Parameter-Efficient Conversational Recommender System as a Language Processing Task

Mathieu Ravaut[1], **Hao Zhang**[1], **Lu Xu**[2], **Aixin Sun**[1], **Yong Liu**[1]
[1]School of Computer Science and Engineering, Nanyang Technological University, Singapore
[2]Singapore University of Technology and Design
mathieuj001@e.ntu.edu.sg

Abstract

Conversational recommender systems (CRS) aim to recommend relevant items to users by eliciting user preference through natural language conversation. Prior work often utilizes external knowledge graphs for items' semantic information, a language model for dialogue generation, and a recommendation module for ranking relevant items. This combination of multiple components suffers from a cumbersome training process, and leads to semantic misalignment issues between dialogue generation and item recommendation. In this paper, we represent items in natural language and formulate CRS as a natural language processing task. Accordingly, we leverage the power of pre-trained language models to encode items, understand user intent via conversation, perform item recommendation through semantic matching, and generate dialogues. As a unified model, our PECRS (Parameter-Efficient CRS), can be optimized in a single stage, without relying on non-textual metadata such as a knowledge graph. Experiments on two benchmark CRS datasets, ReDial and INSPIRED, demonstrate the effectiveness of PECRS on recommendation and conversation. Our code is available at: https://github.com/Ravoxsg/efficient_unified_crs.

1 Introduction

Conversational recommender systems (CRS) have become an active research topic, which leverages both natural language processing and recommendation techniques to provide high-quality recommendations through interactive conversations with users (Jannach et al., 2021; Gao et al., 2021; Pramod and Bafna, 2022).

CRS consists of two sub-tasks: 1) generating natural language responses to interact with user (*conversation*); and 2) recommending desirable items to user based on dialogue context (*recommendation*). An example of CRS data and model prediction is shown in Figure 1. In general, CRS represents a significant advancement in the field of recommendation, which could be applied to various possible use cases, such as e-commerce, entertainment and content platforms.

Dialogue context:
Seeker: Hello.
Recommender: Hi. What movies do you want me to recommend?
Seeker: Deadpool 2, very good movie, I recommend it. Recommend me a movie.
Recommender: Well, movies like that. I recommend The Avengers and X-Men: Apocalypse. They are very very good movies!
Seeker: I head about these movies. I'll keep it in mind. What other could you recommend me?

PECRS: I can recommend Black Panther. Have you seen that one?

Recommender: If you like those movies then do not miss seeing Black Panther. I know you like them.

Figure 1: An example of dialogue from ReDial (Li et al., 2018), where blue color denotes the movie items.

Existing CRS methods can be roughly categorized into *attribute-based* and *generation-based* methods. The attribute-based methods (Lei et al., 2020; Ren et al., 2020; Zou et al., 2020) focus on collecting user preferences on item attributes to narrow down recommendation space to items with desired properties. The generation-based methods (Zhou et al., 2020a, 2022; Wang et al., 2022c) aim to acquire feedback from users, generate natural responses, and establish a comprehensive understanding of conversation to recommend the most desirable items to user. In this work, we focus on generation-based CRS, which was greatly facilitated with the rise of task-specific CRS datasets like ReDial (Li et al., 2018), INSPIRED (Hayati et al., 2020), TG-ReDial (Zhou et al., 2020b) and DuRecDial (Liu et al., 2020).

The key challenge of CRS methods consists in how to jointly model language generation and item recommendation, which are tasks of entirely different natures. Early approaches (Chen et al., 2019; Zhou et al., 2020a; Zhang et al., 2022; Zhou et al., 2022) mainly model conversation and recommendation tasks separately by incorporating external

Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics
Volume 1: Long Papers, pages 152–165

knowledge graphs (KG) for item semantics and designing auxiliary strategies to enhance the interactions between two tasks. They generally treat items as nodes, which neglects the affluent textual information of items. They also sustain semantic misalignment issue due to inconsistent item and word representations, because conversation and recommendation modules are separately learned. Recent approaches (Wang et al., 2022a,b,c; Yang et al., 2022) explore to seamlessly integrate conversation and recommendation modules for better knowledge sharing and semantic alignment via unified frameworks. However, due to the natural gap between recommendation and conversation, they still require multiple training phases (Wang et al., 2022c) and/or additional modules (Wang et al., 2022a; Yang et al., 2022) to integrate the two tasks, failing to reach desired level of integration.

With the rapid development of language models (LMs), LMs for recommendation has gained significant attention. Based on LMs, recent work (Wu et al., 2023; Lin et al., 2023) also shows a growing correlation between recommendation and language tasks. Thus, instead of applying structured KGs, we stick to using item text descriptions together with dialogue contexts for CRS, which formulates the CRS directly as a natural language processing task. Specifically, we devise a **P**arameter-**E**fficient **C**onversational **R**ecommender **S**ystem (**PECRS**), which jointly solves recommendation and conversation by training a single model once, to bypass the shortcomings of prior work in CRS. PECRS only relies on a frozen pre-trained LM as backbone and employs a parameter-efficient plugin module to unify response generation and item recommendation in a simple yet flexible manner. Besides, we design a shared negative sampling strategy to sample negative items across subtasks and data points within the same mini-batch to boost both training efficiency and model performance. Moreover, thanks to the parameter-efficient plugin module, PECRS can easily scale up to larger LM backbones without significantly increasing training parameters. In brief, our contributions are the following:

- To the best of our knowledge, this is the first work solving CRS by optimizing a single model in a single training phase and bypassing the need for either KGs or additional item encoders.
- We demonstrate how to jointly generate response and learn item representations using a single and frozen language model. Through parameter-efficient fine-tuning techniques, our method is with low computation cost, and can easily scale to larger backbones for higher performance.
- Experiments on two benchmark datasets, ReDial and INSPIRED, demonstrate the effectiveness of our proposed PECRS method, which is competitive with SOTA.

2 Related Work

Existing conversational recommender systems (CRS) can be roughly categorized into attribute-based and generation-based CRS methods. The attribute-based CRS methods utilize predefined actions to interact with users and target on accomplishing the recommendation task with fewer turns (Christakopoulou et al., 2016; Sun and Zhang, 2018; Lei et al., 2020; Ren et al., 2020; Zou et al., 2020; Hu et al., 2022a). Our work belongs to the generation-based CRS, which focuses on developing natural language based approaches to make high-quality recommendation and generate human-like responses simultaneously (Li et al., 2018; Hayati et al., 2020; Zhou et al., 2020b; Liu et al., 2020).

Generation-based CRS methods usually devise a recommendation module and a conversation module to implement item recommendation and response generation, respectively. Li et al. (2018) propose the first CRS dataset named ReDial, and solve it via encoder-decoder-based dialogue generator and autoencoder-based recommender. Subsequent work commonly adopts external resources to incorporate sufficient contextual information for better performance. Numerous works (Chen et al., 2019; Zhou et al., 2020a, 2021; Ma et al., 2020; Zhang et al., 2022; Liang et al., 2021; Li et al., 2022; Liu et al., 2023; Zhang et al., 2023b) use knowledge graphs (KG) (Auer et al., 2007; Speer et al., 2017) coupled with graph networks (Schlichtkrull et al., 2018) to enhance the items and user preference understanding by designing sophisticated semantic alignment strategies. RevCore (Lu et al., 2021) and C^2-CRS (Zhou et al., 2022) further incorporate movie reviews to enrich the contextual knowledge via cross-attention (Lu et al., 2021) and contrastive learning (Zhou et al., 2022). Despite consecutive improvements, these works rely on different architectures for conversation and recommendation, making them difficult to be effectively integrated for end-to-end training and knowledge sharing. Consequently, they still suffer from a mismatch between conversation and recommendation

modules as well as inferior efficiency.

To remedy the aforementioned issues, recent approaches explore to jointly learn both conversation and recommendation tasks by pre-trained LMs. UniCRS (Wang et al., 2022c) adopts the DialoGPT (Zhang et al., 2020) for both conversation and recommendation by tuning soft prompts (Lester et al., 2021) dedicated to each task. Nevertheless, UniCRS requires three rounds of optimization, *i.e.,* semantic fusion pre-training, conversation tuning, and recommendation tuning. UniMIND (Deng et al., 2023) follows the UniCRS paradigm with BART (Lewis et al., 2020) as the backbone, which unifies multi-goal CRS, *i.e.,* multi-tasks, using prompting strategy with multiple training stages. RecInDial (Wang et al., 2022a) augments items into DialoGPT vocabulary and designs a pointer mechanism for dynamic word and item prediction to achieve single multi-tasking process. Similarly, BARCOR (Wang et al., 2022b) utilizes BART to recommend items with encoder and generate responses with decoder concurrently. Instead of using KG, MESE (Yang et al., 2022) encodes item representations using metadata and fuses them into dialogue for joint conversation and recommendation learning using GPT-2 (Radford et al., 2019) as the backbone. Although these methods attempt to integrate conversation and recommendation tasks for joint optimization, they rely on extra modules (*e.g.,* R-GCN (Schlichtkrull et al., 2018) and DistilBERT (Sanh et al., 2019)) for either item encoding or semantic fusion, and multi-round training stages. In contrast, our goal is to design a framework to unify the CRS training under a single model optimized in a single training stage.

Our work also employs parameter-efficient fine-tuning (PEFT) strategies. PEFT, including prompt tuning (Lester et al., 2021), Adapters (Houlsby et al., 2019), and LoRA (Hu et al., 2022b), is a series of techniques to adapt (large) LMs with fewer parameters and low computation costs to achieve same or even better performance comparing to the standard fine-tuning on downstream tasks. PEFT has shown great promise in various natural language (Zhang et al., 2023a; Dettmers et al., 2023), computer vision (He et al., 2022; Chen et al., 2023), and recommendation (Fu et al., 2023) tasks, but remains underexplored in CRS area. In this work, we aim to train CRS via PEFT plugins without touching the parameters of the backbone LM.

3 Methodology

In this section, we first describe the problem statement of conversational recommendation systems (CRS). Then we present the proposed **P**arameter-**E**fficient **C**onversational **R**ecommender **S**ystem (**PECRS**) method in detail. The overall architecture of PECRS is shown in Figure 2.

3.1 Problem Formulation

Let $\boldsymbol{\mathcal{I}} = \{I_1, I_2, \ldots, I_{N_{\text{item}}}\}$ represent the item database, which contains N_{item} unique items, and $\boldsymbol{\mathcal{D}} = \{D_1, D_2, \ldots, D_{N_{\text{dial}}}\}$ denote a CRS dataset with N_{dial} dialogues. Each dialogue D consists of n_{utt} utterances denoted by $D = \{u_t\}_{t=1}^{n_{\text{utt}}}$, where u_t represents the utterance at the t-th turn and each utterance $u_t = \{w_j\}_{j=1}^{n}$ contains a sequence of n words. The task of CRS is to generate the response and recommend desirable items based on the given dialogue history and item database. To be specific, given the dialogue history up to the t-th turn $D_t = \{u_i\}_{i=1}^{t-1}$ and the item database $\boldsymbol{\mathcal{I}}$, the CRS needs to *recommend* a set of candidate items $\boldsymbol{\mathcal{I}}_t$ from $\boldsymbol{\mathcal{I}}$, and *generate* the response u_t which includes the items $\boldsymbol{\mathcal{I}}_t$. The recommended candidate items set $\boldsymbol{\mathcal{I}}_t$ could be *empty* when no recommendation is needed, or contain *one* or *more* items depending on the responses.

In this work, we apply our method to the *movie recommendation* (*i.e.,* $\boldsymbol{\mathcal{I}}$ denotes a movie items set), but the process would be identical with other types of items. We follow prior work (Wang et al., 2022c; Yang et al., 2022) to adjust data samples and predict response with *a single recommended movie per utterance*.

3.2 Model Input

In PECRS, items are represented by their textual descriptions, hence both input streams are modeled as text. Nevertheless, we design a few special tokens to distinguish the various elements in PECRS.

Special Tokens. Our PECRS is built upon a pre-trained LM under the decoder-only style, parameterized by θ (e.g, GPT-2). However, LMs generally do not have the capacity for recommendation task. Thus, we define four special tokens, *i.e.,* "[ITEM]", "[SEP]", "[REC]" and "[REC_END]", and add them into the LM's vocabulary to guide the model's understanding of recommended items.

Item Metadata. Prior work (Zhou et al., 2020a; Zhang et al., 2022; Wang et al., 2022a; Zhou et al.,

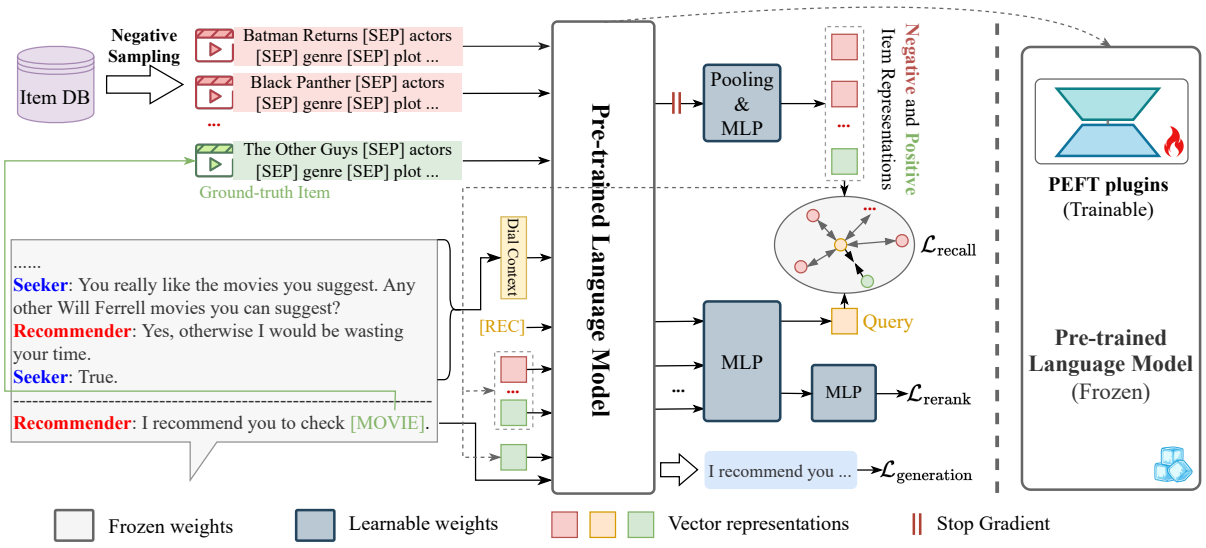

Figure 2: The overall architecture of the proposed Parameter-efficient Conversation Recommendation System (PECRS), where the PEFT denotes the parameter-efficient fine-tuning. Instead of fine-tuning backbone model, we inject PEFT plugins into backbone model and fine-tune the PEFT weights (see the figure in the right).

2022; Wang et al., 2022c) usually exploits external KG to encode item representations. They generally regard items as nodes and model relations among items through R-GCN (Schlichtkrull et al., 2018), but neglect the rich textual descriptions of the items. In contrast, similar to Yang et al. (2022), we explore to use the static textual metadata of items. Item descriptions can be fed into a language model directly, hence bypassing the semantic misalignment issue. To be specific, each item I_j is represented by affluent relevant information of the item rather than just its title. For movie recommendation, we use the following format "*Movie title [SEP] Actors [SEP] Director(s) [SEP] Genre(s) [SEP] Plot*" to describe a movie item, where [SEP] is used to mark the separation among different fields. Note this process can be directly generalized to other domains by using the meta information of items in the target domain. Formally, let $I_j = \{c_{j,k}\}_{k=1}^{l}$ denotes the j-th item textual data with l tokens, its output from LM is $\boldsymbol{I}_j = [\boldsymbol{c}_{j,1}, \ldots, \boldsymbol{c}_{j,l}]$. We further adopt a MLP layer h_{item} with learnable pooling weight $\boldsymbol{w}$ to aggregate the item representation as:

$$\boldsymbol{v}_j = h_{\text{item}}(\boldsymbol{w}^T \cdot \boldsymbol{I}_j). \tag{1}$$

Dialogue Context. The dialogue context is made of all utterances up to the current t-th utterance: $D_t = \{u_i\}_{i=1}^{t-1}$. The word embeddings of the i-th utterance are denoted as $\boldsymbol{u}_i = [\boldsymbol{c}_{i,1}, \ldots, \boldsymbol{c}_{i,n}]$. If any utterance u_i contains an item, it will be replaced by "[ITEM]" token and its item representation is also concatenated to the left side of the utterance's word embeddings. Otherwise, it remains unchanged. Let $\boldsymbol{v}_{\text{sep}}$, $\boldsymbol{v}_{\text{rec}}$ and $\boldsymbol{v}_{\text{rec\_end}}$ denote the token representations of "[SEP]", "[REC]" and "[REC_END]", respectively. Suppose the i-th utterance contains an item, if it is from *seeker*, its token embeddings are represented as $\tilde{\boldsymbol{u}}_i = [\boldsymbol{v}_{\text{sep}}, \boldsymbol{v}_j, \boldsymbol{v}_{\text{sep}}, \boldsymbol{u}_i]$; if it is from *recommender*, its token embeddings are $\tilde{\boldsymbol{u}}_i = [\boldsymbol{v}_{\text{rec}}, \boldsymbol{v}_j, \boldsymbol{v}_{\text{rec\_end}}, \boldsymbol{u}_i]$. Thus, the token embedding sequences of dialogue context are the concatenation of all utterances with $\boldsymbol{v}_{\text{rec}}$ representation:

$$\boldsymbol{D}_t = [\bar{\boldsymbol{u}}_1, \ldots, \bar{\boldsymbol{u}}_{t-1}, \boldsymbol{v}_{\text{rec}}], \tag{2}$$

where $\bar{\boldsymbol{u}}_i = \tilde{\boldsymbol{u}}_i$ if the utterance contains items, otherwise $\bar{\boldsymbol{u}}_i = \boldsymbol{u}_i$.

3.3 Recommendation

The recommendation module contains two processes: retrieval and re-ranking. The retrieval process is to select candidate items relevant to dialogue context from item database. The re-ranking process further re-ranks the selected candidate items after aggregating knowledge from the dialogue context.

Retrieval. We use the movie item in the response to be predicted as the ground-truth item, and sample M negative items from item database. Then, we use their textual descriptions to encode item representations via Equation (1) and derive ground-truth item $\boldsymbol{v}_p$ and negative items $\{\boldsymbol{v}'_j\}_{j=1}^{M}$. As the dialogue context is ended with "[REC]" token (ref. Equation (2)) and decoder-only LM can aggregate all contextual information via causal self-attention,

we utilize LM's output of "[REC]" token, denoted as $\boldsymbol{d}_t$, to represent *query* representation of dialogue context. We adopt a noise-contrastive estimation (NCE) (Gutmann and Hyvärinen, 2012; Mnih and Teh, 2012; Mnih and Kavukcuoglu, 2013) objective to bring together the query $\boldsymbol{d}_t$ with the positive key $\boldsymbol{v}_p$ and push apart M negative (query, key) pairs formed by the set $\boldsymbol{\mathcal{N}} = \{(\boldsymbol{d}_t, \boldsymbol{v}'_j)\}_{j=1}^{M}$.

The NCE objective is written as:

$$\mathcal{E}_{D_t} = \frac{e^{f(\boldsymbol{d}_t)^\top \odot \boldsymbol{v}_p}}{e^{f(\boldsymbol{d}_t)^\top \odot \boldsymbol{v}_p} + \sum\limits_{(\boldsymbol{d}_t, \boldsymbol{v}'_j) \sim \boldsymbol{\mathcal{N}}} e^{f(\boldsymbol{d}_t)^\top \odot \boldsymbol{v}'_j}}, \quad (3)$$

where f is a projection head with two-layer MLP and ReLU activation; $\odot$ denotes the *angular* distance, $\sqrt{2(1-\cos(\boldsymbol{a}, \boldsymbol{b}))}$, which measures the similarity between two vectors, $\boldsymbol{a}$ and $\boldsymbol{b}$. The recall loss for retrieval process is defined as:

$$\mathcal{L}_{\text{recall}} = -\frac{1}{|\boldsymbol{\mathcal{D}}|} \sum_{D_t \in \boldsymbol{\mathcal{D}}} \log(\mathcal{E}_{D_t}). \quad (4)$$

Note we stop the gradients of LM and only optimize the pooling and MLP layers for item representations encoding during training (ref. Figure 2) to accelerate the learning process. The item representations will be reused in re-ranking process and the LM will be optimized at this stage accordingly.

Re-ranking. The item representations derived from retrieval process are reused in the re-ranking process to aggregate the knowledge of dialogue context. To be specific, given both positive and negative items, we concatenate them with the token embeddings of dialogue context as $[\boldsymbol{D}_t, \boldsymbol{v}_p, \boldsymbol{v}'_1, \ldots, \boldsymbol{v}'_M]$ and feed into LM then MLP f to compute the context-aware item representations $[\boldsymbol{q}_p, \boldsymbol{q}_1, \ldots, \boldsymbol{q}_M]$. Note that we adopt a special attention mask to enforce that each item $\boldsymbol{v}_j$ only attends to tokens from $\boldsymbol{D}_t$, and positional embeddings are removed for item tokens to avoid any position leakage. Then another MLP layer g is applied to compute the final item scores as $\boldsymbol{r} = [r_p, r_1, \ldots, r_M]$.

The training objective of re-ranking process is:

$$\mathcal{L}_{\text{rerank}} = \frac{1}{|\boldsymbol{\mathcal{D}}|} \sum_{D_t \in \boldsymbol{\mathcal{D}}} f_{\text{XE}}(\boldsymbol{r}, \boldsymbol{Y}), \quad (5)$$

where $\boldsymbol{Y} = [1, 0, \ldots, 0]$ and f_{XE} denotes cross-entropy loss. Note we shuffle r and $\boldsymbol{Y}$ jointly to avoid the positional bias of ground-truth labels. If a data point has no recommended item in the response, we set $\mathcal{L}_{\text{recall}} = \mathcal{L}_{\text{rerank}} = 0$.

3.4 Response Generation

The response generation aims to predict the current utterance $u_t = \{w_j\}_{j=1}^{n}$ by giving the dialogue context. During training, if the u_t contains an item to be recommended, the representations of the ground-truth item is appended to the corresponding dialogue context to guarantee that the LM generates the response relevant to the item. Then, the input for response generation is:

$$\tilde{\boldsymbol{D}}_t = [\bar{\boldsymbol{u}}_1, \ldots, \bar{\boldsymbol{u}}_{t-1}, \boldsymbol{v}_{\text{rec}}, \boldsymbol{v}_p, \boldsymbol{v}_{\text{rec\_end}}]. \quad (6)$$

Otherwise, the input for response generation stays as $\tilde{\boldsymbol{D}}_t = [\bar{\boldsymbol{u}}_1, \ldots, \bar{\boldsymbol{u}}_{t-1}]$. In general, the response generation is optimized by the standard next-token prediction objective as:

$$\mathcal{L}_{\text{gen}} = -\frac{1}{|\boldsymbol{\mathcal{D}}|} \sum_{D_t \in \boldsymbol{\mathcal{D}}} \frac{1}{n} \sum_{j=1}^{n} \log(p_\theta(w_j | w_{1:(j-1)}, \tilde{\boldsymbol{D}}_t). \quad (7)$$

3.5 Parameter-Efficient Learning

We exploit parameter-efficient fine-tuning (PEFT) techniques for training. PEFT can achieve comparable performance to standard fine-tuning (Hu et al., 2023) with higher training efficiency and avoid the catastrophic forgetting issue of LM. Specifically, we leverage the LoRA (Hu et al., 2022b) method, which incorporates low-rank weight matrices into transformer layers to adapt LM to downstream tasks by fine-tuning the injected weights only. In addition to LoRA layers, we also fine-tune the task-specific MLP layers f, g and h_{item} and the token embeddings of the four special tokens. PECRS only updates a small proportion (around 5%) of the total number of parameters in the model.

3.6 Training and Inference

The PECRS is trained in a *singe-stage* end-to-end manner by minimizing the following loss:

$$\mathcal{L} = \alpha \times \mathcal{L}_{\text{recall}} + \beta \times \mathcal{L}_{\text{rerank}} + \gamma \times \mathcal{L}_{\text{gen}}, \quad (8)$$

where α, β and γ are hyperparameters to balance the three losses. During training, we randomly sample M_{train} negative items and share them for computing the $\mathcal{L}_{\text{recall}}$ and $\mathcal{L}_{\text{rerank}}$ losses. Besides, we share the negative samples across batch elements and ensure that none of them is a positive for the dialogue contexts within a batch.

During inference, we first use PLM to encode the representations of all items in the database, which

Dataset	Unique items	Dialogues	Utterances	Recommender utterances	Rec. utt. w/o rec.	Rec. utt. w/ rec.
ReDial	6,637	11,348	139,557	73,999	31,119	42,880
INSPIRED	1,546	999	21,124	10,122	7,243	2,879

Table 1: Statistics on ReDial and INSPIRED datasets, combined over train, dev and test sets.

are reused for all dialogue contexts. Then the top-$M_{\text{inference}}$ items with highest similarities to the dialogue context query are retrieved via $f(\boldsymbol{d}_t)^\top \odot \boldsymbol{v}_j$ (see Equation (3)). We further re-rank the $M_{\text{inference}}$ items to obtain the top-1 item as the recommendation output. In practice, we set $M_{\text{train}} < M_{\text{inference}}$. We show that M yields an important trade-off between efficiency and recommendation performance both during training and inference in Section 5.2. Moreover, the predicted item is appended at the end of the dialogue context rather than the ground truth in Equation (6) in order to prompt the model for response generation. To determine whether a movie should be recommended at inference, we check whether the "[ITEM]" token is present in the generated response.

4 Experiments

4.1 Experimental Settings

Datasets. We conduct experiments on two commonly used datasets, *i.e.,* ReDial (Li et al., 2018) and INSPIRED (Hayati et al., 2020). ReDial[1] contains $11,348$ conversations ($10,006$ for train and $1,342$ for test) about movie recommendation between *seeker* and *recommender*, which is constructed through crowd-sourcing workers on Amazon Mechanical Turk. INSPIRED[2] is also about movie recommendation with smaller size of 999 (801 for train, 99 for development and 99 for test) and more flexibility given to workers. The statistics of both datasets are summarized in Table 1.

Evaluation Metrics. We follow the common practices (Yang et al., 2022; Wang et al., 2022c) to evaluate PECRS on both recommendation performance and response generation quality. For recommendation subtask, we measure recall with *Recall@K (R@K)* metric, taking $K \in \{1, 10, 50\}$. In order to assess the recommendation coverage, we also report the number of different items predicted by the model over the test set, denoted as *Unique*. ReDial and INSPIRED contain 6,637 and 1,546 unique items in total (Table 1) and 1,872 and 264 items in the test set, respectively.

We use both *reference-based* and *reference-free* metrics to evaluate response generation quality. For reference-based metrics, we adopt *ROUGE@K (RG-K)* (Lin, 2004) with $K \in \{1, 2\}$. To verify whether the model could correctly predict a movie in response when required, we inspect the presence of the "[ITEM]" token in generated responses *w.r.t.* ground truth requirement of movie prediction via *F-1* score. For reference-free metrics, we use *Perplexity (PPL)* to assess the text fluency and *Distinct@K (Dist@K)* with $K \in \{2, 3, 4\}$ to measure the diversity of generated responses.

Implementation. We choose GPT-2 (Radford et al., 2019) as the backbone LM, and experiment with two different model sizes, *i.e.,* GPT-2 small and GPT-2 medium, which enable us to compare against popular CRS approaches. Accordingly, we have **PECRS-small** and **PECRS-medium**. We highlight that PECRS is flexible and can support other choices of decoder-only LMs. We use the public pre-trained checkpoints from HuggingFace *transformers* library (Wolf et al., 2020). We set $M_{\text{train}} = 150$ for training and $M_{\text{infer}} = 700$ for inference. For ReDial, we train for 10 epochs with effective batch size 8; while for INSPIRED, we train for 20 epochs with an effective batch size of 2. Parameter optimization is performed by AdamW (Loshchilov and Hutter, 2019) with linear learning rate warmup strategy. We set maximum learning rate as $3e-5$ for PECRS-small and PECRS-medium and warmup for 1 epoch. During training, we balance losses with $\alpha = 0.15$, $\beta = 0.85$, and $\gamma = 1.0$. We cap dialogue context length at 256 tokens and response length at 64 tokens. We use checkpoint with the highest mean of R@1, R@10 and R@50 for inference. PECRS generates the response with top-k sampling, using $k = 50$. The movie item metadata is obtained from The Movie Database through *tmdbv3api* library[3].

4.2 Comparison with State-of-the-Art

The results on recommendation task are summarized in Table 2. Note that RevCore (Lu et al., 2021) and C^{2}CRS (Zhou et al., 2022) are not directly comparable to our method as they use additional movie review information. PECRS generally outperforms the baselines using KG and extra

[1] https://redialdata.github.io/website/

[2] https://github.com/sweetpeach/Inspired

[3] https://github.com/AnthonyBloomer/tmdbv3api

Model	Metadata			Model Properties			ReDial				INSPIRED			
	KG	Reviews	Description	Extra Model	PEFT	Rounds	R@1	R@10	R@50	Unique	R@1	R@10	R@50	Unique
ReDial (Li et al., 2018)	✗	✗	✗	✓	✗	3	2.4	14.0	32.0	_	_	_	_	_
KBRD (Chen et al., 2019)	✓	✗	✗	✓	✗	2	3.0	16.3	33.8	_	_	_	_	_
KGSF (Zhou et al., 2020a)	✓	✗	✗	✓	✗	3	3.9	18.3	37.8	_	_	_	_	_
KECRS (Zhang et al., 2022)	✓	✗	✗	✓	✗	2	2.3	15.7	36.6	_	_	_	_	_
BARCOR (Wang et al., 2022b)	✓	✗	✗	✓	✗	1	2.5	16.2	35.0	_	_	_	_	_
UniCRS (Wang et al., 2022c)	✓	✗	✗	✓	✓	3	5.1	22.4	42.8	_	**9.4**	**25.0**	**41.0**	_
RecInDial (Wang et al., 2022a)	✓	✗	✗	✓	✗	1	3.1	14.0	27.0	_	_	_	_	_
VRICR (Zhang et al., 2023b)	✓	✗	✗	✓	✗	3	5.7	25.1	41.6	_	_	_	_	_
RevCore (Lu et al., 2021)	✓	✓	✗	✓	✗	2	**6.1**	23.6	45.4	_	_	_	_	_
C^2-CRS (Zhou et al., 2022)	✓	✓	✗	✓	✗	2	5.3	23.3	40.7	_	_	_	_	_
MESE (Yang et al., 2022)	✗	✗	✓	✓	✗	1	5.6	**25.6**	**45.5**	_	4.8	13.5	30.1	_
PECRS-small	✗	✗	✓	✗	✓	1	4.7	20.8	40.5	463	5.4	16.1	33.3	34
PECRS-medium	✗	✗	✓	✗	✓	1	5.8	22.5	41.6	**634**	5.7	17.9	33.7	**72**

Table 2: Results of the recommendation task compared with the state-of-the-art on ReDial and INSPIRED. Results are taken from respective papers. Best numbers are in **bold**, second best underlined.

Model	Reference-based			Reference-free			
	RG-1	RG-2	F-1	PPL	Dist@2	Dist@3	Dist@4
C^2-CRS	_	_	_	_	0.163	0.291	0.417
UniCRS	_	_	_	_	0.492	0.648	0.832
RecInDial	_	_	_	_	0.518	0.624	0.598
MESE	_	_	_	12.9	**0.822**	1.152	1.313
PECRS-small	36.28	14.77	86.04	9.89	0.745	1.462	2.132
PECRS-medium	**36.86**	**15.27**	**86.36**	**8.98**	0.820	**1.552**	**2.154**

Table 3: Results of conversation task compared with the state-of-the-art on ReDial.

Aspect	MESE	PECRS-small	Tie
Fluency	28.00 (1.63)	**46.67** (5.91)	25.33 (6.24)
Relevancy	26.33 (2.62)	**46.00** (0.82)	27.67 (2.87)

Table 4: Human evaluation on 100 random ReDial test data points. We show the average scores for three human raters, with standard deviation in parenthesis.

Model	Time/ batch (s)	Rec.		Conv.	
		R@50	Unique	RG-1	Dist@2
PECRS-small	6.1	40.5	463	36.28	0.745
w/o Recall loss	6.1	19.3	21	**37.67**	0.678
w/o Rerank loss	6.1	12.2	87	36.50	0.745
w/o Generation loss	6.1	39.2	451	7.76	**11.907**
w/o Neg. sharing (batch)	8.6	39.8	291	36.40	0.747
w/o Neg. sharing (tasks)	9.1	**40.8**	434	35.98	0.727
w/o Item pooling	6.1	39.6	**530**	36.60	0.748
w/o Item head	6.1	37.9	453	36.33	0.726
w/o Metadata (just title)	**4.2**	35.8	384	36.38	0.765

Table 5: Models comparison with different modules and optimization strategies on ReDial with PECRS-small.

Removed	None	Title	Actor(s)	Director(s)	Genre(s)	Plot
R@50	**33.3**	29.8	26.9	32.5	30.5	20.7

Table 6: Effect of pruning fields of items metadata at inference on INSPIRED with PECRS-small.

model, such as KGSF (Zhou et al., 2020a) and UniCRS (Wang et al., 2022c), on both datasets. Compared to the baselines with single training stage, PECRS surpasses BARCOR (Wang et al., 2022b) and RecInDial (Wang et al., 2022a). MESE (Yang et al., 2022) also uses the item descriptions and employs two additional modules to encode items. In contrast, our PECRS is simpler and more straightforward, and it is the first approach without using either KG or supplementary module, but only relying on the pre-trained LM. PECRS-medium outperforms MESE for Recall@1 on ReDial, achieving SOTA, and largely surpasses MESE for all metrics on INSPIRED. Besides, PECRS-medium is superior to -small on all metrics, which demonstrates that fine-tuning a larger LM brings more gains thanks to its stronger representation ability.

Table 3 summarizes the results on conversation task, where PECRS achieves promising performance on both types of metrics. Both PECRS-small and -medium surpass all baselines over Dist@3 and Dist@4. Comparing PECRS-small and -medium shows that Dist@K improvements can be achieved by scaling up the backbone model. Thus, we believe that larger LMs can bring better results, and fine-tuning them with plugin style to acquire CRS capability is a promising research direction. A human evaluation (Table 4) for fluency and relevancy on ReDial test set with three volunteer graduate students with professional English proficiency confirms a preference for PECRS-small generated text over MESE outputs.

4.3 Ablation Study

We also conduct ablative experiments to analyze the architecture and optimization design of PECRS. Reported in Table 5, all the components and training strategies contribute to the performance gains on both recommendation and conversation tasks. In particular, recommendation collapses without either loss from its two-stage processes, *i.e.,* retrieval and re-ranking ; and suffers without the genera-

Model	Rec.				Conv.	
	R@1	R@10	R@50	Unique	RG-1	RG-2
PECRS-small	5.4	**16.1**	**33.3**	**34**	**29.72**	**8.26**
Llama-2-7B-chat	**9.3**	*9.3*	*9.3*	26	19.88	2.88
Vicuna-1.5-7B	8.2	*8.2*	*8.2*	23	21.18	3.50

Table 7: Comparison between PECRS-small and two popular LLMs in zero-shot on INSPIRED test set.

Decoding Strategy	Reference-based		Reference-free		
	RG-1	RG-2	Dist@2	Dist@3	Dist@4
Greedy decoding	38.54	16.25	0.208	0.311	0.390
Beam search	38.23	16.83	0.235	0.353	0.444
Diverse beam search (diversity=0.5)	39.94	17.30	0.190	0.287	0.361
Diverse beam search (diversity=1.0)	**40.29**	**17.40**	0.179	0.264	0.320
Diverse beam search (diversity=1.5)	40.07	17.23	0.172	0.246	0.290
Top-k sampling (k=25)	33.54	14.40	0.593	1.177	1.806
Top-k sampling (k=50)	33.37	14.17	**0.647**	1.300	1.989
Top-k sampling (k=75)	33.48	14.15	0.644	**1.303**	**1.992**
Nucleus sampling (p=0.90)	36.35	16.04	0.329	0.555	0.760
Nucleus sampling (p=0.95)	36.44	16.02	0.351	0.594	0.804
Nucleus sampling (p=0.99)	36.60	16.07	0.352	0.593	0.809

Table 8: The conversation performance of PECRS-small with different decoding strategies on ReDial. Except *Greedy decoding*, all other techniques use a beam width of 4.

tion loss. Sharing negative samples across batch elements and tasks leads to significant improvements on training efficiency and marginal gains on recommendation performance.

In Table 6, we conduct a further ablation on the textual fields within items description. We observe that every field contributes to the recommendation performance, especially the plot. This suggests that richer metadata would yield even more recall gains.

4.4 Comparison with Large Language Models

Lastly, we compare our fine-tuning approach with Large Language Models (LLMs). Instruction-tuned LLMs have brought a seismic shift in NLP recently, due to their ability to seamlessly conduct many tasks in a zero-shot fashion through prompts, by-passing the need for task-specific supervised fine-tuning (Sanh et al., 2021; Wei et al., 2021; Ouyang et al., 2022), including in recommender systems (Hou et al., 2023).

We use two popular LLMs: Llama-2-7B-chat[4] (Touvron et al., 2023b), and Vicuna-1.5-7B[5] (Chiang et al., 2023). For each model, we condition on the context, and prompt the LLM to predict the Recommender response, which should include a movie name. We infer in *bfloat16*, decode with greedy decoding, and check if the ground-truth movie name is included in the generated response. As seen in Table 7, the conversational recommendation capability of LLMs in zero-shot is very promising, as they outperform PECRS-small in Recall@1 on INSPIRED. However, due to the lack of a dedicated recommendation module, LLMs used in this fashion cannot suggest a full list of items, hence their recall plateaus at the Recall@1 value. They also tend to recommend fewer different movies (lower **Unique**). Exploring the ranking of a larger list of recommended items with LLMs is a promising future research avenue.

[4]https://huggingface.co/meta-llama/Llama-2-7b-chat-hf
[5]https://huggingface.co/lmsys/vicuna-7b-v1.5

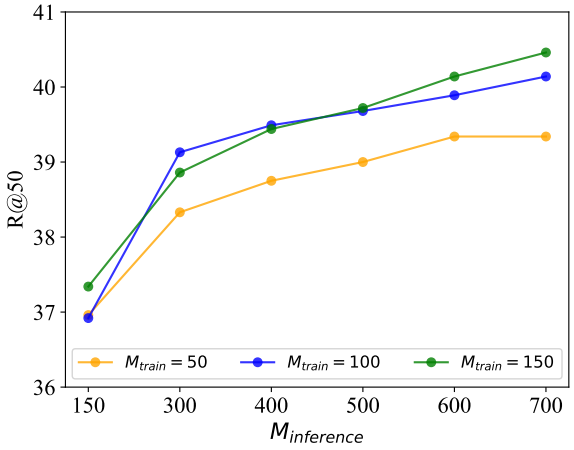

Figure 3: The R@50 results of PECRS-small using the different M_{train} and $M_{\text{inference}}$ pairs on ReDial dataset.

5 Analysis

In this section, we provide more detailed insights about the behavior of PECRS.

5.1 Conversation Evaluation

We first study the effects of different LM's decoding strategies on conversational performance over Dist@K metric. Specifically, we analyze the greedy decoding, beam search, diverse beam search (Vijayakumar et al., 2018), top-k sampling (Fan et al., 2018) and nucleus sampling (Holtzman et al., 2020) strategies on PECRS-small. Reported in Table 8, reference-based metrics (RG-K) show much less variance on different decoding strategies compared to the reference-free metrics (Dist@K). Meanwhile, the correlation between reference-based and reference-free metrics is weak under different decoding strategies. Moreover, PECRS without training for generation can achieve 11.907 on Dist@2 metric (see *w/o Generation loss* in Table 5), but merely 7.76 on RG-1 metric. *This observation implies that Dist@K metrics are not reliable to evaluate the quality of response generation*. Since Dist@K metrics have become the most popular choice in evaluating conversation

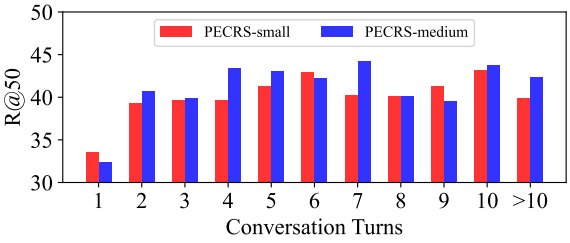

Figure 4: R@50 of PECRS on ReDial per number of conversation turns prior to the CRS response.

performance of CRS (Zhou et al., 2022; Wang et al., 2022c; Yang et al., 2022), we advocate for applying other metrics, in particular reference-based metrics including n-gram overlap like ROUGE or semantic similarity like BERTScore (Zhang et al., 2019), to provide more accurate evaluation on the response generation of CRS.

5.2 Negative Sampling

Now we analyze how the hyper-parameters of negative sampling, *i.e.,* M_{train} and $M_{\text{inference}}$, affect the recommendation performance. Figure 3 illustrates the results of different choices of M_{train} and $M_{\text{inference}}$ pairs. In general, M_{train} and $M_{\text{inference}}$ have significant impacts on the recommendation performance, and larger M_{train} and $M_{\text{inference}}$ lead to better results. However, increasing M will reduce the training and inference efficiency. Thus, there is a trade-off between efficiency and recommendation performance for the selection of M.

5.3 Conversation Turns

Lastly, we investigate how robust is PECRS with regards to the richness of dialogue context. In Figure 4, we group data points by number of utterances happening before the CRS response. We observe that PECRS performs well in recommendation for a wide range of context length, with only a moderate drop when there is only one prior utterance.

6 Conclusion

In this work, we formulate conversational recommendation as a language processing task and propose a unified parameter-efficient CRS (PECRS) framework to solve it in a single-stage end-to-end manner. PECRS effectively addresses the inferior training efficiency via parameter-efficient fine-tuning techniques and semantic misalignment issues via joint conversation and recommendation modeling. Through experiments, we show that PECRS achieves performance competitive with SOTA on both recommendation and response generation on benchmark datasets. Moreover, for response evaluation, we reveal the commonly used Dist@K metrics are not reliable, and advocate for reference-based metrics (e.g ROUGE) for more accurate evaluation. Generally, we show that it is promising to explore unified framework for CRS under the natural language paradigm via language model and rich textual items data.

Limitations

Our work adheres to standard practices for dataset construction and model evaluation. However, we acknowledge three limitations: (1) Recommender utterances containing multiple items are separated into individual data points, which is sub-optimal as the model may only be accurate for the top-ranked item in each data point. (2) If we train PECRS to predict multiple items within the same utterance, it is challenging to compare with current methods, as they do not make simultaneous predictions. (3) All items mentioned by the recommender are considered recommendations, although some may be references to previous discussions or express dislikes rather than actual recommendations.

The maximum context length for the backbone LM is another limitation. We have demonstrated that increasing $M_{\text{inference}}$ yields better recommendation performance (ref. Section 5.2). However, we are constrained by the maximum input length of 1024 for GPT-2, which limits the candidate set size after concatenating with dialogue context. The potential extensions may involve performing inference with multiple forward passes to score batches of $M_{\text{inference}}$ items, or using a backbone that supports longer input lengths, albeit at a higher computational cost. We only experiment with relatively small backbone, *i.e.,* GPT2-small and -medium, due to resource limitation. However, PECRS is flexible and can be seamlessly applied to larger backbones like LLaMA (Touvron et al., 2023a).

References

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. page 722–735. Springer-Verlag.

Qibin Chen, Junyang Lin, Yichang Zhang, Ming Ding, Yukuo Cen, Hongxia Yang, and Jie Tang. 2019. Towards knowledge-based recommender dialog system. In *Proceedings of the 2019 Conference on Empirical*

Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 1803–1813, Hong Kong, China. Association for Computational Linguistics.

Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. 2023. Vision transformer adapter for dense predictions. In *The Eleventh International Conference on Learning Representations*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*.

Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. 2016. Towards conversational recommender systems. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 815–824. Association for Computing Machinery.

Yang Deng, Wenxuan Zhang, Weiwen Xu, Wenqiang Lei, Tat-Seng Chua, and Wai Lam. 2023. A unified multi-task learning framework for multi-goal conversational recommender systems. *ACM Trans. Inf. Syst.*, 41(3).

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *ArXiv*, abs/2305.14314.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898. Association for Computational Linguistics.

Junchen Fu, Fajie Yuan, Yu Song, Zheng Yuan, Mingyue Cheng, Shenghui Cheng, Jiaqi Zhang, Jie Wang, and Yunzhu Pan. 2023. Exploring adapter-based transfer learning for recommender systems: Empirical studies and practical insights. *ArXiv*, abs/2305.15036.

Chongming Gao, Wenqiang Lei, Xiangnan He, M. de Rijke, and Tat-Seng Chua. 2021. Advances and challenges in conversational recommender systems: A survey. *AI Open*, 2:100–126.

Michael U. Gutmann and Aapo Hyvärinen. 2012. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *J. Mach. Learn. Res.*, 13:307–361.

Shirley Anugrah Hayati, Dongyeop Kang, Qingxiaoyang Zhu, Weiyan Shi, and Zhou Yu. 2020. INSPIRED: Toward sociable recommendation dialog systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8142–8152, Online. Association for Computational Linguistics.

Xuehai He, Chunyuan Li, Pengchuan Zhang, Jianwei Yang, and Xin Eric Wang. 2022. Parameter-efficient model adaptation for vision transformers. *ArXiv*, abs/2203.16329.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian McAuley, and Wayne Xin Zhao. 2023. Large language models are zero-shot rankers for recommender systems. *arXiv preprint arXiv:2305.08845*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 2790–2799.

Chenhao Hu, Shuhua Huang, Yansen Zhang, and Yubao Liu. 2022a. Learning to infer user implicit preference in conversational recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 256–266. Association for Computing Machinery.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022b. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Zhiqiang Hu, Yihuai Lan, Lei Wang, Wanyu Xu, Ee-Peng Lim, Roy Ka-Wei Lee, Lidong Bing, Xing Xu, and Soujanya Poria. 2023. Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models. *ArXiv*, abs/2304.01933.

Dietmar Jannach, Ahtsham Manzoor, Wanling Cai, and Li Chen. 2021. A survey on conversational recommender systems. *ACM Comput. Surv.*, 54(5).

Wenqiang Lei, Xiangnan He, Yisong Miao, Qingyun Wu, Richang Hong, Min Yen Kan, and Tat Seng Chua. 2020. Estimation–action–reflection: Towards deep interaction between conversational and recommender systems. In *WSDM 2020 - Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 304–312. Association for Computing Machinery, Inc.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Raymond Li, Samira Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards deep conversational recommendations. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, page 9748–9758. Curran Associates Inc.

Shuokai Li, Ruobing Xie, Yongchun Zhu, Xiang Ao, Fuzhen Zhuang, and Qing He. 2022. User-centric conversational recommendation with multi-aspect user modeling. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 223–233. Association for Computing Machinery.

Zujie Liang, Huang Hu, Can Xu, Jian Miao, Yingying He, Yining Chen, Xiubo Geng, Fan Liang, and Daxin Jiang. 2021. Learning neural templates for recommender dialogue system. *arXiv preprint arXiv:2109.12302*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Jianghao Lin, Xinyi Dai, Yunjia Xi, Weiwen Liu, Bo Chen, Xiangyang Li, Chenxu Zhu, Huifeng Guo, Yong Yu, Ruiming Tang, and Weinan Zhang. 2023. How can recommender systems benefit from large language models: A survey. *ArXiv*, abs/2306.05817.

Zeming Liu, Haifeng Wang, Zheng-Yu Niu, Hua Wu, Wanxiang Che, and Ting Liu. 2020. Towards conversational recommendation over multi-type dialogs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1036–1049, Online. Association for Computational Linguistics.

Zeming Liu, Ding Zhou, Hao Liu, Haifeng Wang, Zheng-Yu Niu, Hua Wu, Wanxiang Che, Ting Liu, and Hui Xiong. 2023. Graph-grounded goal planning for conversational recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 35(5):4923–4939.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. *ArXiv*, abs/1711.05101.

Yu Lu, Junwei Bao, Yan Song, Zichen Ma, Shuguang Cui, Youzheng Wu, and Xiaodong He. 2021. RevCore: Review-augmented conversational recommendation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1161–1173, Online. Association for Computational Linguistics.

Wenchang Ma, Ryuichi Takanobu, and Minlie Huang. 2020. Cr-walker: Tree-structured graph reasoning and dialog acts for conversational recommendation. *arXiv preprint arXiv:2010.10333*.

Andriy Mnih and Koray Kavukcuoglu. 2013. Learning word embeddings efficiently with noise-contrastive estimation. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Andriy Mnih and Yee Whye Teh. 2012. A fast and simple algorithm for training neural probabilistic language models. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, page 419–426.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Dhanya Pramod and Prafulla Bafna. 2022. Conversational recommender systems techniques, tools, acceptance, and adoption: A state of the art review. *Expert Syst. Appl.*, 203(C).

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Xuhui Ren, Hongzhi Yin, Tong Chen, Hao Wang, Nguyen Quoc Viet Hung, Zi Huang, and Xiangliang Zhang. 2020. Crsal: Conversational recommender systems with adversarial learning. *ACM Trans. Inf. Syst.*, 38(4).

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2021. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*.

Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *The Semantic Web*, pages 593–607. Springer International Publishing.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Yueming Sun and Yi Zhang. 2018. Conversational recommender system. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, page 235–244. Association for Computing Machinery.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aur'elien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *ArXiv*, abs/2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R. Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2018. Diverse beam search: Decoding diverse solutions from neural sequence models. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Lingzhi Wang, Huang Hu, Lei Sha, Can Xu, Daxin Jiang, and Kam-Fai Wong. 2022a. RecInDial: A unified framework for conversational recommendation with pretrained language models. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 489–500, Online only. Association for Computational Linguistics.

Ting-Chun Wang, Shang-Yu Su, and Yun-Nung Chen. 2022b. Barcor: Towards a unified framework for conversational recommendation systems. *ArXiv*, abs/2203.14257.

Xiaolei Wang, Kun Zhou, Ji-Rong Wen, and Wayne Xin Zhao. 2022c. Towards unified conversational recommender systems via knowledge-enhanced prompt learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, page 1929–1937. Association for Computing Machinery.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Likang Wu, Zhilan Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, Hui Xiong, and Enhong Chen. 2023. A survey on large language models for recommendation. *ArXiv*, abs/2305.19860.

Bowen Yang, Cong Han, Yu Li, Lei Zuo, and Zhou Yu. 2022. Improving conversational recommendation systems' quality with context-aware item meta-information. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 38–48, Seattle, United States. Association for Computational Linguistics.

Renrui Zhang, Jiaming Han, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, Peng Gao, and Yu Qiao. 2023a. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *ArXiv*, abs/2303.16199.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Tong Zhang, Yong Liu, Boyang Li, Peixiang Zhong, Chen Zhang, Hao Wang, and Chunyan Miao. 2022. Toward knowledge-enriched conversational recommendation systems. In *Proceedings of the 4th Workshop on NLP for Conversational AI*, pages 212–217, Dublin, Ireland. Association for Computational Linguistics.

Xiaoyu Zhang, Xin Xin, Dongdong Li, Wenxuan Liu, Pengjie Ren, Zhumin Chen, Jun Ma, and Zhaochun Ren. 2023b. Variational reasoning over incomplete knowledge graphs for conversational recommendation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. ACM.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. DIALOGPT : Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278, Online. Association for Computational Linguistics.

Jinfeng Zhou, Bo Wang, Ruifang He, and Yuexian Hou. 2021. CRFR: Improving conversational recommender systems via flexible fragments reasoning on knowledge graphs. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4324–4334, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Kun Zhou, Wayne Xin Zhao, Shuqing Bian, Yuanhang Zhou, Ji-Rong Wen, and Jingsong Yu. 2020a. Improving conversational recommender systems via knowledge graph based semantic fusion. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 1006–1014. Association for Computing Machinery.

Kun Zhou, Yuanhang Zhou, Wayne Xin Zhao, Xiaoke Wang, and Ji-Rong Wen. 2020b. Towards topic-guided conversational recommender system. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4128–4139, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Yuanhang Zhou, Kun Zhou, Wayne Xin Zhao, Cheng Wang, Peng Jiang, and He Hu. 2022. C²-crs: Coarse-to-fine contrastive learning for conversational recommender system. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, page 1488–1496. Association for Computing Machinery.

Jie Zou, Yifan Chen, and Evangelos Kanoulas. 2020. Towards question-based recommender systems. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 881–890. Association for Computing Machinery.

A System Outputs

We show an example from PECRS-medium on the INSPIRED dataset, in the same format as Figure 1.

Dialogue context:
Recommender: Hello.
Seeker: Hello.
Recommender: What can I help you with?
Seeker: Can you help recommend a movie trailer to watch?
Recommender: I would recommend horror movies, but I would like to know what movies you like first.
Seeker: Oh, I love action movies.
Recommender: The types with very serious actors (Stallone, etc) or, like Guardians of the Galaxy?
Seeker: I like both, but I'm looking for something super hero related, like Guardians of the Galaxy.
Recommender: Oh! I have a few ideas of what you have in mind. Are comic books characters interesting at all?
Seeker: Yes! I love comic book characters, like Spider man, Ironman and Captain America.

PECRS: Avengers: Endgame is a good one. Avengers: Endgame has a lot of action in it. I think you would like it.

Recommender: Would you like to watch this trailer for The Avengers: Endgame?

Figure 5: An example of dialogue from INSPIRED (Hayati et al., 2020), where blue color denotes the movie items.

Genre	Items (%)	Test set Recommendation (%)	Correctly Predicted (%)
Comedy	24.48	23.74	46.37
Action	21.88	30.67	57.65
Drama	17.89	13.74	32.93
Adventure	6.18	6.02	29.72
Horror	5.82	7.94	46.95
Crime	5.66	4.50	20.56
Animation	5.40	6.71	62.38
Biography	4.11	2.50	12.61
Documentary	3.27	0.76	22.22
Fantasy	1.00	0.61	6.90
Thriller	0.67	0.46	31.82
Family	0.62	0.38	0.00
Mystery	0.47	0.57	7.41
Romance	0.46	0.04	0.00
TV	0.43	0.08	0.00
Music	0.26	0.20	0.00
Western	0.25	0.04	0.00
Science	0.23	0.13	0.00
Short	0.23	0.11	0.00
War	0.21	0.11	0.00
Sci-fi	0.20	0.06	0.00
History	0.11	0.00	–
Musical	0.10	0.23	9.09
Film-noir	0.05	0.08	0.00
Adult	0.02	0.02	0.00

Table 9: Accuracy w.r.t genre prediction on ReDial test set broken down by movie genre.

B Genre Analysis

In this section, we conduct a fine-grained analysis of PECRS top-1 recommendation. We investigate how the model performs on several types of items. To categorize items, we use the first genre tag in the Genre(s) field in the items metadata, yielding a partition of the movies set into 25 unique genres for ReDial, 22 genres for INSPIRED. We report the fraction of data points where the model outputs a top-1 movie of the correct genre per genre on ReDial and INSPIRED in Table 9 and Table 10, respectively.

As we can see, there is wide variance in genres accuracy. Among wrong movie predictions, PECRS-medium outputs the correct genre 41.20% times on ReDial and 30.04% on INSPIRED. Random performance would yield 16.26% and 19.39% accuracy, respectively. The performance is much higher on highly represented genres such as *Comedy*, *Action*, or *Horror*, where it can surpass a ratio of correctly predicted genre of 50%, but quickly falls to 0 for rare genres such as *Romance*. Future work may focus on better handling the long tail distribution in items variety, for instance through data augmentation techniques crafted for rare genres movies.

C Packages

Our framework was implemented in Python 3.8.0. We used the following Python package versions to

Genre	Items (%)	Test set Recommendation (%)	Correctly Predicted (%)
Action	24.01	36.20	50.50
Comedy	22.66	17.92	52.00
Drama	17.67	13.98	10.26
Horror	7.45	9.68	14.81
Adventure	4.86	2.15	66.67
Animation	4.86	4.66	7.69
Crime	4.86	6.09	23.53
Biography	4.50	2.15	0.00
Documentary	3.20	1.79	0.00
Thriller	0.92	0.36	0.00
Fantasy	0.86	0.36	0.00
Romance	0.80	0.36	0.00
Mystery	0.62	0.00	_
TV	0.37	0.00	_
Short	0.37	0.00	_
Science	0.31	0.72	0.00
Music	0.25	0.00	_
Sci-fi	0.25	0.36	0.00
War	0.12	0.00	_
Western	0.12	0.00	_
Musical	0.06	0.00	_
Reality-TV	0.06	0.00	_

Table 10: Accuracy w.r.t genre prediction on INSPIRED test set broken down by movie genre.

conduct all experiments:

- numpy 1.24.3
- torch 1.9.1
- transformers 4.33.2
- rouge-score 0.1.2
- nltk 3.8.1
- peft 0.1.0
- spacy 3.6.0

All packages and datasets used are freely available and open-source, and were used for research purpose only. We refer to the specific papers for more details on the use of each dataset.