Large Language Models are Competitive Near Cold-start Recommenders for Language- and Item-based Preferences

Scott Sanner*
ssanner@mie.utoronto.ca
University of Toronto
Toronto, Canada

Krisztian Balog
krisztianb@google.com
Google
Stavanger, Norway

Filip Radlinski
filiprad@google.com
Google
London, United Kingdom

Ben Wedin
wedin@google.com
Google
Cambridge, MA, United States

Lucas Dixon
ldixon@google.com
Google
Paris, France

ABSTRACT

Traditional recommender systems leverage users' item preference history to recommend novel content that users may like. However, modern dialog interfaces that allow users to express language-based preferences offer a fundamentally different modality for preference input. Inspired by recent successes of prompting paradigms for large language models (LLMs), we study their use for making recommendations from both item-based and language-based preferences in comparison to state-of-the-art item-based collaborative filtering (CF) methods. To support this investigation, we collect a new dataset consisting of both item-based and language-based preferences elicited from users along with their ratings on a variety of (biased) recommended items and (unbiased) random items. Among numerous experimental results, we find that LLMs provide competitive recommendation performance for *pure language-based preferences* (no item preferences) in the near cold-start case in comparison to item-based CF methods, despite having no supervised training for this specific task (zero-shot) or only a few labels (few-shot). This is particularly promising as language-based preference representations are more explainable and scrutable than item-based or vector-based representations.

CCS CONCEPTS

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

KEYWORDS

recommendation; transparency; scrutability; natural language

ACM Reference Format:
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 *Seventeenth ACM Conference on Recommender Systems (RecSys '23), September 18–22, 2023, Singapore, Singapore.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3604915.3608845

*Work done while on sabbatical at Google.

RecSys '23, September 18–22, 2023, Singapore, Singapore

ACM ISBN 979-8-4007-0241-9/23/09.
https://doi.org/10.1145/3604915.3608845

1 INTRODUCTION

The use of language in recommendation scenarios is not a novel concept. Content-based recommenders have been utilizing text associated with items, such as item descriptions and reviews, for about three decades [29]. However, recent advances in conversational recommender systems have placed language at the forefront, as a natural and intuitive means for users to express their preferences and provide feedback on the recommendations they receive [15, 24]. Most recently, the concept of natural language (NL) user profiles, where users express their preferences as NL statements has been proposed [37]. The idea of using text-based user representations is appealing for several reasons: it provides full transparency and allows users to control the system's personalization. Further, in a (near) cold-start setting, where little to no usage data is available, providing a NL summary of preferences may enable a personalized and satisfying experience for users. Yet, controlled quantitative comparisons of such NL preference descriptions against traditional item-based approaches are very limited. Thus, the main research question driving this study is the following: How effective are prompting strategies with large language models (LLMs) for recommendation from natural language-based preference descriptions in comparison to collaborative filtering methods based solely on item ratings?

We address the task of *language-based item recommendation* by building on recent advances in LLMs and prompting-based paradigms that have led to state-of-the-art results in a variety of natural language tasks, and which permit us to exploit rich positive and negative descriptive content and item preferences in a unified framework. We contrast these novel techniques with traditional language-based approaches using information retrieval techniques [3] as well as collaborative filtering-based approaches [14, 42]. Being a novel task, there is no dataset for language-based item recommendation. As one of our main contributions, we present a data collection protocol and build a test collection that comprises natural language descriptions of preferences as well as item ratings. In doing so, we seek to answer the following research questions:

- **RQ1:** Are preferences expressed in natural language sufficient as a replacement for items for (especially) near cold-start recommendation, and how much does performance improve when language is combined with items?
- **RQ2:** How do LLM-based recommendation methods compare with item-based collaborative filtering methods?
- **RQ3:** Which LLM prompting style, be it completion, instructions, or few-shot prompts, performs best?
- **RQ4:** Does the inclusion of natural language *dis*preferences improve language-based recommendation?

Our main contributions are (1) We devise an experimental design that allows language-based item recommendation to be directly compared with state-of-the-art item-based recommendation approaches, and present a novel data collection protocol (Section 3); (2) We propose various prompting methods for LLMs for the task of language-based item recommendation (Section 4); (3) We experimentally compare the proposed prompt-based methods against a set of strong baselines, including both text-based and item-based approaches (Section 5). Ultimately, we observe that LLM-based recommmendation from pure language-based preference descriptions provides a competitive near cold-start recommender system that is based on an explainable and scrutable language-based preference representation.

2 RELATED WORK

Item-Based Recommendation. Traditional recommender systems rely on item ratings. For a new user, these can be provided over time as the user interacts with the recommender, although this means initial performance is poor. Thus, preferences are often solicited with a questionnaire for new users [22, 39, 41]. There has also been work looking at other forms of item-based preferences such as relative preferences between items [10, 39], although approaches that rely on individual item ratings dominate the literature.

Given a corpus of user-item ratings, very many recommendation algorithms exist. These range from methods such as item-based k-Nearest Neighbors [40], where simple similarity to existing users is exploited, to matrix factorization approaches that learn a vector representation for the user [23, 34], through to deep learning and autoencoder approaches that jointly learn user and item vector embeddings [8, 19, 28]. Interestingly, the EASE algorithm [42] is an autoencoder approach that has been found to perform on par with much more complex state-of-the-art approaches.

Natural Language in Recommendation. Following the proposals in [2, 37] to model preferences solely in scrutable natural language, recent work has explored the use of tags as surrogates for NL descriptions with promising results [31]. This contrasts with, for instance Hou et al. [20], who input a (sequence) of natural language item descriptions into an LLM to produce an (inscrutable) user representation for recommendation. Other recent work has sought to use rich, descriptive natural language as the basis for recommendations. At one extreme, we have narrative-driven recommendations [4] that assume very verbose descriptions of specific contextual needs. In a similar vein, user-studies of NL use in recommendation [26] identify a rich taxonomy of recommendation intents and also note that speech-based elicitation is generally more verbose and descriptive than text-based elicitation. In this work, however, we return to the proposal in [37] and assume the user provides a more general-purpose language-based description of their preferences and dispreferences for the purpose of recommendation.

Recently, researchers have begun exploring use of language models (LMs) for recommendation tasks [13]. Radlinski et al. [37] present a theoretical motivation for why LLMs may be useful for recommendations and provide an example prompt, but do not conduct any quantitative evaluation. Mysore et al. [32] generate preference narratives from ratings and reviews, using the narratives to recommend from held-out items. Penha and Hauff [36] show that off-the-shelf pretrained BERT [12] contains both collaborative- and content-based knowledge about items to recommend. They also demonstrate that BERT outperforms information retrieval (IR) baselines for recommendation from language-based descriptions. However, they do not assess the relative performance of language- vs. item-based recommendation from LMs (for which we curate a dataset specifically for this purpose), nor does BERT's encoder-only LM easily permit doing this in a unified prompting framework that we explore here. RecoBERT [30] leverages a custom-trained LM for deriving the similarity between text-based item and description pairs, with the authors finding that this outperforms traditional IR methods. Hou et al. [21] focus on item-based recommendation, with an in-context learning (ICL) approach similar in spirit to our item-only few-shot approach. Similarly, Kang et al. [27] use an LLM to predict ratings of target items. Finally, ReXPlug [17] exploits pretrained LMs to produce explainable recommendations by generating synthetic reviews on behalf of the user. None of these works, however, explore *prompting strategies* in large LMs to *translate actual natural language preferences into new recommendations* compared directly to item-based approaches.

Further, we are unaware of any datasets that capture a user's detailed preferences in natural language, and attempt to rate recommendations on unseen items. Existing datasets such as [2, 7] tend to rely on much simpler characterizations.

Prompting in Large Language Models. Large language models (LLMs) are an expanding area of research with numerous exciting applications. Beyond traditional natural language understanding tasks like summarization, relation mapping, or question answering, LLMs have also proved adept at many tasks such as generating code, generating synthetic data, and multi-lingual tasks [1, 5, 9]. How to prompt these models to generate the best results is a continuing topic of research. Early prompting approaches relied on few-shot prompting, where a small set of training input-output pairs are prepended to the actual input [6]. Through additional tuning of pre-trained models on tasks described via instructions, LLMs also achieve impressive performance in the zero-shot setting (i.e., models are given a task and inputs, without any previous training examples) [44]. Geng et al. [16] test a variety of prompting techniques with a relatively small (less than one billion parameter) LLM trained on a collection of recommendation tasks, finding promising results across multiple tasks and domains, primarily by using item ratings as input.

3 EXPERIMENTAL SETUP

To study the relationship between item-based and language-based preferences, and their utility for recommendation, we require a

parallel corpus from *the same raters* providing both types of information that is *maximally consistent*. There is a lack of existing parallel corpora of this nature, therefore a key contribution of our work is an experiment design that allows such consistent information to be collected. Specifically, we designed a two-phase user study where raters were (1) asked to rate items, *and* to describe their preferences in natural language, then (2) recommendations generated based on both types of preferences were uniformly rated by the raters. Hence we perform our experiments in the movie domain, being frequently used for research as movie recommendation is familiar to numerous user study participants.

A key concern in any parallel corpus of this nature is that people may *say* they like items with particular characteristics, but then consume and positively react to quite different items. For instance, this has been observed where people indicate aspirations (e.g., subscribe to particular podcasts) yet actually consume quite different items (e.g., listen to others) [33]. In general, it has been observed that intentions (such as intending to choose healthy food) often do not lead to actual behaviors [43]. Such disparity between corpora could lead to inaccurate prediction about the utility of particular information for recommendation tasks. As such, one of our key considerations was to maximize consistency.

3.1 Phase 1: Preference Elicitation

Our preference elicitation design collected natural language descriptions of rater interests both at the start and at the end of a questionnaire. Specifically, raters were first asked to write short paragraphs describing the sorts of movies they liked, as well as the sorts of movies they disliked (free-form text, minimum 150 characters). These initial liked (+) and disliked (-) self-descriptions for rater r are respectively denoted as $desc_+^r$ and $desc_-^r$.

Next, raters were asked to name five example items (here, movies) that they like. This was enabled using an online query auto-completion system (similar to a modern search engine) where the rater could start typing the name of a movie and this was completed to specific (fully illustrated) movies. The auto-completion included the top 10,000 movies ranked according to the number of ratings in the MovieLens 25M dataset [18] to ensure coverage of even uncommon movies. As raters made choices, these were placed into a list which could then be modified. Each rater was then asked to repeat this process to select five examples of movies they do not like. These liked (+) and disliked (-) item selections for rater r and item selection index $j \in \{1, \ldots, 5\}$ are respectively denoted as $item_+^{r,j}$ and $item_-^{r,j}$.

Finally, raters were shown the five liked movies and asked again to write the short paragraph describing the sorts of movies they liked (which we refer to as the *final description*). The was repeated for the five disliked movies.

3.2 Phase 2: Recommendation Feedback Collection

To enable a fair comparison of item-based and language-based recommendation algorithms, a second phase of our user study requested raters to assess the quality of recommendations made by a number of recommender algorithms based on the information collected in Phase 1. In particular, past work has observed that completeness of labels is important to ensure fundamentally different algorithms can be compared reliably [2, 25].

Desiderata for recommender selection: We aimed for a mix of item-based, language-based, and unbiased recommendations. Hence, we collected user feedback (had they seen it or would they see it, and a 1–5 rating in either case) on a shuffled set of 40 movies (displaying both a thumbnail and a short plot synopsis) drawn from four sample pools:

- **SP-RandPop**, an unbiased sample of popular items: 10 randomly selected top popular items (ranked 1-1000 in terms of number of MovieLens ratings);
- **SP-RandMidPop**, an unbiased sample of less popular items: 10 randomly selected less popular items (ranked 1001-5000 in terms of number of MovieLens ratings);
- **SP-EASE**, personalized item-based recommendations: Top-10 from the strong baseline EASE [42] collaborative filtering recommender using hyperparameter $\lambda = 5000.0$ tuned on a set of held-out pilot data from 15 users;
- **SP-BM25-Fusion**, personalized language-based recommendations: Top-10 from Sparse Review-based Late Fusion Retrieval that, like [3], computes BM25 match between all item reviews in the Amazon Movie Review corpus (v2) [45] and rater's natural language preferences ($desc_+$), ranking items by maximal BM25-scoring review.

Note that SP-RandPop and SP-RandMidPop have 10 different movies for each rater, and that these are a completely unbiased (as they do not leverage any user information, there can be no preference towards rating items that are more obvious recommendations, or other potential sources of bias). On the other hand, SP-EASE consists of EASE recommendations (based on the user item preferences), which we also evaluate as a recommender—so there is some bias when using this set. We thus refer to the merged set of SP-RandPop and SP-RandMidPop as an **Unbiased Set** in the analysis, with performance on this set being key to our conclusions.

3.3 Design Consequences

Importantly, to ensure a maximally fair comparison of language-based and item-based approaches, consistency of the two types of preferences was key in our data collection approach. As such, we directly crowd-sourced both types of preferences from raters in sequence, with textual descriptions collected twice—before and after self-selected item ratings. This required control means the amount of data per rater must be small. It is also a realistic amount of preference information that may be required of a recommendation recipient in a near-cold-start conversational setting. As a consequence of the manual effort required, the number of raters recruited also took into consideration the required power of the algorithmic comparison, with a key contributions being to the protocol developed rather than data scale.

Our approach thus contrasts with alternatives of extracting reviews or preference descriptions in bulk from online content similarly to [4, 32] (where preferences do not necessarily capture a person's interests fully) and/or relying on item preferences expressed either explicitly or implicitly over time (during which time preferences may change).

4 METHODS

Given our parallel language-based and item-based preferences and ratings of 40 items per rater, we compare a variety of methods to answer our research questions. We present the traditional baselines using either item- or language-based preferences, then novel LLM approaches, using items only, language only, or a combination of items and language.

4.1 Baselines

To leverage the item and language preferences elicited in Phase 1, we evaluate CF methods as well as a language-based baseline previously found particularly effective [2, 11].[1] Most baseline item-based CF methods use the default configuration in MyMediaLite [14], including **MostPopular**: ranking items by the number of ratings in the dataset, **Item-kNN**: Item-based k-Nearest Neighbours [40], **WR-MF**: Weighted Regularized Matrix Factorization, a regularized version of singular value decomposition [23], and **BPR-SLIM**: a Sparse Linear Method (SLIM) that learns a sparse weighting vector over items rated, via a regularized optimization approach [34, 38]. We also compare against our own implementation of the more recent state-of-the-art item-based **EASE** recommender [42]. As a language-based baseline, we compare against **BM25-Fusion**, described in Section 3.2. Finally, we also evaluate a random ordering of items in the rater's pool (**Random**) to calibrate against this uninformed baseline.

4.2 Prompting Methods

We experiment with a variety of prompting strategies using a variant of the PaLM model (62 billion parameters in size, trained over 1.4 trillion tokens) [9], that we denote moving forward as simply LLM. Notationally, we assume t is the specific target rater for the recommendation, whereas r denotes a generic rater. All prompts are presented in two parts: a prefix followed by a suffix which is always the name of the item (movie) to be scored for the target user, denoted as $\langle item^t_* \rangle$. The score is computed as the log likelihood of the suffix and is used to rank all candidate item recommendations.[2] As such, we can evaluate the score given by the LLM to every item in our target set of 40 items collected in Phase 2 of the data collection.

Given this notation, we devise **Completion**, **Zero-shot**, and **Few-shot** prompt templates for the case of **Items only**, **Language only**, and combined **Language+Items** defined as follows:

4.2.1 Items only. The completion approach is analogous to that used for the P5 model [16], except that we leverage a pretrained LLM in place of a custom-trained transformer. The remaining approaches are devised in this work:

- **Completion:** $item^{t,1}_+, item^{t,2}_+, item^{t,3}_+, item^{t,4}_+, item^{t,5}_+, \langle item^t_* \rangle$
- **Zero-shot:** I like the following movies: $item^{t,1}_+, item^{t,2}_+, item^{t,3}_+, item^{t,4}_+, item^{t,5}_+$. Then I would also like $\langle item^t_* \rangle$
- **Few-shot (k):**

Repeat $r \in \{1, \ldots, k\}$ { User Movie Preferences: $item^{r,1}_+, item^{r,2}_+, item^{r,3}_+, item^{r,4}_+$ / Additional User Movie Preference: $item^{r,5}_+$

User Movie Preferences: $item^{t,1}_+, item^{t,2}_+, item^{t,3}_+, item^{t,4}_+, item^{t,5}_+$
Additional User Movie Preference: $\langle item^t_* \rangle$

4.2.2 Language only.

- **Completion:** $desc^t_+ \ \langle item^t_* \rangle$
- **Zero-shot:** I describe the movies I like as follows: $desc^t_+$. Then I would also like $\langle item^t_* \rangle$
- **Few-shot (k):**

Repeat $r \in \{1, \ldots, k\}$ { User Description: $desc^r_+$ / User Movie Preferences: $item^{r,1}_+, item^{r,2}_+, item^{r,3}_+, item^{r,4}_+, item^{r,5}_+$

User Description: $desc^t_+$
User Movie Preferences: $\langle item^t_* \rangle$

4.2.3 Language + item.

- **Completion:** $desc^t_+ \ item^{t,1}_+, item^{t,2}_+, item^{t,3}_+, item^{t,4}_+, item^{t,5}_+, \langle item^t_* \rangle$
- **Zero-shot:** I describe the movies I like as follows: $desc^t_+$. I like the following movies: $item^{t,1}_+, item^{t,2}_+, item^{t,3}_+, item^{t,4}_+, item^{t,5}_+$. Then I would also like $\langle item^t_* \rangle$
- **Few-shot (k):**

Repeat $r \in \{1, \ldots, k\}$ { User Description: $desc^r_+$ / User Movie Preferences: $item^{r,1}_+, item^{r,2}_+, item^{r,3}_+, item^{r,4}_+$ / Additional User Movie Preference: $item^{r,5}_+$

User Description: $desc^t_+$
User Movie Preferences: $item^{t,1}_+, item^{t,2}_+, item^{t,3}_+, item^{t,4}_+, item^{t,5}_+$
Additional User Movie Preference: $\langle item^t_* \rangle$

4.2.4 Negative Language Variants. For the zero-shot cases, we also experimented with negative language variants that inserted the sentences "I dislike the following movies: $item^{t,1}_-, item^{t,2}_-, item^{t,3}_-, item^{t,4}_-, item^{t,5}_-$" for **Item** prompts and "I describe the movies I dislike as follows: $desc^t_-$ " for **Language** prompts after their positive counterparts in the prompts labeled **Pos+Neg**.

5 RESULTS

5.1 Data Analysis

We now briefly analyze the data collected from 153 raters as part of the preference elicitation and rating process.[3] The raters took a median of 67 seconds to write their initial descriptions summarizing what they like, and 38 seconds for their dislikes (median lengths: 241 and 223 characters, respectively). Providing five liked and disliked items took a median of 174 and 175 seconds, respectively. Following this, writing final descriptions of likes and dislikes took a median of 152 and 161 seconds, respectively (median lengths: 205 and 207 characters, respectively). We observe that the initial descriptions were produced 3 to 4 times faster than providing 5 example items, in around one minute. As we will see below, this difference in effort is particularly pertinent as item-based and description-based recommendation are comparable in performance. A sample of initial descriptions are shown in Table 1.

Next, we analyze the ratings collected for the movies from the four pools described in Section 3. From Table 2, we observe: (1) The EASE recommender nearly doubles the rate of recommendations

[1] Notably Dacrema et al. [11] observe that the neural methods do not outperform these baselines.
[2] The full target string scored is the movie name followed by the end-of-string token, which mitigates a potential bias of penalizing longer movie names.
[3] We recruited 160 raters, but discard those (5) that did not complete both phases of the data collection and those (2) who provided uniform ratings on all item recommendations in Phase 2.

Table 1: Example initial self-descriptions provided by three raters.

Rater	Liked Movies	Disliked Movies
#1	I like comedy movies because i feel happy whenever i watch them. We can watch those movies with a group of people. I like to watch comedy movies because there will be a lot of fun and entertainment. Its very exciting to watch with friends and family.so,I always watch comedy movies whenever I get time.	I am not at all interested in watching horror movies because whenever I feel alone it will always disturb me with the characters in the movie. It will be affected by dreams and mood always. SO, mostly i ignore watching them when i stay alone in the home.
#2	Fantasy films often have an element of magic, myth, wonder,and the extraordinary. They may appeal to both children and adults, depending upon the particular film. In fantasy films, the hero often undergoes some kind of mystical experience.	Horror is scary. I don't like the feeling of being terrified. Some are either sensitive to suspense, gore or frightful images, or they may have had an experience in their life that makes horror seem real.
#3	I like comedy genre movies, while watching comedy movies I will feel very happy and relaxed. Comedy films are designed to make the audience laugh. It has different kinds of categories in comedy genres such as horror comedy, romantic comedy, comedy thriller,musical-comedy.	I dislike action genre movies because watching fights gives me a headache and bored me. These kinds of movies mainly concentrate on violence and physical feats.

Table 2: Baseline rating statistics for items in the fully labeled pools of items across all raters.

Sample Pool	Movies Per Rater	Fraction Seen	Average Rating	
			Seen Movies	Unseen Movies
SP-RandPop	10	22%	4.21	2.93
SP-RandMidPop	10	16%	4.00	2.85
SP-EASE	10	46%	4.51	3.16
SP-BM25-Fusion	10	24%	4.38	3.11
SP-Full	40	27%	4.29	3.00

that have already been seen by the rater, which reflects the supervised data on which it is trained where raters only rate what they have seen; (2) There is an inherent positive bias to provide a high ratings for movies the rater has already seen as evidenced by the average 4.29 rating in this case; (3) In contrast, the average rating drops to a neutral 3.00 for unseen items.

5.2 Recommended Items

Our main experimental results are shown in Table 3, using NDCG@10 with exponential gain (a gain of 0 for ratings $s < 3$ and a gain of 2^{s-3} otherwise). We compare the mean performance of various methods using item- and/or language-based preferences (as described in Section 3.1) ranking four different pool-based subsets of the 40 fully judged test recommendation items (as described in Section 3.2), recalling that the pool for each rater is personalized to that rater. The language-based results use only the initial natural language descriptions, which raters produced much faster than either liked and disliked item choices or final descriptions, yet they yield equal performance to final descriptions.

We begin with general observations. First, we note the range of NDCG@10 scores within each subset of items is substantially different, due to both the NDCG normalizer that generally increases with a larger evaluation set size, as well as the average rating of each pool. On the latter note, we previously observed that the subset of **Seen** recommendations in Table 2 has the smallest pool of items and a high positive rating bias that makes it hard to differentiate recommenders on this subset. However, and as also recently argued in [35], in a recommendation setting where an item is typically only consumed once (such as movies), we are much more concerned about recommendation performance on the **Unseen** subset vs. the **Seen** subset. Similarly, we are also concerned with performance on the **Unbiased** set since this subset explores a wide range of popularity and is not biased towards item-based collaborative filtering (CF) methods.

To address our original research questions from Section 1:

RQ1: Can language-based preferences replace or improve on item-based preferences? An initial affirmative answer comes from observing that the LLM Language Few-shot (3) method is competitive with most of the traditional item-based CF methods in this near cold-start setting, which is important since as observed in Section 5.1, language-based preferences took less time to elicit than item-based preferences; furthermore, language-based preferences are transparent and scrutable [37]. However, there seems to be little benefit to combining language- and item-based preferences as the Item+Language LLM methods do not appear to provide a boost in performance.

RQ2: LLM-based methods vs. CF? RQ1 has already established that LLM-based methods are generally competitive with item-based CF methods for the Language variants of the LLMs. However, it should also be noted that in many cases the LLM-based methods can even perform comparatively well to CF methods with only Item-based preferences (i.e., the names of the preferred movies). A critical and surprising result here is that a pretrained LLM makes a competitive recommender without the large amounts of supervised data used to train CF methods.

RQ3: Best prompting methodology? The Few-shot (3) prompting method generally outperforms Zero-shot and Completion prompting methods. The difference between Zero-shot and Completion prompting is less pronounced. While not shown due to space constraints, increasing the number of Few-shot examples did not improve performance.

RQ4: Does inclusion of dispreferences help? In the bottom three rows of Table 3, we show the impact of including negative item or language preferences for LLM-based recommenders. There are no meaningful improvements from including both positive and negative preferences (Pos+Neg) over only positive preferences in these LLM configurations. While not shown due to space constraints, omitting positive preferences and using only negative preferences yields performance at or below the popularity baseline.

Table 3: Main experimental results comparing mean NDCG@10 (± 95% standard error) over raters for all recommendation methods. In each case, the fully judged rater-specific evaluation set is ranked by the given recommendation algorithms. Mean evaluation set sizes are in the first row. Note that performance on the *Unseen* item set is most important in a practical recommendation setting.

	Full Set	Unbiased Set	Items that are	
Evaluation Set	SP-Full	SP-Rand{Pop,MidPop}	Seen	**Unseen**
Mean evaluation set size	*40*	*20*	*10.8*	*29.2*
Recommendation Algorithm				
Random Baseline	0.504 ± 0.032	0.532 ± 0.034	0.876 ± 0.023	0.511 ± 0.038
Popularity Baseline	0.595 ± 0.032	0.624 ± 0.029	0.894 ± 0.020	0.534 ± 0.036
(Item) EASE	**0.673 ± 0.038**	0.592 ± 0.030	0.899 ± 0.023	0.559 ± 0.039
(Item) WRMF	0.644 ± 0.036	0.644 ± 0.029	0.897 ± 0.021	0.573 ± 0.037
(Item) BPR-SLIM	0.672 ± 0.037	0.617 ± 0.029	**0.902 ± 0.021**	0.577 ± 0.037
(Item) KNN Item	0.646 ± 0.038	0.610 ± 0.028	0.889 ± 0.024	0.565 ± 0.037
(Language) BM25-Fusion	0.519 ± 0.032	0.623 ± 0.027	0.868 ± 0.023	0.542 ± 0.036
LLM Item Completion	0.649 ± 0.037	0.610 ± 0.027	0.889 ± 0.022	0.563 ± 0.037
LLM Item Zero-shot	0.659 ± 0.037	0.631 ± 0.028	0.895 ± 0.023	0.571 ± 0.037
LLM Item Few-shot (3)	0.664 ± 0.038	0.636 ± 0.027	0.897 ± 0.022	0.572 ± 0.037
LLM Language Completion	0.617 ± 0.032	0.617 ± 0.029	0.889 ± 0.023	0.559 ± 0.035
LLM Language Zero-shot	0.612 ± 0.034	0.626 ± 0.027	0.885 ± 0.024	0.563 ± 0.034
LLM Language Few-shot (3)	0.640 ± 0.036	**0.650 ± 0.026**	0.891 ± 0.022	0.571 ± 0.038
LLM Item+Language Completion	0.654 ± 0.037	0.639 ± 0.027	0.893 ± 0.022	0.568 ± 0.037
LLM Item+Language Zero-shot	0.660 ± 0.038	0.634 ± 0.028	0.897 ± 0.023	**0.582 ± 0.037**
LLM Item+Language Few-shot (3)	0.663 ± 0.038	0.640 ± 0.028	0.899 ± 0.022	0.570 ± 0.037
LLM Item Zero-shot Pos+Neg	0.647 ± 0.037	0.629 ± 0.027	0.892 ± 0.023	0.569 ± 0.038
LLM Language Zero-shot Pos+Neg	0.612 ± 0.034	0.626 ± 0.027	0.885 ± 0.024	0.563 ± 0.034
LLM Item+Language Zero-shot Pos+Neg	0.662 ± 0.037	0.626 ± 0.028	0.897 ± 0.023	0.577 ± 0.037

6 ETHICAL CONSIDERATIONS

We briefly consider potential ethical considerations. First, it is important to consider biases in the items recommended. For instance, it would be valuable to study how to measure whether language-driven recommenders exhibit more or less unintended bias than classic recommenders, such as perhaps preferring certain classes of items over others. Our task was constructed as ranking a fixed corpus of items. As such, all items were considered and scored by the model. Overall performance numbers would have suffered had there been a strong bias, although given the size of our experiments, the existence of bias cannot be ruled out. Larger scale studies would be needed to bound any possible biases present.

Additionally, our conclusions are based on the preferences of a relatively small pool of 153 raters. The small scale and restriction to English-only preferences means we cannot assess whether the same results would be obtained in other languages or cultures.

Finally, we note that the preference data was provided by paid contractors. They received their standard contracted wage, which is above the living wage in their country of employment.

7 CONCLUSION

In this paper, we collected a dataset containing both item-based and language-based preferences for raters along with their ratings of an independent set of item recommendations. Leveraging a variety of prompting strategies in large language models (LLMs), this dataset allowed us to fairly and quantitatively compare the efficacy of recommendation from pure item- or language-based preferences as well as their combination. In our experimental results, we find that zero-shot and few-shot strategies in LLMs provide remarkably competitive in recommendation performance for *pure language-based preferences* (no item preferences) in the near cold-start case in comparison to item-based collaborative filtering methods. In particular, despite being general-purpose, LLMs perform competitively with fully supervised item-based CF methods when leveraging either item-based or language-based preferences. Finally, we observe that this LLM-based recommendation approach provides a competitive near cold-start recommender system based on an explainable and scrutable language-based preference representation, thus providing a path forward for effective and novel LLM-based recommenders using language-based preferences.

REFERENCES

[1] Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. 2021. Program Synthesis with Large Language Models. arXiv:2108.07732 [cs.PL]

[2] Krisztian Balog, Filip Radlinski, and Shushan Arakelyan. 2019. Transparent, Scrutable and Explainable User Models for Personalized Recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '19)*. 265–274.

[3] Krisztian Balog, Filip Radlinski, and Alexandros Karatzoglou. 2021. On Interpretation and Measurement of Soft Attributes for Recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21)*. 890–899.

[4] Toine Bogers and Marijn Koolen. 2017. Defining and Supporting Narrative-Driven Recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems (RecSys '17)*. 238–242.

[5] Vadim Borisov, Kathrin Seßler, Tobias Leemann, Martin Pawelczyk, and Gjergji Kasneci. 2023. Language Models are Realistic Tabular Data Generators. arXiv:2210.06280 [cs.LG]

[6] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. arXiv:2005.14165 [cs.CL]

[7] Arun Tejasvi Chaganty, Megan Leszczynski, Shu Zhang, Ravi Ganti, Krisztian Balog, and Filip Radlinski. 2023. Beyond Single Items: Exploring User Preferences in Item Sets with the Conversational Playlist Curation Dataset. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23)*. 2754–2764.

[8] Huiyuan Chen, Yusan Lin, Menghai Pan, Lan Wang, Chin-Chia Michael Yeh, Xiaoting Li, Yan Zheng, Fei Wang, and Hao Yang. 2022. Denoising Self-Attentive Sequential Recommendation. In *Proceedings of the 16th ACM Conference on Recommender Systems (RecSys '22)*. 92–101.

[9] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. PaLM: Scaling Language Modeling with Pathways. arXiv:2204.02311 [cs.CL]

[10] Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. 2016. Towards Conversational Recommender Systems. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*. 815–824.

[11] Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach. 2019. Are We Really Making Much Progress? A Worrying Analysis of Recent Neural Recommendation Approaches. In *Proceedings of the 13th ACM Conference on Recommender Systems (RecSys '19)*. 101–109.

[12] 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 Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (NAACL '19)*. 4171–4186.

[13] 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]

[14] Zeno Gantner, Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2011. MyMediaLite: A Free Recommender System Library. In *Proceedings of the Fifth ACM Conference on Recommender Systems (RecSys '11)*. 305–308.

[15] 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.

[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 (RecSys '22)*. 299–315.

[17] Deepesh V Hada and Shirish K Shevade. 2021. ReXPlug: Explainable Recommendation using Plug-and-Play Language Model. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21)*. 81–91.

[18] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems* 5, 4, Article 19 (2015).

[19] 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 (WWW '17)*. 173–182.

[20] Yupeng Hou, Shanlei Mu, Wayne Xin Zhao, Yaliang Li, Bolin Ding, and Ji-Rong Wen. 2022. Towards Universal Sequence Representation Learning for Recommender Systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22)*. 585–593.

[21] Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian McAuley, and Wayne Xin Zhao. 2023. Large Language Models are Zero-Shot Rankers for Recommender Systems. arXiv:2305.08845 [cs.IR]

[22] Fangwei Hu and Yong Yu. 2013. Interview Process Learning for Top-N Recommendation. In *Proceedings of the ACM Conference on Recommender Systems (RecSys '13)*. 331–334.

[23] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining (ICDM '08)*. 263–272.

[24] Dietmar Jannach, Ahtsham Manzoor, Wanling Cai, and Li Chen. 2021. A Survey on Conversational Recommender Systems. *Comput. Surveys* 54, 5 (2021).

[25] Marius Kaminskas and Derek Bridge. 2016. Diversity, Serendipity, Novelty, and Coverage: A Survey and Empirical Analysis of Beyond-Accuracy Objectives in Recommender Systems. *ACM Transactions on Interactive Intelligent Systems* 7, 1 (2016), 1–42.

[26] Jie Kang, Kyle Condiff, Shuo Chang, Joseph A. Konstan, Loren Terveen, and F. Maxwell Harper. 2017. Understanding How People Use Natural Language to Ask for Recommendations. In *Proceedings of the Eleventh ACM Conference on Recommender Systems (RecSys '17)*. 229–237.

[27] 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]

[28] Dawen Liang, Rahul G. Krishnan, Matthew D. Hoffman, and Tony Jebara. 2018. Variational Autoencoders for Collaborative Filtering. In *Proceedings of the 2018 World Wide Web Conference (WWW '18)*. 689–698.

[29] Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. 2011. Content-based Recommender Systems: State of the Art and Trends. In *Recommender Systems Handbook*. Springer, 73–105.

[30] Itzik Malkiel, Oren Barkan, Avi Caciularu, Noam Razin, Ori Katz, and Noam Koenigstein. 2020. RecoBERT: A Catalog Language Model for Text-Based Recommendations. arXiv:2009.13292 [cs.IR]

[31] Sheshera Mysore, Mahmood Jasim, Andrew McCallum, and Hamed Zamani. 2023. Editable User Profiles for Controllable Text Recommendation. arXiv:2304.04250 [cs.IR]

[32] Sheshera Mysore, Andrew McCallum, and Hamed Zamani. 2023. Large Language Model Augmented Narrative Driven Recommendations. arXiv:2306.02250 [cs.IR]

[33] Zahra Nazari, Praveen Chandar, Ghazal Fazelnia, Catherine M. Edwards, Benjamin Carterette, and Mounia Lalmas. 2022. Choice of Implicit Signal Matters: Accounting for User Aspirations in Podcast Recommendations. In *Proceedings of the ACM Web Conference 2022 (WWW '22)*. 2433–2441.

[34] Xia Ning and George Karypis. 2011. SLIM: Sparse Linear Methods for Top-N Recommender Systems. In *Proceedings of the 2011 IEEE 11th International Conference on Data Mining (ICDM '11)*. 497–506.

[35] Roberto Pellegrini, Wenjie Zhao, and Iain Murray. 2022. Don't Recommend the Obvious: Estimate Probability Ratios. In *Proceedings of the 16th ACM Conference on Recommender Systems (RecSys '22)*. 188–197.

[36] Gustavo Penha and Claudia Hauff. 2020. What does BERT know about books, movies and music? Probing BERT for Conversational Recommendation. In *Fourteenth ACM Conference on Recommender Systems (RecSys '20)*. 388–397.

[37] Filip Radlinski, Krisztian Balog, Fernando Diaz, Lucas Dixon, and Ben Wedin. 2022. On Natural Language User Profiles for Transparent and Scrutable Recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22)*. 2863–2874.

[38] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI '09)*. 452–461.

[39] Lior Rokach and Slava Kisilevich. 2012. Initial Profile Generation in Recommender Systems Using Pairwise Comparison. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 42, 6 (2012), 1854–1859.

[40] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based Collaborative Filtering Recommendation Algorithms. In *Proceedings of the 10th International Conference on World Wide Web (WWW '01)*. 285–295.

[41] Anna Sepliarskaia, Julia Kiseleva, Filip Radlinski, and Maarten de Rijke. 2018. Preference Elicitation as an Optimization Problem. In *Proceedings of the ACM Conference on Recommender Systems (RecSys '18)*. 172–180.

[42] Harald Steck. 2019. Embarrassingly Shallow Autoencoders for Sparse Data. In *The World Wide Web Conference (WWW '19)*. 3251–3257.

[43] Bas Verplanken and Suzanne Faes. 1999. Good Intentions, Bad Habits, and Effects of Forming Implementation Intentions on Healthy Eating. *European Journal of Social Psychology* 29, 5-6 (1999), 591–604.

[44] Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned Language Models Are Zero-Shot Learners. arXiv:2109.01652 [cs.CL]

[45] Yury Zemlyanskiy, Sudeep Gandhe, Ruining He, Bhargav Kanagal, Anirudh Ravula, Juraj Gottweis, Fei Sha, and Ilya Eckstein. 2021. DOCENT: Learning Self-Supervised Entity Representations from Large Document Collections. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume (EACL '21)*. 2540–2549.