

# Unveiling User Preferences: A Knowledge Graph and LLM-Driven Approach for Conversational Recommendation

Zhangchi Qiu  
zhangchi.qiu@griffithuni.edu.au  
Griffith University  
Gold Coast, QLD, Australia

Shirui Pan  
s.pan@griffith.edu.au  
Griffith University  
Gold Coast, QLD, Australia

Linhao Luo  
Linhao.Luo@monash.edu  
Monash University  
Melbourne, VIC, Australia

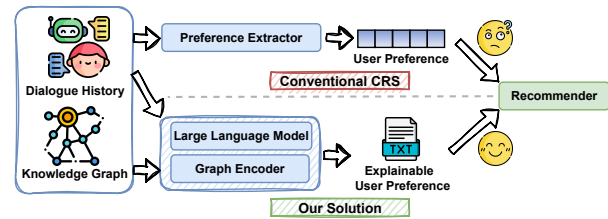
Alan Wee-Chung Liew\*  
a.liew@griffith.edu.au  
Griffith University  
Gold Coast, QLD, Australia

## Abstract

Conversational Recommender Systems (CRSs) aim to provide personalized recommendations through dynamically capturing user preferences in interactive conversations. Conventional CRSs often extract user preferences as hidden representations, which are criticized for their lack of interpretability. This diminishes the transparency and trustworthiness of the recommendation process. Recent works have explored combining the impressive capabilities of Large Language Models (LLMs) with the domain-specific knowledge of Knowledge Graphs (KGs) to generate human-understandable recommendation explanations. Despite these efforts, the integration of LLMs and KGs for CRSs remains challenging due to the modality gap between unstructured dialogues and structured KGs. Moreover, LLMs pre-trained on large-scale corpora may not be well-suited for analyzing user preferences, which require domain-specific knowledge. In this paper, we propose COMPASS (**C**ompact **P**reference **A**nalyzer and **S**ummarization **S**ystem), a plug-and-play framework that synergizes LLMs and KGs to unveil user preferences, enhancing the performance and explainability of existing CRSs. To address integration challenges, COMPASS employs a two-stage training approach: first, it bridges the gap between the structured KG and natural language through an innovative graph entity captioning pre-training mechanism. This enables the LLM to transform KG entities into concise natural language descriptions, allowing them to comprehend domain-specific knowledge. Following, COMPASS optimizes user preference modeling via knowledge-aware instruction fine-tuning, where the LLM learns to reason and summarize user preferences from both dialogue histories and KG-augmented context. This enables COMPASS to perform knowledge-aware reasoning and generate comprehensive and interpretable user preferences that can seamlessly integrate with existing CRS models for improving recommendation performance and explainability. Our experiments on benchmark datasets demonstrate the effectiveness of COMPASS in improving various CRS models.

## CCS Concepts

• **Information systems** → **Recommender systems**.



**Figure 1: Conventional CRSs lack explainability in user preference modeling by extracting hidden representation, while our approach enhances transparency by generating interpretable user preference in text.**

## Keywords

Conversational Recommender System; Large Language Model; Knowledge Graph

## 1 Introduction

Providing personalized, context-aware recommendations that align with users' changing preferences and situational needs is a critical challenge across various domains such as e-commerce, streaming platforms, and other online services. In recent years, conversational recommender systems (CRSs) have emerged as a promising approach, harnessing the power of natural language interactions to unravel user preferences [12, 18]. By engaging the user in interactive conversations, CRSs enable a better understanding of the user's evolving interests and guide them toward products, services, and information that best meet their immediate requirements [4, 24, 26]. Despite the success, CRSs face unique challenges in precisely modeling user preferences from the semantically rich and dynamically evolving dialogues, due to the challenge of natural language understanding [18]. CRSs must capture user's intent and interest from their natural language inputs, which are often ambiguous and context-dependent [24, 48]. Moreover, as the conversation progresses, CRSs need to continuously update and refine their understanding of user preferences in real-time [11, 34]. As the user reveals more about their preferences, CRSs must dynamically adapt their preference modeling to integrate the user's immediate interests expressed in recent interactions with their overall preferences developed over time.

\*Corresponding author.

Conventional CRSs mainly rely on item-centric approaches [3, 48], focusing on items mentioned during the conversation to model user preferences. Recent studies have incorporated pre-trained language models (PLMs) to enhance both natural language understanding and user preference modeling [37, 41]. However, these methods often fail to infer implicit preferences or reason about underlying motivations beyond explicit item mentions, leading to a superficial understanding of user intent. Moreover, they typically represent user preferences as hidden embeddings, leaving it unclear what specific preference is being considered when making a recommendation, as shown at the top of Figure 1. This ambiguity makes it challenging to verify the underlying reasons and results behind the recommendations, hindering the transparency and accountability of the recommender system.

Recent advancements in Large Language Models (LLMs), which significantly scale up PLMs in terms of parameter size and pre-training data, have demonstrated exceptional capabilities in both natural language processing [8, 20, 47] and reasoning [42, 43]. These models have shown promise in generating natural language explanations, particularly in recommendation systems where explainability is critical. Recent works like X-REC [28] and SLIM [42] have demonstrated the LLM’s ability to reason over user historical behaviors and generate interpretable recommendation explanations, enhancing both explainability and performance. Nevertheless, these methods only focus on the traditional recommendation settings, where user preferences are static and inferred from their historical behaviors. CRSs face unique challenges, necessitating real-time preference modeling and recommendations that adapt to users’ changing preferences during ongoing conversations. Although LLMs have shown promise in CRS tasks such as evaluation [40, 50], zero-shot recommendation capabilities [16], and task planning [9, 10, 23], the role of explainable user preferences in LLM-based CRSs still remains unexplored.

While LLMs have shown promise in CRS tasks, they still face limitations in incorporating domain-specific knowledge and keeping up-to-date with item information, which are essential for accurate preference modeling and recommendations. Knowledge Graphs (KGs) have proven effective in addressing these limitations [3, 41, 48], as they provide a rich, structured context of items and their relationships, offering domain-specific insights that enable more accurate and explainable recommendations. Despite their demonstrated potential, integrating the structured knowledge of KGs with the reasoning and language capabilities of LLMs presents a new set of challenges:

- **Modality Gap (Challenge 1):** There exists a significant modality gap between KGs and LLMs, which hinders the LLM’s ability to understand and interpret KG information. While LLMs process sequences of tokens that represent natural language, KGs represent information in a structured, graph-based format. This difference makes it difficult for LLMs to directly interpret the entities and relationships encoded in KGs, limiting their ability to leverage domain knowledge for preference modeling.
- **Cross-Modal Reasoning (Challenge 2):** Effectively reasoning over both KG information and conversation data to infer user preferences is a complex task. LLMs, despite their

strong natural language processing capabilities, are not inherently designed to perform this cross-modal reasoning. Therefore, they face difficulties in analyzing and synthesizing insights from graph-structured knowledge alongside dialogue history. This limitation hinders their ability to identify relevant patterns across both sources and to perform the knowledge-aware reasoning necessary for comprehensive user preference modeling, in which both domain knowledge and user interactions are essential.

To address the above challenges, we propose **Compact Preference Analyzer and Summarization System (COMPASS)**, a novel framework that synergizes LLMs and KGs to unveil user preferences, improving both recommendation performance and the explainability of existing CRS. COMPASS directly tackles the limitations of current CRSs and the challenges of integrating LLMs with KGs through a two-stage process. First, we introduce a *graph entity captioning pre-training* mechanism that transforms KG structures into natural language descriptions. This allows the LLM to comprehend domain-specific information and bridge the **modality gap (Challenge 1)**. We leverage a Graph Neural Network (GNN) to capture structural information from the KG and represent it as entity embeddings. These embeddings are then input into the LLM to produce textual descriptions of the entities along with relevant details from their neighbors. In this way, we align graph-structured knowledge with natural language, enabling the LLM to better interpret KG information. Building upon this alignment, COMPASS employs *knowledge-aware instruction fine-tuning* to improve the LLM’s ability to reason about user preferences from dialogue histories and KG-augmented contexts. These KG-augmented contexts consist of relevant entity information and relationships extracted from the KG, providing a rich background for inference beyond the conversation history alone. Through carefully designed instructions, we enhance the LLM’s capability to perform **cross-modal reasoning (Challenge 2)** by analyzing conversation history and cross-referencing with KG-augmented information. This instruction tuning process enhances the LLM’s ability to extract explicit mentions, infer implicit interests, and reason about preferences in relation to various item attributes. Consequently, as shown in the bottom of Figure 1, COMPASS generates comprehensive and interpretable user preferences in text that capture both overall preferences and current interests. To leverage these insights, we introduce an adaptive gating mechanism that integrates summarized preferences into existing CRS models, boosting recommendation performance and explainability without requiring architectural changes. Our main contributions can be summarized as follows:

- **New framework.** We propose COMPASS, a novel framework for enhancing user preference modeling in CRSs. To the best of our knowledge, this is the first work to leverage LLMs and KGs for explainable preference generation in CRSs.
- **Effective cross-model reasoning and explanation.** We develop a two-stage process that enables the LLM to perform cross-modal reasoning over KGs and conversations, generating explainable user preference summaries. This approach moves beyond abstract vector representations to provide clear, human-readable user preferences.

- **Flexible Plug-in.** COMPASS generates user preference summaries that are compatible with existing CRS architectures without requiring modifications to the system, improving both recommendation performance and explainability.

## 2 Preliminaries

**Conversational Recommendation.** Let  $i$  denote a candidate item from the set of items  $\mathcal{I}$ , and let  $w$  denote a word in the vocabulary  $\mathcal{V}$ . A dialogue  $\mathcal{D}$  between a user and the recommender system consists of a sequence of utterances  $\mathcal{D} = [u_1, \dots, u_T]$ , where  $u_t = [w_1^t, \dots, w_m^t]$  is the  $t$ -th utterance composed of  $m$  words, and  $T$  is the maximum number of turns in the dialogue. As the conversation progresses, the dialogue history up to the turn  $t$  is denoted as  $H_t = [u_1 : u_t]$ , where  $[u_1 : u_t]$  signifies the chronological sequence of utterances from the first to the  $t$ -th turn. The CRS estimates the user’s preferences based on  $H_t$ , then recommend  $K$  items from  $\mathcal{I}$ , which are used to generate the next utterance  $u_{t+1}$ . Note that  $\mathcal{I}_t$  can be empty when no recommendation is needed. In such cases, the CRS may raise a clarification question or generate a casual conversation response.

**Knowledge Graph.** A knowledge graph is defined as  $\mathcal{G} = (\mathcal{E}, \mathcal{A}, \mathcal{X})$  where  $\mathcal{E}, \mathcal{R}$  represents the set of entities and relation types in the graph, respectively.  $\mathcal{A}$  is the adjacency matrix capturing the relationships between entities and  $\mathcal{X}$  represents the textual descriptions of each entity. For each entity  $e \in \mathcal{E}$ , its description is denoted as  $x_e \in \mathcal{X}$ , where  $x_e = [w_1^e, \dots, w_k^e]$ , and  $w_k^e$  represents the  $k$ -th word in the entity description. The entity set  $\mathcal{E}$  encompasses candidate items  $\mathcal{I}$  (e.g., movies) and non-item entities that represent item attributes (e.g., actors, genres, keywords). Formally,  $\mathcal{I} \subseteq \mathcal{E}$ .

**Explainable User Preferences and Recommendations.** Explainable user preferences are crucial for enhancing the transparency and effectiveness of CRSs. Our goal is to generate clear, human-understandable textual user preference summaries that provide insights for both recommendation and explanation. Specifically, for a given dialogue history  $H_t$ , we define the generation of user preference summaries as:

$$\mathcal{P}_t = f(I_p, H_t, \mathcal{E}_t^m, \mathcal{G}), \quad (1)$$

where  $f$  represents a model that reasons over the dialogue history  $H_t$ , the mentioned entities  $\mathcal{E}_t^m \subseteq \mathcal{E}$ , and their associated information from the knowledge graph  $\mathcal{G}$ .  $I_p$  is denoted as instruction prompts. The resulting  $\mathcal{P}_t$  represents the textual explainable user preference summary at the  $t$ -th turn. The summary  $\mathcal{P}_t$  is then encoded using a preference encoder  $g(\cdot)$  and integrated into the base CRS model  $f_{\text{crs}}$ , which uses it to adjust its recommendation strategy. Depending on the model, additional inputs, such as KG information or dialogue history, may also be used. Formally, the recommendation step is represented as:

$$\mathcal{I}_t = f_{\text{crs}}(g(\mathcal{P}_t), H_t, \mathcal{G}), \quad (2)$$

where  $\mathcal{I}_t$  represents the recommended items at turn  $t$ , and  $g(\mathcal{P}_t)$  denotes the encoded preference summary.

## 3 Methodology

We present an overview of COMPASS, followed by detailed descriptions of each component and the training process.

### 3.1 Overview

The primary goal of COMPASS is to synergize the reasoning capabilities of LLMs with the structured knowledge from KGs to analyze and summarize user preferences. COMPASS comprises three core components: (1) *Graph encoder* processes a domain-specific KG, capturing complex relationships between items to augment user preference modeling. (2) *Graph-to-Text adapter* aims to bridge the modality gap between the graph encoder and the LLM, enabling the LLM to comprehend the graph structure and conduct reasoning. (3) *Large Language Model (LLM)* leverages the powerful reasoning and generative capabilities of advanced language models to generate interpretable user preference summaries.

To integrate the components cohesively, we employ a two-stage training process: (1) *Graph entity captioning* aligns KG structures with natural language representations, creating a shared semantic space for the LLM to comprehend and reason with domain-specific knowledge effectively. (2) *Knowledge-aware instruction fine-tuning* optimizes the LLM for cross-modal reasoning, allowing it to generate comprehensive user preference summaries by synthesizing information from dialogue history and KG-augmented context. Once trained, COMPASS can be integrated with existing CRS models to improve their recommendation performance and explainability by generating user preferences and incorporating them into the recommendation process with an adaptive gating mechanism. Figure 2 illustrates COMPASS’s architecture and training process.

### 3.2 Model Architecture

**3.2.1 Graph Encoder.** In COMPASS, the KG is a crucial source of domain-specific information that provides extra context for understanding attributes and relationships of items. To efficiently convert the structured knowledge in the KG into a format understandable by the LLM for preference analysis and summarization, we utilize a Relational Graph Convolutional Network (R-GCN) [33] to capture the complex graph structure and generate entity embeddings. The R-GCN is well-suited for modeling KGs due to its ability to handle multi-relational data and capture higher-order dependencies between entities.

To initialize the entity embeddings, we leverage the textual descriptions  $\mathcal{X}$  associated with the entities in the KG. Specifically, for each entity  $e \in \mathcal{E}$ , we encode its description  $x_e$  using a pre-trained language model (PLM). This provides a rich semantic foundation for the graph learning process. The R-GCN then captures both entity-level information and the overall graph structure through iterative message passing, which is particularly important for understanding the relationships between items and attributes. Formally, the representation of an entity  $e$  at the  $l$ -th layer is calculated as follows:

$$\mathbf{h}_e^{(0)} = \text{PLM}(x_e), \quad (3)$$

$$\mathbf{h}_e^{(l+1)} = \sigma \left( \sum_{r \in \mathcal{R}} \sum_{e' \in \mathcal{E}_e^r} \frac{1}{Z_{e,r}} \mathbf{W}_r^{(l)} \mathbf{h}_{e'}^{(l)} + \mathbf{W}_e^{(l)} \mathbf{h}_e^{(l)} \right), \quad (4)$$

where  $\mathbf{h}_e^l$  is the embedding of entity  $e$  at the  $l$ -th layer,  $\mathcal{E}_e^r$  is the set of neighboring entities connected to  $e$  through relation  $r$ ,  $\mathbf{W}_r^{(l)}$  and  $\mathbf{W}_e^{(l)}$  are learnable weight matrices,  $Z_{e,r}$  is a normalization factor, and  $\sigma$  is an activation function.

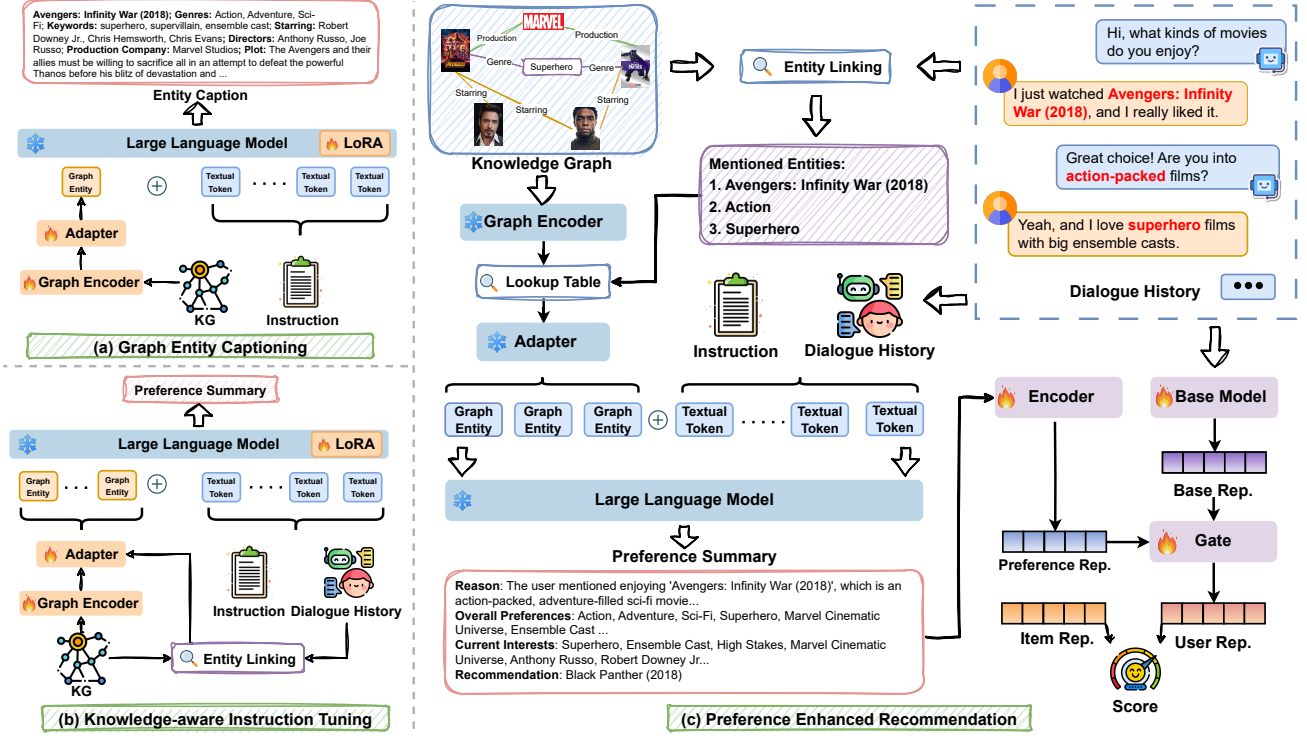


Figure 2: The overall framework of our COMPASS. COMPASS consists of three components: the graph encoder, the adapter, and the LLM. The adapter aligns the knowledge graph to the LLM. COMPASS follows a two-stage training paradigm - (a) *Graph Entity Captioning* and (b) *Knowledge-aware Instruction Tuning*. Once trained, COMPASS can be easily integrated with existing CRS models for (c) *Preference Enhanced Recommendation*.

**3.2.2 Graph-to-Text Adapter.** The entity embeddings produced by the graph encoder, while rich in structural information, exist in a different representational space from the LLM’s textual token representations. This makes it challenging for the LLM to effectively reason with the KG-augmented context to generate user preferences. To bridge this semantic gap and enable effective knowledge integration, we introduce an adapter module that creates a mapping between graph-structured entity embeddings and the LLM’s textual domain. Specifically, the adaptation process is defined as:

$$\mathbf{h}_e^t = f_p(\mathbf{h}_e), \quad (5)$$

where  $\mathbf{h}_e^t$  is the adapted entity embedding aligned with the LLM’s semantic space,  $\mathbf{h}_e$  is the entity embedding generated from graph encoder, and  $f_p$  is the projection function implemented as a linear layer.

**3.2.3 Large Language Model (LLM).** The LLM is the main reasoning engine that generates user preferences by synthesizing information from dialogue histories and KG-augmented contexts. It captures both the explicit user preferences expressed in the dialogue and the implicit preferences inferred from the KG-augmented context for accurate preference analysis. Our framework is compatible with various state-of-the-art LLMs, allowing flexibility in model choice. In this paper, we use Llama3.1-8B [8] for its natural language understanding and generation capabilities. To adapt the chosen LLM efficiently for our task, we employ Low-Rank Adaptation (LoRA) [17] for fine-tuning.

### 3.3 Training Pipeline

**3.3.1 Graph Entity Captioning.** To obtain a Graph-to-Text adapter that effectively bridges the modality gap between the graph entity embeddings and the LLM’s semantic space, inspired by the pre-training strategies employed in vision-language models [5, 25], we introduce a graph entity captioning pretraining mechanism, as shown in Figure 2a. This process creates a strong connection between graph-structured data and natural language by generating entity-specific captions that encompass both the entity’s intrinsic information and aggregated data from its neighboring nodes, simulating a message-passing process within the graph structure.

The pre-training stage enables the LLM to interpret graph entity embeddings in a semantic context, leading to a deeper understanding of the relationships and attributes encoded within the KG, which is crucial for subsequent preference modeling. The caption generation process differentiates between item entities (e.g., movies) and non-item entities (e.g., genres, directors). For item entities, we employ a structured template that captures key attributes and relationships.

#### Example of Item Entity Caption Template

**Movie Title:** <Title>; **Genres:** <Genres>; **Keywords:** <Keywords>; **Starring:** <Actors>; **Directors:** <Directors>; **Production Company:** <Company>; **Plot:** <Plot Summary>

This template ensures comprehensive coverage of item attributes while maintaining a consistent structure across different items. For non-item entities, we adopt a more flexible approach that emphasizes the entity’s role and its connections within the KG <sup>1</sup>.

To format the training data, an input-output pair is constructed for each entity in the knowledge graph. The input consists of the adapted entity embedding  $\mathbf{h}_e^r$  from Equation 5 and a task-specific instruction prompt  $I_c$ . The output is the generated caption  $C_e$  for the entity  $e$ . This format is represented as follows.

#### Entity Captioning

**Input:** <Entity Embeddings  $\mathbf{h}_e^r$ >, <Instruction  $I_c$ >  
**Output:** <Entity Caption  $C_e$ >

In this way, the LLM learns to map the graph-structured input to natural language by reconstructing the entity caption, conditioned on the graph entity embeddings and instructions. This process is optimized by minimizing the negative log-likelihood (NLL) of the generated captions, as expressed by:

$$\mathcal{L}_{caption} = - \sum_{e \in \mathcal{E}} \log P(C_e | \mathbf{h}_e^r, I_c), \quad (6)$$

where  $\mathcal{E}$  is the set of all entities in the KG. Through the training, the LLM learns to interpret graph entity embeddings in a semantic context, leading to a deeper understanding of the relationships and attributes encoded within the KG. This understanding is essential for improving subsequent preference modeling.

**3.3.2 Knowledge-aware Instruction Tuning.** After the graph entity captioning pre-training, the LLM has gained a basic understanding of the KG structure and content. However, it has not yet been explicitly trained to utilize this knowledge for downstream tasks such as preference modeling and recommendation generation. To this end, we introduce a knowledge-aware instruction tuning, which aims to enable the LLM with abilities of reasoning across modalities, integrating information from both the dialogue history and the KG to infer user preferences and interests, as shown in Figure 2b.

The knowledge-aware tuning process employs a carefully crafted instruction prompt  $I_p$  to guide the LLM in synthesizing and reasoning over inputs from multiple sources. Given a dialogue history  $H_t$  and the entities  $\mathcal{E}_t^m$  mentioned within it, the process retrieves the embeddings  $\mathbf{E}_t$  of these entities from Equation 5. These embeddings, along with the full dialogue history, serve as inputs for the LLM. The instruction prompt  $I_p$  directs the LLM to analyze both the KG-derived entity information and the dialogue history to generate a user preference summary  $\mathcal{P}_t$ .

This prompt follows a coarse-to-fine structure containing four key steps: (1) reasoning, providing transparency in the model’s decision-making; (2) overall preferences, offering a broad view of the user’s tastes; (3) current interests, capturing recent and specific preferences to guide subsequent recommendations; and (4) recommendation, leveraging the LLM’s reasoning capabilities to suggest relevant items aligned with user preferences, guiding downstream

CRS models. This structured prompt ensures that the model captures both long-term preferences and immediate interests. To generate ground-truth preference summaries, we utilize an advanced LLM (e.g., ChatGPT) that performs cross-modal reasoning, integrating complete dialogue histories with structured metadata of mentioned items from the KG. The detailed process and instruction templates for this ground-truth generation are provided in Appendix B.3.

The instruction tuning process can be represented in the following format:

#### Knowledge-aware Instruction Tuning

**Input:** <Mentioned Entities Embeddings  $\mathbf{E}_t$ >, <Instruction  $I_p$ >, <Dialogue History  $H_t$ >  
**Output:** <Preference Summary  $\mathcal{P}_t$ >

A detailed example of the preference summary can be found in Table 5 of Appendix C.2.

This instruction tuning process is optimized by minimizing the NLL of the generated preference summaries, as expressed by:

$$\mathcal{L}_{preference} = - \sum_{\mathcal{D} \in \mathcal{C}} \sum_{H_t \in \mathcal{D}} \log P(\mathcal{P}_t | \mathbf{E}_t, I_p, H_t), \quad (7)$$

where  $\mathcal{C}$  represents the set of all dialogues in the training data. Through this process, COMPASS learns to synthesize information from dialogue history and KG-derived entity embeddings, enabling it to generate comprehensive and interpretable preference summaries.

## 3.4 Integration with Existing CRS Models

To enhance the existing CRS models with the user preference summaries generated by COMPASS, we propose a two-step integration process: (1) transforming the natural language preference summaries into a format compatible with CRS models, and (2) incorporating these transformed preferences to enhance the base CRS model’s recommendation performance via an adaptive gating mechanism. Note that COMPASS remains frozen during this process.

**3.4.1 Preference Representation.** Traditional CRSs are not designed to directly utilize natural language preference summaries. Therefore, we explore two methods to encode these summaries into a format suitable for existing CRS architectures:

1) PLM-based representation: This method leverages a PLM to extract rich semantic representations from text. The PLM is adaptable and can be implemented as either a frozen or trainable text encoding model. In our implementation, we employ BERT [6] to encode the user preference summary as follows:

$$\mathbf{s}_c^{\text{text}} = \text{PLM}(\mathcal{P}), \quad (8)$$

where  $\mathbf{s}_c^{\text{text}}$  is the encoded preference from text preference  $\mathcal{P}$ , specifically the [CLS] token embedding. The contextual understanding of language models enables  $\mathbf{s}_c^{\text{text}}$  to capture comprehensive user preference summaries at both coarse-grained (i.e., overall user preferences) and fine-grained levels (i.e., current interests and specific items).

2) EOS representation: Considering the auto-regressive nature of LLMs and the information-rich preference summaries generated by COMPASS, we implement a computationally efficient encoding

<sup>1</sup>Details on caption construction and instruction templates for both item and non-item entities are provided in Appendix B.2.



method. This approach leverages the [EOS] token embedding from the COMPASS-generated summary as it encapsulates the cumulative context of the entire preference description. We then process this embedding through a lightweight two-layer Multi-Layer Perceptron (MLP) to encode the preference as follows:

$$s_c^{\text{EOS}} = \text{MLP}(z_{\text{EOS}}), \quad (9)$$

where  $z_{\text{EOS}}$  is the [EOS] token embedding from the COMPASS-generated preference summary, and  $s_c^{\text{EOS}}$  is the resulting encoded preference. This approach offers computational efficiency while still capturing the essential information from the preference summary, enabling effective integration with existing CRS models. To streamline the notation, we simplify  $s_c^{\text{text}}$  and  $s_c^{\text{EOS}}$  to  $s_c$ .

**3.4.2 Enhanced Recommendation.** We integrate the encoded preferences into the existing CRS models to enhance their recommendation performance. We employ an adaptive gating mechanism to enhance the preference representation  $s_b$  captured from the base CRS model<sup>2</sup> with our COMPASS-generated representation  $s_c$ . Formally, we have:

$$\gamma = \sigma(\mathbf{W}[s_b; s_c]), \quad (10)$$

$$s_u = \gamma s_b + (1 - \gamma) s_c, \quad (11)$$

where  $\mathbf{W}$  are the learnable weight matrices,  $\sigma$  is the sigmoid activation function, and  $\gamma$  represents the gating probability. This adaptive mechanism controls the influence of each representation on the final user preference representation  $s_u$ . The recommendation score for each item is computed using dot-product similarity as:

$$P_{\text{rec}}(i) = \text{softmax}(s_u \cdot I_i), \quad (12)$$

where  $I_i$  are the item representations<sup>3</sup>. Finally, we optimize the recommendation loss  $\mathcal{L}_{\text{rec}}$  as follows:

$$\mathcal{L}_{\text{rec}} = - \sum_{j=1}^N \sum_{i=1}^M y_{ij} \log(P_{\text{rec}}^{(j)}(i)), \quad (13)$$

where  $N$  is the number of conversations,  $M$  is the number of items, and  $y_{ij}$  is the ground truth label indicating whether item  $i$  is relevant to conversation  $j$ .

## 4 Experiments

### 4.1 Experimental Settings

**4.1.1 Datasets.** We conduct our experiments on two widely used English CRS datasets: ReDial [24] and INSPIRED [14] datasets. The ReDial dataset, focused on movie recommendations, contains 11,348 conversations and is constructed through crowdsourcing on Amazon Mechanical Turk (AMT). The INSPIRED dataset contains 999 conversations, also about movie recommendations, but additionally provides recommendation strategies based on social science theories. We constructed the knowledge graph by scraping data from IMDB<sup>4</sup>, using movie titles and release years as key search terms.

<sup>2</sup>The preference representation  $s_b$  is specific to each CRS model as different models employ distinct methods for building user preferences.

<sup>3</sup>The item representations  $I_i$  are specific to each CRS model and may vary depending on the architecture employed by the base CRS.

<sup>4</sup><https://www.imdb.com/>

**4.1.2 Baselines.** We consider a comprehensive range of baseline models, including traditional CRS **ReDial** [24], knowledge-graph based methods such as **KBRD** [3] and **KGSF** [48], language model-based approaches like **BERT** [6], **GPT-2** [31], **Llama3.1-8B** [8], and hybrid models combining KGs with language models such as **BARCOR** [38] and **PECRS** [32].

Since no existing baselines specifically generate user preference summaries, we also compare COMPASS to **Llama3.1-8B** and **GPT-4o**, with **Llama3.1-8B** denoted as **Llama-Summary** when used solely for generating user preference summaries without access to the KG. More details on these models can be found in Appendix B.1.

**4.1.3 Evaluation Metrics.** Our evaluation framework assesses both the performance of the recommendations and the quality of generated user preference summaries. For recommendation tasks, we adopt widely used metrics from previous works [2, 48, 49], including  $\text{HR@K}$ ,  $\text{NDCG@K}$ , and  $\text{MRR@K}$ , with  $K$  set to 10 and 50. To evaluate user preference summaries, we employ three types of metrics: (1) Lexical Similarity: ROUGE-1, ROUGE-2, ROUGE-L, and ROUGE-Sum; (2) Semantic Understanding: Following [36], we use GPT-4o-mini to assess Reasoning Proficiency (RP) and Factual Consistency (FC); (3) User-Centric Evaluation: LLM-based simulated user evaluations, inspired by [13, 19], measure Explainability and User Preference Alignment. More details on the metrics used can be found in Appendix B.4.

**4.1.4 Implementation Details.** We implement COMPASS using the Llama3.1-8B model [8] as the base LLM, which consists of 32 transformer layers with an embedding dimension of 4096. We freeze all parameters of the base LLM and employ LoRA [17] for fine-tuning. For the graph encoder, we set the number of layers to 1, with a hidden dimension of 768. We use batch sizes of 256 for pre-training and 128 for fine-tuning on the Inspired dataset. For the ReDial dataset, we maintain a batch size of 128 for both pre-training and fine-tuning. Early stopping is implemented to optimize training. All experiments are conducted on Nvidia A100 GPUs. For ground truth user preference summaries, we utilize the OpenAI API. More details on this process are provided in Appendix B.3. Our code will be made publicly available upon acceptance.

### 4.2 Recommendation Evaluation

**4.2.1 Improvement over Baseline Models.** COMPASS is designed to be flexible, allowing integration with different CRS models. To assess its effectiveness, we evaluated COMPASS across various CRS models. Performance results are shown in Table 1. We have the following observations: (1) Baseline models incorporating external KGs, such as KBRD and KGSF, consistently outperform simpler language model-based approaches like BERT and GPT-2. This highlights the importance of structured knowledge in capturing user preferences and item relationships, particularly in conversational recommendation settings. (2) Methods that combine language models with KGs, such as BARCOR and PECRS, show further improvements over KG-only models, demonstrating the benefits of integrating both sources of information. (3) Llama3.1-8B demonstrates strong performance, surpassing BARCOR and performing competitively with PECRS on the ReDial dataset while outperforming it on the INSPIRED dataset. This indicates that the advanced

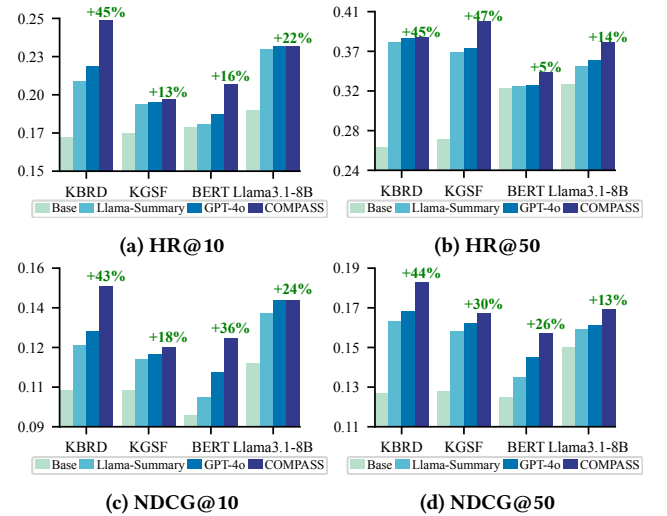
**Table 1: Performance comparison on recommendation tasks. ‘COM.’ denotes models enhanced with the COMPASS approach and ‘Improv.’ indicates the relative improvement of the COMPASS compared to the original base model. The best results are highlighted in bold**

Types	Model	ReDial						INSPIRED					
		HR@10	HR@50	NDCG@10	NDCG@50	MRR@10	MRR@50	HR@10	HR@50	NDCG@10	NDCG@50	MRR@10	MRR@50
Base	BERT [6]	0.143	0.319	0.073	0.108	0.052	0.059	0.179	0.328	0.095	0.125	0.072	0.079
	GPT-2 [31]	0.147	0.327	0.071	0.107	0.051	0.056	0.112	0.278	0.089	0.128	0.063	0.076
	Llama3.1-8B [8]	0.188	0.376	0.103	0.146	0.078	0.087	0.190	0.332	0.118	0.150	0.094	0.102
	ReDial [24]	0.140	0.320	0.061	0.065	0.035	0.045	0.117	0.285	0.035	0.072	0.022	0.048
	KBRD [3]	0.151	0.336	0.099	0.136	0.071	0.079	0.172	0.265	0.106	0.127	0.086	0.091
	KGSF [48]	0.183	0.378	0.098	0.140	0.072	0.081	0.175	0.273	0.106	0.128	0.088	0.093
	BARCOR [38]	0.169	0.374	0.088	0.133	0.063	0.073	0.185	0.339	0.104	0.137	0.080	0.087
	PECRS [32]	0.205	0.399	0.112	0.154	0.083	0.093	0.179	0.337	0.106	0.142	0.084	0.092
Enhanced	COM.+KBRD	0.199	0.412	0.103	0.150	0.075	0.085	0.249	0.392	<b>0.152</b>	<b>0.183</b>	<b>0.123</b>	<b>0.129</b>
	+Improv.	31.79%	22.61%	4.04%	10.29%	5.63%	7.59%	44.76%	47.92%	43.39%	44.09%	43.02%	38.70%
	COM.+KGSF	0.198	<b>0.413</b>	0.105	0.152	0.076	0.088	0.197	<b>0.400</b>	0.125	0.167	0.103	0.110
	+Improv.	8.20%	9.26%	7.14%	8.57%	5.5%	8.64%	12.57%	46.52%	17.92%	30.47%	17.05%	18.28%
	COM.+BERT	0.182	0.382	0.098	0.142	0.073	0.082	0.207	0.345	0.129	0.157	0.105	0.110
	+Improv.	27.27%	19.75%	34.25%	31.48%	40.38%	38.98%	15.64%	5.18%	35.79%	25.60%	45.83%	39.24%
	COM.+Llama3.1-8B	<b>0.215</b>	0.406	<b>0.118</b>	<b>0.161</b>	<b>0.089</b>	<b>0.100</b>	<b>0.232</b>	0.377	0.146	0.169	0.117	0.122
	+Improv.	13.76%	7.98%	14.45%	14.10%	12.82%	14.94%	22.11%	13.55%	23.73%	12.67%	24.47%	19.61%

language understanding and extensive world knowledge of LLMs can effectively capture accurate user preferences, resulting in more accurate recommendations.

Building upon these strong baseline performances, the integration of COMPASS with these models leads to substantial improvements across all evaluation metrics. Notably, when integrated with KBRD on the INSPIRED dataset, COMPASS achieves remarkable relative improvements of 44.76% in HR@10 and 47.92% in HR@50, outperforming all baseline models, including PECRS. When applied to the already strong Llama3-8B model, COMPASS still delivers considerable enhancements, with increases of 13.76% in HR@10 and 14.45% in NDCG@10 on the ReDial dataset, and even greater gains on the INSPIRED dataset. These consistent improvements are attributed to the high-quality, knowledge-enriched user preference representations generated by COMPASS. These representations not only capture user intent more effectively but also provide structured insights that enhance the overall recommendation process, leading to both better performance and explainability.

**4.2.2 Comparison of Enhancement Methods.** To evaluate COMPASS’s effectiveness, we compare it with Llama-Summary and GPT-4o as baseline enhancers, which generate user preference summaries without KG access. Figure 3 illustrates the performance of these methods across various CRS models on the INSPIRED dataset. While all enhancement methods show improvements over the base models, COMPASS consistently outperforms both alternatives. Notably, COMPASS consistently outperforms GPT-4o across all models, with improvements ranging from marginal (for Llama3.1-8B) to substantial (for KBRD and KGSF). This superior performance, achieved despite GPT-4o being a much larger model, demonstrates the importance of integrating domain-specific knowledge into preference modeling. The substantial performance gap between COMPASS and Llama-Summary highlights the effectiveness of our framework in leveraging both structured knowledge and LLM capabilities. These



**Figure 3: Comparison of recommendation performance on the INSPIRED dataset with different enhancers. Green percentages show improvements over baselines.**

results show COMPASS’s ability to generate more accurate and contextually relevant user preference summaries, leading to improved recommendation performance across different CRS architectures.

**4.2.3 Ablation study.** To assess the contribution of different components in COMPASS, we conducted an ablation study by evaluating several model variants across three base models: KBRD, KGSF, and LLaMA3.1-8B. We compare the COMPASS model against three ablated versions: (1) **COM w/o REC**: The generated preference summary does not include recommended items; (2) **COM w/o GEP**: COMPASS without the graph entity captioning pre-training; (3) **COM w/o KG**: COMPASS without the graph encoder.

**Table 2: Ablation study on recommendation task performance. The best-performing results are highlighted in bold.**

Model	ReDial					
	HR@10	HR@50	NDCG@10	NDCG@50	MRR@10	MRR@50
KBRD	0.151	0.336	0.099	0.136	0.071	0.079
+COM w/o KG	0.195	0.405	0.102	0.147	0.074	0.084
+COM w/o REC	0.190	0.395	0.100	0.146	0.075	0.083
+COM w/o GEP	0.188	0.410	0.100	0.148	0.072	0.082
+COMPASS	<b>0.199</b>	<b>0.412</b>	<b>0.103</b>	<b>0.150</b>	<b>0.075</b>	<b>0.085</b>
KGSF	0.183	0.378	0.098	0.140	0.072	0.081
+COM w/o KG	0.196	0.410	0.102	0.150	0.074	0.085
+COM w/o REC	0.192	0.408	0.101	0.150	0.074	0.084
+COM w/o GEP	0.196	0.406	0.102	0.149	0.073	0.084
+COMPASS	<b>0.198</b>	<b>0.413</b>	<b>0.105</b>	<b>0.152</b>	<b>0.076</b>	<b>0.088</b>
Llama3.1-8B	0.188	0.376	0.103	0.146	0.078	0.087
+COM w/o KG	0.209	0.399	0.116	0.159	0.087	0.096
+COM w/o REC	0.202	0.395	0.109	0.151	0.081	0.090
+COM w/o GEP	0.212	0.405	0.117	<b>0.162</b>	0.088	0.099
+COMPASS	<b>0.215</b>	<b>0.406</b>	<b>0.118</b>	0.161	<b>0.089</b>	<b>0.100</b>

**Table 3: Evaluation of generated user preference summaries. RP is reasoning proficiency and FC is factual consistency. The best results are highlighted in bold.**

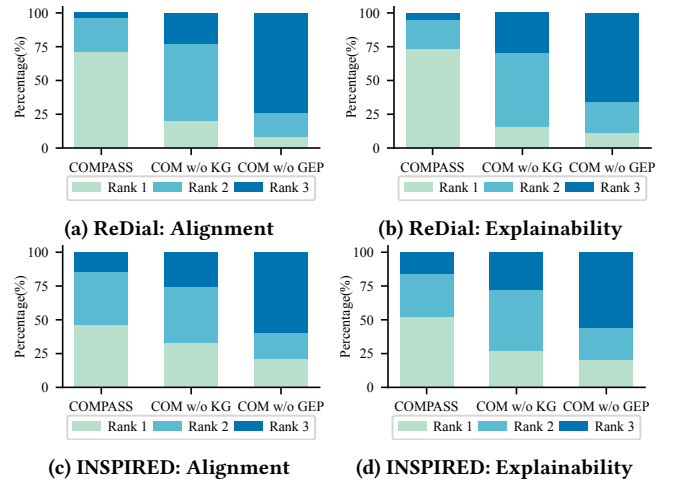
Model	ReDial					
	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-Sum	RP	FC
COM w/o KG	62.06	39.16	49.91	59.33	81.33	82.13
COM w/o NP	61.72	38.95	49.74	59.00	81.16	82.54
COMPASS	<b>62.71</b>	<b>40.61</b>	<b>51.14</b>	<b>60.03</b>	<b>82.20</b>	<b>84.21</b>
Model	INSPIRED					
	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-Sum	RP	FC
COM w/o KG	55.85	32.94	43.36	53.12	82.59	84.33
COM w/o NP	58.20	34.57	44.73	55.08	82.36	84.29
COMPASS	<b>59.37</b>	<b>35.93</b>	<b>46.88</b>	<b>56.64</b>	<b>83.14</b>	<b>85.17</b>

The results in Table 2 show that COMPASS generally achieves the best performance across most metrics, highlighting its superior capability. The COM w/o REC variant shows the largest performance drop, emphasizing the crucial role of including recommended items in preference summaries to guide downstream models. Both COM w/o KG and COM w/o GEP variants demonstrate comparable performance declines, highlighting the crucial roles of KG integration for understanding item relationships and GEP for comprehending KG-augmented context. These findings show the importance of each COMPASS component, with their synergy driving the superior performance of the full model.

### 4.3 Preference Generation Evaluation

This study introduces a novel approach to enhance CRS by generating user preference summaries. As the first attempt in this direction, this evaluation focuses on comparing COMPASS with its variants to assess the effectiveness of its key components. Additionally, a case study that includes examples from Llama-Summary and GPT-4o provides additional qualitative insights.

**4.3.1 Automatic Evaluation.** Table 3 presents a comparative analysis of COMPASS and its variants, COM w/o KG and COM w/o GEP, on the ReDial and INSPIRED datasets. COMPASS consistently outperforms its variants across all metrics on both datasets. Lexical

**Figure 4: LLM-simulated user rankings for User Preference Alignment and Explainability across ReDial and INSPIRED datasets.**

similarity metrics (ROUGE scores) demonstrate that COMPASS generates summaries that align closely with reference texts, while higher reasoning proficiency and factual consistency scores illustrate its enhanced quality of reasoning and factual accuracy.

The performance of COMPASS compared to COM w/o KG indicates the value of KG integration in enhancing summary quality. This suggests that incorporating structured knowledge grounds the model’s output in domain-specific information, which is crucial for accurate preference generation. Moreover, the comparative performance of COM w/o GEP and COMPASS indicates that, despite the presence of structured knowledge from the KG, there remains a modality gap between graph structures and natural language, posing challenges for LLMs in fully understanding and utilizing this information. These results validate the COMPASS framework, and demonstrating the synergistic effect of combining KGs with LLMs, and showing the critical role of our pre-training strategy in enhancing user preference generation.

**4.3.2 LLM-Simulated User Evaluation.** To assess user-perceived quality, we employ LLM-simulated user evaluations on explainability and user preference alignment. Figure 4 presents the results across the ReDial and INSPIRED datasets. COMPASS is consistently preferred over COM w/o KG and COM w/o GEP in both metrics. On the ReDial dataset, COMPASS is most preferred in approximately 75% of the cases for both alignment and explainability. Although the preference gap is smaller on INSPIRED, COMPASS maintains a clear advantage. Notably, COM w/o NP is consistently the least preferred, showing the importance of pre-training in generating meaningful preference summaries. These results demonstrate COMPASS’s superior ability to generate preference summaries that align closely with user preferences while providing clear, user-friendly explanations for recommendations, enhancing both fidelity and transparency in CRSs.



**4.3.3 Case Study.** We present a detailed case study in Table 5 to visualize a sample dialogue along with the corresponding preference summaries generated by different models. It demonstrates that COMPASS accurately identifies the key information for capturing user preference and provides interpretable summaries. The detailed analysis can be found in Appendix C.2.

## 5 Conclusion

In this paper, we introduce COMPASS, a novel framework that synergizes KGs and LLMs to unveil user preferences in Conversational Recommender Systems (CRSs). To address the modality gap between structured knowledge and natural language, we propose a graph entity captioning mechanism that transforms KG structures into LLM-compatible representations. Through knowledge-aware instruction tuning, COMPASS becomes proficient in performing cross-modal reasoning, generating comprehensive and interpretable user preference summaries. Our proposed framework has been extensively evaluated as a plug-and-play enhancement for various existing CRS models across benchmark datasets. The results demonstrate COMPASS's effectiveness in significantly improving both the recommendation performance and the explainability of these base models. The adaptive integration mechanism of COMPASS allows for seamless enhancement of diverse CRS architectures without requiring structural modifications, showcasing its versatility and potential for widespread adoption in the field of CRSs.

## References

- [1] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2018. Neural Attentional Rating Regression with Review-level Explanations. In *Proceedings of the 2018 World Wide Web Conference* (Lyon, France) (WWW '18). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 1583–1592. <https://doi.org/10.1145/3178876.3186070>
- [2] Keyu Chen and Shiliang Sun. 2023. CP-Rec: Contextual Prompting for Conversational Recommender Systems. *Proceedings of the AAAI Conference on Artificial Intelligence* 37, 11 (Jun. 2023), 12635–12643. <https://doi.org/10.1609/aaai.v37i11.26487>
- [3] 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)*. Association for Computational Linguistics, Hong Kong, China, 1803–1813. <https://doi.org/10.18653/v1/D19-1189>
- [4] 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* (San Francisco, California, USA) (KDD '16). Association for Computing Machinery, New York, NY, USA, 815–824. <https://doi.org/10.1145/2939672.2939746>
- [5] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. 2024. InstructBLIP: towards general-purpose vision-language models with instruction tuning. In *Proceedings of the 37th International Conference on Neural Information Processing Systems* (New Orleans, LA, USA) (NIPS '23). Curran Associates Inc., Red Hook, NY, USA, Article 2142, 18 pages.
- [6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Association for Computational Linguistics*, Minneapolis, Minnesota, 4171–4186. <https://doi.org/10.18653/v1/N19-1423>
- [7] Tim Donkers, Benedikt Loepp, and Jürgen Ziegler. 2017. Sequential User-based Recurrent Neural Network Recommendations. In *Proceedings of the Eleventh ACM Conference on Recommender Systems* (Como, Italy) (RecSys '17). Association for Computing Machinery, New York, NY, USA, 152–160. <https://doi.org/10.1145/3109859.3109877>
- [8] Abhimanyu Dubey, Abhinav Jauhri, and et al. 2024. The Llama 3 Herd of Models. arXiv:2407.21783 [cs.AI] <https://arxiv.org/abs/2407.21783>
- [9] Jiabao Fang, Shen Gao, Pengjie Ren, Xiuying Chen, Suzan Verberne, and Zhaochun Ren. 2024. A Multi-Agent Conversational Recommender System. arXiv:2402.01135 [cs.IR] <https://arxiv.org/abs/2402.01135>
- [10] Yue Feng, Shuchang Liu, Zhenghai Xue, Qingpeng Cai, Lantao Hu, Peng Jiang, Kun Gai, and Fei Sun. 2023. A Large Language Model Enhanced Conversational Recommender System. arXiv:2308.06212 [cs.IR] <https://arxiv.org/abs/2308.06212>
- [11] Luke Friedman, Sameer Ahuja, David Allen, Zhenning Tan, Hakim Sidahmed, Changbo Long, Jun Xie, Gabriel Schubiner, Ajay Patel, Harsh Lara, Brian Chu, Zexi Chen, and Manoj Tiwari. 2023. Leveraging Large Language Models in Conversational Recommender Systems. arXiv:2305.07961 [cs.IR] <https://arxiv.org/abs/2305.07961>
- [12] Chongming Gao, Wenqiang Lei, Xiangnan He, Maarten de Rijke, and Tat-Seng Chua. 2021. Advances and challenges in conversational recommender systems: A survey. *AI Open* 2 (2021), 100–126. <https://doi.org/10.1016/j.aiopen.2021.06.002>
- [13] Mingqi Gao, Xinyu Hu, Jie Ruan, Xiao Pu, and Xiaojun Wan. 2024. LLM-based NLG Evaluation: Current Status and Challenges. arXiv:2402.01383 [cs.CL] <https://arxiv.org/abs/2402.01383>
- [14] Shirley Anugrah Hayati, Dongyeop Kang, Qingxiaoyang Zhu, Weiyang 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)*. Association for Computational Linguistics, Online, 8142–8152. <https://www.aclweb.org/anthology/2020.emnlp-main.654>
- [15] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *Proceedings of the 26th International Conference on World Wide Web* (Perth, Australia) (WWW '17). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 173–182. <https://doi.org/10.1145/3038912.3052569>
- [16] Zhanhui He, Zhouhang Xie, Rahul Jha, Harald Steck, Dawen Liang, Yesu Feng, Bodhisattwa Prasad Majumder, Nathan Kallus, and Julian McAuley. 2023. Large Language Models as Zero-Shot Conversational Recommenders. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management* (Birmingham, United Kingdom) (CIKM '23). Association for Computing Machinery, New York, NY, USA, 720–730. <https://doi.org/10.1145/3583780.3614949>
- [17] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.
- [18] Dietmar Jannach, Ahtsham Manzoor, Wanling Cai, and Li Chen. 2021. A Survey on Conversational Recommender Systems. *ACM Comput. Surv.* 54, 5, Article 105 (may 2021), 36 pages. <https://doi.org/10.1145/3453154>
- [19] Yunjie Ji, Xiang Deng, Yan Gong, Yiping Peng, Qiang Niu, Lei Zhang, Baochang Ma, and Xiangang Li. 2023. Exploring the Impact of Instruction Data Scaling on Large Language Models: An Empirical Study on Real-World Use Cases. arXiv:2303.14742 [cs.CL] <https://arxiv.org/abs/2303.14742>
- [20] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Léo Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. arXiv:2310.06825 [cs.CL] <https://arxiv.org/abs/2310.06825>
- [21] Taeho Kim, Juwon Yu, Won-Yong Shin, Hyunyoung Lee, Ji-hui Im, and Sang-Wook Kim. 2023. LATTE: A Framework for Learning Item-Features to Make a Domain-Expert for Effective Conversational Recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Long Beach, CA, USA) (KDD '23). Association for Computing Machinery, New York, NY, USA, 1144–1153. <https://doi.org/10.1145/3580305.3599401>
- [22] 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 *Proc. of ACL*. Association for Computational Linguistics, 7871–7880. <https://doi.org/10.18653/v1/2020.acl-main.703>
- [23] Chuang Li, Yang Deng, Hengchang Hu, Min-Yen Kan, and Haizhou Li. 2024. Incorporating External Knowledge and Goal Guidance for LLM-based Conversational Recommender Systems. arXiv:2405.01868 [cs.CL] <https://arxiv.org/abs/2405.01868>
- [24] 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* (Montréal, Canada) (NIPS '18). Curran Associates Inc., Red Hook, NY, USA, 9748–9758.
- [25] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual Instruction Tuning. In *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (Eds.), Vol. 36. Curran Associates, Inc., 34892–34916. [https://proceedings.neurips.cc/paper\\_files/paper/2023/file/6dcf277ea32ce3288914faf369fe6de0-Paper-Conference.pdf](https://proceedings.neurips.cc/paper_files/paper/2023/file/6dcf277ea32ce3288914faf369fe6de0-Paper-Conference.pdf)
- [26] Yuanxing Liu, Weinan Zhang, Yifan Chen, Yuchi Zhang, Haopeng Bai, Fan Feng, Hengbin Cui, Yongbin Li, and Wanxiang Che. 2023. Conversational Recommender System and Large Language Model Are Made for Each Other in E-commerce Pre-sales Dialogue. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 9587–9605. <https://doi.org/10.18653/v1/>

- 2023.findings-emnlp.643
- [27] 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*. Association for Computational Linguistics, Online, 1161–1173. <https://doi.org/10.18653/v1/2021.findings-acl.99>
- [28] Qiyao Ma, Xubin Ren, and Chao Huang. 2024. XRec: Large Language Models for Explainable Recommendation. *arXiv preprint arXiv:2406.02377* (2024).
- [29] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *arXiv:2203.02155* [cs.CL] <https://arxiv.org/abs/2203.02155>
- [30] Zhangchi Qiu, Ye Tao, Shirui Pan, and Alan Wee-Chung Liew. 2024. Knowledge Graphs and Pretrained Language Models Enhanced Representation Learning for Conversational Recommender Systems. *IEEE Transactions on Neural Networks and Learning Systems* (2024), 1–15. <https://doi.org/10.1109/TNNLS.2024.3395334>
- [31] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. (2019).
- [32] Mathieu Ravaut, Hao Zhang, Lu Xu, Aixun Sun, and Yong Liu. 2024. Parameter-Efficient Conversational Recommender System as a Language Processing Task. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, Yvette Graham and Matthew Purver (Eds.). Association for Computational Linguistics, St. Julian's, Malta, 152–165. <https://aclanthology.org/2024.eacl-long.9>
- [33] Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne vanden Berg, Ivan Titov, and Max Welling. 2018. Modeling Relational Data with Graph Convolutional Networks. In *The Semantic Web*. Springer International Publishing, 593–607.
- [34] Yueming Sun and Yi Zhang. 2018. Conversational Recommender System. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (Ann Arbor, MI, USA) (SIGIR '18)*. Association for Computing Machinery, New York, NY, USA, 235–244. <https://doi.org/10.1145/3209978.3210002>
- [35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, Vol. 30. Curran Associates, Inc. [https://proceedings.neurips.cc/paper\\_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf)
- [36] Jiaan Wang, Yunlong Liang, Fandong Meng, Zengkui Sun, Haoxiang Shi, Zhixu Li, Jinan Xu, Jianfeng Qu, and Jie Zhou. 2023. Is ChatGPT a Good NLG Evaluator? A Preliminary Study. In *Proceedings of the 4th New Frontiers in Summarization Workshop*, Yue Dong, Wen Xiao, Lu Wang, Fei Liu, and Giuseppe Carenini (Eds.). Association for Computational Linguistics, Singapore, 1–11. <https://doi.org/10.18653/v1/2023.newsum-1.1>
- [37] Lingzhi Wang, Huang Hu, Lei Sha, Can Xu, Daxin Jiang, and Kam-Fai Wong. 2022. ReInDial: 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)*, Yulan He, Heng Ji, Sujian Li, Yang Liu, and Chua-Hui Chang (Eds.). Association for Computational Linguistics, Online only, 489–500. <https://aclanthology.org/2022.aacl-main.37>
- [38] Ting-Chun Wang, Shang-Yu Su, and Yun-Nung Chen. 2022. BARCOR: Towards A Unified Framework for Conversational Recommendation Systems. *arXiv:2203.14257* [cs.CL] <https://arxiv.org/abs/2203.14257>
- [39] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. KGAT: Knowledge Graph Attention Network for Recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (Anchorage, AK, USA) (KDD '19)*. Association for Computing Machinery, New York, NY, USA, 950–958. <https://doi.org/10.1145/3292500.3330989>
- [40] Xiaolei Wang, Xinyu Tang, Xin Zhao, Jingyuan Wang, and Ji-Rong Wen. 2023. Rethinking the Evaluation for Conversational Recommendation in the Era of Large Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 10052–10065. <https://doi.org/10.18653/v1/2023.emnlp-main.621>
- [41] Xiaolei Wang, Kun Zhou, Ji-Rong Wen, and Wayne Xin Zhao. 2022. Towards Unified Conversational Recommender Systems via Knowledge-Enhanced Prompt Learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, Washington DC USA, 1929–1937. <https://doi.org/10.1145/3534678.3539382>
- [42] Yuling Wang, Changxin Tian, Binbin Hu, Yanhua Yu, Ziqi Liu, Zhiqiang Zhang, Jun Zhou, Liang Pang, and Xiao Wang. 2024. Can Small Language Models be Good Reasoners for Sequential Recommendation?. In *Proceedings of the ACM Web Conference 2024 (Singapore, Singapore) (WWW '24)*. Association for Computing Machinery, New York, NY, USA, 3876–3887. <https://doi.org/10.1145/3589334.3645671>
- [43] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *arXiv:2201.11903* [cs.CL] <https://arxiv.org/abs/2201.11903>
- [44] Yunjia Xi, Weiwen Liu, Jianghao Lin, Bo Chen, Ruiming Tang, Weinan Zhang, and Yong Yu. 2024. MemoCRS: Memory-enhanced Sequential Conversational Recommender Systems with Large Language Models. *arXiv:2407.04960* [cs.IR] <https://arxiv.org/abs/2407.04960>
- [45] Yongfeng Zhang and Xu Chen. 2020. Explainable Recommendation: A Survey and New Perspectives. *Foundations and Trends® in Information Retrieval* 14, 1 (2020), 1–101. <https://doi.org/10.1561/15000000066>
- [46] 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 *ACL, system demonstration*.
- [47] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A Survey of Large Language Models. *arXiv:2303.18223* [cs.CL] <https://arxiv.org/abs/2303.18223>
- [48] Kun Zhou, Wayne Xin Zhao, Shuqing Bian, Yuanhang Zhou, Ji-Rong Wen, and Jingsong Yu. 2020. Improving Conversational Recommender Systems via Knowledge Graph based Semantic Fusion. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, Virtual Event CA USA, 1006–1014. <https://doi.org/10.1145/3394486.3403143>
- [49] Yuanhang Zhou, Kun Zhou, Wayne Xin Zhao, Cheng Wang, Peng Jiang, and He Hu. 2022. C<sup>2</sup>-CRS: Coarse-to-Fine Contrastive Learning for Conversational Recommender System. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. ACM, Virtual Event AZ USA, 1488–1496. <https://doi.org/10.1145/3488560.3498514>
- [50] Lixi Zhu, Xiaowen Huang, and Jitao Sang. 2024. How Reliable is Your Simulator? Analysis on the Limitations of Current LLM-based User Simulators for Conversational Recommendation. In *Companion Proceedings of the ACM on Web Conference 2024 (Singapore, Singapore) (WWW '24)*. Association for Computing Machinery, New York, NY, USA, 1726–1732. <https://doi.org/10.1145/3589335.3651955>

## A Related Work

### A.1 Conversational Recommender System

Conversational Recommender Systems (CRSs) represent an evolution in recommendation technology, designed to understand user preferences through interactive dialogue and provide personalized recommendations [18]. These systems face a unique challenge in modeling user preferences from typically brief and sparse conversational data. To address this, recent works incorporate external knowledge sources into CRSs to enhance preference modeling. Knowledge graphs (KG) enable systems to infer implicit user interests and capture complex relationships between items [3, 48], while user reviews provide additional insights into item attributes and user opinions [21, 27, 49], helping to address the challenge of limited contextual information. Moreover, the integration of pre-trained language models (PLMs) such as DialoGPT [37, 41, 46] and BART [22, 30] has further improved contextual understanding and language generation capabilities in CRSs.

More recently, the advent of powerful LLMs has introduced new opportunities for CRSs. Studies have focused on improving LLM evaluation through enhanced user simulators [40, 50], exploring zero-shot recommendation capabilities [16], and utilizing LLMs for task planning in CRSs [9, 10, 23]. However, one critical aspect that remains underexplored is the role of explainable user preference in CRSs. Although some conceptual designs propose using LLMs to manage user preferences [11], they lack quantitative validation. [44] addresses user preference memories in sequential CRS but does not fully harness LLM's reasoning abilities for preference generation. Our work is the first to introduce a framework that synergizes the

LLM and the KG for unveiling user preferences in CRSs, enhancing both preference modeling and recommendation explainability.

## A.2 Explainable Recommendation

Explainable recommendation has emerged as a crucial area of research, aiming to enhance user trust, satisfaction, and decision making by providing transparent rationales for recommendations. Early research focused on leveraging user-item interactions and content features to generate simple explanations [45], using techniques like recurrent neural networks [7], attention mechanisms [1], and graph neural networks [39]. Recent developments in LLMs have opened new avenues for explainable recommendation. Studies [28, 42] have demonstrated LLMs’ potential in generating explanations by analyzing user’s past interactions. However, current LLM-based methods primarily operate within the constraints of static user profiles, where preferences are inferred from historical behavior. Our work introduces a novel LLM-KG framework for CRSs, enhancing both preference modeling and recommendation explainability in dynamic interaction contexts.

## B Details of Experimental Setting

### B.1 Model Details

*B.1.1 Backbone Models.* The detailed information for each backbone is as follows:

- ReDial [24]: Introduces the ReDial dataset and model for conversational recommendation, utilizing a denoising auto-encoder[15] as the recommendation module.
- KBRD [3]: This method leverages an external KG to enhance the semantics of entities mentioned in the dialogue history.
- KGSF [48]: This method employs a dual KG approach, integrating semantic information from both word-level and entity-level KGs. It applies a mutual information maximization technique to align these two semantic spaces, enhancing the overall representation.
- BERT [6]: This method utilizes a bidirectional Transformer model [35] pre-trained on a large-scale corpus. We use the [CLS] token representation for the recommendation task.
- GPT-2 [31]: An auto-regressive language model. We fine-tune it on dialogue history and use the last token representation for the recommendation task.
- Llama3.1-8B [8]: This is an open-source LLM that has been trained using both supervised fine-tuning and reinforcement learning with human feedback [29]. For the recommendation task, we fine-tune Llama3.1-8B using LoRA on the dialogue history and use the last token representation as the user preference embedding.
- BARCOR [38]: This model introduces a unified framework that integrates BART with a knowledge graph, enabling both recommendation and response generation within a single model.
- PECRS [32]: This model integrates recommendation and response generation into a single training phase using LoRA [17] with a pre-trained language model, eliminating the need for separate modules or external knowledge graphs.

This study introduces a novel approach to enhance CRS by generating user preference summaries. As there are no existing enhancers specifically designed for this task, we use **Llama3.1-8B**<sup>5</sup> and GPT-4o<sup>6</sup> as baselines for comparison, both of which generate user preference summaries without access to KG information. To clarify, Llama3.1-8B is used in two different capacities in this work: first, as a backbone CRS model adapted through training for the recommendation task, and second, as Llama-Summary for generating user preference summaries without KG access.

*B.1.2 COMPASS Integration.* We enhance CRS models performance by incorporating our COMPASS framework as described in Section 3.4. The integration method is consistent across KBRD, KGSF, and BERT, utilizing the adaptive gating mechanism to combine COMPASS-generated preference summaries with the original model representations. For preference-enhanced Llama3, we employ a different integration strategy. We directly concatenate the generated user preference summary with the conversation history. This approach leverages Llama3.1-8B’s strong language understanding capabilities to process the additional preference information.

### B.2 Instruction Design and Caption Templates

The instruction templates for graph entity captioning and knowledge-aware instruction tuning are presented in Tables 6, 7, and 8. Table 6 shows various instruction templates for different node types in the knowledge graph, including non-item nodes (actors, directors, genres, keywords, production companies) and item nodes (movies). Table 7 presents the structured template for user preference modeling and recommendation, designed to produce JSON-formatted output. Table 8 displays the instruction for Llama3.1-8B to make direct movie recommendations.

The caption templates for various entity types in the KG (such as genres, actors, directors, companies, and keywords) are presented in Table 9. These templates guide the translation of graph structures into natural language descriptions, emphasizing each entity’s relationships within the domain. The templates enable COMPASS to generate contextual descriptions that preserve graph structure while presenting information in a format compatible with the LLM, facilitating effective cross-modal analysis.

### B.3 Ground Truth User Preference Summary Generation

To generate reliable ground truth user preference summaries, we employed OpenAI’s GPT-4o model<sup>7</sup>, with temperature set to 0 for deterministic outputs. The model performed cross-modal reasoning by integrating the complete dialogue history with structured metadata of items mentioned in the conversation, derived from the KG. We developed a structured instruction template, presented in Table 10, to guide this analysis process. This template outlines a step-by-step approach for discerning user preferences through conversation analysis, KG cross-referencing, and insight synthesis, resulting in a comprehensive user preference summary.

<sup>5</sup><https://ollama.com/>

<sup>6</sup>[gpt-4o-2024-05-13](https://openai.com/gpt-4o-2024-05-13)

<sup>7</sup>[gpt-4o-2024-05-13](https://openai.com/gpt-4o-2024-05-13)

**Table 4: Comparison of preference encoding methods for integrating COMPASS generated summaries. The best results are highlighted in bold.**

Model	ReDial					
	HR@10	HR@50	NDCG@10	NDCG@50	MRR@10	MRR@50
KBRD	0.151	0.336	0.099	0.136	0.071	0.079
+EOS	0.188	0.379	0.099	0.141	0.073	0.082
+Text	<b>0.199</b>	<b>0.412</b>	<b>0.103</b>	<b>0.150</b>	<b>0.075</b>	<b>0.085</b>
KGSF	0.183	0.378	0.098	0.140	0.072	0.081
+EOS	0.190	0.383	0.102	0.144	0.075	0.084
+Text	<b>0.198</b>	<b>0.413</b>	<b>0.105</b>	<b>0.152</b>	<b>0.076</b>	<b>0.088</b>
BERT	0.143	0.319	0.073	0.108	0.052	0.059
+EOS	0.162	0.348	0.084	0.125	0.060	0.069
+Text	<b>0.182</b>	<b>0.382</b>	<b>0.098</b>	<b>0.142</b>	<b>0.073</b>	<b>0.082</b>

## B.4 Detailed Evaluation Process

**B.4.1 Semantic Understanding Evaluation.** Following the method proposed by [36], we adopt an LLM-based evaluation approach where GPT-4o-mini serves as an automated scorer to evaluate the reasoning proficiency (RP) and fact consistency (FC) of the user preference summaries generated. The evaluation of RP is centered on three main criteria: (1) Logical Coherence, which assesses the internal consistency of the reasoning process; (2) Accuracy of Inferences, which measures how precisely deductions are made from the dialogue and KG data; and (3) Relevance of Recommendations, which evaluates whether the suggested items are appropriate given the user’s expressed interests. FC evaluation similarly considers three aspects: (1) Dialogue Alignment, which measures fidelity to user-expressed preferences; (2) KG Consistency, assessing the accuracy of preference keywords relative to the KG content; and (3) Claim Validity, checking for unsupported assertions while allowing for reasonable expansions. The evaluation uses the dialogue history, KG data related to the mentioned entities in the dialogue, and the generated preference summary as input. GPT-4o-mini then analyzes these inputs and generates scores for RP and FC on a scale of 0-100. The average scores are calculated for all test samples.

**B.4.2 User-Centric Evaluation.** To complement the quantitative metrics with user-oriented insights, a user-centric evaluation is performed using GPT-4o-mini as a simulated user, following the approach outlined by [13, 19]. We evaluate the Explainability and User Preference Alignment of the generated preference summaries and recommendations. Explainability refers to how clearly the system justifies its summaries and recommendations in a way that is easily understandable from the user’s perspective. User Preference Alignment assesses how accurately the system captures and responds to the preferences expressed by the user in the conversation. The evaluation uses the dialogue history and preference summaries from different models as inputs. GPT-4o-mini ranks these summaries according to the EX and UPA criteria, providing justifications for each ranking.

## C Further Experiment

### C.1 Evaluation of Preference Encoding Methods

Our COMPASS framework demonstrates versatility in generating user preference summaries, offering both comprehensive textual descriptions and compact EOS token embeddings. To integrate these outputs into existing CRS models, we evaluated two encoding methods: a lightweight encoder and a PLM-based text encoder. Table 4 presents results for KBRD, KGSF, and BERT models on the ReDial dataset, which shows the effectiveness of both approaches. The PLM-based text encoder (108M parameters) excels in processing COMPASS’s textual summaries, achieving superior performance across all metrics.

Conversely, the lightweight encoder (16.8M parameters, ~15.5% of PLM-based) efficiently processes COMPASS’s EOS token embeddings. Despite its simplicity, it significantly improves on the baseline performance, increasing KBRD’s HR@10 and HR@50 by 24.5% and 12.8% respectively. This demonstrates COMPASS’s capability to distill user preferences into compact yet informative representations. Both methods enhance performance across different base models, highlighting COMPASS’s adaptability in generating preference summaries. This versatility enables flexible deployment across various computational environments, from resource-constrained systems to high-performance platforms, thereby accommodating a wide range of application scenarios.

### C.2 Case Study

Table 5 presents a detailed visualization of a sample dialogue along with the corresponding preference summaries generated by different models. COMPASS demonstrates compelling effectiveness in capturing and reasoning about user preferences. It accurately identifies the movie genres and lists key cast members and the director of *The Professional (1981)*. Moreover, COMPASS infers relevant themes that align with the movie’s content, such as *professional killers*, and *intense action sequences*. In contrast, other models show varying levels of accuracy and reasoning depth. (1) GPT-4o, without access to the KG, provides a broader interpretation. While it correctly identifies some genres, it also includes unsupported elements. This illustrates that while LLMs like GPT-4o possess some general world knowledge, they may lack the domain-specific information necessary for precise user preference summarization. (2) Llama-Summary, also without a knowledge graph, demonstrates significant inaccuracies. It incorrectly attributes the film to the *French New Wave movement* and lists incorrect personnel. This highlights the difficulties that smaller LLMs face in domain-specific tasks without specialized knowledge. (3) COMP w/o KG, while correctly identifying some genres and themes, introduces factual errors, such as misattributing the film to different actors and directors. This highlights the importance of integrating structured KGs, even when models are fine-tuned on domain-specific tasks. (4) COMP w/o GEP shows improved accuracy in genre identification but continues to struggle with factual details about the cast and crew. Although it makes plausible inferences, it still includes unsupported information. These comparisons emphasize the crucial roles that both KG integration and pre-training play in enabling COMPASS to generate accurate and contextually relevant user preference summaries.

**Table 5: Case Study: Comparison of preference summaries generated by different models. Green shows information supported by the knowledge graph, yellow for correct inferences that are not explicitly detailed in the knowledge graph but align with the content, and red for unsupported and unreasonable details**

Hist.	<p><b>System:</b> Hello! What kind of movies are you into?  <b>User:</b> I loved <i>The Professional (1981)</i> and would love to see something similar.</p>
User Preference Summary	<p><b>COMPASS: Reasoning:</b> The user mentioned ‘The Professional (1981)’ as a movie they loved and would like to see something similar. By analyzing the knowledge graph, we can identify that ‘The Professional’ falls under the genres of <b>Action, Crime, and Thriller</b>, and includes themes such as <b>professional killers, assassins, and intense action sequences</b>. The movie also features notable actors like <b>Paul Belmondo and Michel Beaune</b>, and is directed by <b>Georges Lautner</b>. These elements suggest the user enjoys <b>high-stakes, action-packed narratives with complex characters and moral dilemmas</b>. <i>Overall preferences:</i> <b>Action, Crime, Thriller, professional killers, assassins, intense action sequences, complex characters, moral dilemmas, Jean-Paul Belmondo, Georges Lautner</b>. <i>Current interests:</i> <b>Action, Crime, Thriller, professional killers, assassins, intense action sequences</b>.</p> <p><b>GPT-4o: Reasoning:</b> The user mentioned a specific movie, ‘The Professional (1981)’, which they loved and expressed interest in finding similar movies. This indicates a preference for films with similar themes, styles, or elements. <i>Overall preferences:</i> <b>crime, drama, thriller, intense character development, complex relationships, suspense</b>. <i>Current interests:</i> <b>The Professional (1981), crime drama, thriller, intense character dynamics</b>.</p> <p><b>Llama-Summary: Reasoning:</b> The user’s conversation history indicates a preference for <b>1980s French cinema</b>, particularly from the <b>Nouvelle Vague movement</b>, with an emphasis on <b>intelligent, stylish, and critically-acclaimed films that often involve morally complex situations</b>. <i>Overall preferences:</i> <b>French New Wave, Intellectual, Stylish, Morally Complex</b>. <i>Current interests:</i> <b>Jean-Pierre Mouse, The Professional (1981), Claude Sautet, Claudie Clairat</b>.</p> <p><b>COM w/o KG: Reasoning:</b> The user mentioned ‘The Professional (1981)’ as a movie they loved and would like to see something similar. By analyzing the knowledge graph, we can identify that ‘The Professional’ is an <b>action, crime, and drama film</b> with themes of <b>revenge, vigilantism, and professional killers</b>. The movie stars <b>Burt Lancaster and Robert Duvall</b> and is <b>directed by Sidney Lumet</b>. These elements suggest the user enjoys <b>intense, character-driven narratives with a focus on crime and moral ambiguity</b>. <i>Overall preferences:</i> <b>Action, Crime, Drama, revenge, vigilantism, professional killer, intense character-driven narratives, moral ambiguity, Burt Lancaster, Robert Duvall, Sidney Lumet</b>. <i>Current interests:</i> <b>Action, Crime, Drama, revenge, professional killer, intense character-driven narratives</b>.</p> <p><b>COM w/o GEP: Reasoning:</b> The user mentioned ‘The Professional (1981)’ as a favorite movie and expressed interest in finding something similar. By analyzing the knowledge graph, we can identify that ‘The Professional’ falls under the genres of <b>Action, Crime, and Thriller</b>, and includes themes such as <b>police, political corruption, and organized crime</b>. The movie also features notable actors like <b>Ryan O’Neal and Charles Grodin</b>, and is directed by <b>Donald Shebib</b>. These elements suggest the user enjoys <b>intense, plot-driven narratives with a focus on crime and political intrigue</b>. <i>Overall preferences:</i> <b>Action, Thriller, police, political corruption, organized crime, Ryan O’Neal, Charles Grodin, Donald Shebib</b>. <i>Current interests:</i> <b>Thriller, police, political corruption, organized crime</b>.</p>



**Table 6: Instruction Templates for Graph Entity Captioning**

Node Type	Instruction Templates
Non-item nodes	<p>Please specify the node type (actor, director, genre, keywords, production company) and briefly provide associated details.</p> <p>Specify the category of movie-related personnel or aspect and provide a concise summary. Indicate the node type related to movies and offer a short overview of the associated key details. What is the category of this node, and what are the main details related to it?</p> <p>What is the type of movie-related entity, and what are some key details about it?</p>
Item nodes	<p>Please summarize the details of the movie node, including its title, year, genre, keywords, director, and plot.</p> <p>Provide a concise summary of the movie, including elements such as its title, release year, genre, key themes, director, and a brief plot overview.</p> <p>Detail this movie’s specifics, including title, year, genre, keywords, director, and a short plot summary.</p> <p>Present a summary of this film, noting its title, release year, genre, key themes, director, and plot. Give a brief rundown of the movie, specifying the title, year it was made, genre, key terms, director, and plot.</p>

**Table 7: Instruction Template for User Preference Modeling and Recommendation**

Instruction Template
<p>As a movie recommender, analyze the user’s conversation history to make a movie recommendation. Break down the analysis into clear steps and return in json format:</p> <ol style="list-style-type: none"> <li>reasoning: analyze the user’s conversation history and knowledge graph data to highlight their movie preferences.</li> <li>overall preferences: A list of keywords summarizing the user’s overall preferences.</li> <li>current interests: A list of keywords reflecting user’s current interests.</li> <li>recommendation: a recommended movie title.</li> </ol> <p>History: [Conversation history is inserted here]</p>

**Table 8: Instruction for Direct Recommendation**

Instruction Template
<p>As a movie recommender, please recommend a movie that aligns with the user’s interests discussed in the conversation history provided.</p> <p>Conversation history: [Conversation history is inserted here]</p>

**Table 9: Caption Templates for Different Node Types**

Node Type	Caption Templates
Genre	<p>This node summarizes movies associated with the &lt;genre&gt; genre.</p> <p>This node focuses on &lt;genre&gt; type movies.</p> <p>Movies here are primarily categorized under the &lt;genre&gt; genre.</p> <p>A collection of &lt;genre&gt; films defines this node.</p> <p>This node represents a movie genre that is &lt;genre&gt;.</p>
Actor	<p>This node summarizes movies featuring actor &lt;actor&gt;.</p> <p>This node focuses on films starring &lt;actor&gt;.</p> <p>Movies with &lt;actor&gt; prominently featured make up this node.</p> <p>This node collects major works of actor &lt;actor&gt;.</p> <p>This node highlights films that star &lt;actor&gt;.</p>
Director	<p>This node summarizes movies directed by &lt;director&gt;.</p> <p>This node focuses on films by director &lt;director&gt;.</p> <p>Films directed by &lt;director&gt; are centrally featured in this node.</p> <p>This node is a compilation of &lt;director&gt;'s directorial efforts.</p> <p>A selection of &lt;director&gt;'s films defines this node.</p>
Company	<p>This node summarizes films produced by &lt;company&gt;.</p> <p>This node focuses on movies from the production company &lt;company&gt;.</p> <p>Productions by &lt;company&gt; are highlighted in this node.</p> <p>This node represents the movie production company &lt;company&gt;.</p> <p>Films from &lt;company&gt; prominently make up this node.</p>
Keyword	<p>This node summarizes movies associated with the keyword of &lt;keyword&gt;.</p> <p>This node focuses on movies themed around &lt;keyword&gt;.</p> <p>Films grouped in this node feature the theme: &lt;keyword&gt;.</p> <p>This node aggregates movies revolving around the keyword: &lt;keyword&gt;.</p> <p>This node indicates films related to the keyword: &lt;keyword&gt;.</p>

**Table 10: Instruction Template for Ground Truth Generation**

---

**Instruction Template**

---

As a personalized movie recommender, your task is to analyze the user’s conversation history and relevant movie data from the knowledge graph to discern a balanced mix of specific movie titles and broader thematic and stylistic movie preferences. Think step by step and your analysis should be nuanced and concise, using movie titles as entry points to uncover deeper interests in themes, genres, and styles.

## # Analysis Steps

1. Review Conversation History: Extract mentions of movies and any expressed preferences in genres, keywords, actors, directors, or production companies from the dialogue.
2. Cross-Reference with Knowledge Graph: Correlate the extracted movie preferences from the conversation with the knowledge graph data, looking for patterns in genres, keywords, actors, directorial styles and etc.
3. Synthesize Insights: Combine the conversation and knowledge graph analysis to create a profile of the user’s movie preferences, highlighting both specific movies and general thematic interests.

## # Input Example

## Conversation History

\$User: {}

\$Recommender: {}

## Knowledge Graph

(1). \${movie name}(\$year); Genres: \${Genre}; Keywords: \${Keywords}; Starring: \${Starring}; Director: \${Director}; Company: \${Company}; Plot: \${Plot}

...

# Expected Output

JSON Format Response

{

"reasoning": "A concise explanation reflecting the Analysis Steps, detailing how specific movie and broader thematic and stylistic preferences were identified and integrated.",

"overall preferences": "A short list of keywords summarizing the user’s general thematic and stylistic movie preferences, including genres, notable keywords, preferred actors, directors, narrative styles, etc.",

"current interest": "A concise list of keywords reflecting the user’s most recent and pertinent thematic and stylistic interests, guiding the next recommendations."

}

Conversation History: [Conversation history is inserted here]

Knowledge Graph: [Mentioned Items Meta Information from Knowledge Graph is inserted here]

---