ClusterRAG: Cluster-Based Collaborative Filtering for Personalized Retrieval-Augmented Generation

Anonymous ACL submission

Abstract

Personalized Retrieval-Augmented Generation (RAG) relies on accurately selecting user-relevant documents. In practice, existing approaches often suffer from high retrieval costs and overlook that collaborative signals from similar users can enhance personalized generation for the current user. We propose **ClusterRAG**, a **Cluster**-Based Collaborative Filtering for Personalized **R**etrieval-**A**ugmented **G**eneration. ClusterRAG represents users through their profile documents, organizes users into semantically coherent clusters using density-based clustering, and performs retrieval at both the cluster and document levels via cluster-level similarity and fine-grained ranking. Extensive experiments on the LaMP benchmark demonstrate that jointly leveraging the target user's profile and profiles from top similar users consistently yields the best performance across diverse tasks. Further analysis shows that ClusterRAG integrates seamlessly with different dense retrievers and rankers, and remains effective when paired with both fine-tuned and zero-shot language models.

1 Introduction

Retrieval-Augmented Generation (RAG) has emerged as a powerful paradigm for knowledge-intensive language tasks by combining parametric knowledge in large language models (LLMs) with non-parametric retrieval over external documents, significantly reducing hallucinations and improving factuality in text generation (Lewis et al., 2020). RAG systems typically retrieve documents related to the immediate query, then condition a generative model on those documents to produce responses (Fan et al., 2024; Huang and Huang, 2024; Li et al., 2025b). Despite impressive gains, current RAG pipelines often ignore long-term user information and inter-user relationships when constructing retrieval contexts, limiting personalization and the ability to leverage analogous users' knowledge for improved generation quality.

Personalization is crucial in many real-world applications (e.g., personal assistants, tutoring systems, and personalized search) because user history and preferences strongly influence what information is relevant and how it should be framed (Li et al., 2025a; Ahmad et al., 2025). Existing personalization strategies in RAG largely fall into two extremes: (1) user-only approaches that condition retrieval and prompts solely on a user's own profile, which can be sparse or noisy (Salemi et al., 2024b; Zerhoudi and Granitzer, 2024; Dong et al., 2025), and (2) non-personalized approaches that ignore user history altogether (Lewis et al., 2020; Asai et al., 2024; Yang et al., 2025; Zhang, 2025). Both approaches miss a middle ground where signals from similar users can enrich prompts while preserving user-specific nuance. Recent surveys (Xu et al., 2025; Li et al., 2025a) highlight the potential of end-to-end personalization across the RAG pipeline, from pre-retrieval user modeling to retrieval and generation, but also emphasize practical challenges such as balancing personal vs. collaborative signals.

Collaborative filtering (CF), long established in recommender systems (Xue et al., 2017; Wang et al., 2019; Sgardelis et al., 2025; Shi et al., 2025), naturally complements personalization by exploiting similarities between users to infer missing preferences or relevant content. However, direct application of CF to RAG introduces new questions: (1) *how should users be represented for retrieval tasks?* (2) *how to retrieve similar users to capture heterogeneous behavior at scale?* and (3) *how to leverage both collaborative documents and target user's profile when forming prompts for LLMs?*

In this paper, we propose **Cluster**-Based Collaborative Filtering for Personalized **R**etrieval-**A**ugmented **G**eneration (**ClusterRAG**), a practical pipeline that (1) constructs compact user repre-

sentations by aggregating each user's profile documents into embeddings, (2) groups users into clusters using HDBSCAN (McInnes et al., 2017) to reveal cohorts of similar users and builds a cluster-level ranking matrix by scoring intra-cluster user similarities with effective rankers, e.g., ColBERT (Khattab and Zaharia, 2020), and (3) clusters and retrieves candidate profile documents from the top k similar users to form collaborative, user-only, or hybrid prompts for downstream generation. Our method explicitly leverages cluster structure to reduce search complexity, provide robust neighbor selection in variable-density settings, and enable principled mixing of collaborative and individual signals when constructing prompts. We evaluate multiple retrievers (ColBERT, Contriever (Izacard et al., 2022), BGE (Xiao et al., 2024), BM25 (Robertson et al., 1995), Recency and Random). Beyond methodological novelty, we also demonstrate that ClusterRAG is model-agnostic and robust across architectures and retrieval backbones. ClusterRAG integrates seamlessly with both fine-tuned sequence-to-sequence (seq2seq) encoder-decoder language models and zero-shot LLMs, without requiring any model-specific adaptation.

We validate ClusterRAG on the LaMP benchmark (Salemi et al., 2024b) and report improvements in personalized generation quality compared to non-personalized and naive user-only baselines. We also provide a link to an anonymous project repository for ClusterRAG [1].

The remainder of this paper unfolds as follows. The next section presents related work; Section 3 provides problem formulation; Section 4 describes the ClusterRAG framework; Section 5 presents the experimental evaluation; and Section 6 presents the conclusion, limitations, and ethical considerations. Additional analysis and results are provided in the Appendix.

2 Related Work

Retrieval-Augmented Generation (RAG). The adoption of RAG has demonstrated improvements across a range of tasks, including question answering, dialogue comprehension, and code generation (Lewis et al., 2020; Xu et al., 2023; Zerhoudi and Granitzer, 2024; Fan et al., 2024). Early RAG pipelines typically index a shared corpus (e.g., Wikipedia or domain corpora) and retrieve passages conditioned only on the current query; the retrieved passages are then used to condition an LLM at generation time (Lewis et al., 2020; Zhang, 2025). Subsequent work has explored numerous improvements to retriever architectures, reranking strategies, and retrieval-generation coupling mechanisms (Gao et al., 2023; Siriwardhana et al., 2023; Fan et al., 2024). Even though these works establish strong task-agnostic baselines and sophisticated retriever-generator interfaces, they typically operate in a *user-agnostic* manner and do not leverage a user's long-term profile or cross-user signals when selecting documents for conditioning.

Personalized Retrieval-Augmented Generation. A growing body of work studies how RAG can be adapted for personalized applications (e.g., personal assistants, tutoring, and individualized question answering) by incorporating user history, preferences, or authoring signals into retrieval and prompting (Salemi et al., 2024b; Zerhoudi and Granitzer, 2024; Li et al., 2025a). Some systems personalize LLMs by (1) fine-tuning model parameters (either fully or selectively) for individual users (Li and Liang, 2021; Hu et al., 2022; Zollo et al., 2025), (2) incorporating latent user representations into the model (Ning et al., 2025; Qiu et al., 2025; Huber et al., 2025), and (3) augmenting model prompts with a user profile or a small set of user documents (Zamani et al., 2022; Salemi et al., 2024b,a). The first two strategies require modifying the model's architecture or parameters, which can be expensive, or in some cases infeasible, due to storage, computational, and time constraints. In addition, they cannot perform well for cold-start users. In contrast, the third approach, which is adopted in this work, can be applied to any generative model (Salemi et al., 2024a). User-specific personalization works well when profiles are dense and representative; this notwithstanding, user-only personalization suffers when profiles are sparse, noisy, or unrepresentative of the current intent.

Collaborative Personalized Retrieval Augmented Generation. Collaborative filtering has been extensively studied in recommender systems and consistently shown to be effective (Xue et al., 2017; Wang et al., 2019; Zhang et al., 2024; Shen et al., 2024; Shi et al., 2024; Tang et al., 2025; Xin et al., 2025; Zhang et al., 2025). The core premise is that users with comparable interaction histories tend to exhibit similar preferences; therefore, leveraging items favored by similar users can help generate relevant recommendations for a target user.

[1] https://github.com/academicprojects44/anonymous

Recent efforts have started to marry CF and retrieval for generation (Shi et al., 2025; Zhu et al., 2025). For instance, Shi et al. (2025) employs contrastive learning to generate user embeddings that retrieve similar users and incorporate collaborative signals with a user input for prompt creation.

Despite this progress, two practical challenges remain unresolved in current collaborative RAG research. First, naively computing pairwise similarities across millions of users is costly; clustering users into cohorts can reduce search complexity. ClusterRAG is designed to address this issue by combining user-level clustering with document-level collaborative retrieval and introducing cluster-level ranking matrices that summarize intra-cluster user similarity and thereby enabling robust neighbor selection even in variable-density cohorts. Second, once neighbor users are found, selecting which of their documents to include and how to merge collaborative documents with a target user's own profile remains an open design choice with direct impact on generation quality. ClusterRAG uses flexible prompt fusion modes and evaluates multiple retrievers, showing empirical robustness across both fine-tuned and training-free generative models.

To the best of our knowledge, ClusterRAG is the first framework to integrate user-level clustering with collaborative document retrieval for personalized RAG, explicitly leveraging cross-user similarity to enrich sparse user profiles for personalized generation.

3 Problem Formulation

A standard RAG setup consists of two core components, retrieval and generation: given an input query x, the model predicts the most probable output sequence y conditioned on x and a retrieved document d. Personalized RAG extends this formulation by conditioning generation on a user u, typically represented through a user profile. Formally, we let $T = \{(u_1, x_1, y_1), (u_2, x_2, y_2), \ldots, (u_N, x_N, y_N)\}$ denote a set of N training instances, where each tuple consists of a user u, a user-issued input query x, and a corresponding personalized ground-truth output y. For each user u, a user profile U_p is available and serves as an auxiliary context for personalized generation. The profile $U_p = \{d_1, d_2, \ldots, d_n\}$ is a collection of personal documents or historical records associated with u, such as past queries and generated outputs.

In ClusterRAG, given a target user u, our objective is to identify the top k most similar users using a clustering-assisted ranking strategy. We then retrieve and rank the top m documents from these users using a retriever (ranker) R.

4 ClusterRAG Framework

ClusterRAG is built on the intuition that users with similar behaviors and preferences can provide valuable contextual signals for one another. By combining information from a target user's own profile with carefully selected profiles from similar users, ClusterRAG enhances an LLM's ability to generate accurate and personalized responses. As shown in Figure 1, ClusterRAG framework consists of three main stages: (1) user representation and retrieval, (2) profile retrieval, and (3) personalized generation.

4.1 User Representation and Retrieval

Since explicit user representations are typically unavailable, we first construct user embeddings from observed interaction data. All data instances associated with a user are aggregated into a user-level profile U_p. Each document $d_i \in U_p$ is encoded using a dense embedding model, specifically ColBERTv2 (Santhanam et al., 2022). A compact user representation is then obtained by averaging document embeddings:

$$\mathbf{z}_u = \frac{1}{n_u} \sum_{i=1}^{n_u} f(d_i), \tag{1}$$

where $f(\cdot)$ denotes the embedding function.

Computing similarities between all user pairs is prohibitively expensive at scale. To address this, ClusterRAG groups users into similarity-based cohorts using a hierarchical density-based clustering method, HDBSCAN (McInnes et al., 2017), which automatically identifies clusters of varying density, making it well-suited for collaborative filtering in our setting, where the number of user groups in the user-document collection is unknown a priori.

Clustering alone does not quantify the relative similarity of users within each cluster. Therefore, we compute intra-cluster similarities using a modern reranker, ColBERTv2 (Santhanam et al., 2022), which provides fine-grained token-level interactions inherited from ColBERT (Khattab and Zaharia, 2020) while incorporating residual compression and denoised supervision for improved effi-

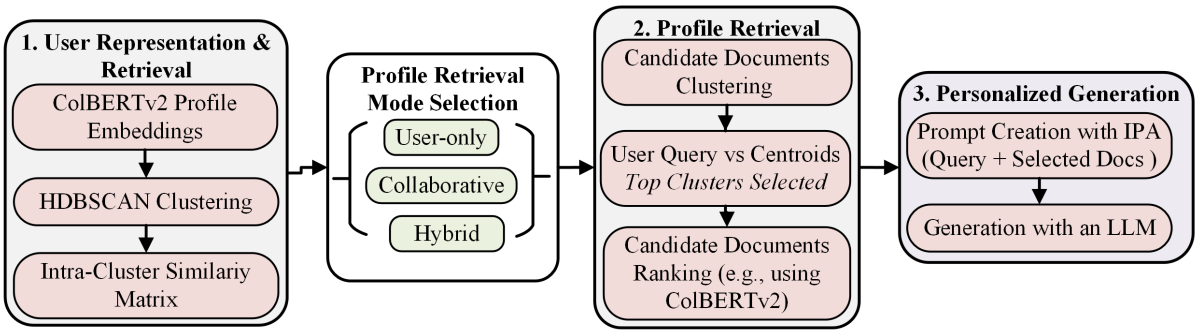

Figure 1: Overview of the ClusterRAG framework.

ciency and generalization. These properties make ColBERTv2 well-suited for robust similarity estimation between user profiles.

Cluster-Level Similarity Ranking. This step aims at restricting similarity computation to cluster members to improve robustness and scalability by focusing comparisons on behaviorally consistent cohorts. For each cluster C, we construct an intra-cluster similarity matrix R^C defined as:

$$R^C_{u,v} = ColBERTv2(\mathbf{z}_u, \mathbf{z}_v), \qquad (2)$$

where $u, v \in C$ and $v \neq u$. The diagonal entries in R^C correspond to self-similarity and are discarded. For each user u, we finally retain an ordered list of the top k most similar users within the same cluster.

4.2 Profile Retrieval

The profile retrieval stage integrates search and ranking to identify documents that are most beneficial for personalized generation (Huang and Huang, 2024). Incorporating collaborative filtering introduces two key challenges: selecting relevant documents from similar users and effectively leveraging both collaborative documents and the target user's documents for personalized RAG. ClusterRAG addresses these challenges by leveraging cluster structure to provide both topical coherence and retrieval efficiency and organizes profile retrieval as follows.

Profile Retrieval Modes. First, given a query q from user u, we retrieve candidate profile documents using one of the following three retrieval modes. (1) *User-only retrieval*: the simplest strategy, which considers only the user's own profile. (2) *Collaborative retrieval*: this mode retrieves documents from the profiles of the top k most similar users. It is particularly beneficial for sparse or cold-start users, whose own profiles may not adequately capture the intent of the current query. (3) *Hybrid retrieval*: this mode combines both user-only and collaborative profiles. As a result, the effective profile U_p used for generation may consist of: (i) documents from the target user only, (ii) documents from similar users only, or (iii)similar users alone, or (iii) a combination of documents from both sources.

Clustering for Topical Organization. After selecting a profile retrieval mode, all candidate profile documents are encoded using a dense retriever, such as ColBERTv2, and partitioned into clusters using HDBSCAN, producing a set of clusters $\mathcal{C} = \{C_1, \ldots, C_K\}$ with corresponding computed centroids $\mathcal{M} = \{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K\}$. This clustering captures latent topical structure and enables ClusterRAG to automatically infer the number of topics present in a user's profile without requiring prior specification.

Cluster-Level Indexing and Retrieval. Finally, ClusterRAG employs a two-stage retrieval strategy. First, a cluster index stores centroid embeddings: given a query q, its embedding e_q is computed similarly as document embeddings, and compared against all centroids, and the top B clusters are selected from $\mathcal{C}$. Second, within each selected cluster, documents are retrieved and reranked by similarity to e_q, and the top m documents are selected for generation.

This hierarchical retrieval reduces complexity from $\mathcal{O}(N)$ to $\mathcal{O}(K + B \cdot N/K)$, where N is the total number of documents, each cluster C_i contains $\approx |N/K|$ documents, and $B \leq K \leq N$.

We use ColBERTv2 as the primary model to retrieve and rerank profile documents; even so, our experiments evaluate additional dense, sparse, heuristic, and random retrievers to demonstrate the framework's retriever-agnostic design.

4.3 Personalized Generation

Effective generation by LLMs critically depends on well-engineered prompts that are tailored to the downstream task. Prompts allow seamless integration of pre-trained models into downstream tasks by eliciting desired model behaviors solely based on the given prompt (Sahoo et al., 2025). To effectively leverage selected user documents from U_p, ClusterRAG adopts *In-Prompt Augmentation (IPA)* (Salemi et al., 2024b), which integrates the user query with all relevant retrieved documents directly within the prompt. IPA is particularly well suited for ClusterRAG, as it can be applied to both training-free (zero-shot) and fine-tuned settings and is compatible with a wide range of model architectures. Accordingly, ClusterRAG supports both fine-tuned LLMs and zero-shot generative models and can be combined with a variety of state-of-the-art document retrievers.

Specifically, to balance user profile context and query specificity, given a maximum prompt length $L_{\max}$, the allocated profile length is computed as:

$$|U_p| = \mathcal{G}_t(L_{\max} - \min(|q|, \lfloor \gamma L_{\max} \rfloor)), \quad (3)$$

where $\gamma \in [0, 1]$ is a tunable mixing parameter, $|q|$ is query length, and $\mathcal{G}_t(\cdot)$ is a task-specific prompt generator (we provide detailed task-specific prompt generators in Section 5.2 and Appendix A). This formulation allows ClusterRAG to incorporate strong personalization signals while preserving the relevance of the current query.

5 Experiments

In this section, we present the experimental setup employed with ClusterRAG, including the datasets, baseline models, evaluation metrics, implementation details, and experimental results. We also provide an overview of the prompts used in our experiments.

5.1 Experimental Setup

Datasets. Our experiments employ the LaMP benchmark (Salemi et al., 2024b), a publicly available dataset that covers a broad range of personalized text generation tasks. The benchmark consists of three personalized text classification tasks and four personalized text generation tasks. One of the four text generation tasks, **LaMP-6** (Personalized Email Subject Generation), is excluded in this work because its data is not publicly available. Specifically, the remaining tasks include: **LaMP-1:** *Personalized Citation Identification*, formulated as a binary classification task; **LaMP-2:** *Personalized Movie Tagging*, a 15-class categorical classification task; **LaMP-3:** *Personalized Product Rating*, an ordinal classification task predicting ratings from one to five stars for e-commerce products; **LaMP-4:** *Personalized News Headline Generation*; **LaMP-5:** *Personalized Scholarly Title Generation*; and **LaMP-7:** *Personalized Tweet Paraphrasing*. ClusterRAG experiments follow the time-based LaMP split to partition the data into training, validation, and test sets. We provide statistics of the dataset in Table 1, detailed dataset statistics in Table 6 in Appendix B, and detailed task descriptions in Appendix C. We additionally provide dataset licensing information in Appendix B.

Task	#users	#train	#dev	#test
LaMP-1	6542	6542	1500	1500
LaMP-2	929	5073	1410	1557
LaMP-3	20000	20000	2500	2500
LaMP-4	1643	12500	1500	1800
LaMP-5	14682	14682	1500	1500
LaMP-7	13437	13437	1498	1500

Table 1: Statistics of the LaMP benchmark with time-based data split.

Evaluation Metrics. Following previous work (Salemi et al., 2024b,a; Shi et al., 2025), we evaluate LaMP-1 and LaMP-2 using Accuracy and F1-measure, and LaMP-3 using Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). We evaluate text generation performance on LaMP-4, LaMP-5, and LaMP-7 using ROUGE-1 (R-1) and ROUGE-L (R-L) (Lin, 2004).

Baseline Models. We firstly compare ClusterRAG to **no personalization** and call this **vanillaRAG**. In this baseline, the generative model is presented with the original task's input without any profile documents to assess whether personalization improves the model effectiveness. Then we consider personalized baselines: (1) **User-only** models, which include (a) **ROPG** (Salemi et al., 2024a): it optimizes the dense retrieval model based on the results generated by a LLM and (b) **LaMP-IPA**: it was introduced with the LaMP benchmark, and it uses in-prompt augmentation for prompt creation; (2) **Collaborative** baseline, **CFRAG** (Shi et al., 2025), that uses contrastive learning to find similar users. To the best of our

knowledge, CFRAG is the only existing collaborative work for personalized RAG; therefore, we compare these baselines with three versions of ClusterRAG: user-only, collaborative, and hybrid profile retrieval modes.

Implementation details. We implement ClusterRAG using the HuggingFace transformers framework (Wolf et al., 2020) and the PyTorch library (Paszke et al., 2019). ClusterRAG adopts a fine-tuned FlanT5-base (Chung et al., 2024) for generation; unless explicitly stated otherwise (in experiments with zero-shot LLMs), it uses a causal Qwen2-7B-Instruct (Yang et al., 2024) or a seq2seq FlanT5-XXL (Chung et al., 2024); all models are open-source. Training is performed using the Trainer or Seq2SeqTrainer APIs[2], depending on whether the LLM backbone is a causal or seq2seq model. FlanT5-base has 250M parameters; Qwen2-7B-Instruct uses 7.07B parameters; and FlanT5-XXL has 11B parameters.

We use AdamW (Loshchilov and Hutter, 2019) optimization with a learning rate of 5×10^{-5}, weight decay of 10^{-4}, linear learning rate scheduling, and a warm-up ratio of 0.05. Models are trained for up to 30 epochs, with evaluation and checkpointing conducted at the end of each epoch. The maximum prompt and output lengths ($L_{\max}$ and $|\bar{y}|$) of generative models is set to 512 and 128 tokens, respectively. Maximum sequence length for embedding models is set to 256 tokens. We set the number of collaborative (similar) users, k, to 1 and retrieved profile documents, m, to 2. γ for prompt formulation in Equation 3 is set to 0.55, while batch size is set to 16. Beam search (Freitag and Al-Onaizan, 2017) with a beam size of 4 is employed for text generation. All hyperparameters for the baseline models are searched according to the settings in the original papers. We select the optimal hyperparameters for ClusterRAG via grid search and report the corresponding tuning grid in Table 7 in Appendix D. All experiments are conducted on a Quadro RTX 8000 GPUs, 48 GB VRAM for a range of 10-24 hours per experiment depending on the task.

5.2 Prompts Used in ClusterRAG

This subsection presents the prompt templates employed during generation. Each prompt contains an instruction, input, and profile. We provide the template for LaMP-1 only below; templates for other tasks are presented in Appendix A. In the template, {*Paper abstract*} and {*Reference list*}, {*Movie description*} represent user input for the corresponding LaMP tasks, while {*Paper abstract*} represent user profile entries. The remaining text is the instruction guiding an LLM to generate the intended output.

> **LaMP-1 Prompt Template:** Given an author who has previously written papers {*Paper list*} and now has written {*Paper abstract*}. Which reference below is related? Just answer with [1] or [2] without explanation. {*Reference list*}

5.3 Experimental Results

When reporting experimental results, we identify statistically significant differences of ClusterRAG performance using a two-tailed paired t-test for generation and ordinal classification evaluation (ROUGE-1, ROUGE-L, MAE, and RMSE) and McNemar test for categorical text classification evaluation (Accuracy and F1). We report results comparing ClusterRAG with baselines, analyzing its retriever-agnostic design, language model versatility, and ablation study. In all tables, the symbol $\uparrow$ indicates that higher values are better, while the symbol $\downarrow$ indicates that lower values are better; all results presented are obtained from a single experimental run.

5.3.1 Comparison with Baselines

Comparison results are presented in Table 2, which shows that ClusterRAG consistently outperforms all baselines across the LaMP benchmark. Importantly, the hybrid variant (*ClusterRAG-H*) achieves the best performance on every task. These improvements indicate that retrieving and aggregating documents from similar users substantially enhances RAG, providing more relevant and personalized evidence than standard RAG pipelines. Notably, ClusterRAG achieves strong performance using only two profile documents, whereas baseline methods require at least four documents to reach their optimal results, indicating that ClusterRAG is well suited for low-resource personalization settings. While *ClusterRAG-C* (collaborative) and *ClusterRAG-U* (user-only) achieve competitive second-best results on several tasks, their combination in the hybrid model yields the most robust and consistent gains across diverse task settings.

[2] https://github.com/huggingface/transformers

Models	LaMP-1		LaMP-2		LaMP-3		LaMP-4		LaMP-5		LaMP-7	
	Acc.↑	F1↑	Acc.↑	F1↑	MAE↓	RMSE↓	R-1↑	R-L↑	R-1↑	R-L↑	R-1↑	R-L↑
vanillaRAG	0.630	0.630	0.520	0.440	0.371	0.709	0.171	0.154	0.462	0.413	0.310	0.273
LaMP-IPA	0.674	0.664	0.570	0.522	0.289	0.608	0.175	0.169	0.472	0.423	0.508	0.457
ROPG	0.644	0.322	0.468	0.031	0.346	0.692	0.184	0.163	0.464	0.396	0.353	0.288
CFRAG	0.633	0.327	0.534	0.036	0.354	0.707	0.162	0.141	0.473	0.425	0.375	0.306
ClusterRAG-C	0.674*	0.673*	0.644	0.607	0.284	0.624	0.179	0.157	0.480*	0.430*	0.507	0.454
ClusterRAG-U	0.645	0.645	0.649*	0.612*	0.271*	0.599*	0.184*	0.165*	0.475	0.425	0.514*	0.464*
ClusterRAG-H	**0.690**	**0.690**	**0.661**	**0.620**	**0.270**	**0.594**	**0.190**	**0.176**	**0.490**	**0.440**	**0.521**	**0.470**

Table 2: Comparison of the performance of ClusterRAG with baselines on the LaMP benchmark. Best results are shown in **bold** and second-best in underlined; boldface indicates statistically significant improvements over the second-best ($p < 0.05$). The symbol * denotes the second-best ClusterRAG variant.

5.3.2 ClusterRAG Retriever-Agnostic Design

In addition to *ColBERTv2*, ClusterRAG explores five more retrievers: (1) a dense unsupervised dual-encoder retriever, *Contriever* (Izacard et al., 2022), (2) a fine-tuned multilingual dense retriever optimized for semantic similarity and retrieval tasks, *BGE* (Xiao et al., 2024), (3) a classical sparse lexical retriever based on term frequency-inverse document frequency (TF-IDF), *BM25* (Robertson et al., 1995), (4) a heuristic retriever that ranks documents solely based on temporal proximity to the query time, favoring the most recently published documents, *Recency*, and (5) a non-informative baseline that samples documents uniformly at random, *Random*. We provide retriever-agnostic design results in Table 3 for LaMP-(1,2,7) and Figure 2 for LaMP-5. The table demonstrates that ClusterRAG consistently benefits from stronger retrievers, with dense semantic models outperforming sparse and heuristic baselines across all LaMP tasks. *ColBERTv2 (ColBERT or Col)* achieves the best overall performance on all tasks, highlighting the advantage of late-interaction matching in personalized retrieval. *BGE* and *Contriever (Con)* provide competitive performance, confirming the retriever-agnostic nature of ClusterRAG, while *BM25* and *Recency (Rec)* offer modest gains over *Random (Ran)* but lag behind dense methods. These results indicate that ClusterRAG is robust across retrieval paradigms, yet most effectively leverages high-capacity dense retrievers to maximize personalization and generation quality.

5.3.3 ClusterRAG LLM Versatility

Table 4 reports the performance of ClusterRAG on LaMP-(1,2,5) when paired with zero-shot LLMs: FlanT5-XXL and Qwen2-7B-Instruct. For each LLM, we compare a non-personalized variant (*nFlan*, *nQwen2*) against its personalized counterpart (*pFlan*, *pQwen2*). As shown in Table 4, personalized variants consistently outperform their non-personalized counterparts across all tasks, demonstrating the effectiveness of ClusterRAG in injecting user-specific and collaborative signals into generation. Notably, *pFlan* achieves the strongest overall performance on LaMP-1, while *pQwen2* attains the best results on LaMP-2 and LaMP-5, indicating that the benefits of ClusterRAG generalize across model architectures, providing consistent gains without requiring additional model fine-tuning.

Retrievers	LaMP-1		LaMP-2		LaMP-7	
	Acc.↑	F1↑	Acc.↑	F1↑	R-1↑	R-L↑
Random	0.640	0.639	0.609	0.608	0.500	0.449
Recency	0.659	0.650	0.618	0.610	0.507	0.456
BM25	0.662	0.658	0.629	0.621	0.510	0.460
Contriever	0.681	0.681	0.649	0.623	0.511	0.459
BGE	0.684	0.682	0.658	0.613	0.509	0.461
ColBERT	**0.690**	**0.690**	**0.661**	**0.620**	**0.521**	**0.470**

Table 3: Comparison of the performance of ClusterRAG under different retrievers on LaMP-(1,2,7).

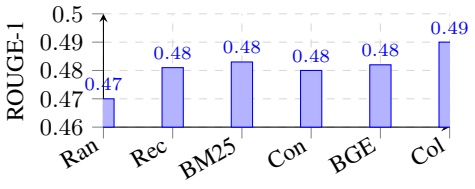

Figure 2: Retrivers' ROGUE-1 scores on LaMP-5.

5.3.4 Ablation Study

The integral components of ClusterRAG are user representation and retrieval and profile retrieval. By systematically removing or replacing these in-

LLMs	LaMP-1		LaMP-2		LaMP-5	
	Acc.↑	F1↑	Acc.↑	F1↑	R-1↑	R-L↑
nFlan	0.546	0.540	0.451	0.448	0.431	0.398
pFlan	**0.648**	**0.647**	0.601	0.601	0.484	0.432
nQwen2	0.602	0.600	0.521	0.521	0.457	0.419
pQwen2	0.639	0.635	**0.610**	**0.606**	**0.488**	**0.447**

Table 4: Performance comparison of ClusterRAG using different LLMs on LaMP-(1,2,5).

Derivatives	LaMP-3		LaMP-7	
	MAE↓	RMSE↓	R-1↑	R-L↑
w/o user clustering	0.320	0.637	0.458	0.371
w/o intra-cluster sim	0.329	0.639	0.501	0.442
w/o doc ranking	0.331	0.642	0.462	0.413
Centroids only	0.400	0.643	0.472	0.438
k-means	0.291	0.610	0.502	0.453
ClusterRAG	**0.270**	**0.594**	**0.521**	**0.470**

Table 5: Ablation study of ClusterRAG on LaMP-(3,7).

dividual components, we assess their impact on personalized generation performance on selected LaMP tasks in Table 5.

First, we examine the role of collaborative user modeling. Replacing clustering-based neighbor selection with random user sampling (*w/o user clustering*) leads to a substantial degradation across all tasks, highlighting the importance of structured user grouping. Similarly, removing intra-cluster similarity ranking (*w/o intra-cluster sim*) consistently reduces performance, indicating that fine-grained similarity estimation within clusters is critical for identifying truly relevant collaborative signals. Lastly, we analyze the profile retrieval module. Using cluster centroids alone to represent user profiles (*Centroids only*) results in the largest performance drop, demonstrating that document-level evidence is essential. Excluding document ranking (*w/o doc ranking*) further degrades results, confirming that effective reranking is necessary to prioritize high-quality contextual evidence. Replacing HDBSCAN with *k-means* (Na et al., 2010) clustering yields slightly weaker yet competitive performance, suggesting that while ClusterRAG is robust to the choice of clustering algorithm, density-aware clustering provides additional benefits.

5.4 Discussion

Computational Effectiveness. ClusterRAG consistently improves personalized generation across diverse tasks and evaluation metrics by jointly leveraging user-specific history and collaborative signals from similar users. The gains observed in both classification and generation settings indicate that clustering-based collaborative filtering provides complementary information beyond individual user profiles, particularly for sparse or ambiguous queries.

Computational Efficiency. ClusterRAG is computationally efficient due to its modular and lightweight design. User and document clustering is performed using HDBSCAN, a non-parametric algorithm without learnable parameters, enabling fast and scalable user grouping. The primary retriever, ColBERTv2, maintains a parameter size comparable to BERT by introducing only a small linear projection layer (approximately 0.1M parameters), resulting in a total model size of roughly 110M parameters. Since user and document embeddings can be precomputed offline, inference primarily involves similarity computations, yielding low latency and minimal overhead. This efficiency allows ClusterRAG to scale effectively and generalize rapidly to new datasets.

We further investigate the sensitivity of ClusterRAG to the number of similar users and the size of the retrieved profile in Appendix E. In addition, Appendix F evaluates cluster cohesion for collaborative user retrieval, and Appendix G presents a qualitative case study that illustrates the performance of ClusterRAG.

6 Conclusion

This work introduced ClusterRAG, a collaborative framework that organizes users and their documents into semantically coherent clusters and performs retrieval at both the cluster and document levels, effectively reducing search complexity while preserving retrieval quality. Extensive experiments on the LaMP benchmark demonstrate that the hybrid profile retrieval mode, which jointly leverages the target user's profile and profiles from top similar users, is the most effective configuration, yielding the best overall performance. Additionally, the experiments indicate that ClusterRAG is retriever-agnostic, allowing seamless integration with different dense retrievers and rankers, and remains effective when paired with both fine-tuned and zero-shot language models, highlighting its robustness and generality. Overall, ClusterRAG offers a design approach that improves effectiveness without incurring significant computational overhead.

Limitations

While ClusterRAG's primary goal is to retrieve similar-user documents and enhance personalized RAG, we highlight a few factors that may affect the model's performance. First, ClusterRAG relies on prompt-based generation, and the adopted IPA strategy may not be optimal; more advanced and structured prompt formulation techniques could further improve personalized generation performance. Nevertheless, prompt engineering, which is crucial for performance of LLMs, is not the central objective of this study. Second, the LaMP-1 (Personalized Citation Identification) and LaMP-5 (Personalized Scholarly Title Generation) tasks provide only paper abstracts rather than full-text content, which may limit the contextual information available to LLMs when selecting citations or generating titles, even though abstracts offer useful sequence constraints. Third, our evaluation is restricted to English, text-only datasets, leaving the effectiveness of ClusterRAG in multilingual and multimodal settings unexplored. Finally, ClusterRAG's end performance depends on the underlying language model, which may introduce additional limitations inherited from the backend LLM. Future work will focus on designing more sophisticated prompt generation strategies and extending ClusterRAG to multilingual and multimodal personalized RAG scenarios.

Ethical Considerations

ClusterRAG leverages user interaction histories and collaborative signals, which raises considerations related to privacy, consent, and potential bias. Although the framework operates on anonymized user profiles and does not require access to explicit personal identifiers, improper handling of user-generated data could still risk unintended information leakage. Moreover, collaborative filtering may amplify existing biases if certain user groups or preferences are overrepresented in the data, potentially affecting fairness in personalized outputs. To mitigate these risks, ClusterRAG can be deployed with standard data governance practices, including data anonymization, access control, and bias-aware evaluation. Importantly, ClusterRAG is a model-agnostic retrieval framework rather than a user profiling system, and it does not infer sensitive attributes beyond observed interactions, helping limit ethical exposure while enabling effective personalization.

References

Aleena Ahmad, Gibson Nkhata, Abdul Rafay Bajwa, Hannah Marsico, Bryan Le, and Susan Gauch. 2025. Colbert-based user profiles for personalized information retrieval. In *Proceedings of the Seventeenth International Conference on Information, Process, and Knowledge Management*, eKNOW '25, pages 51–58, Nice, France.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*, Vienna, Austria.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, and 16 others. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Guanting Dong, Yutao Zhu, Chenghao Zhang, Zechen Wang, Ji-Rong Wen, and Zhicheng Dou. 2025. Understand what llm needs: Dual preference alignment for retrieval-augmented generation. In *Proceedings of the ACM on Web Conference 2025*, WWW '25, page 4206–4225, New York, NY, USA. Association for Computing Machinery.

Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 6491–6501, New York, NY, USA. Association for Computing Machinery.

Markus Freitag and Yaser Al-Onaizan. 2017. Beam search strategies for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 56–60, Vancouver. Association for Computational Linguistics.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *ArXiv*, abs/2312.10997.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Yizheng Huang and Jimmy X. Huang. 2024. A survey on retrieval-augmented text generation for large language models. *ArXiv*, abs/2404.10981.

Bernd Huber, Ghazal Fazelnia, Andreas Damianou, Sebastian Peleato, Max Lefarov, Praveen Ravichandran, Marco De Nadai, Mounia Lalmas-Roellke, and Paul N. Bennett. 2025. Embedding-to-prefix: Parameter-efficient personalization for pre-trained large language models. *Preprint*, arXiv:2505.17051.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*, 2022.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 39–48, New York, NY, USA. Association for Computing Machinery.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Vladimir Karpukhin, Naman Goyal, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.

Xiaopeng Li, Pengyue Jia, Derong Xu, Yi Wen, Yingyi Zhang, Wenlin Zhang, Wanyu Wang, Yichao Wang, Zhaocheng Du, Xiangyang Li, Yong Liu, Huifeng Guo, Ruiming Tang, and Xiangyu Zhao. 2025a. A survey of personalization: From rag to agent. *Preprint*, arXiv:2504.10147.

Zongxi Li, Zijian Wang, Weiming Wang, Kevin Hung, Haoran Xie, and Fu Lee Wang. 2025b. Retrieval-augmented generation for educational application: A systematic survey. *Computers and Education: Artificial Intelligence*, 8:100417.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. *Preprint*, arXiv:1711.05101.

Leland McInnes, John Healy, and Sean Astels. 2017. hdbscan: Hierarchical density based clustering. *Journal of Open Source Software (JOSS)*, 2(11).

Shi Na, Liu Xumin, and Guan Yong. 2010. Research on k-means clustering algorithm: An improved k-means clustering algorithm. In *2010 Third International Symposium on Intelligent Information Technology and Security Informatics*, pages 63–67, Jian, China.

Lin Ning, Luyang Liu, Jiaxing Wu, Neo Wu, Devora Berlowitz, Sushant Prakash, Bradley Green, Shawn O'Banion, and Jun Xie. 2025. User-llm: Efficient llm contextualization with user embeddings. In *Companion Proceedings of the ACM on Web Conference 2025*, WWW '25, page 1219–1223, New York, NY, USA. Association for Computing Machinery.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, and 1 others. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Yilun Qiu, Tianhao Shi, Xiaoyan Zhao, Fengbin Zhu, Yang Zhang, and Fuli Feng. 2025. Latent inter-user difference modeling for LLM personalization. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 10610–10628, Suzhou, China. Association for Computational Linguistics.

Stephen Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1995. Okapi at trec-3. In *In Proceedings of the Third Text REtrieval Conference*, pages 109–126, Gaithersburg, MD: NIST. TREC-3.

Peter J. Rousseeuw. 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65.

Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. 2025. A systematic survey of prompt engineering in large language models: Techniques and applications. *Preprint*, arXiv:2402.07927.

Alireza Salemi, Surya Kallumadi, and Hamed Zamani. 2024a. Optimization methods for personalizing large language models through retrieval augmentation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 752–762, New York, NY, USA. Association for Computing Machinery.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. 2024b. LaMP: When large language models meet personalization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7370–7392, Bangkok, Thailand. Association for Computational Linguistics.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. ColBERTv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of*

the Association for Computational Linguistics: Human Language Technologies, pages 3715–3734, Seattle, United States. Association for Computational Linguistics.

Kiriakos Sgardelis, Dionisis Margaris, Dimitris Spiliotopoulos, and Costas Vassilakis. 2025. An evaluation review of user similarity metrics in sparse collaborative filtering datasets. *International Journal of Data Science and Analytics*, 20:6665–6693.

Chenglei Shen, Xiao Zhang, Teng Shi, Changshuo Zhang, Guofu Xie, and Jun Xu. 2024. A survey of controllable learning: Methods and applications in information retrieval. *ArXiv*, abs/2407.06083.

Teng Shi, Zihua Si, Jun Xu, Xiao Zhang, Xiaoxue Zang, Kai Zheng, Dewei Leng, Yanan Niu, and Yang Song. 2024. Unisar: Modeling user transition behaviors between search and recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 1029–1039, New York, NY, USA. Association for Computing Machinery.

Teng Shi, Jun Xu, Xiao Zhang, Xiaoxue Zang, Kai Zheng, Yang Song, and Han Li. 2025. Retrieval augmented generation with collaborative filtering for personalized text generation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '25, page 1294–1304, New York, NY, USA. Association for Computing Machinery.

Shamane Siriwardhana, Rivindu Weerasekera, Elliott Wen, Tharindu Kaluarachchi, Rajib Rana, and Suranga Nanayakkara. 2023. Improving the domain adaptation of retrieval augmented generation (RAG) models for open domain question answering. *Transactions of the Association for Computational Linguistics*, 11:1–17.

Jiakai Tang, Sunhao Dai, Teng Shi, Jun Xu, Xu Chen, Wen Chen, Wu Jian, and Yuning Jiang. 2025. Think before recommend: Unleashing the latent reasoning power for sequential recommendation. *ArXiv*, abs/2503.22675.

Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 165–174, New York, NY, USA. Association for Computing Machinery.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 641–649, New York, NY, USA. Association for Computing Machinery.

Haoran Xin, Ying Sun, Chao Wang, and Hui Xiong. 2025. Llmcdsr: Enhancing cross-domain sequential recommendation with large language models. *ACM Transactions on Information Systems*, 43(5).

Fangyuan Xu, Weijia Shi, and Eunsol Choi. 2023. Recomp: Improving retrieval-augmented lms with compression and selective augmentation. *ArXiv*, abs/2310.04408.

Yiyan Xu, Jinghao Zhang, Alireza Salemi, Xinting Hu, Wenjie Wang, Fuli Feng, Hamed Zamani, Xiangnan He, and Tat-Seng Chua. 2025. Personalized generation in large model era: A survey. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 24607–24649, Vienna, Austria. Association for Computational Linguistics.

Hong-Jian Xue, Xinyu Dai, Jianbing Zhang, Shujian Huang, and Jiajun Chen. 2017. Deep matrix factorization models for recommender systems. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 3203–3209.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 43 others. 2024. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

Peiru Yang, Xintian Li, Zhiyang Hu, Jiapeng Wang, Jinhua Yin, Huili Wang, Lizhi He, Shuai Yang, Shangguang Wang, Yongfeng Huang, and Tao Qi. 2025. Heterag: A heterogeneous retrieval-augmented generation framework with decoupled knowledge representations. *Preprint*, arXiv:2504.10529.

Hamed Zamani, Fernando Diaz, Mostafa Dehghani, Donald Metzler, and Michael Bendersky. 2022. Retrieval-enhanced machine learning. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 2875–2886, New York, NY, USA. Association for Computing Machinery.

Saber Zerhoudi and Michael Granitzer. 2024. Personarag: Enhancing retrieval-augmented generation systems with user-centric agents. *Preprint*, arXiv:2407.09394.

Changshuo Zhang, Teng Shi, Xiao Zhang, Yanping Zheng, Ruobing Xie, Qi Liu, Jun Xu, and Jirong Wen. 2024. Qagcf: Graph collaborative filtering for q&a recommendation. *ArXiv*, abs/2406.04828.

Changshuo Zhang, Xiao Zhang, Teng Shi, Jun Xu, and Jirong Wen. 2025. Test-time alignment for tracking user interest shifts in sequential recommendation. *ArXiv*, abs/2504.01489.

Yangxiao Zhang. 2025. A retrieval-augmented generation framework with retriever and generator modules for enhancing factual consistency. *Applied and Computational Engineering*, 166:149–155.

Yaochen Zhu, Chao Wan, Harald Steck, Dawen Liang, Yesu Feng, Nathan Kallus, and Jundong Li. 2025. Collaborative retrieval for large language model-based conversational recommender systems. In *Proceedings of the ACM on Web Conference 2025*, WWW '25, page 3323–3334, New York, NY, USA. Association for Computing Machinery.

Thomas P Zollo, Andrew Wei Tung Siah, Naimeng Ye, Ang Li, and Hongseok Namkoong. 2025. Personal-LLM: Tailoring LLMs to individual preferences. In *The Thirteenth International Conference on Learning Representations*.

A Prompts Used in ClusterRAG

This subsection presents the prompt templates employed during generation for the remaining LaMP tasks. As already stated, each prompt contains an instruction, input (query), and profile. We provide the templates below. In the templates, {*Movie description*} and {*Movie tags*}, {*Review*}, {*Article*}, {*Paper abstract*}, and {*Tweet*} represent user input for the corresponding LaMP tasks, while the rest of the italicized text represent user profile entries. The remaining text is the instruction guiding an LLM to generate the intended output.

LaMP-2 (Personalized Movie Tagging) Prompt Template: Given the user previous movie tag pairs: The tag for the movie description: <*Movie_1_Description*> is <Tag_1>, the tag for the movie description: <*Movie_2_Description*> is <*Tag_2*>, . . . , the tag for the movie description: <*Movie_M_Description*> is <*Tag_M*>, which tag does the movie description: {*Movie description*} relate to among the following tags? Just answer with the tag name without further explanation. Movie tags: {*Movie tags*}

LaMP-3 (Personalized Product Rating) Prompt Template: Given the user previous review-score pairs: <*Score_1*> is the score for <*Review_1 Text*>, <*Score_2*> is the score for <*Review_2 Text*>, . . . , <*Score_M*> is the score for <*Review_M Text*>. What is the score of the following review on a scale of 1 to 5? Just answer with 1, 2, 3, 4, or 5 without further explanation. Review: {*Review*}

LaMP-4 (Personalized News Headline Generation) Prompt Template: Given the user's previous article-headline pairs: <*Headline_1*> is the title for <*Article_1_Text*>, <*Headline_2*> is the title for <*Article_2_Text*>, . . . , <*Headline_M*> is the title for <*Article_M_Text*>. Generate a headline for the following article. Article: {*Article*}

LaMP-5 (Personalized Scholarly Title Generation) Prompt Template: Given the user's previous abstract-title pairs: <*Title_1*> is a title for <*Abstract_1_Text*>, <*Title_2*> is a title for <*Abstract_2_Text*>, . . . , <*Title_M*> is a title for <*Abstract_M_Text*>. Generate a title for the following abstract of a paper. Abstract:{*Paper abstract*}

LaMP-7 (Personalized Tweet Paraphrasing) Prompt Template: Given the user's previous tweets: <*Tweet_1*>, <*Tweet_2*>, . . . , <*Tweet_M*>. Paraphrase the following tweet without any explanation before or after it following the user's tweeting patterns. Tweet: {*Tweet*}

B Detailed Dataset Statistics and Licensing Information

Table 6 below provides detailed statistics of the LaMP benchmark based on the time-based split, as stated in Section 5.1. We additionally summarize the licensing information and terms of use for each LaMP task considered in this study as follows:

1. Personalized Citation Identification (LaMP-1): CC BYNC-SA 4.0.
2. Personalized Movie Tagging (LaMP-2): Educational or academic research, NON COMMERCIAL USE.

3. Personalized Product Rating (LaMP-3): CC BY-NC-SA 4.0.

4. Personalized news Headline Generation (LaMP-4): CC BY-NC-SA 4.0.

5. Personalized Scholarly Title Generation (LaMP-5): CC BY-NC-SA 4.0.

6. Personalized Tweet Paraphrasing (LaMP-7): CC BYNC-SA 4.0.

C Task Descriptions

The LaMP benchmark contains English and text-only data. The documents used in each LaMP task do not contain personally identifiable information that could otherwise compromise privacy issues. Below, we provide detailed descriptions of each LaMP task included in our evaluation, outlining the task objectives, input-output formulations, and the specific aspects of personalization each task is designed to assess.

LaMP-1: Personalized Citation Identification. This task frames citation recommendation as a binary classification task and assesses the ability of a language model to identify user preferences for citations. Specifically, if the user u writes a paper x, a language model must determine which of two given candidate papers (a or b) u will cite in x. The profile of each user encompasses all the papers they have authored. Only the title and abstract of each paper are retained in the user's profile; it uses scientific papers.

LaMP-2: Personalized Movie Tagging. This task recasts movie tagging as a multi-class classification task. Given a movie description x and a user's historical movie-tag pairs, a language model must predict one of 15 tags for x. The movie tags are: sci-fi, based on a book, comedy, action, twist ending, dystopia, dark comedy, classic, psychology, fantasy, romance, thought-provoking, social commentary, violence, and true story.

LaMP-3: Personalized Product Rating. LaMP-3 is also framed as a multi-class classification task. In particular, given the user u's historical review and rating pairs of products and an input review x, the model must predict an integer rating (from 1 to 5) of the review.

LaMP-4: Personalized News Headline Generation. This is a generative task that evaluates the ability of an LLM to capture the stylistic patterns of an author u by requiring it to generate a headline for an input news article, x, given a user profile of the authors' historical article-title pairs.

LaMP-5: Personalized Scholarly Title Generation. LaMP-5 is another generative task that requires a generative model to generate titles for an input article x, given a user profile of historical article-title pairs for an author. It is similar to LaMP-4, but it uses different corpus domain, scientific papers (similar to LaMP-1).

LaMP-7: Personalized Tweet Paraphrasing. This is framed as a generative personalized tweet paraphrasing task, which requires an LLM to generate a tweet in the style of a user u given an input tweet x, and a user profile of historical tweets by the user.

D Hyperparameter Tuning Grid

Table 7 summarizes the hyperparameter search space explored during model selection, detailing the ranges and candidate values used to tune ClusterRAG across optimization, training, and decoding configurations. As described in Section 5.1, the optimal hyperparameters were identified via grid search with early stopping applied on LaMP-1 and LaMP-7, representing classification and generative tasks, respectively.

E Impact of Similar User Size and Profile Size

This section investigates the sensitivity of ClusterRAG to two key design parameters that govern collaborative context construction: the number of similar users (k) incorporated during retrieval and the number of profile documents (m) selected per user. By systematically varying these parameters, we analyze how the breadth of collaborative signals and the depth of user profile information affect model performance across different LaMP tasks.

Impact of the Number of Similar Users (k). Table 8 shows that incorporating a small number of similar users consistently improves ClusterRAG performance across all LaMP tasks. Performance generally increases as k grows from 1 to 3, where most metrics achieve their peak or near-peak values, indicating that a limited set of highly simi-

Task	#users	#train	#dev	#test	Input Length	Output Length	#Profile Size	#classes
LaMP-1	6542	6542	1500	1500	51.43 ± 5.70	–	84.15 ± 47.54	2
LaMP-2	929	5073	1410	1557	92.39 ± 21.95	–	86.76 ± 189.52	15
LaMP-3	20000	20000	2500	2500	128.18 ± 146.25	–	185.40 ± 129.30	5
LaMP-4	1643	12500	1500	1800	29.97 ± 12.09	10.07 ± 3.10	204.59 ± 250.75	–
LaMP-5	14682	14682	1500	1500	162.34 ± 65.63	9.71 ± 3.21	87.88 ± 53.63	–
LaMP-7	13437	13437	1498	1500	29.72 ± 7.01	16.96 ± 5.67	15.71 ± 14.86	–

Table 6: Detailed statistics of the LaMP benchmark with time-based data split.

Hyperparameter	Tested values
Learning rate	5×10^{-5}, 3×10^{-3}, 10^{-3}, 10^{-4}
Weight decay	5×10^{-6}, 10^{-4}, 10^{-3}
Warm-up ratio	0.05 to 0.10
Batch size	8, 16, 32, 64
Epochs	10, 20 30, 50, 70, 100
Max seq length	64, 128, 256, 512
Beam size	$1 - 6$
γ	$0.1 - 0.9$
k	$1 - 5$
m	$1 - 12$
$L_{\max}$	64, 128, 256, 512, 1024
$\lvert\bar{y}\rvert$	32, 64, 128, 256, 512

Table 7: Hyperparameter tuning grid used for training and optimizing ClusterRAG.

lar users provides the most informative collaborative signals. Beyond $k = 3$, gains saturate or slightly decline, suggesting that adding more users introduces weaker or noisier preferences that dilute personalization benefits. This trend highlights the importance of selectively leveraging collaborative information rather than aggregating large numbers of loosely related users.

Impact of the Number of Retrieved Profile Documents (m). As shown in Table 9, increasing the number of retrieved profile documents leads to steady performance improvements up to a moderate range ($m \approx 6$–7), after which the gains plateau. Larger profile sizes consistently reduce prediction error (MAE/RMSE) and improve generation quality (ROUGE scores), reflecting richer contextual grounding. At the same time, marginal benefits diminish for larger m, indicating that excessively long profiles provide limited additional signal while increasing prompt complexity. Overall, these results suggest that ClusterRAG is robust to the choice of m and performs best when balancing sufficient contextual coverage with prompt efficiency.

F Cluster Cohesion for Collaborative User Retrieval

To evaluate whether ClusterRAG forms meaningful user clusters for collaborative filtering, we measure the Silhouette score (Rousseeuw, 1987) of user clusters produced by HDBSCAN and k-means and report results in Table 10. The Silhouette score is an internal clustering metric that jointly captures intra-cluster cohesion and inter-cluster separation, with values ranging from -1 to 1, where higher scores indicate better-defined clusters.

As shown in Table 10, ClusterRAG consistently achieves higher Silhouette scores with HDBSCAN than with k-means, whose scores remain below 0.5 across all tasks. In high-dimensional embedding spaces, moderately positive Silhouette scores are common and still reflect meaningful structure. Therefore, the scores above 0.5 obtained with HDBSCAN indicate that ClusterRAG learns cohesive and well-separated user clusters, enabling the retrieval of more similar users for collaborative filtering in personalized RAG. These findings further corroborate the superior performance of ClusterRAG when combined with HDBSCAN.

G Case Study

We randomly sample a case from **LaMP_2 (Personalized Movie Tagging)** in Table 11 to illustrate the effectiveness of ClusterRAG in leveraging both target and similar user profiles for personalized generation. In this task, the *User Query* corresponds to a movie description, and the objective is to generate an appropriate movie tag based on this description and the user's historical tagging behavior. Due to space constraints, we include three profile documents per user. The target user's historical tags are *twist ending* and *action*, while the similar user's historical tags include *true story*, *action*, and *violence*. The gold label for the given query is *violence*.

When we use a non-personalized prompt or the target user profile only, ClusterRAG incorrectly pre-

k value	LaMP-1		LaMP-2		LaMP-3		LaMP-4		LaMP-5		LaMP-7	
	Acc.↑	F1↑	Acc.↑	F1↑	MAE↓	RMSE↓	R-1↑	R-L↑	R-1↑	R-L↑	R-1↑	R-L↑
1	0.690	0.690	0.661	0.620	0.270	0.594	0.190	0.176	0.490	0.440	0.521	**0.470**
2	0.692	0.697	0.665	0.631	**0.269**	**0.582**	0.193	0.179	0.496	0.444	0.524	0.469
3	**0.700**	**0.700**	**0.668**	**0.642**	0.270	0.595	**0.196**	**0.180**	0.495	**0.445**	**0.528**	0.468
4	0.690	0.688	0.660	0.621	0.272	0.595	0.192	0.178	**0.497**	0.441	0.523	0.469
5	0.689	0.690	0.658	0.619	0.273	0.596	0.191	0.177	0.492	0.438	0.521	0.467

Table 8: Effect of the number of similar users (k) on ClusterRAG performance in hybrid mode across LaMP tasks.

m value	LaMP-1		LaMP-2		LaMP-3		LaMP-4		LaMP-5		LaMP-7	
	Acc.↑	F1↑	Acc.↑	F1↑	MAE↓	RMSE↓	R-1↑	R-L↑	R-1↑	R-L↑	R-1↑	R-L↑
1	0.674	0.673	0.648	0.612	0.289	0.608	0.182	0.166	0.480	0.431	0.515	0.464
2	0.690	0.690	0.661	0.620	0.270	0.594	0.190	0.176	0.490	0.440	0.521	0.470
3	0.691	0.691	0.665	0.632	0.269	0.591	0.191	0.178	0.501	0.465	0.528	0.474
4	0.693	0.692	0.669	0.639	0.268	0.590	0.196	0.179	0.510	0.471	0.530	0.473
5	0.695	0.695	**0.675**	**0.650**	0.267	0.586	0.198	0.180	0.512	0.473	0.533	0.475
6	**0.707**	**0.707**	0.673	0.649	0.262	0.584	**0.200**	**0.182**	**0.514**	**0.479**	0.534	0.476
7	0.703	0.700	0.672	0.647	**0.260**	**0.581**	0.199	0.180	0.511	0.474	**0.538**	**0.481**
8	0.704	0.701	0.670	0.645	0.263	0.583	0.197	0.178	0.509	0.472	0.536	0.478
9	0.707	0.701	0.671	0.643	0.263	0.585	0.196	0.177	0.511	0.474	0.534	0.476
10	0.706	0.705	0.671	0.644	0.261	0.582	0.191	0.174	0.508	0.471	0.533	0.476
11	0.701	0.700	0.669	0.641	0.263	0.584	0.191	0.173	0.504	0.469	0.534	0.475
12	0.700	0.700	0.670	0.664	0.264	0.584	0.192	0.174	0.504	0.470	0.532	0.473

Table 9: Effect of the number of retrieved profile documents (m) on ClusterRAG performance in hybrid mode across LaMP tasks.

Task	HDBSCAN Score	k-means Score
LaMP-1	0.601	0.389
LaMP-2	0.535	0.326
LaMP-3	0.551	0.328
LaMP-4	0.570	0.274
LaMP-5	0.562	0.347
LaMP-7	0.537	0.323

Table 10: Silhouette scores of user clusters produced by ClusterRAG using HDBSCAN and k-means on the LaMP benchmark.

dicts the movie tag as *action*, driven by its higher frequency in the current user's history. However, when similar-user information is incorporated via hybrid profile retrieval, the model correctly predicts *violence*, as the movie tag. This occurs because the top-ranked retrieved profile, originating from a similar user, emphasizes personal and retaliatory violence that closely aligns with the query, whereas the lower-ranked *action*-tagged profile reflects more stylized narratives less relevant to the description.

H AI Assistance Usage

In this work, ChatGPT has been used solely as a writing assistant. Specifically, draft passages were provided to the tool for paraphrasing and language refinement, after which we manually reviewed, edited, and finalized the text.

User Query:

Movie description: When the Davison family comes under attack during their wedding anniversary getaway, the gang of mysterious killers soon learns that one of their victims harbors a secret talent for fighting back.

Gold Output: Violence

Target User Profile:

(1) **Movie tag**: "twist ending", **Movie description**: "Soon after his insufferably arrogant father wins the Nobel Prize for chemistry, Barkley Michaelson is kidnapped by Thaddeus James, a young genius who claims to be Barkley's illegitimate half-brother. Motivated not so much by money as revenge, Thaddeus tries to convince Barkley to help him carry out a multimillion-dollar extortion plot against their patriarch."

(2) ***Movie tag**: "action", **Movie description**: "The Bride unwaveringly continues on her roaring rampage of revenge against the band of assassins who had tried to kill her and her unborn child. She visits each of her former associates one-by-one, checking off the victims on her Death List Five until there's nothing left to do . . . but kill Bill."*

(3) **Movie tag**: "action", **Movie description**: "NYPD cop John McClane's plan to reconcile with his estranged wife is thrown for a serious loop when, minutes after he arrives at her office, the entire building is overtaken by a group of terrorists. With little help from the LAPD, wisecracking McClane sets out to single-handedly rescue the hostages and bring the bad guys down."

Similar User Profile:

(1) **Movie tag**: "true story", **Movie description**: "The mostly true story of the legendary 'worst director of all time', who, with the help of his strange friends, filmed countless B-movies without ever becoming famous or successful."

(2) **Movie tag**: "action", **Movie description**: "Liu Jian, an elite Chinese police officer, comes to Paris to arrest a Chinese drug lord. When Jian is betrayed by a French officer and framed for murder, he must go into hiding and find new allies."

(3) ***Movie tag**: "violence", **Movie description**: "An elderly ex-serviceman and widower looks to avenge his best friend's murder by doling out his own form of justice."*

Top Ranked Documents: Doc # (3) from similar user then doc # (2) from current user.

Personalized Prompt:

Given the user previous movie tag pairs: The tag for the movie description: "*An elderly ex-serviceman and widower looks to avenge his best friend's murder by doling out his own form of justice*" is "*violence*", and the tag for the movie description: "*The Bride unwaveringly continues on her roaring rampage of revenge against the band of assassins who had tried to kill her and her unborn child. She visits each of her former associates one-by-one, checking off the victims on her Death List Five until there's nothing left to do . . . but kill Bill.*" is "*action*", which tag does the movie description: "*When the Davison family comes under attack during their wedding anniversary getaway, the gang of mysterious killers soon learns that one of their victims harbors a secret talent for fighting back.*" relate to among the following tags? Just answer with the tag name without further explanation. Movie tags: "*sci-fi, based on a book, comedy, action, twist ending, dystopia, dark comedy, classic, psychology, fantasy, romance, thought-provoking, social commentary, violence, and true story*"

Generated Output: Violence

Table 11: A case study illustrating how ClusterRAG leverages user profile and similar-user information for personalized RAG.