PeerJ Computer Science

PluDG: enhancing task-oriented dialogue system with knowledge graph plug-in module

Xuelian Dong and Jiale Chen

School of Computer Science, University of South China, Hunan, China

ABSTRACT

Task-oriented dialogue systems continue to face significant challenges as they require not only an understanding of dialogue history but also domain-specific knowledge. However, knowledge is often dynamic, making it difficult to effectively integrate into the learning process. Existing large language model approaches primarily treat knowledge bases as textual resources, neglecting to capture the underlying relationships between facts within the knowledge base. To address this limitation, we propose a novel dialogue system called PluDG. We regard the knowledge as a knowledge graph and propose a knowledge extraction plug-in, Kg-Plug, to capture the features of the graph and generate prompt entities to assist the system's dialogue generation. Besides, we propose Unified Memory Integration, a module that enhances the comprehension of the sentence's internal structure and optimizes the knowledge base's encoding location. We conduct experiments on three public datasets and compare PluDG with several state-of-the-art dialogue models. The experimental results indicate that PluDG achieves significant improvements in both accuracy and diversity, outperforming the current state-of-the-art dialogue system models and achieving state-of-the-art performance.

Subjects Human-Computer Interaction, Artificial Intelligence, Data Mining and Machine Learning, Data Science, Natural Language and Speech
Keywords Artificial intelligence, Natural language processing, Data science, Graph neural networks, Dialogue systems

Submitted 8 September 2023
Accepted 27 October 2023
Published 24 November 2023

Corresponding author
Jiale Chen, jalorc@163.com

Academic editor
Binh Nguyen

Additional Information and Declarations can be found on page 14

DOI **10.7717/peerj-cs.1707**

INTRODUCTION

Building task-oriented dialogue systems has become a prevalent research subject in both academic and business settings. The commonly used method to create a dialogue system is developing an end-to-end system, which increases efficiency by generating responses directly from a knowledge base and dialogue history (*Lu et al., 2023a*; *Lu et al., 2023b*; *Liu et al., 2023b*). Figure 1 depicts the whole data needed by the task-oriented dialogue system. To make full use of the external knowledge base information, *Madotto, Wu & Fung (2018)* proposed Mem2Seq. The model enhances the MemNN framework (*Sukhbaatar et al., 2015*) using a sequence generation framework and incorporates a global multi-hop attention mechanism to replicate words directly from the dialogue history or knowledge base. In addition, some researchers propose that entities' relationships in an external knowledge base should be considered rather than treated as isolated triples. *Banerjee & Khapra (2019)* achieved state-of-the-art results in goal-directed dialog systems using GCN

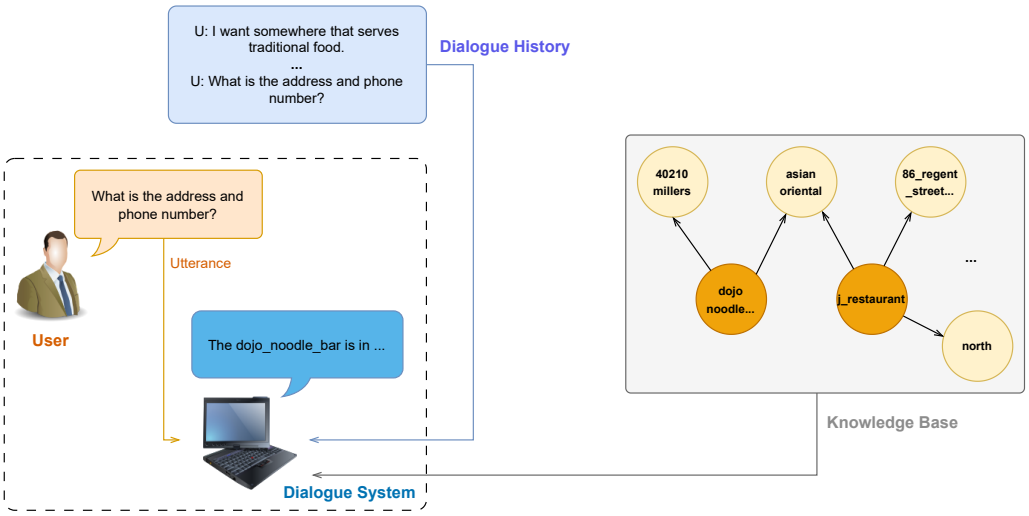


Figure 1 **Illustration of a task-oriented dialogue system about navigation.**
Full-size DOI: 10.7717/peerjcs.1707/fig-1

(*Kipf & Welling, 2016*) to combine structural information with encoded sequences and developed contextual graphs for constructing hybrid dialogues in different languages. Later, *Zhao et al. (2023)* proposed a multi task learning method based on graph attention networks for modeling a multi-domain task-oriented dialogue system.

On the other hand, researchers have also utilized large language models (LLMs) in task-oriented dialogue systems by treating the response as the natural language generation task. One such system is UBAR (*Yang, Li & Quan, 2021*), a modularly designed task-based dialogue system based on GPT-2 that facilitates module replacement and functional extensions for different domains and scenarios. *Rony, Usbeck & Lehmann (2022)* proposed DialoKG, a model that incorporates knowledge into the GPT-2 architecture. To achieve this, the model leverages the structural information of the knowledge base by treating each entity as a sequence and calculating its weight for the dialogue history with the help of RoBERTa (*Liu et al., 2019*). Nevertheless, LLMs may face challenges in capturing these structured relations when processing knowledge bases to treat entities as sequences since the information contained in knowledge bases is usually structured, consisting of entities and their relations (*Shen et al., 2021*; *Liu et al., 2023a*).

To address this limitation, this article presents a novel method called PluDG (PLUgins-Assisted Dialogue Generation). Specifically, we designed a plug-and-play module called Kg-Plug, which treats knowledge as a knowledge graph. Kg-Plug utilizes LR-GCN modules to leverage low-dimensional decomposition for feature extraction. Furthermore, it employs the attention mechanism to align with the dialogue history to get the prompt entities, which are inferred from the dialogue history and knowledge base and are related to the user's true intent. Subsequently, prompt entities are generated and provided to the decoder for dialogue generation. Additionally, we employ a GPT-2-based decoder for generating responses. We enhance it by incorporating an entity memory ensemble embedding, which

utilizes special tokens and embeddings to improve GPT-2′s ability to produce contextually appropriate results.

Our article outlines several major contributions:

- We proposed PluDG, a task-oriented dialogue system that integrates a plug-and-play Kg-Plug component into a GPT-2-based decoder. PluDG learns intrinsic graph structure information from the knowledge base and gets entity hints to pass to the decoder for better response generation.
- We proposed a novel embedding technique for GPT-2, named Unified Memory Integration (UMI), which utilizes multi-layered and position embeddings that are aware of the structure of the dialogue history, knowledge base, and prompt entities.
- Experiment results on three benchmark datasets show the superior performance of PluDG compared to other state-of-the-art models. Our model outperforms existing approaches based on metrics, particularly in complex knowledge-base information datasets.

RELATED WORKS

A task-oriented dialog system has been employed with an end-to-end approach. Originally, researchers considered the KB and dialogue history as sequences. Lately, many researchers have emphasized the importance of preserving the connection between entities in the KB to achieve improved bot responses. The most recent studies have applied pre-trained language models to enhance dialog systems.

RNN-based dialogue systems. *Wen et al. (2016)* proposed a web-based task-oriented dialogue system capable of directly learning parameters from raw data. Later, *Wu, Socher & Xiong (2019)* proposed GLMP that integrates the external knowledge base. The external knowledge utilized an end-to-end memory network (MN), storing word-level information about the knowledge base and conversation history. Regrettably, prior studies have failed to acknowledge the plentiful structural information present in knowledge bases, specifically the graph structural information formed by entity-entity relationships.

Knowledge graph-augmented dialogue systems. Graph neural networks are also used by some researchers to encode knowledge-base entities. *He et al. (2020)* developed Fg2Seq, which can integrate the latent semantics of conversation history, improving the description of entities and enabling better inference of knowledge related to conversation history. *Wu, Harris & Zhao (2022)* employed a GMN to comprehend the intrinsic patterns in the dialogue history and their connection with the KB. Although this method treats the KB as a graph, their decoders are still based on RNN, which does not provide a superior understanding of contextual information compared to the GPT.

Pretrain-language-model-based dialogue systems. *Madotto et al. (2020)* employed a strategy called knowledge embedding to embed knowledge bases directly into model parameters. This approach does not require dialogue state tracking or template responses as inputs and can dynamically update its knowledge base through fine-tuning. Recently, *Huang, Quan & Wang (2022)* proposed a task-oriented dialog model that employs an Auto-regressive Entity Generation technique, which consists of three major components:

a GPT-2 that generates replies, an entity generator that identifies entities in the responses, and a final stage that embeds the entities to generate the ultimate dialog response. It is an end-to-end task-oriented dialogue model that combines natural language processing and generation methods.

In contrast to previous studies, our work introduces PluDG, a novel task-oriented dialogue system. PluDG incorporates Kg-Plug, a plug-and-play component, to extract features from the knowledge base and align them with the dialogue context before passing prompt entities to the decoder. Additionally, to enhance the decoder's comprehension of the underlying semantic information, we employ the UMI module to provide the structure of the knowledge base and dialogue history.

METHOD

Prior to presenting the complete method, we provide a description of the problem.

For the given dialogue history, we regard the utterance of the user as U and the system's response as S. For given turn i, dialogue consists of T_i, which is made up of U_i and S_i: $T_i = (U_i, S_i)$. If we assume that there have been K turns in the dialogue history, then the entire history can be defined as $T = [T_1, T_2, T_3, \ldots, T_K]$.

Regarding the knowledge base, we utilize the triple format $G = (e, r, o)$ to represent various entities and their relationships. Note that, e refers to the entities, r represents the relationships, and o represents objects. For instance, in the case of the ith potential triple G_i, $G_i = (j_{restaurant}, place, north)$.

Suppose there are n entities for a given turn i, then we use K_i to denote the given knowledge base construct by the format above mentioned $K_i = (G_1, G_2, \ldots, G_n)$.

The probability distribution of responses generated by the language model in the ith turn is formally defined as follows:

$$P(S_i|T_{1:i-1}, U_i, K_i) = \prod_{j=1}^{N} P(s_j|s_{1:j-1}, T_{1:i-1}, U_i, K_i), \tag{1}$$

where $S_i = [s_1, \ldots, s_n]$ represents the response generated from the ith round of the system, and N is the maximum number of words in the response S_i. The $1 : j-1$ represents elements 1 to $j-1$.

Overview

To address the problem that LLMs may face challenges in capturing these structured relations when processing knowledge bases, treat entities as sequences. We propose a model called PluDG. This model is composed of three components: the Kg-Plug and the Decoder. More details are shown in Fig. 2.

Kg-Plug module

The Kg-Plug module is designed as a plug-and-play component, as illustrated in Fig. 3. It treats the provided knowledge as a graph and employs LR-GCN for feature extraction.

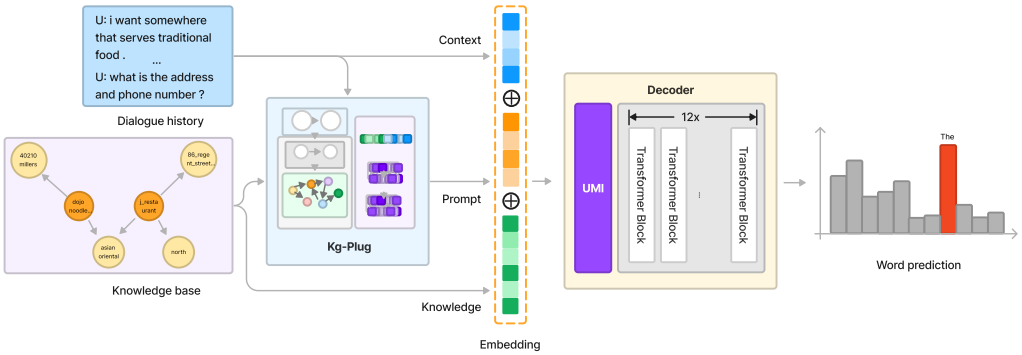


Figure 2 **Overview of the architecture of the PluDG model.**
Full-size DOI: 10.7717/peerjcs.1707/fig-2

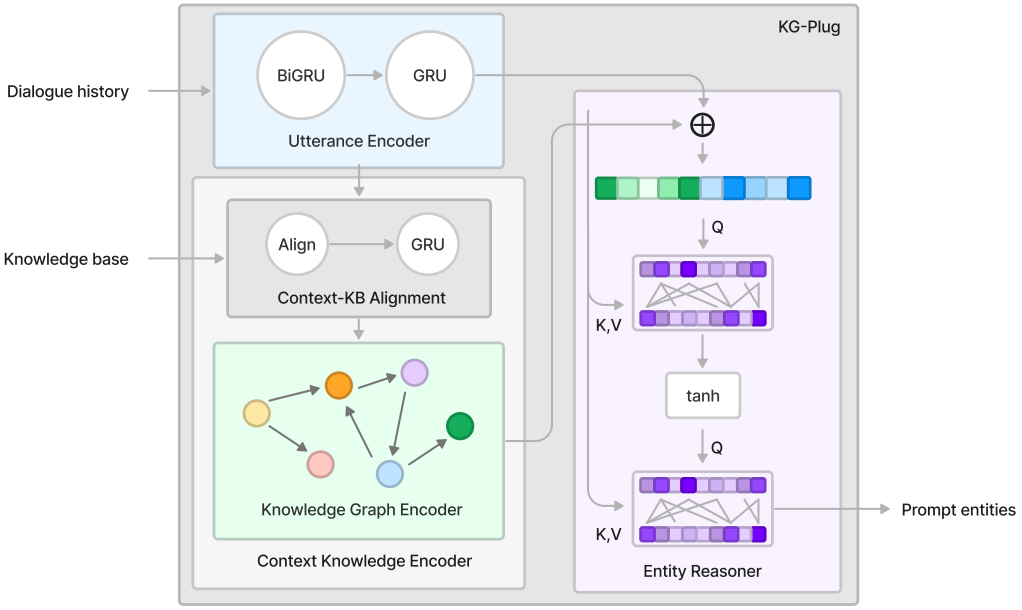


Figure 3 **The schematic diagram of the Kg-Plug module includes the utterance encoder, context knowledge encoder, and entity reasoner components.**
Full-size DOI: 10.7717/peerjcs.1707/fig-3

Subsequently, it infers the most probable entity hints based on the dialogue history information. Finally, the prompt entities are passed to the decoder.

Utterance encoder

Assuming that there are K turns in the dialogue history, the history contains $2K-1$ utterances, where each utterance includes L_i words. The words in the ith utterance are represented by word w_{il}, where $L \in [1, L_i]$. First, a Bi-GRU, which includes both a forward unit and a backward unit, is used to obtain the hidden representation of the sentences:

$$H_i = h_{i1}, h_{i2}, h_{i3}, \ldots, h_{il} = BiGRU(Emb(w_{il})), \tag{2}$$

where $Emb(w_{il})$ represents the embedding state of the word w_{il}.

Next, a self-attention unit is utilized to capture the contextual information of each token in order to obtain a comprehensible semantic representation of the utterance, as shown below:

$$\mu_{il} = \tanh(W_w h_{il} + b_w), \tag{3}$$

$$\alpha_{il} = \frac{\exp(\mu_{il} u_w)}{\sum_l \exp(\mu_{il} u_w)}, \tag{4}$$

$$v_i = \sum_l \alpha_{il} h_{il}, \tag{5}$$

where W_w, b_w, u_w are trainable parameters of the model.

Lastly, a GRU is utilized to encode the utterance vector v_i:

$$H_i^c = GRU(v_i), i \in [1, 2K-1]. \tag{6}$$

Context knowledge encoder

The Context Knowledge Encoder is employed to extract hidden information from both the dialogue history and knowledge base.

Context-KB Alignment. Following *Chen et al. (2017)*, the Context-KB Alignment module aims to capture the alignment representation of each entity in the knowledge base through the incorporation of dialogue history. To achieve this goal, an attention mechanism is employed to align the dialogue history embedding with the knowledge base entity embedding, allowing for the creation of a coherent representation of the graph. Specifically, the module concatenates each word w_{il} with the entity representation e, applies a tanh activation, and derives attention scores through a Softmax operation. These scores are then multiplied with the corresponding words and summed to generate an aligned representation of the entity's conversation history:

$$c_{il} = \tanh(W_e[Emb(e); Emb(w_{il})] + b_e), \tag{7}$$

$$\alpha_{il} = \frac{\exp(c_{il} u_e)}{\sum_l \exp(c_{il} u_e)}, \tag{8}$$

$$f_{align}^i(e) = \sum_l \alpha_{il} Emb(w_{il}), \tag{9}$$

where W_e, b_e, and u_e are trainable weight parameters, and $[;]$ denotes the concatenation.

Next, the jth entity embedding $Emb(e_j)$ is concatenated with its correspondingly aligned embedding $f_{align}^i(e_j)$. In this way, we obtain a sequence of history-alignment entity input

representations. Then, the sequence is passed to the GRU unit to obtain a more robust history-alignment entity representations. Formally, for each entity e_j, the representation f_{ij} is obtained as follow:

$$f_{ij} = GRU([Emb(e_j); f^i_{align}(e_j)]). \tag{10}$$

Knowledge Graph Encoder. In this section, we introduce a GCN (*Kipf & Welling, 2016*) to extract the intrinsic features of the knowledge graph. However, inspired by *Hu et al. (2021)*, we leveraged the low-rank decomposition into the weights of GCN and named this new module LR-GCN. For given weights $W_0 \in \Re^{x \times y}$, we use $W_o + \Delta W = W_o + BA$ to replace the update, where $B \in \Re^{x \times y}$, $A \in \Re^{y \times z}$, and $y << \min(x, z)$.

In this section, we represent each entity as a node, where N represents the set of nodes. The relationships between entities are denoted as edges, and R represents the set of edges. Following the Context-KB Alignment operation, each entity in the dialog history has $2K-1$ representations, which correspond to the $2K-1$ utterances spoken. To capture the features from each node and its neighborhoods, we employ the GCN in the Graph operation:

$$g_{ij} = \sigma\left(\sum_{r \in R}\sum_{v \in N^r_i} \frac{1}{|N^r_i|} W_r f_{iv} + W_0 f_{ij}\right). \tag{11}$$

In Eq. (11), N^r_i denotes the set of neighborhood-indices of entity i under relation r, $r \in R$; W_r and W_o are trainable parameters. An activation function $\sigma()$ is adopted in this research, and ReLU is the specific function utilized.

Finally, an appropriate pooling method is used to fuse the data in g_{ij} and f_{ij} to obtain a question and text representation matrix G^f:

$$\vartheta_{ij} = \tanh(W_g[f_{ij}; g_{ij}] + b_g), \tag{12}$$

$$\alpha_{il} = \frac{\exp(\vartheta_{ij} u_g)}{\sum_l \exp(\vartheta_{ij} u_g)}, \tag{13}$$

$$g^f_i = \sum_l \alpha_{il}[f_{ij}; g_{ij}], \tag{14}$$

where W_g, b_g, and u_g are trainable weight parameters, $[;]$ denotes the concatenation, and $G^f = [g^f_1, \ldots, g^f_{(2K-1)}]$.

Entity reasoner

The entity reasoner is an important component of the Kg-Plug. In this component, we concatenate the Utterance Encoder's output and Context Knowledge Encoder's output as q^r_0, formally:

$$q^r_0 = [H^c; G], \tag{15}$$

then use to two-hop attention to get the final entity probability.

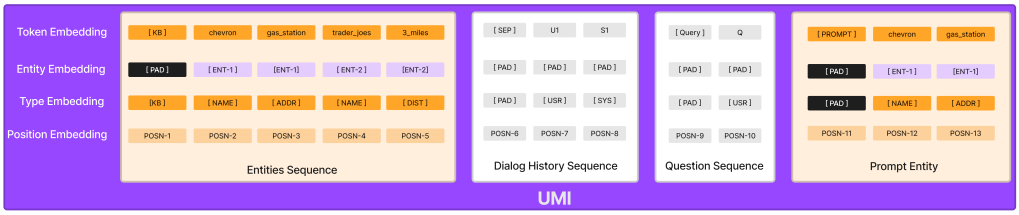


Figure 4 **Illustration of Unified Memory Integration.**
Full-size DOI: 10.7717/peerjcs.1707/fig-4

Two-Hop update. In the reason stage, followed by the MemNN, we design a two-hop update mechanism to get the precise entity. For the sake of clarity in our description, we denote the number of hops as X, where $X = 2$. For the given hidden state, q_0^r, we use learnable attention to search for deeper information. In each hop have the following:

$$q_{i+1}^r = \tanh(W_q q_i^r), i \in [0, X]. \tag{16}$$

In the final hop, we use the Softmax function to get the final entity probability p^{ent}:

$$p^{ent} = Softmax(G^T W_e q_X^r). \tag{17}$$

Decoder

In this study, the decoder is based on the GPT-2 model and is responsible for generating the final response.

Unified Memory Integration. As shown in Fig. 4, to incorporate entity structural information from the Knowledge base and prompt entities from Kg-Plug into the GPT-2, we use various embedding techniques, including entity embedding, type embedding, as well as the traditional word token and positional embedding. These techniques enable the decoder to extract the knowledge graph structure, which is linearized into a sequence as input, with special tokens ([NAME] and [ADDR] etc.) to separate the subject, relation, and object of an entity. The entity embedding layers capture entity-level separate information about the word token, and the type embedding distinguishes the relevant tokens. Furthermore, we incorporate speaker information into the dialogue history. To differentiate between the system's response and the user's utterance, we employ the [SYS] token for system responses and the [USR] token for user utterances. Additionally, we use [Query] to indicate the user's current utterance for clear separation.

For generating responses, the GPT-2 decoder relies heavily on the input sequence, and the sequence of tokens plays a crucial role in determining the output. We position the prompt entities after the history, as shown in Fig. 4, in order to enhance the generation process. By doing so, we hope the decoder can draw upon a more precise context, which improves its ability to understand user queries and generate appropriate responses.

Calculate the modeling response word's probability p^{final} by using the embedding token as follows:

$$h_0^t = e_{x'} W_v + W_p, \tag{18}$$

$$h_l^t = TransformerBlock(h_{l-1}^t), \tag{19}$$

$$p^{final} = Softmax(h_l^t W_v), \tag{20}$$

where $e_{x'}$ presents x' in one-hot representation, W_v presents the word vector mask, W_p is the position mask, and $l \in L$ presents the Transformer layers.

EXPERIMENTS

Datasets

We evaluate our model on three publicly available benchmark datasets: CamRest (*Wen et al., 2016*), In-Car Assistant (*Eric & Manning, 2017*), MultiWOZ 2.1 (*Budzianowski et al., 2018*). Details of each dataset are provided below:

- **CamRest.** The dataset comprises dialogs in the restaurant reservation domain, consisting of 676 multi-turn dialogs with an average of five turns per dialog. Additionally, each dialog has an average of 22.5 KB of triples. To conduct our experiments, following *Rony, Usbeck & Lehmann (2022)*, we partitioned the dataset into training, validation, and test sets, with 406, 135, and 135 dialogs, respectively.
- **In-Car Assistant.** The dataset contains 3,031 multi-turn dialogs across three distinct domains: weather, navigation, and schedule. On average, each dialog comprises 2.6 turns, but the knowledge base (KB) information in every dialog has an average of 62.3 triples. Following *Rony, Usbeck & Lehmann (2022)*, we partitioned the In-Car Assistant dataset into training, validation, and test sets, consisting of 2,425, 302, and 304 dialogs, respectively, for use in our experiments.
- **Multi-WOZ 2.1.** The dataset comprises three distinct domains: attractions, hotels, and restaurants. Each dialog in the dataset has an average of 5.6 turns and 54.4 KB of triples. To process the data, we followed the method used by *Rony, Usbeck & Lehmann (2022)* and divided the dataset into training, validation, and test sets, each containing 1,839, 117, and 141 dialogs, respectively.

Baselines

For our PluDG model, we employ some of the recently proposed state-of-the-art models, including GLMP (*Wu, Socher & Xiong, 2019*), DF-Net (*Qin et al., 2020*), Fg2Seq (*He et al., 2020*), GPT-2+KE (*Madotto et al., 2020*), CDNet (*Raghu et al., 2021*), GraphMemDialog (GMD) (*Wu, Harris & Zhao, 2022*), and DialoKG (*Rony, Usbeck & Lehmann, 2022*). All comparison models were evaluated in the same experimental environment.

Metrics

We utilize two popular evaluation metrics in dialogue studies to evaluate our model: BLEU (*Papineni et al., 2001*) and Entity F1. To ensure a fair comparison with previous work, we adopted these widely used metrics in the community.

Table 1 Training parameters.

Parameter	Kg-Plug	GPT-2+UMI+Kg-Plug
Batch size	8	2
Learning rate	$1e^{-4}$	$6.25e^{-5}$
Epoch	20	10
Dropout	0.2	0.2
Embedding size	128	768
Max gradient norm	1	1

- **BLEU.** The Bilingual Evaluation Understudy (BLEU) metric measures the n-gram overlap between generated responses and gold standard responses.
- **Entity F1.** We use Entity F1 to assess the system's ability to produce relevant entities that can accomplish specific tasks by retrieving accurate entities from the provided knowledge base. To compute the Entity F1 score, we micro-average the precision and recall over knowledge base entities of the generated responses.

Model training

The cross-entropy is utilized to direct the model-training process. Specifically, the negative log likelihood is calculated between the predicted and actual distributions of the training data:

$$L(D) = -\sum_{j}^{|D|}\sum_{i}^{n}\log p(s_i^j|s_{1:i}^j, T, K), \tag{21}$$

where D is the dialogue dataset consisting of $D_1, D_2, \ldots, D_i$.$|D|$ is the number of the dialogue datasets. Let s_i^j be the response generated by the model at D_j, corresponding to the words output by the ith time step of the model. Here, n represents the maximum response length, while dialogue history T and knowledge base K are given by D_j.

Training settings

We employed the PyTorch framework to implement our model, which was trained on an NVIDIA GeForce GTX 3070 with 8 GB of GPU memory. Our experiments entailed setting the Kg-Plug's embedding dimensions and hidden units to 128, while the batch size was set to 8. Additionally, we set the number of hops for the Entity Reasoner at 2.

For the decoder, we used the normal pretrained GPT-2 with 137M parameters. The model underwent end-to-end training utilizing the AdamW23 optimizer, with the learning rate was set to $6.25e^{-5}$ and the decay was set to $1e^{-8}$. For all the datasets, the dropout ratio was set at 0.2. More hyper-parameters used to train PluDG are listed in Table 1.

Evaluation results

Table 2 illustrates the superior performance of our model compared to baseline models on three datasets, as demonstrated by both BLEU (*Papineni et al., 2001*) and Entity-F1 metrics. Additionally, we present the architectures the models utilized. Our experimental results indicate that PluDG achieves a BLEU score of 23.0 and an F1 score of 76.9 on the CamRest dataset, along with a significantly improved BLEU score of 21.6 and 69.5 Entity-F1 score on

Table 2 Comparison of generation results on three datasets.

Model	Structure	CamRest		In-Car Assistant		MultiWOZ 2.1	
		BLEU	Entity F1	BLEU	Entity F1	BLEU	Entity F1
GLMP	RNN	–	–	8.5	58.4	–	–
DFNet	RNN	–	–	9.00	62.7	3.4	34.8
FG2Seq	RNN+GCN	13.2	62.2	10.4	62.0	–	–
GPT-2+KE	GPT-2+KE	17.8	54.0	16.8	58.6	12.7	35.6
CDNet	DNN	19.1	63.1	16.0	57.4	10.5	30.6
GMD*	GMN+GAT	22.3	64.4	18.8	64.5	14.9	40.2
DialoKG	GPT-2+RoBerta	22.5	75.4	18.4	64.9	7.4	39.1
PluDG (Ours)	GPT-2+Plug-in	23.0	76.9	21.6	69.5	9.2	42.4

Table 3 Results of the ablation study.

Model	CamRest		In-Car assistant		MultiWOZ 2.1	
	BLEU	Entity F1	BLEU	Entity F1	BLEU	Entity F1
PluDG	23.0	76.9	21.6	69.5	9.2	42.4
w/o Kg-Plug	22.4	73.2	20.3	64.7	8.8	40.5
w/o UMI	22.8	75.2	6.3	72.7	8.2	41.9
w/o Both	21.7	74.6	18.2	64.8	7.4	39.0

the In-Car Assistant dataset, showcasing its capability to generate more fluent responses. Moreover, we achieve a higher Entity-F1 score of 42.4 on WOZ2.1, despite obtaining a BLEU score of 9.2. Notably, PluDG outperforms the previously similarly structured DialoKG (*Rony, Usbeck & Lehmann, 2022*) model in different domains, highlighting the effectiveness of Kg-Plug in constructing knowledge graph extraction features and providing effective prompt entities. Additionally, UMI modules effectively leverage deep semantic information, further contributing to the model's response efficacy. The same trend of improvement is observed in three datasets, indicating the generalization ability of our model.

Ablation study

To assess the necessity of each component in PluDG, we conducted an ablation study by removing the Kg-Plug and Unified Memory Integration (UMI) modules and analyzing their impact on the performance of the framework. As shown in Table 3, our results indicate that these two modules are essential for achieving high performance in task-oriented dialogue generation tasks.

After removing Kg-Plug, a component added as a plug-in to the model, we observed a significant drop in performance for various evaluation indicators, particularly Entity F1 of CamRest and In-Car Assistant, both decreasing by more than three points. We speculate that the Prompt entities provided to the GPT-2 decoder play a vital role in generating responses. Conversely, removing the UMI module leads to a performance drop across all the three datasets. Although the BLEU index of the In-Car dataset experienced the most significant drop, exceeding 15 points, the Entity F1 indicator increased. Thus, we

Table 4 Result of the significance test.

Metrics	*t*-statistics	*p*-value
BLEU	3.7871	0.0193
Entity F1	5.8948	0.0042
Both	3.5881	0.0049

conjecture that the GPT-2 model heavily relies on input sequences for response generation, and labeled information significantly impacts the output. By incorporating more semantic information, the GPT-2 model obtains a more accurate and comprehensive context, leading to more relevant responses. Additionally, when all the extra modules were removed, we observed a drop in all indicators, performing even worse than the previous baseline model. In conclusion, our ablation study emphasizes the critical importance of Kg-Plug and UMI in PluDG, as they are essential for achieving state-of-the-art performance in task-oriented dialogue tasks.

Significance test

To rigorously assess the significance of the performance improvement in our proposed method, we conducted an evaluation using the *t*-test method. We compared PluDG with the best model. The comparison is divided into BLEU significance, Entity F1 significance, and the significance comparison of both. The results, presented in Table 4, demonstrate that our PluDG exhibits significantly improved performance metrics compared to the best baselines, with all *p*-values below the 0.05 significance level.

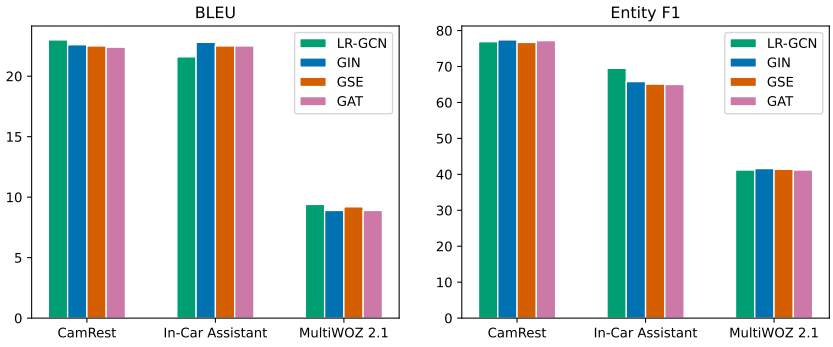
Comparison with other GNN models

Our proposed approach, PluDG, exhibits significant improvements over existing baselines. We hypothesize that this improvement can be traced to the Kg-Plug for the powerful graph feature extraction. To test our hypothesis, we compared four different GNNs, including GIN24, GSE25, and GAT26, all of which were modified to directly replace our original LR-GCN modules, ensuring a fair evaluation.

Figure 5A illustrates that LR-GCN outperforms other GNNs in terms of BLEU on the Camrest and MultiWoZ2.1 datasets, but its score is comparatively lower on the In-Car Assistant dataset. In Fig. 5B, LR-GCN exhibits a slightly higher Entity F1 score compared to other GNNs on the Camrest and MultiWoZ2.1 datasets, and significantly outperforms them on the In-Car Assistant dataset. Overall, while different GNNs offer unique advantages for specific datasets, our LR-GCN approach demonstrates the most substantial cumulative improvement in two evaluation metrics across all three datasets. We attribute this observation to LR-GCN's utilization of low-rank matrix weight factorization to prevent overfitting and potentially better capture the global characteristics of the entire graph.

Case study

Figure 6 displays the responses of PluDG along with multiple baseline models over 3 rounds of the In-Car Assistant dataset. In the first round, PluDG accurately answered the



(a) (b)

Figure 5 **Performance comparison of Kg-Plug with representative GNNs.** (A) Comparison results in BLEU. (B) Comparison results in Entity F1.

Full-size DOI: 10.7717/peerjcs.1707/fig-5

Knowledge Triples

Subject	Relation	Object
trader_joes	distance	5_miles
trader_joes	traffic_info	heavy_traffic
trader_joes	poi_type	grocery_store
home	distance	3_miles
home	traffic_info	heavy_traffic
home	poi_type	home
home	address	56_cadwell_street
teavana	distance	3_miles
dominos	distance	3_miles
dominos	poi_type	pizza_restaurant
whole_foods	distance	3_miles
whole_foods	address	819_alma_st
comfort_inn	traffic_info	no_traffic
comfort_inn	poi_type	rest_stop
panda_express	distance	2_miles
panda_express	traffic_info	no_traffic
tai_pan	distance	4_miles
tai_pan	traffic_info	road_block_nearby
tai_pan	address	830_almanor_ln

Round 1

User	where is home
Truth	home is where is your heart is , at 56_cadwell_street
DialoKG	your home is 3_miles away , i sent its position on your screen , drive carefully !
PluDG	your home is at 56_cadwell_street
Fg2Seq	home is 3_miles away at 56_cadwell_street

Round 2

User	okay , how far away am i and send me the quickest route
Truth	you are 3_miles away from home with heavy_traffic noted
DialoKG	there is heavy_traffic but you should be able to reach home in just few minutes you ' re welcome
PluDG	home is 3_miles away at 56_cadwell_street , there is heavy_traffic on the way but i sent on your screen the best route available
Fg2Seq	home is 3_miles away at 56_cadwell_street

Round 3

User	okay , how far away am i and send me the quickest route
Truth	you are welcome driver !
DialoKG	you ' re welcome !
PluDG	you ' re welcome !
Fg2Seq	you ' re welcome !

Figure 6 **PluDG and the two baselines generated responses using the In-Car assistant dataset.**

Full-size DOI: 10.7717/peerjcs.1707/fig-6

question, but with less romance compared to the ground truth. In contrast, DialoKG's response was slightly less satisfactory, while Fg2Seq provided a more comprehensive, yet mechanical, reply. In the second round, DialoKG barely met the expectations of the ground truth. Conversely, Fg2Seq mechanically responded to the first-round responses. On the other hand, PluDG offered nearly correct answers and generated smoother, more engaging responses. In the third round, it appears that all three models responded similarly.

Overall, the responses generated by PluDG are more contextually appropriate and comprehensible to humans. Combining these three cases, despite the remaining gap between the sentences generated by PluDG and the Reference Entities of real responses, the first two cases demonstrate PluDG's capability to generate semantically similar responses and provide more informative replies.

DISCUSSION

The research results here show that the plug-in we designed can provide effective prompt entities for the decoder. After adding modules such as the Kg-Plug, the model has been greatly improved on the three data sets. Therefore, in the future, we can design some plug-and-play lightweight plug-ins to assist large language models in different domain knowledge areas to generate results. Furthermore, we believe that, apart from enhancing the reply's accuracy, exploring ways to enhance its engagement and amusement could be a valuable area for future research, as evidenced by the results of this case study.

CONCLUSIONS

In this article, we introduce a novel task-oriented dialog system called PluDG, which utilizes a plug-and-play plug-in named Kg-Plug to assist GPT-2 in extracting knowledge base features. To enable GPT-2′s full exploration of the internal relationship of the selected knowledge base, we propose Unified Memory Integration, a method that enhances the comprehension of the sentence's internal structure and optimizes the knowledge base encoding location, thus improving the accuracy and fluency of the responses. Our experiments on three standard datasets demonstrate that our proposed model surpasses existing state-of-the-art models, particularly on datasets with complex knowledge base information. Additionally, we perform further ablation experiments to investigate the contribution of each module to the overall model. We aspire that our research findings will make a valuable contribution to the domain of task-oriented dialogue systems.

ADDITIONAL INFORMATION AND DECLARATIONS

Funding

The authors received no funding for this work.

Competing Interests

The authors declare there are no competing interests.

Author Contributions

- Xuelian Dong conceived and designed the experiments, performed the experiments, analyzed the data, performed the computation work, prepared figures and/or tables, authored or reviewed drafts of the article, and approved the final draft.
- Jiale Chen conceived and designed the experiments, performed the experiments, analyzed the data, performed the computation work, prepared figures and/or tables, authored or reviewed drafts of the article, and approved the final draft.

Data Availability

The following information was supplied regarding data availability:

The data and code are available in the Supplemental File.

The data used in this study are from Wen et al., 2016, EricManning, 2017, and Budzianowski et al., 2018. They are available at:

- https://paperswithcode.com/dataset/wizard-of-oz
- https://nlp.stanford.edu/blog/a-new-multi-turn-multi-domain-task-oriented-dialogue-dataset/
- https://paperswithcode.com/dataset/multiwoz

Supplemental Information

Supplemental information for this article can be found online at http://dx.doi.org/10.7717/peerj-cs.1707#supplemental-information.

REFERENCES

Banerjee S, Khapra MM. 2019. Graph convolutional network with sequential attention for goal-oriented dialogue systems. *Transactions of the Association for Computational Linguistics* **7**:485–500 DOI 10.1162/tacl_a_00284.

Budzianowski P, Wen T-H, Tseng B-H, Casanueva I, Ultes S, Ramadan O, Gašić M. 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*. Stroudsburg: Association for Computational Linguistics DOI 10.18653/v1/d18-1547.

Chen D, Fisch A, Weston J, Bordes A. 2017. Reading wikipedia to answer open-domain questions. In: *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: long papers)*. Stroudsburg: Association for Computational Linguistics DOI 10.18653/v1/p17-1171.

Eric M, Manning CD. 2017. Key-value retrieval networks for task-oriented dialogue. ArXiv arXiv:1705.05414.

He Z, He Y, Wu Q, Chen J. 2020. Fg2seq: effectively encoding knowledge for end-to-end task-oriented dialog. In: *Proceedings of the 2020 IEEE international conference on acoustics, speech and signal processing (ICASSP 2020)*. Piscataway: IEEE DOI 10.1109/icassp40776.2020.9053667.

Hu EJ, Shen Y, Wallis P, Allen-Zhu Z, Li Y, Wang S, Wang L, Chen W. 2021. LoRA: low-rank adaptation of large language models. ArXiv arXiv:2106.09685.

Huang G, Quan X, Wang Q. 2022. Autoregressive entity generation for end-to-end task-oriented dialog. ArXiv arXiv:2209.08708.

Kipf TN, Welling M. 2016. Semi-supervised classification with graph convolutional networks. ArXiv arXiv:1609.02907.

Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, Levy O, Lewis M, Zettlemoyer L, Stoyanov V. 2019. RoBERTa: a robustly optimized BERT pretraining approach. ArXiv arXiv:1907.11692.

Liu X, Shi T, Zhou G, Liu M, Yin Z, Yin L, Zheng W. 2023a. Emotion classification for short texts: an improved multi-label method. *Humanities and Social Sciences Communications* **10**:306 DOI 10.1057/s41599-023-01816-6.

Liu X, Wang S, Lu S, Yin Z, Li X, Yin L, Tian J, Zheng W. 2023b. Adapting feature selection algorithms for the classification of chinese texts. *Systems* **11(9)**:483 DOI 10.3390/systems11090483.

Lu S, Ding Y, Liu M, Yin Z, Yin L, Zheng W. 2023a. Multiscale feature extraction and fusion of image and text in VQA. *International Journal of Computational Intelligence Systems* **16**:54 DOI 10.1007/s44196-023-00233-6.

Lu S, Liu M, Yin L, Yin Z, Liu X, Zheng W. 2023b. The multi-modal fusion in visual question answering: a review of attention mechanisms. *PeerJ Computer Science* **9**:e1400 DOI 10.7717/peerj-cs.1400.

Madotto A, Cahyawijaya S, Winata GI, Xu Y, Liu Z, Lin Z, Fung P. 2020. Learning knowledge bases with parameters for task-oriented dialogue systems. In: *Findings of the association for computational linguistics: EMNLP 2020*. Stroudsburg: Association for Computational Linguistics DOI 10.18653/v1/2020.findings-emnlp.215.

Madotto A, Wu C-S, Fung P. 2018. Mem2Seq: effectively incorporating knowledge bases into end-to-end task-oriented dialog systems. In: *Proceedings of the 56th annual meeting of the association for computational linguistics (volume 1: long papers)*. Stroudsburg: Association for Computational Linguistics DOI 10.18653/v1/p18-1136.

Papineni K, Roukos S, Ward T, Zhu W-J. 2001. BLEU: a method for automatic evaluation of machine translation. In: *Proceedings of the 40th annual meeting on association for computational linguistics—ACL'02*. Stroudsburg: Association for Computational Linguistics DOI 10.3115/1073083.1073135.

Qin L, Xu X, Che W, Zhang Y, Liu T. 2020. Dynamic fusion network for multi-domain end-to-end task-oriented dialog. In: *Proceedings of the 58th annual meeting of the association for computational linguistics*. Stroudsburg: Association for Computational Linguistics DOI 10.18653/v1/2020.acl-main.565.

Raghu D, Jain A, Mausam , Joshi S. 2021. Constraint based knowledge base distillation in end-to-end task oriented dialogs. In: *Findings of the association for computational linguistics: ACL-IJCNLP 2021*. Stroudsburg: Association for Computational Linguistics DOI 10.18653/v1/2021.findings-acl.448.

Rony M, Usbeck R, Lehmann J. 2022. DialoKG: knowledge-structure aware task-oriented dialogue generation. In: *Findings of the association for computational linguistics: NAACL 2022*. Stroudsburg: Association for Computational Linguistics, 2557–2571 DOI 10.18653/v1/2022.findings-naacl.195.

Shen Y, Ding N, Zheng H-T, Li Y, Yang M. 2021. Modeling relation paths for knowledge graph completion. *IEEE Transactions on Knowledge and Data Engineering* **33(11)**:3607–3617 DOI 10.1109/tkde.2020.2970044.

Sukhbaatar S, Szlam A, Weston J, Fergus R. 2015. End-to-end memory networks. ArXiv arXiv:1503.08895.

Wen T-H, Vandyke D, Mrksic N, Gasic M, Rojas-Barahona LM, Su P-H, Ultes S, Young S. 2016. A network-based end-to-end trainable task-oriented dialogue system. ArXiv arXiv:1604.04562.

Wu J, Harris IG, Zhao H. 2022. GraphMemDialog: optimizing end-to-end task-oriented dialog systems using graph memory networks. *Proceedings of the AAAI Conference on Artificial Intelligence* **36(10)**:11504–11512 DOI 10.1609/aaai.v36i10.21403.

Wu C-S, Socher R, Xiong C. 2019. Global-to-local memory pointer networks for task-oriented dialogue. ArXiv arXiv:1901.04713.

Yang Y, Li Y, Quan X. 2021. UBAR: towards fully end-to-end task-oriented dialog system with GPT-2. *Proceedings of the AAAI Conference on Artificial Intelligence* **35(16)**:14230–14238 DOI 10.1609/aaai.v35i16.17674.

Zhao M, Wang L, Jiang Z, Li R, Lu X, Hu Z. 2023. Multi-task learning with graph attention networks for multi-domain task-oriented dialogue systems. *Knowledge-Based Systems* **259**:110069 DOI 10.1016/j.knosys.2022.110069.