A Flash in the Pan: Better Prompting Strategies to Deploy Out-of-the-Box LLMs as Conversational Recommendation Systems

Gustavo Adolpho Lucas de Carvalho[1] **Simon Benigeri**[2] **Jennifer Healey**[3]
Victor Bursztyn[3] **David Demeter**[2] **Lawrence Birnbaum**[2]

[1]University of Southern California, {lucasdec}@usc.edu
[2]Northwestern University, {simon.benigeri, david.jr1, l-birnbaum}@northwestern.edu
[3]Adobe Research, {jehealey, victor.bursztyn}@adobe.com

Abstract

Conversational Recommendation Systems (CRSs) are a particularly interesting application for out-of-the-box LLMs due to their potential for eliciting user preferences and making recommendations in natural language across a wide set of domains. Somewhat surprisingly, we find however that in such a conversational application, the more questions a user answers about their preferences, the worse the model's recommendations become. We demonstrate this phenomenon on a previously published dataset as well as two novel datasets which we contribute. We also explain why earlier benchmarks failed to detect this round-over-round performance loss, highlighting the importance of the evaluation strategy we use and expanding upon Li et al. (2023a). We also present preference elicitation and recommendation strategies that mitigate this degradation in performance, beating state-of-the-art results, and show how three underlying models, GPT-3.5, GPT-4, and Claude 3.5 Sonnet, differently impact these strategies. Our datasets and code are available at https://github.com/CtrlVGustavo/A-Flash-in-the-Pan-CRS.

1 Introduction

The advent of large language models (LLMs) has revolutionized conversational recommendation systems (CRSs). Recent works have shown that using conversation history can improve both question generation and product recommendation in naturalistic, multi-round conversational recommendation settings (Li et al., 2023a; Deldjoo, 2024; Li et al., 2023c). Somewhat surprisingly, however, we show that round-over-round recommendation accuracy tends to *decrease* using prior methods. That is, as more information is gathered about the user's preferences, recommendations become less and less reliable. We develop two additional datasets to test this observation in different domains and find similar results. To our knowledge, our datasets are the only multi-turn, profile-based CRS benchmarks with human-generated labels.

In this paper, we introduce strategies to both (1) generate better questions to elicit human preferences, and (2) make recommendations using the information gained through these questions, aimed in part at overcoming the problem described above. We evaluate our method on an article recommendation benchmark from (Li et al., 2023a), as well as two novel datasets we contribute, one for movie recommendation and another for book recommendation (described in Appendix A). We report on the depreciation of model performance as more questions are asked and the synergistic effect of combining different prompting strategies. Lastly, we show that our method outperforms SOTA results (Li et al., 2023a) on these three distinct datasets.

The key idea behind our improvements in preference elicitation is to nudge the LLM to ask questions concerning item examples, for example what movies the user enjoys, rather than more abstract questions. As described in Section 3.1, we do this in two ways: by starting the conversation with a predetermined question, and by directly prompt the model to ask questions about item examples.

We use three different prompting strategies, detailed in Section 3.2, in our LLM-based recommender. In the first round, we use a prompt designed to take advantage of information we know will be yielded by the predetermined question. Subsequently, we keep predictions based on previous question-answer pairs in the context window of the LLM. Finally, we introduce an additional reasoning step that uses the prediction made in the previous iteration to help the LLM assess how the most recent question-answer pair should affect the current prediction. These prompting strategies taken together yield improved preference estimates.

Proceedings of the 31st International Conference on Computational Linguistics, pages 8385–8398
January 19–24, 2025.

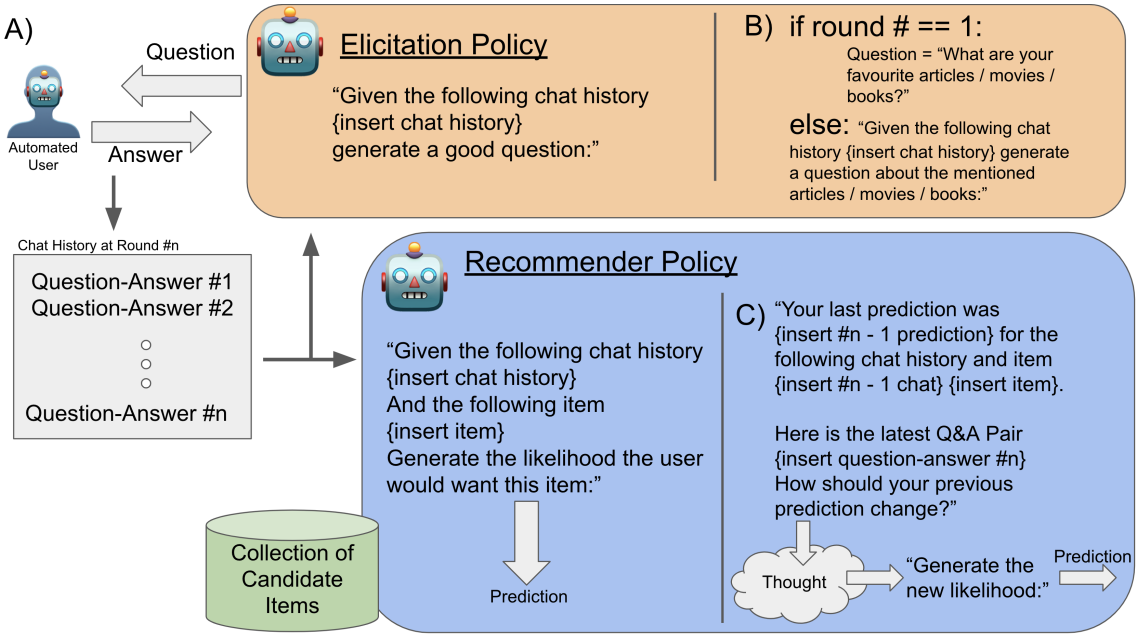

Figure 1: **General Overview of Our Paradigm**, methods, and benchmark. **(A)** Illustrates how our benchmarks work. An LLM-simulated user answers questions from an Elicitation Policy. This interaction generates a chat history which is fed back to the Elicitation Policy to generate the next question, as well as a Recommender Policy to generate a prediction. For both policies, the text to the left represents the baseline prompt and the text to the right represents our methods. **(B)** Exemplifies our two prompting methods for the Elicitation Policy, the use of a predetermined question for round #1 and Item-Centered Question generation. **(C)** Describes the broad strokes of one of our prompting strategies for the Recommender, Incremental Prediction, which introduces an intermediary reasoning step before generating a prediction.

Contributions

Our work identifies a perhaps somewhat counter-intuitive tendency of out-of-the-box LLM-based CRSs towards performance degradation when using multiple rounds of questions and answers to elicit user preferences, while also introducing methods for utilizing conversation history to mitigate this problem. We achieve better than state-of-the-art results on an article recommendation dataset (Li et al., 2023a) and show that our strategy also works better on a new movie dataset and a new book dataset. To summarize we contribute:

- Novel findings regarding the behavior of out-of-the-box LLM-based CRSs.
- A novel method for generating better information with a differentiated preference Elicitation Policy.
- Multiple novel methods to develop better preference estimates for recommendations using known information.
- Two unique, human-annotated datasets of user preferences, one for movie recommendation and another for book recommendation.

2 Prior Work

Conversational recommendation systems are a well-studied application in NLP (Jannach et al., 2020; Pramod and Bafna, 2022). Previous work has established this task can be carried out by LLMs without any extra supervised training (Palma et al., 2023; Sanner et al., 2023; Liu et al., 2023; Huang et al., 2024). Palma et al. (2023) and Liu et al. (2023) demonstrate that LLMs can recommend an item from a selection of items and predict user ratings for a particular item.

A common approach to capturing user preferences leverages user interaction data. *GPT4Rec* (Li et al., 2023b) provides a recommender LLM with a basic user profile: a list of items and corresponding user feedback. Such user profiles can also be enriched/augmented using LLMs; for example, *PALR* (Yang et al., 2023) prompts an LLM to summarize the profile, and *LLM-Rec* (Lyu et al., 2023) prompts an LLM to describe items in the profile.

Another approach is to hold a conversation with a user to elicit their preferences. For example, *RecLLM* (Friedman et al., 2023) and *SalesForce vs*

Domain	Model	Round					
		1	2	3	4	5	Highest-Scoring
Online Articles	Baseline GPT-3.5	66.0 ± 0.6	65.3 ± 1.6	64.3 ± 2.1	63.0 ± 2.0	63.0 ± 1.9	66.0 *(round 1)*
	Baseline GPT-4	68.9 ± 0.3	67.0 ± 0.5	67.8 ± 0.6	67.1 ± 0.8	66.4 ± 0.9	68.9 *(round 1)*
	Baseline Claude 3.5 Sonnet	67.2 ± 1.1	67.2 ± 1.0	65.8 ± 2.1	64.8 ± 0.6	64.6 ± 0.3	67.2 *(round 2)*
	Our Method GPT-3.5	**70.2 ± 2.2**	68.3 ± 2.7	67.0 ± 1.7	66.8 ± 1.0	66.0 ± 0.9	**70.2 *(round 1)***
	Our Method GPT-4	69.6 ± 2.0	**69.8 ± 1.7**	**69.8 ± 1.5**	**69.7 ± 1.5**	**69.5 ± 1.5**	69.8 *(round 3)*
	Our Method Claude 3.5 Sonnet	68.9 ± 1.6	69.6 ± 1.6	68.5 ± 2.4	68.6 ± 1.6	67.2 ± 2.2	69.6 *(round 2)*
Movies	Baseline GPT-3.5	**70.0 ± 0.3**	66.7 ± 0.3	66.3 ± 1.2	66.4 ± 0.8	66.1 ± 0.5	70.0 *(round 1)*
	Baseline GPT-4	68.8 ± 0.2	68.2 ± 0.4	68.0 ± 0.4	68.0 ± 0.1	68.5 ± 0.7	68.8 *(round 1)*
	Baseline Claude 3.5 Sonnet	66.2 ± 0.1	67.6 ± 0.6	67.9 ± 0.6	67.5 ± 0.6	67.4 ± 0.3	67.9 *(round 3)*
	Our Method GPT-3.5	69.3 ± 0.5	71.7 ± 1.2	71.8 ± 1.0	71.8 ± 0.8	71.6 ± 0.6	71.8 *(round 4)*
	Our Method GPT-4	69.9 ± 0.2	70.3 ± 0.4	70.8 ± 0.4	71.0 ± 0.6	71.0 ± 0.3	71.0 *(round 5)*
	Our Method Claude 3.5 Sonnet	**70.0 ± 0.3**	**71.8 ± 0.4**	**72.3 ± 0.1**	**72.6 ± 0.6**	**72.7 ± 0.6**	**72.7 *(round 5)***
Books	Baseline GPT-3.5	69.0 ± 0.5	67.7 ± 0.5	67.5 ± 1.1	67.2 ± 1.2	66.6 ± 0.7	69.0 *(round 1)*
	Baseline GPT-4	66.5 ± 1.2	67.1 ± 0.8	66.8 ± 0.6	66.7 ± 0.8	66.8 ± 0.9	67.1 *(round 2)*
	Baseline Claude 3.5 Sonnet	64.9 ± 0.5	65.6 ± 0.5	65.2 ± 0.5	65.0 ± 0.3	65.1 ± 0.2	65.6 *(round 2)*
	Our Method GPT-3.5	**69.8 ± 0.4**	**69.8 ± 0.5**	68.9 ± 1.7	68.0 ± 1.9	67.3 ± 1.9	**69.8 *(round 2)***
	Our Method GPT-4	67.8 ± 0.6	68.3 ± 0.8	68.3 ± 1.2	68.6 ± 1.2	68.5 ± 1.3	68.6 *(round 4)*
	Our Method Claude 3.5 Sonnet	67.8 ± 0.1	69.1 ± 0.3	**69.6 ± 0.4**	**69.6 ± 0.3**	**69.3 ± 0.3**	69.6 *(round 4)*
Average Across All Domains	Baseline GPT-3.5	68.3 ± 1.9	66.6 ± 1.1	66.0 ± 1.5	65.5 ± 2.1	65.2 ± 1.8	68.3 *(round 1)*
	Baseline GPT-4	68.1 ± 1.3	67.4 ± 0.6	67.5 ± 0.6	67.3 ± 0.6	67.2 ± 1.0	68.1 *(round 1)*
	Baseline Claude 3.5 Sonnet	66.1 ± 1.1	66.8 ± 1.0	66.3 ± 1.3	65.8 ± 1.4	65.7 ± 1.4	66.8 *(round 2)*
	Our Method GPT-3.5	**69.8 ± 0.4**	69.9 ± 1.6	69.2 ± 2.2	68.9 ± 2.4	68.3 ± 2.7	69.9 *(round 2)*
	Our Method GPT-4	69.1 ± 1.0	69.5 ± 1.0	69.6 ± 1.2	69.8 ± 1.1	**69.7 ± 1.2**	69.8 *(round 4)*
	Our Method Claude 3.5 Sonnet	68.9 ± 1.0	**70.2 ± 1.3**	**70.1 ± 1.8**	**70.3 ± 1.9**	**69.7 ± 2.6**	**70.3 *(round 4)***

Table 1: P(correct) at each round for our method and the baseline, using GPT-3.5, GPT-4, or Claude 3.5 Sonnet as the backbone LLM for the CRS, for different domains. The highest score per round for each domain is in bold.

SalesBot (Murakhovs'ka et al., 2023) update a user profile with facts and preferences extracted from a conversation transcript. These preferences can also be inferred from the transcript, as in *GATE* (Li et al., 2023a) and *IERL* (Hong et al., 2023). This raises the question: *How can we get LLMs to ask questions such that the conversation transcript provides a good representation of the user's preferences?*

Murakhovs'ka et al. (2023) source the questions from buying guides, which provide a list of common questions to ask a customer interested in buying a category of item. Hong et al. (2023) frame conversational recommendation as goal-directed conversation. They use LLMs to simulate conversation data used as training data for reinforcement learning. Li et al. (2023a) aim for for "goal-directed" conversations by prompting an LLM to ask a user about their preferences given the the conversation thus far.

Our work expands on Li et al. (2023a) in the following ways:

1. Utilizing a predetermined first question to initiate the conversation, and only then use the LLM to generate subsequent questions.

2. Preference elicitation questions are centered on specific items, producing question-and-answer transcripts that result in better recommendations.

3. Rather than using a single prompt to estimate the user's preference given a transcript of the conversation so far, and an item, we introduce more elaborate prompting strategies to produce better preference estimates.

4. Of the datasets provided by Li et al. (2023a), we test our approach only on the articles dataset. (The other two datasets, for email regexes and moral reasoning, do not fit our task of interest, i.e., preference elicitation for recommendation.)

5. We also test our approach on two new datasets, introduced in Section 4.3. Both datasets expand on the evaluation done in Li et al. (2023a) by using profiles and labels based on real human users and recommenders.

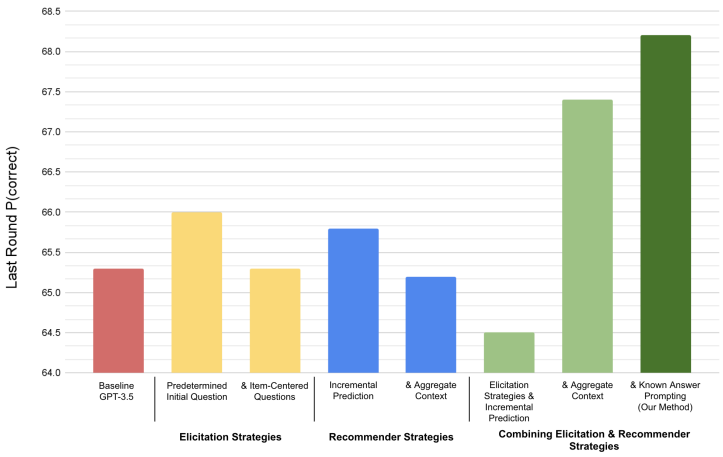

Figure 2: Ablation study detailing the Average P(correct) on all domains for different combinations of strategies.

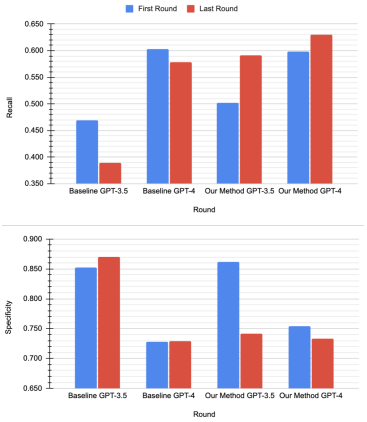

Figure 3: Top chart shows the average recall and the bottom shows specificity across all domains for the first and last round.

3 Methods

Following Li et al. (2023a) and other prior work, our goal is to design an LLM-based CRS by breaking the task into an Elicitation Policy and a Recommender. The Elicitation Policy interacts with a human user to produce a free-form preference representation (e.g., a question-answer transcript) that is cumulative over turns in the conversation. The Recommender maps this representation and some possible suggestion to the probability that the user would agree to that suggestion.

3.1 Elicitation Policy

We introduce two distinct but interrelated ideas: Utilizing a *predetermined first question*, and generating *item-focused questions*. Both the predetermined first question and the prompt used for the LLM-generated questions are designed to produce conversations that concern specific items embodying the user's preferences.

Predetermined Initial Question: The key advantage of using LLM-generated questions instead of predetermined questions is that the LLM can adapt based on the user's previous responses. This clearly isn't possible when generating the initial question, however. Given that, using a predetermined question to start the conversation offers significant benefits, namely, it can be designed to promote conversations yielding superior informational value for the Recommender. Here, we use an initial question that simply asks the user to enumerate examples of items they like in the domain.

Item-Centered Questions: All subsequent questions are generated by prompting an LLM as in prior work (e.g., Li et al. (2023a)). However, the prompt we utilize encourages the LLM to focus on questions about the user's preferences regarding the items mentioned in previous response. For example, questions such as *"Could you expand upon why you like Dune's world-building,"* and *"What specifically did you enjoy about the movie Saltburn,"* are preferred over broad questions such as *"What is your favorite genre,"* or *"Do you prefer intellectual or goofy movies."* Given that LLMs can improve at tasks using a Few-shot scenario (Brown et al., 2020), we believe that focusing on information about specific items in preference elicitation will lead to better predictions by an LLM-based Recommender that ultimately is required to reason about user preferences regarding specific items.

Thus, our initial question gets the user to name items that reflect their preferences, while our prompt directs the LLM to ask clarification questions regarding the items the user has mentioned. The initial question for each domain as well as all prompts are described in full detail in Appendix B.

3.2 Recommender

We introduce three prompting strategies to improve upon the Recommender in Li et al. (2023a).

Incremental Prediction: Compared to the baseline method that simply utilizes the transcript of questions and answers to a given point, Incremental Prediction introduces two key changes. First, in all rounds except for the first, it also includes the prediction produced by the Recommender in the previous round as part of the input to recommen-

dation in the current round.

Second, our Recommender generates probability assessments in two phases. The first starts by describing the prediction in the previous round and the the conversational transcript up to that point; it then asks the model *how* the most recent question-answer pair should change the probability assigned by the prediction. The resulting answer is then appended to the context window, together with a second prompt that instructs the LLM to generate a new prediction for the current round (see Figure 1).

We speculate that by introducing this intermediate reasoning step, Incremental Prediction aids the LLM by acting as a sort of structured Zero-shot Chain of Thought (CoT) (Kojima et al., 2022).

Known Answer Prompting: One disadvantage of using LLM-generated questions is that, since LLMs can adapt their questions to the user's previous responses, it's hard to predict the kind of information that will be elicited. This leads to the use vague prompts for the Recommender, e.g., "use the preference information to make a prediction." When using a Predetermined Initial Question in the first round, however, we know the nature of the information that is likely to be present in the user's response, and hence can provide a more specific prompt to the LLM about how to use that information in making a prediction. Specifically, we prompt the Recommender to assign high probability to any item under consideration that is similar to one of the items listed as a favorite by the user.

Aggregate Context: Aggregate Context simply means that we do not remove previous predictions from the context window. At any given turn, the Recommender receives as input not only the entire transcript of questions and answers to that point, but also the recommendation and reasoning (Incremental Prediction) made at each prior turn. Hence, in each round another example of the inputs to and outputs of the Recommender to that point is added to the context. Together with Incremental Prediction, this prompting strategy functions as a sort of dynamically generated Few-shot CoT prompt to improve performance.

4 Evaluation and Metric

We evaluate our approach in three different domains, online article recommendation, movie recommendation, and book recommendation. For all three datasets, we adopt the metric used by Li et al. (2023a), which measures how well the CRS can estimate the user's preferences. Specifically, *P(correct)* is the probability our recommender assigns the user-preferred answer. For example, for some interaction history and suggested recommendation, if the user would answer *yes* to that recommendation and the probability assigned by the Recommender is 80%, then *P(correct)* is 80%. On the other hand, if the user would say *no* to the recommendation, *P(correct)* would be 20%. We selected this metric instead of accuracy since guessing the user's preferences may not always be possible and modeling this uncertainty is useful. Having said that, a significant distinction between our work and Li et al. (2023a) is that while they measured the *cumulative P(correct) over all rounds of dialogue*, we opted to report the *P(correct) for each individual round*.

4.1 Online Articles Domain Dataset

In our first dataset, from Li et al. (2023a), the items of interest are online articles. The dataset consists of 5 different *personae*, each with 16 article descriptions that are labeled *True* or *False* corresponding to whether the specific *persona* would want to be recommended the article. An LLM is prompted with a description of a human *persona* to create a simulated user that approximates the responses of a human user with the same characteristics described in the prompt. The Elicitation Policy interacts with each simulated user for some number of rounds, generating a set of preference representations for each round, each being the cumulative collection of the QA pairs up to that round. Differently from Li et al. (2023a), we report *P(correct)* for every round. Hence, for each preference representation in the set of suggested recommendations, we compute the average *P(correct)* given all the labeled examples. We evaluate the performance of an Elicitation strategy and Recommender taken from Li et al. (2023a) as a baseline, as well as the performance of our methods.

4.2 Uniqueness of the Evaluation Approach

In contrast with Li et al. (2023a), most evaluation methods for LLM-based CRSs are not designed with a multi-turn interactive setting in mind, and instead consist of prompting the LLM with past user interaction or dialogue data (Li et al., 2023b; He et al., 2023; Palma et al., 2023; Liu et al., 2023; Lyu et al., 2023). In other words, most previous works do not implement and evaluate LLMs as Elicitation Policies, but only as Recommenders.

Method	Round #1	Round #2	Round #3	Round #4	Round #5
Baseline GPT-3.5	68.3	66.6	66.1	65.5	65.3
Zero-shot CoT	66.6 (- 1.7)	65.3 (- 1.3)	65.3 (- 0.8)	64.9 (- 0.6)	65.3 (+ 0.0)
Incremental Prediction	68.3 (+ 0.0)	67.0 (+ 0.4)	67.3 (+ 1.2)	66.9 (+ 1.4)	65.8 (+ 0.5)

Table 2: Average P(correct) at each round across all domains of the GPT-3.5 baseline, Zero-shot CoT, and Incremental Prediction. Performance gains compared to the baseline are highlighted in green text and performance loss in red.

Lei et al. (2020) and Zhang and Balog (2020) are examples of methods that also implement Elicitation Policies in a setting with simulated users. However their Elicitation Policies depend on predefined conversation flows and templates.

Other than Li et al. (2023a), the only work to our knowledge that implements and evaluates LLMs as Recommenders and Elicitation Policies in a multi-turn interactive environment is Wang et al. (2023). Instead of simulating users with prompts containing *personae*, Wang et al. (2023) defines user profiles by providing the LLM with a set of target items. We discuss difficulties raised by this approach in Section 6.2.

Another factor that differentiates the online articles domain dataset from previous benchmarks is that it contains both positive and negative labels. While most prior methods only use recall as their performance metric, we believe that having negative labels improves the quality of the evaluation since it allows for a more comprehensive analysis of model performance (see Section 6.1).

4.3 Movie Domain and Book Domain Datasets

The online articles dataset used by Li et al. (2023a) has several limitations, including the small number of *personae* and the lack of information on how these *personae* were created. To address these problems, we developed a dataset in the movie recommendation domain *Reddit Movie Personae* and a dataset in the book recommendation domain *Reddit Book Personae*. Both have the same format as the dataset from Li et al. (2023a). Our books dataset has 42 *personae* and our movies dataset has *64 personae*, i.e., more than 8x and 10x the number of *personae* in the orginal online articles dataset, respectively. For both datasets, each *persona* has 10 items labeled *True* or *False*. Every *persona* and movie/book recommendation was derived from real human movie/book requests and suggestions scraped from Reddit (He et al., 2023; Penha and Hauff, 2020). We also evaluated our *personae* with over 40 independent raters to verify that they were reasonable representations of the real human requests upon which they were based. For details on the datasets verification and creation see Appendix A.

5 Results

For each task domain, we performed 3 trials of the experiment described in Section 4.1. The averages of those trials, along with their corresponding 95% confidence interval, are reported in Table 1. We report the *P(correct)* for the baseline and for our method, which comprises all the strategies described in Section 3. Our results show the following:

1. **Our method outperforms the baseline on all task domains** by an average of 3.1% with GPT-3.5, 2.5% with GPT-4, and 4.0% with Claude 3.5 Sonnet after the last round of questioning.

2. Our method's better performance is largely, but not entirely, due to a somewhat surprising (and certainly problematic) behavior of the baseline strategy, namely, that **the baseline's performance goes down as more questions are asked.** Our approach does not display this effect with GPT-4 or Claude 3.5 Sonnet, increasing its performance by an average of 0.6% and 0.8% (respectively) between the first and last questions. The baseline for both models, on the other hand, has a 0.9% and 0.4% performance decrease on average. The baseline with GPT-3.5 suffers even more from this problem, losing 3.1% between the first and last rounds. Our method also degrades over rounds using GPT-3.5, albeit less so, losing 1.5%. This problematic behavior is not the only reason our method outperforms the baseline: Even considering only the highest-scoring round, our method still outperforms the baseline by an average of 1.6% with GPT-

3.5, 1.7% with GPT-4, and 3.5% with Claude 3.5 Sonnet.

3. **Our method with GPT-3.5 outperforms the GPT-4 and Claude 3.5 Sonnet baselines**. On average, our method with GPT-3.5 bests the baseline with GPT-4 by 1.1% and Claude 3.5 Sonnet by 2.6% w.r.t. to the last round. It also slightly outperforms our method with GPT-4 by 0.1% w.r.t. the highest-scoring round, but this may not be statistically significant.

6 Discussion

To further understand why our method helps mitigate or eliminate the degradation in performance we found over rounds, we consider the difference between *recall* (i.e. the true positive rate) and *specificity* (i.e. the true negative rate) of the first and last rounds of conversation between the baseline and our methods. We compare GPT-3.5 and GPT-4 as they match each other's performance w.r.t their highest-scoring round with our method, except GPT-3.5 experiences degradation while GPT-4 mostly does not.

6.1 Exploring Depreciation Over Rounds

As seen in Figure 3, the recall for the baseline methods decreases between the first and last rounds of conversation, while the specificity either increases or stays the same. The recall for our method, on the other hand, increases from the first to the last round of conversation. This implies that, without a dedicated prompting strategy, **extra information does not help the model make good recommendations, although it does prevent bad ones.** Furthermore, specificity is always *at least* 10% higher than recall, demonstrating how, in general, the model is better at avoiding bad recommendations than at making good ones. It is also important to note that our method loses specificity between the first and last rounds when using GPT-3.5; this may be part of the reason it still leads to a loss in P(correct) as the rounds go on. Our method with GPT-4 also has a marginal loss in specificity between the first and last rounds, although both rounds are still higher than the specificity of the baseline.

6.2 A Case Study on User Simulation for CRS

Given that most prior evaluation methods for LLM-based CRMs were not multi-turn, interactive benchmarks and that Li et al. (2023a) measured the cumulative performance over all rounds (see Section 4), it is reasonable that this behavior of round-over-round performance loss has gone unreported. However, even though Wang et al. (2023) had a similar approach to our evaluation setting, they reported round-over-round *increase* in performance. To investigate this discrepancy we adapted our setup to mirror theirs and found a straightforward explanation that highlights an advantage of our benchmark design.

The most significant difference between the two settings is our choice of prompt for the LLM simulated users. While we adapted real user descriptions of their preferences into *persona* for LLMs to emulate (see Figure 4 for an example), Wang et al. (2023) provides a set of target items and instructs the LLM to provide information about the items when asked for their preferences. The general framework of their user simulation prompt is as follows, *"You are a user chatting with a recommender for recommendation. Your target items: {}. You should never directly tell the target item title. If the recommender asks for your preference, you should provide the information about {}."*, where *"{}"* is replaced with a list of target items. Given that the list of target items in the user prompt is the same as the positive labels for the Recommender, we suspected that this choice of prompt design might lead to a failure mode in which the simulated user describes the positive labels to the Recommender, instead of describing the preferences that are associated with those positive labels.

To verify this, we implemented our own version of Wang et al. (2023) prompt for user simulation by taking the positive labels from each profile in the *Reddit Movie Personae* dataset and using them as the main part of the user prompt. We found that, out of 64 conversations, all of them had at least one response where the simulated user described the positive labels. For example, when one of the simulated users was asked "What are some of your favorite movie quotes or one-liners that always make you laugh or stay with you?", it responded with "'I see dead people.' from a 2001 parody film" and "'Yarp!' from a British action comedy film in 2007" as well as other quotes from their list of positive labels. Hence, even though it did not explicitly inform the Recommender of the titles of the positive labels, it provided information that was specific enough to potentially make the recommendation task much easier to modern LLMs. This failure mode is not possible within

our *persona* framework. This entire situation illustrates the nuances involved in designing prompting schemes for user simulation in LLM-based CRSs benchmarks.

6.3 Ablation Study

Given that our method consists of several different prompting strategies, we also ran an ablation study to evaluate better how the specific component strategies interact. The results can be seen in Figure 2. Surprisingly, our method's score is highly dependent on interactions between our Elicitation and Recommender prompting strategies.

1. **Both our elicitation strategies, as well as our recommender strategies, lose performance when paired with each other**. The Predetermined Initial Question strategy scores %0.7 higher by itself than when combined with Item-Centered Questions. Similarly, Incremental Prediction loses %0.6 when paired with Aggregate Context.

2. **The combination of the two elicitation strategies and Incremental Prediction demonstrates even worse performance**, scoring %0.8 lower than the baseline.

3. **However, performance increases when we combine all the Elicitation and Recommender strategies we propose**. The combination of using a Predetermined Initial Question, Item-Centered Questions, Incremental Prediction, and Aggregate Context, outperforms the independent use of any of the strategies alone.

6.4 Studying the Effect of Zero-shot Reasoning

Lastly, we also wanted to compare our prompting strategies with others that are more well-known, such as Zero-shot CoT. In Table 2 we compare one of our prompting strategies, Incremental Prediction, with Zero-shot CoT, and find that, despite the success of the reasoning used in Incremental Prediction, Zero-shot CoT prompting loses performance compared to the baseline. This further demonstrates the complexity of this task, in that commonly used strategies for improving performance in many task domains seem to be not only ineffective but actually counterproductive. We hypothesize that prompting the model to reason by itself is not sufficient and that instead this task requires being prompted to reason in a more structured way (as we do in Incremental Prediction).

7 Conclusion and Future Work

In this paper, we contribute 2 prompting strategies to improve user elicitation and 3 strategies for item recommendation in an LLM-based CRS. We demonstrate a previously unreported phenomenon with LLM-based CRSs, the degradation of performance as more questions are asked and answered. We show that our method mitigates and sometimes even eliminates this undesired behavior. Additionally, we illuminate some of the mechanisms behind this behavior by showing that the deprecation of performance is due to a loss in recall, but not specificity. We conclude that although the extra information helps filter out bad recommendations, it also causes the Recommender to lose confidence in good recommendations. Future research could further explore how this behavior can be eliminated, as well as verify that this phenomenon occurs when LLM-based CRSs interact with real users, given that our results are limited by our use of automated benchmarks (eun Yoon et al., 2024).

We also find that our method's improvement over the baseline is due to the combination of these prompting strategies, as using them independently does not work. This suggests that finding new prompting strategies for LLM-based CRSs through trial and error will be very challenging. Further work could experiment with more systematic ways to find prompting strategies for CRSs as well as explore how synergies arise between them.

Lastly, we also show that our method with GPT-3.5 outperforms the baseline with GPT-4 and Claude 3.5 Sonnet, with the caveat that its degradation round-over-round is worse. This is indicative of the potential of prompting strategies to allow older models to compete with more sophisticated models as CRSs, while also implying that this degradation behavior can only be addressed via promoting when using newer models. Future research is needed to explore these hypotheses.

In conclusion, our work contributes 5 novel strategies for LLM-based CRS, 2 new unique benchmarks for automated testing, as well as several empirical observations regarding the behavior of LLM-based CRSs. In sum, the application of out-of-the-box LLMs for conversational recommendation holds a lot of promise, but still requires a lot of future work.

References

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Yashar Deldjoo. 2024. Understanding biases in chatgpt-based recommender systems: Provider fairness, temporal stability, and recency. *ArXiv*, abs/2401.10545.

Se eun Yoon, Zhankui He, Jessica Maria Echterhoff, and Julian McAuley. 2024. Evaluating large language models as generative user simulators for conversational recommendation. *ArXiv*, abs/2403.09738.

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. *Preprint*, arXiv:2305.07961.

Zhankui He, Zhouhang Xie, Rahul Jha, Harald Steck, Dawen Liang, Yesu Feng, Bodhisattwa Prasad Majumder, Nathan Kallus, and Julian Mcauley. 2023. Large language models as zero-shot conversational recommenders. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, CIKM '23, page 720–730, New York, NY, USA. Association for Computing Machinery.

Joey Hong, Sergey Levine, and Anca Dragan. 2023. Zero-shot goal-directed dialogue via rl on imagined conversations. *ArXiv*, abs/2311.05584.

Chengkai Huang, Tong Yu, Kaige Xie, Shuai Zhang, Lina Yao, and Julian McAuley. 2024. Foundation models for recommender systems: A survey and new perspectives. *ArXiv*, abs/2402.11143.

D. Jannach, Ahtsham Manzoor, Wanling Cai, and Li Chen. 2020. A survey on conversational recommender systems. *ACM Computing Surveys (CSUR)*, 54:1 – 36.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *ArXiv*, abs/2205.11916.

Wenqiang Lei, Xiangnan He, Yisong Miao, Qingyun Wu, Richang Hong, Min-Yen Kan, and Tat-Seng Chua. 2020. Estimation-action-reflection: Towards deep interaction between conversational and recommender systems. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, WSDM '20. ACM.

Belinda Z. Li, Alex Tamkin, Noah Goodman, and Jacob Andreas. 2023a. Eliciting human preferences with language models. *Preprint*, arXiv:2310.11589.

Jinming Li, Wentao Zhang, Tiantian Wang, Guanglei Xiong, Alan Lu, and Gérard G. Medioni. 2023b. Gpt4rec: A generative framework for personalized recommendation and user interests interpretation. *ArXiv*, abs/2304.03879.

Xinyi Li, Yongfeng Zhang, and Edward C. Malthouse. 2023c. A preliminary study of chatgpt on news recommendation: Personalization, provider fairness, fake news. *ArXiv*, abs/2306.10702.

Junling Liu, Chaoyong Liu, Renjie Lv, Kangdi Zhou, and Yan Bin Zhang. 2023. Is chatgpt a good recommender? a preliminary study. *ArXiv*, abs/2304.10149.

Hanjia Lyu, Song Jiang, Hanqing Zeng, Yinglong Xia, and Jiebo Luo. 2023. Llm-rec: Personalized recommendation via prompting large language models. *ArXiv*, abs/2307.15780.

Lidiya Murakhovs'ka, Philippe Laban, Tian Xie, Caiming Xiong, and Chien-Sheng Wu. 2023. Salespeople vs salesbot: Exploring the role of educational value in conversational recommender systems. *ArXiv*, abs/2310.17749.

Dario Di Palma, Giovanni Maria Biancofiore, Vito Walter Anelli, Fedelucio Narducci, T. D. Noia, and Eugenio Di Sciascio. 2023. Evaluating chatgpt as a recommender system: A rigorous approach. *ArXiv*, abs/2309.03613.

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, page 388–397, New York, NY, USA. Association for Computing Machinery.

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

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

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 ACM Conference on Recommender Systems (RecSys '23)*.

Xiaolei Wang, Xinyu Tang, Xin Zhao, Jingyuan Wang, and Ji-Rong Wen. 2023. Rethinking the evaluation for conversational recommendation in the era of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language*

Processing. Association for Computational Linguistics.

Fan Yang, Zheng Chen, Ziyan Jiang, Eunah Cho, Xiaojiang Huang, and Yanbin Lu. 2023. Palr: Personalization aware llms for recommendation. *Preprint*, arXiv:2305.07622.

Shuo Zhang and Krisztian Balog. 2020. Evaluating conversational recommender systems via user simulation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery amp; Data Mining*, volume 5 of *KDD '20*, page 1512–1520. ACM.

A Dataset Creation

Below we describe how the Movies and Books datasets were created.

A.1 Movies Dataset

Derived from the Reddit Movies dataset (He et al., 2023), which scraped a large pool of Reddit posts of users asking and receiving movie recommendations, our dataset consists of *personae* (or *user profiles*) and *postive/negative recommendations*.

The personae were based on posts made by Reddit users where they detail the kind of movie they would want to watch. Out of all the posts in the Reddit Movies dataset, we only considered the 200 longest posts in terms of word count since we assumed that the longer requests were more likely to contain comprehensive information for the creation of profiles.

From those posts, we further filtered out those that did not contain at least 5 movie suggestions with 3 upvotes or more. Movie suggestions were extracted from the comments made under the post, thus the amount of upvotes a given movie received was just the amount of upvotes the comment containing the movie received. Upvotes are a form of "like" in Reddit, so we assumed we could be more confident that suggestions with upvotes are good since more than one user thought the suggestion was good. Out of the 200 longest posts, only 79 had 5 suggested movies with at least 3 upvotes. We then manually checked the 79 posts and removed 12 since they were either (1) not a movie suggestions request or (2) had been edited to include the movies the person who made the request decided to watch (i.e. the request was contaminated with the movie recommendations we wanted our system to find by itself).

Finally, we used GPT-4 to generate a profile for each of the 67 posts. The prompt used to generate the profiles was the following: "Your task is to use this user's movie request to create a description of the user's movie preferences. If they give examples, mention them in the description. User Request: [POST] User Description:", where [POST] was substituted by a post out of the 67. An example of a Reddit post and its respective profile can be seen in Figure 4.

Each profile was given 5 positive labels (i.e. good movie recommendations), sampled from the group of upvoted movie suggestions. We then assigned 5 negative labels to each profile by first

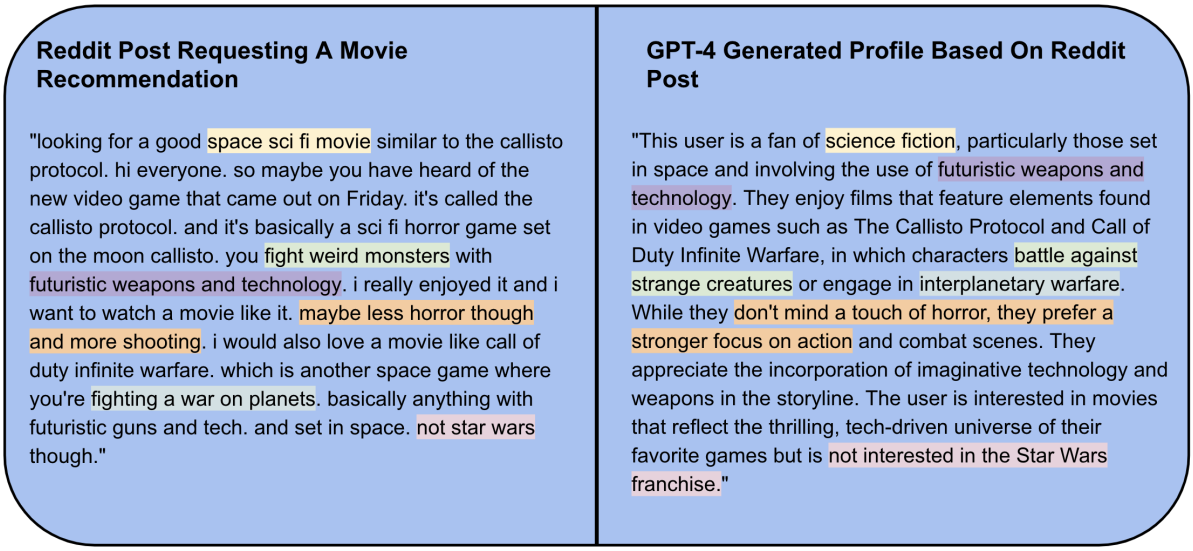

Figure 4: An example of a post requesting movie suggestions and the profile generated from it. Examples of movie preferences that are represented in both the post and the profile are highlighted.

creating an embedding for each post using the MPNET model from Reimers and Gurevych (2019), then sampling 5 movies from the positive labels of the 5 furthest posts according to cosine distance.

Lastly, a survey was conducted to certify that the profiles generated by GPT-4 represented the preferences in their respective posts. Given the survey's results, 3 profiles were removed. For details on the survey see Appendix A.3

A.2 Books Dataset

The Books Dataset was constructed in a similar way to the Movies Dataset. It was adapted from an existing dataset of book suggestion requests scraped from Reddit, filtered in terms of length and availability of good suggestions, manually checked, and tested via a crowdsourced survey.

There were, however, a couple of differences. Firstly, the dataset with the posts and book suggestions (Penha and Hauff, 2020) did not contain the upvotes but instead a "relevancy score" which serves a similar function. A book suggestion was considered good if its relevancy score was 10 or higher. By filtering out all the book requests that did not have at least 5 suggestions with a relevancy score of 10 or more, we decreased the number of posts in consideration from 44100 to 279. We then only considered the 60 longest posts out of the 279. We manually checked each of the book requests, generated profiles with GPT-4 with the same prompt as the one used for the Movies Dataset (substituting the word "movie" with "book"), and ran a survey to ensure the profiles matched the posts (details in Appendix A.3). Finally, we ran the same procedure for sampling negative labels for each profile as we did for the Movies Dataset, except instead of embedding the posts we embedded the profiles. We ended up with a total of 42 profiles, each with 5 positive and 5 negative labels.

A.3 Crowdsourced Evaluation

We used Prolific, a crowdsourcing platform, to evaluate how well the generated profiles represented the movie/book preferences in the posts. For each profile, we assigned at least 10 workers to evaluate all the statements that composed the profile. Each statement was given a score of "1" for "Yes," "0" for "Maybe" and -1 for "No" in regards to whether the contents of the statement were reflected in the post (See Figure 5).

A preponderance of agreement would have been any positive number, but we chose 0.6 as a threshold for automatically accepting the statement indicating that the average response was more definitely positive (1) than uncertain (0) for each statement. We manually reviewed all statements and posts with scores under 0.6, deciding whether we should just remove the statement from the profile or remove the entire profile from the dataset.

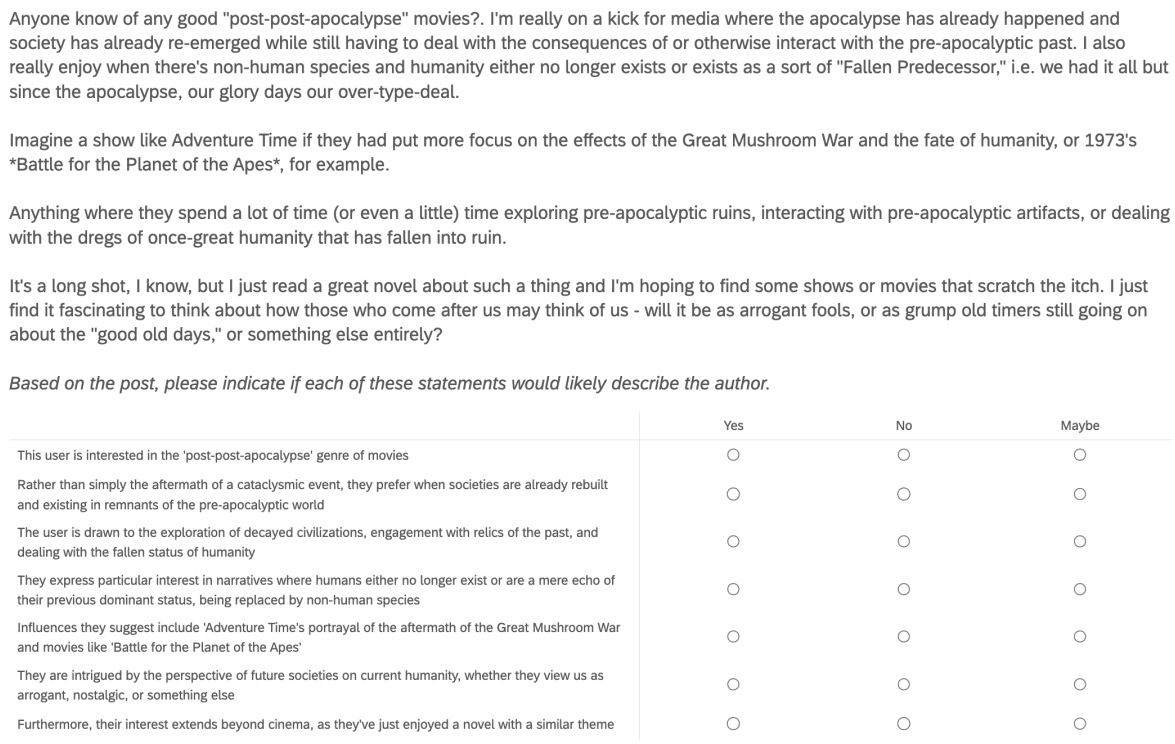

Figure 5: An example of a question in the survey that was used to evaluate if the generated profiles matched the posts they were based on.

B Prompts

Below are all the prompts used for the simulated users, Elicitation Policy, and Recommender.

B.1 Simulated User

The prompt used to simulate a user was the following:

"User: [PROFILE] Task: Answer the question as if you were the user. Answer the question in the shortest way with minimal additional explanation. Question: [QUESTION] Answer:"

Where [PROFILE] would be substituted by some profile in the dataset and [QUESTION] would be substituted by the latest question generated by the Elicitation Policy.

B.2 Elicitation Policy

The Baseline Elicitation Policy used the following prompt:

"Ask a question to help determine what [DOMAIN ITEM]s to recommend to a user. Your task is to come up with the best question possible to discover the user's preferences when it comes to [DOMAIN ITEM]s. Previous questions: [HISTORY] As you saw above, you already asked the user some questions and you should not repeat them. Make sure the question is short and friendly. Focus on trying to discover more information:"

Where [DOMAIN ITEM] would be "online article", "movie", or "book" depending on the domain, and [HISTORY] would be substituted by the question-answer pairs from the previous rounds of the conversation. For the first round, where there are no previous question-answer pairs, the prompt used was the following:

"Ask a question to help determine what [DOMAIN ITEM]s to recommend to a user. Ask a broad question to discover a lot about the user's preferences when it comes to [DOMAIN ITEM]s. Make sure the question is short and friendly. Focus on trying to discover more information:"

The structure of [HISTORY] was taken from Li et al. (2023a). Pairs consisted of the question asked, followed by an arrow symbol "->", and then the

answer to the question. For example, "Do you like Horror movies? -> Yes". The question-answer pairs were separated from one another by newlines and ordered from oldest to newest.

The exact **Predetermined Initial Question** used was the following:

"What are some of your favorite [DOMAIN ITEM]s? If you can't think of examples, what kind of [DOMAIN ITEM] do you like?"

Where [DOMAIN ITEM] is the same as previous prompts. The second clause, "what kind of [DOMAIN ITEM] do you like", was added so the question would be robust to profiles that had no examples of previous articles/movies/books that fit their preferences.

The Elicitation Policy with **Item Centered Questions** used the following prompt:

"You need to determine what [DOMAIN ITEM]s to recommend to a user. Your task is to come up with the best question possible to discover the user's preferences when it comes to [DOMAIN ITEM]s. Previous questions: [HISTORY] As you saw above, you already asked the user some questions and you should not repeat them. Ask one of two types of questions. 1. Ask a broad question to get the user to elaborate more about what [DOMAIN ITEM]s they are looking for. 2. Ask a question about one of the [DOMAIN ITEM]s the user mentioned to get further insight into the user's preferences regarding [DOMAIN ITEM]s. Make sure the question is short and friendly. Focus on trying to discover more information:"

Where [DOMAIN ITEM] and [HISTORY] are the same as previous prompts, except in the first round instead of using a separate prompt [HISTORY] is just empty.

B.3 Recommender

The prompt used for **The Baseline Recommender** was the following:

"A user has a particular set of preferences over what [DOMAIN ITEM]s they want to [DOMAIN ACTION]. They have specified their preferences below: [HISTORY] Based on these preferences, would the user be interested in [DOMAIN ACTION]ing the following [DOMAIN ITEM]? Answer with a probability between 0 and 1, where 0 means 'definitely not interested' and 1 means 'definitely interested'. Only output the probability and nothing else. If uncertain, make your best guess. [SUGGESTION]"

Where [DOMAIN ITEM] and [HISTORY] are the same as previous prompts, [DOMAIN ACTION] would be "read" or "watch" depending on the domain, and [SUGGESTION] would be one of the items in the profile's positive/negative labels.

With **Known Answer Prompting**, we used the same prompt as the baseline recommender, except we would include the following text before "Answer with a probability between 0 and 1...":

"The preferences above have examples of the kind of [DOMAIN ITEM]s the user likes. If the [DOMAIN ITEM] below is similar to any one of the [DOMAIN ITEM]s above, that means the user is interested."

For **Incremental Prediction** the prompt from the Baseline Recommender was used for the first round. For all subsequent rounds, two prompts were used in sequence. The first of these prompts was the following:

"A user has a particular set of preferences over what [DOMAIN ITEM]s they want to [DOMAIN ACTION]. They have specified their preferences below: [PREV HISTORY] Based on these preferences, the probability is [PREV PROB] out of 1.0 that the user was interested in the following [DOMAIN ITEM]: [SUGGESTION] You have asked the user another question and gotten the following answer: [NEWEST PAIR] How does this information change the probability the user is interested in the [DOMAIN ITEM]? Let's think step by step."

Where [DOMAIN ITEM], [DOMAIN ACTION], and [SUGGESTION] are the same as previous prompts. [PREV HISTORY] would be substituted by the previous question-answer pairs except for the newest pair, while [NEWEST PAIR] would be the newest pair. [PREV PROB] would be the probability predicted for [SUGGESTION] in the previous round of conversation.

This prompt would be used to generate a rationale which would be appended to the context. Then, together with the first prompt and the rationale, the following text would be appended to the context to create the final prompt:

"What is the probability the user is interested in the [DOMAIN ITEM] now? Answer with a probability between 0.0 and 1.0, where 0.0 means 'definitely not interested' and 1.0 means 'definitely interested'. If uncertain, make your best guess. Only output the probability and nothing else. Probability:"