IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 17 July 2024, accepted 24 July 2024, date of publication 29 July 2024, date of current version 7 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3434720

RESEARCH ARTICLE

CollRec: Pre-Trained Language Models and Knowledge Graphs Collaborate to Enhance Conversational Recommendation System

SHUANG LIU[1], ZHIZHUO AO[1], PENG CHEN[2], AND SIMON KOLMANIČ[3]
[1]School of Computer Science and Engineering, Dalian Minzu University, Dalian 116600, China
[2]School of Computer and Software, Dalian Neusoft University of Information, Dalian 116023, China
[3]Faculty of Electrical Engineering and Computer Science, University of Maribor, 2000 Maribor, Slovenia

Corresponding author: Shuang Liu (liushuang@dlnu.edu.cn)

This work was supported by the 2023 Humanities and Social Sciences Research and Planning Fund of the Ministry of Education with under Grant 23YJA860010.

ABSTRACT Existing conversational recommender systems (CRS) use insufficient generality in incorporating external information using knowledge graphs. The recommendation module and generation module are loosely connected during model training and shallowly integrated during inference. A simple switching or copying mechanism is used to merge recommended items into generated responses. These problems significantly degrade the recommendation performance. To alleviate this problem, we propose a novel unified framework for collaboratively enhancing conversational recommendations using pre-trained language models and knowledge graphs (CollRec). We use a fine-tuned pre-trained language model to efficiently extract knowledge graphs from conversational text descriptions, perform entity-based recommendations based on the generated graph nodes and edges, and fine-tune a large-scale pre-trained language model to generate fluent and diverse responses. Experimental results on the WebNLG 2020 Challenge dataset, ReDial dataset, and Reddit-Movie dataset show that our CollRec model significantly outperforms the state-of-the-art methods.

INDEX TERMS Conversational recommendation system, knowledge graph, large language model, end-to-end generation, fine-tuning, ReDial, WebNLG 2020 challenge.

I. INTRODUCTION

With the increasing popularity of smart assistants in users' daily lives, research interest in conversational recommendation systems (CRS) has grown rapidly. The concept of conversational recommender system originated from the earliest reviews of conversational recommender systems published by Li et al. [1], Sun and Zhang [2], and Zhou et al. [3] in 2018. CRS can be used to make precise recommendations based on users' previous implicit feedback (such as click or purchase history), while traditional recommendation systems make personalized recommendations based on users' implicit feedback. Although existing research provides user-specific recommendations through dialogue, CRS remains challenging.

There are two reasons for this: (1) Typical conversations are short and lack sufficient project information to capture user preferences. These include the methods proposed by Chen et al. [4] and Zhou et al. [3]. (2) Difficulty in generating informative responses with project-related descriptions. These include the methods proposed by Shao et al. [5], Ghazvininejad et al. [6] and Wang et al. [7]. Therefore,

The associate editor coordinating the review of this manuscript and approving it for publication was Sangsoon Lim.

FIGURE 1. The dialogue between a user and the system regarding movie recommendations is illustrative. Red indicates comments that match other sentiments. A bolded item (movie) or entity (such as an actor) is shown.

by leveraging rich entity information contained in structured knowledge graphs (KG), external information has been introduced recently to enhance item representation. These include the methods proposed by Chen et al. [4] and Zhou et al. [3]. Although KG-based methods improve the performance of CRS to a certain extent, they still suffer from the following limitations: (1) poor versatility due to the high cost of KG construction. (2) insufficient integration of knowledge and response generation, including the method of Lin et al. [8]. Furthermore, most existing CRS datasets are relatively small due to expensive crowdsourcing costs, including datasets constructed by Li et al. [1], Zhou et al. [9], and Liu et al. [10]. End-to-end neural network models trained on these datasets often overfit and generate unrealistic responses in conversations.

Therefore, in order to better connect external knowledge with recommendations in conversations and improve the generality of the knowledge graph used in training, in this paper, a novel framework based on pre-trained language models (PLM) is proposed, that is, a unified framework (CollRec) with pre-trained language models and knowledge graphs to collaboratively enhance conversational recommendation systems. Specifically, CollRec first builds a more versatile knowledge end-to-end by using PLM (T5) [11] Graphs, including the generation of node and edge relationships. Given an input text, the generation of the knowledge graph is divided into two steps. In the first step, we leverage the representation power of a pre-trained language model (T5) and fine-tune it on the task of entity (graph nodes) extraction, while in the second stage, node relations (graph edges) are generated using available entity information. The entire knowledge graph generation is end-to-end trainable. By combining the generation of nodes and edges, efficient information transfer between the two modules is achieved, avoiding the involvement of any external NLP pipeline. Next, in the conversation recommendation part, CollRec integrates item recommendation into dialogue generation in pretrain-finetune mode. To represent project-oriented knowledge graphs as nodes in RGCNs, powerful PLMs are used, such as DialoGPT [12]. Through the former, PLM generates fluent and diverse dialogue responses in light of its powerful language generation capabilities, while the latter facilitates item recommendation through more accurate structural node representations.

Using Figure 1, we illustrate the motivation for our work, since the CRS system may not have a deep understanding of the items mentioned by the user, so the system will respond uninformatively with "It's Great" since it lacks the necessary knowledge to make recommendations. Chatting doesn't help with recommendations either.

Furthermore, our proposal is to evaluate the performance of recommendations by checking whether the final response contains the target items and checking the construction effect of the knowledge graphs. Separate evaluations of knowledge graph generation and dialogue recommendation are conducted in our work. We conduct extensive experiments on text-to-RDF generation on the WebNLG+2020 challenge [13] and the benchmark ReDial [1] and Reddit-Movie [55]. Our CollRec model performs well in both recommendation accuracy and session quality. Further ablation studies also demonstrated the superior performance of our method.

In this work, the following contributions can be made:

- CollRec is proposed, a conversational push framework based on PLM. CRS challenges can be solved through the fine-tuning of large-scale PLM using graph convolutional networks by CollRec.
- By constructing the knowledge graph in an end-to-end manner, we use the representation ability of the pre-trained language model to fine-tune the entity (graph node) extraction task to obtain nodes and node features, and then use the available entity information relationships (graph edges) to generate The edge of the graph.
- The results of extensive experiments demonstrate that CollRec can significantly outperform state-of-the-art methods when it comes to assessing recommendation accuracy and quality of sessions.

II. RELATED WORK

Recently, the NLP community's language model based on Transformer has achieved success. Among them, Vaswani et al. [14] completely abandoned the traditional RNN and convolutional network and used the self-attention mechanism to build the model structure. Devlin et al. proposed the pre-trained language model BERT [15]. Raffel et al. [11] used a unified text-to-text converter based on Transformer to explore the limits of transfer learning. These models has been pre-trained on a large text corpus, spawning a series of downstream tasks based on KG, and CRS is one of them. Currently, RCS work can be classified into two categories: attribute-based RCS and open RCS.

In terms of knowledge graph construction, some of these methods have studied a relatively simple graph completion problem. Li et al. [16] selected sentences with the highest log-likelihood based on a pre-trained language model to complete the graph. Yao et al. [17] completed the knowledge graph by fine-tuning BERT. Malaviya et al. [18]

completed the graph by learning from the local graph structure and combining transfer learning from the pre-trained language model to the knowledge graph. All of the above methods are given partial triples, one of the entities is usually missing, and the goal is to generate the missing entity relationship or triplets are formed by ranking entities. Due to their limitations, these methods are not suitable for building the entire global graph structure, but are only suitable to extend the existing knowledge graph. Furthermore, other works include Petroni et al. [19] who discussed the potential of pre-trained language models in unsupervised open-domain knowledge graph question answering systems. Roberts et al. [20] measured the knowledge information stored in pre-trained language models through question answering tasks without context/external knowledge. Jiang et al. [21] proposed mining and paraphrase-based methods to automatically generate high-quality and diverse prompts, as well as a collection method that combines answers from different prompts. Shin et al. [22] proposed an automated method called AUTOPROMPT to create appropriate prompt templates for a variety of tasks without additional parameters or fine-tuning. Li and Liang [23] proposed prefix-tuning, a lightweight alternative method for natural language generation tasks that keeps the language model parameters frozen but optimizes a small (continuous) task-specific vector (called prefix). The extraction of learned facts and common sense knowledge from pre-trained models has been proposed in these works. As these methods are unable to perceive global graph structures like in the past, they are usually only useful when patching local graphs.

CRSs with attribute-based recommendations can be viewed as task-oriented dialogue systems [24] driven by questions. Using such systems, the user preferences are inferred from querying items about their attributes and the best candidates for recommendation are then identified. Using such systems, the user preferences are inferred from querying items about their attributes and the best candidates for recommendation are then identified. Existing works have studied various query strategies, such as reinforcement learning methods based on adversarial learning. Chen et al. [28] discussed 20 ''questions'' to show whether it is possible to reduce the labor cost (making the manual construction process less boring) while ensuring the ''quality'' of the constructed knowledge base. Lei et al. [29] proposed a new CRS framework called ''Estimate-Action-Reflection'' or EAR to fill the gap in the missing interaction framework (what questions to ask about item attributes, when to recommend items, and how to adapt to users' online feedback). Deng et al. [30] developed an RL method based on dynamic weighted graphs to learn a strategy for selecting actions in each dialogue round, whether it is asking about attributes or recommending items. Methods based on generalized binary search, Zou and Kanoulas [26] proposed a novel interactive method to effectively locate the best matching products between users and retailers in e-commerce. Zou et al. [27] also proposed a novel question-based recommendation method Qrec, which helps users find items interactively by answering questions that are automatically constructed and algorithmically selected. In addition, Ren et al. [31] proposed a conversational recommendation system based on adversarial learning (CRSAL), which designed a completely statistical conversation state tracker and combined a neural policy agent to accurately capture the intention of each user from limited conversation data and generate conversational recommendation actions. Wu et al. [25] proposed an entropy-based ranking method to calculate the recommendation algorithm of the target sequence. Under this premise, graph-based methods have emerged, including the CRSAL model of Ren et al. [31], conversational recommendation through user memory reasoning of Xu et al. [32], and conversational recommendation using interactive path reasoning on the graph of Lei et al. [33]. As well as some methods that use bandit online recommendation to solve cold start scenarios, including the contextual gambling method for personalized news article recommendation by Li et al. [34], a review of conversational recommendation systems by Christakopoulou et al. [35], and conversational recommendation for cold start users by Li et al. [36]. Open CRS is explored through more free-form conversations, including the following existing works, among which Li et al. [1] produced the ReDial dataset containing more than 10,000 conversations on the topic of providing movie recommendations, and used this dataset to explore multiple aspects of conversational recommendation. In particular, new neural architectures, mechanisms, and methods suitable for forming conversational recommendation systems were explored. Liu et al. [10] proposed a new task of conversational recommendation based on multi-type conversations, in which the robot can actively and naturally guide the conversation from non-recommendation conversations (such as QA) to recommendation conversations while taking into account user interests and feedback. Jiang et al. [37] proposed a frequency-aware cross entropy (FACE) loss function that improves the CE loss function by incorporating a weighting mechanism conditioned on the frequency of tokens. Hayati et al. [38] proposed a new dataset INSPIRED, which contains 1,001 movie recommendation conversations between people, and measured the degree of successful recommendations. Ma et al. [39] discussed bridging the gap between conversational reasoning and interactive recommendation. Wang et al. [40] proposed a pioneering conversational recommendation model by jointly modeling user preferences for recommendations with entity and context representations captured by a pre-trained language model, where a time-aware attention mechanism is designed to emphasize the most recently appeared items in the entity-level representation. CRS has released several datasets in this direction to help promote research in this area. Such as ReDial [1], TG-REDIAL (Chinese) [3], INSPIRED [38], DuRecDial [10] and Reddit-Movie [55]. In subsequent research, KBRD [4] utilized various external knowledge to improve the performance of open CRS. According to CR-Walker [39], an approach to introduce related items would be

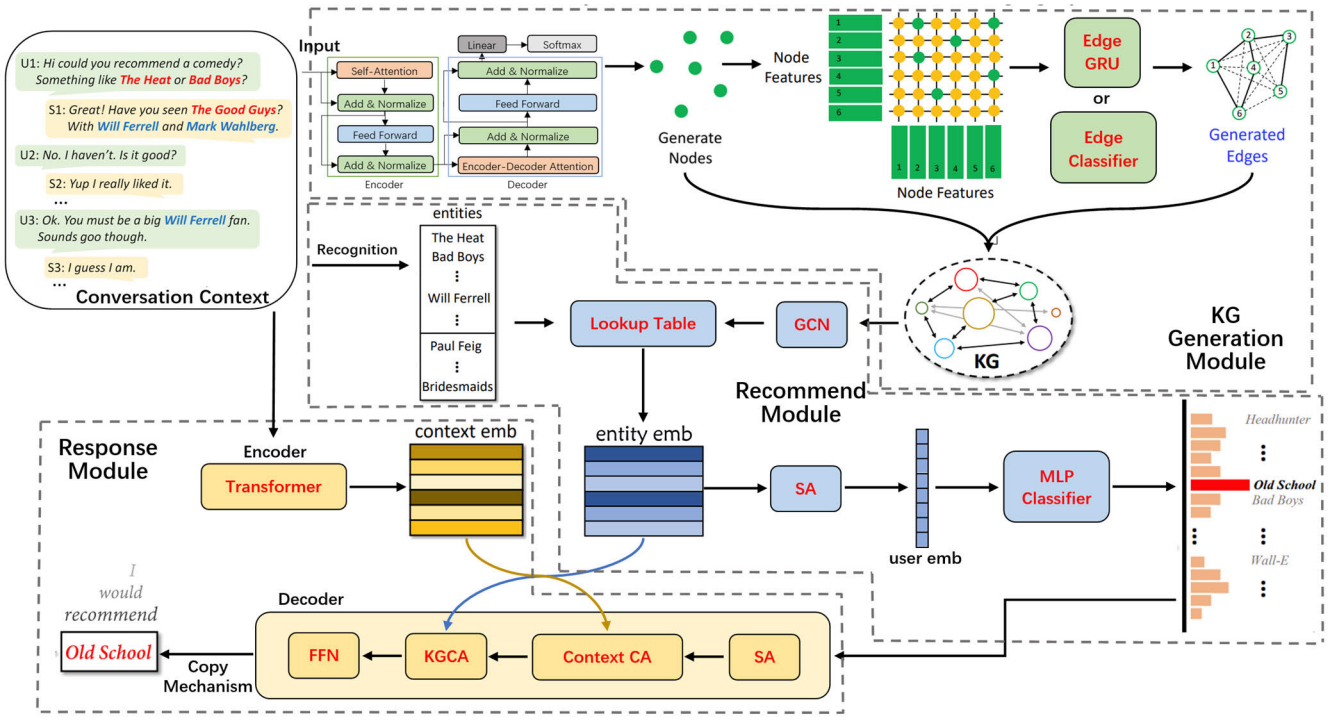


FIGURE 2. Model overview of CollRec.

tree-structured reasoning, while in MGCG [10], a transition strategy from non-recommended dialogues to recommended dialogues is discussed. Our work lies in the research of open CRS.

III. METHODOLOGY

In this section, we introduce the details of the proposed CollRec method, first describing the generation of knowledge graphs that can be used by subsequent methods in an end-to-end fashion through pre-trained language models. Secondly, a unified framework for conversation recommendation using pre-trained language models and knowledge graphs is described. The overall framework of the CollRec method is shown in Figure 2.

A. USE PRE-TRAINED LANGUAGE MODELS TO GENERATE KNOWLEDGE GRAPHS

Using the pretrained encoder-decoder language model T5, where the system is fine-tuned to convert text, our method generates nodes from text input using a sequence-to-sequence problem to generate a unique set of nodes that constitute the graph. As a result of the conversion, nodes are formed, separated by special tokens, *i.e.*, <PAD> NODE_1 <NODE_SEP> NODE_2 <NODE_SEP> NODE_3 </S>, where each word is represented by the special token NODE_i.

1) NODE GENERATION

We first learn node queries to obtain node features, and then estimate the arrangement to align with the target node order. As shown in Figure 3.

An embedding matrix is created from the set of learnable node queries that are given as input to the decoder. We also disabled causal masking to ensure that Transformer can handle all node queries simultaneously. While traditional encoder-decoder architectures typically use causal masks as input embeddings of target sequences during training, or embed self-generated sequences during inference, our work can be distinguished from traditional encoder-decoder architectures used to form Compared. Decoder output can now be read directly as Nd-dimensional node features $F_n \in \mathbb{R}^{d\times N}$ and passed to the prediction head (LSTM or GRU) to be decoded into node logits $L_n \in \mathbb{R}^{S\times V\times N}$, where S is generated The node sequence length, V is the vocabulary size.

A permutation-invariant system avoids remembering a specific order of target nodes and is configured as:

$$L'_n(s) = L_n(s)P, F'_n = F_nP \tag{1}$$

For $s = 1, \ldots, S$, where $P \in \mathbb{R}^{N\times N}$ is the permutation matrix obtained using the bisection matching algorithm between the target node and the greedy decoding node. For the bipartite matching algorithm to obtain the permutation matrix, we refer to the bipartite graph matching algorithm in the DETR (Detection Transformer) [56] method in the field of target detection. We have made some improvements to the algorithm, specifically matching the bounding box and the groundtruth. The model will output a fixed number of prediction boxes. If the number of predicted values is insufficient, it will be padded with ∅. The groundtruth of the label will be consistent with the number of output values, and the insufficient number will also be padded with ∅. Using the

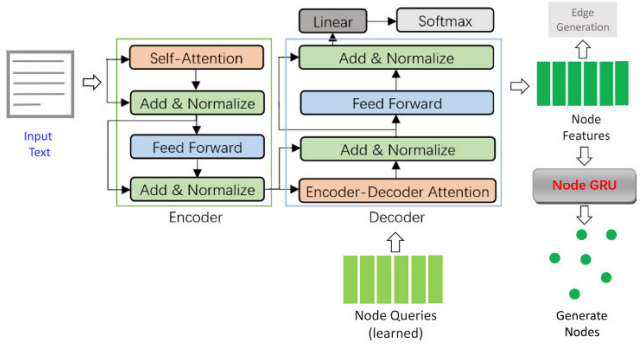


FIGURE 3. Nodes are generated using the learned query vectors. The node features are formed by transforming the input text and query vectors into embedding matrices. By using node generation heads such as GRU or LSTM, they are decoded into graph nodes. Edge building blocks receive the same features.

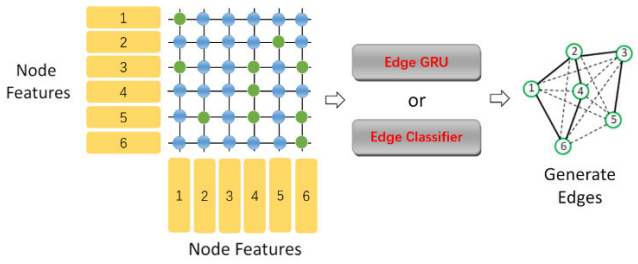


FIGURE 4. Edge construction, using generator (Edge GRU) or classifier heads. Circles in green indicate graph edges (solid lines), while circles in blue indicate <NO_EDGE> features (dashed lines).

set loss (calculated using the Hungarian algorithm), even if the same target is predicted and output multiple times, there can only be one groundtruth corresponding to it, similar to NMS. Finally, the model will learn to give the same number of unclassified predictions $\emptyset$ on both sides through training, because if an output that should not be obtained is given, it will be punished. Repeat this process to obtain the permutation matrix of the target node and the greedy decoding node. Next we use the cross entropy loss as the matching cost function. With the new alignment feature F'_n, the edge generation phase can use aligned nodes to generate edges.

2) EDGE GENERATION

We use the node feature set generated in the previous step for edge generation in this module. This step is schematically illustrated in Figure 4. In order to determine whether an edge exists between two nodes, the prediction head takes into account the pair of node features. Edges can be generated as a sequence of tokens using a header (LSTM or GRU) by constructing a sequence of edges, including those not seen during training.

Since knowledge graphs are usually represented as directed graphs, it is critical to ensure the correct order (subject-object) between two nodes. For this purpose, a simple difference between eigenvectors is suggested using: $F'_n(:,i) - F'_n(:,j)$, for the case where node i is the parent of node j. This approach allows the model to learn that $F'_n(:,i) - F'_n(:,j)$ means $i \to j$, and $F'_n(:,j) - F'_n(:,i)$ means $j \to i$.

We have to generate or predict at least N^2 edges, where is the number of nodes, in order to check if edges exist between all pairs of nodes. Ignoring self-edges and ignoring edges when some of the generated nodes have the <NO_NOD E> flag can result in some small savings. When no edge exists between two nodes, we indicate it with the special tag <NO_EDGE>. For <NO_EDGE> tags/classes, generation and classification are unbalanced because actual edges are usually small while <NO_EDGE> is large. We solve this problem by modifying the cross-entropy loss.

Here we use Our loss instead of the traditional Cross Entropy (CE) loss [41], the main idea is to reduce the CE loss weight of well-classified samples (<NO_EDGE> in our example), and increase the CE loss of misclassified samples, as follows, the probability distribution p corresponds to a single edge and t is the target category:

$$\mathrm{CE}(p,t) = -\log(p_t) \tag{2}$$

$$\mathrm{FL}(p,t,\gamma) = -(1-p_t)^{\gamma}\log(p_t) \tag{3}$$

where $\gamma \geq 0$ is a weighting factor such that when $\gamma = 0$ the two losses are equal. On the classification head, this loss can easily be applied; on the generation head, it must be modified by arithmetic to obtain the equivalent of p_t over the edge sequence length.

B. KNOWLEDGE GRAPH ENHANCED RECOMMENDATION

The KG-based framework is used to construct the recommender component, with all entities in the context being extracted to create a user profile embedding. To enhance the accuracy of our recommendations, we use the retrieved contextual keywords in our method to enrich entity information.

First, use GCN to extract the embeddings of all items in the knowledge graph from the knowledge graph generated by the pre-trained language model in Section A, and obtain the embedding set ε. Then according to the context context C, extract all the entity items that appear in the conversation, then find the embedding of these entity items in ε, and then splice them into a matrix $\mathbf{E}^{(C)} \in \mathbb{R}^{l^{(C)} \times d}$. The number of rows of the matrix is the number of entities $l^{(C)}$ and the number of columns is The dimension d of embedding. Through the self-attention layer, the embedding matrix $E^{(C)}$ is aggregated into the user's preference representation $U^{(C)}$, the formula is as follows:

$$\mathbf{u}^{(C)} = \mathbf{E}^{(C)} \cdot \boldsymbol{\alpha} \tag{4}$$

$$\boldsymbol{\alpha} = \mathrm{softmax}(\mathbf{b}^{\top} \cdot \tanh(\mathbf{W}_{\alpha}\mathbf{E}^{(C)})) \tag{5}$$

$\boldsymbol{\alpha}$ in the formula represents the self-attention mechanism score vector of each entity, and $\boldsymbol{W_a}$ and $\boldsymbol{b}$ are the parameter matrix and vector for linear projection and bias. Input the user preference representation $\boldsymbol{U^{(C)}}$ into the MLP, and then undergo softmax normalization to obtain the probability set $\boldsymbol{p}$ of the candidate entities. Finally, the entity with the highest

probability in $\boldsymbol{p}$ is selected as the target to be recommended:

$$\mathrm{p} = \mathrm{softmax}(\mathrm{MLP}(\mathrm{u}^{(\mathrm{C})})) \tag{6}$$

Finally, the cross-entropy loss $\mathcal{L}_{rec}$ between the probability p of the recommended target and the target category p^* is calculated:

$$L_{rec} = -\frac{1}{M}\sum_{i=1}^{M} log\ p_i^* \tag{7}$$

where M is the number of recommendations and p_i^* is the prediction probability of the target category in the i_{th} recommendation.

While constructing the dataset, the annotators of some movies may have little or no familiarity with the dialogue history as a result of sparse entities in the dialogue history. By adding more entity words to $E^{(C)}$ from the retrieved reviews, $E^{(C)}$ can be further enhanced. The process of obtaining comment-rich entities can be expressed as:

$$E_C^{(C)} = \mathrm{Extract}\,(C) \tag{8}$$

$$E^{(C)} = \{E_C^{(C)}\} \tag{9}$$

where $extract(\cdot)$ defines the entity extraction operation, and $E_c^{(C)}$ represents the entity extracted from the context. Based on rich entities, user embeddings are expected to be better represented to produce more accurate recommendation results.

C. GENERATING RESPONSES BASED ON PLM

Assuming that an input (***i.e.*** conversation history context $\{\boldsymbol{t_1}, \boldsymbol{t_2}, \ldots, \boldsymbol{t_m}\}$) is accompanied by a set of historical utterances, we can concatenate historical utterances into a variety of contexts $\boldsymbol{C} = \{\boldsymbol{w_1}, \boldsymbol{w_2}, \ldots, \boldsymbol{w_n}\}$, where $\boldsymbol{n}$ is the total number of contexts mark. Therefore, the probability of generating a response $\boldsymbol{R} = \{\boldsymbol{w_{n+1}}, \boldsymbol{w_{n+2}}, \ldots, \boldsymbol{w_{n+k}}\}$ is:

$$\boldsymbol{PLM}(\boldsymbol{R} \mid \boldsymbol{C}) = \prod_{i=n+1}^{n+k} \boldsymbol{p}(\boldsymbol{w_i} \mid \boldsymbol{w_1}, \ldots, \boldsymbol{w_{i-1}}) \tag{10}$$

where $\boldsymbol{PLM}(\cdot \mid \cdot)$ represents the PLM of the Transformer [14] architecture. $\boldsymbol{N}$ such context-response pairs can be constructed for multiturn conversations, where $\boldsymbol{N}$ is the quantity of utterances from the recommender. Using all possible $(\boldsymbol{C}, \boldsymbol{R})$ pairs constructed from the conversation corpus, we then fine-tune the PLMs. As a result, our model not only inherits the powerful language generation capabilities of PLMs, but is also able to learn how to generate recommended utterances from an extremely small CRS dataset.

D. LEARNING OBJECTIVES

In this section, two goals are being pursued: learning node representations on the knowledge graph, and fine-tuning the model for generating responses. By optimizing the cross-entropy of item predictions, we optimize the R-GCN and self-attention network for the former:

$$\mathcal{L}_{kg} = \sum_{(u,i)\in D_1} -\log\left(\frac{\exp\left(\boldsymbol{t}_u \boldsymbol{H}^T\right)_i}{\sum_j \exp(\boldsymbol{t}_u \boldsymbol{H}^T)_j}\right) \tag{11}$$

TABLE 1. Dataset WebNLG (Text-to-RDF).

	Train	Dev	Test
RDF triple sets	13211	1667	752
Texts	35426	4464	2155

TABLE 2. Dataset statistics for the dataset ReDial. A number is symbolized by "#" and an average by "avg".

Conversations		Movies	
# of convs	10006	# of mentions	51699
# of utterances	182150	# of movies	6924
# of users	956	avg mentions	7.5
avg token length	6.8	max mentions	1024
avg turn #	18.2	min mentions	1

where item i is a real item, u is the corresponding user history, and D_1 contains all training instances and $\boldsymbol{t}_u \boldsymbol{H}^T \in \mathbb{R}^{|\varepsilon|}$.

A further cross-entropy loss, referred to as R, is optimized for all generated responses. The following formula summarizes the process:

$$\mathcal{L}_{gen} = \sum_{(C,R)\in\mathcal{D}_2} \sum_{w_i \in R} -\log\left(p\left(w_i \mid w_{<i}, C\right)\right) \tag{12}$$

where $p(w_i)$ refers to the C in Section C and D_2 contain all (C, R) pairs constructed from the dataset. With the joint action of two objectives $\mathcal{L}_{kg} + \mathcal{L}_{gen}$, we train the entire model end-to-end.

IV. EXPERIMENT

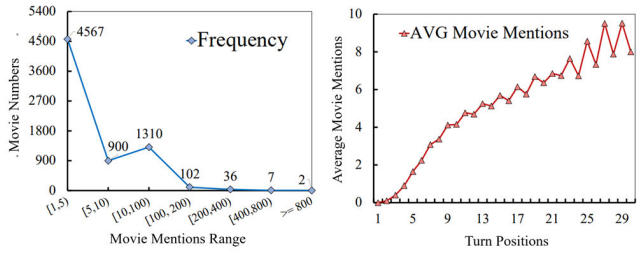
In order to evaluate the effect of CollRec, we set up two experiments to test the effect of the model. First, we conducted an experimental evaluation on the knowledge graph generated by the pre-trained language model. Next, we conducted an experimental evaluation on the conversational recommendation system based on the collaboration of the pre-trained language model and the knowledge graph.

A. DATASET

1) KNOWLEDGE GRAPH GENERATION

In this part we first use the small-scale WebNLG+ 2020 [13] dataset. In the 2020 WebNLG Challenge, WebNLG+ corpus v3.0 is part of a set of two tasks: one generates text based on a set of RDF triples (subject-predicate-object), and another analyzes semantics. Used to convert a text description into a set of RDF triples. In our work, we evaluate the algorithm on the text-to-RDF task, whose statistics are shown in Table 1. Each triple is associated with one or more words, so when the triples are assigned to all words, the size of the total training, validation, and test set splits is shown in the second row of Table 1. The data consists of 16 DBpedia categories: 11 categories are used as training and validation sets, and 5 unseen categories are used as test sets.

In order to reduce noise in the data, underscores and surrounding quotes were removed as part of the preprocessing. Due to a vocabulary coverage mismatch between T5's tokenizer and WebNLG's dataset, some WebNLG characters are ignored during tokenization because they don't appear



(a) Movie # Distribution (b) Position Distribution

FIGURE 5. For Figure 5(a), X-axis: the number of times the movie is mentioned; Y-axis: movie number. For Figure 5(b), X-axis: position of dialogue turns; Y-axis: average number of movie mentions.

TABLE 3. Dataset statistics for the dataset Reddit-Movie. A number is symbolized by "#".

	Total	Train + Validation	Test
#Conv.	634,392	570,955	63,437
#Turns	1,669,720	1,514,537	155,183
#Users	36,247	32,676	4,559
#Items	51,203	48,838	20,275

in T5. For this, we formatted the data to map those missing characters to the closest available characters.

2) CONVERSATION RECOMMENDATIONS

In this section, we evaluate our CollRec model on the benchmark dataset ReDial [1], a dataset built around movies. Since real-world data in this aspect is quite scarce, most previous works [1,4,3] also conducted experiments on this dataset. A summary of the ReDial dataset statistics can be found in Table 2, and a detailed chart of movie occurrences can be found in Figure 5(a). It is obvious that this dataset has a problem of data imbalance because most movies appear less than five times in the dataset. According to figure 5(b), the number of dialogues found in a text is related to the number of occurrences in a movie. We can see that when the number of dialogue turns is less than 5, the number of movies presented is less than 2. In addition, we evaluate our CollRec model again on the Reddit-Movie [54], [55] dataset, which is still centered around movies and their related conversation data. Compared with ReDial, the conversations in the Reddit-Movie dataset tend to contain more complex and detailed user preferences because they are derived from real conversations on Reddit, enriching the conversation recommendation dataset with diverse discussions. The amount of data in the Reddit-Movie dataset is shown in Table 3. Before training, we followed the method of [1] to randomly divide the dataset into a ratio of 8:1:1 and use it as a training set, a validation set, and a test set.

B. PARAMETER SETTING

1) KNOWLEDGE GRAPH GENERATION

Our pretrained language model uses the T5 "large" (770M parameters in total) from HuggingFace, Inc [42]. For query node generation, we also define an embedding matrix $\boldsymbol{M} \in \mathbb{R}^{\boldsymbol{H}\times\boldsymbol{N}}$ for learnable queries, with a hidden size of 1024 for the T5 model and an 8-node graph is the maximum number possible. A single-layer GRU decoder with $\boldsymbol{H_{GRU}} = 1024$ is used for node generation. This is followed by a linear transformation projection to a vocabulary of size 32, 128 in the node generation head. The edge generation head uses the same GRU settings, setting the maximum number of edges to 7. Using a dropout probability of 0.5, we define four fully connected layers with ReLU nonlinearity for edge class classification, which is projected into a 407-class edge space.

We trained all parameters of the fine-tuned model using the AdamW optimizer with the learning rate set to 10^{-4}. $\boldsymbol{\beta} = [0.9, 0.999]$ and decrement to 10^{-2}. A single NVIDIA A100 GPU was used for WebNLG training with a batch size of 10 samples.

2) CONVERSATION RECOMMENDATIONS

We improved the 12 transformer layer DialoGPT model2, which is a small-sized pretrained model. The embedding dimension is 768. Using R-GCN, we have set the layer count to 1 and both the entity embedding size and hidden representation size to 128 for the knowledge graph. For the GPT-2 baseline, we fine-tune small model4. For the BART baseline, we fine-tune each encoder and decoder of base model 3 to 6 layers with a hidden size of 1024. For the training of all models, we adopt the Adam optimizer and the learning rate is selected from $\{\boldsymbol{1e-5}, \boldsymbol{1e-4}\}$. The gradient accumulation step is set to 8, the batch size is chosen from {32, 64}, and the preheating step is chosen from {500, 800,1000}.

C. BASELINES

1) KNOWLEDGE GRAPH GENERATION

In order to evaluate the performance of the knowledge graph generation module in CollRec, we use the best-performing team on the WebNLG 2020 challenge leaderboard as the baseline, including the following contestants: (1) **Amazon AI** (Shanghai) [43] is the Text-to-RDF task Challenge winner. In this procedure, the entities present in the text are entity linked with the DBpedia ontology, and then the relationships between these entities are extracted from the DBpedia database. (2) **CycleGT** [44], an off-the-shelf entity extractor is used to identify all entities present in the input text in the knowledge base building part of CycleGT 's unsupervised text-to-graph and graph-to-text generation methods, while a multi-label classifier is used to predict how entities are related. (3) **BT5** [45], concatenate the object-predicate-subject triples using a large pre-trained T5 model, and transform the entire text-to-graph problem into a sequence-to-sequence problem. (4) **Stanford CoreNLP Open IE** [46], the method extracts subjects, relations, and objects from the input text portion of the test set unsupervised. (5) **ReGen** [47], a linearized graph representation approach is used to generate bidirectional text-to-graphs and graph-to-texts using a T5 pre-trained language model.

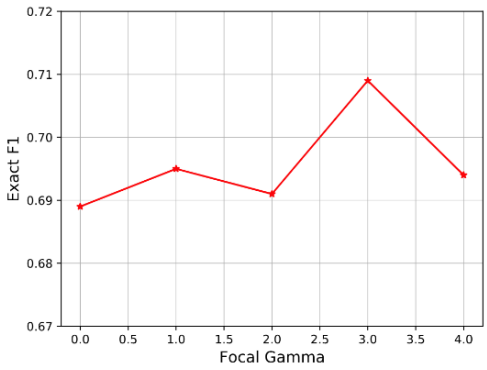


FIGURE 6. The impact of the focal parameter γ on the performance of CollRec (KG), measured by the F1 score of the exact match.

2) Conversation recommendations

In this section, we introduce two baselines of dialogue and recommendation modules. (1) **Transformer** [14]. Using a transformer-based encoder-decoder, the system generates responses without the need for a recommendation module. (2) **Popularity**. By using the historical frequency of movie items in the training set instead of dialogue modules, it ranks movie items.

Experimental comparisons are then made between these baseline models: (3) **ReDial**. Among its components are a dialogue generation module [48] that uses HRED, a recommendation module [49] that uses autoencoders, and a sentiment analysis module [50]. (4) **GPT-2**. By integrating GPT-2's vocabulary with the project's vocabulary, we refine GPT-2 directly. (5) **DialoGPT**. As part of this fine-tuning, DialoGPT's vocabulary is extended to include the same items as DialoGPT's. (6) **BART**. To incorporate the same project vocabulary into BART, we directly refine it and expand its vocabulary. (7) **KGSF** [3]. To learn better semantic representations of user preferences, a word-level and entity-level knowledge graph is merged and fused. (8) **KBRD** [4]. Modeling relationship knowledge between context items or entities is done through DBpedia's knowledge graph, and dialogue generation is based on Transformer.

D. EVALUATION METRICS

1) KNOWLEDGE GRAPH GENERATION

We used the built-in evaluation script in WebNLG 2020 Challenge [13] to evaluate the generated knowledge graph based on real-world scenarios, which calculates precision, recall and F1 scores. However, since the order of the real triples and the generated triples should not have an impact on the results, the script searches for the best between the reference triples and each candidate by ranking all possible (hypothesis-reference pairs) Best alignment. We then evaluate precision, recall, and F1-score using metrics based on named entity evaluation [50] in four different ways. Exact: It does not matter the type of the triple (subject, predicate, object), the candidate triple should exactly match the reference triple.Partial: In whatever type (subject, predicate, or object), the candidate triple must at least partially match the reference triple. Strict: There should be exact matches between the candidate triple and the reference triple, as well as between the element types (subject, predicate, object).

TABLE 4. Dataset evaluation results for WebNLG + 2020. The bottom is the experimental results of the knowledge graph generation module of our proposed CollRec. Bold black and blue show the best and second best performance respectively.

	Match	F1	Precision	Recall
Amazon AI	Exact	0.689	0.689	0.690
	Partial	0.696	0.696	0.698
	Strict	0.686	0.686	0.687
BT5	Exact	0.682	0.670	0.701
	Partial	0.712	0.700	0.736
	Strict	0.675	0.663	0.695
CycleGT	Exact	0.342	0.338	0.349
	Partial	0.360	0.355	0.372
	Strict	0.309	0.306	0.315
Open IE	Exact	0.158	0.154	0.164
	Partial	0.200	0.194	0.211
	Strict	0.127	0.125	0.130
ReGen	Exact	**0.723**	**0.714**	**0.738**
	Partial	**0.767**	**0.755**	**0.788**
	Strict	**0.720**	**0.713**	**0.735**
CollRec（KG）(Ours)	Exact	**0.709**	**0.702**	**0.720**
	Partial	**0.735**	**0.725**	**0.750**
	Strict	**0.706**	**0.700**	**0.717**

2) CONVERSATION RECOMMENDATIONS

The recommendation module and the dialogue module are evaluated separately according to what we have described so far. Following the previous setup [4,3], we evaluate the recommendation module via Recall@k (k = 1, 10, 50). The final generated response is also evaluated end-to-end, ***i.e.***, in order to determine whether the target item is present in the final generated response.

In such a setting, Recall@K is calculated based on both whether the recommended item is successfully injected into a generated sentence and whether the real item is listed in the top K recommendations. Consequently, K = 1, 10, 50 is a reasonable number of models for end-to-end evaluation. Metrics automatically generated by the dialogue module include: (1) **Fluency**: Depending on how confident you are in the generated response, you can measure the Perplexity Level (PPL). (2) **Correlation**: BLEU-2/4 [51] and Rouge-L [52]. (3) **Diversity**: This number of different n-grams is known as distinct-n, which is denoted as the number of different words divided by the number of distinct-n (Dist-n) [53]. To evaluate the diversity of responses, Dist-2/3/4 is used at the sentence level. In addition, the response ratio is also measured using the item ratio as per KGSF [3].

E. EXPERIMENTAL RESULTS

1) KNOWLEDGE GRAPH GENERATION

Using the WebNLG test set, Table 4 summarizes the main results of all compared methods. And we visualize it in the

TABLE 5. Main comparison results on recommendations. R@k refers to Recall@k. CollRec significantly outperforms the baseline.

Models	ReDial						Reddit-Movie					
	Eval on Rec Module			End-to-End Eval			Eval on Rec Module			End-to-End Eval		
	R@1	R@10	R@50	R@1	R@10	R@50	R@1	R@10	R@50	R@1	R@10	R@50
Popularity	1.2	6.1	17.9	1.2	6.1	17.9	1.1	5.8	16.4	1.1	5.8	16.4
ReDial	2.4	14.0	32.0	0.7	4.4	10.0	2.2	13.5	31.3	0.6	3.9	9.0
KBRD	3.1	15.0	33.6	0.8	3.8	8.8	3.0	14.8	32.8	0.8	3.6	8.5
KGSF	3.9	18.3	37.8	0.9	4.2	8.8	3.7	17.6	35.4	0.8	3.9	8.6
GPT-2	-	-	-	1.4	6.5	14.4	-	-	-	1.2	5.9	13.6
BART	-	-	-	1.5	-	-	-	-	-	1.4	-	-
DialoGPT	-	-	-	1.7	7.1	13.8	-	-	-	1.7	6.1	12.3
CollRec	-	-	-	**3.1**	**14.0**	**27.0**	**-**	**-**	**-**	**2.9**	**13.4**	**25.6**

TABLE 6. Metric performance for the response generation part. where IR stands for item ratio.

Models	ReDial								Reddit-Movie							
	Dist-2	Dist-3	Dist-4	IR	BL-2	BL-4	Rouge-L	PPL↓	Dist-2	Dist-3	Dist-4	IR	BL-2	BL-4	Rouge-L	PPL↓
Transformer	14.7	15.1	13.8	19.4	-	-	-	-	14.3	14.5	12.5	17.8	-	-	-	-
ReDial	22.4	23.6	22.7	15.8	17.8	7.4	16.9	61.7	22.1	22.9	22.2	14.6	16.7	6.9	15.7	57.4
KBRD	26.3	36.8	42.3	29.6	18.5	7.4	17.1	58.8	25.4	34.9	40.3	28.7	18.0	7.0	15.7	56.8
KGSF	28.8	43.4	51.7	32.4	16.4	7.4	14.4	131.1	27.3	40.9	46.8	31.0	15.4	6.9	13.6	124.1
GPT-2	35.4	48.6	44.1	14.5	17.1	7.6	11.2	56.3	33.5	44.7	40.3	13.2	15.9	7.2	10.1	53.1
BART	37.6	49.1	43.5	16.0	17.9	9.3	13.1	55.6	35.3	45.0	39.8	14.9	16.3	8.3	11.8	52.4
DialoGPT	47.6	55.9	48.7	15.9	16.7	7.8	12.4	56.0	44.7	53.5	44.1	14.9	15.3	7.0	11.2	52.7
CollRec	**51.8**	**62.4**	**59.8**	**43.5**	**20.4**	**11.0**	**17.6**	**54.1**	**49.3**	**60.1**	**57.4**	**39.5**	**18.7**	**10.1**	**16.3**	**51.5**

form of a chart to view the scores of each model more intuitively, as shown in Figure 7. It can be seen that the knowledge graph generation module in our CollRec model achieves the second best performance based on querying nodes and class edges and checking the variability of results through multiple random initializations, and is very close to the best-performing SOTA ReGen, indicating that our CollRec has entered the best ranks in the knowledge graph generation task. In addition, our CollRec model's experimental scores in the three types of Exact, Partial, and Strict are slightly higher than Amazon AI (Shanghai). Since Amazon AI (Shanghai) is the winner of the WebNLG 2020 Challenge in the text-to-RDF task, it can be seen that our CollRec model has good performance in the text-to-RDF task. Furthermore, this module utilizes focal loss to correct edge imbalances in the training process. With such small graphs and training data, the T5 model pre-trained on the text corpus of this module can better handle entity extraction since its representation ability is four orders of magnitude greater. In addition to allowing the module to extract nodes, if it is constructed from an unreliable set of nodes, subsequent stages of edge generation will also perform poorly.

For the edge imbalance problem, Cross-entropy is replaced using focal loss. In Figure 6 we show the dependence of the F1 score (under exact matching) on the focal length parameter $\gamma \geq 0$, defined previously in Section A in the Methods section. Among them, γ reduces the relative loss of well-classified examples while placing more emphasis on difficult, misclassified examples. We see that the performance is sensitive to the choice of γ, with $\gamma = 3$ achieving better results.

TABLE 7. Comparison results on ablation study.

Models	R@1	R@10	R@50	IR	BLEU	Rouge-L
CollRec	**3.0**	**14.1**	**27.2**	**43.3**	**20.6**	**17.8**
CollRec w/o KG	2.2	9.3	20.3	39.7	17.5	12.7
CollRec w/o PLM	1.8	8.8	19.5	17.8	18.5	14.6

2) CONVERSATION RECOMMENDATIONS

As shown in Table 5, we examined the main experimental results in both our CollRec and baseline models. The recommended modules and the final integrated system perform differently, as can be seen by the performance gap. In the evaluation of the recommendation module, the KGSF model achieved a Recall@1 of 3.9%, while in the evaluation of the final generated response, it achieved a Recall@1 of 0.9%. As a result, previous solutions to session recommendation significantly degraded performance because of the integration strategy used.

In addition, fine-tuning PLM on the CRS dataset is effective. We can see that compared with non-PLM-based methods, whether fine-tuning BART/DialoGPT/GPT2 directly on ReDial or on Reddit-Movie has achieved significant performance improvements in recommendation, which shows that our CollRec has good model generalization ability. However, the experimental data on Reddit-Movie is lower than ReDial to a certain extent, indicating that when the number of conversations and item recommendations is small, it is easy to cause a certain degree of overfitting of the recommendation model.

Whether on ReDial or on Reddit-Movie,Our model significantly outperforms SOTA in recommendation performance.

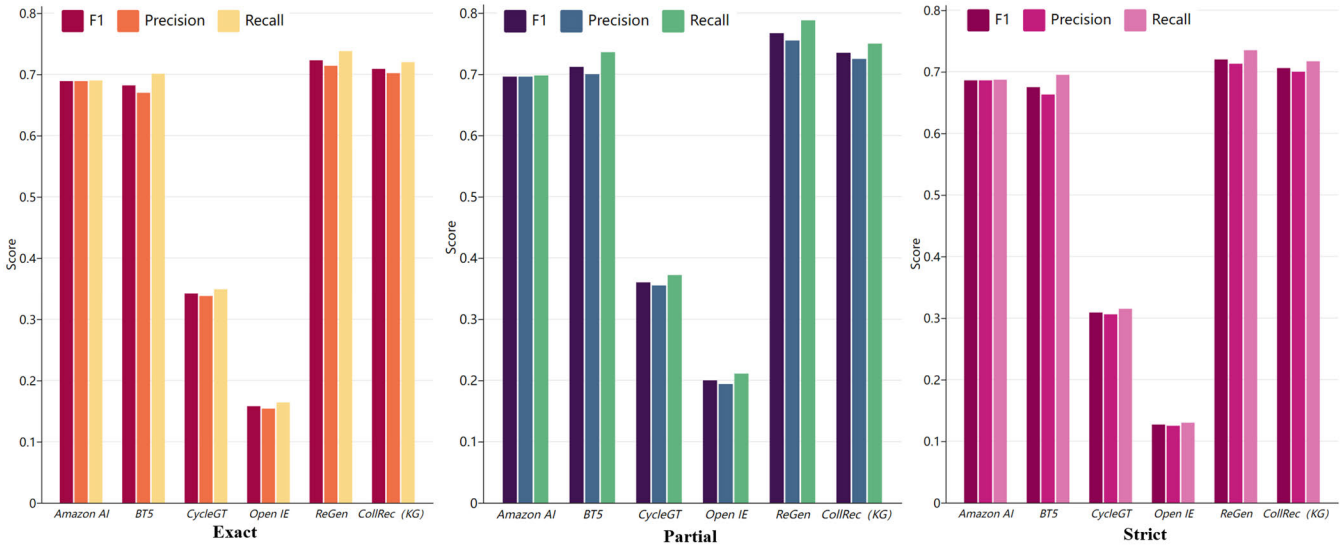
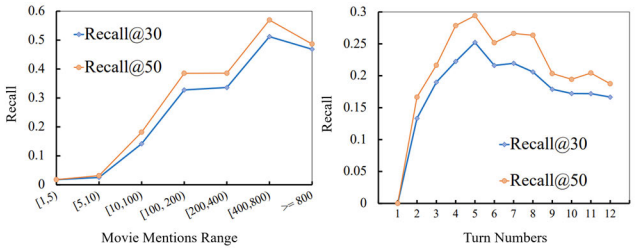


FIGURE 7. The performance of six models (including our CollRec) on the evaluation indicators F1, Precision, and Recall under Exact, Partial, and Strict respectively.



(a) Recall over Mentions (b) Recall over Turn #

FIGURE 8. Y-axis: memories. For Figure 9(a), X-axis: movie mention range. For Figure 9(b), X-axis: turns.

The CollRec score of Recall@k (k = 1, 10, 50) achieved under end-to-end evaluation reveals the superior performance of PLM through a unified framework that integrates design into a single system.

Table 6 shows Dist-2/3/4, BLEU-2/4, Rouge-L and PPL. In this regard, we can see PLM enables CollRec to generate more diverse responses than other baselines on Dist-n. Previous work suffers from insufficient resources due to small CRS data sets and often produces meaningless and single responses. Conversely, CollRec has a stronger item ratio score than the baseline, which suggests that our model makes more frequent recommendations. We also found that CollRec and PLM-based methods consistently perform better than non-PLM-based methods on all metrics. In dialogue generation, PLM performs better than other solutions.

F. ABLATION EXPERIMENT

For our CollRec model, in addition to the full model (1) **CollRec**, we evaluate three variants: (2) **CollRec w/o KG**, where we remove the knowledge graph part; (3)**CollRec w/o PLM**, where based The recommendation part of PLM was deleted, and only recommendations based on the knowledge graph were used.

We then show the performance comparison of CollRec variants. Table 7 shows the recommendation performance and generation results for both variants as well as the overall model. Deleting the word recommendation part based on PLM will cause R@k and Item Ratio to decrease significantly. CollRec's PLM-based recommendation component is crucial to the performance of project recommendations. In addition, we can find that the fine-tuning of recommendation methods enhanced by knowledge graphs helps achieve better recommendation performance. In addition to improving recommendation performance, node representations learned from CollRec's knowledge graph can be used to model user preferences and act as better targeting effects for various types of data.

G. QUALITATIVE ANALYSIS

A conversational example is provided in this section to demonstrate how our model operates.

In Figure 8, Explorer states that he likes horror movies. CollRec Responses generated with our model are more consistent with the context of the keywords scary and recommend the horror movie "It(2017)" whereas KGSF generated a safe response "Hello!". Besides, our CollRec recommended another horror movie released in the last century, 'Psycho' (1960), after the explorer stated he watched the old version of "It (1990)". This is probably because CollRec indicates the searcher is interested in old horror films. It is apparent from Figure 8 that the CollRec tends to generate richer responses than the KGSF. Additionally, we found that KGSF always generates "I would recommend item" (in this case Get out (2017) replaces item) and "I would recommend it". Movie items were successfully integrated into the first response pattern, but incomplete recommendations were given by the

Conversation Example	
User:	I like to watch scary movies
Human:	Tell me what kind of movie do you like? Did you watch ***It (2017)***? And ***It (1990)***?
KBRD:	Hi, what kind of movies do you like?
KGSF:	Hello!
CollRec:	Have you seen ***It (2017)***?
User:	I have seen the old one
Human:	Did you watch ***The Exorcist (1973)***?
KBRD:	***Get Out (2017)***
KGSF:	I would recommend ***Get Out (2017)***
CollRec:	How about ***Psycho (1960)***?
User:	I have not seen the new one I do like the exorcist
Human:	Ok, you should watch the new one even though I would rather the old one. And did you watch ***The Last Exorcism (2010)***? I hope I have helped
KBRD:	I haven't seen that one
KGSF:	I would recommend it
CollRec:	Yes, it was good. I hope you enjoy it. Have a good day. Bye

FIGURE 9. A conversation example to illustrate CollRec.

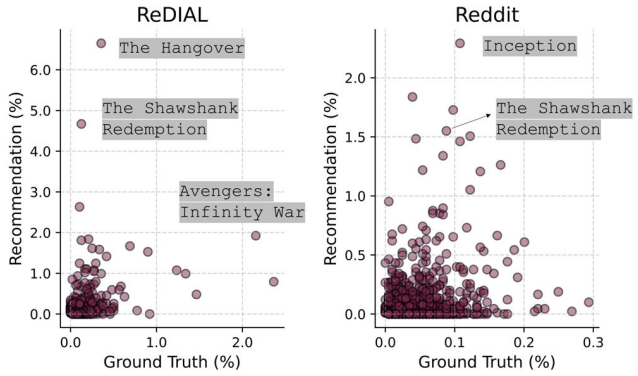


FIGURE 10. Scatter plot of proposals generated by the pr-trained language model and ground-truth item frequencies.

second pattern, which resulted in a flaw in the KGSF replication algorithm.

H. FURTHER ANALYSIS

1) DATA IMBALANCE ANALYSIS

A disbalanced distribution of movie frequency can be observed among different movies, as we discussed earlier. Figure 9(a) shows the Recall@30 and Recall@50 scores for movie mentions to study the effect. We can see that low-frequency movies (less than 10 mentions) have much lower recall scores than high-frequency movies (>100 mentions). Nonetheless, ReDial's dataset contains a majority of low-frequency movies (5470 out of 6924), leading to low overall performance.

2) COLD START ANALYSIS

The ReDial dataset has a cold start problem. The model has difficulty recommending precise items in the first few rounds of the conversation. In Figure 9(b), we compare CollRec scores across different dialogue turns at Recall@30 and Recall@50. Generally speaking, we can see that the recall rate gets higher and higher as richer information is gradually obtained from the conversational interaction. When it exceeds 5 rounds, the score starts to decrease. Due to the progress of the conversation, Seekers may become less satisfied with high-frequency recommendations and prefer more personalized ones, making prediction more difficult.

V. CONCLUSION

In this work, we address the problem of pre-trained language models and knowledge graphs synergistically enhancing the performance of conversational recommendation systems, for which a novel multi-stage system called CollRec is proposed. The proposed system divides the entire session recommendation task into two steps. In the first step, a pretrained language model is used to generate nodes from the input text. The generated node features will then be used in the subsequent steps of edge generation to build a tailored, smaller-scale knowledge graph that satisfies the entity scope. In the second step, we integrate item recommendations into the generation process. Specifically, we fine-tune large-scale PLM and relational graph convolutional networks on the knowledge graph constructed by the first step. Experiments with the CollRec system on text-to-RDF task generation on the smaller WebNLG dataset and extensive experiments on the CRS benchmark dataset ReDial demonstrate that by building targeted knowledge graphs and unifying response generation and item recommendation into existing In PLM, CollRec significantly outperforms state-of-the-art methods.

VI. DISCUSSION AND FUTURE WORK

In this section, we discuss the strengths and limitations of our CollRec model and suggest possible future work.

A. ADVANTAGES AND LIMITATIONS

The advantages of our CollRec model include using its own knowledge graph built based on user context information as external information when making recommendations. By matching the key words in the user context with the trained corpus in the pre-trained language model, the graph nodes and features related to the recommendation are efficiently extracted and used for graph relationship generation. This method can effectively reduce the size of the knowledge graph without missing relevant nodes, improving the interpretability of the CollRec model in the recommendation results. In addition, as shown in Figure 8, since CollRec uses a fine-tuned large-scale pre-trained language model for response generation, the generated responses can more accurately understand the questions raised by users and generate replies that are closer to real humans. This makes the recommendations of the CollRec model richer and more humane.

However, our CollRec model also suffers from popularity bias. Popularity bias refers to a phenomenon where popular items are recommended more often than their popularity would indicate [57]. Figure 10 shows the popularity bias in the recommendations of a pre-trained language model, although it may not be biased towards popular items in the target dataset. On ReDIAL, the most popular movie, such as Avengers: Infinity War, appears about 2% of all real items; on Reddit-Movie, the most popular movie, such as Everything Everywhere All at Once, appears less than 0.3% of the time in real items. But for the recommendations generated by the pre-trained language model (most LLMs have similar trends), the most popular item, such as The Shawshank Redemption, appears about 5% of the time on ReDIAL and about 1.5% of the time on Reddit. Compared to the target dataset, the recommendations containing the pre-trained language model are more concentrated on popular items, which may lead to further problems such as bias amplification loops [57]. In addition, the recommended popular items are similar across different datasets, which may reflect the popularity of items in the pre-trained language model corpus.

B. FUTURE WORK

About CRS. Our results show that recommendation performance is affected by dataset size and popularity bias. Next, we need new datasets from different sources, such as crowdsourcing platforms, discussion forums, and real-world CRS applications with various domains, languages, and cultures. More CRS models should be systematically re-benchmarked to fully understand their recommendation capabilities and the characteristics of CRS tasks.

REFERENCES

[1] R. Li, S. E. Kahou, H. Schulz, V. Michalski, L. Charlin, and C. Pal, "Towards deep conversational recommendations," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, 2018, pp. 9748–9758.

[2] Y. Sun and Y. Zhang, "Conversational recom-mender system," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Ann Arbor, MI, USA, Jul. 2018, pp. 235–244.

[3] K. Zhou, W. X. Zhao, S. Bian, Y. Zhou, J.-R. Wen, and J. Yu, "Improving conversational recommender systems via knowledge graph based semantic fusion," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Virtual Event, CA, USA, Aug. 2020, pp. 1006–1014.

[4] Q. Chen, J. Lin, Y. Zhang, M. Ding, Y. Cen, H. Yang, and J. Tang, "Towards knowledge-based recommender dialog system," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, Hong Kong, 2019, pp. 1803–1813.

[5] Y. Shao, S. Gouws, D. Britz, A. Goldie, B. Strope, and R. Kurzweil, "Generating high-quality and informative conversation responses with sequence-to-sequence models," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Copenhagen, Denmark, 2017, pp. 2210–2219.

[6] M. Ghazvininejad, C. Brockett, M.-W. Chang, B. Dolan, J. Gao, W. Yih, and M. Galley, "A knowledge-grounded neural conversation model," in *Proc. 32nd AAAI Conf. Artif. Intell. (AAAI), 30th Innov. Appl. Artif. Intell. (IAAI), 8th AAAI Symp. Educ. Adv. Artif. Intell. (EAAI)*, New Orleans, LA, USA, 2018, pp. 5110–5117.

[7] X. Wang, D. Wang, C. Xu, X. He, Y. Cao, and T.-S. Chua, "Explainable reasoning over knowledge graphs for recommendation," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 5329–5336.

[8] X. Lin, W. Jian, J. He, T. Wang, and W. Chu, "Generating informative conversational response using recurrent knowledge-interaction and knowledge-copy," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 41–52.

[9] K. Zhou, Y. Zhou, W. X. Zhao, X. Wang, and J.-R. Wen, "Towards topic-guided conversational recommender system," in *Proc. 28th Int. Conf. Comput. Linguistics*, Barcelona, Spain, 2020, pp. 4128–4139.

[10] Z. Liu, H. Wang, Z.-Y. Niu, H. Wu, W. Che, and T. Liu, "Towards conversational recommendation over multi-type dialogs," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 1036–1049.

[11] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, pp. 1–67, Jan. 2020.

[12] Y. Zhang, S. Sun, M. Galley, Y.-C. Chen, C. Brockett, X. Gao, J. Gao, J. Liu, and B. Dolan, "DIALOGPT: Large-scale generative pre-training for conversational response generation," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics, Syst. Demonstrations*, 2020, pp. 270–278.

[13] T. C. Ferreira, C. Gardent, N. Ilinykh, C. Lee, S. Mille, D. Moussallem, and A. Shimorina, "The 2020 bilingual, bi-directional WebNLG+ shared task overview and evaluation results (WebNLG+2020)," in *Proc. Int. Workshop Natural Lang. Gener. From Semantic Web*, 2020, pp. 1–23.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1–11.

[15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, vol. 1, 2019, pp. 4171–4186.

[16] X. Li, A. Taheri, L. Tu, and K. Gimpel, "Com-monsense knowledge base completion," in *Proc. Annu. Meeting ACL*, 2016, pp. 1445–1455.

[17] L. Yao, C. Mao, and Y. Luo, "KG-BERT: BERT for knowledge graph completion," 2019, *arXiv:1909.03193*.

[18] C. Malaviya, C. Bhagavatula, A. Bosselut, and Y. Choi, "Commonsense knowledge base completion with structural and semantic context," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2020, pp. 2925–2933.

[19] F. Petroni, T. Rocktäschel, S. Riedel, P. Lewis, A. Bakhtin, Y. Wu, and A. Miller, "Language models as knowledge bases?" in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 1–11.

[20] A. Roberts, C. Raffel, and N. Shazeer, "How much knowledge can you pack into the parameters of a language model?" in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2020, pp. 1–9.

[21] Z. Jiang, F. F. Xu, J. Araki, and G. Neubig, "How can we know what language models know?" *Trans. Assoc. for Comput. Linguistics*, vol. 8, pp. 423–438, Dec. 2020.

[22] T. Shin, Y. Razeghi, R. L. Logan, E. Wallace, and S. Singh, "AutoPrompt: Eliciting knowledge from language models with automatically generated prompts," 2020, *arXiv:2010.15980*.

[23] X. L. Li and P. Liang, "Prefix-tuning: Optimizing con-tinuous prompts for generation," 2101, *arXiv:2101.00190*.

[24] Y. Zhang, X. Chen, Q. Ai, L. Yang, and W. B. Croft, "Towards conversational search and recommendation: System ask, user respond," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2018, pp. 177–186.

[25] X. Wu, H. Hu, M. Klyen, K. Tomita, and Z. Chen, "Q20: Rinna riddles your mind by asking 20 questions," in *Proc. NLP*, 2018, pp. 1–4.

[26] J. Zou and E. Kanoulas, "Learning to ask: Question-based sequential Bayesian product search," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Beijing, China, Nov. 2019, pp. 369–378.

[27] J. Zou, Y. Chen, and E. Kanoulas, "Towards question-based recommender systems," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Virtual Event, China, Jul. 2020, pp. 881–890.

[28] Y. Chen, B. Chen, X. Duan, J.-G. Lou, Y. Wang, W. Zhu, and Y. Cao, "Learning-to-ask: Knowledge acquisition via 20 questions," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 1216–1225.

[29] W. Lei, X. He, Y. Miao, Q. Wu, R. Hong, M.-Y. Kan, and T.-S. Chua, "Estimation-action-reflection: Towards deep interaction between conversational and recommender systems," in *Proc. 13th Int. Conf. Web Search Data Mining*, Houston, TX, USA, Jan. 2020, pp. 304–312.

[30] Y. Deng, Y. Li, F. Sun, B. Ding, and W. Lam, "Unified conversational recommendation policy learning via graph-based reinforcement learning," 2021, *arXiv:2105.09710*.

[31] X. Ren, H. Yin, T. Chen, H. Wang, N. Q. V. Hung, Z. Huang, and X. Zhang, "CRSAL: Conversational recommender systems with adversarial learning," *ACM Trans. Inf. Syst.*, vol. 38, no. 4, pp. 1–40, Oct. 2020.

[32] H. Xu, S. Moon, H. Liu, B. Liu, P. Shah, B. Liu, and P. Yu, "User memory reasoning for conversational recommendation," in *Proc. 28th Int. Conf. Comput. Linguistics*, Barcelona, Spain, 2020, pp. 5288–5308.

[33] W. Lei, G. Zhang, X. He, Y. Miao, X. Wang, L. Chen, and T.-S. Chua, "Interactive path reasoning on graph for conversational recommendation," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Virtual Event, CA, USA, Aug. 2020, pp. 2073–2083.

[34] L. Li, W. Chu, J. Langford, and R. E. Schapire, "A contextual-bandit approach to personalized news article recommendation," in *Proc. 19th Int. Conf. World Wide Web*, Apr. 2010, pp. 661–670.

[35] K. Christakopoulou, F. Radlinski, and K. Hofmann, "Towards conversational recommender systems," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, San Francisco, CA, USA, Aug. 2016, pp. 815–824.

[36] S. Li, W. Lei, Q. Wu, X. He, P. Jiang, and T.-S. Chua, "Seamlessly unifying attributes and items: Conversational recommendation for cold-start users," 2020, *arXiv:2005.12979*.

[37] S. Jiang, P. Ren, C. Monz, and M. de Rijke, "Improving neural response diversity with frequency-aware cross-entropy loss," in *Proc. World Wide Web Conf.*, San Francisco, CA, USA, May 2019, pp. 2879–2885.

[38] S. A. Hayati, D. Kang, Q. Zhu, W. Shi, and Z. Yu, "INSPIRED: Toward sociable recommendation dialog systems," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2020, pp. 8142–8152.

[39] W. Ma, R. Takanobu, and M. Huang, "CR-Walker: Tree-structured graph reasoning and dialog acts for conversational recommendation," 2020, *arXiv:2010.10333*.

[40] L. Wang, S. Joty, W. Gao, X. Zeng, and K.-F. Wong, "Improving conversational recommender system via contextual and time-aware modeling with less domain-specific knowledge," 2022, *arXiv:2209.11386*.

[41] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, Feb. 2020.

[42] T. Wolf, J. Chaumond, L. Debut, V. Sanh, C. Delangue, A. Moi, P. Cistac, M. Funtowicz, J. Davison, and S. Shleifer, "Transformers: Stateof-the-art natural language processing," in *Proc. Conf. Empirical Methods Natural Lang. Process., Syst. Demonstrations*, 2020, pp. 38–45.

[43] Q. Guo, Z. Jin, N. Dai, X. Qiu, X. Xue, D. Wipf, and Z. Zhang, "*P*2: A plan-and-pretrain approach for knowledge graph-to-text generation," in *Proc. Int. Workshop Natural Lang. Gener. Semantic Web*, 2020, pp. 100–106.

[44] Q. Guo, Z. Jin, X. Qiu, W. Zhang, D. Wipf, and Z. Zhang, "CycleGT: Unsupervised graph-to-text and text-to-graph generation via cycle training," 2020, *arXiv:2006.04702*.

[45] O. Agarwal, M. Kale, H. Ge, S. Shakeri, and R. Al-Rfou, "Machine translation aided bilingual data-to-text generation and semantic parsing," in *Proc. Int. Workshop Natural Lang. Gener. Semantic Web*, 2020, pp. 125–130.

[46] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, "The Stanford CoreNLP natural language processing toolkit," in *Proc. Assoc. Comput. Linguistics*, 2014, pp. 55–60.

[47] P. L. Dognin, I. Padhi, I. Melnyk, and P. Das, "ReGEN: Reinforcement learning for text and knowledge base generation using pretrained language models," 2021, *arXiv:2108.12472*.

[48] I. V. Serban, A. Sordoni, R. Lowe, L. Charlin, J. Pineau, A. C. Courville, and Y. Bengio, "A hierarchical latent variable encoder–decoder model for generating dialogues," in *Proc. 31st AAAI Conf. Artif. Intell.*, San Francisco, CA, USA, 2017, pp. 3295–3301.

[49] J. He, H. Hankui Zhuo, and J. Law, "Distributed-representation based hybrid recommender system with short item descriptions," 2017, *arXiv:1703.04854*.

[50] I. Segura-Bedmar, P. Martínez, and M. Herrero-Zazo, "SemEval-2013 task 9: Extraction of drug-drug interactions from biomedical texts (DDIExtraction 2013)," in *Proc. SemEval@NAACL-HLT*, 2013, pp. 341–350.

[51] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: A method for automatic evaluation of machine translation," in *Proc. 40th Annu. Meeting Assoc. Comput. Linguistics*, Philadelphia, PA, USA, 2001, pp. 311–318.

[52] C.-Y. Lin, "ROUGE: A package for automatic evalu-ation of summaries," in *Proc. Text Summarization Branches Out*, Barcelona, Spain, 2004, pp. 74–81.

[53] J. Li, M. Galley, C. Brockett, J. Gao, and B. Dolan, "A diversity-promoting objective function for neural conversation models," in *Proc. Conf. North Amer. Chapter Assoc. for Comput. Linguistics: Human Lang. Technol.*, 2016, pp. 110–119.

[54] Z. He, Z. Xie, R. Jha, H. Steck, D. Liang, Y. Feng, B. P. Majumder, N. Kallus, and J. Mcauley, "Large language models as zero-shot conversational recommenders," in *Proc. 32nd ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2023, pp. 720–730.

[55] J. Baumgartner, S. Zannettou, B. Keegan, M. Squire, and J. Blackburn, "Thepushshift Reddit dataset," in *Proc. Int. AAAI Conf. Web Social Media*, vol. 14, 2020, pp. 830–839.

[56] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," 2020, *arXiv:2005.12872*.

[57] J. Chen, H. Dong, X. Wang, F. Feng, M. Wang, and X. He, "Bias and debias in recommender system: A survey and future directions," *ACM Trans. Inf. Syst.*, vol. 41, no. 3, pp. 1–39, Jul. 2023.

SHUANG LIU was born in Jinzhou, China, in 1977. She received the Ph.D. degree in traffic information engineering and control from Dalian Maritime University, in 2006. From 2012 to 2015, she was a Postdoctoral Fellow with the Computer Science and Technology Postdoctoral Station, Dalian University of Technology. She is currently a Professor with the School of Computer Science and Engineering, Dalian Minzu University, Dalian, China. Her academic papers have been published in both national and international journals and conferences, such as IEEE Access, in 2022, IJCIS, in 2021, and *Information*, in 2020. Her research interests include machine learning, object detection, knowledge graphs, and scientific visualization.

ZHIZHUO AO was born in Qiqihar, Heilongjiang, China, in 1999. He received the B.E. degree in computer science and technology from Dalian Minzu University, China, where he is currently pursuing the master's degree in electronic Information. His primary research interests include knowledge graph, large language model, and conversational recommendation systems.

PENG CHEN was born in Xuzhou, Jiangsu, China. He received the master's degree in computer science from Liaoning Shihua University, in 2003. He is currently a Professor with the School of Computer and Software, Dalian Neusoft University of Information. His research interests include knowledge graphs and machine learning algorithms. His academic papers are published in both national and international journals and conferences, such as *Journal of Database Management* and *Applied Sciences*.

SIMON KOLMANIČ was born in Ormož, Slovenia, in 1972. He received the Ph.D. degree in computer science and informatics from the School of Electrical Engineering and Computer Science, University of Maribor, Slovenia, in 2006. He is currently teaching with the School of Electrical Engineering and Computer Science, University of Maribor. He teaches subjects of algorithmic fundamentals, computer graphics, and animation, directing subjects in Cinema 4D and Blender: Introduction to Geometric Modeling and Computer Graphics, Media Computer Animation, and *Computer Graphics and Image Processing*.