Instructing and Prompting Large Language Models for Explainable Cross-domain Recommendations

Alessandro Petruzzelli
alessandro.petruzzelli@uniba.it
University of Bari Aldo Moro, Italy

Cataldo Musto*
cataldo.musto@uniba.it
University of Bari Aldo Moro, Italy

Lucrezia Laraspata
l.laraspata3@studenti.uniba.it
University of Bari Aldo Moro, Italy

Ivan Rinaldi
i.rinaldi4@studenti.uniba.it
University of Bari Aldo Moro, Italy

Marco de Gemmis
marco.degemmis@uniba.it
University of Bari Aldo Moro, Italy

Pasquale Lops
pasquale.lops@uniba.it
University of Bari Aldo Moro, Italy

Giovanni Semeraro
giovanni.semeraro@uniba.it
University of Bari Aldo Moro, Italy

ABSTRACT

In this paper, we present a strategy to provide users with explainable cross-domain recommendations (CDR) that exploits large language models (LLMs). Generally speaking, CDR is a task that is hard to tackle, mainly due to *data sparsity* issues. Indeed, CDR models require a large amount of data labeled in both *source* and *target* domains, which are not easy to collect. Accordingly, our approach relies on the intuition that the knowledge that is already encoded in LLMs can be used to more easily bridge the domains and seamlessly provide users with personalized cross-domain suggestions.

To this end, we designed a pipeline to: *(a)* instruct a LLM to handle a CDR task; *(b)* design a personalized prompt, based on the preferences of the user in a *source* domain, and a list of items to be ranked in *target* domain; *(c)* feed the LLM with the prompt, in both *zero-shot* and *one-shot* settings, and process the answer in order to extract the recommendations and a natural language explanation. As shown in the experimental evaluation, our approach beats several established state-of-the-art baselines for CDR in most of the experimental settings, thus showing the effectiveness of LLMs also in this novel and scarcely investigated scenario.

KEYWORDS

Cross-domain Recommendations; Recommender Systems; Large Language Models; Instruction Tuning

ACM Reference Format:
Alessandro Petruzzelli, Cataldo Musto, Lucrezia Laraspata, Ivan Rinaldi, Marco de Gemmis, Pasquale Lops, and Giovanni Semeraro. 2018. Instructing and Prompting Large Language Models for Explainable Cross-domain Recommendations. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX).* ACM, New York, NY, USA, 10 pages. https://doi.org/XXXXXXX.XXXXXXX

*Corresponding author.

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Recommender systems (RSs) have emerged as essential tools to support users in decision-making [39]. These systems leverage user preferences and historical behavior to suggest items of potential interest, ranging from movies and music to products and news articles [20]. Generally speaking, RSs employ various algorithms, including collaborative filtering, content-based filtering, and hybrid methods, to generate personalized recommendations tailored to individual users [40]. Cross-domain recommender systems (CDR) [10] represent a specialized class of RSs designed to address the challenge of recommendation in heterogeneous domains. Unlike traditional RSs, that operate within a single domain, cross-domain recommender systems extend their capabilities across multiple domains (typically referred to as *source* and *target* domains), accommodating diverse types of items and user preferences. By leveraging knowledge transfer and domain adaptation techniques [24], these systems enable the transfer of insights and recommendations from one domain to another. However, CDR typically suffer of *data sparsity* issues [6], since they require a large amount of labeled data in both the domains, which is not easy to collect. Moreover, some works [8] showed that the actual *transfer* of knowledge that happens between the domains is often over-claimed. A suitable solution is represented by content-based features and item metadata, whose exploitation in the area of RSs is largely established [30, 34]. As shown by several works, [12, 17], this information be used to better link the knowledge across the domains and provide accurate CDRs.

In this research direction, Large Language Models (LLMs) represent a promising means to provide CDRs with the knowledge that is needed to transfer preferences across different domains. Indeed, LLMs are trained on vast corpora of text data to learn complex patterns and relationships within language. These models, such as GPT [1, 54] and LLaMa [48, 49], excel at natural language understanding and generation tasks, including machine translation [2, 7], summarization [25, 47], and dialogue generation [51]. While some attempts towards the exploitation of LLMs for recommendation tasks has been proposed [9, 15, 16], up to our knowledge the use of LLMs to tackle CDR tasks has been scarcely investigated. Conversely, in our conjecture, the huge amount of knowledge that is already encoded in LLMs can be used to more easily bridge the domains and seamlessly provide users with cross-domain suggestions.

Accordingly, this paper aims to fill in this gap by proposing a pipeline to: *(a)* instruct open LLMs to handle a CDR task; *(b)* design a personalized prompt, based on the preferences of the user in a *source* domain, and a list of items to be ranked in *target* domain; *(c)* feed the LLM with the prompt, in both *zero-shot* and *one-shot* settings, and process the answer to extract the recommendations together with a natural language explanation. As shown in the experiments, based on the comparison of three different state-of-the-art (SOTA) LLMs, *i.e.,* GPT, LLama and Mistral [1, 21, 48, 49, 54], our approach beats several baselines for CDR in most of the experimental settings, thus showing the effectiveness of LLMs also in this scarcely investigated scenario. To sum up, this paper provides the following contributions: (1)We introduce a pipeline that allows LLMs to handle a CDR task, based on instruction tuning techniques; (2) We design a strategy to prompt LLMs and obtain suitable CDRs, together with a natural language explanation; (3) We compare our approach to several baselines for CDR, and we guarantee the reproducibility of the protocol by releasing our source code.

The rest of the paper is organized as follows: in Section 2 we discuss related work. Next, Section 3 introduces the problem and Section 4 describes our workflow to obtain CDRs based on LLMs. Finally, in Section 5 we present our experiments, and in Section 6 we provide the conclusions and sketch future research directions.

2 RELATED WORK

In this section, we provide some basics of LLMs and we present an overview of RSs based on these models, by emphasizing the distinctive traits of our work.

2.1 Large Language Models

Large language models (LLMs) have revolutionized the landscape of artificial intelligence research, including RSs. These models, which put down roots in the area of distributional semantics [35], are typically based on Transformers [50] and are trained on huge amount of textual data. The hallmark of Transformers is the use of *attention mechanisms* to influence the representation of a piece of text based on the semantics of the (most relevant) surrounding words. This allows to learn representations that are very accurate and encode many nuances of the language [18]. Early LLMs, such as BERT [11], exploited the Transformers in the so-called *encoder-only* fashion, *i.e.*, the output of the process was an embedding representing word or sentences. Conversely, more recent models, such as GPT [1, 54], LLaMa [48, 49], T5 [37], and Mistral [21], rely on *encoder-decoder* or *decoder-only* architectures. In other terms, these LLMs can also generate textual content that is coherent to a generic input (*i.e.,* the answer to a question, or the completion of a sentence). Such a *generation* capability allowed LLMs to handle many and diverse downstream tasks, including recommendations, and gave rise to systems such as ChatGPT.

In order to fully exploit the knowledge available in LLMs, it is necessary to *prompt* these models. Generally speaking, the concept of *prompt engineering* [5] refers to process of crafting specific instructions, called prompts, to guide a LLM towards a desired outcome. Based on the amount of information that is provided to the LLMs, prompting strategies can be roughly split into *zero-shot* and *one-shot* (or, more in general, *few-shot*) prompting. In the first case, the prompt itself contains all the necessary information for the LLM to complete the task. As an example, a zero-shot prompt for a recommendation task may just include the preferences of the user and the request for a list of suggestions. Conversely, in the latter case, also referred to as *in-context learning*, the prompt is accompanied by one or more examples that show the desired output, in terms of content, format and style. This extra information helps the LLM to better understand the task and return a correct output.

However, the effectiveness of prompting strategies often clashes with the number of parameters of the models. Indeed, huge models such as GPT3.5 or GPT-4 (consisting of approx. 175B and 1.8T of parameters[1], respectively) are able to effectively tackle very diverse tasks, even in zero-shot settings. Conversely, relatively smaller models, such as LLaMa (consisting of 7 or 13 billions of parameters), do not have sufficient generalization capabilities to be equally accurate. In these cases, before prompting a model it is often necessary to *specialize* them in the task to be tackled. This is done through an approach that is referred to as *fine-tuning*, which involves an additional training step of the LLM by injecting a set of new and labeled data that are specific to the task. When fine-tuning is used to let LLMs handle new tasks it was not originally trained on, this strategy is known as *instruction tuning* [56]. As regards training strategies, when re-training is too computationally expensive, strategies for parameter-efficient fine tuning (PEFT) such as LORA [19] and prompt tuning [26] can be exploited. These models either work on an approximation of the original weight matrices or learn a smaller portion of the original LLM. However, *it is necessary to point out that fine-tuning and instruction-tuning strategies can be only run on open-sourced LLMs, such as LLaMa.*

In this paper, we put together all the above mentioned pieces and we propose a framework based on LLMs to tackle a CDR task. This is done by first exploiting *instruction tuning* techniques to let LLMs adapt to our scopes. Subsequently, we explored *prompt engineering* to design a proper prompt that returns a list of recommendations together with an explanation. All the details will provided next.

2.2 LLMs for Recommender Systems

While early attempts have exploited LLMs and Transformers as encoders [36, 46], recent works either use them by directly prompting for recommendations or be fine-tuning for the task.

As for fine-tuned models, the first attempt to jointly address recommendation tasks through LLMs is represented by P5 [16]. This model is based on T5 [37] and introduces different prompts to tackle different recommendation tasks, including rating prediction, sequential recommendation and so on. A similar attempt is proposed in M6-Rec [9], built on the M6 architecture [27]. While both these models obtained remarkable results, it was not possible to consider them as *baselines* since CDR is not among the tasks these models can handle. Moreover, our work is not directly comparable since we designed a *discrete* prompt where preferences are represented in a textual form, rather than using only item IDs.

[1] https://the-decoder.com/gpt-4-architecture-datasets-costs-and-more-leaked/

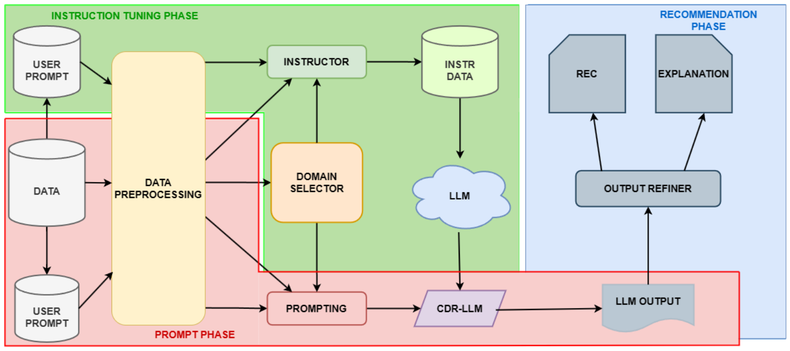

Figure 1: Workflow of the proposed model

Next, as regards models directly prompting LLMs to obtain recommendations, in [23] the authors evaluate models belonging to the GPT family [55]. This is done by feeding the LLM with a prompt representing the user profile based on their past item ratings. Even without fine-tuning, the model achieved performance comparable to baseline models. Another study [41] focused on sequential recommendation tasks, considering both item descriptions and user preferences. Their findings suggest that in-context learning outperforms the zero-shot approach. Finally, in [15] the authors propose a framework utilizing GPT for both rating and ranking prediction, showcasing remarkable capabilities in the movie domain.

While our approach shares with these works the idea of prompting LLMs for recommendations, **the novelty of this paper lies in the exploitation of LLMs for cross-domain recommendation, in both zero-shot and one-shot settings.** Up to our knowledge, this research direction has been never investigated in literature. **Moreover, we also used LLMs to generate an explanation supporting the recommendations**. With the exception of some early work [42], this direction is scarcely investigated as well.

3 PROBLEM FORMULATION

Unlike traditional RSs, CDR tasks involve two domains: a source domain $\mathcal{D}_\mathcal{S}$ and a target domain $\mathcal{D}_\mathcal{T}$ (*i.e.,* movies and books). Let $\mathcal{I}_\mathcal{S} = \{i_{1S}, \ldots, i_{nS}\}$ and $\mathcal{I}_\mathcal{T} = \{i_{1T}, \ldots, i_{mT}\}$, be the sets of the items in the different domains. For each $i \in \mathcal{I}_\mathcal{S}$ or $\mathcal{I}_\mathcal{T}$ we can assume that a list $\mathcal{F}_i = \{f_1, \ldots, f_z\}$ of descriptive features of the item (i.e., genre, categories, author, etc.) exists.

Next, let $\mathcal{U}_\mathcal{S} = \{u_{1S}, \ldots, i_{jS}\}$ and $\mathcal{U}_\mathcal{T} = \{i_{1T}, \ldots, i_{kT}\}$, be the sets of the *users* who interacted with the items in the source and target domains, respectively. As regards the users, it is worth pointing out that our evaluation will rely on the users who have provided positive ratings in *both* source and target domains[2]. Formally, target users are represented by the overlapping set $\mathcal{U} = \mathcal{U}_\mathcal{S} \cap \mathcal{U}_\mathcal{T}$.

In our workflow, for each $u \in \mathcal{U}$, we first collect their positive and negative preferences. Let $\mathcal{P}_{u,\mathcal{S}}$ and $\mathcal{N}_{u,\mathcal{S}} \subseteq \mathcal{I}_\mathcal{S}$ be the sets of the items liked and disliked by the user u in the *source* domain, and let $\mathcal{R}_{u,\mathcal{T}} \subseteq \mathcal{I}_\mathcal{T}$ be the set of the items to be ranked in the target domain, *i.e.,* the ground truth. Given a generic LLM, we design a prompt that takes as input the items in $\mathcal{P}_{u,\mathcal{S}}$, $\mathcal{N}_{u,\mathcal{S}}$ and $\mathcal{R}_{u,\mathcal{T}}$, together with their descriptive features $\mathcal{F}$, and returns as output a list $\widetilde{\mathcal{R}}$ that re-ranks the items in $\mathcal{R}_{u,\mathcal{T}}$. Formally:

$$\widetilde{\mathcal{R}} = prompt(LLM, \mathcal{P}_{u,\mathcal{S}}, \mathcal{N}_{u,\mathcal{S}}, \mathcal{R}_{u,\mathcal{T}}, \mathcal{F}) \tag{1}$$

In the experiments, we assess to what extent $\widetilde{\mathcal{R}}$ is close to the correct ranking of the items in $\mathcal{R}_{u,\mathcal{T}}$, based on the ground truth.

4 METHODOLOGY

The workflow carried out by our strategy is presented in Figure 1. As shown in the Figure, it can be roughly split into four phases: *(a)* **data pre-processing**, whose goal is to prepare the data in a form that is suitable by a LLM; *(b)* **instruction tuning**, that uses a portion of the data to train the model to correctly tackle a CDR task; *(c)* **prompting**, that aims at building the natural language request for the LLM; *(d)* **refinement and recommendation**, where the output of the LLM is processed in order to extract the list of suggestions together with the explanations.

4.1 Data Pre-Processing

The first step that is carried out is the data pre-processing. In particular, this step takes as input all the data that are available (regardless of belonging to the *source* or *target* domains) and prepares them in a form that is suitable by our workflow. A fundamental preliminary step is the split of the set $\mathcal{U}$, defined as in Section 3, in two disjoint subsets $\mathcal{U}_I$ and $\mathcal{U}_P$. In particular, $\mathcal{U}_I$ will be used for the *instruction tuning* step, while $\mathcal{U}_p$ will be used for the *prompting* phase. It is important to point out that $\mathcal{U}_I \cap \mathcal{U}_T = \emptyset$. This design choice prevents *information leakage* and ensures a fair and unbiased evaluation of the model, since we guarantee that the users used to instruct the model are not used to evaluate the model itself[3]. Moreover, for each user u in either $\mathcal{U}_I$ and $\mathcal{U}_P$, the sets $\mathcal{P}_{u,\mathcal{S}}, \mathcal{N}_{u,\mathcal{S}}, \mathcal{R}_{u,\mathcal{T}}$ are built for every source and target domains. Finally, the set $\mathcal{F}$ containing of the descriptive features of the items is obtained. As previously stated, all these sets represent the *input* of the LLM, in both the instruction tuning and prompting phases.

4.2 Instruction Tuning

As we stated throughout the paper, the goal of *instruction tuning* is to specialize a LLM to tackle new tasks. In our case, we provide a LLM with data regarding CDR scenarios, and we adapt the model.

[2]While this is not a requirement for a *real* CDR settings, in our scenario it becomes mandatory since we need ground truth data, *i.e.,* items evaluated by the user in the target domain, to evaluate the accuracy of the recommendations.

[3]Once the model has been fine-tuned, it is not necessary to repeat this step for every run. It is mandatory only when instruction tuning has to be carried out.

Instruction Input	
System prompt	You are a cross-domain recommender system User provides you a list of items from a source domain that they liked and a list of items such source domain that they disliked Using the same format as the user, you must re-rank the candidate items based on the user liked items and return the *top k <target domain>* Answer using the format: \n Items ranking: <list of top *k* recommended *<domain>* ids>. \n Explanation: <explanation>.
User prompt	I liked these books: Title: The Great Gatsby Brand: F. Scott Fitzgerald Categories: [Classic Literature, Fiction] Title: Harry Potter and the Philosopher's Stone Brand: J.K. Rowling Categories: [Fantasy, Young Adult] Instead, I disliked these books: Title: Fifty Shades of Grey Brand: E.L. James Categories: [Romance, Erotic Fiction] Title: Twilight Brand: Stephenie Meyer Categories: [Fantasy, Romance] Rank and return the top k movies from: Id: 1234 Title: The Great Gatsby Brand: Warner Bros. Pictures Categories: [Drama, Romance] Id: 5678 Title: Harry Potter and the Sorcerer's Stone Brand: Warner Bros. Pictures Categories: [Fantasy, Adventure] Id: 91011 Title: Gone Girl Brand: 20th Century Fox Categories: [Mystery, Thriller]
Instruction Output	
Model Output	Items ranking: 1234, 5678, 91011 Explanation: Based on the user's liked books, which include "The Great Gatsby" and "Harry Potter and the Philosopher's Stone", it's apparent that they enjoy classic books with elements of fiction and fantasy, particularly in the young adult genre.

Table 1: A complete example of prompt *zero-shot setting.*

The output of this step is a new LLM, that is specialized on CDRs. From now on, we will refer to the model as CDR-LLM.

In a nutshell, the Instructor model transforms the data coming from the pre-processing step in a form that is suitable by a LLM, and uses them to fill in a set of *adaptation prompts*, *i.e.*, prompts that are used to adapt the LLM to new tasks, before fine-tuning again the model. In our case, we design a *discrete* prompt, *i.e.*, we lexicalize all the information regarding the task, the preferences of the user and the features of the items as a *string*, and we concatenate all of them. To this end, the process is further split into two steps: first, the structure of the adaptation prompt is designed. In particular, the prompt is organized in three main parts: **System Prompt.** This part provides general instructions about the CDR task and the domains. **User Prompt.** This part details the user interaction history. It is further divided into three parts: (a) *User-liked items* in the source domain, with their features. (b) *User-disliked items* in the source domain, along their features. (c) *Candidate items* in the target domain, along their features. **Model Output.** This specifies the expected output for the LLM, which is the list of re-ranked candidate items in the target domain based on user preferences and an explanation of the provided recommendations.

Then, these parts are concatenated one to each other. Formally:

$$prompt_{adapt} = \text{SystemPrompt} \oplus \text{UserPrompt} \oplus \text{ModelOutput} \quad (2)$$

Where $\oplus$ indicates concatenation. Once the structure of the prompt has been defined, each prompt can be filled in based on the information about user preferences and item features. A complete example of prompt is provided in Table 1.

As shown in the example, the **System Prompt** part remains *static*, while the **User Prompt** is dynamically populated based on the lexicalization of the preferences. Finally, the **Model Output** contains the correct ranking based on the ground truth and an explanation generated by the LLM itself. In other terms, the goal of the adaptation prompts is to provide LLMs with a description of the task together with a set of examples that also contain the *correct* answer the LLM should be able to provide, *i.e.,* the correct ranking and a suitable explanation. In this way, we let the LLM exploit its outstanding generalization capabilities and adapt its behavior to such a new and unseen task, even with a few number of examples.

This process is repeated for all the users in $\mathcal{U}_I$ and over all the combinations of source and target domains. This choice is justified by the fact that we want to specialize the model in a *general* cross-domain recommendation task, regardless of the specific source and target domains. Conversely, as we will show in the next step, in the *prompting* phase a specific couple of source and target domains is picked, since we want to provide CDRs in a specific setting.

Once all the adaptation prompts are generated, our Instruction Data (see Figure 1) are obtained. Such data are used to fine-tune the model and output our CDR-LLM model. As regards this phase, we opted for a *full parameter fine-tuning* of the model. More details about this will be provided next. In particular, the objective of instruction tuning is to minimize the average loss over the entire set of instruction data defined as:

$$\min_{\theta \in \theta_{LLM}} \frac{1}{N} \sum_{i=1}^{N} L(LLM(p_i; \theta), r_i) \quad (3)$$

where p_i is an adaptation prompt (system and user parts), r_i is the expected response (the instruction output)), L is a loss function, (generally the Cross Entropy), θ represents the parameters of the LLM model, and θ_{LLM} is the parameter space. In other terms, by minimizing this loss function, LLMs learn to accurately respond to novel and unseen prompts.

4.3 Prompting LLMs for CDR

Once the CDR-LLM model has been learnt, it is possible to prompt the model to return a list of CDRs based on the preferences of a user. As shown in the workflow, the prompting starts with the Domain Selector component, whose goal is to simply pick a *source* and a *target* domain to be used. Next, given the domains, for each user u in $\mathcal{U}_\mathcal{P}$ we build the sets $\mathcal{P}_{u,\mathcal{S}}, \mathcal{N}_{u,\mathcal{S}}, \mathcal{R}_{u,\mathcal{T}}$, and we collect the descriptive features of the items. Next, we feed CDR-LLM with a personalized prompt based on these sets. In this case, the prompt follows the same structure presented in Table 1, with the exception of the model output, which is not provided. Indeed, in this case the answer will be obviously generated by the model. Formally,

$$prompt_{test} = \text{System Prompt} \oplus \text{User Prompt} \quad (4)$$

In other terms, in this step we simply prompt the LLM we have previously specialized, and we check to what extent it is able to provide accurate CDRs. It is important to emphasize again that the users in $\mathcal{U}_\mathcal{P}$ are completely disjoint from those in $\mathcal{U}_\mathcal{I}$. Moreover, in order to also assess the effectiveness of *in-context learning* in cross-domain recommendations, we repeated the process in a *one-shot* setting *i.e.*, we extended the user prompt by also adding an example of correct recommendation and explanation based on the profile of *another* user in $\mathcal{U}_{\mathcal{P}}$. Due to space reasons, we cannot provide an example of this prompt. However, in the experiments we will compare the behavior of each LLM in both the settings.

4.4 Output Refinement and Recommendation

Despite the instruction tuning phase, the output of CDR-LLM may not always strictly follow the desired format. To address this, we introduced in our workflow an OUTPUT REFINEMENT step that guarantees the model's recommendations align with this format.

The refinement process consists of two steps. First, we analyze the IDs returned as output to avoid *hallucinations* [38] that are typical of LLMs. In particular, we compare the elements returned by CDR-LLM with those in $\mathcal{R}_{u,\mathcal{T}}$, and we filter those that did not occur in the original set of candidate items. Then, the final recommendation list corresponds to the list of filtered IDs, in the same order they appeared in the response of the LLM. Next, the explanation section is analyzed. In this case, we extract all the text after the word *"Explanation:"* occuring in the answer of the LLM. As shown in Table 1, the natural language explanations generated by CDR-LLM represent another distinctive trait of the current approach. Indeed, they clearly show the remarkable ability of the LLMs to seamlessy to exploits the huge body of knowledge they encode to identify patterns and connections between items belonging to different domains. In the experimental evaluation, we will also partially evaluate this aspect.

4.5 Discussion

In this section, we have presented our framework to provide users with CDRs based on LLMs. Before going into details of the experiments, it is necessary to point out the following aspects: (1)We designed a very general framework, that can be adapted to every LLM. Accordingly, in the evaluation we will compare several LLMs at the SOTA, by considering both *open* and *close* LLMs, as well as smaller and larger LLMs (in terms of parameters). This guarantees the solidity of our findings; (2) While the framework is described as a pipeline, it is not necessary (and, sometimes, even not possible) to run all the steps. Indeed, *instruction tuning* can be only run for *open-sourced* LLMs, such as LLaMa. Closed models such as those belonging to the GPT family can be only directly prompted, since it is not possible to fine-tune them. (3) While instruction tuning cannot be run for all the models, this does not jeopardize the outcomes of the experiments. Indeed, as we will show in the next section, the huge number of parameters of GPT (*i.e.*, 25x w.r.t. LLaMa) allows it to correctly handle a CDR task even without the adaptation step. On the contrary, *instruction tuning* is mandatory for smaller models. Without this step, they do not show the capability of generating an answer in the correct format. (4) In our experiments, we run instruction tuning and prompting by considering *all* the features that are available for the items in the source and target domains. While we agree that this choice completely falls into the concept of *prompt engineering* and should be carefully considered, due to space reasons it was not possible to discuss and report more results. This analysis is left as future work.

5 EXPERIMENTAL EVALUATION

Our experiments were designed to answer the following research questions (RQs): **(RQ1)** How do different LLMs perform in the CDR task, in terms of accuracy, ranking, popularity bias and quality of explanations? **(RQ2)** What is the effect of *in-context learning* on the different metrics we compared? **(RQ3)** How does the performance of LLMs in the CDR setting compare to that of SOTA models?

5.1 Experimental Design

Datasets. Following prior works in the area of CDR [3, 52, 58], we conducted experiments on Amazon dataset[4] [33]. These data are particularly suitable for CDR since they have overlapping users across multiple domains. Specifically, we selected three datasets, *i.e.,* "Movies", "CDs", and "Books". Each item is provided with a title, a list of categories and an author and/or a brand. From these datasets, we generated six distinct cross-domain scenarios by interchanging their roles as either source or target domains. All these datasets have been preliminary processed in order to filter out users having: *(a)* less than 5 preferences; *(b)* less than 10 items in the ground truth; *(c)* More than 30 items, in either training or test sets. The first heuristics was set to remove non-significant users. The second was necessary to obtain significant results (we used $k = 5$ as cut-off for our metrics). Finally, the third heuristics was due to the limits in the length of the prompts (4096 tokens at most) that characterize current LLMs. Statistics of the datasets are provided in Table 2. All the datasets are released on our repository[5].

Protocol. To run *instruction tuning*, we picked 5,642 users. Users were randomly selected, as long as the proportion among the different scenarios is guaranteed. The remaining users were used to test the framework in the prompting phase. To prepare the prompts, we established that items rated with a score of 5 out of 5 were considered as *positive*, while all the others were considered as *negative*. This guarantees a more balanced distribution of ratings. As we anticipated throughout the paper, to generate the recommendations we followed the *TestRatings* strategy [4], *i.e.*, we asked the LLM to re-rank the items in the test set, and we returned as recommendation the list of *top-k* items ranked by the LLM, by filtering out non-existing items and hallucinations (see Section 4.4).

Evaluation metrics. The recommendation lists have been evaluated through the ClayRS framework [31], ensuring reproducibility and repeatability. We considered Precision, Recall, F1, nDCG as *accuracy* metrics, as well as the average popularity of the recommended items. Additionally, we evaluate *explanation quality* using perplexity score[6]. This metric reflects the model's uncertainty in predicting the next word in a textual output, i.e, the explanations, in our case. A lower perplexity score indicates that the model is more confident in the generated text, suggesting a more coherent and well-structured explanation. Finally, for each metric, we assessed statistical significance by running *t-test*. Also in this case, we used the implementation available in ClayRS and we compared the scored obtained by each user for each metric.

Implementation Details. To run our experiments, we compared the performance of three SOTA LLMs. As regards GPT, we exploited

[4]http://jmcauley.ucsd.edu/data/amazon/
[5]https://github.com/petruzzellialessandro/RecSys_2024_CDR_LLM
[6]https://huggingface.co/docs/transformers/perplexity

Scenario	#user	Source					Target				
		#items	#interactions	sparsity	#likes_base	#dislikes_base	#items	#interactions	sparsity	#likes_target	#dislikes_target
Movies → Books	2145	10,366	20,289	99.91%	11430	8859	29,325	36,472	99.94%	20464	16008
Books → CDs	1940	19,828	24,049	99.94%	13288	10761	16,745	24,853	99.92%	15997	8856
CDs → Books	2269	14,134	20,294	99.94%	13251	7043	30,565	38,448	99.94%	21507	16941
CDs → Movies	2098	13422	19569	99.93%	12742	6827	12120	27216	99.89%	15205	12011
Books → Movies	1933	20278	24499	99.94%	13663	10836	11819	24959	99.89%	13955	11004
Movies → CDs	1965	9503	18937	99.90%	10517	8420	16659	25317	99.92%	16199	9118

Table 2: Statistics of the Datasets

GPT3.5 Turbo by using GPT's APIs[7]. Next, we exploited LLaMa2-7B-chat[8] for LLaMa, while Mistral-7B Instruct v0.2[9] was used as reference model for Mistral. In the *instruction tuning* step, we fine-tuned these models for 15 training epochs and a batch size of 64. This was distributed as 16 instances per device across 4 Nvidia A100 GPUs using DeepSpeed [44]. We employed the Adam optimizer with a learning rate of 5e-5, weight decay of 1e-4, and a maximum gradient norm of 1.0. The training time per model was approximately 3 hours. As previously stated, instruction tuning was only possible for *open* LLMs such as LLama and Mistral.

Baselines. We evaluate the effectiveness of our proposed methods by comparing them with several competitive baselines available in RecBole [57]. In particular, we considered: **(1) EMCDR** [32]: This method utilizes a multi-layer perceptron to capture non-linear relationships across domains, enabling flexible learning of domain-specific features. **(2) BiTGCF** [28]: This approach relies on graph collaborative filtering and tackles data sparsity by leveraging domain knowledge and facilitating knowledge transfer between domains. It incorporates feature propagation to capture high-order connectivity in user-item graphs. **(3) DeepAPF** [53]: This attention-based CDR method learns non-uniform importance weights for both cross-domain commonalities and domain-specific user interests. **(4) CMF** [43]: This method introduces a relational learning architecture that performs simultaneous matrix factorization on multiple user-item relationships. Crucially, it shares latent factors across these matrices when users participate in multiple domains, allowing information from one domain to enhance prediction accuracy in another. **(5) CLFM** [14]: This method considers the diversity among related domains and incorporates both common and domain-specific rating patterns through joint non-negative matrix tri-factorization, enabling learning of shared patterns across domains while preserving domain-specific information. **(6) SSCDR** [22] This method employs a semi-supervised mapping approach, learning latent vectors for users and items in each domain and then training a cross-domain mapping function using labeled data from overlapping users and unlabeled data from all items; As regards the choice of the baselines, it is necessary to point out that we limit our comparison to the CDR models available in RecBole. This was done to guarantee a fair comparison and the reproducibility. Moreover, as regards potential baselines based on LLMs, methods such as P5 [16] and M6-Rec [9] were not considered since they did not support CDR. No other methods for CDR based on LLMs were found.

5.2 RQ1: Comparing LLMs for CDR

To answer RQ1, we compared the effectiveness of different LLMs in the task of providing users with CDRs. Results are presented in Table 3, 4 and 5, showing the performance in the movies, books and cd domains, respectively. **The first important answer to RQ1 is that the results did not show a LLM clearly beating the others. On the contrary, the results show a connection between the best-performing LLM and the target domain.**

Indeed, as regards the movie domain (Table 3), LLaMa2 consistently surpasses GPT and Mistral on multiple metrics. It achieves superior Precision@5, Recall@5, and F1@5. Notably, the best performance is obtained in the zero-shot setting, and this confirms the remarkable capability of LLM to quickly adapt to new downstream tasks, without any fine tuning. While this is not the focus of this contribution, it is also worth pointing out that zero-shot strategies also allow to implement more sustainable [45] and energy-efficient solutions. Another tendency that emerged from the experiments is the performance of Mistral in recommending less popular items compared to GPT and LLaMa2. It is worth to note that the findings are completely aligned regardless of the *source* domain which is considered, *i.e.,* books or cd. This means that LLMs have the ability of findings connections between the domains by relying on the knowledge that is already encoded in these models. This confirms our intuitions, showing that LLMs can be a suitable solution to bridge heterogeneous domains and provide users with CDRs.

Next, as regards the book domain, the best-performing model is GPT, which overcomes both LLaMa2 and Mistral in all the settings. Such a different behavior can be probably explained in the light of the different body of knowledge that is used to train such models. Indeed, as showed by recent studies [29], it is likely that a significant portion of the text which is used to train GPT-3 comes from (copyrighted) books. Accordingly, it is not surprising that in our setting such a knowledge can be effectively exploited to better provide book recommendations. Moreover, differently from what we noted for movie recommendations, the best performance are obtained in 1-shot settings. This means that GPT can better exploit the information provided through *in-context learning* and even improve the accuracy of the recommendation. This behavior is probably due to the huge amount of parameters encoded in GPT (25x w.r.t. to LLama2, as previously stated) that fosters the generalization and adaptation capability of such models, even without instruction tuning. Finally, as for CD recommendation, Table 5 shows that GPT and LLaMa emerged as best-performing LLMs, significantly overcoming Mistral. However, while GPT achieves the highest Precision@5, Mistral excels again in recommending less popular items. Finally, LLaMa2 maintains good performance across precision, recall, and ranking metrics.

To sum up, this experiment showed that the different LLMs obtain results that are generally comparable. However, an important finding lies in the connection between the knowledge encoded in the LLMs and the overall performance, **since models trained on a larger amount of data of a particular kind (i.e., books on GPT)**

[7]https://platform.openai.com/docs/models/gpt-3-5-turbo
[8]https://huggingface.co/meta-llama/Llama-2-7b-chat-hf/
[9]https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2

tend to have better performance on that particular domain. While this outcome needs to be further investigated, such behavior can be helpful to pick the most effective LLM for a particular task.

Another distinctive trait of our work lies in the generation of a natural language explanation supporting the recommendation. To this end, we evaluated the quality of the LLM-generated explanations based on perplexity values (see Table 6). These values provide insights into the clarity of explanations generated by the LLMs. Comparing perplexity values across different scenarios and LLMs, it is clear that Mistral consistently demonstrates lower perplexity compared to LLaMa2. Furthermore, within each LLM, the 1-shot scenario generally exhibits lower perplexity than the 0-shot scenario. **This is an interesting insight, showing that LLMs are able to improve the quality of the explanations they generate with just one example provided in the prompt.** However, it is important to point out that perplexity measure the uncertainty or surprise of a language model in predicting a sequence of words. While lower perplexity generally corresponds to clearer explanations, it does not include other aspects such as coherence, relevance, or user satisfaction. Thus, although Mistral may have lower perplexity, it does not necessarily imply a *better* explanation. This is confirmed by analyzing the length of the explanations, since those based on Mistral are generally shorter in all the settings.

Accordingly, we also carried out a very preliminary qualitative analysis of the explanations generated by CDR-LLM. An example of explanations is provided in Table 8. As expected, while having lower perplexity, the explanations generated through Mistral tend to be very concise. Conversely, LLaMa2 showed a very interesting capability of improving the quality of the explanation by relying on the example provided in the prompt in the 1-shot setting. Finally, GPT provides very good explanations in both 0-shot and 1-shot scenarios. However, this is not surprising due the high number of parameters of the model. **To conclude, we can state that such a preliminary qualitative analysis showed that the explanations generated by our framework seem to effectively support the recommendations generated by CDR-LLM, finding the connections between source and target domains in an effective way**. A more precise analysis of the explanations, carried out through a user study, will be investigated in future work.

5.3 RQ2: Role of In-Context Learning

Next, to answer RQ2 we compared the performance of each LLM in the 1-shot setting w.r.t. the 0-shot setting. As regards GPT, results show that the introduction of in-context learning improved the accuracy of the recommendations, in terms of Precision, Recall and F1. As previously stated, this is not surprising since the huge amount of parameters of GPT allows this LLM to have a great generalization capability by just exploiting one single example. However, the use of examples tend to increase the popularity bias of CDR based on GPT. Conversely, the analysis of the results on LLaMa and Mistral led to more interesting results. In this case, a different behavior was noted for accuracy (*i.e.,* precision, recall, F1) and ranking metrics. As for the first, a decrease was noted. Conversely, the use of an example allows to increase the average NDCG of the models. Also in this case, the results can be explained in the light of the complexity of such models, which are significantly smaller w.r.t. GPT. As for accuracy metrics, it is likely that one single example is not enough to trigger generalization in LLaMa and Mistral. As future work, we will carry out further experiments with more input examples to support our conjecture. On the contrary, NDCG is positively impacted by the use of *shots*, since the nature of Transformers, which LLMs strongly rely on, is inherently *sequential*. Accordingly, while it is not surprising that a LLM need more information to learn how to *select* good items (as in precision, recall and F1), it is equally not surprising that a task that is closer to the principles of Transformers, *i.e.,* ranking, benefits more of the introduction of examples in the prompts. **Overall, this experiment showed that larger LLMs take more advantage from in-context learning. However, such a strategy has a positive impact on ranking and popularity bias on smaller LLMs as well.**

Scenario	Metric	GPT		LLaMa2		Mistral	
		0-Shot	1-Shot	0-Shot	1-Shot	0-Shot	1-Shot
Books ↓ Movies	Precision@5 ↑	0.5171	0.5221	**0.5316**	0.5207	0.5148	0.5215
	Recall@5 ↑	0.4731	0.4810	**0.5067**	0.4718	0.4628	0.4874
	F1@5 ↑	0.4542	0.4604	**0.4725**	0.4537	0.4473	0.4623
	NDCG@5 ↑	0.8295	**0.8999***	0.8421	0.8601	0.8607	0.8321
	AvgPop@5 ↓	12.0516	12.7576	10.8146	10.4246	10.2708	**10.0620***
CDs ↓ Movies	Precision@5 ↑	0.5153	0.5217	**0.5447**	0.5436	0.5246	0.5317
	Recall@5 ↑	0.4873	0.4915	**0.5313***	0.4877	0.4709	0.5049
	F1@5 ↑	0.4602	0.4666	**0.4901**	0.4699	0.4580	0.4761
	NDCG@5 ↑	0.8377	**0.9117***	0.8476	0.8695	0.8650	0.8337
	AvgPop@5 ↓	12.4699	13.1049	12.1721	11.8152	11.7139	**10.6980***

Table 3: Result on Scenarios with Movies as Target Domain

Scenario	Metric	GPT		LLaMa2		Mistral	
		0-Shot	1-Shot	0-Shot	1-Shot	0-Shot	1-Shot
Movies ↓ Books	Precision@5 ↑	**0.5406**	0.5383	0.5262	0.5174	0.5326	0.5332
	Recall@5 ↑	0.3488	**0.3536**	0.3408	0.3414	0.3462	0.3451
	F1@5 ↑	**0.3942**	**0.3942**	0.3833	0.3804	0.3887	0.3887
	NDCG@5 ↑	0.8326	**0.8980***	0.8403	0.8474	0.8410	0.8301
	AvgPop@5 ↓	2.9333	3.2536	2.5785	**2.5747**	3.0626	2.6859
CDs ↓ Books	Precision@5 ↑	0.5040	**0.5148**	0.5040	0.5098	0.5084	0.5068
	Recall@5 ↑	0.3396	**0.3528**	0.3458	0.3493	0.3393	0.3440
	F1@5 ↑	0.3750	**0.3861**	0.3773	0.3813	0.3773	0.3780
	NDCG@5 ↑	0.8271	**0.9017***	0.8365	0.8444	0.8376	0.8262
	AvgPop@5 ↓	2.9046	3.0115	**2.4776***	2.6149	2.8793	2.8051

Table 4: Result on Scenarios with Books as Target Domain

Scenario	Metric	GPT		LLaMa2		Mistral	
		0-Shot	1-Shot	0-Shot	1-Shot	0-Shot	1-Shot
Movies ↓ CDs	Precision@5 ↑	**0.502**4	0.4785	0.4823	0.4642	0.4385	0.3438
	Recall@5 ↑	**0.6493***	0.6292	0.6261	0.6365	0.6033	0.5887
	F1@5 ↑	**0.5282**	0.5105	0.5116	0.5021	0.4797	0.4003
	NDCG@5 ↑	0.8993	**0.9431***	0.8929	0.8995	0.8874	0.9326
	AvgPop@5 ↓	3.9946	4.2190	3.8950	3.7799	3.9789	**3.1788***
Books ↓ CDs	Precision@5 ↑	0.6459	0.6463	0.6438	**0.6482**	0.6064	0.5683
	Recall@5 ↑	0.4855	0.4861	**0.4921**	0.4769	0.4377	0.3461
	F1@5 ↑	0.5161	0.5179	**0.5196**	0.5153	0.4760	0.3970
	NDCG@5 ↑	0.8871	0.9243	0.8886	0.8984	0.8837	**0.9284***
	AvgPop@5 ↓	4.0859	4.2715	3.9445	3.7175	3.8166	**3.1693***

Table 5: Result on Scenarios with CDs as Target Domain

5.4 RQ3: Comparison to Baselines

Finally, we compared CDR-LLM to several baselines in the area of CDR. Results are presented in 7. For each setting, the baselines are compared to the best-performing configuration emerged from RQ1.

As shown in the Table, our approach based on LLMs tend to outperform the baselines in most of the experimental settings. In particular, methods based on LLMs beat *all* the baselines on *all* the metrics for the CD-Movies, Movies-Books and Movies-CD setting, and most of the baselines in the Books-CD and CD-Books. Overall, in 4 out of 6 comparisons our framework got the best results in

Scenario	Metric	LLaMa2		Mistral	
		0-Shot	1-Shot	0-Shot	1-Shot
Books → Movies	Perplexity	16.41	10.87	5.76	4.11
	Mean length	280.90	275.06	176.67	153.21
	Std length	184.25	190.29	71.05	51.43
CDs → Movies	Perplexity	20.10	11.97	6.88	4.49
	Mean length	270.10	259.12	174.93	149.93
	Std length	170.00	174.52	85.59	45.34
CDs → Books	Perplexity	13.85	10.24	5.32	4.05
	Mean length	253.83	261.32	178.53	177.37
	Std length	148.25	154.85	80.39	84.02
Movies → Books	Perplexity	13.83	10.23	5.29	3.95
	Mean length	260.50	250.26	170.70	177.78
	Std length	123.85	138.99	72.91	81.39
Movies → CDs	Perplexity	17.89	11.19	6.30	4.31
	Mean length	298.20	295.24	180.05	175.94
	Std length	173.06	191.02	91.11	79.67
Books → CDs	Perplexity	15.71	10.70	5.64	4.13
	Mean length	310.00	308.77	179.55	147.22
	Std length	200.06	216.51	96.48	62.18

Table 6: Perplexity and length statistics of explanations.

Dataset	Model	Precision@5 ↑	Recall@5 ↑	F1@5 ↑	NDCdG@5 ↑	AvgPop@5 ↓
Books ↓ Movies	DeepAPF	0.5250	0.4807	0.4631	0.8291	16.8534
	EMCDR	0.5274	0.4903	0.4664	0.8323	11.9061
	SSCDR	**0.5423**	0.5081	**0.4815**	0.8339	21.9568
	CMF	0.5303	0.4908	0.4684	0.8323	18.6123
	BiTGCF	0.5136	0.4702	0.4515	0.8274	13.1767
	CLFM	0.5386	0.5018	0.4763	0.8352	19.9481
	LLaMa2-0-shot	0.5316	**0.5067**	0.4725	**0.8421**	**10.8146***
CDs ↓ Movies	DeepAPF	0.5232	0.4996	0.4692	0.8396	18.8728
	EMCDR	0.5207	0.4913	0.4650	0.8351	12.2243
	SSCDR	0.5421	0.5171	0.4855	0.8424	23.1046
	CMF	0.5353	0.5116	0.4807	0.8421	19.4315
	BiTGCF	0.5186	0.4955	0.4646	0.8366	17.3814
	CLFM	0.5334	0.5104	0.4791	0.8418	20.1796
	LLaMa2-0-shot	**0.5447**	**0.5313**	**0.4901**	**0.8476***	**12.1721***
Movies ↓ Books	DeepAPF	0.5291	0.3358	0.3834	0.8298	4.2974
	EMCDR	0.5343	0.3482	0.3902	0.8350	**3.2135**
	SSCDR	0.5262	0.3380	0.3825	0.8268	7.2063
	CMF	0.5268	0.3418	0.3846	0.8293	5.9964
	BiTGCF	0.5246	0.3448	0.3847	0.8286	5.3548
	CLFM	0.5358	0.3492	0.3912	0.8308	6.5011
	GPT-1-Shot	**0.5383**	**0.3536**	**0.3942**	**0.8980**	3.2536
CDs ↓ Books	DeepAPF	0.5105	0.3470	0.3816	0.8262	**2.0268***
	EMCDR	0.5042	0.3384	0.3752	0.8256	3.2876
	SSCDR	0.5120	0.3501	0.3839	0.8192	7.0432
	CMF	0.5090	0.3507	0.3823	0.8223	5.5861
	BiTGCF	0.5053	0.3404	0.3761	0.8229	5.5628
	CLFM	0.5095	**0.3532**	0.3827	0.8179	6.1075
	GPT-1-Shot	**0.5148**	0.3528	**0.3861**	**0.9017***	3.0115
Movies ↓ CDs	DeepAPF	0.4723	0.6218	0.5033	0.8807	5.2178
	EMCDR	0.4815	0.6221	0.5064	0.8869	4.5212
	SSCDR	0.4745	0.6191	0.5007	0.8780	7.4475
	CMF	0.4801	0.6242	0.5069	0.8833	6.1815
	BiTGCF	0.4878	0.6334	0.5146	0.8879	4.9420
	CLFM	0.4830	0.6257	0.5076	0.8818	6.3444
	GPT-0-Shot	**0.5024**	**0.6493**	**0.5282**	**0.8993**	**3.9946***
Books ↓ CDs	DeepAPF	0.6377	0.4786	0.5104	0.8813	5.2003
	EMCDR	0.6338	0.4758	0.5074	0.8788	4.5029
	SSCDR	0.6349	0.4723	0.5067	0.8799	7.9277
	CMF	0.6408	0.4825	0.5139	0.8814	6.4865
	BiTGCF	**0.6456**	**0.4861**	**0.5188**	0.8858	5.2717
	CLFM	0.6408	0.4792	0.5131	0.8840	5.4844
	GPT-1-Shot	0.6463	**0.4861**	0.5179	**0.9243***	**4.2715***

Table 7: Comparison of CDR baselines with LLM's best result. Boldface metric indicates highest score. *: significantly better than all baselines with t-test p<0.05.

terms of Precision and F1, while in 5 out of 6 comparisons we obtained the best Recall. However, the most interesting findings regard the impact of our strategy in terms of ranking of the items. Indeed, we always obtained the best results in terms of NDCG (with a statistically significant increase in 3 out of 6 scenarios). Finally, it is also remarkable the impact in terms of AveragePopularity. These results definitely confirm the effectiveness of our approach, since we showed that: *(a)* the sequential nature of Transformers is particularly suitable to tackle CDR as a ranking task, *i.e.,* to put the items in the right sequence, since it perfectly borrows the principles of Transformers to CDR. *(b)* the knowledge encoded in the LLMs, as well as the ability of finding non-trivial connections between items in the source and target domains leads to recommendations that are less prone to popularity bias. **Overall, the use of LLMs for CDR seem to be very promising in *multi-objective* fashion, since our recommendations showed a good trade-off between the quality of the rankings and their popularity bias. Moreover, the opportunity of generating a natural language explanation is another trait that supports the use of LLMs for CDR.**

6 CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a framework for explainable CDR based on LLMs. Our framework first exploits instruction tuning to adapt LLMs to better handle a CDR task, then uses prompt engineering to ask a LLM retun a list of recommendations together with a natural language explanation. Our experiments showed that our framework can provide results that often beat several SOTA baselines for CDR recommendations, and can also provide a suitable explanation that can make the CDR process more transparent. Moreover, results showed the importance of in-context learning, especially for ranking and explanation. As future work, we will analyze different strategies for optimization [13], instruction tuning and prompting, in order to better drive LLMs. Of course, we will also evaluate more LLMs and different domains.

ACKNOWLEDGMENTS

We acknowledge the support of the PNRR project FAIR - Future AI Research (PE00000013), Spoke 6 - Symbiotic AI (CUP H97G22000210007) under the NRRP MUR program funded by the NextGenerationEU and project PHaSE (CUP H53D23003530006) - Promoting Healthy and Sustainable Eating through Interactive and Explainable AI Methods, funded by MUR under the PRIN program

Model	0-shot	1-Shot
LLaMa2	Based on the user's liked CDs, which include rock, progressive rock, and alternative rock genres, I have ranked the movies accordingly.	Based on the user's liked books, which include genres like literature & fiction, contemporary, and humor, I have recommended CDs that are related to the metal genre. This recommendation is made based on the assumption that the user's preference for literature & fiction and contemporary genres may align with the intense and energetic nature of metal music. Additionally, the recommendation includes CDs from bands like Iced Earth and Shadows Fall, which are known for their technical and progressive musical style, which might appeal to the user's interest in contemporary literature.
Mistral	Based on the user's liked and disliked movies, we can infer that the user has a preference for animation and children's books. Therefore, the recommended books are selected based on their categories.	Based on the userś liked movies, which include "Birdy VHS" and "Mean Streets VHS," it seems that the user has an interest in movies with a gritty and intense atmosphere. Therefore, I would recommend books that have a similar theme or tone
GPT	Based on your liked books, which are mainly children's books and literature, I have recommended classical music albums that may resonate with the storytelling and emotional depth found in those genres. The recommended albums feature vocal performances and compositions that could evoke similar feelings to the narratives and themes present in the books you enjoyed.	Based on your liked books, which include a mix of fiction, humor, and biographies, the recommended movies are aligned with alternative rock and indie music genres, which often offer a diverse and unique storytelling experience through their lyrics and melodies. These recommendations aim to provide you with a creative and thought-provoking entertainment option that resonates with the variety in your reading preferences.

Table 8: Examples of explanations from LLMs for *0-shot* and *1-shot* recommendation tasks.

REFERENCES

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[2] Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. *arXiv preprint arXiv:1903.00089* (2019).

[3] Nawaf Alharbi and Doina Caragea. 2022. Cross-Domain Attentive Sequential Recommendations based on General and Current User Preferences (CD-ASR). In *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology* (Melbourne, VIC, Australia) *(WI-IAT '21)*. Association for Computing Machinery, New York, NY, USA, 48–55. https://doi.org/10.1145/3486622.3493949

[4] Alejandro Bellogin, Pablo Castells, and Ivan Cantador. 2011. Precision-oriented evaluation of recommender systems: an algorithmic comparison. In *Proceedings of the fifth ACM conference on Recommender systems*. 333–336.

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[6] Iván Cantador, Ignacio Fernández-Tobías, Shlomo Berkovsky, and Paolo Cremonesi. 2015. Cross-domain recommender systems. *Recommender systems handbook* (2015), 919–959.

[7] Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, George Foster, Llion Jones, Niki Parmar, Mike Schuster, Zhifeng Chen, et al. 2018. The best of both worlds: Combining recent advances in neural machine translation. *arXiv preprint arXiv:1804.09849* (2018).

[8] Paolo Cremonesi and Massimo Quadrana. 2014. Cross-domain recommendations without overlapping data: Myth or reality?. In *Proceedings of the 8th ACM Conference on Recommender systems*. 297–300.

[9] Zeyu Cui, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. 2022. M6-rec: Generative pretrained language models are open-ended recommender systems. *arXiv preprint arXiv:2205.08084* (2022).

[10] Maurizio Ferrari Dacrema, Iván Cantador, Ignacio Fernández-Tobías, Shlomo Berkovsky, and Paolo Cremonesi. 2012. Design and evaluation of cross-domain recommender systems. In *Recommender Systems Handbook*. Springer, 485–516.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[12] Ignacio Fernández-Tobías, Iván Cantador, Paolo Tomeo, Vito Walter Anelli, and Tommaso Di Noia. 2019. Addressing the user cold start with cross-domain collaborative filtering: exploiting item metadata in matrix factorization. *User modeling and user-adapted interaction* 29 (2019), 443–486.

[13] Allegra De Filippo, Michele Lombardi, and Michela Milano. 2021. Integrated Offline and Online Decision Making under Uncertainty. *J. Artif. Intell. Res.* 70 (2021), 77–117. https://doi.org/10.1613/JAIR.1.12333

[14] Sheng Gao, Hao Luo, Da Chen, Shantao Li, Patrick Gallinari, and Jun Guo. 2013. Cross-Domain Recommendation via Cluster-Level Latent Factor Model. In *Machine Learning and Knowledge Discovery in Databases*, Hendrik Blockeel, Kristian Kersting, Siegfried Nijssen, and Filip Železný (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 161–176.

[15] Yunfan Gao, Tao Sheng, Youlin Xiang, Yun Xiong, Haofen Wang, and Jiawei Zhang. 2023. Chat-rec: Towards interactive and explainable llms-augmented recommender system. *arXiv preprint arXiv:2303.14524* (2023).

[16] Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2022. Recommendation as Language Processing (RLP): A Unified Pretrain, Personalized Prompt & Predict Paradigm (P5). In *Proceedings of the 16th ACM Conference on Recommender Systems* (<conf-loc>, <city>Seattle</city>, <state>WA</state>, <country>USA</country>, </conf-loc>) *(RecSys '22)*. Association for Computing Machinery, New York, NY, USA, 299–315. https://doi.org/10.1145/3523227.3546767

[17] Wenxing Hong, Nannan Zheng, Ziang Xiong, and Zhiqiang Hu. 2020. A parallel deep neural network using reviews and item metadata for cross-domain recommendation. *IEEE Access* 8 (2020), 41774–41783.

[18] Dichao Hu. 2020. An introductory survey on attention mechanisms in NLP problems. In *Intelligent Systems and Applications: Proceedings of the 2019 Intelligent Systems Conference (IntelliSys) Volume 2*. Springer, 432–448.

[19] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685* (2021).

[20] Anthony Jameson, Martijn C Willemsen, Alexander Felfernig, Marco de Gemmis, Pasquale Lops, Giovanni Semeraro, and Li Chen. 2015. Human decision making and recommender systems. In *Recommender Systems Handbook*. ACM New York, NY, USA, 611–648.

[21] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio 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]

[22] SeongKu Kang, Junyoung Hwang, Dongha Lee, and Hwanjo Yu. 2019. Semi-Supervised Learning for Cross-Domain Recommendation to Cold-Start Users. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (Beijing, China) *(CIKM '19)*. Association for Computing Machinery, New York, NY, USA, 1563–1572. https://doi.org/10.1145/3357384.3357914

[23] Wang-Cheng Kang, Jianmo Ni, Nikhil Mehta, Maheswaran Sathiamoorthy, Lichan Hong, Ed Chi, and Derek Zhiyuan Cheng. 2023. Do LLMs Understand User Preferences? Evaluating LLMs On User Rating Prediction. arXiv:2305.06474 [cs.IR]

[24] Muhammad Murad Khan, Roliana Ibrahim, and Imran Ghani. 2017. Cross domain recommender systems: A systematic literature review. *ACM Computing Surveys (CSUR)* 50, 3 (2017), 1–34.

[25] Philippe Laban, Wojciech Kryściński, Divyansh Agarwal, Alexander Richard Fabbri, Caiming Xiong, Shafiq Joty, and Chien-Sheng Wu. 2023. SummEdits: Measuring LLM ability at factual reasoning through the lens of summarization. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. 9662–9676.

[26] Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691* (2021).

[27] Junyang Lin, An Yang, Jinze Bai, Chang Zhou, Le Jiang, Xianyan Jia, Ang Wang, Jie Zhang, Yong Li, Wei Lin, et al. 2021. M6-10t: A sharing-delinking paradigm for efficient multi-trillion parameter pretraining. *arXiv preprint arXiv:2110.03888* (2021).

[28] Meng Liu, Jianjun Li, Guohui Li, and Peng Pan. 2020. Cross Domain Recommendation via Bi-directional Transfer Graph Collaborative Filtering Networks. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management* (Virtual Event, Ireland) *(CIKM '20)*. Association for Computing Machinery, New York, NY, USA, 885–894. https://doi.org/10.1145/3340531.3412012

[29] Yang Liu, Yuanshun Yao, Jean-Francois Ton, Xiaoying Zhang, Ruocheng Guo Hao Cheng, Yegor Klochkov, Muhammad Faaiz Taufiq, and Hang Li. 2023. Trustworthy LLMs: a Survey and Guideline for Evaluating Large Language Models' Alignment.

arXiv preprint arXiv:2308.05374 (2023).
[30] Pasquale Lops, Marco de Gemmis, Giovanni Semeraro, Cataldo Musto, Fedelucio Narducci, and Massimo Bux. 2009. A Semantic Content-Based Recommender System Integrating Folksonomies for Personalized Access. In *Web Personalization in Intelligent Environments*, Giovanna Castellano, Lakhmi C. Jain, and Anna Maria Fanelli (Eds.). Studies in Computational Intelligence, Vol. 229. 27–47. https://doi.org/10.1007/978-3-642-02794-9_2
[31] Pasquale Lops, Marco Polignano, Cataldo Musto, Antonio Silletti, and Giovanni Semeraro. 2023. ClayRS: An end-to-end framework for reproducible knowledge-aware recommender systems. *Information Systems* 119 (2023), 102273.
[32] Tong Man, Huawei Shen, Xiaolong Jin, and Xueqi Cheng. 2017. Cross-Domain Recommendation: An Embedding and Mapping Approach. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*. 2464–2470. https://doi.org/10.24963/ijcai.2017/343
[33] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-Based Recommendations on Styles and Substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Santiago, Chile) *(SIGIR '15)*. Association for Computing Machinery, New York, NY, USA, 43–52. https://doi.org/10.1145/2766462.2767755
[34] Cataldo Musto, Tiziano Franza, Giovanni Semeraro, Marco de Gemmis, and Pasquale Lops. 2018. Deep Content-based Recommender Systems Exploiting Recurrent Neural Networks and Linked Open Data. In *Adjunct Publication of the 26th Conference on User Modeling, Adaptation and Personalization, UMAP 2018, Singapore, July 08-11, 2018*, Tanja Mitrovic, Jie Zhang, Li Chen, and David Chin (Eds.). ACM, 239–244. https://doi.org/10.1145/3213586.3225230
[35] Cataldo Musto, Giovanni Semeraro, Pasquale Lops, and Marco de Gemmis. 2014. Combining Distributional Semantics and Entity Linking for Context-Aware Content-Based Recommendation. In *User Modeling, Adaptation, and Personalization - 22nd International Conference, UMAP 2014, Aalborg, Denmark, July 7-11, 2014. Proceedings (Lecture Notes in Computer Science, Vol. 8538)*, Vania Dimitrova, Tsvi Kuflik, David Chin, Francesco Ricci, Peter Dolog, and Geert-Jan Houben (Eds.). Springer, 381–392. https://doi.org/10.1007/978-3-319-08786-3_34
[36] Alessandro Petruzzelli, Alessandro Francesco Maria Martina, Giuseppe Spillo, Cataldo Musto, Marco de Gemmis, Pasquale Lops, and Giovanni Semeraro. 2024. Improving Transformer-based Sequential Conversational Recommendations through Knowledge Graph Embeddings. In *Proceedings of the 32nd ACM Conference on User Modeling, Adaptation and Personalization, UMAP 2024, Cagliari, Italy, July 1-4, 2024*. ACM, 172–182. https://doi.org/10.1145/3627043.3659565
[37] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research* 21, 140 (2020), 1–67.
[38] Vipula Rawte, Amit Sheth, and Amitava Das. 2023. A Survey of Hallucination in Large Foundation Models. arXiv:2309.05922 [cs.AI]
[39] Paul Resnick and Hal R Varian. 1997. Recommender systems. *Commun. ACM* 40, 3 (1997), 56–58.
[40] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2022. Recommender Systems: Techniques, Applications, and Challenges. *Recommender Systems Handbook* (2022), 1–35.
[41] Scott Sanner, Krisztian Balog, Filip Radlinski, Ben Wedin, and Lucas Dixon. 2023. Large Language Models are Competitive Near Cold-start Recommenders for Language- and Item-based Preferences. In *Proceedings of the 17th ACM Conference on Recommender Systems* (Singapore, Singapore) *(RecSys '23)*. Association for Computing Machinery, New York, NY, USA, 890–896. https://doi.org/10.1145/3604915.3608845
[42] Itallo Silva, Alan Said, Leandro Balby Marinho, and Martijn Willemsen. 2022. Leveraging Large Language Models for Recommendation and Explanation. In *Proceedings of the 10th Joint Workshop on Interfaces and Human Decision Making for Recommender Systems (IntRS 2023)*. 74–81.
[43] Ajit P. Singh and Geoffrey J. Gordon. 2008. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Las Vegas, Nevada, USA) *(KDD '08)*. Association for Computing Machinery, New York, NY, USA, 650–658. https://doi.org/10.1145/1401890.1401969
[44] Shuaiwen Leon Song, Bonnie Kruft, Minjia Zhang, Conglong Li, Shiyang Chen, Chengming Zhang, Masahiro Tanaka, Xiaoxia Wu, Jeff Rasley, Ammar Ahmad Awan, Connor Holmes, Martin Cai, Adam Ghanem, Zhongzhu Zhou, Yuxiong He, Pete Luferenko, Divya Kumar, Jonathan Weyn, Ruixiong Zhang, Sylwester Klocek, Volodymyr Vragov, Mohammed AlQuraishi, Gustaf Ahdritz, Christina Floristean, Cristina Negri, Rao Kotamarthi, Venkatram Vishwanath, Arvind Ramanathan, Sam Foreman, Kyle Hippe, Troy Arcomano, Romit Maulik, Maxim Zvyagin, Alexander Brace, Bin Zhang, Cindy Orozco Bohorquez, Austin Clyde, Bharat Kale, Danilo Perez-Rivera, Heng Ma, Carla M. Mann, Michael Irvin, J. Gregory Pauloski, Logan Ward, Valerie Hayot, Murali Emani, Zhen Xie, Diangen Lin, Maulik Shukla, Ian Foster, James J. Davis, Michael E. Papka, Thomas Brettin, Prasanna Balaprakash, Gina Tourassi, John Gounley, Heidi Hanson, Thomas E Potok, Massimiliano Lupo Pasini, Kate Evans, Dan Lu, Dalton Lunga, Junqi Yin, Sajal Dash, Feiyi Wang, Mallikarjun Shankar, Isaac Lyngaas, Xiao Wang, Guojing Cong, Pei Zhang, Ming Fan, Siyan Liu, Adolfy Hoisie, Shinjae Yoo, Yihui Ren, William Tang, Kyle Felker, Alexey Svyatkovskiy, Hang Liu, Ashwin Aji, Angela Dalton, Michael Schulte, Karl Schulz, Yuntian Deng, Weili Nie, Josh Romero, Christian Dallago, Arash Vahdat, Chaowei Xiao, Thomas Gibbs, Anima Anandkumar, and Rick Stevens. 2023. DeepSpeed4Science Initiative: Enabling Large-Scale Scientific Discovery through Sophisticated AI System Technologies. arXiv:2310.04610 [cs.AI]
[45] Giuseppe Spillo, Allegra De Filippo, Cataldo Musto, Michela Milano, and Giovanni Semeraro. 2023. Towards Sustainability-aware Recommender Systems: Analyzing the Trade-off Between Algorithms Performance and Carbon Footprint. In *Proceedings of the 17th ACM Conference on Recommender Systems, RecSys 2023, Singapore, Singapore, September 18-22, 2023*, Jie Zhang, Li Chen, Shlomo Berkovsky, Min Zhang, Tommaso Di Noia, Justin Basilico, Luiz Pizzato, and Yang Song (Eds.). ACM, 856–862. https://doi.org/10.1145/3604915.3608840
[46] Giuseppe Spillo, Cataldo Musto, Marco Polignano, Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro. 2023. Combining Graph Neural Networks and Sentence Encoders for Knowledge-aware Recommendations. In *Proceedings of the 31st ACM Conference on User Modeling, Adaptation and Personalization, UMAP 2023, Limassol, Cyprus, June 26-29, 2023*. ACM, 1–12. https://doi.org/10.1145/3565472.3592965
[47] Liyan Tang, Igor Shalyminov, Amy Wing-mei Wong, Jon Burnsky, Jake W Vincent, Yu'an Yang, Siffi Singh, Song Feng, Hwanjun Song, Hang Su, et al. 2024. TofuEval: Evaluating Hallucinations of LLMs on Topic-Focused Dialogue Summarization. *arXiv preprint arXiv:2402.13249* (2024).
[48] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).
[49] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* (2023).
[50] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).
[51] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. 2023. Autogen: Enabling next-gen llm applications via multi-agent conversation framework. *arXiv preprint arXiv:2308.08155* (2023).
[52] Zitao Xu, Weike Pan, and Zhong Ming. 2023. A multi-view graph contrastive learning framework for cross-domain sequential recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*. 491–501.
[53] Huan Yan, Xiangning Chen, Chen Gao, Yong Li, and Depeng Jin. 2019. DeepAPF: Deep Attentive Probabilistic Factorization for Multi-site Video Recommendation. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, 1459–1465. https://doi.org/10.24963/ijcai.2019/202
[54] Junjie Ye, Xuanting Chen, Nuo Xu, Can Zu, Zekai Shao, Shichun Liu, Yuhan Cui, Zeyang Zhou, Chao Gong, Yang Shen, et al. 2023. A comprehensive capability analysis of gpt-3 and gpt-3.5 series models. *arXiv preprint arXiv:2303.10420* (2023).
[55] Gokul Yenduri, Ramalingam M, Chemmalar Selvi G, Supriya Y, Gautam Srivastava, Praveen Kumar Reddy Maddikunta, Deepti Raj G, Rutvij H Jhaveri, Prabadevi B, Weizheng Wang, Athanasios V. Vasilakos, and Thippa Reddy Gadekallu. 2023. Generative Pre-trained Transformer: A Comprehensive Review on Enabling Technologies, Potential Applications, Emerging Challenges, and Future Directions. arXiv:2305.10435 [cs.CL]
[56] Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. 2024. Instruction Tuning for Large Language Models: A Survey. arXiv:2308.10792 [cs.CL]
[57] Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Kaiyuan Li, Yushuo Chen, Yujie Lu, Hui Wang, Changxin Tian, Xingyu Pan, Yingqian Min, Zhichao Feng, Xinyan Fan, Xu Chen, Pengfei Wang, Wendi Ji, Yaliang Li, Xiaoling Wang, and Ji-Rong Wen. 2021. Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms. In *CIKM*.
[58] Xiaolin Zheng, Jiajie Su, Weiming Liu, and Chaochao Chen. 2022. DDGHM: dual dynamic graph with hybrid metric training for cross-domain sequential recommendation. In *Proceedings of the 30th ACM International Conference on Multimedia*. 471–481.