Filtering Discomforting Recommendations with Large Language Models

Jiahao Liu
Fudan University
Shanghai, China
jiahaoliu23@m.fudan.edu.cn

Yiyang Shao
Fudan University
Shanghai, China
yyshao22@m.fudan.edu.cn

Peng Zhang*
Fudan University
Shanghai, China
zhangpeng_@fudan.edu.cn

Dongsheng Li
Microsoft Research Asia
Shanghai, China
dongsli@microsoft.com

Hansu Gu
Independent
Seattle, United States
hansug@acm.org

Chao Chen
Shanghai Jiao Tong University
Shanghai, China
chao.chen@sjtu.edu.cn

Longzhi Du
Alibaba
Shanghai, China
du.dlz@alibaba-inc.com

Tun Lu*
Fudan University
Shanghai, China
lutun@fudan.edu.cn

Ning Gu
Fudan University
Shanghai, China
ninggu@fudan.edu.cn

Abstract

Personalized algorithms can inadvertently expose users to discomforting recommendations, potentially triggering negative consequences. The subjectivity of discomfort and the black-box nature of these algorithms make it challenging to effectively identify and filter such content. To address this, we first conducted a formative study to understand users' practices and expectations regarding discomforting recommendation filtering. Then, we designed a Large Language Model (LLM)-based tool named DiscomfortFilter, which constructs an editable preference profile for a user and helps the user express filtering needs through conversation to mask discomforting preferences within the profile. Based on the edited profile, DiscomfortFilter facilitates the discomforting recommendations filtering in a plug-and-play manner, maintaining flexibility and transparency. The constructed preference profile improves LLM reasoning and simplifies user alignment, enabling a 3.8B open-source LLM to rival top commercial models in an offline proxy task. A one-week user study with 24 participants demonstrated the effectiveness of DiscomfortFilter, while also highlighting its potential impact on platform recommendation outcomes. We conclude by discussing the ongoing challenges, highlighting its relevance to broader research, assessing stakeholder impact, and outlining future research directions.

CCS Concepts

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

Keywords

discomforting recommendation filtering, large language model

ACM Reference Format:
Jiahao Liu, Yiyang Shao, Peng Zhang, Dongsheng Li, Hansu Gu, Chao Chen, Longzhi Du, Tun Lu, and Ning Gu. 2025. Filtering Discomforting Recommendations with Large Language Models. In *Proceedings of the ACM Web Conference 2025 (WWW '25), April 28-May 2, 2025, Sydney, NSW, Australia.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3696410.3714850

*Corresponding author.

WWW '25, Sydney, NSW, Australia

ACM ISBN 979-8-4007-1274-6/25/04
https://doi.org/10.1145/3696410.3714850

1 Introduction

Personalized algorithms, which analyze user preferences to deliver tailored content and thereby support human decision-making, are indispensable across web platforms [16–20, 43]. While these algorithms are designed to enhance user experience, they can inadvertently expose users to discomforting recommendations [24]. For example, if a user searches for sensitive topics like health issues, the algorithm might suggest related health products, which could be perceived as a breach of privacy and cause unease [38]. Similarly, when a user is experiencing emotional distress, such as after a breakup, algorithms lacking contextual awareness might recommend content that evokes painful memories, potentially worsening the user's emotional state [26]. Such recommendations may not only fail to engage users but also lead to negative emotional consequences, such as anxiety, unease, or distress [24]. The perception of discomfort is highly subjective, meaning that content one user finds enjoyable may be discomforting to another [24, 25, 34, 36]. This subjectivity underscores the urgent need for a more nuanced approach to discomforting content identification that aligns more closely with individual user experiences [10, 27, 31].

In this paper, we aim to design a tool that helps users filter out discomforting recommendations. This task has two key challenges: (1) users' perceptions of discomfort are highly subjective, and (2) the algorithms recommending such content operate as black-box systems. As illustrated in Figure 2, we formalize the problem as follows: black-box personalized algorithms recommend items to a user based on the inferred preference profile (often implicit in

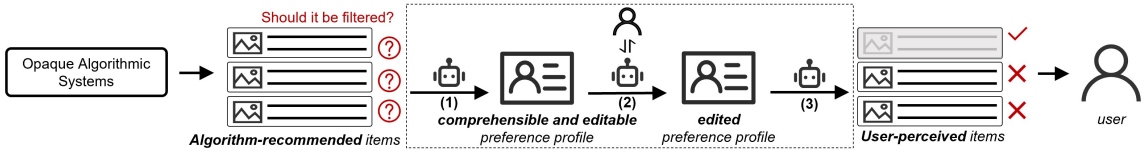

Figure 1: The workflow of DiscomfortFilter.

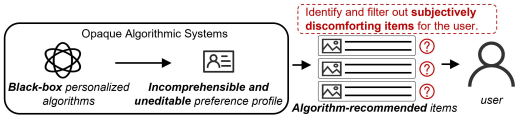

Figure 2: Problem formulation.

the embeddings), and our objective is to identify and filter out subjectively discomforting items for the user. Due to the opacity of algorithmic systems, the preference profile is both incomprehensible and uneditable, making it challenging for the user to influence the algorithm's decisions.

To inform our design process, we conducted a formative study to gain insights into the current landscape and user expectations (Section 3). Initially, we identified several key factors contributing to discomforting recommendations, including deviations in user behavior, biases in algorithmic modeling, and conflicting interests among stakeholders. We then examined the limitations of current feedback mechanisms, emphasizing shortcomings such as insufficient personalization, inflexibility, and a lack of transparency. Based on these findings, we established four design goals to guide the design of our tool: support conversational configuration, provide preference explanations, provide feedback channels, and operate in a plug-and-play manner.

Given the natural language understanding, reasoning, and generation capabilities demonstrated by LLMs [15, 47], we propose that LLM provide a promising solution for achieving these design goals. To this end, we designed an LLM-based tool named DiscomfortFilter, specifically aimed at helping users filter out discomforting recommendations (Section 4). Figure 1 illustrates the workflow of DiscomfortFilter: (1) DiscomfortFilter identifies algorithm-recommended items based on the user's personalized perceptions, integrates the user's pairwise preferences, and ranks them to construct a comprehensible and editable preference profile tailored for the user; (2) Through a guided conversation, DiscomfortFilter assists the user in expressing personalized filtering needs, and then masks the discomforting preferences with in the profile; (3) DiscomfortFilter filters out discomforting recommendations based on the edited preference profile in a plug-and-play manner, ensuring that user-perceived items no longer includes discomforting elements. Additionally, DiscomfortFilter provides the user with access to filtering logs recorded during step (3), assisting the user in refining filtering needs in a manner similar to step (2). Overall, DiscomfortFilter empowers the user to actively influence the decisions made by personalized algorithms, enhancing control over the algorithms.

We validated the efficacy of the constructed preference profile using an offline proxy task. The findings suggest that it enhances the reasoning process of LLMs, markedly decreases the challenge of aligning them with users, and allows a 3.8B open-source LLM to rival top commercial models. Additionally, we conducted a one-week user study on Zhihu (a platform similar to Quora), China's largest Q&A community, with 24 participants (Section 5). The results demonstrate that our design goals effectively help users express their filtering needs and filter out discomforting recommendations, with DiscomfortFilter successfully achieving these goals. We also analyzed how DiscomfortFilter impacts platform recommendation outcomes by influencing the exposure of discomforting items. Finally, we conducted an in-depth discussion (Section 6), covering: challenges of filtering discomforting recommendations with LLMs, relevance to research topics in recommender systems, potential impact on platforms, and limitations and future work.

The key contributions of this work are outlined below:

- We conducted a formative study with 15 participants to examine the current status and user expectations regarding the filtering of discomforting recommendations.
- We designed an LLM-based tool named DiscomfortFilter to assist users in filtering out discomforting recommendations, which is the first attempt to leverage LLMs in this important task, to the best of our knowledge.
- We evaluated DiscomfortFilter through an offline proxy experiment and a user study and the results showed that DiscomfortFilter can effectively help users express their filtering needs and filter out discomforting recommendations.
- We discussed the challenges and opportunities of using LLMs for filtering discomforting recommendations and shed light on its broader implications.

2 Related Work

Personalized algorithms primarily derive user preferences from behavioral data, leading to incomplete user modeling and discomforting recommendations [29]. In Appendix E, we provide a detailed review of studies illustrating this phenomenon across various scenarios, including privacy invasion [38], lack of contextual understanding [26], popularity bias [6], and information bubbles [30], along with their negative outcomes [36]. Although previous research has identified scenarios and causes of discomfort, **few studies have focused on designing systems to identify and filter these recommendations, which our work aims to address**.

In Appendix E, we provide a detailed review of two potential solutions for filtering discomforting recommendations: interactive recommendation systems [7, 11, 21] and content moderation systems [3, 9, 12]. Both allow users to modify recommendations to mitigate discomfort. However, interactive systems often struggle to scale in opaque environments due to their reliance on specific algorithm designs, while moderation systems focus on objectively

harmful content, which may be less effective for addressing subjective discomfort. Thus, **the subjectivity of discomfort perception and the opacity of algorithms present notable challenges**.

In Appendix E, we present a detailed review of studies on LLMs as personal assistants [15], including commercial applications [23] and frameworks for human-centered recommender systems [33]. The impressive capabilities of LLMs, especially in handling personal data and services, underscore their potential for effectively filtering discomforting recommendations. **To our knowledge, this is the first exploration of using LLMs for this specific purpose.**

3 Formative Study

We conducted semi-structured interviews and participatory design sessions with **15** participants, each lasting approximately **one hour**. Additional details about the process are provided in Appendix A.

3.1 Findings from Semi-Structured Interviews

During the semi-structured interviews, we explored participants' experiences with discomforting recommendations and the issues they faced when using the "Not Interested" button[1] for feedback. Our analysis identified two key findings from their responses.

F1: Users may encounter discomforting recommendations for three reasons. (1) User behavior deviation. Curiosity-driven search behavior and clickbait-induced clicks may fail to reflect a user's true long-term interests, leading inaccurate user preference modeling. For example, P03 said "*Out of curiosity, I once searched for adult products, and now they keep showing up in my recommendations—so embarrassing.*" **(2) Algorithmic modeling bias.** Personalized algorithms cannot fully capture the nuanced interests[2] and contexts of users. For example, P06 said "*Getting horror content at night is awful, even if I watch it during the day.*" **(3) Conflicting interests.** For instance, platforms may promote content designed to boost user engagement, even if it may cause discomfort. Seven participants mentioned scenarios where this was the case.

F2: Platforms' "Not Interested" button faces three major limitations that reduce user engagement. (1) Lack of personalization. Thirteen participants found the options too vague, making it difficult for them to articulate their specific reasons and potentially leading to the unintended exclusion of content they might otherwise enjoy. **(2) Lack of flexibility.** Eleven participants expressed concern that content they temporarily wish to hide might be permanently removed from their feed. **(3) Lack of transparency.** Twelve participants expressed dissatisfaction with the uncertainty about whether their feedback was being processed effectively.

3.2 Design Goals Established through Participatory Design

In response to the issues mentioned above, during the participatory design process, we further discussed participants' specific expectations for the tool and summarized the following four design goals.

G1: Support conversational configuration. Thirteen participants indicated that they prefer expressing their filtering needs using natural language because it "*isn't limited by predefined options*" (P11) and "*allows for more accurate and personalized expression*" (P04). Additionally, ten participants expressed a desire to communicate their filtering needs through conversation with the tool, as it feels "*more natural*" (P09). By supporting conversational configuration, the issue of "lack of personalization" is addressed.

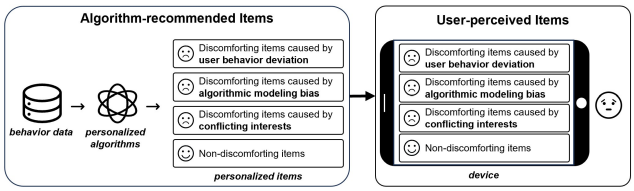

Figure 3: The process of presenting items to a user before introducing DiscomfortFilter.

G2: Provide preference explanations. As the input shifts from predefined options to open conversations, 10 participants reported difficulty in proactively articulating filtering needs. However, all participants agreed that understanding the preferences reflected in platform recommendations and their own behavior would encourage them to express these needs. For example, P05 said, "*I can review it and then provide targeted feedback on any inaccuracies.*"

G3: Provide feedback channels. All participants expressed the need for the tool to exhibit transparency and be contestable. Being informed about the filtered content and the corresponding reasons can "*enhance trust in the tool*" (P13), while allowing corrections to the tool's behavior helps "*refine filtering needs*" (P12). By providing feedback channels, the issue of "lack of transparency" is addressed.

G4: Operate in a plug-and-play manner. The plug-and-play approach means that the tool operates independently of specific personalized algorithms and directly affects the outputs of these algorithms. Three key factors support this: (1) Participants recognized that their filtering needs are dynamic, as "*discomforting content varies by state*" (P06); (2) Nine participants highlighted that the tool should be user-managed, enabling it to "*work across platforms*" (P14); (3) Participants were more concerned with the discomfort caused by personalized algorithmic outputs than with understanding the algorithms themselves. By operating in a plug-and-play manner, the issue of "lack of flexibility" is addressed.

4 DiscomfortFilter

We designed and implemented an LLM-based tool, DiscomfortFilter, which meets the four design goals established in the formative study and aims to assist users in filtering discomforting recommendations. The workflow of DiscomfortFilter has already been illustrated in Figure 1, and this section will provide a detailed introduction.

4.1 Overview

Figure 3 illustrates how personalized algorithms present items to a user before the introduction of DiscomfortFilter. These algorithms analyze user behavior to recommend items, which may include both discomforting and non-discomforting items. The personalized items is then displayed on the user's device for **passive consumption**.

[1]Figure 8 shows the button interfaces of the four platforms mentioned most frequently.
[2]This stems from the fact that collaborative filtering algorithms mainly retain low-frequency information during model training [32].

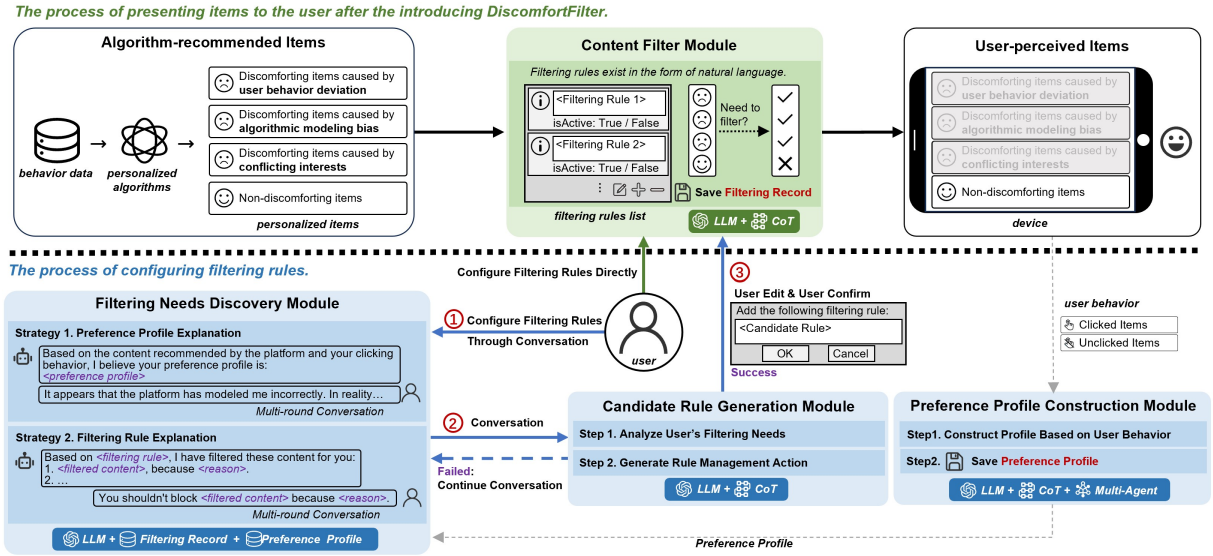

Figure 4: Detailed design of DiscomfortFilter.

The upper part of Figure 4 illustrates how personalized algorithms present items to the user after the introduction of DiscomfortFilter. By integrating a *Content Filter Module* into the original items presentation flow, DiscomfortFilter removes discomforting recommendations, ensuring that only non-discomforting items are ultimately displayed to the user. The identification of discomfort is based on user-configured *filtering rules*, giving the user **an ability of control** over the process.

The filtering rules, existing in natural language form, are crucial for the operation of DiscomfortFilter. The lower part of Figure 4 illustrates how the user configures these rules. The user can manage them directly (green arrow) or utilize the *Filtering Needs Discovery Module* for conversational rule configuration (blue arrow). This conversational agent employs two strategies to help the user identify filtering needs: the first strategy relies on the preference profile constructed by the *Preference Profile Construction Module*, while the second strategy relies on filtering records from the Content Filter Module. The *Candidate Rule Generation Module* analyzes the user's filtering needs from the conversations and translates them into management actions for the filtering rules, which the user can then edit and confirm.

4.2 Module Details

We provide a detailed introduction to the four modules that make up the DiscomfortFilter.

4.2.1 Content Filter Module. A user can manage filtering rules directly through this module. These rules are described in natural language, specifying the discomforting recommendations the user wishes to avoid. Each filtering rule has an associated activation option. During the filtering phase, the module reviews each recommendation against all active filtering rules to determine if the content matches any discomforting criteria. Only recommendations that do not match any filtering rules will be shown to the user; otherwise, they will be filtered out (G4). The identification process uses a chain-of-thought (CoT) method, with the prompt detailed in Appendix C. Filtering records will be saved and forwarded to the Filtering Needs Discovery Module.

4.2.2 Preference Profile Construction Module. This module constructs a user's preference profile by analyzing the user's clicking behavior on recommendations in chronological order. This process differs from traditional personalized algorithm research in three key aspects: (1) It solely models preferences based on **individual user behavior**, rather than on the behavior of all users. (2) It offers **more comprehensive implicit feedback** by capturing both clicked items and the recommended items users choose to ignore. (3) It must be conducted in **real-time**, responding instantly to user clicks. The key to this process is summarizing the user's clicking behavior on recommended content into a preference profile made up of features and maintaining that profile over time.

As illustrated in Figure 5, we propose an LLM-based multi-agent pipeline to complete this process. We adopt the general assumption of **pairwise rank learning** [28]—when two items are displayed simultaneously, the one that is clicked is more appealing to the user. For each clicked item (denoted as *pos* for positive), the module randomly samples an unclicked item that appeared simultaneously with *pos* as a negative item (denoted as *neg*), and then constructs an ordered pair <*pos*, *neg*>. Note that *pos* and *neg* are unprocessed raw contents. Then, this pair is processed by the pipeline.

First, the **Perceive Agent** identifies *pos* and *neg* from the user's perspective, analyzing why the user clicked on *pos* but not on *neg* based on the current preference profile. Different users focus on different aspects of the same item. **By incorporating a user's preference profile, the Perceive Agent can accurately identify the aspects that matter most to each individual user.**

Second, the **Summary Agent** distills the reasons for selecting *pos* over *neg* into m *pos features* and n *neg features*, drawing from the analysis of the Perceive Agent. It then forms $m \times n$ ordered pairs of

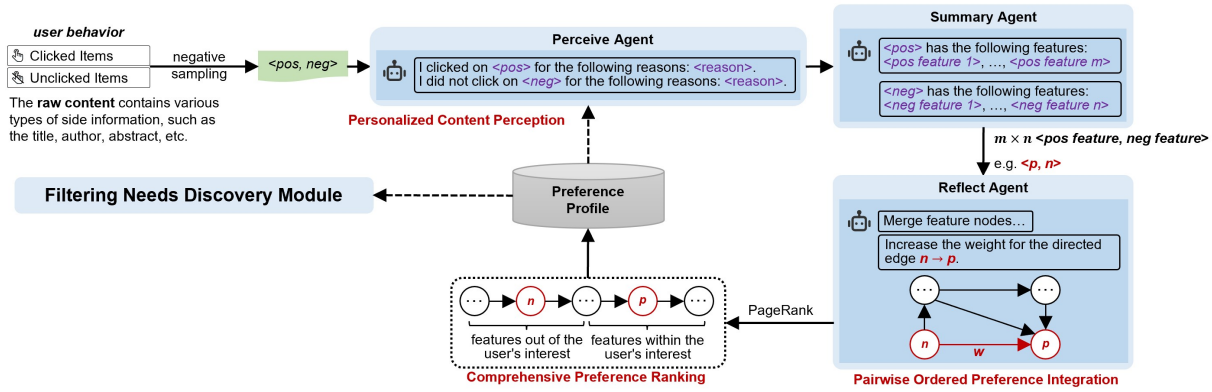

Figure 5: The Preference Profile Construction Module is a multi-agent pipeline powered by LLMs.

<pos feature, neg feature> via the Cartesian product, indicating that for each pair, the user prefers the *pos feature* over the corresponding *neg feature*. **The ordered pairs that describe the partial order relationships between these features are the fundamental basis for modeling user preferences.**

Third, the **Reflect Agent** maintains a directed graph, where edges point from *neg feature* to *pos feature*, with edge weights denoting the frequency of each *<pos feature, neg feature>* pair. Upon receiving pairs from the Summary Agent, it first merges similar feature nodes and subsequently integrates them into the graph. During the merge process, candidate features for merging are first identified based on semantic similarity, and then the final merge result is obtained using LLM. **The directed graph integrates independent ordered pairs from each user click, enabling the modeling of user preferences through comprehensive structural information.**

Finally, the feature nodes are ranked using the PageRank algorithm, which **provides a comprehensive ranking of preferences**. Features with higher rankings are generally more aligned with the user's interests, while lower-ranked features tend to be less relevant.

Overall, this Module has three key characteristics: (1) Preference profile is constructed from **features** (rather than raw contents); (2) The features are summarized through **personalized content perception**; (3) Features are **globally ranked** using the PageRank algorithm. **The preference profile constructed with these three designs helps refine the reasoning process of LLMs and significantly reduce the difficulty of aligning LLMs with user preferences**.

The preference profile is stored and sent to the Filtering Needs Discovery Module to improve the personalization of the conversational agent and provide users with clear explanations. The prompt used for this multi-agent pipeline can be found in Appendix C.

4.2.3 Filtering Needs Discovery Module. This module is a conversational agent designed to help users identify potential filtering needs with two strategies (G1). The content of each conversation round will be forwarded to the Candidate Rule Generation Module for further analysis.

Strategy 1: Preference Profile Explanation. This strategy begins by informing a user about the preference profile constructed by the Preference Profile Construction Module (G2). The user can then engage in multiple rounds of conversation with the conversational agent to express filtering needs, particularly where the preference profile do not match the user's expectations.

Strategy 2: Filtering Record Explanation. This strategy begins by informing a user about the filtering records from the Content Filter Module, including the filtered content and the reasons for filtering (G3). The user can then engage in multiple rounds of conversation with the conversational agent to refine filtering rules, particularly where the filtering records do not match the user's expectations.

It is important to emphasize that integrating the preference profile aligns the conversational agent with the user, ensuring that interactions between them are highly personalized.

4.2.4 Candidate Rule Generation Module. During the interaction between the conversational agent and a user, this module continuously analyzes the user's filtering needs from the conversation. Once these needs are successfully identified, the module evaluates their relevance to existing filtering rules and generates corresponding management actions. By generating actions based on relevance, the module helps prevent conflicts and redundancy. For example, if the new filtering needs are related to existing rules, the module will generate an update action rather than create a new rule. The user can then edit and confirm the generated management actions. If no management action is confirmed (either because no needs were identified or because the user did not confirm), the conversational agent will continue interacting with the user. Figure 10 provides a detailed illustration of the above process through a flowchart.

4.3 Summary

Contestability refers to the ability to challenge decisions made by algorithms, serving as a crucial safeguard against power imbalances between users and algorithms [21]. However, this concept is often overlooked [21]. Although DiscomfortFilter does not interfere with the platform's algorithm, it fundamentally aims to **enhance user-perceived contestability in interactions with personalized algorithms**. By constructing a comprehensible and editable

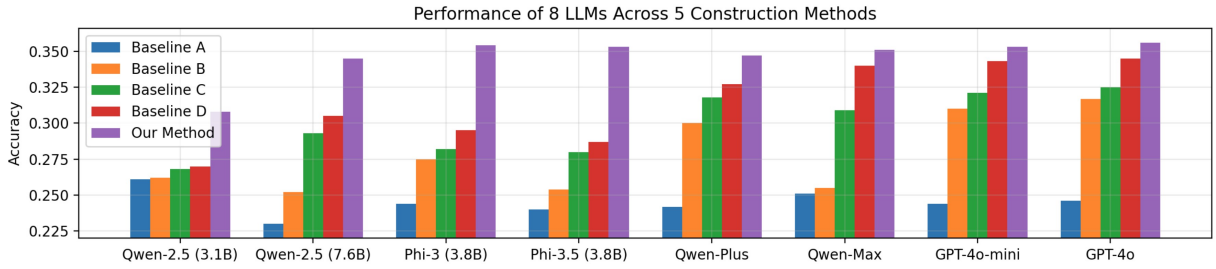

Figure 6: Performance of 8 LLMs across 5 preference profile construction methods with K=4.

Table 1: The statistics of participants configure the filtering rules through the Content Filter Module and the two strategies of the conversational agent.

	# Messages	# Add	# Update
Content Filter Module	-	2.9	2.1
Strategy 1	13.0	4.4	2.7
Strategy 2	22.3	3.2	7.3

Table 2: Confusion matrix.

	Predicted Positive	Predicted Negative
Actual Positive	173	32
Actual Negative	67	208

preference profile and allowing the user to mask any discomforting preferences, DiscomfortFilter **narrows the gap between algorithmic recommendations and user expectations**. Importantly, DiscomfortFilter inherently possesses contestability—**it does not introduce any new uncontestability while enhancing contestability in personalized algorithms**.

5 Evaluation

We first validate the Preference Profile Construction Module through offline experiments, then conduct a user study to assess whether the design goals help users filter discomforting recommendations and whether DiscomfortFilter meets those goals.

5.1 Effectiveness of Preference Profile Construction Module

5.1.1 Task. We validate the effectiveness of the Preference Profile Construction Module through a proxy task. Specifically, when presenting K items to a user, we instruct the LLMs to predict which item the user is most likely to click based on the preference profile, with accuracy as the evaluation metric. If the preference profile is sufficiently effective, the LLMs should accurately predict the user's behavior. For each interaction sample from a user (in chronological order), DiscomfortFilter will initially predict the user's behavior and then observe the actual click behavior to continuously update the preference profile. Notably, DiscomfortFilter only accesses the interaction records of the individual user during this process.

5.1.2 Dataset. We conducted offline experiments using the MIND dataset [41], which details negative samples recommended to users during interactions, along with side information. Given the high cost of API calls, we sampled a subset of ~5,000 users for experimentation. To account for the varying interaction frequencies, we first grouped users by interaction frequency into intervals: $[0, 10)$, $[10, 20)$, and up to $[100, +\infty)$. Then, from each group, we randomly selected users, ensuring a total of at least 10,000 interactions per group.

5.1.3 Baselines. We conducted an ablation study comparing four heuristic in-context learning baselines for preference profile construction: (A) Preference profile exclude any user-specific information; (B) Raw contents are directly used as preference profile; (C) Features extracted from raw contents are used to construct preference profile; (D) Features are globally ranked using PageRank, but without adapting to personalized perception.

5.1.4 Results. We used four open-source LLMs (Qwen-2.5 (3.1B), Qwen-2.5 (7.6B), Phi-3 (3.8B), and Phi-3.5 (3.8B)) and four commercial LLMs (Qwen-Plus, Qwen-Max, GPT-4o-mini, and GPT-4o) for evaluation. Figure 6 shows their performance across various construction methods, with our method consistently achieving the best results, thus demonstrating its superior effectiveness.

To understand the reasons behind the inferior performance of other baselines, we conducted an analysis as follows: *A* fails to incorporate user-specific data, leading to essentially random predictions; *B* lacks a summary of features, and the LLMs struggle to deliver precise predictions based solely on raw content; *C* omits a summary of unclicked features, which prevents LLMs from fully modeling user preferences; *D* fails to perceive content in a personalized manner, leading to an inability to capture truly important features. **Our findings indicate that simply providing LLMs with extensive input is insufficient; instead, carefully curated and personalized inputs are essential for optimal performance.**

The profiles constructed by several ablation baselines require sufficiently powerful LLMs to achieve better performance, leading commercial models to outperform open-source LLMs on these baseline methods. However, **our construction method reduces task complexity by streamlining the reasoning process through a well-constructed preference profile**. This enhancement enables open-source models, e.g., Phi-3, to match or even surpass the performance of certain commercial models, e.g., GPT-4o-mini.

5.2 User Study

5.2.1 Implementation and Usage. We implemented DiscomfortFilter as a third-party tool on Zhihu[3], the largest Chinese Q&A community (similar to Quora[4]). We selected Zhihu because participants frequently mentioned it during the formative study, and many participants indicated that they regularly browse personalized content on it. Furthermore, Q&A platforms are rich in user-generated content and diverse topics, and the three types of discomforting recommendations mentioned in F1 have all been observed on Zhihu.

DiscomfortFilter was implemented as a browser extension that filters discomforting recommendations by dynamically editing the DOM. We deployed Qwen2-72B-Instruct to provide LLM services for DiscomfortFilter. Aside from the LLM service, all other services run on user devices, with data stored locally. We provide three user stories and their corresponding interfaces in Appendix D to show the implementation and usage of DiscomfortFilter.

5.2.2 Process. We recruited **24** participants to use DiscomfortFilter freely for **one week**. Afterward, participants completed a questionnaire and randomly selected 10 filtered and 10 unfiltered items to label whether DiscomfortFilter correctly identified them. Finally, we collected usage statistics from the participants' devices and conducted **30-minute** interviews with each participant. Additional details about the process are provided in Appendix B.

5.2.3 Preliminary Statistical Results. On average, DiscomfortFilter processed 1,093 items per participant, filtering out 124 of them. Table 1 presents the interaction data between participants and DiscomfortFilter. The overall acceptance rate for the management actions on candidate filtering rules generated by DiscomfortFilter (via strategy 1 and strategy 2) was 93.8%.

5.2.4 Results of the questionnaire. Guided by the Technology Acceptance Model (TAM) [4, 22], we developed three evaluation questions for each design goal and the overall tool, focusing on perceived usefulness (PU), perceived ease of use (PEOU), and behavioral intention (BI). Figure 9 shows the specific questions and the corresponding score distributions. For each question, at least three-quarters of participants rated it 4 or 5, indicating overall satisfaction with the role of DiscomfortFilter in assisting users to filter out discomforting recommendations. With a Cronbach's α of 0.80, we believe the results are reliable for subsequent analysis.

5.2.5 Results of the interview. We compared the platform's "Not Interested" button with DiscomfortFilter and asked participants for their opinions. All participants unanimously preferred DiscomfortFilter for filtering discomforting recommendations, which can be summarized into the following five key results.

R1. Natural language filtering rules helped participants in personalizing their filtering needs (G1). Some participants also found emotional value in being able to "*complain to the tool about the platform's recommendations*" (P31) and "*set rules regarding mood*" (P38). However, **false associations in LLMs sometimes hindered accurate identification of discomforting recommendations.** Seven participants noted that DiscomfortFilter occasionally overextended the rules, leading to unintended content filtering. This reduced precision — as shown in Table 2, where 24 participants annotated 480 recommended contents, resulting in a precision rate of 173/240=72.1% and a recall rate of 173/(173+32)=84.4%.

R2. Providing preference explanations helps participants identify and articulate their filtering needs (G2). Participants were sometimes unclear about their filtering needs, and the preference explanations enabled them to "*discuss with the tool how to establish rules*" (P31). However, two participants felt that **the preference profile lacked sufficient detail**, offering only a "*broad overview of the recommended content*" (P17).

R3. Feedback channels help participants refine their filtering needs and build trust in DiscomfortFilter (G3). Participants noted that some filtering needs were "*hard to express precisely in one attempt*" (P18). When DiscomfortFilter behaved unexpectedly, feedback channels convinced participants that "*the tool had the ability to evolve*" (P31), and encouraged them to "*refine the rules instead of abandoning the tool*" (P28).

R4. Participants exhibited varying usage habits for the two strategies in the conversational agent (G1, G2, G3). Participants indicated that configuring filtering rules was the most challenging aspect, and the conversational agent helped simplify this process. Most participants reported a preference for using strategy 1 to create new filtering rules because "*the gaps highlighted new filtering needs*" (P31), while strategy 2 was typically used to update existing rules due to "*errors prompting corrections to the filtering rules*" (P23). This observation is further supported by Table 1.

R5. The plug-and-play approach facilitates flexible configuration of dynamic filtering needs (G4), adapting to contextual changes and short-term interests. However, three participants expressed dissatisfaction with the absence of **an efficient method for managing numerous filtering rules**, noting that "*reviewing all filtering rules to decide which to activate is cumbersome*" (P33).

5.2.6 Case Study. To demonstrate how participants use DiscomfortFilter, we provide an example. After searching related topics, P35 found through strategy 1 that the platform mistakenly assumed she was interested in mother-in-law and daughter-in-law relationships, which also raised privacy concerns. She then used strategy 1 to set a filtering rule: "*I do not want to see content related to mother-in-law and daughter-in-law relationships.*" Later, through strategy 2, P35 realized that DiscomfortFilter had also unintentionally filtered out content from a novel she liked (caused by false association in LLMs). She then adjusted the rule using strategy 2: "*I do not want to see content related to mother-in-law and daughter-in-law relationships, except for the fictional content in novels.*"

5.2.7 Impact on platform recommendation outcomes. Assuming DiscomfortFilter processes N items using a filtering rule, with n items identified as discomforting, we define n/N as the *filtering burden* of this rule. We calculated the daily average filtering burden for all filtering rules that remained active for more than five days during the seven-day user study. The trend of daily average filtering burden from the day they were configured is shown in Figure 7. Over time, the filtering burden steadily declined, indicating that the platform was recommending progressively fewer discomforting items. This decline can be attributed to the introduction of DiscomfortFilter, which reduces users' exposure to discomforting recommendations, thereby resulting in fewer interactions with such

[3]https://www.zhihu.com/
[4]https://www.quora.com/

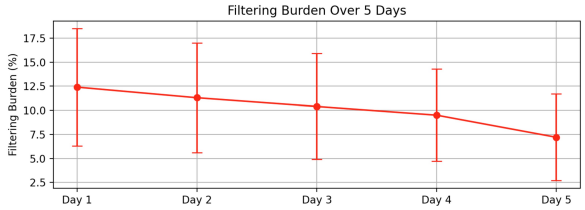

Figure 7: Average filtering burden over 5 days.

items over time. As a result, the platform's recommender systems can dynamically adjust to users' evolving preferences, further reducing the likelihood of discomforting items being recommended. It is important to emphasize that, from the users' perspective, discomforting recommendations vanish immediately, as they were filtered out entirely.

6 Discussion

6.1 Challenges of Filtering Discomforting Recommendations with LLMs

Our evaluation has identified two main challenges in filtering discomforting recommendations using LLMs. **(1) False association in LLMs.** Despite careful design, LLMs sometimes misinterpret non-discomforting recommendations as containing discomforting elements, leading to unintended exclusions (R1). While LLMs' associative abilities enhance creativity, they require careful control when making decisions for users. **(2) Insufficient perceptual alignment.** Despite our meticulous design of the Preference Profile Construction Module, LLMs struggle to fully grasp users' subjective experiences, undermining the effectiveness of profile explanation (R2). A significant gap remains in helping LLMs transition from "seeing what users see" to "perceiving what users perceive".

6.2 Relevance to Research Topics in Recommender Systems

Recent recommender system studies increasingly emphasize user experience. **Recommendation unlearning** [14, 45], a process that allows models to forget specific user interests, enhances transparency [44] and controllability [39]. **Context-aware recommendations** [1] improve user satisfaction by accurately modeling contextual factors. **Bias-mitigating recommendations** [2] address information cocoons by prioritizing diversity [13] and fairness [40].

Our study emphasizes human-centered aspects of recommender system design, enhancing previous research primarily focused on algorithmic design. Most existing recommendation unlearning techniques [14, 45] achieve only approximate forgetting; however, integrating them with DiscomfortFilter **enables exact interest forgetting**. Traditional context-aware recommendations [1] rely solely on passively collected data, such as spatio-temporal information, while combining with DiscomfortFilter can better **meet the personalized needs of users**. Recent studies indicate that the perceived diversity among users cannot be achieved merely by providing diverse recommendations but instead relies on their active exploration [46]. While the impact of our study on information cocoons remains uncertain, we believe that, with appropriate guidance, DiscomfortFilter can enhance users' understanding of recommended content and **help them escape these cocoons actively**.

6.3 Potential Impact on Platforms

While DiscomfortFilter does not directly modify the platform's algorithms, it influences the items that users encounter. This influence can alter user behavior and, in turn, affect the platform's data collection and user modeling indirectly. We believe that this influence is beneficial for two primary reasons. **First**, studies indicate that even minimal control over recommendations significantly enhances users' willingness to engage [5]. DiscomfortFilter empowers users with greater control over the recommendation process, fostering trust and increasing their willingness to use the platform. **Second**, preventing data collection from users' interactions with discomforting recommendations enables the platform to model users more accurately.

6.4 Limitations and Future Work

We present the limitations and potential improvements from four aspects: **(1) Performance.** The two challenges outlined in Section 6.1 primarily arise from LLMs not being aligned with users. One potential solution is to introduce an interactive verification process that aligns LLMs based on user feedback. **(2) Function.** Our study currently focuses exclusively on user click behavior and text content. Future developments could incorporate other behaviors and multimodal content. Additionally, an efficient rule management solution is necessary. **(3) Evaluation.** Most participants recruited for this study hold bachelor's degrees and were assessed on a single platform over a period of one week. A large-scale deployment is needed for a long-term evaluation that encompasses a broader demographic and multiple platforms. **(4) Application.** DiscomfortFilter could be extended to other scenarios, such as parental monitoring and controlling the online content accessible to children. This requires targeted design and careful consideration of legal and ethical issues.

7 Conclusion

Building on insights from a formative study, we developed an LLM-based tool named DiscomfortFilter to assist users in identifying and filtering discomforting recommendations from recommender systems. Results from an offline experiment and a user study demonstrated DiscomfortFilter's effectiveness. In-depth discussions illuminated future directions and the potential broad impact of our work. We believe our study can strengthen the recommender systems community's focus on human-centered design.

Acknowledgments

This work is supported by National Natural Science Foundation of China (NSFC) under the Grant No. 61932007, 62372113, and 62172106. Tun Lu is also a faculty of Shanghai Key Laboratory of Data Science, Fudan Institute on Aging, MOE Laboratory for National Development and Intelligent Governance, and Shanghai Institute of Intelligent Electronics & Systems, Fudan University.

References

[1] Gediminas Adomavicius and Alexander Tuzhilin. 2010. Context-aware recommender systems. In *Recommender systems handbook*. Springer, 217–253.

[2] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2023. Bias and debias in recommender system: A survey and future directions. *ACM Transactions on Information Systems* 41, 3 (2023), 1–39.

[3] Stefano Cresci, Amaury Trujillo, and Tiziano Fagni. 2022. Personalized interventions for online moderation. In *Proceedings of the 33rd ACM Conference on Hypertext and Social Media*. 248–251.

[4] Fred D Davis et al. 1989. Technology acceptance model: TAM. *Al-Suqri, MN, Al-Aufi, AS: Information Seeking Behavior and Technology Adoption* 205 (1989), 219.

[5] Berkeley J Dietvorst, Joseph P Simmons, and Cade Massey. 2018. Overcoming algorithm aversion: People will use imperfect algorithms if they can (even slightly) modify them. *Management science* 64, 3 (2018), 1155–1170.

[6] Andres Ferraro. 2019. Music cold-start and long-tail recommendation: bias in deep representations. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 586–590.

[7] Yingqiang Ge, Shuchang Liu, Zuohui Fu, Juntao Tan, Zelong Li, Shuyuan Xu, Yunqi Li, Yikun Xian, and Yongfeng Zhang. 2022. A survey on trustworthy recommender systems. *ACM Transactions on Recommender Systems* (2022).

[8] Ziwei Gu, Jing Nathan Yan, and Jeffrey M Rzeszotarski. 2021. Understanding user sensemaking in machine learning fairness assessment systems. In *Proceedings of the Web Conference 2021*. 658–668.

[9] Necdet Gurkan, Mohammed Almarzouq, and Pon Rahul Murugaraj. 2024. Personalized Content Moderation and Emergent Outcomes. *arXiv preprint arXiv:2405.09640* (2024).

[10] Richard Jackson Harris and Lindsay Cook. 2011. How content and co-viewers elicit emotional discomfort in moviegoing experiences: Where does the discomfort come from and how is it handled? *Applied Cognitive Psychology* 25, 6 (2011), 850–861.

[11] Chen He, Denis Parra, and Katrien Verbert. 2016. Interactive recommender systems: A survey of the state of the art and future research challenges and opportunities. *Expert Systems with Applications* 56 (2016), 9–27.

[12] Shagun Jhaver, Alice Qian Zhang, Quan Ze Chen, Nikhila Natarajan, Ruotong Wang, and Amy X Zhang. 2023. Personalizing content moderation on social media: User perspectives on moderation choices, interface design, and labor. *Proceedings of the ACM on Human-Computer Interaction* 7, CSCW2 (2023), 1–33.

[13] Matevž Kunaver and Tomaž Požrl. 2017. Diversity in recommender systems–A survey. *Knowledge-based systems* 123 (2017), 154–162.

[14] Yuyuan Li, Chaochao Chen, Xiaolin Zheng, Yizhao Zhang, Biao Gong, Jun Wang, and Linxun Chen. 2023. Selective and collaborative influence function for efficient recommendation unlearning. *Expert Systems with Applications* 234 (2023), 121025.

[15] Yuanchun Li, Hao Wen, Weijun Wang, Xiangyu Li, Yizhen Yuan, Guohong Liu, Jiacheng Liu, Wenxing Xu, Xiang Wang, Yi Sun, et al. 2024. Personal llm agents: Insights and survey about the capability, efficiency and security. *arXiv preprint arXiv:2401.05459* (2024).

[16] Jiahao Liu, Dongsheng Li, Hansu Gu, Tun Lu, Jiongran Wu, Peng Zhang, Li Shang, and Ning Gu. 2023. Recommendation unlearning via matrix correction. *arXiv preprint arXiv:2307.15960* (2023).

[17] Jiahao Liu, Dongsheng Li, Hansu Gu, Tun Lu, Peng Zhang, and Ning Gu. 2022. Parameter-free dynamic graph embedding for link prediction. *Advances in Neural Information Processing Systems* 35 (2022), 27623–27635.

[18] Jiahao Liu, Dongsheng Li, Hansu Gu, Tun Lu, Peng Zhang, Li Shang, and Ning Gu. 2023. Personalized graph signal processing for collaborative filtering. In *Proceedings of the ACM Web Conference 2023*. 1264–1272.

[19] Jiahao Liu, Dongsheng Li, Hansu Gu, Tun Lu, Peng Zhang, Li Shang, and Ning Gu. 2023. Triple structural information modelling for accurate, explainable and interactive recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1086–1095.

[20] Sijia Liu, Jiahao Liu, Hansu Gu, Dongsheng Li, Tun Lu, Peng Zhang, and Ning Gu. 2023. Autoseqrec: Autoencoder for efficient sequential recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 1493–1502.

[21] Henrietta Lyons, Eduardo Velloso, and Tim Miller. 2021. Conceptualising contestability: Perspectives on contesting algorithmic decisions. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW1 (2021), 1–25.

[22] Nikola Marangunić and Andrina Granić. 2015. Technology acceptance model: a literature review from 1986 to 2013. *Universal access in the information society* 14 (2015), 81–95.

[23] Yusuf Mehdi. 2023. Announcing Microsoft Copilot, your everyday AI companion. *Official Microsoft Blog* (2023).

[24] Miran Park, Kyuri Park, Hyewon Cho, Hwan Choi, and Hajin Lim. 2024. Exploring Design Approaches for Reducing Viewers' Discomfort with Distressing Short-form Videos. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. 1–8.

[25] James Pierce, Sarah Fox, Nick Merrill, and Richmond Wong. 2018. Differential vulnerabilities and a diversity of tactics: What toolkits teach us about cybersecurity. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW (2018), 1–24.

[26] Anthony T Pinter, Jialun Aaron Jiang, Katie Z Gach, Melanie M Sidwell, James E Dykes, and Jed R Brubaker. 2019. " Am I Never Going to Be Free of All This Crap?" Upsetting Encounters with Algorithmically Curated Content About Ex-Partners. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (2019), 1–23.

[27] Adelais Reichmann, Ines Bauda, Bettina Pfeffer, Andreas Goreis, Mercedes Bock, Paul Plener, Oswald D Kothgassner, et al. 2023. Post-Traumatic Stress after Corona Virus Disease 19 (COVID-19): The Role of Gender and Distressing Social Media Exposure as Risk Factors. *Digital Psychology* 4, 1 (2023), 14–26.

[28] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012).

[29] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2021. Recommender systems: Techniques, applications, and challenges. *Recommender systems handbook* (2021), 1–35.

[30] Fred Rowland. 2011. The filter bubble: what the internet is hiding from you. *portal: Libraries and the Academy* 11, 4 (2011), 1009–1011.

[31] Gautam Kishore Shahi and William Kana Tsoplefack. 2022. Mitigating harmful content on social media using an interactive user interface. In *International Conference on Social Informatics*. Springer, 490–505.

[32] Yifei Shen, Yongji Wu, Yao Zhang, Caihua Shan, Jun Zhang, B Khaled Letaief, and Dongsheng Li. 2021. How powerful is graph convolution for recommendation?. In *Proceedings of the 30th ACM international conference on information & knowledge management*. 1619–1629.

[33] Yubo Shu, Haonan Zhang, Hansu Gu, Peng Zhang, Tun Lu, Dongsheng Li, and Ning Gu. 2024. RAH! RecSys–Assistant–Human: A Human-Centered Recommendation Framework With LLM Agents. *IEEE Transactions on Computational Social Systems* (2024).

[34] Spandana Singh. 2019. Everything in moderation: An analysis of how Internet platforms are using artificial intelligence to moderate user-generated content. *New America* 22 (2019), 1–42.

[35] Jessie J Smith, Lex Beattie, and Henriette Cramer. 2023. Scoping fairness objectives and identifying fairness metrics for recommender systems: The practitioners' perspective. In *Proceedings of the ACM Web Conference 2023*. 3648–3659.

[36] Jessie J Smith, Lucia Jayne, and Robin Burke. 2022. Recommender systems and algorithmic hate. In *Proceedings of the 16th ACM Conference on Recommender Systems*. 592–597.

[37] Armin Toroghi, Griffin Floto, Zhenwei Tang, and Scott Sanner. 2023. Bayesian Knowledge-driven Critiquing with Indirect Evidence. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1838–1842.

[38] Blase Ur, Pedro Giovanni Leon, Lorrie Faith Cranor, Richard Shay, and Yang Wang. 2012. Smart, useful, scary, creepy: perceptions of online behavioral advertising. In *proceedings of the eighth symposium on usable privacy and security*. 1–15.

[39] Wenjie Wang, Fuli Feng, Liqiang Nie, and Tat-Seng Chua. 2022. User-controllable recommendation against filter bubbles. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*. 1251–1261.

[40] Yifan Wang, Weizhi Ma, Min Zhang, Yiqun Liu, and Shaoping Ma. 2023. A survey on the fairness of recommender systems. *ACM Transactions on Information Systems* 41, 3 (2023), 1–43.

[41] Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, et al. 2020. Mind: A large-scale dataset for news recommendation. In *Proceedings of the 58th annual meeting of the association for computational linguistics*. 3597–3606.

[42] Ga Wu, Kai Luo, Scott Sanner, and Harold Soh. 2019. Deep language-based critiquing for recommender systems. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 137–145.

[43] Jiafeng Xia, Dongsheng Li, Hansu Gu, Jiahao Liu, Tun Lu, and Ning Gu. 2022. FIRE: Fast incremental recommendation with graph signal processing. In *Proceedings of the ACM Web Conference 2022*. 2360–2369.

[44] Yongfeng Zhang, Xu Chen, et al. 2020. Explainable recommendation: A survey and new perspectives. *Foundations and Trends® in Information Retrieval* 14, 1 (2020), 1–101.

[45] Yang Zhang, Zhiyu Hu, Yimeng Bai, Fuli Feng, Jiancan Wu, Qifan Wang, and Xiangnan He. 2023. Recommendation unlearning via influence function. *arXiv preprint arXiv:2307.02147* (2023).

[46] Yu Zhang, Jingwei Sun, Li Feng, Cen Yao, Mingming Fan, Liuxin Zhang, Qianying Wang, Xin Geng, and Yong Rui. 2024. See Widely, Think Wisely: Toward Designing a Generative Multi-agent System to Burst Filter Bubbles. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–24.

[47] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223* (2023).

A The Detailed Process of the Formative Study

For the full version, visit https://arxiv.org/abs/2410.05411.

B The Detailed Process of the User Study

For the full version, visit https://arxiv.org/abs/2410.05411.

C Prompt

For the full version, visit https://arxiv.org/abs/2410.05411.

D User Stories

For the full version, visit https://arxiv.org/abs/2410.05411.

E Detailed Review of Related Work

For the full version, visit https://arxiv.org/abs/2410.05411.

F More Discussion

Here are some discussions that help readers gain a deeper understanding of our work.

F.0.1 Generalizability to non-text-based items. Expanding to non-text scenarios is a natural progression, requiring the use of multimodal models to understand such content. However, the efficiency of processing, especially for videos, remains a concern. Preprocessing non-text content offline may offer a viable solution.

F.0.2 Differences between discomforting and disliked items. We believe this primarily manifests in the impact perceived by users, with "discomforting" items having a greater impact. For uninteresting or disliked items, users may simply choose to ignore them. However, as we mentioned in the introduction, discomforting recommendations may not only fail to engage users but also lead to negative emotional consequences, such as anxiety, unease, or distress. This is a highly subjective matter and has a dynamic nature. For example, as illustrated in our formative study (Section 3.1, F1), one participant mentioned that he might search for horror content to watch during the day, but encountering the same horror content at night would make him feel very uncomfortable, even to the extent of affecting his sleep.

In our formative study, we observed that most users do not express strong hostility toward uninteresting (disliked) content but are more averse to "discomforting" content. Thus, from the users' perspective, filtering discomforting content is a more pressing issue than merely addressing disinterest, and this study focuses on addressing the former.

F.0.3 Discussion of critiquing-based recommender systems. Critiquing-based systems [37, 42] adjust recommendations end-to-end based on feedback, whereas our work adopts a plug-and-play filtering approach, enabling immediate real-world applicability. As noted in Section 6.2, our work complements algorithmic designs like critiquing systems.

F.0.4 Computational overhead and scalability. There are three areas where LLMs are needed: (1) to determine whether an item should be filtered, (2) for conversational agents, and (3) to build preference profiles. Among these, (1) and (3) have higher real-time requirements.

For (1), assume a user has m filtering rules. Processing each item requires $1 + m$ LLM calls, including one call for analyzing the item and m calls to check individually whether the item should be filtered according to each filtering rule. In the user study, we recruited 24 participants, and they did not notice any significant delay with the LLM service during their use.

For (3), suppose the algorithm recommends N items, and a user clicks on n of them ($N \gg n$). One direct way to construct a user's preference profile is to analyze all the items recommended by the algorithm, treating the clicked items as positive samples and the unclicked ones as negative samples. This process is similar to pointwise matrix factorization in collaborative filtering, with a complexity of $O(N)$. However, we adopted the pairwise ranking approach, where for each positive sample clicked by the user, a negative sample is sampled, reducing the complexity to $O(2n)$ (with n positive samples and corresponding n negative samples). This process is similar to the matrix factorization of BPR Loss in collaborative filtering with a negative sampling ratio equal to 1. Additionally, in the user study, we found that users could smoothly use the tool without experiencing delays, indicating good real-time performance.

We are working hard to publish our work to the extension store of the Chrome browser, where users can access commercial LLM services through a private API Key. Meanwhile, we are exploring collaborations with commercial partners, and when our work is applied to real-world applications, certain engineering designs will be considered, such as processing items offline and using different sizes of LLMs for different tasks. In the future, we believe that all our services, including LLM services, have the potential to run entirely on mobile devices. This is also why we chose to conduct offline experiments with the Phi series LLM, which has 3.8 billion parameters. From its inception, the Phi series model was designed with running on mobile devices in mind, and we believe that LLMs on user devices will become a reality in the near future.

F.0.5 Discussion on the echo chamber problem. Our work provides users with understandable and editable preference profiles, giving them greater agency over the recommended content. In this context, if systems similar to the critiquing-based recommender systems can effectively capture users' preferences in real time and then deliver diverse recommendations, it becomes possible to prevent the formation of information silos. This discussion further underscores why we believe our human-centered approach complements algorithm-centered work in the research and development of recommender systems.

F.0.6 Comparison with recommendation unlearning. Recommendation unlearning emphasizes the model's capability, with a primary focus on the model's completeness, efficiency, and performance after forgetting. These methods typically predefine which interactions should be forgotten, assess the completeness of forgetting through Membership Inference Attacks (MIA), measure the efficiency of forgetting through runtime analysis, and evaluate post-forgetting performance through recommendation accuracy. In contrast, DiscomfortFilter filters interactions from the perspective of user needs—due to its subjectivity, it is difficult to clearly define which interactions should be filtered in advance. Therefore, it is hard to make a quantitative comparison with recommendation unlearning, as current unlearning methods cannot handle subjective user inputs. We believe that these two types of work focus

Figure 8: The "Not Interested" button on the four most frequently mentioned social media platforms lacks personalization, flexibility, and transparency, resulting in barriers to user engagement.

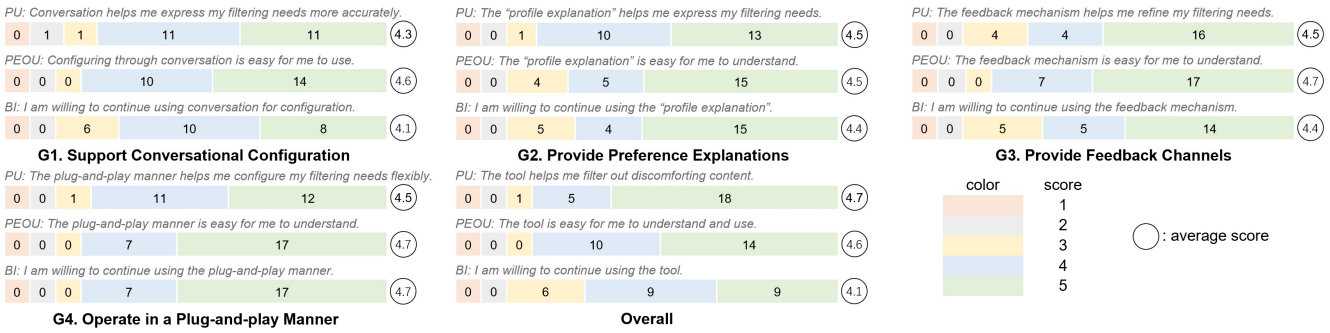

Figure 9: Score distribution from the questionnaire: the integers in the colored bars represent the number of participants for each score, with the color-to-score relationship indicated in the bottom-right corner.

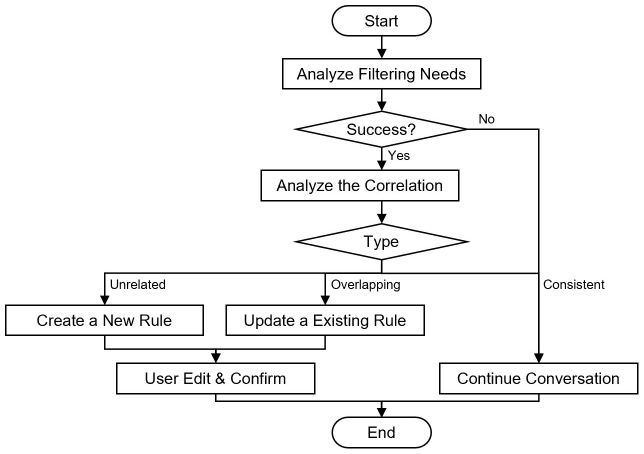

Figure 10: The workflow of the Candidate Rule Generation Module.

on model design and human-centered design, respectively, and are complementary to each other.

F.0.7 The effect of ephemeral filtering criteria. We first selected filtering rules with more than 600 items processed. For each group of 30 items, we calculated the filtering burden of the rules. The results showed that the average filtering burden of the rules decreased in oscillations, and the corresponding fitted curve yielded conclusions similar to those from the analysis at a daily granularity.

F.0.8 Only the MIND dataset was used in the offline experiment. DiscomfortFilter constructs positive and negative sample pairs by observing the content recommended by the algorithm and the user's click behavior on that content. Specifically, our negative sampling is not performed randomly from the entire item set; instead, it is selected from items that appeared alongside the clicked content but were not clicked. This approach places high demands on the dataset and is often difficult to satisfy. Among commonly used datasets, we found that only the MIND dataset provides both unclicked items that appeared alongside the clicked ones and side information for those items. Additionally, some less common datasets, such as the ZhihuRec-Dataset, provide unclicked items but lack detailed side information, offering only token vectors. Furthermore, while classic CTR datasets include labels indicating whether a user clicked on an item, they are not suitable for our purpose, as their features are anonymous and do not contain textual information.

F.0.9 Insufficient diversity of platform and user in user study. During the user study, we ceased recruiting participants when we observed that additional participants were no longer providing new insights. Due to the substantial time investment required for user research (including interviews, transcription, coding, and analysis), it became challenging to expand the study further. For instance, two related studies recruited 15 and 12 participants, respectively [8, 35].

We are currently trying to collaborate with industry partners to implement the DiscomfortFilter concept in real-world recommender systems, which will allow for broader research in this area.

F.0.10 Can DiscomfortFilter be used to capture interest? We believe the process can be understood as re-ranking algorithmically recommended content on the user side. We agree that the DiscomfortFilter could indeed be applied for such purposes and believe this might be a very promising direction for the future.

F.0.11 Is it possible to use the "Not Interested" button to enhance the feedback? While we agree that the "Not Interested" button can enhance feedback, we find it inappropriate for our goals. In the formative study, we identified drawbacks of the "Not Interested" button, such as its lack of personalization, which prevents users from providing nuanced feedback, and its lack of flexibility, which can result in content disappearing permanently even if the disinterest is temporary. Including such a button could introduce uncontrollability and potentially harm users.

F.0.12 DiscomfortFilter or DisinterestFilter? In our formative study, we observed that most users do not have strong hostility toward uninteresting content but are more averse to "discomforting" content. Thus, from users' perspective, filtering discomforting content is a more pressing issue than merely addressing disinterest, and the study focuses on the former. That said, DiscomfortFilter could also be used to filter uninteresting content—the tool's functionality depends on the user.

F.0.13 The A/B test for the control group. Our findings revealed that DiscomfortFilter's design inherently accounts for contestability, ensuring transparency and user control over its decisions. For instance, users can review filtering log and understand the rationale behind each filtering record, with the ability to make adjustments if the filtering records do not meet their expectations. Consequently, when we presented users with a mock filtering interface—where no actual or only random filtering occurred—they quickly discerned they were part of the control group, leading to immediate expressions of dissatisfaction. Moreover, during the user study in current version, participants expressed satisfaction with the performance of the functionalities, not just their design. In summary, we posit that the mere presence of a filtering interface is unlikely to enhance user satisfaction. Instead, satisfaction levels are significantly influenced by the efficacy of the actual filtering performance.

Regarding other design elements, such as explanations of preference profiles and filtering rules, these strategies emerged from our participatory design process as necessities identified by participants. Furthermore, during the evaluation phase, these features garnered high satisfaction in both questionnaires and interviews with regards to their effects and design. Therefore, we believe that our current conclusions sufficiently demonstrate the effectiveness of these designs without necessitating a control group experiment.

In conclusion, the results of the A/B test do not alter our original conclusions.