Improving Retrieval Accuracy Using Clustering to Personalize Large Language Models

MASHRUR RASHIK* and AMIT SARKER*, University of Massachusetts Amherst, USA

In this project, we work on improving how Large Language Models (LLMs) personalize their responses. Our main goal is to find out an effective way to retrieve user profile information for LLM personalization. In this project, we contain our focus on the retrieval aspect. We utilize Flan-T5-base as the LLM to generate our output. We use baseline models like BM25 and more sophisticated approaches like topic-modeling and contriever reranking to retrieve profile information. Finally, we propose a novel retrieval strategy based on clustering. We test these models with the Citation Identification and News Categorization data from the LaMP benchmark. Our findings suggest that topic-modeling can be an alternative to generate shorter inputs for LLMs. In addition, we found that our clustering-based approach outperforms the rest of the retrieval strategies. We also discuss the implications of our results which includes scaling retrieval time using reranking and clustering models, inherent task complexity in the LAMP benchmark, going beyond similarity measures for ranking, and opportunities for self-supervised learning-to-rank models.

Additional Key Words and Phrases: Personalization, LLM, Rerank

ACM Reference Format:
Mashrur Rashik and Amit Sarker. 2024. Improving Retrieval Accuracy Using Clustering to Personalize Large Language Models. 1, 1 (May 2024), 5 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

The field of information retrieval is evolving, and the integration of personalization within Large Language Models (LLMs) has emerged as an important area of research. This project delves into this domain, addressing two research questions (RQs) that tackle the challenges of personalizing LLM outputs. RQ1 focuses on the extraction of relevant information from user profiles, and RQ2 aims to optimize LLM prompts for large datasets.

- **RQ1**: How to extract relevant information from a user profile to personalize the output of an LLM?
- **RQ2**: How can the prompt to personalize an LLM output be optimized when the relevant user information is large?

Our approach to **RQ1** involves a clustering-based IR method. This approach recognizes that efficient and accurate user profile data extraction is crucial for customized LLM outputs. The clustering technique allows us to handle diverse and extensive user data effectively. This enhances the personalization aspect of the model. We hypothesize that this approach will lead to more relevant and user-centric information retrieval compared to traditional methods.

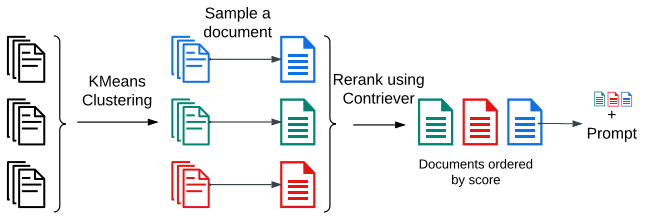

Fig. 1. This figure shows our clustering approach for retrieval. In our approach, we first cluster the documents using KMeans. We then sample a data point or document from a cluster. Finally, we rerank the documents using contriever.

For **RQ2**, we explore topic modeling as a means to optimize prompts for LLMs in the middle of extensive user data. The complexity and volume of information in large user profiles are extensive. Therefore, concise and focused prompts are hypothesized to improve the LLM's efficiency and output accuracy. This part of the research is important for ensuring that the LLM can handle large amounts of data without compromising the relevance and precision of the information retrieved.

In this project, we used various models and datasets to investigate advanced retrieval techniques. We used the Flan-T5-base language model (LM), trained and evaluated through Hugging Face, and various retrieval models, including a baseline BM25 model, a topic-model based retrieval model, a reranking model using contriever, and a novel clustering-based retrieval model. The experiments were conducted using the Personalized Citation Identification and News Categorization datasets from the LAMP benchmark. The retrieval models processed user profile data and queries differently: BM25 used a concatenation of the user's profile data with the query, the topic-model approach used the topics from the profile data, the reranking model used contriever to rerank BM25 results, and the clustering approach combined sampling from clusters with contriever reranking.

In the results, we found that the reranking model and BM25 performed best with an accuracy of 0.67 using user-based separation with k=4 for citation identification. The topic-model had a slightly lower accuracy. In news categorization, the reranking model achieved the highest accuracy (0.8) for both k=1 and k=4 under user-based separation. However, the clustering model outperforms the other models without profile tuning. The clustering model reached an accuracy of 0.69 for k=4 in user-based separation in citation identification. In news categorization, it achieved an accuracy of 0.8 for k=4 under the same condition. These results indicate the effectiveness of the clustering approach. The clustering model's performance in news categorization was comparable to the best-performing models from Salemi et al. [13].

*Both authors contributed equally to this research.

Authors' address: Mashrur Rashik, mrashik@umass.edu; Amit Sarker, asarker@umass.edu, University of Massachusetts Amherst, P.O. Box 1212, Amherst, MA, USA, 01007.

2 RELATED WORK

The recent advancements in natural language processing (NLP) have increasingly emphasized the importance of personalization in language models. The LaMP paper [13] introduces the LaMP benchmark that is designed to evaluate and train large language models (LLMs) for personalized output generation. This benchmark is critical in understanding how user-specific data can be integrated into model training to produce more relevant and context-aware responses. This concept is central to our project.

Our project aligns with recent advancements in personalized information retrieval and recommendation systems. Incorporating implicit user profiles for personalized chatbot interactions demonstrates the significance of user-specific data in enhancing response relevance [11]. We used this principle in our personalized citation and news categorization tasks. The study [11] introduces the IMPChat model, which learns a user's personalized language style and preferences to select contextually relevant responses. Furthermore, Jawaheer et al. [6] outlines a framework for utilizing explicit and implicit user feedback. This approach is relevant to our method of leveraging user interaction data for personalization. The paper reviews state-of-the-art techniques to improve user feedback in recommender systems and formulates challenges for future research in enhancing the performance of recommender systems through better user feedback. In [10], the authors explored different ways of giving prompts or instructions to enhance the recommendations made by Large Language Models (LLMs). This approach offers valuable insights into our method, particularly when working with LLMs. The paper emphasizes the importance of diverse prompts and input augmentation techniques to enhance LLM capabilities in recommendation systems.

The idea of group-based personalization, introduced in this study [1], aligns well with our method. It offers useful information on how to effectively group users together for information retrieval purposes. It introduces a model that uses Latent Dirichlet Allocation (LDA) [2] for topic modeling and K-means clustering [4] for grouping users. The similarity between users is assessed using symmetric Kullback–Leibler divergence. This approach addresses data sparsity issues, enhances the relevance of search results, and alleviates privacy concerns by using group profiles instead of individual user profiles. Moreover, the research by Yao et al. [15] highlights the importance of matching word meanings with user interests. The paper shows that by training personal word embeddings on a user's search history, it's possible to greatly improve how personalized a search is. This is done by making sure the meanings of words match the specific interests of each individual user. The approach of our project is further supported by current research, including the progress in cluster-based retrieval outlined by Liu and Croft [8] and the proven effectiveness of pairwise ranking in Large Language Models (LLMs) discussed by Qin et al. [12]. These studies provide a solid foundation for our project's methodology. These studies collectively inform our project's design and implementation, contributing to our understanding of personalization in IR and recommendation systems. [8] demonstrates that cluster-based retrieval can achieve significant improvements over document-based retrieval. Moreover, cluster-based retrieval offers a consistent performance across collections of realistic size. The proposed Pairwise Ranking Prompting (PRP) technique in [12] simplifies the task complexity for LLMs and addresses calibration issues found in traditional pointwise and listwise ranking approaches.

3 PROBLEM FORMULATION

In this project, we will evaluate retrieval models to retrieve information from a user's profile and personalize a Large Language Model (LLM). More formally, for a user profile, P, and a query to an LLM q, we want to retrieve $x \in P$ to create $q^{'} = \phi(q, x)$. Here ϕ is a concatenation function. For concatenation and formulation of q, we followed Salemi et. al. [13]. For brevity, we do not discuss them in this report, but we share a Google Drive link that contains our data and models [1]. The objective is to use $q^{'}$ as an input to an LLM, to answer q accurately.

4 EXPERIMENTAL DESIGN

This section will provide an overview of the models, the datasets used to evaluate the models, and the experimental setup.

4.1 Models

For our LLM, we used Flan-T5-base in all our experiments. We used huggingface to train and evaluate our LMs [2]. For the retrieval model, we used a BM25 model as the baseline following Salemi et al. [13]. In addition to BM25, we used a topic-model based retrieval model and a reranking model where we rerank the output of BM25 using contriever [5]. Finally, we introduce our novel clustering-based retrieval model. We discuss each of these models in the sections that follow.

4.1.1 BM25. This is our baseline retrieval model. The input to this model is a subsequence of the query and every datapoint ($x \in P$) from the user's profile. We concatenate the top-k datapoints (x) to q to form $q^{'}$. We used the publicly available implementation of Okapi BM25 on PyPI [3].

4.1.2 Topic-model Based Retrieval. In this model, we first used BM25 to rank the data points from the user's profile. Next, instead of using the entire text to concatenate to q, we concatenate the topics of that data point. To get the topics of the text data, we used BERTtopic [3]. The idea behind using a topic-model is to reduce the length of the queries, where instead of using the entire query, we would use the relevant topics associated with it. Our hypothesis is:

H1. The efficacy of concatenating the topics of a text into the retrieval query would be comparable to that of using the entire text.

4.1.3 Reranking. In this model, we first use BM25 to rank the data points. Then, using contriever, we select the minimum number of data points in the user profile or ten datapoints and rerank them. We use the implementation of contriever on huggingface [4].

[1] https://drive.google.com/drive/folders/1ItwkTrQR97-fFrbtXysJXrV54A2tFZDy?usp=sharing
[2] https://huggingface.co/docs/transformers/model_doc/flan-t5
[3] https://pypi.org/project/rank-bm25/
[4] https://huggingface.co/facebook/contriever

Table 1. This table shows the results from our experiment. In this table, we report the accuracy, where a higher score is better.

Dataset	K = 1			K = 4		
	BM25	Topic Modeling	Reranking (BM25 + Contriever)	BM25	Topic Modeling	Reranking (BM25 + Contriever)
Citation Identification (User Separation)	0.65	0.63	0.65	0.67	0.64	**0.67**
Citation Identification (Time Separation)	0.65	0.63	0.65	0.67	0.64	**0.67**
News Categorization (User Separation)	0.78	0.78	0.8	0.79	0.78	**0.8**
News Categorization (Time Separation)	0.78	0.78	**0.8**	0.79	0.78	0.79

4.1.4 Clustering. This is our novel retrieval approach (see Figure 1). This is inspired by prior works on clustering [9]. In prior works, researchers used scoring methods to rank the clusters. However, in our approach, we don't directly rank the clusters. Instead, we first sample a single data point from a cluster and then rerank them using contriever. This is based on the assumption that the cluster elements are similar to each other. Our hypothesis is:

H2. The clustering-based retrieval technique outperforms other techniques for retrieval.

For clustering, we used the implementation of K-Means from Scikit-Learn [5]. We used ten or six for the number of clusters, depending on the dataset. After finding the clusters, we sample a single data point from each cluster and then rerank them using contriever.

4.2 Datasets

In this project, we used the Personalized Citation Identification and News Categorization datasets from the LAMP benchmark [6].

4.3 Experimental Setup

We ran our experiments on Google Colab [7]. We used V100 and T4 to train the LLM. Due to resource and time constraints, we separated the retrieval and LLM models. We ran separate notebooks to prepare the input and target data for training the LLM. This process included creating $q^{'}$ from q using each retrieval technique on both datasets. In our clustering approach, we used an initial cluster size of ten for Personalized Citation Identification and a cluster size of six for Personalized News Categorization. While retrieving relevant profile information, for top-k, we use $k = 1$ and $k = 4$.

For hyperparameters, we followed Salemi et. al. [13] and used a learning rate of $5x10^{-5}$, weight decay of $1x10^{-4}$, warmup ratio of 0.05, and max input and output token size of 512. However, we trained our models for 5 epochs since we observed that after 5 epochs, the validation loss started exceeding the training loss (overfit). The overall training process took approximately five days on three parallel colab notebooks.

[5]https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html
[6]https://lamp-benchmark.github.io/download
[7]https://colab.google/

5 RESULTS

We evaluated our models on the development data and reported the accuracy for each. The results of our experiments using BM25, topic-model, and reranking are summarized in Table 1. For user-based separation in citation identification, using $k = 1$, the reranking model got an accuracy of 0.65, whereas BM25 and topic-model had an accuracy of 0.65 and 0.63, respectively. From Salemi et al. [13], we found that the best-performing model using $k = 1$ is contriever which gets an accuracy of 0.69. For $k = 4$, the reranking model got an accuracy of 0.67, whereas BM25 and topic-model had an accuracy of 0.67 and 0.64, respectively. For time-based separation, the ranking model had an accuracy of 0.65 for $k = 1$ and 0.67 for $k = 4$. BM25 got a similar score for time-based separation. The results show that, for citation identification, BM25 and the reranking model achieved the best performance using $k = 4$ for both user-based and time-based separation of the data.

For news categorization and user-based separation, the reranking model got an accuracy of 0.8 for $k = 1$ and 0.8 for $k = 4$. BM25 had a slightly lower accuracy of 0.78 for $k = 1$ and 0.79 for $k = 4$. The topic-model got a similar accuracy of 0.78 for both $k = 1$ and $k = 4$. For time-based separation, the reranking model got an accuracy of 0.8 for $k = 1$ and 0.79 for $k = 4$. This is the first instance where a model using $k = 1$ outperforms $k = 4$. BM25 got an accuracy of 0.78 for $k = 1$ and 0.79 for $k = 4$. Finally, topic-model got an accuracy of 0.78 for both values of k. In all our experiments, we found that topic-model gets similar performance for both values of k, and the performance varies by at most 3% in comparison to BM25 and 4% compared to reranking. Therefore, our $H1$ is partially supported by the results.

The results for our clustering technique are summarized in Table 2. For citation identification and user-based separation, the clustering model got an accuracy of 0.64 for $k = 1$ and 0.69 for $k = 4$. For time-based separation, the clustering model got an accuracy of 0.63 for $k = 1$ and 0.7 for $k = 4$. The results show that, for news categorization, the cluster model performs best for $k = 4$. From Salemi et al. [13], we found that the best-performing models use a tuned profile and have an accuracy of 0.73 and 0.71, respectively, for the user and time-based separation on citation identification. The

clustering model, without any profile tuning, achieves a comparable accuracy for the citation identification data.

For news categorization using the clustering model, for user based-separation, we found an accuracy of 0.79 and 0.8, respectively, for $k = 1$ and $k = 4$. For time-based separation, we found an accuracy of 0.79 and 0.804, respectively, for $k = 1$ and $k = 4$. From Salemi et al. [13], we found that the best-performing models use a tuned profile and have an accuracy of 0.76 and 0.806, respectively, for the user and time-based separation on news categorization. Similar to citation identification, without any profile tuning, our clustering approach achieves comparable accuracy to the best-performing models from Salemi et al. [13] [8]. Finally, we found that the clustering model outperforms all the models from Table 1. Therefore, our *H2* is supported.

Table 2. This table shows the results for our clustering models. In this table, we report the accuracy, where a higher score is better.

Dataset	K = 1	K = 4
Citation Identification (User Separation)	0.64	**0.69**
Citation Identification (Time Separation)	0.63	**0.7**
News Categorization (User Separation)	0.79	**0.8**
News Categorization (Time Separation)	0.79	**0.804**

6 DISCUSSIONS AND FUTURE WORK

This section discusses the implications of our results and opportunities for future work.

6.1 Scaling retrieval time using reranking and clustering models

Our experiment's results showed us that reranking with contriever can reach comparable accuracy to employing only contriever. We found that contriever consumes a lot of GPU resources, making it difficult to train with a contriever retriever, even with a high-end V100 GPU. Because reranking has comparable accuracy, it may be more time efficient to rank the profile data first with a faster model like BM25 and subsequently with a superior scorer like contriever. Furthermore, clustering-based models can be a useful alternative to resource-intensive models such as contriever. Our findings suggest that clustering models perform nearly as well as contriever. While we employed KMeans in our experiment, future research could look into the usefulness of other clustering algorithms.

6.2 Inherent task complexities in the LAMP benchmark

We discovered that news categorization has a greater accuracy than citation identification in our studies. However, the citation identification task is a binary classification task, but the news categorization data is a multi-class classification. This could be related to the intrinsic task complexity of identifying citations using only past articles. It may be beneficial to provide extra profile information such as the user's subject of research and publication venues. Future research in this area may look into techniques for generating an implicit user profile from history in order to personalize LLMs. Furthermore, it could be interesting to investigate the impact of various prompts in improving the accuracy of these models.

6.3 Going beyond similarity measures for ranking

In our experiments, we employed a similarity measure to rank the user profile information. However, there are alternative measures, such as informativeness [14], that indicate the depth of a text data's information content. Previous works have effectively employed these measures as a signal for data quality, particularly in conversational systems [7, 14]. Future research could look into how effective these metrics are at ranking profile information for retrieval.

6.4 Self supervised learning-to-rank models

Our findings suggest that self-supervised algorithms like KMeans perform well when it comes to grouping similar profile information together. It will be interesting to see how self-supervised learning algorithms may be used to rank such profile information for personalization in the future.

7 LIMITATIONS

One of our project's primary drawbacks is a lack of resources and time. We were unable to employ the contriever model for ranking due to a lack of adequate GPU resources. Furthermore, because our LLM took a long time to train on the data, our tests were limited to employing a single LLM, Flant-T5-base. However, we would like to put our clustering strategy to the test on larger models. We also acknowledge that our present retrieval techniques may not be ideal for commercial applications due to the time required to generate an output. However, this time can be lowered by employing better GPU resources.

8 CONCLUSION

In this project, we evaluate existing ranking algorithms and suggest two of our own. The first strategy is topic modeling, which tries to reduce the query input size. Our findings suggest that topic modeling retrieval can perform better than existing baselines, although there is still potential for improvement. Then, using clustering to group documents, we present a clustering approach that trades off GPU resources. Our clustering method outperforms all others. Finally, we explore the ramifications of our findings and future work opportunities.

REFERENCES

[1] Sara Abri, Rayan Abri, and Salih Çetin. 2020. Group-based personalization using topical user profile. In *Adjunct Publication of the 28th ACM Conference on User Modeling, Adaptation and Personalization*. 181–186.

[2] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.

[8] By best-performing model, we refer to the best-performing retrieval model for Flan-T5-base

[3] Maarten Grootendorst. 2022. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. *arXiv preprint arXiv:2203.05794* (2022).

[4] John A Hartigan and Manchek A Wong. 1979. Algorithm AS 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)* 28, 1 (1979), 100–108.

[5] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118* (2021).

[6] Gawesh Jawaheer, Peter Weller, and Patty Kostkova. 2014. Modeling user preferences in recommender systems: A classification framework for explicit and implicit user feedback. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 4, 2 (2014), 1–26.

[7] Zhiqiu Jiang, Mashrur Rashik, Kunjal Panchal, Mahmood Jasim, Ali Sarvghad, Pari Riahi, Erica DeWitt, Fey Thurber, and Narges Mahyar. 2023. CommunityBots: Creating and Evaluating A Multi-Agent Chatbot Platform for Public Input Elicitation. *Proceedings of the ACM on Human-Computer Interaction* 7, CSCW1 (2023), 1–32.

[8] Xiaoyong Liu and W Bruce Croft. 2004. Cluster-based retrieval using language models. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*. 186–193.

[9] Xiaoyong Liu and W. Bruce Croft. 2004. Cluster-Based Retrieval Using Language Models. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Sheffield, United Kingdom) *(SIGIR '04)*. Association for Computing Machinery, New York, NY, USA, 186–193. https://doi.org/10.1145/1008992.1009026

[10] Hanjia Lyu, Song Jiang, Hanqing Zeng, Yinglong Xia, and Jiebo Luo. 2023. Llm-rec: Personalized recommendation via prompting large language models. *arXiv preprint arXiv:2307.15780* (2023).

[11] Hongjin Qian, Zhicheng Dou, Yutao Zhu, Yueyuan Ma, and Ji-Rong Wen. 2021. Learning implicit user profile for personalized retrieval-based chatbot. In *proceedings of the 30th ACM international conference on Information & Knowledge Management*. 1467–1477.

[12] Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, Xuanhui Wang, et al. 2023. Large language models are effective text rankers with pairwise ranking prompting. *arXiv preprint arXiv:2306.17563* (2023).

[13] Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. 2023. LaMP: When Large Language Models Meet Personalization. *arXiv preprint arXiv:2304.11406* (2023).

[14] Ziang Xiao, Michelle X. Zhou, Q. Vera Liao, Gloria Mark, Changyan Chi, Wenxi Chen, and Huahai Yang. 2020. Tell Me About Yourself: Using an AI-Powered Chatbot to Conduct Conversational Surveys with Open-Ended Questions. *ACM Trans. Comput.-Hum. Interact.* 27, 3, Article 15 (jun 2020), 37 pages. https://doi.org/10.1145/3381804

[15] Jing Yao, Zhicheng Dou, and Ji-Rong Wen. 2020. Employing personal word embeddings for personalized search. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*. 1359–1368.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009