Filtering Discomforting Recommendations with Large Language Models

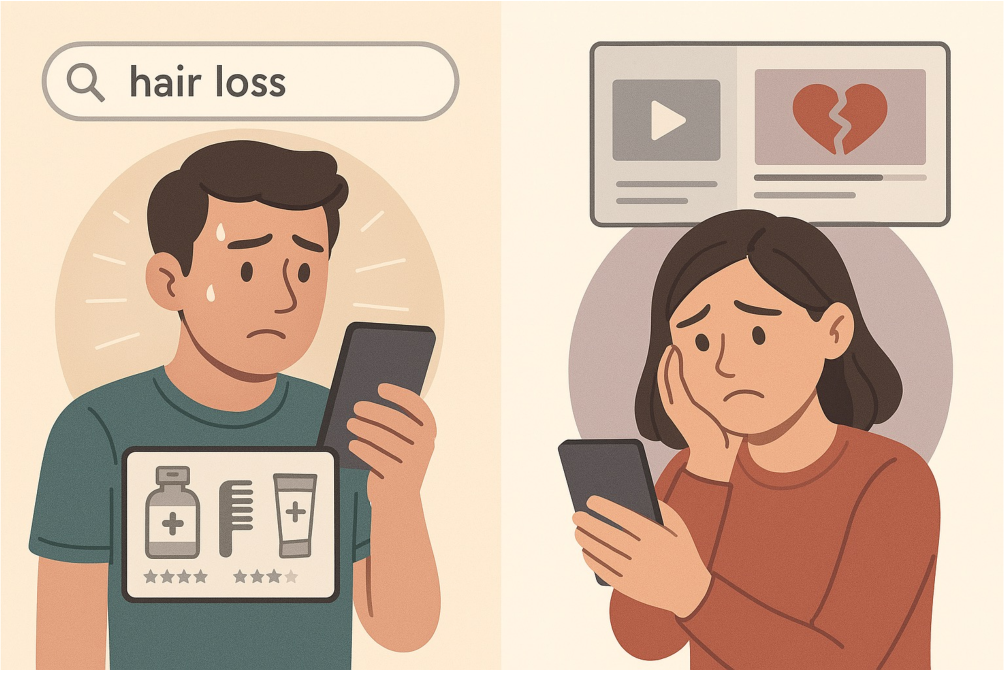
Jiahao Liu[1 5], Yiyang Shao[1], Peng Zhang[1 6], Dongsheng Li[2], Hansu Gu, Chao Chen[3], Longzhi Du[4], Tun Lu[1 6], and Ning Gu[1]
[1] Fudan University, [2] Microsoft Research Asia, [3] Shanghai Jiao Tong University, [4] Alibaba
[5] Email of the First Author: jiahaoliu23@m.fudan.edu.cn
[6] Emails of Corresponding Authors: zhangpeng_@fudan.edu.cn, lutun@fudan.edu.cn

Accepted to *TheWebConf 2025 Research Track*
Presented at *the 1st Workshop on Human-Centered Recommender Systems @ TheWebConf 2025*
Presenter: *Jiahao Liu*

2025/4

Background

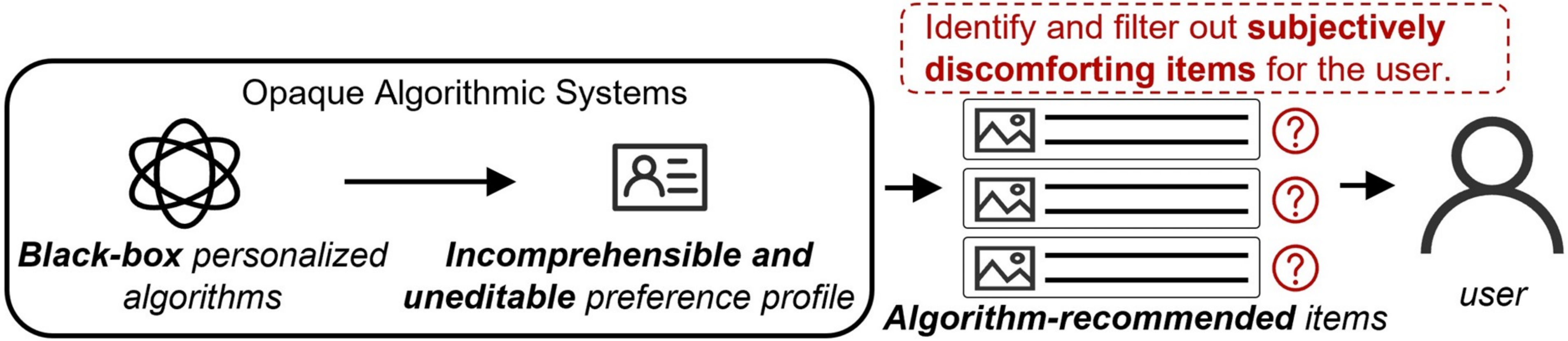
- Personalized algorithms can inadvertently expose users to **discomforting recommendations**, leading to negative emotional consequences.
  - Search for sensitive topics (e.g., hair loss) —— Privacy leakage
  - Experience emotional distress (e.g., a breakup) —— Worsen emotional state



- **Our Goal**: Help users filter out discomforting recommendations.

Problem Formulation & Challenges

- **Problem Formulation**: black-box personalized algorithms recommend items to a user based on the inferred preference profile (often implicit in the embeddings), and our objective is to identify and filter out subjectively discomforting items for the user.
- **Challenges**
  - The perception of discomfort is highly subjective, meaning that content one user finds enjoyable may be discomforting to another.
  - The preference profile is both incomprehensible and uneditable, making it challenging for the user to influence the algorithm's decisions.



Formative Study

- **Methods:** Conducted semi-structured interviews with 15 participants.
- **Finding 1**: Users may encounter discomforting recommendations for three reasons.
  - 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: “Out of curiosity, I once searched for adult products, and now they keep showing up in my recommendations—so embarrassing.”)
  - Algorithmic Modeling Bias (Personalized algorithms **cannot fully capture the nuanced interests and contexts of users**: “Getting horror content at night is awful, even if I watch it during the day.”)
  - Conflicting Interests (Platforms may **promote content designed to boost user engagement, even if it may cause discomfort**. Seven participants mentioned scenarios where this was the case.)

Formative Study

- **Finding 2**: Platforms' "Not Interested" button faces three major limitations that reduce user engagement
  - Lack of personalization: Only preset options can be selected.
  - Lack of flexibility: The duration of the block cannot be set.
  - Lack of transparency: The effect cannot be known.

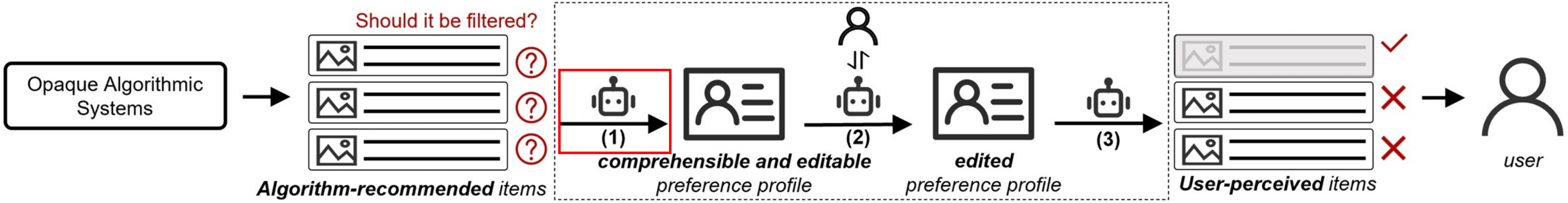
Zhihu Weibo Bilibili Xiaohongshu

Formative Study

- **Design Goals**
  - Support Conversational Configuration (Participants prefer **expressing filtering needs in natural language** as it is unrestricted and personalized. Participants also find conversation-based interaction more natural, addressing the issue of “lack of personalization”.)
  - Provide Preference Explanations (While participants struggle to articulate filtering needs proactively, all agree that **understanding platform recommendations and personal behaviors aids expression**. Reviewing and correcting recommendations enhances clarity and accuracy.)
  - Provide Feedback Channels (All participants emphasize the need for transparency and contestability. **Knowing what content is filtered and why builds trust, while enabling corrections refines filtering needs**, addressing “lack of transparency.”)
  - Operate in a Plug-and-play Manner (The tool should **work independently of specific algorithms and directly affect outputs**. Key factors: (1) Filtering needs are dynamic; (2) It should be user-managed across platforms; (3) Users prioritize mitigating discomfort over algorithm understanding. This approach enhances flexibility.)

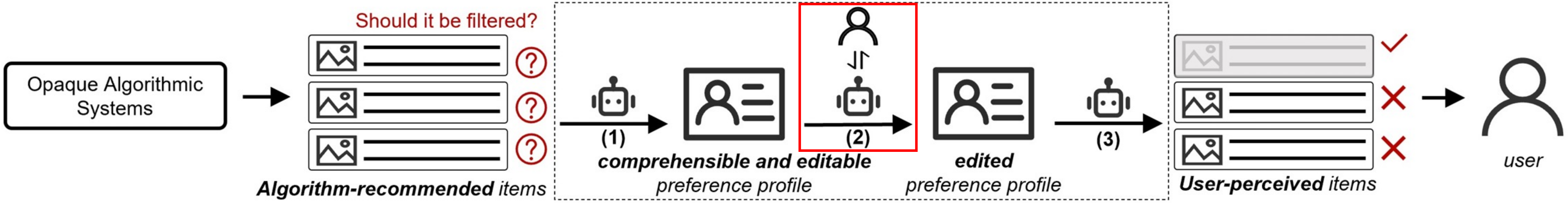
DiscomforFilter

- Empowers the user to actively influence the decisions made by personalized algorithms
  - **(1) Construct a comprehensible and editable preference profile based on user click behavior**



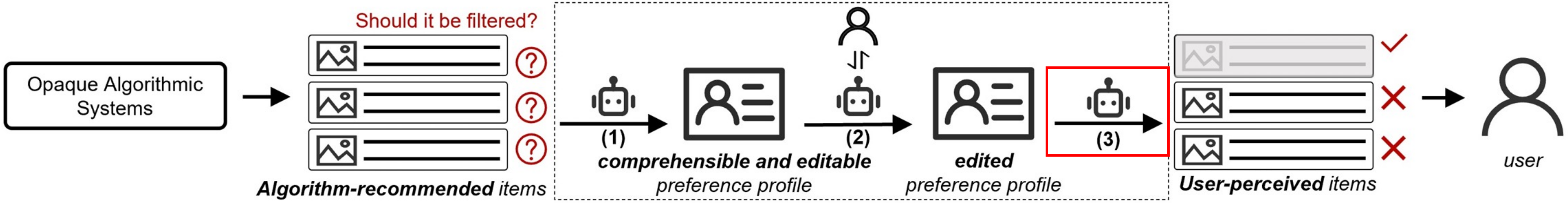
DiscomforFilter

- Empowers the user to actively influence the decisions made by personalized algorithms
  - (1) Construct a comprehensible and editable preference profile based on user click behavior
  - **(2) assists the user in expressing filtering needs, and then masks the discomforting preferences**



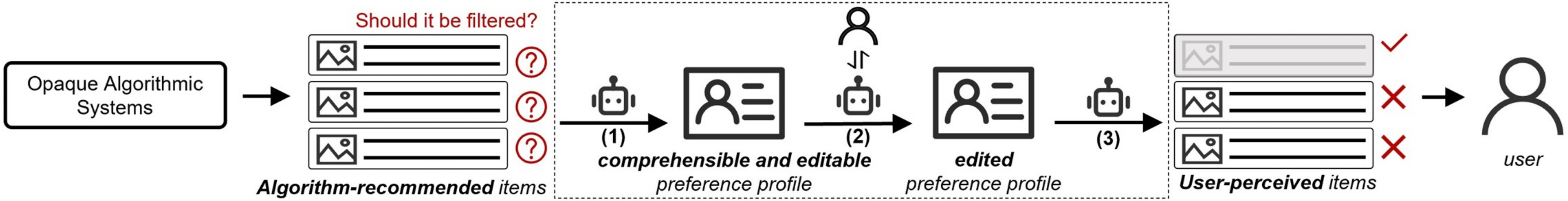
DiscomforFilter

- Empowers the user to actively influence the decisions made by personalized algorithms
  - (1) Construct a comprehensible and editable preference profile based on user click behavior
  - (2) assists the user in expressing filtering needs, and then masks the discomforting preferences
  - **(3) filters out discomforting recommendations based on the edited preference profile**



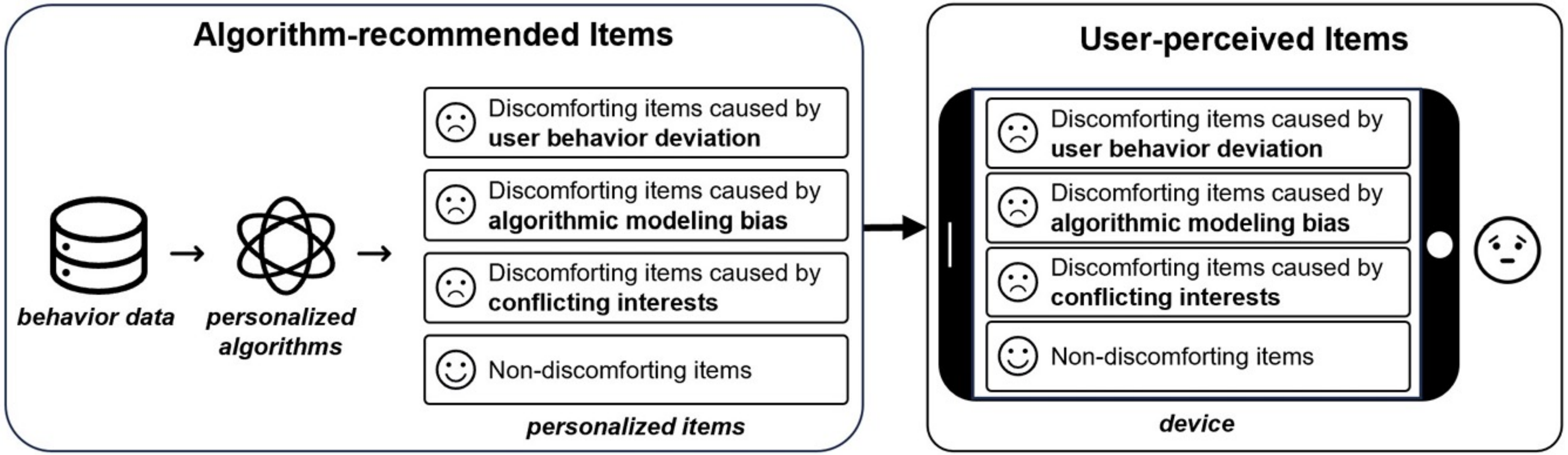
DiscomforFilter

- Empowers the user to actively influence the decisions made by personalized algorithms
  - (1) Construct a comprehensible and editable preference profile based on user click behavior
  - (2) assists the user in expressing filtering needs, and then masks the discomforting preferences
  - (3) filters out discomforting recommendations based on the edited preference profile
- **Overall, DiscomfortFilter empowers the user to actively influence the decisions made by personalized algorithms, enhancing control over the algorithms.**



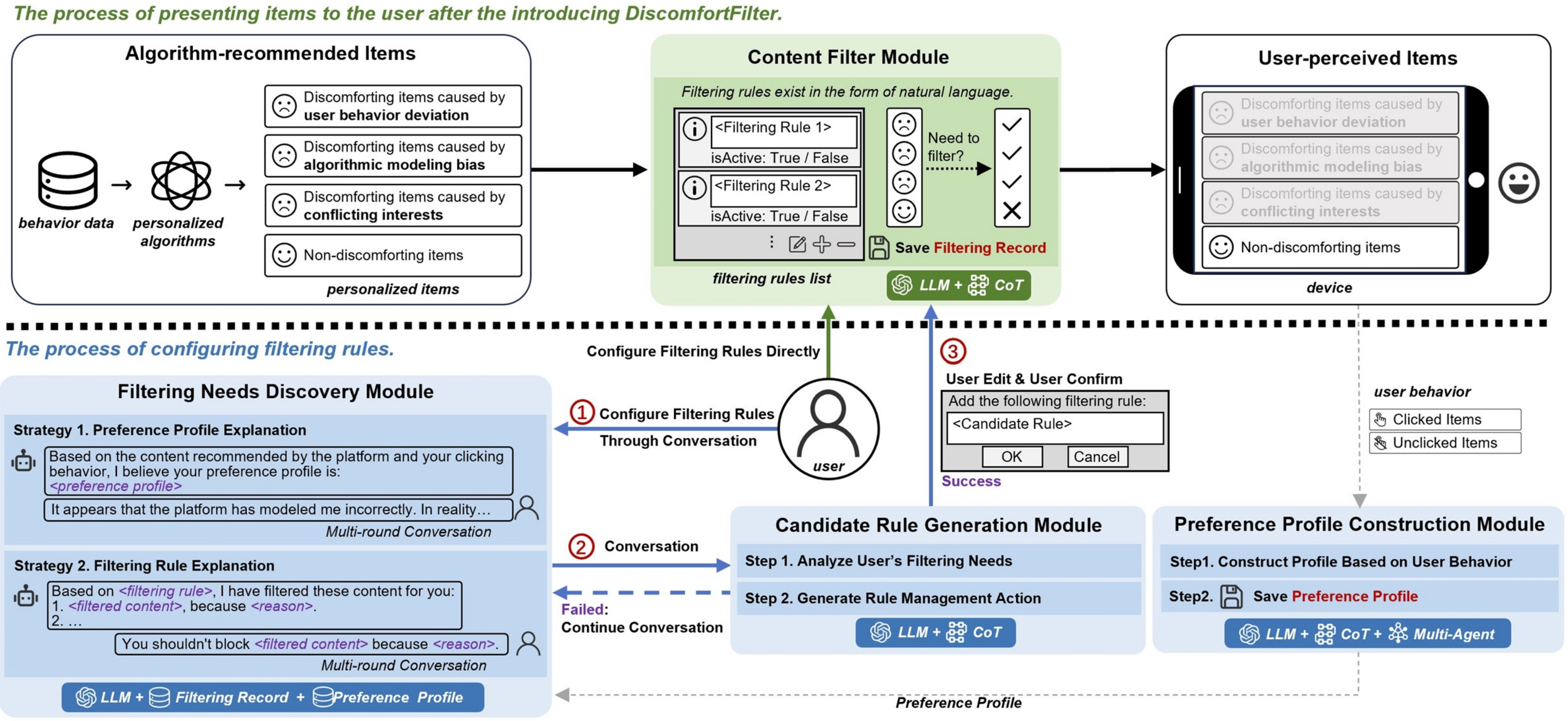
DiscomforFilter

- The process of presenting items to a user before introducing DiscomfortFilter, where items are presented directly to the user and may contain elements that are discomforting.



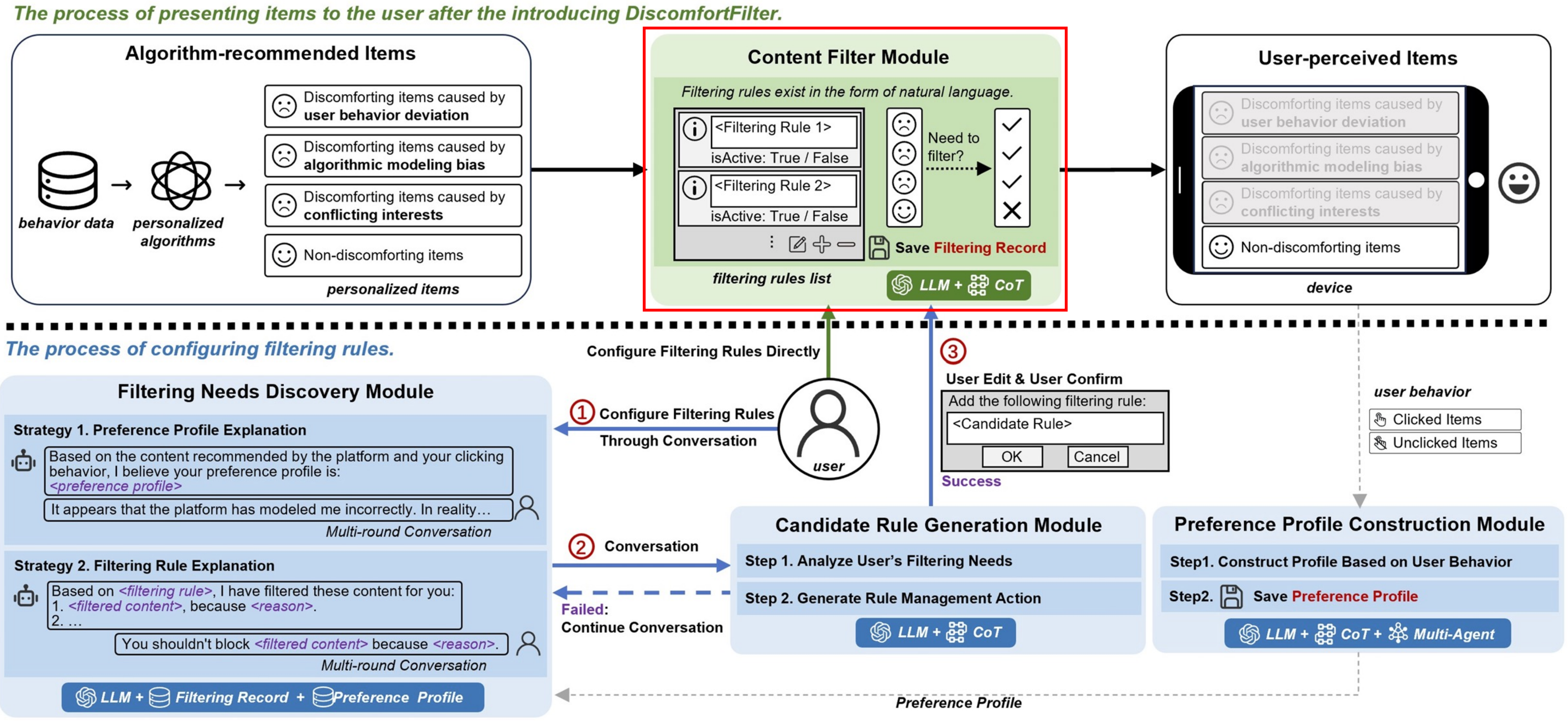
DiscomforFilter

- The process of presenting items to a user after applying the DiscomfortFilter, ensuring that only non-discomforting items are ultimately shown.



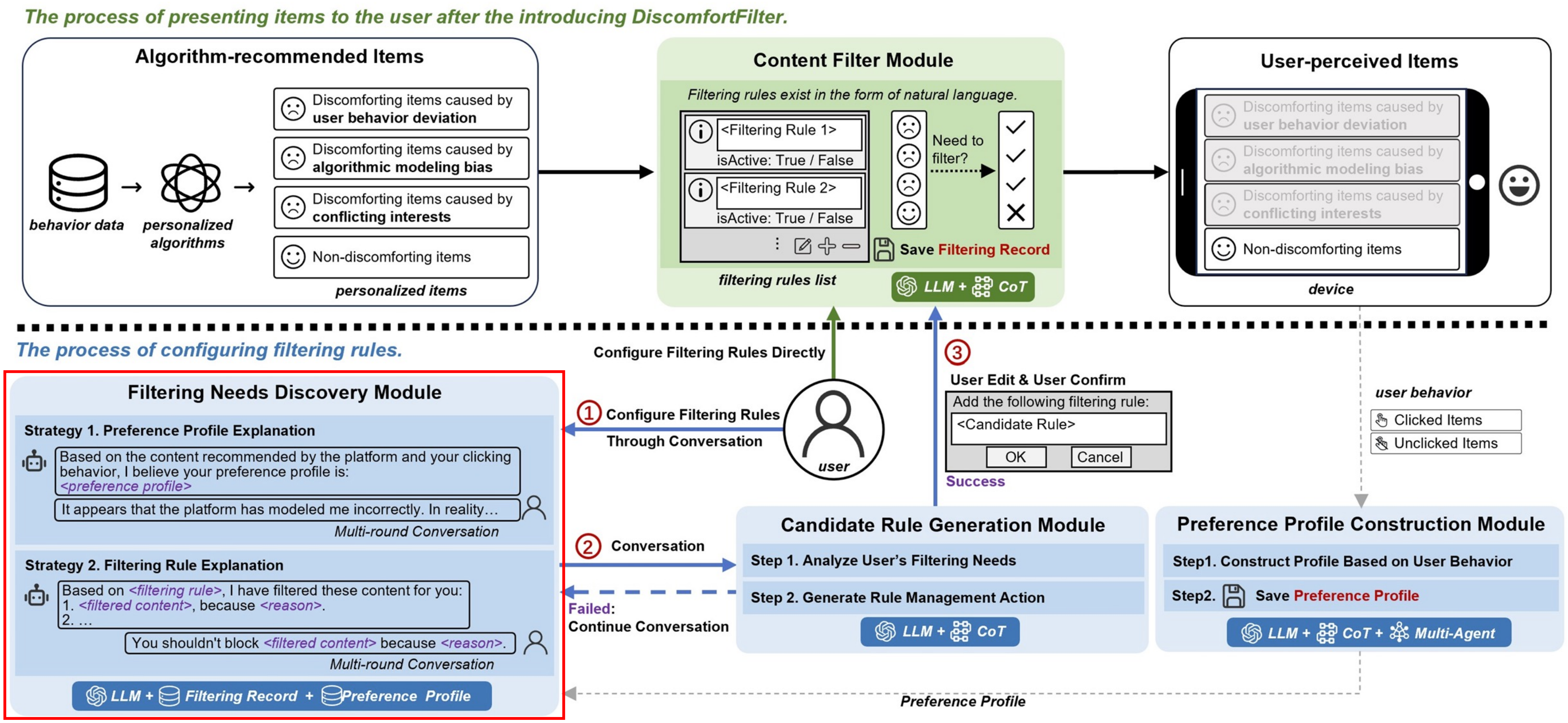
DiscomforFilter

- **Content Filter Module:** Identify and block recommended content based on the filtering rules configured by the user.



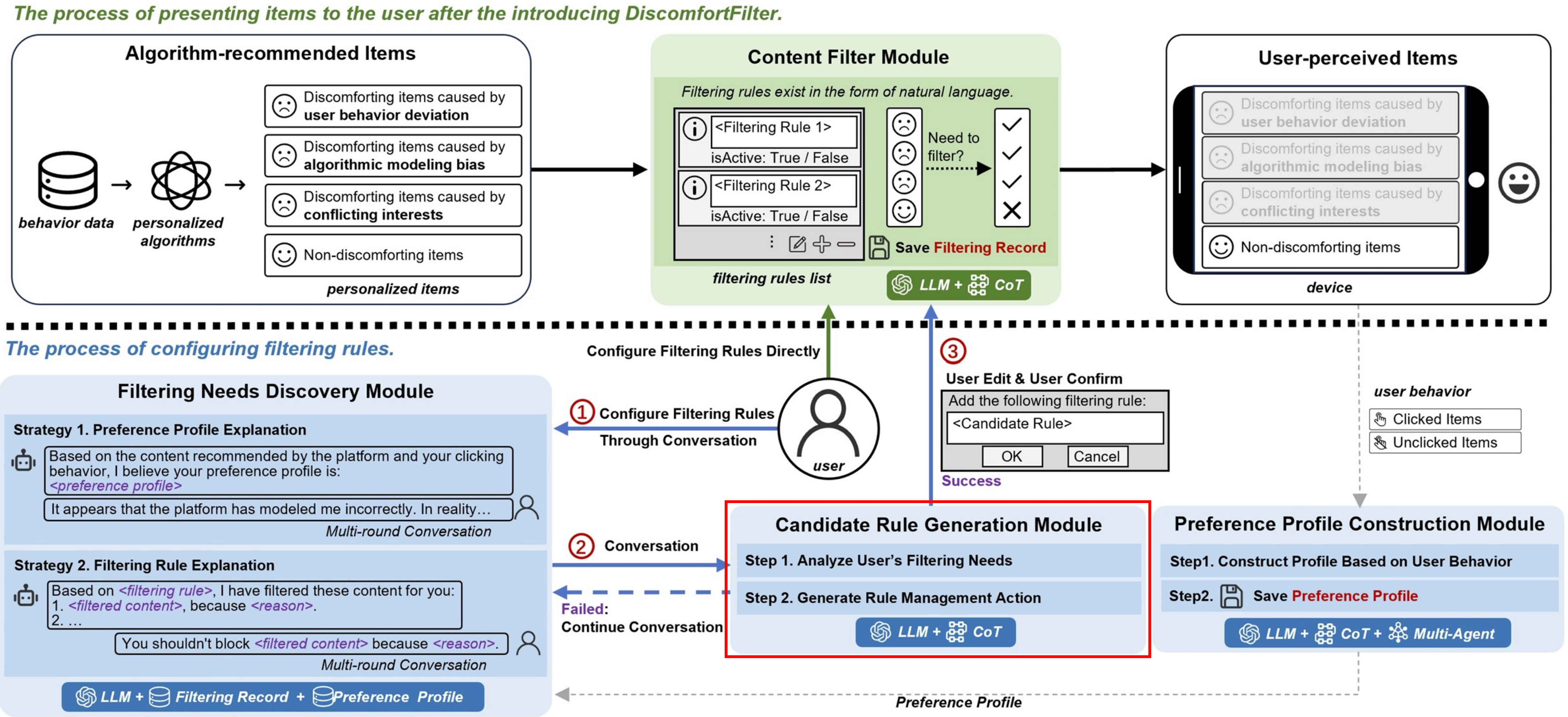
DiscomforFilter

- **Filtering Needs Discovery Module**: Help users express their filtering preferences by explaining the recommended content and the preference profile reflected by their behavior.



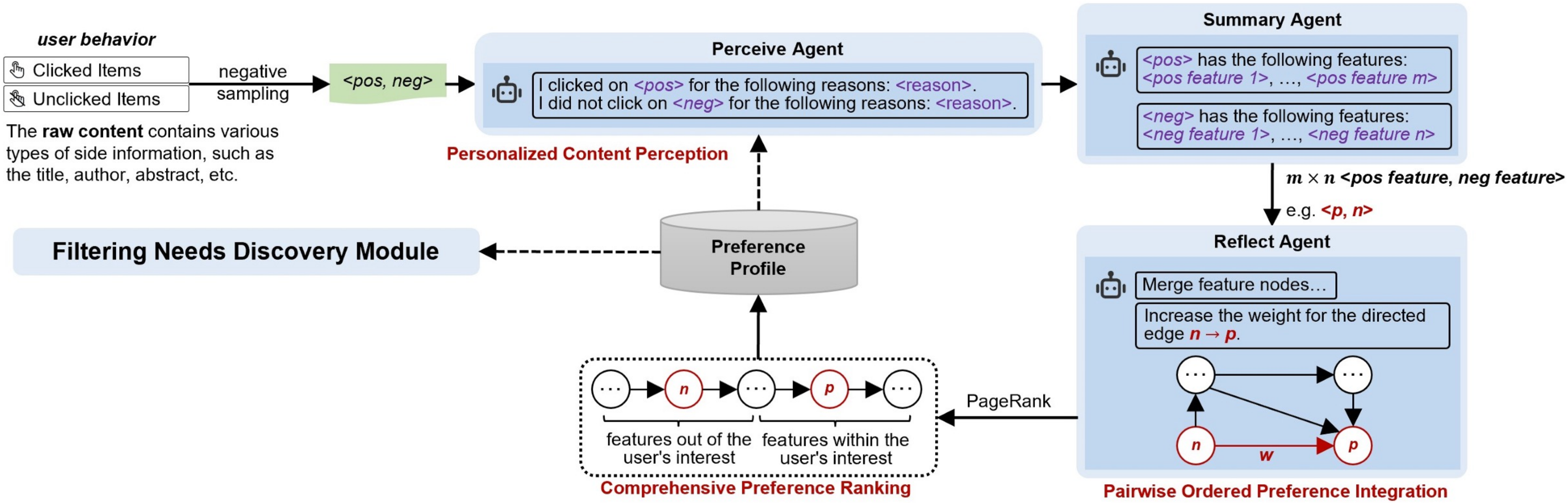
DiscomforFilter

- **Candidate Rule Generation Module**: Analyze the dialogue content to extract the user's filtering needs.



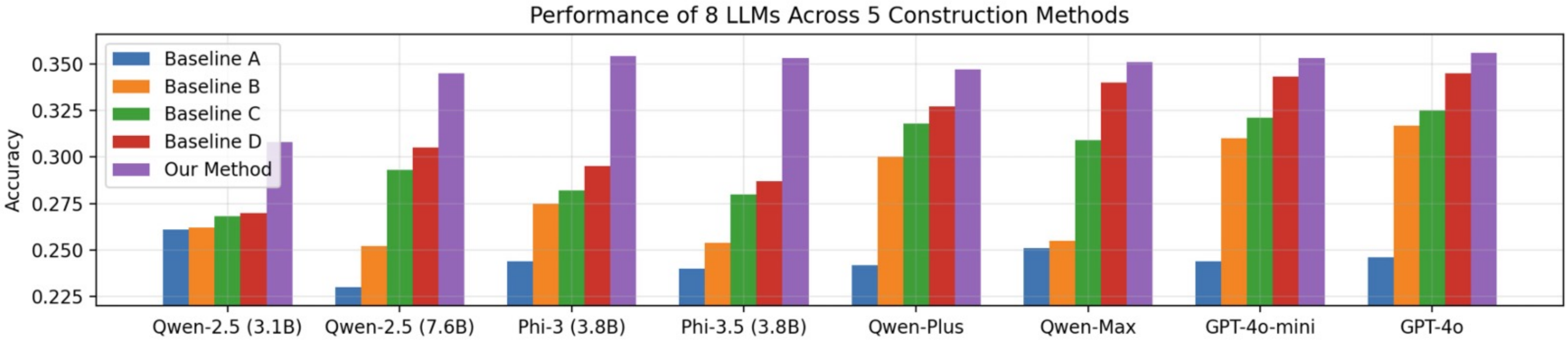
DiscomforFilter

- Preference Profile Construction Module (constructs a user's preference profile by analyzing the user's clicking behavior on recommendations in chronological order. It applies pairwise ranking to generate sample pairs from clicked items and extracts corresponding feature-level sample pairs. These pairs are then used to construct a directed graph, where features are ranked using the PageRank algorithm)

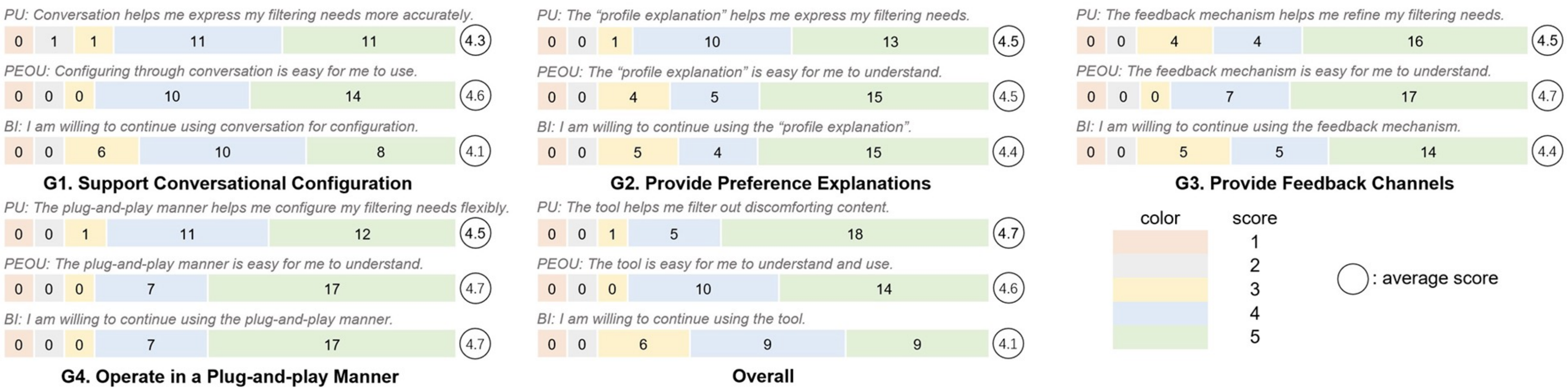


Evaluation – Offline Proxy Task

- **Task:** Predict a user's next click from K options based on their preferences.
- **Metric:** Accuracy
- **Result**: The constructed preference profile not only delivers state-of-the-art performance across various LLMs but also empowers open-source models with fewer parameters to rival proprietary commercial models in effectiveness.

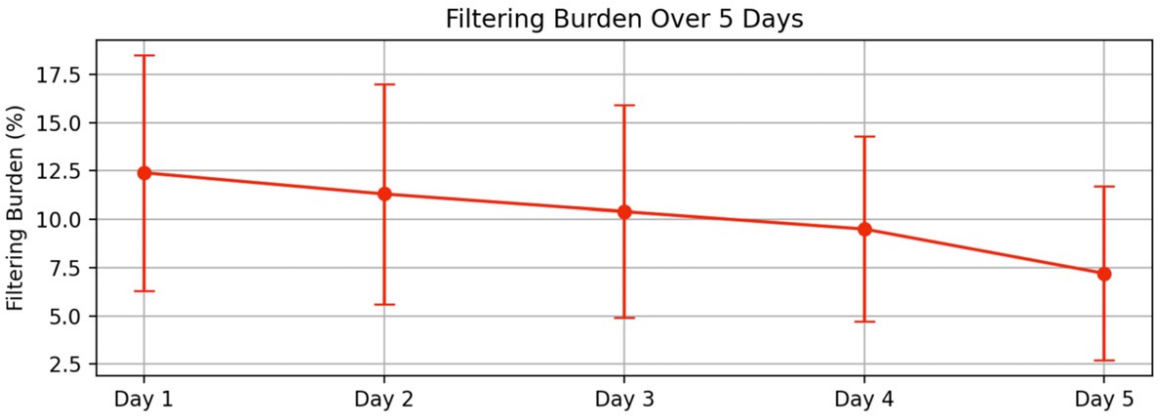


Evaluation – User Study



- Platform: **Zhihu**
- Duration: **One week**
- Number of participants: **24**
- Result: **DiscomfortFilter effectively help users express their filtering needs and filter out discomforting recommendations**

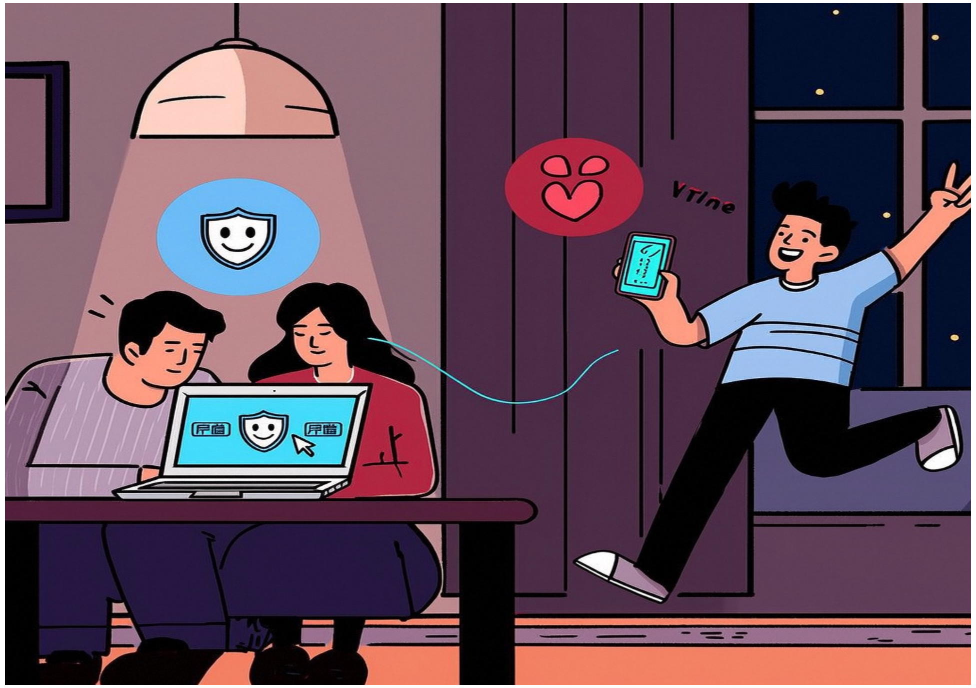
- **DiscomfortFilter impacts platform recommendation outcomes by influencing the exposure of discomforting items.**



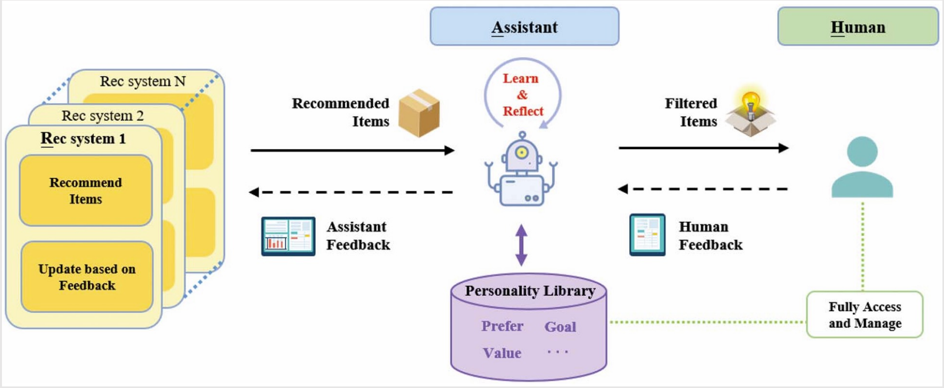
Assuming DiscomfortFilter processes *N* items using a filtering rule, with *n* items identified as discomforting, we define *n/N* as the **filtering burden** of this rule. Over time, the filtering burden steadily declined, indicating that the platform was recommending progressively fewer discomforting items.

Limitations & Future Work

- **Main Limitations**: From seeing to perceiving — still a long way to go.
  - False Association in LLMs
  - Insufficient Perceptual Alignment
- **Future Work**: Parental control over children's online content (configured by parents, used by children) — requires targeted design



Related Work from Our Group



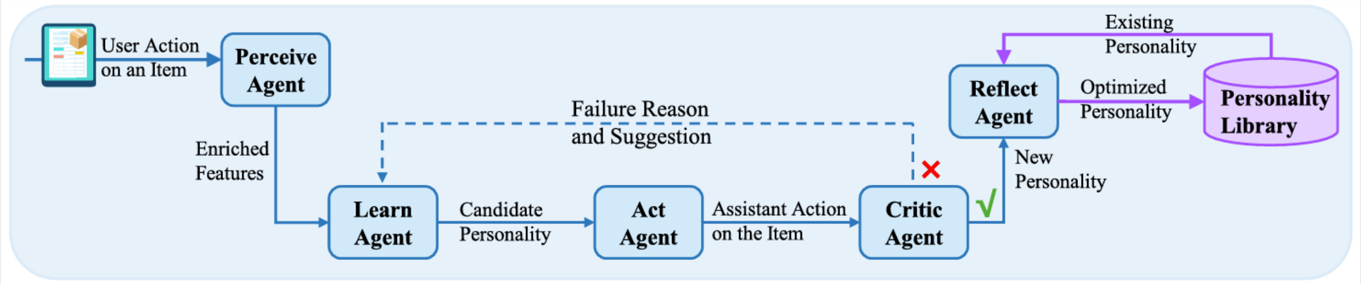
RAH! (RecSys–Assistant–Human)

A human-centered recommendation framework

Assistant Features:

1. Aligning with User Preferences
2. Filtering Items Recommended by the System
3. Simulating User Clicks

The framework utilizes the **learn-act-critic loop** and a **reflection mechanism** for improving user alignment:



Perceive Agent: Analyzes recommendations and feedback.
Learn Agent: Captures user personalities from behaviors.
Act Agent: Filters and personalizes content.
Critic Agent: Evaluates and adjusts actions.
Reflect Agent: Optimizes personality data.

[Human]
User Action: Like The Depression Cure: The 6-Step Program to Beat Depression without Drugs
[Assistant]
It can have a potential risk of privacy leakage. Suggest two personality confusion strategies.
- **Strategy I** (pretend a psychologist) Assistant will automatically express more Like on professional psychology textbooks to the recommender system.
- **Strategy II** (pretend a shared account) Assistant will automatically express random Like and Dislike.

[Human]
(select and enable a protection strategy)
[Rec System]
(recommend several items)
[Assistant]
Act
- For the user: filter recommended items from the recommender systems to remain accurate.
- For the recommender system: selectively express user real feedback and create some extra feedback to protect privacy.

1. Protecting Privacy

[Human]
User Action: Dislike the Incredibles (Pixar film)
User Comment: *Usually **watch films with my kid**. The film is **too dark for children**, yet **too childish for adults**. It's pretty much for the most part just **mindless violence** throughout the film.*
[Assistant]
Learn:
| **Prefer:** family movies | **Disprefer:** heavy dark elements, too childish, lots of violence | ……
[Rec System]
Recommend: (1) Coco (2) Ironman (3) Batman: The Dark Knight
[Assistant]
Act
(1) Like, pass to the user
(2) Not Sure, pass to the user to learn from human feedback
(3) Dislike, proxy feedback to the recommender system

2. Enhancing Control

More Capabilities:

3. Debiasing
4. Cold Start
5. ……

RAH! RecSys–Assistant–Human: A Human-Centered Recommendation Framework With LLM Agents. IEEE TCSS 2024.
Yubo Shu, Haonan Zhang, Hansu Gu, Peng Zhang, Tun Lu, Dongsheng Li, and Ning Gu.

Thanks!