seen as a retrieval system that benefits from the graph structure to capture relevant dialogue turn indexes. By leveraging the graph as a form of meta-information, our graph model enhances retrieval performance.

We developed a baseline retriever model based on DPR, using ALBERT (Lan et al., 2019) as an encoder and extracting the output embedding from the $[CLS]$ token. The similarity between the current and previous turns was calculated by taking the dot product of their embeddings, and the most similar turn was chosen as the relevant one. Similar to the dialogue graph, the baseline retriever selected the most pertinent dialogue turn d^{m_s}, and the remaining process of dialogue state tracking was the same in both experiments. However, this baseline model did not understand the connections between dialogue turns, focusing only on individual turn embeddings. As shown in Table 4, the baseline retriever achieved a JGA of only 54.05%. This performance was significantly lower, with a degradation of 11.96%p, compared to our proposed model that incorporates more enhanced semantic information between turns. This highlights the substantial improvement achieved by our approach to selecting relevant information.

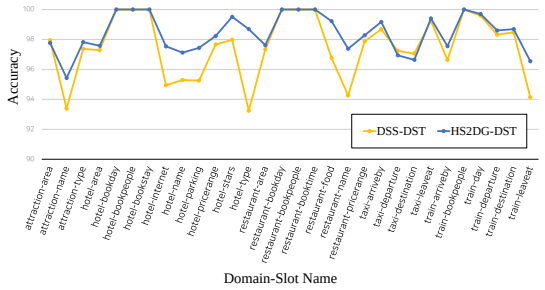

Figure 4: Accuracy per slot name compared to DSS-DST in the test set of MultiWOZ 2.2.

Domain	MultiWOZ 2.2		
	DSS-DST	DiCoS-DST	Our Model
Attraction	79.88	78.79	**80.14**
Hotel	62.47	58.02	**71.75**
Restaurant	75.79	75.14	**81.63**
Taxi	54.84	**56.33**	42.71
Train	76.25	77.26	**79.87**

Table 5: Domain specific accuracy of our model and other baselines on the test data of MultiWOZ 2.2.

4.5 Analysis

4.5.1 Slot Type Classification

To assess the effectiveness of hierarchical slot selector for fine-grained value predictions, we conducted experiments on each slot accuracy compared to DSS-DST. As shown in Figure 4, our model consistently outperformed DSS-DST in terms of slot accuracy. Our model achieved over 95.4% accuracy for all slots, while DSS-DST occasionally fell below 94% accuracy. This demonstrates the stability of our approach in accurately predicting each slot. In Table 1, DSS-DST demonstrated weak performance in span-joint metrics. This suggests a lack of consideration for slot classification in DSS-DST. In contrast, our model hierarchically classifies slots, resulting in the best overall performance with stability and consistency across both metrics. These findings highlight the effectiveness of an elaborate approach in achieving better and more consistent results.

4.5.2 Effect of Dialogue Information

In Table 2, when we exclude all the information provided by the two graphs (w/o dual dynamic graph), the performance is 61.52%, which is similar to the 61.13% of DiCoS-DST. This indicates that the information selected by the graphs directly contributes to performance improvement. In Table 3, when the model use only the dialogue graph (w/o value graph), the performance is 65.05%. This indicates that the dialogue graph alone, which includes dialogue information, achieved a performance of 65%. Furthermore, in order to assess how effectively the proposed graph provides dialogue information, we conducted experiments comparing it to DPR in Section 4.4.3 and the performance is 54.05%. This indicates that if a weak retrieval model provides incorrect dialogue turns, it can have a detrimental impact on performance. In conclusion, the experimental results presented above demonstrate the performance benefits of utilizing dialogue turn information.

4.5.3 Domain-Specific Accuracy

Table 5 presents the domain-specific results, the accuracy measured on subsets of the predicted state specific to each domain. Our model achieved best performance in four domains, with notable improvements in the *hotel* and *restaurant* domains, which have many span slots. However, the performance in the *taxi* domain was comparatively lower than the other domains. Because we extract the span labels from $Dial_t$, the performance of predicting span slots relies heavily on selecting relevant dialogue turns using the dialogue graph. The superior performance of our model in the *hotel* and *restaurant* domains demonstrated the model's effectiveness in selecting relevant dialogue turns. However, in the MultiWOZ datasets,the *taxi* domain frequently emerges in the last turn of a conversation. The larger number of dialogue turns presents challenges in accurately determining its relevance. This may explain the relatively lower performance in the *taxi* domain. The statistical analysis of the graphs can be found in Appendix A.2 Graph Analysis.

5 Conclusion

In this paper, we proposed a novel hierarchical slot selection framework via a dual dynamic graph for multi-domain dialogue state tracking. Our approach involves fine-grained value prediction by classifying slots into multiple types and incorporating complementary knowledge for target slots. The proposed graphs effectively manage semantic information through a semantic-aware graph structure that determines relevant information for target slots. Against the state-of-the-art DST methods, experimental results on two variant multi-domain datasets demonstrate the effectiveness of hierarchical slot selection and dual dynamic graph.

Limitations

This paper proposes HS2DG-DST, a framework that utilizes an elaborate slot classification and optimized information retrieval for value prediction. However, it currently relies solely on extracting span labels from dialogue turns, without incorporating previously predicted values obtained from the previous dialogue state. This approach faces challenges as the number of dialogue turns increases, complicating the selection of relevant dialogues. To mitigate this limitation, there is a need to develop a concise form of dialogue graph. Utilizing a more efficiently summarized form of the graph could offer a solution to this issue.

Ethics Statement

Improving the DST module in dialogue systems can enhance their ability to understand user requirements and increase user satisfaction. Our proposed framework has the potential to enhance DST performance in industrial and commercial dialogue systems. Additionally, the concepts and techniques employed in our frameworks, such as hierarchical slot selection and dual dynamic graph, can be applied to other natural language processing and machine learning applications, leading to performance improvements in various tasks.

References

Samuel Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 2018. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026.

Guan-Lin Chao and Ian Lane. 2019. Bert-dst: Scalable end-to-end dialogue state tracking with bidirectional encoder representations from transformer. *arXiv preprint arXiv:1907.03040*.

Lu Chen, Boer Lv, Chi Wang, Su Zhu, Bowen Tan, and Kai Yu. 2020. Schema-guided multi-domain dialogue state tracking with graph attention neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7521–7528.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, Adarsh Kumar, Anuj Goyal, Peter Ku, and Dilek Hakkani-Tur. 2020. Multiwoz 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 422–428.

Yue Feng, Aldo Lipani, Fanghua Ye, Qiang Zhang, and Emine Yilmaz. 2022. Dynamic schema graph fusion network for multi-domain dialogue state tracking. *arXiv preprint arXiv:2204.06677*.

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

Shuyang Gao, Sanchit Agarwal, Di Jin, Tagyoung Chung, and Dilek Hakkani-Tur. 2020. From machine reading comprehension to dialogue state tracking: Bridging the gap. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 79–89.

Shuyang Gao, Abhishek Sethi, Sanchit Agarwal, Tagyoung Chung, and Dilek Hakkani-Tur. 2019. Dialog state tracking: A neural reading comprehension approach. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 264–273.

Jinyu Guo, Kai Shuang, Jijie Li, and Zihan Wang. 2021. 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 Processing (Volume 1: Long Papers)*, pages 139–151.

Jinyu Guo, Kai Shuang, Jijie Li, Zihan Wang, and Yixuan Liu. 2022. Beyond the granularity: Multi-perspective dialogue collaborative selection for dialogue state tracking. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2320–2332.

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

Matthew Henderson, Blaise Thomson, and Steve Young. 2014. Word-based dialog state tracking with recurrent neural networks. In *Proceedings of the 15th annual meeting of the special interest group on discourse and dialogue (SIGDIAL)*, pages 292–299.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the*

2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 6769–6781.

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

Adarsh Kumar, Peter Ku, Anuj Goyal, Angeliki Metallinou, and Dilek Hakkani-Tur. 2020. Ma-dst: Multi-attention-based scalable dialog state tracking. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8107–8114.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Chia-Hsuan Lee, Hao Cheng, and Mari Ostendorf. 2021. Dialogue state tracking with a language model using schema-driven prompting. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4937–4949.

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*, pages 5478–5483.

Wenqiang Lei, Xisen Jin, Min-Yen Kan, Zhaochun Ren, Xiangnan He, and Dawei Yin. 2018. Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1437–1447.

Qingbiao Li, Weizhe Lin, Zhe Liu, and Amanda Prorok. 2021. Message-aware graph attention networks for large-scale multi-robot path planning. *IEEE Robotics and Automation Letters*, 6(3):5533–5540.

Sha Li, Mahdi Namazifar, Di Jin, Mohit Bansal, Heng Ji, Yang Liu, and Dilek Hakkani-Tur. 2022. Enhancing knowledge selection for grounded dialogues via document semantic graphs. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2810–2823.

Weizhe Lin, Bo-Hsiang Tseng, and Bill Byrne. 2021. Knowledge-aware graph-enhanced gpt-2 for dialogue state tracking. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7871–7881.

Ilya Loshchilov and Frank Hutter. 2018. Fixing weight decay regularization in adam.

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

Liliang Ren, Jianmo Ni, and Julian McAuley. 2019. Scalable and accurate dialogue state tracking via hierarchical sequence generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1876–1885.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable multi-domain state generator for task-oriented dialogue systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 808–819.

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. *ACL 2020*, page 109.

Yan Zeng and Jian-Yun Nie. 2020. Multi-domain dialogue state tracking based on state graph. *arXiv preprint arXiv:2010.11137*.

Haoning Zhang, Junwei Bao, Haipeng Sun, Youzheng Wu, Wenye Li, Shuguang Cui, and Xiaodong He. 2022. Monet: Tackle state momentum via noise-enhanced training for dialogue state tracking. *arXiv preprint arXiv:2211.05503*.

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

Yichi Zhang, Zhijian Ou, Min Hu, and Junlan Feng. 2020b. A probabilistic end-to-end task-oriented dialog model with latent belief states towards semi-supervised learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9207–9219.

Zheng Zhang, Ryuichi Takanobu, Qi Zhu, MinLie Huang, and XiaoYan Zhu. 2020c. Recent advances and challenges in task-oriented dialog systems. *Science China Technological Sciences*, 63(10):2011–2027.

Li Zhou and Kevin Small. 2019. Multi-domain dialogue state tracking as dynamic knowledge graph enhanced question answering. *In NeurIPS 2019 Workshop on Conversational AI.*

A Appendix

Type	Sub Type	Slot Name
Cate Slot	pricerange	hotel-pricerange, restaurant-pricerange
	area	attraction-area, hotel-area, restaurant-area
	number	hotel-bookpeople, hotel-bookstay, hotel-stars, restaurant-bookpeople, train-bookpeople
	day	hotel-bookday, train-day, restaurant-bookday
	boolean	hotel-internet, hotel-parking
	station	train-departure, train-destination
	type	hotel-type, attraction-type
Span Slot	name	attraction-name, hotel-name, restaurant-name, restaurant-food
	location	taxi-departure, taxi-destination
	time	restaurant-booktime, taxi-arriveby, taxi-leaveat, train-arriveby, train-leaveat

Table 6: Type of Slots.

Sub Type	Slot Name
pricerange	cheap, expensive, moderate
area	centre, east, north, south, west
number	0, 1, 2, 3, 4, 5, 6, 7,8, 9, 10, 15
day	monday, tuesday, wednesday, thursday, friday, saturday, sunday
boolean	yes, no
station	birmingham new street, bishops stortford, broxbourne, cambridge, ely, kings lynn, leicester, london kings cross, london liverpool street, norwich, peterborough, stansted airport, stevenage
hotel type	guesthouse, hotel
attraction type	architecture, boat, cinema, college, concerthall, entertainment, museum, multiple sports, nightclub, park, swimmingpool, theatre

Table 7: Possible Value Sets.

A.1 Classification of Slots

In Table 6, we present the classification of all tracked slots, which can be further categorized into subtypes based on their meanings. For example, slots like *bookpeople* and *stars*, which have numerical values, are classified as a subtypes "number".

Table 7 provides the possible value sets for each categorical slots. Although *train station* or *attraction type* may be more suitable for span slots, they were classified as categorical slots in this study. This decision was made because the MultiWOZ datasets only mention a limited set of values for these slots, and it is reasonable to predefine the possible value sets.

Graph Analysis	Train	Valid	Test
total # of example	7900	1000	999
total # of value graph	7300	914	916
total # of dialogue graph	7614	981	978
avg # of dialogue turn node	5.63	7.37	7.37
max # of dialogue turn node	20	17	18
avg # of updated slot node	6.02	6.31	6.18
max # of updated slot node	14	13	13

Table 8: Statistical analysis of graph in the training, validation, and test data of MultiWOZ 2.2.

A.2 Graph Analysis

In our analysis of the graph model in Table 8, we find that among all dialogue examples, the majority have dialogue graphs generated, while slightly fewer examples have value graphs generated. This suggests that most examples have both types of graphs, although there are instances where only one type is presented. On average, there are approximately 5 dialogue turn nodes in the training datasets and 6 updated slot nodes in the dialogue graphs. It is worth noting that the maximum number of nodes in the dialogue graph is limited to a maximum of 34.

A.3 Implementation Details

We utilize a pre-trained ALBERT-base-uncased model (Lan et al., 2019) with a hidden size 768 as our encoder. The AdamW optimizer (Loshchilov and Hutter, 2018) is employed with a warmup proportion of 0.01 and an L2 weight decay of 0.01. The peak learning rate for the state update predictor is set to the same value as Guo et al. (2021). The dual dynamic graph and state generator are trained jointly, with initial learning rates of 1e-3 and 2e-5 for the two major components. Word dropout is applied by randomly replacing input tokens with the special $[UNK]$ token (Bowman et al., 2016) with a probability of 0.1 (Srivastava et al., 2014). The maximum sequence length for all inputs is fixed at 512. During training, the ground truth updated slots are used instead of predicted ones for the dual dynamic graph and state generator. The training process consists of 5 epochs.

The graph attention networks are trained with 768 hidden dimensions, the same as the encoder. All GATs layers have output dimensions equal to the input dimensions. The number of layers is 4,

the number of heads per layer is 4, and the number of hops is 2.

Furthermore, MultiWOZ 2.1 has no annotated span labels for slots. To address this, we preprocess the MultiWOZ datasets by converting value labels to span labels. We identify the occurrence of a value label in the dialogue and use it as the span start and end labels.

We trained the entire model using a single RTX 3090 GPU. The average time required for training 1 epoch is approximately 2 to 3 hours, with variations possible depending on different development environments. The source code is available for reference on the official GitHub repository at https://github.com/HS2DG-DST.

A.4 Experiment Details

Due to disparate experimental conditions and the constraints of reproducing experiments, particularly when the reusability of the source code was limited. Consequently, it was unfeasible to reconduct all baseline experiments. Meanwhile, as noted in Table 1, the two most recent models (DSS-DST and DiCoS-DST) were re-experimented in the same experimental setting as ours. We found that the obtained outcomes closely align with the performance reported in those respective papers. Thus we've added the results from the baselines (TRADE, DSTQA, DS-DST, SOM-DST, TripPy, DST-as-Prompting) to the Table, as we believe they are comparable, albeit not reimplemented.