Journal Pre-proof

Advanced Dialog State Tracking with Noetic Graphs for Complex Human-Machine Interactions

Jingyang Li , Shengli Song , Sitong Yan , Guangneng Hu , Chengen Lai , Yulong Zhou

PII: S0031-3203(25)00502-3
DOI: https://doi.org/10.1016/j.patcog.2025.111842
Reference: PR 111842

To appear in: *Pattern Recognition*

Received date: 29 October 2024
Revised date: 29 April 2025
Accepted date: 10 May 2025

Please cite this article as: Jingyang Li , Shengli Song , Sitong Yan , Guangneng Hu , Chengen Lai , Yulong Zhou , Advanced Dialog State Tracking with Noetic Graphs for Complex Human-Machine Interactions, *Pattern Recognition* (2025), doi: https://doi.org/10.1016/j.patcog.2025.111842

This is a PDF file of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability, but it is not yet the definitive version of record. This version will undergo additional copyediting, typesetting and review before it is published in its final form, but we are providing this version to give early visibility of the article. Please note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2025 Published by Elsevier Ltd.

Journal Pre-proof

Highlights

- Introduces Noetic State Representation Graph for dialog state tracking.
- Enhances reasoning using DialoGPT for encoding utterance sequences.
- Utilizes graph attention for explicit dialog context representation.
- Achieves superior performance on SGD and MultiWOZ datasets.

Journal Pre-proof

Journal Pre-proof

Advanced Dialog State Tracking with Noetic Graphs for Complex Human-Machine Interactions

[1] Jingyang Li, [1*] Shengli Song, [1] Sitong Yan, [1] Guangneng Hu,
[1]Chengen Lai, [1]Yulong Zhou
[1] School of Computer Science and Technology, Xidian University,
Xi'an, China

*Corresponding author: Shengli Song;
Email : shenglisong.id@gmail.com; shlsong@xidian.edu.cn

Ethical Statements:

Funding: Not Applicable

Conflict of interest

The authors declare that they have no conflict of interest.

Human and Animal Rights

This article does not contain any studies with human or animal subjects performed by any of the authors.

Informed Consent

Informed consent was obtained from all individual participants included in the study.

Consent to participate: Not applicable

Consent for publication: Not applicable

Availability of data and material:

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Journal Pre-proof

Journal Pre-proof

Abstract

Dialog state tracking (DST) is a crucial process in task-oriented dialog systems, evaluating the current state of a conversation based on preceding interactions. Two effective techniques for DST are Large Language Models (LLMs) and neural networks. However, conventional neural networks lack explainability and reasoning abilities, limiting their adaptability to unknown domains and complex scenarios, posing challenges in real-time human-machine interfaces. This paper formulates the DST problem by assembling and representing the belief state using an explicit model and integrating it to enhance the neural network's DST capabilities. A novel technique, the Noetic State Representation (NSR) Graph, is proposed to address these challenges. The NSR graph offers a dynamic and explicit representation of dialog context that synchronizes with multi-turn dialogues. To improve reasoning ability and semantic augmentation, a pre-trained language model, DialoGPT, is employed as the encoder for utterance sequences. The core NSR graph is built and encoded using a graph attention network to ensure the explicit representation of dialog context. To generate the belief state, the proposed model utilizes a classical sequence decoder, which is guided by the context information from the NSR graph and utterances. Experimental results demonstrate the effectiveness of this approach, achieving a 0.8% improvement on unknown domains and a 1.7% improvement across all domains in the Schema-Guided Dialogue (SGD) dataset, outperforming advanced techniques and showing strong results on the MultiWOZ dataset.

Keywords: Task-oriented dialog system; Dialog state tracking; Noetic State Representation Graph;DialoGPT; Graph Attention Network; Multiple domains.

1. Introduction

Dialog Management (DM) is a crucial component of task-oriented dialog systems [1]. Since the 8th Dialog System Technology Challenge was held, dialog management research has expanded from single-domain to multiple and cross-

Journal Pre-proof

domain [2]. As the core module of the DM, Dialog State Tracking (DST) extracts user goals and relevant information at each turn based on the preceding dialog and additionally provides a corresponding dialog state for dialog Policy Learning (PL), which helps the agent determine the appropriate actions to take [3]. DST is especially important since its results directly determine the quality of the responses. Since the DSTC8 was held, an end-to-end neural network dialog system based on fine-tuning Generative Pre-trained Transformer 2 (GPT-2) [4] has achieved the best performance in human conversation evaluation, followed by many state-of-the-art end-to-end task-oriented dialog systems that outperform pipeline methods. Such improvements indicate the potential of end-to-end neural network models, which largely benefit from the effective incorporation of rich semantic information from pre-trained language models and the exploitation of fine-tuning modes on DST. For instance, Bidirectional Encoder Representations from Transformers- DST (BERT-DST) [5] applies an end-to-end framework as the backbone and introduces representation from the transformer (BERT) as a bidirectional encoder. DST via Entity Adaptive pre-training (DSTEA) [6] presents Natural Language Understanding (NLU) benchmarks of a task-oriented dialog system.

Moreover, external knowledge and schema information have been exploited to improve DST performance [7, 8]. External knowledge is incorporated for dialog understanding [9]. The Multi-view Graph Convolution and multi-agent Reinforcement Learning (MGCRL) method [10] presents a dialog state graph for reconstructing domain-slot pairs with a graph mode and enhances the DST performance by using a copy mechanism. However, these methods consider limited dialog turns instead of entire conversations. Previous research has shown that employing Pre-trained Language Models (PLMs) and external knowledge are feasible approaches for dialog state tracking tasks. However, in real-world engineering applications, rule-based dialog systems still be the mainstream method for the following reasons. BERT-based PLMs offer powerful semantic information, but their bidirectional structure and contextual dependencies are inconsistent with the one-directional nature of task-oriented dialogues, which limits their ability to capture temporal conversation flow. Additionally, previous models heavily rely on predefined ontologies and schemas, which blocks their capacity to augment dynamic

Journal Pre-proof

dialog state information and extend to unseen domains. Overall, neural network models still lack controllability, interpretability, and generalization, rendering them impractical. To solve these deficiencies, the proposed method focuses on the following three aspects: The Noetic State Representation (NSR) Graph building process is synchronized with each dialog turn, as shown in Figure 1, and it gradually evolves into a Noetic state representation graph that contains domain, slot, and value nodes, when the dialog turns iterate from 1st to 5th.

- ***Complex Scenario:*** Practical applications often involve complex scenarios. Cross-domain and multi-domain scenarios are relatively universal in real applications and indicate a new trend in task-driven dialog. Consequently, the foremost problem with the DST task in complex scenes is solving the complexity of the dynamic dialog state. To represent the state of the dialog context, the NSR Graph is presented with entity granularity and is built to synchronize with the multi-turn dialog proceeding.
- ***Reasoning Explainability:*** End-to-end neural networks fail to explicitly explain their decision-making process. Despite some other Natural Language Processing (NLP) tasks incorporating knowledge graphs to explain the reasoning path, they can realize only reverse causal interpretation. In this paper, the NSR Graph is exploited in both the encoding and decoding procedures to guide the dialog system for DST.
- ***Model Generalization:*** Previous neural network models rely heavily on predefined ontologies and thus have poor generalizability in few-shot or zero-shot settings. The proposed update mechanism of the NSR Graph could capture the new domain and slots in dialog flow.

Overall, the proposed model is an end-to-end method that incorporates the NSR Graph to represent dialogue context and enhance semantic information. Figure 1 illustrates how the model explicitly represents and integrates dialogue contexts and relevant semantic knowledge from schemas to improve the model's generalizability.

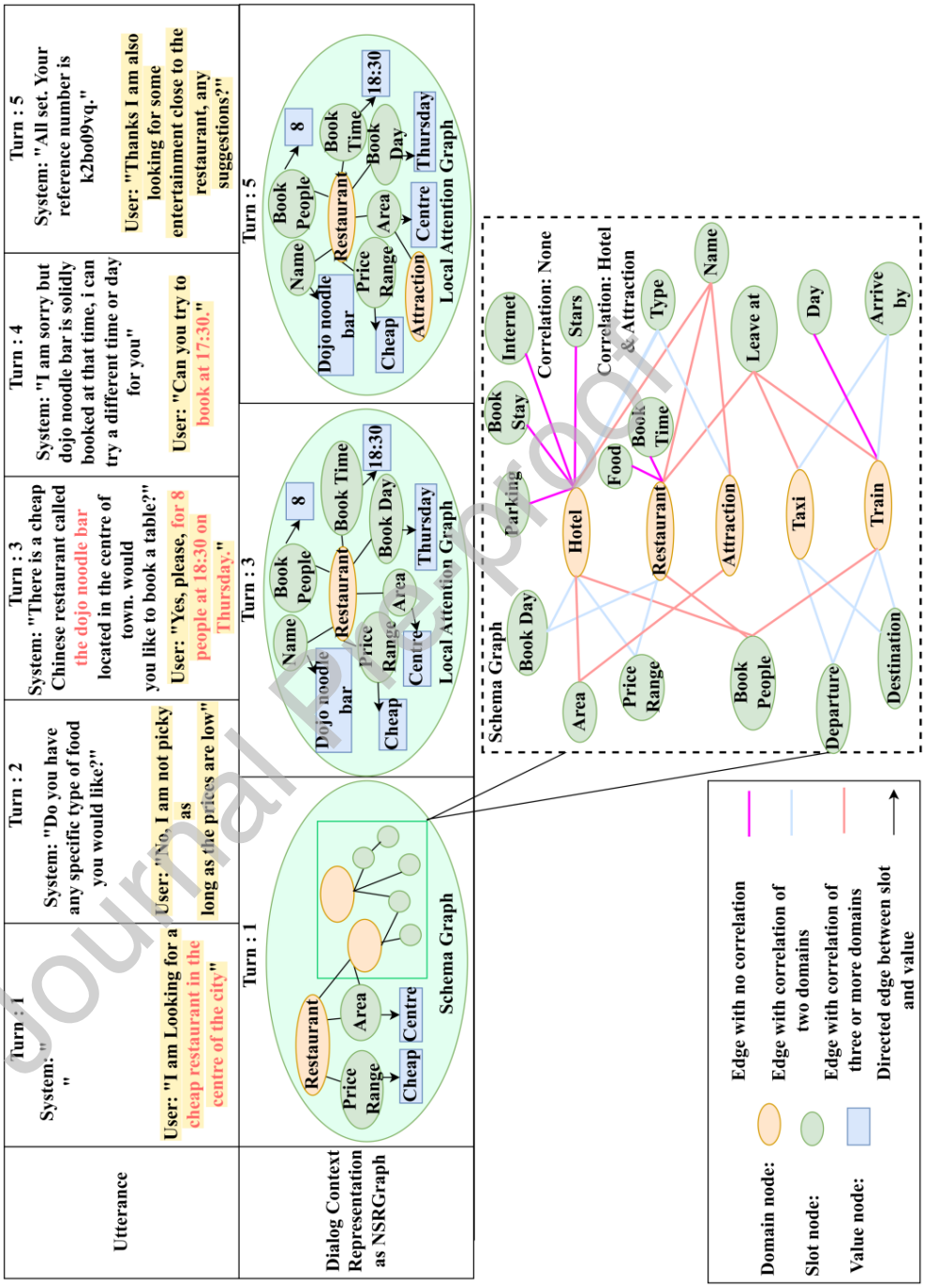

Figure 1: Construction of NSR Graph

Journal Pre-proof

Journal Pre-proof

1.1. Contribution

The main contributions of this paper are as follows:

- A novel dialog context representation structure, NSR Graph, is introduced, which differs significantly from traditional graph structures that only consider entity nodes and edges. NSR Graph effectively captures dynamic dialog states synchronized with dialog proceedings, enhancing DST.
- NSR Graph is integrated into an end-to-end neural network to solve DST tasks in a generative manner with an open-vocabulary setting. Specifically, DialoGPT, a pre-trained transformer-based language model, is incorporated, and a domain-slot-value sequence is designed for embedding generation. The method leverages NSR Graph to fuse domain-slot correlations and employs a graph attention network to guide DST sequence generation.
- The NSR Graph offers flexibility across multiple domains, making it robust for analyzing various datasets in multi-domain dialog tracking. Additionally, hyperparameters are optimized using the Adam optimizer, improving performance in multi-domain DST.
- Experimental results on benchmark datasets demonstrate strong performance in complex scenes and unseen domains. On the MultiWOZ dataset, the approach achieves competitive results compared to baselines, while on the Schema-Guided Dialogue (SGD) dataset, it excels in solving few-shot problems and adapting to unseen domain-slot pairs. The results confirm that incorporating an NSR Graph enhances explainability and generalization in neural network models for DST.

1.2. Organization

The rest of the paper is organized as follows: Section 2 contains a brief description of previous works related to multi-domain dialog state tracking. Section 3 presents the problem formulation of this work. Section 4 elaborates on the proposed methodology. Section 5 contains the experimental results of this work. Section 6 presents the ablation study. Section 7 contains the discussions and implications. This

Journal Pre-proof

paper concludes with Section 8.

2. Related works

For tracking the multi-domain dialog state, Khan et al. [11] employed Long Short Term Memory (LSTM). The dialog data obtained from various dialog domains were used. Without considering the availability of the in-domain data for the models' training, this procedure enhanced the performance of belief tracking. Tracking the correct beliefs using this procedure was complicated. Le et al. [12] established a dialogue state tracking multimodally. From the evaluations done, it was found that the proposed multimodal dialogue systems showed their superiority. The video domains were not explored in this work. For tracking the scalable and universal beliefs, Guo et al. [13] developed Enriching Sub-words Information Explicitly with BERT (ESIE-BERT) for slot filling and intent classification. The experimental result showed significant performance. The proposed model was not tested with the varying domain ontology. To forecast the slot value on multi-domain dialog state tracking, Jia et al. [14] developed a dual strategy. The experimental analysis attained superior performance. However, only a limited number of datasets were explored in this work. For the dialogue systems based on task, Zhao et al. [15] developed a Graph ATtention (GAT) network. The cross-domain slot problem was alleviated using the ontology schema subgraph and dialogue context subgraph. An experimental analysis was performed. However, the performance was poor. Khan et al. [16] developed a scalable DST based on multi-attention. For the clients to finish their tasks, a natural language processing interface was provided by the task-oriented dialogue agents. The encoding of dependencies by the model was an important parameter to perform accurate DST in multi-domain. Therefore, a new framework to encode the slot semantics and conversation history was developed in this paper. From the experiments done, it was noted that in the full dataset the proposed technique increased the Joint Goal Accuracy (JointGA). Efficient approaches were not used to collect the correlations and dependencies among the slots. Zhu et al. [17] developed Efficient Context and Domain Guidance based on DST (ECDG-DST) for smart dialogue systems. The efficient context was developed to minimize the amount of data and also utilized for high refinement of historical dialogue data to

Journal Pre-proof

protect the key details. A slot gate was developed to encourage the domain guide, as well as enhance the accuracy of value generation. You et al. [18] focused on a Turn-level Contrastive Learning Network (TCLNet) combined with a reranking module for (DST). These methods correct the in-between stages of the lengthy dialogues sequentially by differentiating data points at a finer approach. The capability of the model to handle long dialogue series was improved and attained superior performance. Jeon et al. [19] explained the Dialogue system by Optimizing a Recurrent Action policy (DORA) for multiple domain-related dialog systems, which utilizes Supervised Learning (SL) and successively, reinforcement learning was also applied to fine-tune the dialogue systems by utilizing recurrent dialogue policy and role of dialog history was efficiently performed. Lee et al. [20] developed DSTEA to learn the significant details of DST. Enhanced representation through knowledge Integration was utilized for pretraining and the selective knowledge masking method was developed to learn phase and word entities more commonly than other non-entities. This evaluation shows this entity extraction enhances the performance of DST.

2.1. Research gap

Utilizing NSR graphs for tracking the dialog state in multi-domain is an advanced topic that contains more advantages in dialog state tracking. The existing techniques contain a few limitations like complicated processes, lack of exploration of video domains, lack of exploration of various domain ontology, reduced datasets, and absence of advanced optimization techniques. To overcome these limitations an NSR Graph is proposed in this work. The research gaps are:

Time consumption: The time consumed to track the dialog from various domains is an important parameter that decides the cost of the operation and the energy consumption. Due to the lack of better datasets the existing techniques consumed more time for its operations.

Cost consumption: The cost of an operation mainly depends on the time consumed to execute the operation. The complexities in the operations of the existing works increase the time consumption and thereby increase the cost consumption.

Accuracy: Accuracy is a parameter that decides the efficiency of any technique.

Journal Pre-proof

The existing techniques do not contain effective advanced techniques which lead to a decrease in the accuracy.

3. Problem formulation

The dialog belief state at time t, denoted as B_t, is a set of slot-value pairs capturing the user's intent. It is formally defined as:

$$B_t = \left\{ \{ d - s_i : v_i \} | s_i \in S_d, v_i \in V_d, d \in D, i = 1,2,..,N \right\} \quad (1)$$

where d is domain name (eg., taxi, restaurant, flight, etc), D is set of all possible domains, S_d is the set of all possible slot names (eg., departure, destination, arrival time) in domain d, V_d is the set of all possible values corresponding to slots in domain d, N is the total number of slot-value pairs at time t .($s_i : v_i$) represents the i^{th} slot-value pair. For example, the user input is "I need a taxi from Cambridge to Gardenia at 12:15 PM" and the corresponding dialog belief state is given as:

$$B_t = \left\{ \begin{matrix} taxi - destination : Gardenia, taxi - departure : \\ Cambridge, taxi - arriveby : 12:15 \end{matrix} \right\} \quad (2)$$

The structured sequence representation Y_t is a linearized format of B_t, where slot-value pairs are concatenated using a special separator token [*SEP*].

$$Y_t = \left\{ \begin{matrix} taxi - destination : Gardenia, [SEP], taxi - departure : \\ Cambridge, [SEP], taxi - arriveby : 12:15 \end{matrix} \right\} \quad (3)$$

To generate the belief state B_t, the probability of the sequence Y_t is modeled as a product of conditional probabilities, considering previous belief states B_{t-1} and additional contextual information Y_t. This is expressed as:

Journal Pre-proof

$$P(Y_t) = \prod_{i=0}^{n} y_i | B_{t-1}, E_t \tag{4}$$

where $y_i \in Vocab(Y)$, n represents the predefined maximum sequence length or the actual sequence length determined by the [*EOS*] token and $Vocab(Y)$ typically refers to the vocabulary of the target variable Y , which includes all unique words (or tokens) present in the dataset. The generated sequence Y_t is then transformed back into B_t using predefined transformation rules. In Equation (4), the input consists of the previous belief states B_{t-1} and additional contextual information E_t . The belief states B_{t-1} capture the essential information about the dialogue's history, including the system's understanding of the conversation up until the previous time step. This historical belief state already encodes the relevant content of the utterances indirectly, as it reflects both the user inputs and the system's responses. The additional contextual information E_t , which includes things like the system's state or other features, provides supplementary details necessary for generating the next belief state. Instead of directly using the raw utterances as input, the model leverages the belief states and contextual information to update the dialogue state, as the belief states already implicitly carry the necessary information derived from the utterances. Thus, this approach reduces redundancy by not explicitly reintroducing the utterances themselves but still ensuring that all relevant information is captured through the belief states and context.

4. Proposed Methodology

In this section, a detailed explanation of the technique proposed in this work is presented. It includes the architecture of the proposed methodology and the various components in it.

Journal Pre-proof

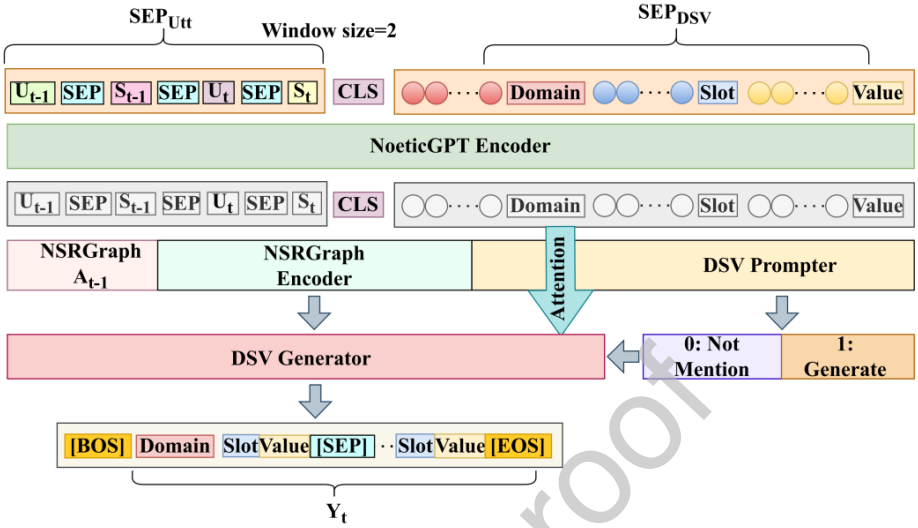

Figure 2: Overall framework of the proposed NSR Graph.

4.1. Architecture overview

The overall framework designed and depicted in Figure 2, follows a structured pipeline to track and update the belief state in a dialogue system, ensuring seamless interaction between its components. It consists of four key modules such as NoeticGPT encoder, DSV prompter, NSR graph encoder, and DSV generator. The framework processes three main inputs at each time step t: history dialogue utterance (user utterance and system response) within a window size $w = 2$, Domain-Slot-Value (DSV) string representing the structures belief state, and NSR graph adjacency matrix A_{t-1} from the previous time step. The process begins with the NoeticGPT Encoder, which takes as input the user utterance, system response, and a formatted Domain-Slot-Value (DSV) string. The DSV string represents the structured belief state, where each DSV node corresponds to a specific domain (e.g., restaurant, taxi), slot (e.g., location, time), and value (e.g., Cambridge, 12:15 PM) in the dialogue. The DSV string's role is to encode these components into a shared semantic space, ensuring contextual dependencies are preserved while generating fixed-size sequence representations. The output is a set of contextualized embeddings that represent both the utterance and DSV nodes in a unified space.

Journal Pre-proof

Journal Pre-proof

These embeddings serve as input to the NSR Graph Encoder, which constructs an explicit belief state representation using an adjacency matrix that defines relationships between domains, slots, and values. By leveraging a GAT, it dynamically encodes DSV sequences, facilitating context exchange and aggregation across dialogue turns. The output is a graph-structured belief state embedding enriched with contextual relationships. This embedding is then passed to the DSV Prompter, a binary classifier that determines if the current dialogue turn contains slot-value pairs that need extraction. If no relevant slot-value pairs are detected, the classifier outputs 0, halting further processing. If slot-value pairs are present, it outputs 1, activating the DSV Generator. The DSV Generator takes as input the NSR Graph embeddings and utterance embeddings, fusing them using an attention mechanism to generate the updated belief state. The generator iteratively predicts tokens step-by-step by integrating extracted embeddings from both the utterance sequence and NSR Graph hidden states. The next token is generated upon concatenation through token embedding, continuing until the End-of-Sequence (EOS) token is reached or the maximum sequence length is met. The output is a structured DSV sequence representing the updated belief state for the current dialogue turn. This updated belief state is then fed back into the system to guide response generation, action prediction, and overall dialogue management. By integrating PLMs, GNNs, and attention mechanisms, the framework effectively captures local and global contextual dependencies, ensuring precise and robust dialogue state tracking. The seamless information flow and interdependence between modules ensure that belief state tracking remains dynamic, contextually relevant, and continuously refined across turns, thereby enhancing the robustness of the dialogue system.

4.2. NoeticGPT Encoder

NoeticGPT Encoder is based on the DialoGPT [21], a generative pre-trained dialogue transformer designed specifically for dialog response generation tasks. In each dialog turn, the system and user utterance SEP_{Utt} are formalized within a sliding window. Here, U_t represents the user utterance at turn t, while S_t

represents the system response at turn t. The separator character is denoted by [SEP]. To balance the model performance and complexity, the window size n is set to 2 and the maximum token length to 128 in practice. The utterance formulation is as follows.

$$SEP_{Utt} = \{U_{t-n+1}, [SEP], S_{t-n+1}, [SEP], \ldots, U_t, [SEP], S_t\}, window\ size = n \quad (5)$$

The reason for separating the user utterance and system response is to enable the classification model to perform the classification task based on each separate part of the system and user utterances. Moreover, the model needs to determine whether the current utterance requires span information extraction. In addition, the NoeticGPT Encoder provides the initial node embeddings for the NSR Graph. A formatted string SEP_{DSV} is presented, comprising domain, slot, and value predefined in the ontology. To be more adaptable to open-vocabulary settings, the domains and slots are represented with independent tokens without specific descriptions, despite the predefined ontology incorporated. Here, abstract placeholders are designed to describe non-categorical slots. Specifically, numerical slots are represented by $number_{value}$, name slots are represented by $name_{value}$, time slots are represented by $time_{value}$, and place slots are denoted by $place_{value}$.

$$SEP_{DSV} = \begin{Bmatrix} domain: hotel, taxi, \ldots, attraction, slot: type, \\ price, .., time, value: cheap, Cambridge, \ldots, \\ number_{value}, name_{value}, time_{value}, place_{value} \end{Bmatrix} \quad (6)$$

Taking the NSR Graph on the MultiWOZ dataset as an example, there is a total of 69 nodes involved, which corresponds to a length of 70 for SEP_{DSV} . Compound words in 63 nodes are replaced by their first words or synonyms. Then, descriptors are added for the domain, slot, value, and four types of placeholders. The length of SEP_{DSV} is fixed to enable indexing of the 63 nodes in the NSR Graph based on their position after encoding.

$$H_t = NoeticGPT(\{SEP_{Utt}, [CLS], SEP_{DSV}\}) \quad (7)$$

Journal Pre-proof

Journal Pre-proof

NoeticGPT Encoder concurrently encodes both SEP_{Utt} and SEP_{DSV} for subsequent modules incorporating the causal information between them. The special token [CLS] is used to separate SEP_{Utt} and SEP_{DSV} in the input. Then the NoeticGPT Encoder produces the hidden states H_t , and the hidden states at [CLS] position are then used as input to the DSV Prompter. The hidden states at SEP_{Utt} position are sent into the DSV Generator, and the hidden states at SEP_{DSV} position are used as input to the NSR Graph encoder.

4.3. DSV Prompter

DSV Prompter aims to prompt the decoder to learn DSV generation correctly, which also prevents unnecessary tracking of DSV generation during the inference process. This module consists of two-layer linear networks with ReLU activation and a softmax layer. It takes the hidden state H_t on the index of [CLS] as input.

$$P\left(\mathrm{Pr}ompter_t \left| H_t^{CLS} \right.\right) = softmax\left(\mathrm{Re}lu\left(w * H_t^{CLS} + b\right)\right) \tag{8}$$

where $Prompter_t$ is the output at turn t , w and b represent learnable parameters, and *Relu* is the activation function that introduces non-linearity to the network. The output of the softmax function indicates whether the current dialogue turn requires tracking and generation of slot-value pairs. If the DSV Prompter outputs 0, the decoder skips generating any slot-value pairs for this turn, effectively preventing unnecessary tracking. During training, if the output is 0, the decoder generates an empty string, signaling that no slot-value pair needs to be generated. However, if the DSV Prompter outputs 1, the DSV Generator will be activated in the inference stage to generate the corresponding slot-value pairs as required.

4.4 NSR Graph

An example of the step-by-step building process of the NSR Graph is illustrated in Figure 3. The NSR Graph represents the evolving dialog context and serves as a structured guide for generating the DSV sequence. It consists of nodes and edges, where nodes represent domains, slots, and values, while edges capture their

relationships. The primary goal of the NSR Graph is to model dependencies between these components as the dialog progresses. Different datasets have varying structures: for instance, the MultiWOZ dataset consists of 37 slots in 7 domains, while the SGD dataset contains 215 slots across 16 domains. Certain domain-slot relationships are naturally stronger than others. For example, in a travel-related conversation, the slot "stay" is frequently associated with the domain "hotel", while "stars" is less relevant to "hotel" and "destination" is unrelated. These structured correlations play a crucial role in accurate DSV prediction. To effectively capture both static schema knowledge and dynamic dialog context, the NSR Graph is constructed with a static schema graph and a dynamic dialog context graph. A Static Schema Graph is derived from predefined dataset ontologies, capturing fundamental domain-slot-value relationships at initialization. A Dynamic Dialog Context Graph evolves at each dialog turn, updating dependencies between domains, slots, and values based on the conversation flow.

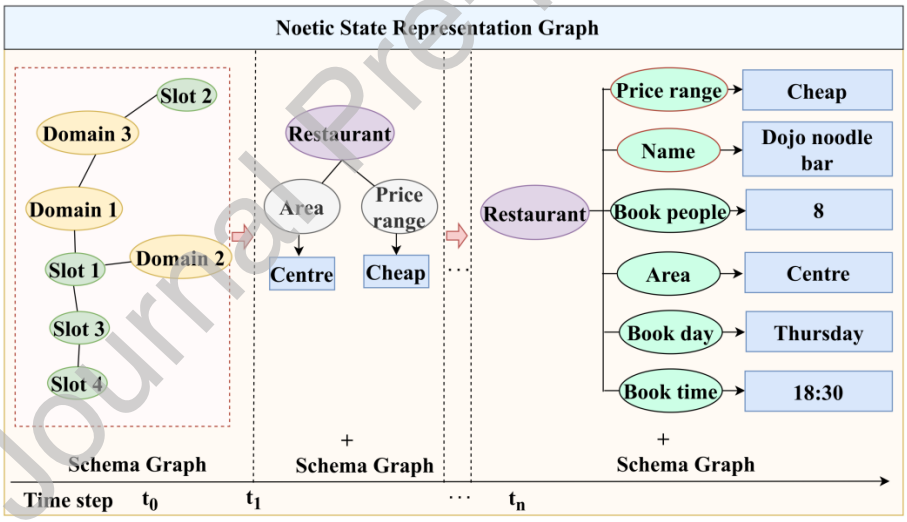

Figure 3: Building process of NSR Graph

Graph Representation and Evolution

At any given dialog turn t, the NSR Graph is represented as a directed graph $G_t = (V_t, E_t)$, where V_t represents the set of nodes (domains, slots, and values) at time t and E_t represents the set of edges (relationships between nodes). The

structure of G_t is dynamically updated as new slot-value pairs appear in the conversation. At the initial time step ($t=0$), the graph is initialized using the schema structure of the dataset. As the conversation progresses from t to $t+1$, new slot-value pairs are incorporated into the graph, and the adjacency matrix A_{t-1} (from the previous time step) is updated to A_t.

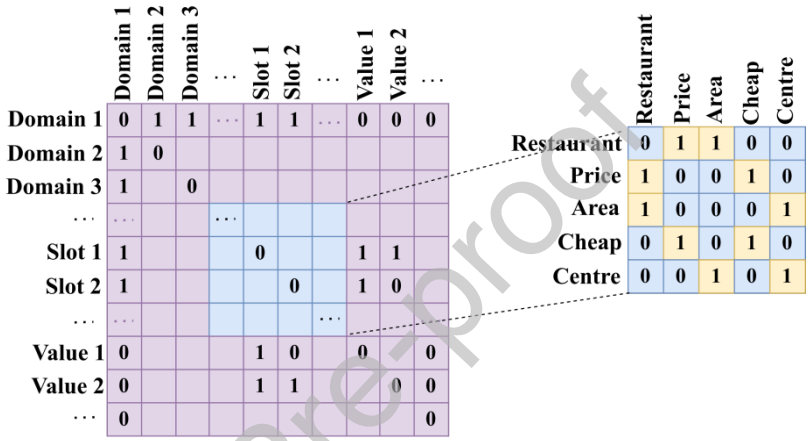

Figure 4: Local attention of adjacency matrix representation

Adjacency matrix update mechanism

The adjacency matrix A_t encodes the relationships between nodes in the NSR Graph. The matrix evolves as follows: If a new slot-value pair is observed at time t, the corresponding node and edge are added to the graph, updating A_t. If the slot already has an existing value connection, the previous edge weight is reduced to 0, and a new connection is established. Mathematically, if a new slot-value pair (s,v) appears in the conversation at turn t, the adjacency matrix is updated as:

$$A_t(s,v)=\begin{cases}1, & if\ (s,v)\ is\ a\ new\ slot-value\ pair\ at\ time\ t\\ 0, & if\ s\ had\ a\ previous\ value\ connection\\ A_{t-1}(s,v)+1, & if\ the\ edge\ already\ existed\end{cases} \quad (9)$$

This incremental update mechanism ensures that the NSR Graph maintains both local dialog dependencies (recent slot-value pairs) and global contextual information

Journal Pre-proof

Journal Pre-proof

(previous interactions). Figure 4 demonstrates local attention applied to the adjacency matrix representation, allowing the model to focus on meaningful dialog dependencies.

Graph Embedding and Attention Mechanism

At each dialog turn t, the dialog state is encoded as a vector representation:

$$h_t = f(A_t, h_{t-1}, w, b) \tag{10}$$

where h_t represents the dialog state vector at time t, A_t represents updated adjacency matrix, h_{t-1} is the previous dialog state, $f(\cdot)$ represents a transformation function that integrates structural dependencies and, w and b are the learnable weight and bias parameters. The dialog state vector is then transformed into a raw graph embedding using a GAT to refine structural information:

$$h'_t = GAT(h_t, A_t) \tag{11}$$

where h'_t is the refined embedding incorporating both local and global graph attention. GAT enables the model to focus on relevant domain-slot-value dependencies while suppressing irrelevant connections. The final NSR graph embedding at the time step t is computed as:

$$H_t = \sigma(w_h h'_t + b_h) \tag{12}$$

where w_h and b_h are trainable parameters and σ is an activation function. This embedding is subsequently passed to the DSV generator for sequence generation. Overall the NSR Graph provides a structured and evolving representation of dialog context by integrating static schema knowledge with dynamic dialog dependencies. Through adjacency matrix updates and graph attention mechanisms, the model effectively captures both short-term slot-value dependencies and long-range contextual information. This structured approach enhances DSV prediction accuracy, ensuring coherent and contextually relevant responses.

4.5. DSVGenerator

Journal Pre-proof

Journal Pre-proof

The DSV Generator module is responsible for generating the DSV sequence Y_t based on the outputs from previous modules. Instead of using a pre-trained generation model such as T5, the classical LSTM [22] is chosen as the final decoder option for two main reasons. First, LSTM reduces the dictionary space of the target sequence and requires fewer dictionary-mapping layer parameters. Second, LSTM is better suited for combining the utterance sequence and NSR Graph embeddings and requires fewer parameters to achieve comparable performance to transformer decoders. This is because target spans are generally shorter in length, with a maximum length of 23 and an average length of 5.14. By utilizing LSTM, the proposed model requires fewer parameters and enables efficient training, even with a single GPU.

Token-Level DSV Generation

During the decoding process, the DSV sequence is generated at the token level. The input at the generation time step t, denoted as h_t^{DSV}, is a concatenated vector that integrates: the contextual information from NSR graph embeddings (h_t^{NSR}), the utterance encoding from previous turns (h_t^{Utt}), and the last generated token (y_{t-1}). Thus, the token representation at the time step t is given by:

$$h_t^{DSV} = f\left(h_t^{NSR}, h_t^{Utt}, y_{t-1}\right) \tag{13}$$

The next token is predicted using a softmax function over the vocabulary:

$$y_t = softmax\left(wh_t^{DSV} + b\right) \tag{14}$$

where w is the learnable weight matrix, and b is the bias parameter. This token-level DSV Generator efficiently extracts slot-value pairs from the input utterance, leveraging a span-generation approach for direct slot disambiguation. An example of slot disambiguation through span generation is presented below. Consider the utterance: "I need to book a hotel in the east that has 4 stars."Here, the phrase "in the" indirectly suggests the area slot without explicitly stating it. The proposed end-to-end sequence generation model identifies this correspondence without requiring additional slot alignment mechanisms. This approach also enables generalization to unseen slot values during inference.

Guiding the Decoding Process with NSR Graph

A critical component of the decoding process is the initial cell state of the LSTM, which is embedded with the NSR Graph output:

$$c_0^{LSTM} = g\left(h^{NSR}\right) \tag{15}$$

Where c_0^{LSTM} is the initial LSTM cell state and $g\left(h^{NSR}\right)$ represents function mapping NSR graph embeddings to the initial LSTM state. This allows the NSR Graph representation to influence span generation, ensuring contextual alignment between slot-value relationships in the dataset and the generated output. The target sequence in the proposed approach is structured as:

$$Y = \left(d, s_1, v_1, s_2, v_2, ..., s_n, v_n\right) \tag{16}$$

where $(s_1, v_1), (s_2, v_2), \ldots, (s_n, v_n)$ represents slot-value pairs, and d is the domain token. This format ensures smooth decoding while constraining sequence length, as predefined slot values do not require explicit delimiters. For example, the utterance: "I am looking for a cheap restaurant in the center of the city." corresponds to the target sequence:

$$Y = \left("restaurant", "price", "cheap", "area", "center"\right) \tag{17}$$

To enhance decoding stability, special start and end tokens, [*BOS*] and [*EOS*], respectively, are employed during training:

$$Y = \left(\left[BOS\right], d, s_1, v_1, \ldots, s_n, v_n, \left[EOS\right]\right) \tag{18}$$

Decoding Strategies

The probability transition matrix, P, is computed over the target vocabulary space, V, guiding token generation as:

$$P\left(y_t \middle| y < t\right) = softmax\left(wh_t^{DSV} + b\right) \tag{19}$$

where h_t^{DSV} is updated at each step based on the previously generated tokens.

Journal Pre-proof

Training and Inference Strategy

During training, the teacher forcing mechanism is applied with a ratio of 0.2, where the model is guided by ground truth tokens 80% of the time. For inference, sequence decoding algorithms such as greedy search and beam search are used. Greedy Search selects the token with the highest probability at each step and the Beam Search maintains multiple candidate sequences, choosing the most probable global sequence. The Beam Search outperforms Greedy Search in DSV sequence generation:

$$Y_t = \underset{Y}{argmax} \prod_{t=1}^{T} P\left(y_t | y_{1:t-1}\right) \tag{20}$$

where Y_t is the final optimal predicted sequence, chosen by maximizing the probability over all candidate sequences Y. $\underset{Y}{argmax}$ selects the sequence Y that maximizes the probability, $\prod_{t=1}^{T}$ represents the product over all time steps t, from 1 to T. $P\left(y_t | y_{1:t-1}\right)$ represents the probability of generating a token y_t given all previous tokens $y_{1:t-1}$, y_t is the predicted token at time step t, and $y_{1:t-1}$ represents the sequence of all tokens generated before time step t. This formulation is used in Beam Search, which maintains multiple candidate sequences and selects the one with the highest probability, ensuring optimal generation of DSV sequences.

4.6. Objective functions

The training objectives consist of two parts: one for the DSV Prompter module and another for the DSV Generator module. Both objective functions are based on the cross-entropy function. During the training process, the aforementioned modules are jointly trained and optimized using the summation of the respective losses. The loss function is defined as follows.

$$Loss = loss_{Generator} + \lambda * Loss_{prompter*} \tag{21}$$

In particular, when λ = 0.01, both modules achieve the best performance and avoid

Journal Pre-proof

overfitting. An intuitive reason for this is that the sequence generation task has a higher optimization complexity than the binary classification task, making it more sensitive to overfitting.

5. Experimental results

The experiments conducted in this work and their results are discussed in detail in this section and it also includes the graphical representations.

5.1. Datasets

This paper focuses on a complex cross-domain DST task and conducts experiments on three large-scale multi-domain goal-oriented datasets: SGD [20, 23], MultiWOZ 2.1 [24, 25], and MultiWOZ 2.2 [24, 26]. MultiWOZ 2.1, a widely used benchmark for dialog state tracking tasks, comprises over 10,000 dialogs, spanning seven distinct domains, and 30 corresponding domain-slot pairs. MultiWOZ 2.2 improves upon MultiWOZ 2.1 by fixing dialog state annotation errors across 17.3% of the utterances and enhancing the ontology definition and slot annotations. SGD is the most challenging DST testbed, comprising over 16,000 multi-domain conversations spanning 16 domains. Moreover, the dataset contains unseen domains and services in the evaluation set to assess performance in zero-shot or few-shot settings. Table 1 summarizes the statistics of the datasets used in the proposed experiments.

Table 1. Statistics of datasets used. The numbers indicate the number of data points used in the training datasets.

Characteristics	MultiWOZ2. 1	MultiWOZ2. 2	SGD
No. of domains	7	8	16
No. of dialogs	8, 438	8, 438	16, 142
Total no. of turns	113, 556	113, 556	329, 964
Avg. turns per dialog	13. 46	13. 46	20. 44
Avg. tokens per turn	13. 38	13. 13	9. 75
No. of slots	37	61	215

Journal Pre-proof

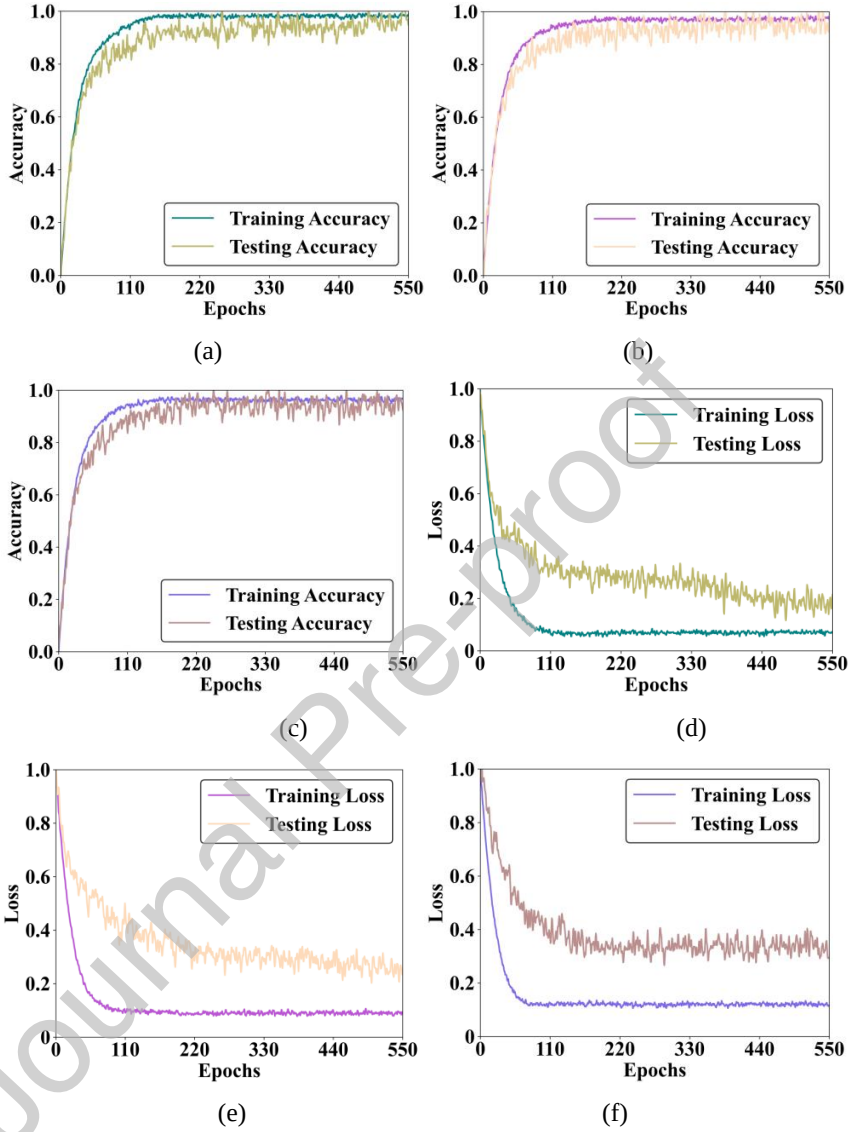

Figure 5: Comprehensive accuracy and loss graph. (a) Accuracy of MultiWOZ 2.1 (b) Accuracy of MultiWOZ 2.2 (c) Accuracy of SGD (d) Loss of MultiWOZ 2.1 (e) Loss of MultiWOZ 2.2 (f) Loss of SGD

5.2 Performance analysis

The training and testing accuracy and loss values are presented in Figure 5 (a-f).

Journal Pre-proof

Journal Pre-proof

Training accuracy is generally very high for all datasets, with MultiWOZ 2.1 achieving the highest training accuracy at 0.982, followed by MultiWOZ 2.2 achieving an accuracy of 0.971, and SGD achieving an accuracy of 0.963. Similarly, testing accuracy is highest for MultiWOZ 2.1 (0.968) but slightly lower for MultiWOZ 2.2 (0.951) and lowest for SGD (0.943). The low training losses indicate that the model fits the training data well. The training loss of three datasets such as MultiWOZ 2.1, MultiWOZ 2.2, and SGD are 0.07, 0.09, and 0.12 while the testing loss is 0.18, 0.25, and 0.34, respectively.

5.3. Training settings

The data preprocessing procedure follows a method similar to that used in [27] for both MultiWOZ 2.1 and 2.2 datasets. Since existing DST datasets do not involve span labels, the belief state is converted to available span labels at the beginning of data preprocessing. The proposed NSR Graph model is implemented using the AllenNLP [28] framework, which enables efficient development with its flexible code structure. With the benefit of the window input setting, the maximum length of the input sequence only needs to be 128 when the window size is two. The DialoGPT [29] model with 768 hidden units is used as the encoder, incorporating a multilayer GAT [30] network for graph embedding and a 3-layer LSTM sequence as the decoder. The best-performing model in the proposed experiments has 145 million parameters and a dropout of 0.2 (including the DialoGPT). The models are trained using the Adam optimizer with a learning rate of $1e^{-5}$ and 300 warmup steps. The batch size is set to 32 with four steps of gradient accumulation, and the gradient clip is set to 10. The proposed model aims to generate span and classification labels, with a 0.01 weight for classification labels. After training for two days on a Tesla A100 GPU for 115 epochs, NSR Graph achieves the best performance. The number of layers for the GAT network is set to 2 with 16 attention heads for optimal graph embedding. The output sequence length is constrained to a maximum of 23 tokens, with an average length of 5.14. During training, a teacher forcing ratio of 0.2 is used, and a beam search width of 3 is employed for improved decoding. Additionally, the dropout rate is set to 0.1, and a batch size of 32 is adopted during training.

Journal Pre-proof

Journal Pre-proof

5.4. Main results

The main evaluation metric for the DST tasks was JointGA, defined as the ratio of dialog turns in a dataset for which all slots have been filled correctly according to the ground truth. It is a widely used metric for evaluating task-oriented DST models. Table 2 presents the performance of the proposed approach compared to the baselines on various datasets.

Table 2. JointGA of the NSR Graph and baselines on the MultiWOZ2.1, MultiWOZ2.2, and SGD datasets.

Models	**MultiWOZ2.1**	**MultiWOZ2.2**	**SGD Unseen Domains**	**SGD All Domains**
LSTM	46.0%	45.4%	-	-
ESIE-BERT	45.7%	45.2%	-	-
ECDG-DST	52.1%	51.4%	23.5%	30.1%
TCLNet	49.0%	47.9%	-	-
DORA	43.4%	42.0%	20.0%	25.4%
DSTEA	50.8%	48.8%	-	-
OPAL	48.5%	49.3%	-	-
MSPN	47.8%	48.2%	-	-
SeKnow-S2S & SeKnowPLM	51.6%	50.9%	23.7%	29.8%
Proposed	**53.5%**	**52.9%**	**24.3%**	**31.8%**

The performance is measured using JointGA, which assesses the percentage of dialogue turns where the predicted state matches the ground truth. Some of the existing end-to-end models for DST include the Ontology-Aware Pretrained Language Model (OPAL) [31], the Multi-Span Prediction Network (MSPN) [32], and the Semi-Structured Knowledge Management-Based Sequence-to-Sequence (SeKnow-S2S) model and SeKnowPLM [33]. The existing end-to-end DST models have several advantages and limitations. OPAL benefits from an ontology-aware approach, ensuring robustness in structured environments but limiting adaptability to unseen domains. MSPN improves span extraction with multi-span prediction yet

Journal Pre-proof

struggles with complex multi-turn dialogues. SeKnow-S2S & SeKnowPLM enhance knowledge integration, leading to better generalization in unseen domains but still fall short of optimal accuracy. Traditional models like LSTM [11] and ESIE-BERT [13] show significantly lower performance, highlighting the need for more advanced architectures. TCLNet [18] is effective in structured dialogues but lacks adaptability to unseen scenarios.

The proposed model achieves the highest accuracy across all datasets, with 53.5% on MultiWOZ 2.1, 52.9% on MultiWOZ 2.2, 24.3% on SGD Unseen Domains, and 31.8% on SGD All Domains, demonstrating its superiority over existing approaches. Among state-of-the-art models, ECDG-DST performs competitively with 52.1% and 51.4% on MultiWOZ datasets but lags behind in unseen domains (23.5%). SeKnow-S2S & SeKnowPLM show strong performance on SGD Unseen Domains (23.7%) and SGD All Domains (29.8%) but still fall short of the proposed model. End-to-end models such as OPAL (49.3%), and MSPN (48.2%) achieve competitive results but do not surpass the proposed model. OPAL leverages an ontology-aware approach, making it robust in structured environments but less adaptable to unseen domains. MSPN relies on multi-span prediction, which improves span extraction but struggles with complex multi-turn dialogues. Models like DORA [19] and DSTEA [20] provide moderate improvements but still fail to match state-of-the-art end-to-end approaches. TCLNet achieves 49.0% on MultiWOZ 2.1 and 47.9% on MultiWOZ 2.2, demonstrating its effectiveness in structured dialogues but lacking adaptability to unseen domains. The proposed model overcomes these limitations by integrating the NSR Graph for better schema-based reasoning, enhancing robustness in complex dialogues, and improving generalization to unseen domains. It effectively incorporates schema and background knowledge while maintaining high accuracy across structured and unseen datasets. However, its reliance on structured schema knowledge limits its adaptability to open-domain conversations, and its performance in zero-shot and few-shot DST remains uncertain. Future work should focus on developing schema-free DST mechanisms, optimizing graph-based reasoning for efficiency, integrating few-shot learning strategies, and incorporating human feedback mechanisms to enhance adaptability and generalization in practical applications.

Journal Pre-proof

The metrics presented in Table 3 assess the performance and efficiency of various models in DST. Parameters (M) represent the total number of learnable weights in the model, indicating its complexity. FLOPs (B) measure the number of floating-point operations, reflecting the computational cost during inference and training. Training Time (Hours) tracks the duration required to train the model, highlighting the time efficiency of the training process. Computational Time (Seconds) measures the time taken for the model to process input and make predictions, which is crucial for real-time performance in practical applications. Together, these metrics provide a comprehensive evaluation of the models' complexity, computational demands, training efficiency, and real-time performance.

Table 3. Performance analysis of complexity

Techniques	Parameters (M)	FLOPs (B)	Training time (Hours)	Computational time (seconds)
Proposed	**14**	**125.76**	**20**	**0.12**
LSTM	23	136.22	29	3.45
ESIE-BERT	17	133.45	26	5.32
ECDG-DST	21	138.26	31	1.23
TCLNet	15	129.35	24	4.67
DORA	19	127.42	22	2.43
DSTEA	18	132.57	24	2.65
OPAL	16.6	129.42	25	1.28
MSPN	17.4	130.49	27	1.31
SeKnow-S2S & SeKnowPLM	14.5	126.48	22	1.03

Journal Pre-proof

6. Ablation study

There are four basic modules in the proposed model. The ablation studies for the NSR Graph, the DSV prompter, and NoeticGPT are implemented individually to learn the effects of these modules related to the DSV generation. The impact of utilizing the DialoGPT encoder with causal characteristics, the NSR Graph with comprehensive semantic knowledge, and the DSV prompter with prompt data for

enhancing DST performance is analyzed. The variation of span accuracy under multiple settings during the training process is shown in Figure 6.

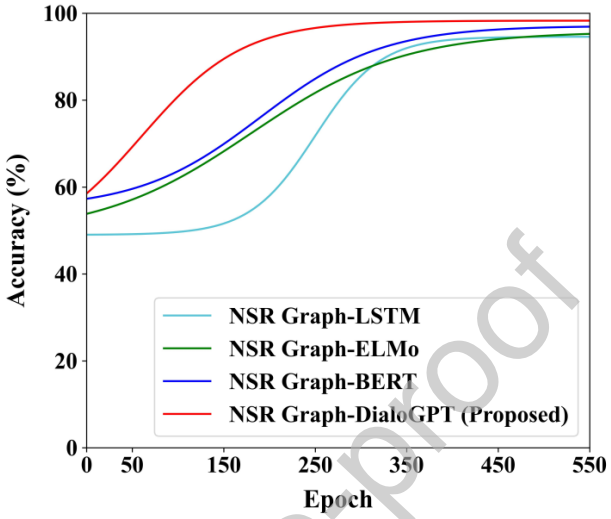

Figure 6: Span accuracy variation under different settings on the test dataset during the training process

The variations in the span accuracy for various settings like NSR Graph-LSTM, NSR Graph-ELMo, NSR Graph-BERT, and NSR Graph-DialoGPT (proposed) with concerning the test dataset at the time of training are shown in the above graph. The four curves in the graph indicate the four settings respectively as mentioned in the graph. From the accuracy curves in the graph, the curve of NSR Graph-DialoGPT reaches the highest accuracy value of 98.3% compared to the curves of other techniques.

6.1. The effect of NoeticGPT Encoder

A comparison is conducted between the NoeticGPT encoder and other encoders, such as LSTM, ELMo word embedding, and BERT. Based on the results presented in Figure 6, BERT performed similarly to DialoGPT in terms of span generation during the training process. However, as the epoch increased, BERT underperforms DialoGPT, which implies that DialoGPT exhibits stronger learning capability as the training becomes more detailed. The experiment results shown in Table 4 reveal that

Journal Pre-proof

Journal Pre-proof

the random encoding technique, when combined with LSTM, fails to learn sufficient semantic information from the training data. Although ELMo word embedding improves embedding performance, its unsophisticated approach to obtaining word embeddings limits its ability to capture complex semantic relations, resulting in only a modest performance gain. BERT, as a pre-trained model, excels in learning-rich semantic knowledge due to its extensive pretraining, leading to improved overall performance across the board. However, the NoeticGPT encoder, incorporating a pretraining procedure based on the inference DialoGPT model, enhances the causal reasoning capabilities, resulting in the best overall performance among the encoders. These results suggest that the ability to reason causally during encoding is crucial in the DST model.

Table 4. Ablation studies for encoders on MultiWOZ2.1 with the JointGA and accuracy of each turn.

Models	JointGA	Turn Accuracy
NSR Graph-LSTM	45.9%	96.4%
NSR Graph-ELMo	48.5%	96.9%
NSR Graph-BERT	51.4%	97.8%
NSR Graph-DialoGPT	**53.5%**	**98.3%**

6.2. The effect of NSR Graph

NSR Graph facilitates the generation of belief states by providing contextual representations during decoding. The ablation study is presented in Table 5 to evaluate the effectiveness of the NSR Graph using four different generation modes: no- NSR Graph, simple-NSR Graph, moderate-NSR Graph, and complex-NSR Graph.

NSR Graph-L2H3 outperforms the other three NSR Graph modes, indicating its effective intervention and instruction in predicting DST results, particularly in unseen domains. Intuitively, this is because the NSR Graph enriches the implicit dependencies and background knowledge of various slots. However, Table 5 shows that increasing the number of layers and heads does not improve results, despite the increasing network complexity. The experimental results suggest that L2H3

Journal Pre-proof

achieves the best performance by capturing second-order neighbors within three types of relations, which provides an appropriate representation of graphs for embedding slot-value relationships in datasets. Furthermore, it is observed that a complicated network can lead to overfitting and reduced generalization abilities.

Table 5. Ablation studies for the NSR Graph on SGD with JointGA. L stands for Layers, and H stands for Heads, for example, L1H1 indicates that the GAT network has one layer and one head.

Models	Unseen Domain	All Domain
w/o NSR Graph	22.9%	26.8%
NSR Graph-L1H1	23.5%	29.4%
NSR Graph-L2H3	**24.3%**	**31.8%**
NSR Graph-L4H5	24.2%	31.6%

6.3. The effect of the DSV Prompter

The DSV Prompter aims to ensure the generative properties of the decoder and effectively avoid confusion in decoding caused by the fake DSV sequence from turns that are unnecessarily tracked. The ablation outcomes for the DSV Prompter are shown in Table 6.

To test the effectiveness of the DSV Prompter, ablation experiments are conducted by removing the module and examining the impact on classification and the main experimental indicator JointGA under different coefficients. As shown in Table 6, after removing the classification module, the JointGA can only achieve 31.4%, while immediately increasing by 14.1% after completing the module. There are two main reasons for this. First, the overall model setting is affected. Without the classification module, the DSV Generator generates indiscriminately, resulting in a lot of noise that involves the learning and extraction of meaningful slot values. This is reflected in the prediction stage as frequent omission of slot value information and generation of 'None' values. However, with the DSV Prompter module, such noise phenomena can be effectively eliminated. Second, the DSV Prompter provides guidance similar to prompt learning during the training process, influencing the backpropagation process of the generator and enabling it to learn more profound

Journal Pre-proof

knowledge, i.e., to generate responses under the guidance of whether there is valuable information in the script. After multiple experiments, it was found that setting the weight to 0.1 when the lambda ranged from 0.01 to 0.20 could achieve the optimal gain for JointGA.

The ablation studies conducted demonstrate the effectiveness of the modules in the proposed method. In conclusion, the use of DialoGPT significantly enhances the performance of DST in complex scenes with causal and semantic-rich properties. The NSR Graph module effectively represents the dialog context and dynamically captures key DST information, introducing schema and background knowledge to new scenes and improving unseen domain adaptation. The integration of the modules in the proposed framework results in a complementary end-to-end DST span generation. As a result, the proposed methodology is suitable for hybrid causal inference and graph patterns.

Table 6. Ablation studies for the DSV Prompter effect of the Main Experiment, λ is the weight of DSV Prompter Loss in Total Loss, Classification Accuracy, and JointGA on MulitoWoz 2.1

Setting	Classification Accuracy	JointGA
w/o Span Classifier	-	31.4%
λ=0.01	92.5%	45.4%
λ=0.02	95.1%	46.1%
λ=0.05	97.7%	49.7%
λ=0.10	98.9%	**53.5%**
λ=0.15	**99.3%**	53.2%
λ=0.20	**99.3%**	51.4%

7. Discussion and implications

This study presents a novel approach to the DST task by introducing a noetic state representation graph (NSR-Graph). The proposed method leverages pre-trained language models, specifically DialoGPT, to learn schema knowledge through a predefined DSV sequence. By integrating these features, the proposed method generates a DST sequence that can dynamically capture the dialog state during

Journal Pre-proof

multi-turn dialog. The explicit graph representation of the NSR Graph enhances the model's ability to capture and retain the context of the dialog. Compared to previous graph-based methods, NSR Graph is unique for combining static and dynamic elements to predict dialog states in open-vocabulary settings. This feature allows the proposed model to incorporate both knowledge and dialog context, providing promising results for practical use.

For the theoretical implications, the proposed findings demonstrate the following: (1) the graph structure and model architecture of NSR Graph enable flexible access and utilization of external knowledge for downstream tasks. (2) NSR Graph is highly efficient in capturing the evolution of dialog dynamics and enhances model explainability with its explicit representation properties. To knowledge, the proposed study is the first to effectively combine static knowledge and dynamic dialog context to solve the DST task in a sequence generation mode, which is a significant contribution to the field.

For the practical implications, the proposed method enables easier error-tracking on downstream task models and performs better model controllability. Other graph-based algorithms can even be used for automatic error detection and correction in practical scenarios. Empirical results show the effectiveness of the proposed methodology. Besides, NSR Graph can be used in various downstream human-machine conversation tasks.

8. Conclusion

A dialog state representation graph, the NSR Graph, is proposed and incorporated into an end-to-end framework for DSV sequence generation. Compared with traditional methods that represent dialog states with only dialog history, the dialog context is explicitly formalized with the dynamic construction of the NSR Graph, which introduces schema and background knowledge of dialog context, effectively guiding DST. Experiments show that the NSR Graph achieves competitive results on the MultiWOZ dataset compared to traditional methods and even outperforms them on the SGD dataset. This indicates that the proposed method can guarantee high JointGA in the DST task with explainability and generalizability, making it potentially applicable in practical scenarios.

Journal Pre-proof

Journal Pre-proof

The NSR Graph has been applied in real-world scenarios, but the current reliance on structured schema knowledge still poses limitations, particularly in open-domain conversations. This reliance may restrict the model's ability to effectively handle diverse, unstructured dialog scenarios, making it less adaptable to zero-shot and few-shot DST tasks. To address these limitations, the development of schema-free DST mechanisms is a key focus, enabling better handling of open-domain dialogues without the need for predefined schemas. Additionally, optimizing graph-based reasoning for increased efficiency and scalability will be critical for real-time applications. Future work will also involve integrating few-shot learning strategies to improve performance in low-data scenarios, crucial for applications encountering infrequent or novel dialog states. Another direction is the incorporation of human feedback mechanisms, similar to those used in systems like ChatGPT, to enhance adaptability, refine responses, and facilitate continuous learning. These mechanisms will allow the system to adjust and improve over time based on user interactions. Furthermore, integrating the CopyNet approach, akin to SaCLog, is planned to enhance DST JointGA by improving named entity recognition and dialog state extraction. These improvements will strengthen the system's ability to capture and respond to complex dialog states, ensuring a more accurate and effective DST process. The NSR Graph will also be incorporated into dialog policy learning and response creation, enhancing the overall flow and coherence of responses. By adopting the NSR Graph alongside dialog policies, the system can track dialog states accurately while generating appropriate and contextually relevant responses.

References

1. J. Sun, J. Kou, W. Hou, Y. Bai,. A multi-agent curiosity reward model for task-oriented dialogue systems. Pattern Recognit. 157(2025) p.110884.
2. H. Yu, Y. Ko, Enriching the dialogue state tracking model with a asyntactic discourse graph. Pattern Recognit. Lett. 169(2023) 81-86.
3. M. Heck, N. Lubis, C.V. Niekerk, S. Feng, C. Geishauser, H.C. Lin, M. Gašić, Robust dialogue state tracking with weak supervision and sparse data. Trans. Assoc. Comput. Linguist. 10(2022) 1175-1192.
4. A. Ohashi, R. Higashinaka, Optimizing pipeline task-oriented dialogue

Journal Pre-proof

Journal Pre-proof

systems using post-processing networks. Comput. Speech Lang. 90(2025) p.101742.

5. Y. Park, Y. Ko, J. Seo, BERT-based response selection in dialogue systems using utterance attention mechanisms. Expert Syst. Appl. 209(2022) p.118277.
6. M. Zhao, L. Wang, H. Ji, Z. Jiang, R. Li, X. Lu, Z. Hu, Mutually improved response generation and dialogue summarization for multi-domain task-oriented dialogue systems. Knowl. Based Syst. 279 (2023) p.110927.
7. 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) p.101460.
8. Y. Yang, H. Huang, Y. Gao, J. Li, Building knowledge-grounded dialogue systems with graph-based semantic modelling. Knowl. Based Syst. 298 (2024) p.111943.
9. H. Xie, J. Chen, Y. Lin, L. Zhang, G. Wang, K. Xie, External knowledge document retrieval strategy based on intention-guided and meta-learning for task-oriented dialogues. Adv. Eng. Inform. 56(2023) p.102020.
10. Z. Huang, F. Li, J. Yao, Z. Chen, MGCRL: Multi-view graph convolution and multi-agent reinforcement learning for dialogue state tracking. Neural Comput. Appl. 36(9) (2023) 4829-4846.
11. M.A. Khan, Y. Huang, J. Feng, B.K. Prasad, Z. Ali, I. Ullah, P. Kefalas, A multi-attention approach using BERT and stacked bidirectional LSTM for improved dialogue state tracking. Appl. Sci. 13(3) (2023) p.1775.
12. Li, J., Song, S., Li, Y., Zhang, H. and Hu, G., 2024. ChatMDG: A discourse parsing graph fusion based approach for multi-party dialogue generation. Information Fusion, 110, p.102469.
13. Y. Guo, Z. Xie, X. Chen, H. Chen, L. Wang, H. Du, S. Wei, Y. Zhao, Q. Li, G. Wu, ESIE-BERT: Enriching sub-words information explicitly with BERT for intent classification and slot filling. Neurocomputing. 591 (2024) p.127725.
14. X. Jia, R. Zhang, M. Peng, Multi-domain gate and interactive dual attention for multi-domain dialogue state tracking. Knowl. Based Syst. 286 (2024)

Journal Pre-proof

p.111383.

15. 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) p.110069.
16. M.A. Khan, B.K. Prasad, G. Qi, W.Song, F. Ye, Z. Ali, I. Ullah, P. Kefalas, UTMGAT: a unified transformer with memory encoder and graph attention networks for multidomain dialogue state tracking. Appl. Intell. 54(17) (2024) 8347-8366.
17. M. Zhu, X. Xu, ECDG-DST: A dialogue state tracking model based on efficient context and domain guidance for smart dialogue systems. Comput. Speech Lang. (2024) p.101741.
18. C. You, D. Xiong, TCLNet: Turn-level contrastive learning network with reranking for dialogue state tracking. Knowl. Based Syst. 302(2024) p.112308.
19. H. Jeon, G.G. Lee, DORA: Towards policy optimization for task-oriented dialogue system with efficient context. Comput. Speech Lang. 72(2022) p.101310.
20. Y. Lee, T. Kim, H. Yoon, P. Kang, J. Bang, M. Kim, DSTEA: Improving Dialogue State Tracking via Entity Adaptive pre-training. Knowledge-Based Systems, 290 (2024) p.111542.
21. S. Cao, Y. Jia, C. Niu, H. Zan, Y. Ma, S. Xu, Generating emotional responses with dialogpt-based multi-task learning. In CCF International Conference on Natural Language Processing and Chinese Computing (pp. 485-496) (2022 September). Cham: Springer International Publishing.
22. J. Zhang, Y. Feng, J. Zhang, Y. Li, The Short Time Prediction of the Dst Index Based on the Long-Short Time Memory and Empirical Mode Decomposition–Long-Short Time Memory Models. Appl. Sci. 13(21) (2023) p.11824.
23. Dataset available at: https://github.com/google-research-datasets/dstc8-schema-guided-dialogue
24. Dataset available at: https://github.com/budzianowski/multiwoz/tree/master/data

Journal Pre-proof

25. 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.
26. J. Qiu, Z. Lin, H. Zhang, Y. Yang, Hierarchical temporal slot interactions for dialogue state tracking. Neural Comput. Appl. 35(8) (2023) 5791-5805.
27. P. Paromita, A. Khader, S. Begerowski, S.T. Bell, T. Chaspari, Linguistic and vocal markers of microbehaviors between team members during analog space exploration missions. IEEE Pervasive Comput. 22(2) (2023) 7-18.
28. A. Dunn, D. Inkpen, R. Andonie, Designing and Evaluating Context-Sensitive Visualization Models for Deep Learning Text Classifiers. In Artificial Intelligence and Visualization: Advancing Visual Knowledge Discovery (pp. 399-421) (2024). Cham: Springer Nature Switzerland.
29. T. Nguyen-Mau, A.C. Le, D.H. Pham, V.N. Huynh, An information fusion based approach to context-based fine-tuning of GPT models. Inf. Fusion, 104 (2024) p.102202.
30. X. Zhou, T. Zhang, C. Cheng, S. Song, Dynamic multichannel fusion mechanism based on a graph attention network and BERT for aspect-based sentiment classification. Appl. Intell. 53(6) (2023) 6800-6813.
31. Z. Chen, Y. Liu, L. Chen, S. Zhu, M. Wu, K. Yu, Opal: Ontology-aware pretrained language model for end-to-end task-oriented dialogue. Trans. Assoc. Comput. Linguist. 11(2023) 68-84.
32. Q.B. Liu, S.Z. He, C. Liu, K. Liu, J. Zhao, Unsupervised Dialogue State Tracking for End-to-End Task-Oriented Dialogue with a Multi-Span Prediction Network. J. Comput. Sci. Technol. 38(4) (2023), 834-852.
33. S. Gao, R. Takanobu, A. Bosselut, M. Huang, End-to-end task-oriented dialog modeling with semi-structured knowledge management. IEEE/ACM Trans. Audio Speech Lang. Process. 30 (2022) 2173-2187.

Journal Pre-proof

Journal Pre-proof

Declaration of interests

- ✓ 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.

- ✓ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

None

Journal Pre-proof