# MADREC: A Multi-Aspect Driven LLM Agent for Explainable and Adaptive Recommendation

**Anonymous ACL submission**


## Abstract

Recent attempts to integrate large language models (LLMs) into recommender systems have gained momentum, but most remain limited to simple text generation or static prompt-based inference, failing to capture the complexity of user preferences and real-world interactions. This study proposes the Multi-Aspect Driven LLM Agent (MADREC), an autonomous LLM-based recommender that constructs user and item profiles by unsupervised extraction of multi-aspect information from reviews and performs direct recommendation, sequential recommendation, and explanation generation. MADREC generates structured profiles via aspect-category-based summarization and applies RE-RANKING to construct high-density inputs. When the ground-truth item is missing from the output, the SELF-FEEDBACK mechanism dynamically adjusts the inference criteria. Experiments across multiple domains show that MADREC outperforms traditional and LLM-based baselines in both precision and explainability, with human evaluation further confirming the persuasiveness of the generated explanations.

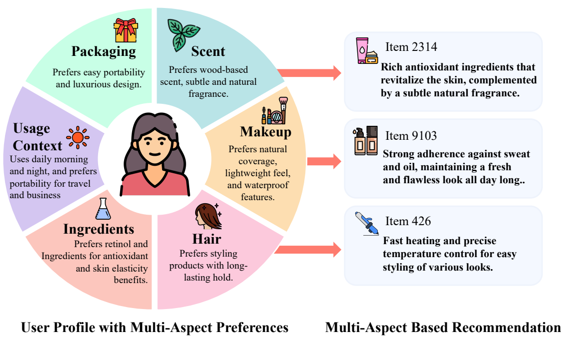




Figure 1: Multi-aspect user profiles and explainable recommendations grounded in aspect-based reasoning.

## 1 Introduction

Recommender systems have become a core technology for enhancing user experience across various online platforms, primarily by predicting items a user is likely to prefer based on their interaction history with items (Wei et al., 2019; Tsagkias et al., 2021; Singh et al., 2022; Xie et al., 2022). Recently, more sophisticated recommendation methods have emerged by incorporating various information such as metadata, domain knowledge, and user review texts (Gazdar and Hidri, 2020; Pérez-Almaguer et al., 2021). However, existing models are often specialized for specific recommendation tasks, requiring new data collection and model training for each new task, leading to an inefficient structure (Yang et al., 2023). This limitation hinders achieving generalizability and scalability required in real service environments. To address this, recent efforts have explored incorporating the strong representational power of Pretrained Language Models (PLMs) into recommender systems (Geng et al., 2023). In particular, Large Language Models (LLMs) such as `GPT-3` (Brown et al., 2020), `GPT-4` (OpenAI et al., 2024), and `LLaMA` (Touvron et al., 2023) have significantly improved the ability to understand sentence context and reason about relationships between words and concepts through large-scale text training. These capabilities are also meaningfully applicable to recommender systems.

However, most existing research utilizing LLMs has been limited to text response generation, and high-level use cases involving tool integration, external knowledge reference, and user feedback have not been sufficiently explored (Geng et al., 2023). Moreover, users expect systems that go beyond simple item recommendations to provide explainable personalized recommendations reflecting the detailed preferences of individual users, along with persuasive explanations. For instance, user reviews often contain information across various aspects such as texture, effectiveness, and usability in natural language expressions like *"It applies smoothly and has excellent pigmentation, great for* *dry skin"* (Tang et al., 2024). Such information can serve as key clues for inferring user preferences, as well as effectively conveying the reasons behind recommendations (Park and Kim, 2025). However, traditional collaborative filtering and content-based approaches struggle to structure and interpret such unstructured and multidimensional text data.

In this study, we propose MADREC (Multi-Aspect Driven LLM Agent), a framework that integrates multi-aspect-based unsupervised learning techniques with an LLM agent architecture to support a scalable, multi-domain recommendation system using LLMs (see Figure 1). First, aspect terms and categories are extracted from reviews using the Aspect Extraction Module. Then, reviews labeled with the same category are clustered, and category-specific summary sentences are generated using the Aspect Summary Module to construct user and item profiles. These user and item profiles are then re-ranked through the RE-RANKING tool, and the top-ranked candidate items are provided as input to the LLM to generate recommendation results and explanations. A SELF-FEEDBACK mechanism is applied based on recommendation results to further enhance model performance. To validate the effectiveness of the proposed framework, we conducted experiments using real review data from three domains collected from Amazon. We conducted quantitative evaluations of our framework across three key tasks—direct recommendation, sequential recommendation, and explanation generation. Additionally, we compared its performance against traditional recommendation models and recent LLM-based baselines in each task, demonstrating that our framework yields competitive results not only in terms of accuracy but also in explainability and user-personalized reasoning.

The main contributions of our work are:

- **Proposal of an unsupervised multi-aspect profile generation method**: We extract meaningful multidimensional information from unlabeled review texts and automatically generate user and item profiles, laying the foundation for explainable personalized recommendations.
- **Introduction of aspect-based RE-RANKING strategy**: We design a RE-RANKING tool that utilizes profile information generated by the ASPECT SUMMARY TOOL to evaluate the importance of candidate recommendation items and reorder them so that key items appear at the top of the LLM input.
- **Implementation of an LLM-based explainable personalized agent architecture**: We construct an active agent architecture integrating reasoning, memory, tools, SELF-FEEDBACK, and RE-RANKING, enabling flexible execution of various recommendation tasks within a single framework.

## 2 Related Work

**LLM-based Recommendation System** LLMs, leveraging their linguistic expressiveness and pre-trained knowledge, are capable of understanding user preferences at the natural language level, and research efforts have increasingly aimed to integrate them into recommender systems (Zhang et al., 2021; Cui et al., 2022; Geng et al., 2022). Early approaches proposed reformatting user–item interactions or metadata into sentence form, allowing recommendation tasks to be handled within a text-to-text paradigm (Geng et al., 2022). Subsequent methods modeled item attributes and user sequences as sentence-level inputs to Transformer-based architectures (Li et al., 2023). In studies where LLMs are used directly as recommenders, their performance has generally been found to be limited compared to traditional recommendation models (Liu et al., 2023a), prompting follow-up work on evaluating their ability to understand personalization and on applying fine-tuning strategies (Kang et al., 2023). Other efforts have explored prompt structures to enhance interactivity and explainability (Gao et al., 2023), as well as zero-shot ranking approaches (Wang and Lim, 2023). Fine-tuning large-scale models for personalized recommendation based on natural language user histories has also shown competitive performance (Yang et al., 2023).

**LLM-based Agents in Recommendation Systems** Recent research has actively explored extending LLMs into autonomous problem-solving agents. ReAct alternates between generating thoughts and external actions to establish a sophisticated problem-solving flow (Yao et al., 2023), while Toolformer proposes a structure in which the model autonomously determines when to invoke external tools (Schick et al., 2023). AutoGPT and BabyAGI aim to autonomously decompose high-level goals into sub-tasks, and LangChain

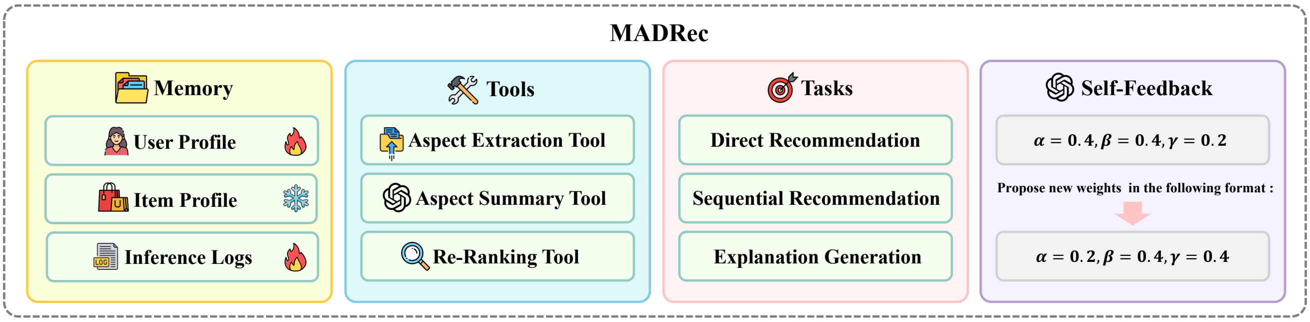



Figure 2: The structure of the MADRec framework. The system consists of Memory, Tools, Tasks, and Self-Feedback.

has been utilized as a framework for implementing agent workflows (Significant Gravitas, 2023; Nakajima, 2023; Chase, 2023). In the context of recommender systems, TallRec improves efficiency through domain-specific prompt tuning (Bao et al., 2023), while other studies have demonstrated the potential of zero-shot ranking (Hou et al., 2024) and interactive recommendation structures (Gao et al., 2023).

While prior studies have largely focused on limited functionalities or static workflows, this work proposes an active agent architecture that integrates LLM reasoning capabilities, external tool usage, and a Self-Feedback mechanism. This enables seamless execution of multi-aspect-based user preference inference, candidate Re-Ranking, and explanation generation within a unified framework.

## 3 MADRec Framework

The framework proposed in this paper, which combines multi-aspect-based unsupervised learning with an LLM-based agent architecture, is illustrated in Figure 2.

### 3.1 Memory

Memory is a core module that stores and provides multidimensional information about users and items, allowing LLMs to reference them during recommendation tasks. The user profile is dynamically generated at each recommendation point based on reviews and purchase history, using the Aspect Extraction Tool and the Aspect Summary Tool, and is subsequently updated in Memory. In contrast, item profiles are constructed in advance using the same tools and are statically stored in Memory, simulating a real-world service environment. Detailed descriptions of these tools are provided in Section 3.2. The user and item profiles stored in Memory consist of summary sentences organized by aspect category. Based on these profiles stored in Memory, the LLM evaluates candidate items and generates recommendation explanations. Furthermore, the inference results output by the LLM during the recommendation task, as well as the weight adjustments and re-recommendation history performed in the Self-Feedback phase, are also logged in Memory. This structure enables flexible adaptation to evolving user preferences, provides essential information for LLM reasoning in a structured manner, and facilitates record-based improvement strategies for enhancing future recommendation performance.

### 3.2 Tools

**Aspect Extraction Tool** This is an unsupervised module that automatically extracts key aspect categories and terms from review texts. In this study, to ensure functionality without predefined labels or domain-specific formats, we apply an unsupervised clustering model to review word embeddings to group semantically similar terms, which are then used as initial candidates for aspect categories. Subsequently, multi-head attention and max-margin loss are applied to refine contextual understanding, and finally, interpretable aspect categories are assigned to each cluster by combining domain knowledge-based rules with a `GPT`-based language model. This tool is implemented with reference to the `MUSCAD` framework proposed by Park and Kim (2025), and is designed for extensibility across various domains. For example, in the Beauty domain, words such as *evening, morning, night, daily* are grouped into the Usage Context category; *aging, elasticity, reduce, dryness* into the Improvement category; and *tropical, fruity, musk, sandalwood* into the Scent category. The extracted categories and terms serve as the foundational basis for constructing user and item profiles. Examples of the extracted aspect categories and terms are presented in Appendix C, Tables C.1, C.2, and C.3.

**ASPECT SUMMARY TOOL** This tool utilizes the aspect categories and terms extracted by the ASPECT EXTRACTION TOOL to label each review sentence with the corresponding aspect category (Table C.4 in Appendix C). It then groups sentences belonging to the same category and summarizes them using an LLM on a per-category basis (Figure D.1). The resulting summary sentences are stored in MEMORY as part of the user and item profiles. These summaries are subsequently included in the LLM input prompts and serve as key conditions for performing various recommendation tasks. For instance, a multi-aspect summary for a single product may appear in the form shown in Figure 3.

**User Profile Example**

Satisfaction: Values quality, durability, and variety in nail products. Usage Context: Prefers long-lasting products suitable for frequent nail changes. Beauty Tools: User values durability and effectiveness for nail care products. Makeup: Prefers long-lasting products with daily maintenance for durability. Quantity: Prefers sets with a mix of liked and lesser plates. Packaging: User values attractive and quality packaging for plates.

**Item Profile Example**

Satisfaction: Light, soft scent loved for daily wear, despite not being show-stopping. Usage Context: Customers appreciate the light and charming scent for daily wear, despite its subtle nature. Scent: Delicate and charming scent, not overpowering but pleasant for daily wear. Purchase: Customers repeatedly buy the fresh, dainty scent for its charm and travel-friendly packaging.

Figure 3: Examples of user and item profiles constructed with our aspect-based framework. The texts highlighted in teal indicate aspect categories, and the following sentences are the summary statements generated for each category.

**RE-RANKING Tool** This tool quantifies the relevance between users and items to select the final candidate items that will be used as input for the LLM. It computes scores for items in the initial candidate pool and selects the top-$k$ items, thereby forming an input with high information density, which plays a crucial role in improving the inference quality of the LLM. This design reflects the finding that not only the inclusion but also the position of information within the input can significantly affect the accuracy of LLM outputs when processing long contexts (Liu et al., 2023b). Accordingly, the tool places high-scoring core items at the beginning of the LLM input to support more precise reasoning within the model's limited context window. The final score $S_i$ for a candidate item $i$ is defined as follows:

$$S_i = \alpha{\cdot}\mathrm{Sim}(u,i){+}\beta{\cdot}\mathrm{Sim}(C(u),C(i)){+}\gamma{\cdot}\mathrm{Pop}(i)$$

Here, $\mathrm{Sim}(u,i)$ denotes the cosine similarity between the user profile and the item profile, $\mathrm{Sim}(C(u),C(i))$ represents the similarity between the set of aspect categories associated with the user's past purchases $C(u)$ and those of the candidate item $C(i)$, and $\mathrm{Pop}(i)$ is a relative popularity indicator calculated based on the number of reviews for item $i$.

In this study, we leverage multi-aspect-based user and item profiles—summarized at the aspect category level—for the RE-RANKING computation, enabling a finer-grained reflection of user preferences compared to simple keyword matching or frequency-based ranking. In other words, the multidimensional characteristics extracted from reviews are actively incorporated into the scoring process, effectively capturing subtle differences in individual user preferences.

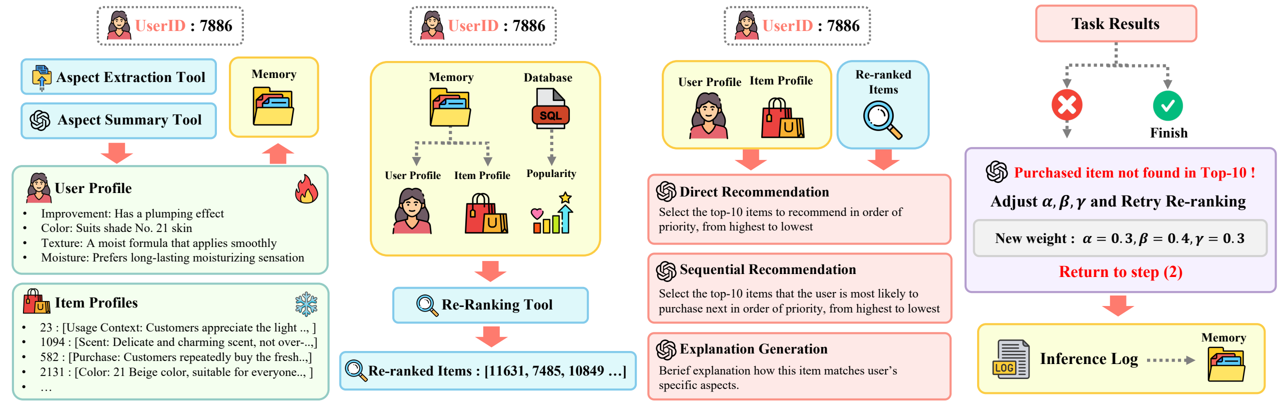
### 3.3 Tasks

The MADREC framework performs three main recommendation tasks centered around the LLM: direct recommendation, sequential recommendation, and explanation generation.

**Direct Recommendation** This task directly recommends the most suitable items at the current point in time based on the user profile. The prompt includes the user profile and a refined list of candidate items, and the LLM selects the recommended items in order of priority and responds in natural language.

**Sequential Recommendation** This task predicts the items that the user is most likely to prefer next, based on their sequential purchase history. The input prompt contains the most recent five past items sorted in chronological order, the user profile, and the refined candidate item list. Based on this information, the LLM generates top recommended items.

**Explanation Generation** For each recommended item, the LLM generates a natural language sentence explaining why the item is suitable for the user, organized by aspect category. The LLM receives the user profile and the multi-aspect summary profile of each recommended item as input.




(1) Generating a Multi-Aspect User Profile (2) Multi Factor Re-Ranking (3) Tasks (4) Self-Feedback

Figure 4: Overall pipeline of the proposed MADREC framework. The system proceeds in four stages: (1) Generating a Multi-Aspect User Profile, (2) Multi Factor Re-Ranking, (3) Tasks, and (4) Self-Feedback.

Each task is formulated as a prompt that includes input information such as the user and item profiles, candidate item list, and past interactions, which is then fed into the LLM. The LLM performs reasoning in a step-by-step `Chain-of-Thought (CoT)` manner and generates responses in natural language. The prompts used for all recommendation tasks are illustrated in Figure D.2 in Appendix D.

### 3.4 SELF-FEEDBACK

If the user's actual purchase item is not included in the recommendation results, the SELF-FEEDBACK mechanism is activated. This simulates user behaviors such as re-searching or adjusting filters to find the desired product. Specifically, when the correct item is not included in the recommendation output, the weight coefficients $\alpha$, $\beta$, and $\gamma$ in the SELF-FEEDBACK formula are adjusted to dynamically revise the recommendation criteria, and the LLM is prompted to re-rank and recommend based on a new set of candidates. This structure enables the LLM to reflect on and refine its initial reasoning, and each step is logged and stored in MEMORY, where it can be used for future recommendations and agent decision-making. The SELF-FEEDBACK prompt is shown in Figure D.3 in Appendix D.

## 4 MADREC-Based Recommendation Pipeline

Based on the components described in Section 3, the MADREC-based recommendation pipeline consists of the following four steps, as illustrated in Figure 4:

**Step 1. Generating a Multi-Aspect User Profile**: As shown in Figure 4 (a), when a recommendation request is received, the user's reviews are processed through the ASPECT EXTRACTION TOOL and ASPECT SUMMARY TOOL to dynamically generate a user profile organized by aspect categories. Item profiles are pre-generated in the same way and stored in MEMORY.

**Step 2. Multi-Factor RE-RANKING**: The RE-RANKING TOOL computes a score for each item by combining the similarity between user and item profiles, category overlap, and popularity, and selects the top 30 candidate items (See Figure 4 (b)).

**Step 3. Tasks**: As illustrated in Figure 4 (c), the selected candidate items are used as input to the LLM, and three tasks are performed: direct recommendation, sequential recommendation, and explanation generation (see Section 3.3).

**Step 4. SELF-FEEDBACK**: If the actual purchased item is not included in the recommendation results, the SELF-FEEDBACK module is triggered, as shown in Figure 4 (d), to adjust the RE-RANKING weights and repeat the recommendation task.

## 5 Experimental Setup

### 5.1 Datasets

This study conducts evaluations using three real-world datasets with varying domains and levels of data sparsity. The data were collected from Amazon.com [1], containing user reviews and ratings across a wide range of product categories. Among them, three categories—Beauty, Sports, and Toys—were selected for the experiments. After preprocessing, the statistics of each dataset are summarized in Table 1.

[1] https://nijianmo.github.io/amazon/

| Statistics | Beauty | Sports | Toys |
|---|---|---|---|
| # Users | 22,363 | 25,598 | 19,412 |
| # Items | 12,101 | 18,357 | 11,924 |
| # Actions/User | 8.9 | 8.3 | 8.1 |
| # Actions/Item | 16.4 | 16.1 | 14.1 |
| # Actions | 198,502 | 296,337 | 167,597 |
| Sparsity | 99.93% | 99.95% | 99.93% |

Table 1: Statistics of the datasets after preprocessing. #Actions/User and #Actions/Item denote the average number of interactions per user and item, respectively. Sparsity indicates the proportion of missing entries in the user-item matrix

### 5.2 Evaluation Metrics

To quantitatively evaluate the performance of the proposed system, this study adopts a leave-one-out strategy, where one item is repeatedly excluded from each user's interaction sequence and set as the prediction target. This approach assesses how accurately the model can predict the excluded item. For the evaluation of direct and sequential recommendation tasks, we use HR@n (Hit Ratio) and NDCG@n (Normalized Discounted Cumulative Gain) as performance metrics, with $n$ set to 5 and 10 to account for both the hit rate and the ranking of top recommendations. For the explanation generation task, we employ n-gram-based automatic evaluation metrics such as BLEU-n and ROUGE-n to assess the quality of the generated natural language explanations. Additionally, we use the pretrained language model-based BERT-Score to provide a more fine-grained assessment of semantic similarity.

### 5.3 Baselines

To compare the performance of the proposed model, we follow the experimental settings of Geng et al. (2022); Zhou et al. (2020); Liu et al. (2023a) and select the following representative baseline models.

For the direct recommendation task, we use `ENMF` (Chen et al., 2019), `SimpleX` (Mao et al., 2021), `P5` (Geng et al., 2022), and `ChatGPT` (Liu et al., 2023a) as baselines. For the sequential recommendation task, we include `P5`, `ChatGPT`, `S`$^3$`-Rec` (Zhou et al., 2020), and `SAS-Rec` (Kang and McAuley, 2018). For the explanation generation task, we compare with `P5` and `ChatGPT`.

Our framework uses `GPT-4.1-nano` (Schulman et al., 2022) as the core language model, and to efficiently reference domain-specific information, the entire review dataset is stored in a MySQL database. This database consists of tables that include product metadata, user interaction histories, and profile information pre-generated by the tools. Detailed descriptions of each baseline model can be found in Appendix A.

### 5.4 Training Details

In the RE-RANKING stage for candidate item selection, scores are computed using weights of $\alpha = 0.4$, $\beta = 0.4$, and $\gamma = 0.2$, and the top 30 items are extracted and fed into the LLM prompt.

## 6 Experimental Results

### 6.1 Results on Recommendation Tasks

The proposed framework was evaluated across three key recommendation tasks—direct recommendation, sequential recommendation, and explanation generation. The direct recommendation task involves predicting the Top-N items, including the ground-truth, from a pool of 100 candidates. The sequential recommendation task aims to predict the next likely item based on the user's purchase history. As shown in Table 2 and Table 3, our proposed system (RR+SF) consistently outperformed all baseline models across all domains. This demonstrates that, unlike conventional models limited to static inference or pretraining-based reasoning, our framework benefits from an active processing structure that combines RE-RANKING and SELF-FEEDBACK, resulting in more robust and adaptive performance.

The explanation generation task was introduced to go beyond item recommendation and provide users with clear, natural language explanations for the recommendations. Specifically, the LLM generates explanations based on the relationship between the user and item profiles, focusing on relevant aspect categories. Examples of generated explanations are shown in Figure B. Since this task is conditioned on the final recommendation result and the aspect profile of each item, RE-RANKING and SELF-FEEDBACK influence the outcome only indirectly. Thus, we compare the generation quality of RR+SF against existing LLM-based baselines. As shown in Table 4, the proposed model achieved the highest performance across all domains.

### 6.2 Ablation Study on RE-RANKING and SELF-FEEDBACK Modules

To quantitatively analyze the effectiveness of the two core components of our proposed system—RE-

| Methods | Beauty | | | | Sports | | | | Toys | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | HR@5 | NDCG@5 | HR@10 | NDCG@10 | HR@5 | NDCG@5 | HR@10 | NDCG@10 | HR@5 | NDCG@5 | HR@10 | NDCG@10 |
| ENMF | 0.020 | 0.016 | 0.050 | 0.025 | 0.096 | 0.062 | 0.144 | 0.078 | 0.066 | 0.042 | 0.128 | 0.062 |
| P5 | 0.090 | 0.053 | 0.166 | 0.079 | 0.100 | 0.066 | 0.170 | 0.079 | 0.110 | 0.071 | 0.174 | 0.092 |
| SimpleX | 0.040 | 0.017 | 0.082 | 0.026 | 0.034 | 0.013 | 0.054 | 0.018 | 0.050 | 0.029 | 0.086 | 0.036 |
| ChatGPT | 0.044 | 0.029 | 0.078 | 0.040 | 0.043 | 0.082 | 0.022 | 0.035 | 0.045 | 0.025 | 0.076 | 0.035 |
| RR + SF (ours) | **0.252** | **0.152** | **0.364** | **0.188** | **0.188** | **0.117** | **0.310** | **0.156** | **0.200** | **0.131** | **0.334** | **0.174** |
| RR + No-SF | 0.218 | 0.133 | 0.296 | 0.158 | 0.162 | 0.103 | 0.264 | 0.132 | 0.174 | 0.114 | 0.260 | 0.142 |
| No-RR + SF | 0.132 | 0.090 | 0.246 | 0.126 | 0.150 | 0.098 | 0.258 | 0.132 | 0.106 | 0.070 | 0.214 | 0.104 |
| No-RR + No-SF | 0.110 | 0.074 | 0.186 | 0.099 | 0.108 | 0.072 | 0.180 | 0.095 | 0.100 | 0.066 | 0.152 | 0.083 |

Table 2: Performance comparison direct recommendation on Beauty, Sports, and Toys domains. Bold indicates the best score, underline the second-best. RR and SF denote Re-Ranking and Self-Feedback.

| Methods | Beauty | | | | Sports | | | | Toys | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | HR@5 | NDCG@5 | HR@10 | NDCG@10 | HR@5 | NDCG@5 | HR@10 | NDCG@10 | HR@5 | NDCG@5 | HR@10 | NDCG@10 |
| P5 | 0.046 | 0.029 | 0.048 | 0.030 | 0.072 | 0.042 | 0.116 | 0.056 | 0.066 | 0.041 | 0.110 | 0.055 |
| $S^3$-Rec | 0.056 | 0.034 | 0.106 | 0.049 | 0.046 | 0.025 | 0.104 | 0.043 | 0.046 | 0.027 | 0.088 | 0.040 |
| SAS-Rec | 0.070 | 0.048 | 0.135 | 0.069 | 0.103 | 0.058 | 0.169 | 0.099 | 0.090 | 0.054 | 0.128 | 0.081 |
| ChatGPT | 0.018 | 0.012 | 0.046 | 0.023 | 0.022 | 0.019 | 0.032 | 0.026 | 0.029 | 0.014 | 0.038 | 0.018 |
| RR + SF (ours) | **0.234** | **0.155** | **0.362** | **0.196** | **0.230** | **0.142** | **0.368** | **0.186** | **0.202** | **0.136** | **0.336** | **0.178** |
| RR + No-SF | 0.206 | 0.142 | 0.312 | 0.177 | 0.180 | 0.115 | 0.268 | 0.142 | 0.178 | 0.120 | 0.278 | 0.152 |
| No-RR + SF | 0.136 | 0.086 | 0.246 | 0.121 | 0.118 | 0.073 | 0.206 | 0.101 | 0.128 | 0.089 | 0.200 | 0.112 |
| No-RR + No-SF | 0.104 | 0.068 | 0.188 | 0.095 | 0.104 | 0.072 | 0.140 | 0.083 | 0.104 | 0.069 | 0.144 | 0.082 |

Table 3: Performance comparison sequential recommendation evaluation on Beauty, Sports, and Toys domains.

| Methods | Beauty | | | | | Sports | | | | | Toys | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BLEU2 | R-1 | R-2 | R-L | BERTS | BLEU2 | R-1 | R-2 | R-L | BERTS | BLEU2 | R-1 | R-2 | R-L | BERTS |
| RR + SF (ours) | 0.473 | **15.632** | **6.298** | **12.689** | **84.831** | **0.103** | **14.165** | **3.437** | **10.355** | **85.004** | **0.277** | **15.558** | **4.412** | **10.765** | **85.160** |
| ChatGPT | **1.160** | 14.981 | 3.041 | 10.874 | 82.642 | 0.023 | 8.162 | 1.196 | 6.504 | 83.410 | 0.085 | 9.735 | 1.433 | 7.342 | 83.673 |
| P5 | 0.006 | 2.162 | 0.120 | 2.070 | 8.535 | 0.001 | 2.577 | 0.113 | 2.296 | 9.984 | 0.001 | 2.407 | 0.113 | 2.176 | 8.596 |

Table 4: Performance comparison for explanation generation across three domains. BLEU2: bi-gram precision; R-1/R-2/R-L: ROUGE scores for unigram, bigram, and longest sequence matches; BERTScore: semantic similarity.

Ranking and Self-Feedback—we conducted experiments on the following four combinations. All experiments were performed under the same dataset, prompt structure, and LLM architecture. Detailed descriptions of the prompts used in each setting are provided in Appendix D. The results are summarized in Table 2, Table 3, and Table 4. In the No-RR+SF setting, Re-Ranking is omitted and recommendations are generated in the original candidate order, followed by the application of Self-Feedback. In RR+No-SF, only Re-Ranking is applied without any feedback on the recommendation outcome. The No-RR+No-SF setting disables both modules and represents the most basic recommendation structure that directly infers over unranked candidates. Across all domains and tasks, the RR+SF configuration—where both Re-Ranking and Self-Feedback are applied—achieved the best performance. In the direct recommendation task, RR+SF showed relative improvements over No-RR+No-SF of 95.7% in Beauty, 72.2% in Sports, and 119.7% in Toys. In the sequential recommendation task, the improvements were 92.6%, 162.9%, and 133.3%, respectively.

To visualize the individual and combined effects of Re-Ranking and Self-Feedback, Figure 5 presents HR@10 scores from two perspectives. The figure compares the performance of all four configurations and `ChatGPT` across the Beauty, Sports, and Toys domains, clearly showing that RR+SF (ours) consistently outperforms all other baselines. A notable observation is that the simple prompt-based LLM approach (`ChatGPT`) yields the lowest performance in all domains, demonstrating the superiority of leveraging aspect-based user and item profiles. In particular, the direct recommendation task in the Sports domain reveals an approximately 8× performance gap between `ChatGPT` (0.022) and No-RR+No-SF (0.180), highlighting the especially pronounced shortcomings of prompt-only methods in this task.

The RR+No-SF and No-RR+SF configurations allow for a clear analysis of the individual contributions of each module. RR+No-SF achieved substantial improvements over No-RR+No-SF across all domains, indicating that the Re-Ranking module plays a more significant role in overall performance. Specifically, Re-Ranking sorts the candidate items based on multi-aspect profile similarity, category overlap, and popularity, selecting the top 30 most informative items as input to the LLM.

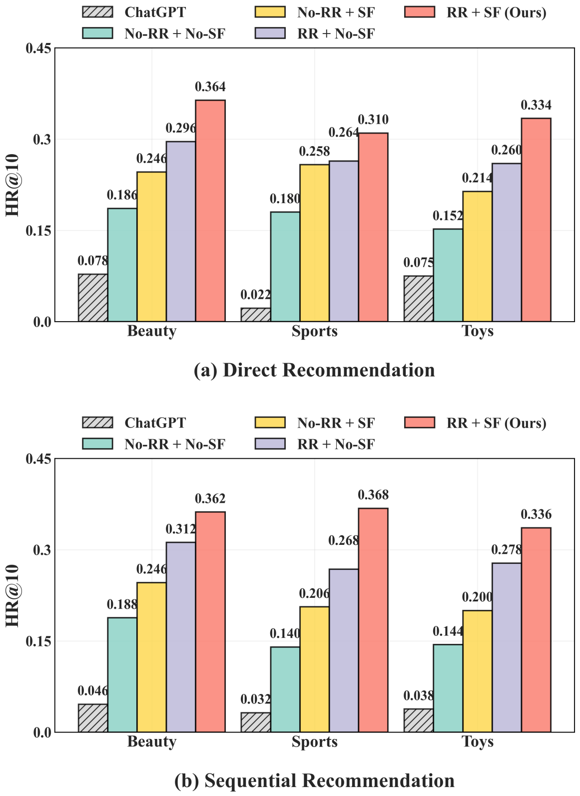



Figure 5: Performance comparison (HR@10) across Beauty, Sports, and Toys domains for four model variants and `ChatGPT`. RR + SF (ours) consistently outperforms all baselines, while `ChatGPT` exhibits limited effectiveness, particularly in sequential recommendation.

This enables the model to perform more accurate reasoning within the limited context window. Similarly, No-RR+SF also outperformed No-RR+No-SF in all domains, demonstrating the effectiveness of the SELF-FEEDBACK module. When recommendations are suboptimal, SELF-FEEDBACK automatically adjusts the scoring criteria and re-invokes inference, mimicking real user behaviors such as re-searching or re-filtering, and enabling *iterative refinement*. Finally, RR+SF achieved the largest performance gains compared to No-RR+No-SF, empirically demonstrating that the two modules work synergistically, producing a greater effect than their individual contributions alone. These results confirm that using both modules together yields the strongest performance and highlight a key structural advantage over conventional systems that rely solely on static inference.

### 6.3 Human Evaluation

Since the linguistic quality and persuasiveness of recommendation explanations are difficult to fully evaluate using automatic metrics alone, we additionally conducted a human evaluation. Specifically, three independent evaluators (Evaluator 1, 2, and 3) were asked to compare the explanations generated by `P5`, `ChatGPT`, and our proposed model (RR+SF) across 50 test cases. Each evaluator ranked the three explanations for each case, and Table 5 reports the percentage of times each method was selected as the top-1 explanation by each evaluator. The results show that the proposed model was consistently rated highest by all evaluators. This indicates that our model is able to generate more specific and persuasive explanations by grounding its reasoning in aspect-level user preferences.

| Methods | Evaluator | | | Average |
|---|---|---|---|---|
| | Eva_1 | Eva_2 | Eva_3 | |
| P5 | 0.34 | 0.34 | 0.36 | 0.35 |
| ChatGPT | 0.00 | 0.00 | 0.00 | 0.00 |
| RR + SF (Ours) | 0.66 | 0.66 | 0.64 | **0.65** |

Table 5: Human evaluation results of explanation quality, rated by three independent evaluators. RR + SF (Ours) significantly outperforms P5 and ChatGPT in terms of average human preference.

## 7 Conclusion

In this study, we propose MADREC, a Multi-Aspect Driven LLM Agent for explainable and personalized recommendation. The framework extracts multidimensional aspect information from user reviews in an unsupervised manner and generates structured user and item profiles that reflect diverse preference dimensions. By combining unsupervised multi-aspect learning with an LLM-based agent architecture, MADREC identifies aspect terms and categories, summarizes category-specific content, and constructs interpretable profiles. These profiles are refined using a RE-RANKING TOOL and provided as input to the LLM, while the SELF-FEEDBACK module dynamically adjusts recommendation criteria based on previous outputs, enabling iterative improvement. Evaluations on three recommendation tasks show that MADREC consistently outperforms traditional and LLM-based baselines, not only in accuracy but also in explainability. Human evaluation further confirms that our model delivers the most persuasive explanations. In future work, we plan to improve the adaptability and interactivity of the system by incorporating user feedback-driven learning and integrating external tools.

## 8 Limitations

This study proposes an LLM-based active recommendation framework and demonstrates meaningful performance improvements across various recommendation tasks. Nevertheless, several limitations remain. First, the multi-stage inference pipeline introduced by RE-RANKING and SELF-FEEDBACK may increase computational cost and response time, requiring further optimization for real-time applications. Second, aspect-based inputs can be constrained by context length limits, necessitating input compression or selection strategies. Third, while SELF-FEEDBACK enables iterative recommendation, it currently relies on static criteria rather than real user responses, indicating a need for future integration with interaction logs and user behavior signals.

## 9 Ethics Statement

The training process of our proposed architecture does not involve any socially sensitive or ethically inappropriate elements. Accordingly, this study raises no ethical concerns.

## References


Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023. Tallrec: An effective and efficient tuning framework to align large language model with recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, RecSys '23, page 1007–1014, New York, NY, USA. Association for Computing Machinery.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Harrison Chase. 2023. langchain. GitHub repository.

Chong Chen, Min Zhang, Chenyang Wang, Weizhi Ma, Minming Li, Yiqun Liu, and Shaoping Ma. 2019. An efficient adaptive transfer neural network for social-aware recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 225–234, New York, NY, USA. Association for Computing Machinery.

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

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

Achraf Gazdar and Lotfi Hidri. 2020. A new similarity measure for collaborative filtering based recommender systems. *Know.-Based Syst.*, 188(C).

Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2022. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). In *Proceedings of the 16th ACM Conference on Recommender Systems*, RecSys '22, page 299–315, New York, NY, USA. Association for Computing Machinery.

Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2023. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). *Preprint*, arXiv:2203.13366.

Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian McAuley, and Wayne Xin Zhao. 2024. Large language models are zero-shot rankers for recommender systems. In *Advances in Information Retrieval: 46th European Conference on Information Retrieval, ECIR 2024, Glasgow, UK, March 24–28, 2024, Proceedings, Part II*, page 364–381, Berlin, Heidelberg. Springer-Verlag.

Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. *Preprint*, arXiv:1808.09781.

Wang-Cheng Kang, Jianmo Ni, Nikhil Mehta, Maheswaran Sathiamoorthy, Lichan Hong, Ed Chi, and Derek Zhiyuan Cheng. 2023. Do llms understand user preferences? evaluating llms on user rating prediction. *Preprint*, arXiv:2305.06474.

Jiacheng Li, Ming Wang, Jin Li, Jinmiao Fu, Xin Shen, Jingbo Shang, and Julian McAuley. 2023. Text is all you need: Learning language representations for sequential recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, page 1258–1267, New York, NY, USA. Association for Computing Machinery.

Junling Liu, Chao Liu, Peilin Zhou, Renjie Lv, Kang Zhou, and Yan Zhang. 2023a. Is chatgpt a good recommender? a preliminary study. *Preprint*, arXiv:2304.10149.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023b. Lost in the middle: How language models use long contexts. *Preprint*, arXiv:2307.03172.

Kelong Mao, Jieming Zhu, Jinpeng Wang, Quanyu Dai, Zhenhua Dong, Xi Xiao, and Xiuqiang He. 2021. Simplex: A simple and strong baseline for collaborative filtering. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21, page 1243–1252, New York, NY, USA. Association for Computing Machinery.

Yohei Nakajima. 2023. babyagi. GitHub repository.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Jiin Park and Misuk Kim. 2025. A scalable unsupervised framework for multi-aspect labeling of multilingual and multi-domain review data. *Preprint*, arXiv:2505.09286.

Yilena Pérez-Almaguer, Raciel Yera, Ahmad A. Alzahrani, and Luis Martínez. 2021. Content-based group recommender systems: A general taxonomy and further improvements. *Expert Systems with Applications*, 179:115092. Received 22 October 2020, Revised 14 May 2021, Accepted 12 June 2021, Available online 30 June 2021.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *Preprint*, arXiv:2302.04761.

John Schulman, Barret Zoph, Christina Kim, Jacob Hilton, Jacob Menick, Jiayi Weng, Juan Felipe Ceron Uribe, Liam Fedus, Luke Metz, Michael Pokorny, and 1 others. 2022. Chatgpt: Optimizing language models for dialogue. `https://openai.com/blog/chatgpt`. OpenAI Blog.

Significant Gravitas. 2023. Auto-gpt. GitHub repository.

Jagendra Singh, Mohammad Sajid, Chandra Shekhar Yadav, Shashank Sheshar Singh, and Manthan Saini. 2022. A novel deep neural-based music recommendation method considering user and song data. In *2022 6th International Conference on Trends in Electronics and Informatics (ICOEI)*, pages 1–7.

An Tang, Xiuzhen Zhang, and Minh Dinh. 2024. Aspect-based key point analysis for quantitative summarization of reviews. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 1419–1433, St. Julian's, Malta. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Manos Tsagkias, Tracy Holloway King, Surya Kallumadi, Vanessa Murdock, and Maarten de Rijke. 2021. Challenges and research opportunities in ecommerce search and recommendations. *SIGIR Forum*, 54(1).

Lei Wang and Ee-Peng Lim. 2023. Zero-shot next-item recommendation using large pretrained language models. *Preprint*, arXiv:2304.03153.

Yinwei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, Richang Hong, and Tat-Seng Chua. 2019. Mmgcn: Multi-modal graph convolution network for personalized recommendation of micro-video. In *Proceedings of the 27th ACM International Conference on Multimedia*, MM '19, page 1437–1445, New York, NY, USA. Association for Computing Machinery.

Yueqi Xie, Peilin Zhou, and Sunghun Kim. 2022. Decoupled side information fusion for sequential recommendation. *Preprint*, arXiv:2204.11046.

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

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. *Preprint*, arXiv:2210.03629.

Yuhui Zhang, HAO DING, Zeren Shui, Yifei Ma, James Zou, Anoop Deoras, and Hao Wang. 2021. Language models as recommender systems: Evaluations and limitations. In *I (Still) Can't Believe It's Not Better! NeurIPS 2021 Workshop*.

Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, page 1893–1902, New York, NY, USA. Association for Computing Machinery.

## A Baseline Model Details

The baseline models used for comparison in this study are described in detail as follows

### A.1 Direct Recommendation Model

- **ENMF (Efficient Neural Matrix Factorization)**: A matrix factorization-based model that effectively utilizes all observed data. It offers balanced performance in terms of computational efficiency and recommendation accuracy, and shows stable results even on sparse datasets.

- **SimpleX**: A structurally simple collaborative filtering model that incorporates a strong cosine contrastive loss, achieving performance comparable to more complex state-of-the-art models. It is particularly advantageous in terms of efficiency and interpretability.

- **P5 (Personalized Prompt for Personalization)**: A prompt-based framework that handles various recommendation tasks in a text-to-text format. It effectively encodes user preferences and item characteristics using natural language processing techniques, and supports generalizable performance through multi-task learning.

- **ChatGPT**: A few-shot recommendation approach based on a large language model, which generates recommendations using prompts without additional fine-tuning. User preferences and item attributes are processed in natural language and provided directly in the prompt.

### A.2 Sequential Recommendation Model

- **SASRec (Self-Attentive Sequential Recommendation)**: A sequential recommendation model based on the self-attention mechanism that effectively captures important signals from users' temporal behavior patterns. It models both short- and long-term dependencies, delivering stable performance across various sequence lengths.

- **S$^3$-Rec (Self-Supervised Sequential Recommendation)**: A model that integrates multiple self-supervised learning objectives to capture rich correlations in user–item sequences. It enhances representational power by jointly optimizing item attributes, sequence patterns, and user preferences.

These baseline models represent widely adopted approaches in current recommender systems research and were selected as comparison points to fairly evaluate the performance of the proposed MADREC framework.

## B Example of Explanation Generation

**Explanation Generation Example**

Based on the user profile, the user values products that are powerful, effective, organic, and have pleasant scents, especially in hair products, with quick and efficient usage. They also prefer affordable items with high demand and utility, and they favor products that reduce frizz, smell good, and are effective for hair and skin care.

1353 : Effective for frizz reduction, pleasant scent, high utility.

Figure B.1: Example of explanation generation based on a user profile and a recommended item. The upper part shows the summarized user preferences, and the lower part provides the natural language explanation for why item 1353 fits the user's needs.

## C Aspect Term & Category

| Aspect Categories and Terms from Beauty Reviews | |
|---|---|
| **Aspect Category** | **Aspect Terms** |
| **Makeup** | shadow, liner, concealer, eyeliner, mascara, eyeshadow, brow, blush, highlighter, primer, bronzer, foundation, palette, lipgloss, powder |
| **Ingredients** | helianthus, annuus, kernel, vegetable, hydrogenated, bran, ester, sunflower, tocopheryl, acetate, glycine, argania, soja, tocopherol, panthenol |
| **Color** | pink, purple, nude, bright, yellow, blue, metallic, beige, gold, shimmer, red, vibrant, coral, bronze, satin |
| **Hair** | wavy, curly, straight, braid, strand, frizzy, ponytail, layered, heat, curl, styling, volume, rinse, shampoo, comb |
| **Beauty Tools** | file, buffer, clipper, cutter, filing, cuticle, pedicure, scissors, drill, electric, grooming, trimming, tweezer, trimmer, manicure |
| **Scent** | musk, sandalwood, mint, aroma, vanilla, jasmine, floral, cinnamon, citrus, lavender, coconut, honey, berry, peppermint, perfume |
| **Purchase** | amazon, cost, expensive, bargain, budget, cheaper, online, overpriced, price, seller, buy, cheapest, pricing, purchase, repurchase |
| **Usage Context** | evening, morning, night, daily, routine, weekend, bedtime, afternoon, overnight, weekly, daytime, frequently, outdoors, workout, wedding |
| **Improvement** | aging, elasticity, reduce, inflammation, dryness, soothe, wrinkle, firmness, collagen, repair, brightening, hydrate, protect, rejuvenate, calming |
| **Packaging** | zipper, case, sealed, magnetic, cardboard, pocket, compartment, pouch, box, sleeve, sturdy, envelope, clip, bag, resealable |
| **Quantity** | four, ten, five, six, three, twenty, ml, oz, seven, eight, two, half, nine, ounce, dozen |
| **Usage Method** | cleansing, washcloth, foam, pat, massage, toner, cleanser, exfoliating, scrub, wiping, towel, rubbing, soaking, dab, blotting |
| **Satisfaction** | nice, great, wonderful, awesome, impressive, excellent, amazing, fantastic, best, perfect, comfortable, attractive, exceptional, durable, unique |

Table C.1: **Extracted Aspect Categories and Terms from Beauty Reviews.** This table presents 13 distinct aspect categories automatically identified from unlabeled Beauty reviews, along with their 15 most representative terms. These categories reveal the key dimensions consumers focus on when evaluating beauty products, ranging from makeup characteristics to scent preferences and improvement effects.

| Aspect Categories and Terms from Sports Reviews | |
|---|---|
| **Aspect Category** | **Aspect Terms** |
| **Functionality** | exceptional, usability, impressive, excellent, robust, improves, outstanding, innovative, efficient, superior, practical, versatile, durable, reliable, strong |
| **Brand** | officially, supreme, luminox, rogue, submariner, fabulous, hydroflask, omega, elite, priceless, british, multiuse, rocksolid, branding, legendary |
| **Usage Context** | vacation, boating, campground, canoeing, concert, festival, adventure, camping, hiking, beach, picnic, weekend, outdoors, trail, snorkeling |
| **Satisfaction** | trust, rely, willing, honest, impressed, interested, believe, aware, expect, hoping, curious, committed, determined, satisfied, pleased |
| **Technology** | bluetooth, wireless, wifi, gps, usb, smartphone, app, network, software, touchscreen, led, charger, sensor, rechargeable, device |
| **Service** | vendor, contacted, request, representative, emailed, distributor, supplier, seller, dealer, merchant, manufacturer, employee, shipped, customer, returned |
| **Quantity/ Measurement** | fifty, ten, twelve, thirty, twenty, approximate, half, couple, three, quarter, two, four, dozen, maximum, ml |
| **Fit** | stretchy, baggy, waistband, roomy, elastic, compression, breathable, padded, expandable, cinched, comfy, spacious, supportive, snug, fitted |
| **Ease of assembly** | screw, clamp, fastener, tighten, bolt, nut, insert, attach, locking, quick-release, pivot, knob, hinge, mounting, latch |
| **Durability** | cracking, tearing, peeling, ripping, scraping, deform, crushed, grinding, scuff, bruised, bending, chipping, snapping, abrasion, damaged |

Table C.2: **Extracted Aspect Categories and Terms from Sports Reviews.** This table presents 10 distinct aspect categories automatically identified from unlabeled Sports and Outdoors reviews, along with their 15 most representative terms. These categories highlight the key dimensions consumers consider when evaluating sports equipment, from functionality and durability to brand reputation and usage contexts.

| Aspect Categories and Terms from Toys Reviews | |
|---|---|
| **Aspect Category** | **Aspect Terms** |
| **Purchase** | amazon, walmart, retailer, seller, discount, refund, sale, coupon, shipping, return, cost, price, purchase, bargain, online |
| **Character** | avenger, batman, bumblebee, megazord, superman, spiderman, joker, catwoman, thor, jedi, darth, hulk, yoda, deadpool, venom |
| **Electronic** | transmitter, controller, signal, frequency, mechanism, adjustment, automatic, manual, remote, controllable, electric, battery, wireless, motorized, joystick |
| **Gameplay** | strategy, player, opponent, mission, scoring, victory, tactic, mechanic, challenge, cooperation, turn, deck, card, phase, role |
| **Food** | pasta, pepper, cupcake, frosting, dough, icing, sprinkles, chocolate, baking, cookie, candy, pizza, cake, muffin, chocolate |
| **Movement** | lift, slide, rotate, tilt, flip, fold, bump, push, pull, wobble, spin, lean, climb, snap, hinge |
| **Age Range** | three, four, five, six, seven, eight, nine, ten, eleven, twelve, thirteen, fourteen, fifteen, sixteen, eighteen |
| **Educational** | leapreader, software, ebooks, touchscreen, tablet, app, phonics, flashcard, workbook, smartphone, digital, headphone, programming, language, instructional |
| **Accessories** | earring, headband, ribbon, scarf, necklace, bracelet, tiara, belt, glove, hat, sunglasses, pouch, mask, hairclip, pendant |
| **Safety** | careful, cautious, supervise, supervision, guidance, injured, danger, responsible, calm, un-supervised, help, tough, nervous, stress, patience |
| **Packaging** | fit, aligned, snap, lock, stored, attach, glued, fasten, folded, screw, stacked, sealed, labeled, carry, wrapped |
| **Animal** | puppy, rabbit, monkey, doggy, kitty, bunny, elephant, panda, giraffe, tiger, owl, kitten, lion, bear, dolphin |

Table C.3: **Extracted Aspect Categories and Representative Terms from Toys Reviews.** This table presents 12 distinct aspect categories automatically identified from unlabeled Toys and Games reviews, along with their 15 most representative terms. These categories reveal the key dimensions consumers focus on when evaluating toys and games, ranging from character-based features to educational value and safety considerations.

| Multi-Aspect Labeling Examples | |
|---|---|
| **Beauty Products** | |
| **Review** | **Multi-Aspect Category** |
| This is the first curling iron i ever used.. and i am not planning to purchase anything else. I had a problem with the Auto on/off button at the beginning since my hand kept on pushing it by mistake, but now that i know the proper way of holding it it doesn't bother me much. I use a heat protectant so i didn't notice any damage to my hair, on the contrary, my curls ended up being soft and shiny! | Improvement, Hair, Purchase |
| Love this stuff. It's perfect for keeping my face soft and smooth, without breaking out. I especially like to use it at night. | Usage Context |
| I have been using this lotion for over a month now and I really like it. I researched new lotions online and this came up as dermatologist recommended so I took a chance and ordered it. It is perfect for moisturizing before putting on make-up because it does not leave the skin oily or greasy. I have sensitive skin and it seems to be perfect for me. | Makeup, Usage Context, Purchase |
| **Sports and Outdoors** | |
| They work really well you can use them in any way they even work out with pull-up bars and can attach it bench and use for reverse push-ups. | Ease of assembly |
| I bought 3 of these to replace the key locks on my weapons. No more having to look for the key or need to turn on the light. If you preset the combo off open, you can open this in the dark. I also like the rubberized center contacts that prevent scratching the finish. | Ease of assembly, Durability |
| These are hands down the best kids goggles out there as they stay put on little faces. The large coverage area also seems to give kids more security in the water and also leaves less chances of them falling off. The material is tacky without being sticky, which is great for holding on to little kids in motion. The many colors are also nice so that each kid can have their own color. They aren't indestructible and the lens can scratch so a bit of care is a good idea, but as far as kids goggles go, this is a good investment to make. | Fit, Durability |
| **Toys and Games** | |
| My nephew (14) suggested this game for my son (7). It couldn't have been a better suggestion. Our son loves trains and understand math well enough to enjoy this game. It's actually fun for me, too. It's really a smarter version of Monopoly. | Age Range, Gameplay |
| This Sabretooth statue, is very nice and menacing. A great pick up for the Wolverine and Sabretooth admirers out there. | Character |
| We are all fans of TinkerBell in my house and I was thrilled to find this for my 4 year old's Innotab 2. It has great games and creative features and is by far her favorite cartridge. The best part is that more than once I have also caught my 17 year daughter playing it as well. | Age Range, Gameplay, Educational |

Table C.4: **Examples of Automatically Assigned Multi-Aspect Categories for Reviews in Beauty, Sports, and Toys Domains.** This table presents sample reviews from the Beauty, Sports, and Toys domains, along with the automatically assigned multi-aspect category labels. These labels are generated by the ASPECT SUMMARY TOOL prior to constructing user and item profiles.

# D Additional Implementation Details

**Aspect Summary Generation Prompt**

You are an intelligent assistant that builds personalized user profiles for a recommendation system.

Your job is to summarize what the user values most regarding the aspect "{aspect}", based on the reviews below.
Only extract information that is directly related to the aspect "{aspect}".
Ignore general praise, irrelevant sentences, or duplicated expressions.

Focus on capturing the user's unique preferences and patterns for this aspect.
Summarize the user's preference or priority into one sentence within 10 words, reflecting what kind of features the user tends to like or look for.

Reviews:
"""
{combined_text}
"""

Answer format:
Aspect: {aspect}
Summary: <Your 10-word sentence here>

Figure D.1: Aspect-based user profiling prompt used in the ASPECT SUMMARY TOOL.

**Direct Recommendation Prompt**

You are a smart recommendation agent.

[User Profile]
Summarize what the user values in products: {user_profile_text}

[Candidate Items]
You are given {len(item_data)} candidate items. Each includes a category and aspect-based profile summary.

{item_blocks.strip()}

[Task]
Based on the user profile and the information for each item, select the top-{top_k} items that best match the user's preferences. For each item, consider how it matches with the user's specific aspects and preferences.

Think **step by step** before making a final decision. Choose the top {top_k} products to recommend in order of priority, from highest to lowest.

**Sequential Recommendation Prompt**

You are a smart recommendation agent.

[User Profile]
Summarize what the user values in products: {user_profile_text}

[User Purchase History]
The user has recently purchased these items in this exact order (oldest to newest):{recent_items_text}

[Candidate Items]
You are given {len(item_data)} candidate items. Each includes a category and aspect-based profile summary.

{item_blocks.strip()}

[Task]
Based on both the user's profile and purchase sequence/pattern, predict the next item the user is most likely to purchase. The sequential pattern and evolution of the user's preferences over time. The user's aspect-based preferences from their profile

Think **step by step** before making a final decision, Choose the top {top_k} products to recommend in order of priority, from highest to lowest.

**Explanation Generation Prompt**

You are a smart recommendation agent.

[User Profile]
Summarize what the user values in products: {user_profile_text}

[Candidate Items]
You are given {len(item_data)} candidate items. Each includes a category and aspect-based profile summary.

{item_blocks.strip()}

[Task]
Based on the user profile and the information for each item, select the top-{top_k} items that best match the user's preferences and explain the recommendation reason based on aspects. For each item, consider how it matches with the user's specific aspects and preferences.

Think **step by step** before making a final decision, Choose the top {top_k} products to recommend in order of priority, from highest to lowest.

[Example]
Explanation:
- id1: Brief explanation how this item matches user's specific aspects (15 words max)

Figure D.2: Prompt templates used for recommendation tasks, including direct recommendation, sequential prediction, and human evaluation criteria, illustrating the input structure and task instructions for each scenario.

**SELF-FEEDBACK Prompt for RE-RANKING**

You are a recommendation system weight analysis expert.

[User Profile]
{user_profile_text}

[Previously Recommendation]
{'\n'.join([f"- item['title'] (item['category'])" for item in prev_recommended_info])}

[Current Weights]
- Profile similarity: 0.4
- Category similarity: 0.4
- Popularity: 0.2

Analysis:
1. What are the differences between the actually selected item and recommended items?
2. How should weights be adjusted to rank the actual item higher?

Propose new weights in the following format:
{
  "profile_similarity": 0.X,
  "category_similarity": 0.X,
  "popularity": 0.X,
  "reasoning": "Explanation for weight adjustment"
}

**SELF-FEEDBACK Prompt For No RE-RANKING**

You are a recommendation system that needs to improve its strategy.

[User Profile]
{user_profile_text}

[Previous Recommendation]
You previously recommended these items, but the customer didn't choose any of them:
{'\n'.join([f"- item['title'] (item['category'])" for item in prev_recommendations_details])}

[All Candidate Items]
{item_blocks.strip()}

[Task]
Since the customer didn't choose any of your previous recommendations, you need to:
Reconsider your recommendation strategy
Think about different aspects or categories that might better match the user's preferences
Select {top_k} different items that could better satisfy the customer's needs

Try to recommend items from different categories or with different characteristics than before.

Choose the top {top_k} products to recommend in order of priority, from highest to lowest.

Figure D.3: SELF-FEEDBACK prompt templates used in MADREC differ in feedback format depending on whether RE-RANKING is applied or not.