LaMP-QA: A Benchmark for Personalized Long-form Question Answering

Alireza Salemi and Hamed Zamani
Center for Intelligent Information Retrieval
University of Massachusetts Amherst
{asalemi,zamani}@cs.umass.edu

Data: https://hf.co/datasets/alireza7/LaMP-QA
Code: https://github.com/LaMP-Benchmark/LaMP-QA

Abstract

Personalization is essential for question answering systems that are user-centric. Despite its importance, personalization in answer generation has been relatively underexplored. This is mainly due to lack of resources for training and evaluating personalized question answering systems. We address this gap by introducing LaMP-QA—a benchmark designed for evaluating personalized long-form answer generation. The benchmark covers questions from three major categories: (1) Arts & Entertainment, (2) Lifestyle & Personal Development, and (3) Society & Culture, encompassing over 45 subcategories in total. To assess the quality and potential impact of the LaMP-QA benchmark for personalized question answering, we conduct comprehensive human and automatic evaluations, to compare multiple evaluation strategies for evaluating generated personalized responses and measure their alignment with human preferences. Furthermore, we benchmark a number of non-personalized and personalized approaches based on open-source and proprietary large language models. Our results show that incorporating the personalized context provided leads to up to 39% performance improvements. The benchmark is publicly released to support future research in this area.

1 Introduction

Personalization plays a key role in many applications such as search (Xue et al., 2009), recommendation (Naumov et al., 2019), and generation (Salemi et al., 2024b, 2025b; Xu et al., 2025), as it contributes to improving user satisfaction and trust in the system. For information seeking, personalization is valuable as it enables systems to generate responses that are tailored to the intent, background, and preferences of the user, producing more accurate and user-specific responses. Research on personalized information seeking has predominantly focused on personalized retrieval (Kasela et al., 2024; Guo et al., 2021; Eugene et al., 2013), while the *generation* aspect of the problem remains underexplored. This gap has become increasingly important with the advent of large language models (LLMs) that interact with users through natural language, making personalized text generation a critical area of study. Recently, researchers developed LaMP (Salemi et al., 2024b) and LongLaMP (Kumar et al., 2024) benchmarks for personalized text generation, however, they exclusively focus on content generation tasks, such as email generation based on user's writing style, overlooking *information seeking tasks*. This leaves a critical gap in evaluating how well LLMs can generate tailored responses to users' information needs.

A potential approach for constructing a personalized question answering dataset is human annotation, e.g., where a user poses a question and selects their preferred response from a set of generated responses. This faces two major limitations. First, the user's selected response represents only a single sample from a broader distribution of potentially suitable responses for the user. Since the user does not have access to the full space of all possible responses, their choice may not reflect their true preference. Second, previous work has shown that effective personalization is based on access to the user's historical interactions with the system (Kasela et al., 2024; Salemi and Zamani, 2024). Collecting such history in a single annotation session is challenging, making this method difficult to scale for large and realistic datasets. A more scalable alternative is to use data from forums or community question answering (CQA) platforms. These platforms often feature questions accompanied by detailed post descriptions—serving as the *question narrative*—where users explicitly articulate their specific information needs. Users often ask multiple questions over time, enabling the creation of histories that support personalized modeling and provide rich, time-based interaction data.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 1140–1160
November 4-9, 2025

In addition, other users in the community can respond to the question, and the question asker has the option to select a preferred "accepted" answer. However, relying on a single selected answer for evaluation still has limitations, as users may not see the full range of suitable responses.

An alternative evaluation strategy is to use the question narrative that accompanies the question on these platforms as a personalized question narrative. These question narratives include rich contextual cues, such as the motivation behind the question, the specific aspects the user expects to see addressed, and their primary concerns. Instead of asking users to pick a single preferred response, the question narrative can serve as a personalized evaluation rubric, defining what makes an answer satisfactory. This allows for a more principled, fine-grained evaluation based on alignment with user-specified criteria. This approach goes beyond binary or ordinal preferences, enabling systematic scoring of responses across multiple personalized criteria. This evaluation approach is also grounded in long-standing research on dataset creation in the TREC community (Balog et al., 2010; Shen et al., 2007; Allan, 2003; Lawrie et al., 2024; Buckley et al., 1998). In many TREC tasks, annotators are provided not only with the query but also with a detailed description or narrative that clarifies the underlying intent and contextual nuances of the information need. This additional information ensures more accurate and fair evaluation by helping annotators judge relevance based on the full scope of the user's expectations. Our proposed evaluation method extends this idea to the generation setting with a focus on *personalized narratives*.

This approach results in the Language Model Personalized Question Answering benchmark (LaMP-QA) for training and evaluating long-form personalized answer generation systems. To collect our dataset, we begin with the SE-PQA (Kasela et al., 2024) dataset, designed for personalized retrieval and extracted from the StackExchange website. We filter out questions that do not require personalization, assessed by a capable LLM, Gemini (Gemini-Team, 2024), to make sure usefulness for studying personalization. Next, for the remaining questions, we filter out those where the corresponding question narrative (i.e., the post detail) do not provide details regarding the specific information needs of the user, using the same LLM. This step is crucial, as responses are evaluated based on how well they address the user's specific information needs outlined in narratives. Since question narratives are hidden during generation and used only for evaluation, we filter out questions lacking sufficient detail for a meaningful assessment. The outputs of these steps are sampled and then quality-checked by human annotators to ensure high quality.

Inspired by prior work on personalized search (Kasela et al., 2024) and generation (Salemi et al., 2024b), which leverage a user's historical interactions as the profile, we construct our benchmark by treating the user's current question as the input, their previously asked questions as the user profile, and the key aspects extracted from the question narrative as the evaluation criteria. To evaluate a generated response, we use an LLM to assess how well the response addresses each personalized rubric aspect aspect (Examples in Figure 8 and Figure 9 in Appendix A). This enables us to evaluate how well the response aligns with the user's needs and preferences. We divide the dataset into three categories: (1) Arts & Entertainment, (2) Lifestyle & Personal Development, and (3) Society & Culture, with over 45 subcategories as shown in Figure 5 in Appendix A. Examples of our benchmark are provided in Figure 8 and Figure 9 in Appendix A, while the statistics are detailed in Table 1.

To assess the benchmark's quality for personalization, we run a series of experiments. First, we assess the quality of automatically extracted rubric aspects—used as evaluation rubrics—from the user's question narratives. Human annotators assign an average rating of 4.9 out of 5 on the quality of the extracted aspects, demonstrating the high quality of the aspect extraction process. We compare our proposed personalized, aspect-based evaluation method against two alternative strategies: pairwise comparison and aspect-free scoring (both also based on the question narrative). Our method achieves the highest alignment with human judgment, validating its effectiveness as an evaluation approach. Moreover, an essential characteristic of a high-quality personalized QA dataset is that the included user profile should yield better performance when used with the corresponding user's query compared to either (1) no personalization, or (2) using a profile from another user. Our results confirm this: using the target user's profile improves performance by up to 39% over the non-personalized setting, and by up to 62% compared to using a mismatched profile. These findings show that both the profiles and evaluation rubrics are user-specific and effectively support personaliza-

tion. To establish baselines, we evaluate a range of open-source and proprietary LLMs—including Gemma 2 (Gemma-Team, 2024), Qwen 2.5 (Qwen et al., 2025), and GPT-4o (OpenAI, 2024)—in both personalized (using RAG) and non-personalized settings on the LaMP-QA benchmark. All data and code is publicly released to encourage further research in personalized question answering.[1]

2 Problem Formulation

We consider a scenario where a user u poses a question x_u. Following prior work that incorporates long-term user history as a user profile (Kasela et al., 2024; Salemi et al., 2024b), we assume $P_u = \{p_i\}_{i=1}^{n_u}$, the user profile, consists of n_u user's previously asked questions along with the detailed descriptions of the question written by the user. This information facilitates a better understanding of user preferences. The objective is to use this personalized information alongside the question x_u to generate a personalized response using a QA model M, expressed as $\hat{y}_u = M(x_u, P_u)$. To evaluate the quality of the generated response, we assume access to a set of n_{x_u} personalized aspects $E_{x_u} = \{e_i\}_{i=1}^{n_{x_u}}$ that the user expects to be addressed in response to the question x_u. These aspects are extracted from a personalized question narrative r_{x_u} provided by the user. Importantly, these aspects are used exclusively for evaluation and are not accessible to the model during response generation. Finally, a metric $\mu(x_u, \hat{y}_u, E_{x_u}, r_{x_u})$ quantifies response quality based on the extent to which the expected aspects are covered. Since these aspects are explicitly derived from user-provided requirements, this evaluation framework enables an assessment of how well the generated response is personalized to the user's information needs.

3 The LaMP-QA Benchmark

Personalized text generation has been studied in short- and long-form content generation, such as email writing, review writing, and email title generation, in benchmarks like LaMP (Salemi et al., 2024b) and LongLaMP (Kumar et al., 2024). However, personalization in information-seeking differs fundamentally from these tasks. While personalized content generation focuses on mimicking the user's writing style and preferences when generating text on their behalf, personalization in information-seeking is centered on tailoring the response to align with the user's information needs and preferences. Although datasets for personalized retrieval, as a form of information-seeking, exist (Kasela et al., 2024; Guo et al., 2021; Eugene et al., 2013), to the best of our knowledge, no dataset focuses on answer generation.

To construct the LaMP-QA benchmark, we begin with the SE-PQA dataset (Kasela et al., 2024),[2] which has already collected data from the StackExchange platform for retrieval tasks, to avoid scraping the website again. This website is a community-based question-answering platform where users post their questions. Each post consists of a title—phrased as a question—and a detailed description that clarifies the user's information needs. This structure allows for the formulation of a personalized question-answering task. Specifically, the post title serves as the user's question, while the detailed description outlines the key information necessary for generating an effective response for the user. Since these descriptions are written by the users themselves, they provide direct insight into their expectations, making them a valuable resource for evaluating how well a generated response aligns with their needs. Furthermore, a user's previously asked questions can be leveraged to construct a user profile, capturing their information-seeking behavior and past interactions with the system.

3.1 Data Collection

The LaMP-QA benchmark is created by adapting the SE-PQA dataset, originally designed for personalized retrieval tasks, to a personalized question answering task through the following steps:

Filtering out factoid questions that do not require personalization: Since our objective is to construct a dataset for personalized question answering, we exclude questions that are purely factual—those whose answers remain unchanged regardless of the individual asking them. In other words, we exclude factoid questions that do benefit from personalized user information. To achieve this, we use a capable LLM, employing the prompt illustrated in Figure 6 in Appendix A. This prompt instructs the LLM to identify and label factoid questions that do not require personalization. For the test and validation sets, to ensure high-quality out-

[1] Code is available at: https://github.com/LaMP-Benchmark/LaMP-QA and the data is available at: https://hf.co/datasets/alireza7/LaMP-QA

[2] Open access under cc-by-sa 4.0 license.

Method	Arts & Entertainment			Lifestyle & Personal Development			Society & Culture		
	train	validation	test	train	validation	test	train	validation	test
#Questions (users)	9349	801	767	7370	892	989	7614	810	1074
#Evaluation Aspects	2.7 ± 0.9	4.7 ± 1.2	4.6 ± 1.2	3.1 ± 1.0	5.1 ± 1.1	5.1 ± 1.2	2.9 ± 0.9	4.8 ± 1.1	4.8 ± 1.0
Profile Size	106.7 ± 127.3	129.0 ± 183.7	159.1 ± 203.0	116.6 ± 162.0	98.2 ± 198.6	111.6 ± 220.3	141.3 ± 194.7	110.5 ± 210.6	115.8 ± 203.6
Question Length	13.0 ± 2.9	10.6 ± 4.0	10.0 ± 3.8	13.6 ± 3.3	11.3 ± 4.4	11.6 ± 4.6	14.2 ± 3.6	12.1 ± 4.9	12.9 ± 5.4
Narrative Length	113.1 ± 98.2	166.1 ± 167.6	144.7 ± 146.0	132.2 ± 104.1	159.2 ± 138.5	169.4 ± 145.2	144.6 ± 117.9	161.6 ± 158.2	167.9 ± 143.4

Table 1: Dataset statistics of the each category in the LaMP-QA benchmark.

puts, we utilize Gemini 1.5 Pro[3] (Gemini-Team, 2024) as the LLM. For the training set, we use Gemma 2 (Gemma-Team, 2024) with 27 billion parameters to reduce computational costs while maintaining strong effectiveness. To validate the effectiveness of this filtering, we manually review a sample of 500 questions from the remaining questions. This quality check ensures that the remaining questions after filtering require personalized information to improve response generation. Based on our observations, almost all the questions in this sample benefit from personalization and require some personalized context to generate high-quality responses that effectively answer the question.

Filtering out questions lacking sufficient information in question narrative about the response requirements: To evaluate a response to a given question, we extract the user's information needs from the question narrative and assess how well the response addresses them. However, some question narratives (i.e., post details) lack sufficient detail for this evaluation and do not provide clear rubrics. Thus, we filter out such questions. To achieve this, we use a capable LLM with the prompt shown in Figure 7 in Appendix A. This prompt determines whether a question narrative explicitly specifies aspects that a response should address. If it does, the model extracts these aspects, provides supporting evidence, and explains their significance. The extracted aspects correspond to the set E_{x_u} defined in Section 2, which we use to evaluate response personalization. Since these aspects are derived directly from user-written question narrative detailing their information needs, they are inherently personalized. For the test and validation sets, to ensure high-quality outputs, we use Gemini 1.5 Pro as the LLM. For the training set, we use Gemma 2 with 27 billion parameters to reduce costs. To evaluate the quality of the extracted rubric aspects, we conduct a human evaluation study. We sample 100 questions that passed this filtering step and present them to annotators, who rate the extracted aspects on a 1–5 scale based on their alignment with the user's stated information needs in the question narratives. Each example is independently reviewed by two annotators. The inter-annotator agreement, measured using Cohen's kappa, is 0.87, indicating a high level of consistency between reviewers. The results show an average score of 4.9 out of 5 for the extracted aspects, demonstrating their strong alignment with the information needs specified in the question narratives and confirming the high quality of the extraction. The detailed guidelines used for the human evaluation are included in Appendix B.

Forming the LaMP-QA benchmark: We use the SE-PQA train, validation, and test sets, applying the filtering steps to retain only questions that are suitable for personalized evaluation. This ensures the dataset includes questions that benefit from personalization and contain aspects reflecting the user's specific information needs. For each remaining post from user u, we treat the post's question as the input x_u and extract the relevant aspects, denoted as E_{x_u}, from the question narrative r_{x_u}. To construct the user profile P_u, we gather previous posts from the same user, capturing their information-seeking behavior. For a more fine-grained evaluation, we categorize the filtered questions into three categories: **(1) Art & Entertainment**, **(2) Lifestyle & Personal Development**, and **(3) Society & Culture**. Each of the main categories consists of subcategories, totaling over 45 subcategories. The distribution of subcategories is shown in Figure 5 in Appendix A. The dataset statistics are provided in Table 1. Two examples from our dataset, which includes the question, question narrative, and the extracted key aspects relevant to the user, is presented in Figures 8 and 9 in Appendix A.

3.2 Evaluation Metric

Inspired by recent advances in the evaluation of personalized text generation (Salemi et al., 2025a) and

[3]Available at: https://ai.google.dev/gemini-api/docs/models/gemini#gemini-1.5-pro

aspect-based evaluation (Min et al., 2023; Samarinas et al., 2025), we evaluate a generated personalized response $\hat{y}_u$ to a user query x_u using extracted aspects from the question narrative, denoted as E_{x_u}. These aspects remain hidden during response generation and are only utilized for evaluation. Specifically, we evaluate the generated response $\hat{y}_u$ for each aspect $e \in E_{x_u}$ using an LLM. In this paper, we use the instruction-tuned Qwen 2.5 (Qwen et al., 2025) model with 32 billion parameters as the LLM for evaluation unless stated otherwise. The response is rated on a scale from 0 to 2 for each aspect, following the prompt shown in Figure 12 in Appendix C. The scores are then normalized by dividing by 2. The final evaluation score for the generated response is computed as the average of the normalized scores across all aspects $e \in E_{x_u}$.

4 Baselines for the LaMP-QA Benchmark

This section introduces both existing and newly proposed baselines on the LaMP-QA benchmark.

No-Personalization. For this baseline, the question x_u is provided directly to the LLM M to generate a response $\hat{y}_u = M(x_u)$, without incorporating any personalized information, using the prompt shown in Figure 13 in Appendix D. Since this method doesn't use the user's profile during generation, the response is generic and not personalized.

RAG-Personalization. Following Salemi et al. (2024b), we adopt RAG to incorporate personalized context from the user's profile when answering a question. Specifically, the question x_u is used as a query to retrieve the top k relevant entries from the user's profile P_u using a retrieval model R. The retrieved content is then concatenated with the question and passed to the LLM M to generate a personalized response, denoted as: $\hat{y}_u = M(x_u, R(x_u, P_u, k))$, using the prompt shown in Figure 14 in Appendix D. This approach allows the model to condition its generation on user-specific information, enabling the model to learn about the user preferences from its history to generate a more relevant and tailored response.

Plan-RAG Personalization (PlanPers) This method extends the RAG-Personalization approach by introducing a planning step prior to response generation. Given the question x_u and the top k retrieved items from the user's profile $R(x_u, P_u, k)$, a planner model M_{plan} uses the prompt shown in Figure 15 in Appendix D to generate a set of aspects that are likely important to the user in the context of the question. These aspects are represented as a plan $p_{x_u} = M_{\text{plan}}(x_u, R(x_u, P_u, k))$. The final response is then generated by the LLM M using the prompt shown in Figure 15 in Appendix D, conditioned on the original question, the retrieved personalized context, and the generated plan: $\hat{y}_u = M(x_u, R(x_u, P_u, k), p_{x_u})$. This process aims to guide response generation by first explicitly inferring the aspects that the user is likely to expect in the answer, based on their question and personal history. These inferred aspects are then incorporated into the generation process, encouraging the model to produce responses that more effectively address the user's information needs.

Since the LaMP-QA benchmark does not include reference responses, directly training the LLM M for response generation is not feasible. However, as described in Section 3, the dataset provides a set of rubric aspects that reflect the user's expectations for the response. Leveraging this, we train a planner M_{plan} to predict these aspects based on the question x_u and the retrieved personalized context $R(x_u, P_u, k)$. We frame this as a sequence-to-sequence (Sutskever et al., 2014) problem and train M_{plan} using cross-entropy loss to generate the expected aspects, with each aspect title separated by a newline. During inference, the planner model predicts expected rubric aspects given the question and the personalized context. These aspects can then be used to guide the generation process, helping the model produce responses that are more aligned with the user's specific information needs.

5 Experiments

5.1 Experimental Setup

We use a combination of open and proprietary models for the generator LLM M. We employ instruction-tuned Gemma 2 (9B parameters) (Gemma-Team, 2024), instruction-tuned Qwen 2.5 (7B parameters) (Qwen et al., 2025), and GPT-4o-mini (OpenAI, 2024) as the proprietary model. For the planner model M_{plan}, we use Qwen 2.5 with 7B parameters. Training details for the planner using LoRA (Hu et al., 2022) are provided in Appendix E. All models operate with a maximum input-output token limit of 8192 and generate responses using nucleus sampling (Holtzman et al., 2020) with a temperature of 0.1. For retriever, we use Contriever (Izacard et al., 2022), fine-tuned on MS MARCO

Method	Arts & Entertainment	Lifestyle & Personal Development	Society & Culture	Average (macro)
Gemma 2 Instruct (9B)				
No-Personalization	0.1860	0.3858	0.4094	0.3270
RAG-Personalization (Random P)	0.1708	0.3415	0.3310	0.2811
RAG-Personalization (Asker's P_u)	0.2929	0.4232	0.4834	0.3998
PlanPers	**0.3548**†	**0.4671**†	**0.5481**†	**0.4566**†
Qwen 2.5 Instruct (7B)				
No-Personalization	0.3129	0.4582	0.4769	0.416
RAG-Personalization (Random P)	0.2547	0.3829	0.4037	0.3471
RAG-Personalization (Asker's P_u)	0.3397	0.4481	0.4967	0.4281
PlanPers	**0.3518**†	**0.4818**†	**0.5240**†	**0.4525**†
GPT 4o-mini				
No-Personalization	0.3713	0.5112	0.5218	0.4681
RAG-Personalization (Random P)	0.2881	0.4148	0.4202	0.3743
RAG-Personalization (Asker's P_u)	0.3931	0.4884	0.5310	0.4708
PlanPers	**0.4490**†	**0.5442**†	**0.6084**†	**0.5338**†

Table 2: Performance on the test set. † shows a statistically significant difference between the best-performing baseline and the others using t-test ($p < 0.05$). The results on the validation set are reported in Table 3 in Appendix F.

(Bajaj et al., 2018), to retrieve $k = 10$ items from the user profile, unless otherwise noted. Implementation details are presented in Appendix E.

5.2 Main Findings

Baselines Performance. The performance of the baselines on the LaMP-QA benchmark is reported in Table 2 for the test set and in Table 3 in Appendix F for the validation set. The results in Table 2 demonstrate that Plan-RAG Personalization significantly outperforms all baselines across all evaluation categories. This highlights the utility of the training data provided by the LaMP-QA benchmark for training an effective planner model that can infer the key information users expect in response to their questions. Furthermore, the results show that all personalized baselines leveraging the asker's profile to tailor the LLM's response outperform the non-personalized model in nearly all cases. This indicates that incorporating user-specific context enhances the relevance and quality of the generated responses for each user, underscoring the importance of personalization in question answering.

Moreover, we observe that the best performance is achieved when GPT-4o-mini is used as the backbone LLM for generation. Notably, the highest performance occurs when Plan-RAG Personalization is applied to this model, highlighting the effectiveness of the planning step in understanding user preferences and incorporating them into the generated response. This demonstrates that even with a strong underlying LLM, explicitly modeling user intent through planning provides substantial gains in personalized question answering.

To further assess the comparative quality of generated responses, we conducted a human evaluation between the two strongest baselines, PlanPers and RAG-Personalization, both with Qwen 2.5 Instruct as the backbone. We randomly sampled 100 outputs from each system and asked two human annotators to evaluate them. For each instance, annotators were provided with the question, its narrative, and the personalized rubrics, and were asked to select the better response or indicate a tie. The results, shown in Figure 1, indicate that PlanPers was preferred in 35% of cases, while RAG-Personalization was preferred in 26% of cases. The remaining instances were judged as ties. These findings suggest that PlanPers produces responses that are more consistently aligned with the question narrative and personalized rubrics from a human perspective.

Effect of Personal Data in the LaMP-QA Benchmark on the Performance. A crucial criterion

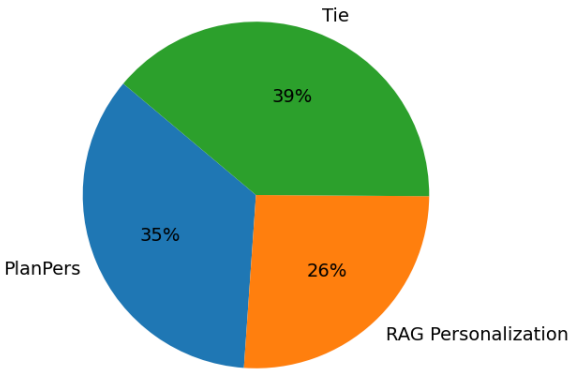

Figure 1: Results of pairwise human evaluation between RAG-Personalization and PlanPers with Qwen 2.5 Instruct 7B as the backbone LLM. Each slice reports the winning percentage of the corresponding method.

for a dataset designed for personalized question answering is that incorporating personalized context, such as a user's history, into the response generation process should lead to improvements in the preferability of the generated response, as assessed by a metric grounded in the user's information needs. To assess the LaMP-QA benchmark under this criterion, we conduct three experiments.

In the first experiment, we compare a non-personalized LLM with a personalized LLM that uses RAG without any additional training. As shown in Table 2, incorporating personalized information consistently improves performance across almost all cases. This demonstrates that the user-specific context provided in the LaMP-QA benchmark is effective in enabling LLMs to generate more tailored and relevant responses.

In the second experiment, we aim to assess whether the user profiles provided in the LaMP-QA benchmark are indeed tailored to individual users and beneficial for generating personalized responses. To this end, we use the RAG approach but retrieve information from a random user profile rather than from the actual asker's profile. The results, presented in Table 2 for the test set and Table 3 in Appendix F for the validation set, reveal a significant performance drop when using random profiles. In fact, this setting performs worse than the non-personalized baseline, highlighting that the retrieved context must be user-specific to be helpful. These findings confirm that the user profiles in the LaMP-QA benchmark are aligned with the users' rubric-based expectations and are useful for studying personalized question answering.

In the third experiment, we investigate the effect of varying the amount of retrieved information from the user profile on the performance of Plan-RAG Personalization, the best-performing baseline. Specifically, we vary the number of retrieved items $k \in \{0, 2, 4, 6, 8, 10\}$ and report the model's performance on the test set in Figure 2, and on the validation set in Figure 16 in Appendix F. The results show improvement in performance as more items are retrieved from the user profile, suggesting that incorporating a larger amount of personalized context helps the model better infer and address the user's preferences. This highlights the importance of rich user history in effective personalization.[4]

Headroom Analysis for Improving Planning in PlanPers. To evaluate the potential upper bound of plan quality in PlanPers, we conduct an experiment comparing our trained planner with an oracle setting. In the oracle condition, instead of generating plans using the learned planner, we directly use gold-standard plans derived from the rubric aspects associated with each user. These gold plans are then provided to the model to generate responses. The results of this comparison, shown in Figure 3 for Qwen 2.5 Instruct, indicate that access to gold plans substantially improves performance, yielding gains of 155%–208% on different tasks. While the trained planner already achieves significant improvements over baseline methods (Table 2), this gap highlights considerable headroom for further enhancing the planner's ability to generate high-quality and user-specific plans.

Effect of Evaluation Method and Auto-Rater Size. One might ask about alternative methods for evaluating generated responses. We compare three approaches, each aiming to assess how well a response aligns with the user's information needs. First, inspired by prior work (Kocmi and Federmann, 2023; Liu et al., 2023), we use a direct scoring approach in which, given the question, the question narrative representing the user's stated information need, and the generated response, the LLM assigns a score between 0 and 1 using the prompt shown in Figure 10 in Appendix C. Second, we evaluate responses in a pairwise setup, where the LLM is provided with the question, the user's stated information need, and two candidate

[4]Due to the 8192-token limit of Gemma 2, we restrict the number of retrieved items to a maximum of 10.

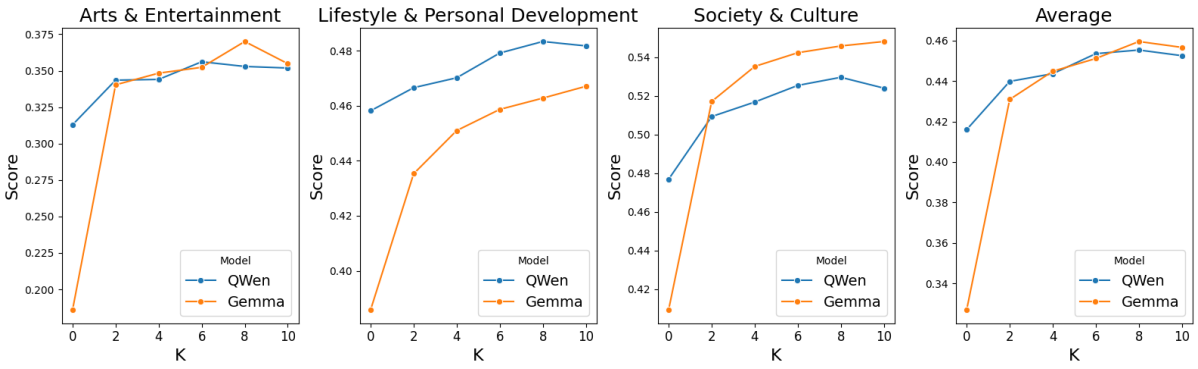

Figure 2: Effect of number of items (K) from the user's profile on the performance of Plan-RAG Personalization on the test set. The results for the validation set is shown in Figure 16 in Appendix F.

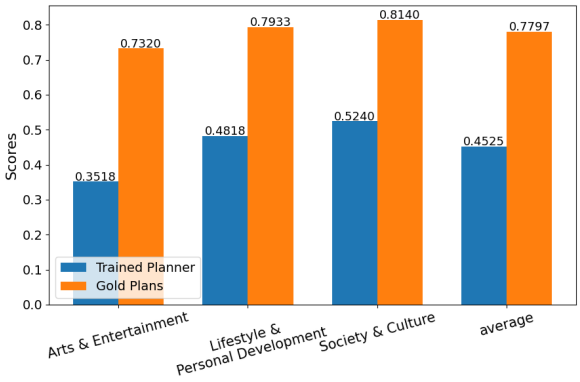

Figure 3: Comparison between trained planner and utilizing gold plans on the performance of PlanPers (Qwen 2.5 Instruct 7B) on the test set of LaMP-QA.

responses, and is asked to choose the better one using the prompt in Figure 11. Lastly, we use our proposed evaluation method in Section 3.2, which scores responses based on how well they address the individual aspects extracted from the user's information needs. The implementation details for all evaluation approaches are provided in Appendix C.

In the pairwise setting, we observe a strong position bias that undermines the reliability of this approach. Consistent with prior findings by Salemi et al. (2025a), our experiments show that instruction-tuned Qwen 2.5 (32B parameters) changes its preferred response in 78% of cases when the order of the two responses is reversed. This high sensitivity to position shows that the model's judgments are heavily influenced by presentation order, making this method unsuitable for robust and fair assessment of responses.

To assess the effectiveness of the other two methods, we present 100 examples from the test set to human annotators. Each example includes two generated responses—one from Plan-RAG Personalization and one from RAG-Personalization, both using Qwen 2.5. Annotators are instructed to choose the response that better addresses the information need from the question narrative. Each example is independently evaluated by two raters. The inter-annotator agreement, measured using Cohen's kappa, is 0.726, indicating moderate to substantial agreement and validating the reliability of the human evaluation process. The results show that in 73% of cases, the aspect-based evaluation method selects the same response as human annotators, while the direct scoring method aligns with human preferences in only 58% of cases. This demonstrates that the proposed aspect-based evaluation approach achieves a higher alignment with human judgments, highlighting its effectiveness as a more reliable and fine-grained evaluation strategy for personalized question answering.

In another experiment, we observed that the number of parameters of the evaluator LLM significantly impacts both its alignment with human judgment and the scores it assigns to the responses. We report the alignment with human preferences and the average score assigned to the generated responses using the aspect-based evaluation approach with Qwen2.5 models of varying sizes—0.5B, 3B, 7B, and 32B parameters—in Figure 4. The results show a clear trend: as the size of the evaluator LLM increases, its alignment with human judgment improves, while the average score it assigns to outputs decreases. Specifically, with a 0.5B parameter model, alignment with human preferences is only 48%, and the average score assigned is relatively high at 0.94. In contrast, the 32B evaluator achieves a much higher alignment of 73% but assigns a significantly lower average score of 0.44. This suggests that larger evaluator LLMs are better at capturing nuanced quality signals aligned with human expectations. We found that smaller LLMs are less capable of distinguishing varying

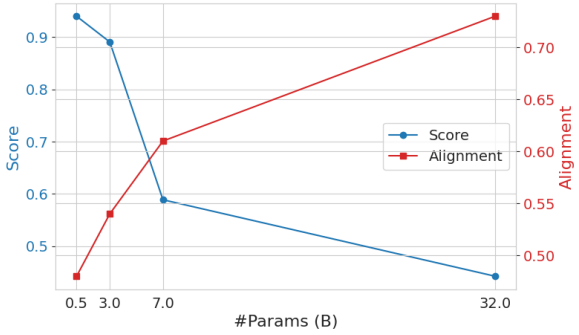

Figure 4: Trade-off between assigned score and alignment with human using the aspect-based evaluation metric in Section 3.2 across different evaluator sizes.

degrees to which a response addresses a particular aspect. Instead, they tend to behave like binary classifiers without the ability to assess how well it is addressed. This leads to inflated scores and weaker alignment with human evaluators.

6 Related Work

Personalized LLMs. Personalization plays a central role in search, recommendation, and text generation (Fowler et al., 2015; Xue et al., 2009; Naumov et al., 2019; Salemi et al., 2024b). To personalize an LLM, Salemi et al. (2024b) proposed a RAG framework that retrieves information from the user profile and incorporates it into the prompt provided to the LLM. Existing methods span a range of strategies, including training retrievers with relevance feedback (Salemi et al., 2024a), optimizing LLMs using user-specific supervision (Jang et al., 2023), and creating personalized prompts tailored to the user (Li et al., 2024). Parameter-efficient fine-tuning have been proposed for personalized generation (Tan et al., 2024), with recent work integrating such methods into RAG pipelines (Salemi and Zamani, 2024). In addition, reasoning and self-training have shown effectiveness in improving long-form personalized generation (Salemi et al., 2025b). Personalized assistants have also been investigated in recent work (Li et al., 2023; Mysore et al., 2023; Lu et al., 2024; Zhang et al., 2024). Despite growing interest in personalized NLP, personalized question answering remains underexplored.

Personalized Information Seeking. Personalized text generation has been explored in both short- and long-form generation settings, including tasks such as email subject line generation and social media post generation, as demonstrated in benchmarks like LaMP (Salemi et al., 2024b) and LongLaMP (Kumar et al., 2024). However, these benchmarks focus solely on content generation and do not cover information-seeking. On the other hand, personalized information-seeking has been studied in the context of retrieval, using datasets such as SE-PQA (Kasela et al., 2024), AOL4PS (Guo et al., 2021), and the Personalized Web Search Challenge (Eugene et al., 2013). With the increasing adoption of generative AI systems, it is crucial to revisit this problem from a generation perspective—an area that remains underexplored despite its importance. This paper addresses this gap by introducing a the LaMP-QA benchmark specifically designed to evaluate personalized question answering with LLMs.

7 Conclusion

This paper introduced the LaMP-QA benchmark, specifically designed to evaluate personalized question answering with LLMs. The benchmark spans three broad domains: Arts & Entertainment, Lifestyle & Personal Development, and Society & Culture. We conducted a comprehensive analysis to assess the benchmark's effectiveness in facilitating the evaluation of personalization in question answering. We investigate multiple evaluation strategies and find that aspect-based evaluation achieves the highest alignment with human judgment. We evaluate standard RAG baselines and introduce novel planning-based methods for generating personalized responses that align more with the user's information needs. Experimental results demonstrate that leveraging personalized context alongside planning-based personalized response generation leads to substantial improvements in response quality and personalization.

Acknowledgment

We would like to thank Cheng Li, Mingyang Zhang, Qiaozhu Mei, Weize Kong, Tao Chen, Zhuowan Li, Spurthi Amba Hombaiah, and Michael Bendersky for their valuable feedback and generous support in preparing this work and shaping the formulation of the problem. This work was supported in part by the Center for Intelligent Information Retrieval, in part by NSF grant #2143434, in part by NSF grant #2402873, and in part by Google. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

Limitations

This work has the following limitations:

On the Automatic Evaluation of Personalization. Although the proposed evaluation method demonstrates strong alignment with human judgments, the automatic evaluation of personalized text generation remains a challenging problem (Salemi et al., 2024b, 2025a,b; Kumar et al., 2024), as the most reliable evaluator is the original user who posed the question. However, consistently obtaining feedback from the same user across different studies is impractical, costly, and in many cases impossible. As a result, automatic evaluation methods are necessary to enable scalable and reproducible assessment of personalization quality. Note that this challenge is inherent to the personalization problem itself and is not specific to our or any personalization evaluation dataset, as also noted by previous work (Salemi et al., 2025a; Kumar et al., 2024).

On the Completeness of Question Narratives. One underlying assumption in this work is that users articulate all of their information needs within the question narrative. However, this is not always the case—users may be uncertain about what they expect in a response or may omit relevant aspects unknowingly. To address this, and following prior work on nugget-based evaluation of response generation (Pradeep et al., 2025), we adopt a recall-oriented evaluation strategy. Under this approach, a response is considered satisfactory if it successfully addresses all explicitly stated user aspects, without penalizing the model for not covering unstated or implicit needs. Capturing the information needs that users are unaware of remains a difficult and often infeasible task. Nonetheless, developing techniques to handle such cases represents an important future direction for more comprehensive evaluation of personalization in question answering.

On the Privacy Considerations of Personalization. Personalizing LLMs necessitates access to user-specific data in order to effectively tailor responses. However, the use and sharing of personal data raises important privacy concerns that must be carefully addressed in the development of effective personalized language models (Li et al., 2023; Salemi and Zamani, 2024; Volokh, 2000). While this paper does not investigate privacy-preserving approaches, it focuses solely on enabling research by providing publicly available data for studying personalized question answering. Addressing the privacy implications of personalized LLMs and resolving them remains an important future work.

On the Model Size and Families. There exist various families of open-source and proprietary large language models (LLMs) with different model sizes. However, due to computational and financial constraints, this paper focuses on evaluating two widely used open-source model families, Gemma (Gemma-Team, 2024) and Qwen (Qwen et al., 2025), at a specific size. While a comprehensive comparison across model architectures and sizes would be valuable, it falls beyond the scope of this work. Our primary goal is not to benchmark model performance exhaustively, but rather to establish representative baselines for the proposed `LaMP-QA` benchmark. Nonetheless, we acknowledge this as a limitation of our study, though not a fundamental flaw of the dataset itself.

References

James Allan. 2003. Hard track overview in trec 2003: High accuracy retrieval from documents. In *Text Retrieval Conference*.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2018. Ms marco: A human generated machine reading comprehension dataset. *Preprint*, arXiv:1611.09268.

Krisztian Balog, Pavel Serdyukov, and Arjen P. de Vries. 2010. Overview of the trec 2010 entity track. In *Text Retrieval Conference*.

Chris Buckley, Mandar Mitra, Janet A. Walz, and Claire Cardie. 1998. Smart high precision: Trec 7. In *Text Retrieval Conference*.

Eugene, serdyukovpv, and Will Cukierski. 2013. Personalized web search challenge. `https://kaggle.com/competitions/yandex-personalized-web-search-challenge`. Kaggle.

Andrew Fowler, Kurt Partridge, Ciprian Chelba, Xiaojun Bi, Tom Ouyang, and Shumin Zhai. 2015. Effects of language modeling and its personalization on touchscreen typing performance. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, page 649–658, New York, NY, USA. Association for Computing Machinery.

Gemini-Team. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *Preprint*, arXiv:2403.05530.

Gemma-Team. 2024. Gemma 2: Improving open language models at a practical size. *Preprint*, arXiv:2408.00118.

Qian Guo, Wei Chen, and Huaiyu Wan. 2021. Aol4ps: A large-scale data set for personalized search. *Data Intelligence*, 3(4):548–567.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

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*.

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*.

Joel Jang, Seungone Kim, Bill Yuchen Lin, Yizhong Wang, Jack Hessel, Luke Zettlemoyer, Hannaneh Hajishirzi, Yejin Choi, and Prithviraj Ammanabrolu. 2023. Personalized soups: Personalized large language model alignment via post-hoc parameter merging. *Preprint*, arXiv:2310.11564.

Pranav Kasela, Marco Braga, Gabriella Pasi, and Raffaele Perego. 2024. Se-pqa: Personalized community question answering. In *Companion Proceedings of the ACM Web Conference 2024*, WWW '24, page 1095–1098, New York, NY, USA. Association for Computing Machinery.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Tom Kocmi and Christian Federmann. 2023. Large language models are state-of-the-art evaluators of translation quality. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 193–203, Tampere, Finland. European Association for Machine Translation.

Ishita Kumar, Snigdha Viswanathan, Sushrita Yerra, Alireza Salemi, Ryan A. Rossi, Franck Dernoncourt, Hanieh Deilamsalehy, Xiang Chen, Ruiyi Zhang, Shubham Agarwal, Nedim Lipka, Chien Van Nguyen, Thien Huu Nguyen, and Hamed Zamani. 2024. Longlamp: A benchmark for personalized long-form text generation. *Preprint*, arXiv:2407.11016.

Dawn Lawrie, Sean MacAvaney, James Mayfield, Paul McNamee, Douglas W. Oard, Luca Soldaini, and Eugene Yang. 2024. Overview of the trec 2023 neuclir track. *Preprint*, arXiv:2404.08071.

Cheng Li, Mingyang Zhang, Qiaozhu Mei, Weize Kong, and Michael Bendersky. 2024. Learning to rewrite prompts for personalized text generation. In *Proceedings of the ACM on Web Conference 2024*, WWW '24. ACM.

Cheng Li, Mingyang Zhang, Qiaozhu Mei, Yaqing Wang, Spurthi Amba Hombaiah, Yi Liang, and Michael Bendersky. 2023. Teach llms to personalize – an approach inspired by writing education. *Preprint*, arXiv:2308.07968.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

Zhuoran Lu, Sheshera Mysore, Tara Safavi, Jennifer Neville, Longqi Yang, and Mengting Wan. 2024. Corporate communication companion (ccc): An llm-empowered writing assistant for workplace social media. *Preprint*, arXiv:2405.04656.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.

Sheshera Mysore, Zhuoran Lu, Mengting Wan, Longqi Yang, Steve Menezes, Tina Baghaee, Emmanuel Barajas Gonzalez, Jennifer Neville, and Tara Safavi. 2023. Pearl: Personalizing large language model writing assistants with generation-calibrated retrievers. *Preprint*, arXiv:2311.09180.

Maxim Naumov, Dheevatsa Mudigere, Hao-Jun Michael Shi, Jianyu Huang, Narayanan Sundaraman, Jongsoo Park, Xiaodong Wang, Udit Gupta, Carole-Jean Wu, Alisson G. Azzolini, Dmytro Dzhulgakov, Andrey Mallevich, Ilia Cherniavskii, Yinghai Lu, Raghuraman Krishnamoorthi, Ansha Yu, Volodymyr Kondratenko, Stephanie Pereira, Xianjie Chen, and 5 others. 2019. Deep learning recommendation model for personalization and recommendation systems. *Preprint*, arXiv:1906.00091.

OpenAI. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Ronak Pradeep, Nandan Thakur, Shivani Upadhyay, Daniel Campos, Nick Craswell, and Jimmy Lin. 2025. The great nugget recall: Automating fact extraction and rag evaluation with large language models. *Preprint*, arXiv:2504.15068.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin

Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

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, Julian Killingback, and Hamed Zamani. 2025a. Expert: Effective and explainable evaluation of personalized long-form text generation. *Preprint*, arXiv:2501.14956.

Alireza Salemi, Cheng Li, Mingyang Zhang, Qiaozhu Mei, Weize Kong, Tao Chen, Zhuowan Li, Michael Bendersky, and Hamed Zamani. 2025b. Reasoning-enhanced self-training for long-form personalized text generation. *Preprint*, arXiv:2501.04167.

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.

Alireza Salemi and Hamed Zamani. 2024. Comparing retrieval-augmentation and parameter-efficient fine-tuning for privacy-preserving personalization of large language models. *Preprint*, arXiv:2409.09510.

Chris Samarinas, Alexander Krubner, Alireza Salemi, Youngwoo Kim, and Hamed Zamani. 2025. Beyond factual accuracy: Evaluating coverage of diverse factual information in long-form text generation. *Preprint*, arXiv:2501.03545.

Huawei Shen, Guoyao Chen, Haiqiang Chen, Yue Liu, and Xueqi Cheng. 2007. Research on enterprise track of trec 2007. In *TREC*.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 3104–3112, Cambridge, MA, USA. MIT Press.

Zhaoxuan Tan, Zheyuan Liu, and Meng Jiang. 2024. Personalized pieces: Efficient personalized large language models through collaborative efforts. *Preprint*, arXiv:2406.10471.

Eugene Volokh. 2000. Personalization and privacy. *Commun. ACM*, 43(8):84–88.

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. *Preprint*, arXiv:2503.02614.

Gui-Rong Xue, Jie Han, Yong Yu, and Qiang Yang. 2009. User language model for collaborative personalized search. *ACM Trans. Inf. Syst.*, 27(2).

Kai Zhang, Yangyang Kang, Fubang Zhao, and Xiaozhong Liu. 2024. LLM-based medical assistant personalization with short- and long-term memory coordination. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2386–2398, Mexico City, Mexico. Association for Computational Linguistics.

A Filtering Prompts & Examples from the LaMP-QA Benchmark

The LaMP-QA benchmark employs Gemini 1.5 Pro to refine the SE-PQA (Kasela et al., 2024) dataset by filtering out factoid questions that do not benefit from personalization, using the prompt provided in Figure 6. Furthermore, the same model is used to exclude questions that lack sufficient information in the question narrative (i.e., post details) necessary for evaluating response requirements, based on the prompt illustrated in Figure 7.

To illustrate the structure of questions, question narrative, and rubric aspects in the LaMP-QA benchmark, Figures 8 and 9 present two representative examples from the dataset. These examples demonstrate how the benchmark captures user-specific information needs and the corresponding personalized evaluation criteria.

B Human Annotation Instructions

We conduct two types of human annotation in this study. In the first experiment, we present annotators with a question, its corresponding question narrative, and the set of automatically extracted aspects obtained using Gemini 1.5 Pro as the extraction model. Annotators are instructed to evaluate the quality of the extracted aspects based on how well they reflect the user's stated information needs, as described in the question narrative, using the following criteria:

5: The generated aspects contain all the important information needs mentioned in the post detail.

4: The generated aspects contain most of the important information needs mentioned in the post detail.

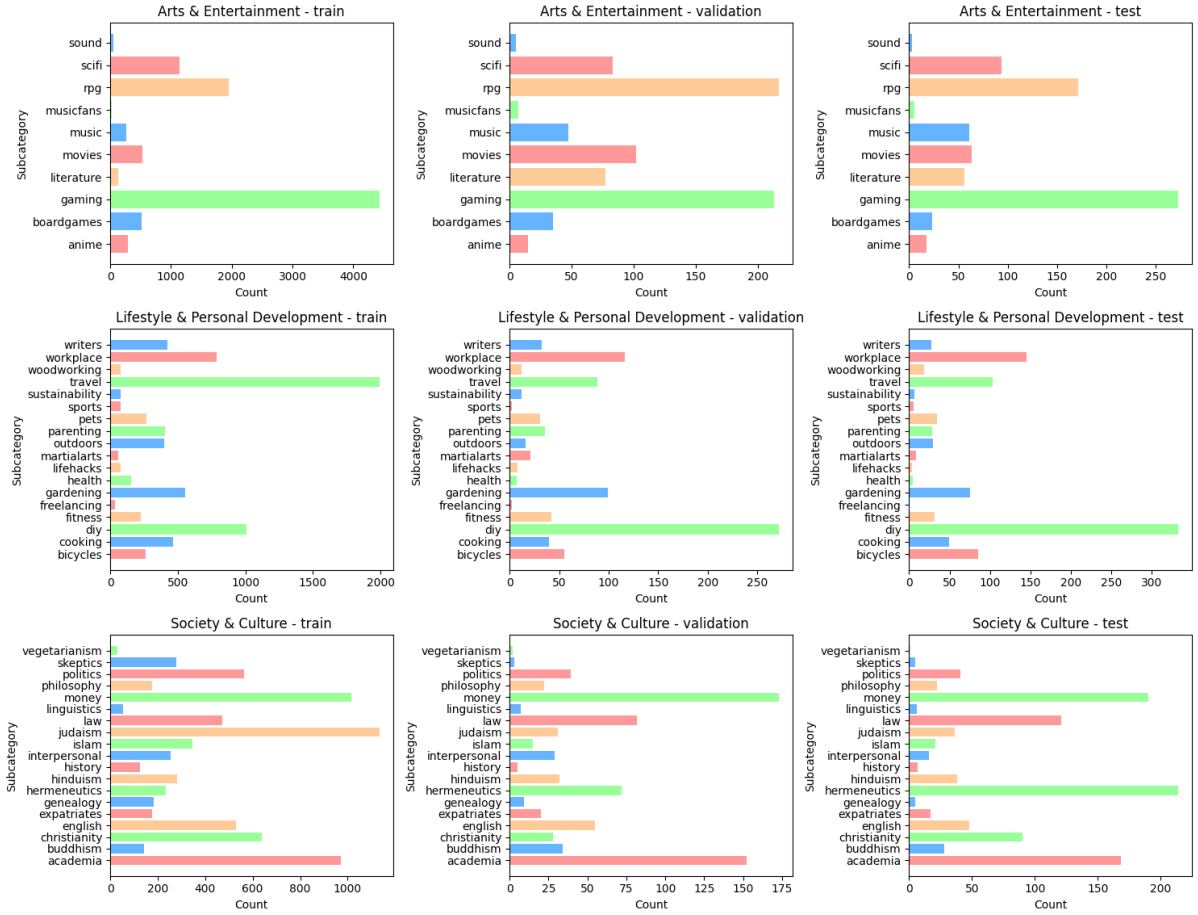

Figure 5: Distribution of subcategories in train, validation, and test sets of the LaMP-QA benchmark.

Prompt for Filtering Personalizable Questions

You are a helpful assistant with amazing judgment capabilities. Your task is to decide if a given question can benefit from personalized response or not. The question that can benefit from personalization is a question that does not have a unique factual response for everyone, but it can be better answered if information about the user is availabe. The question that does not need personalization is a question that has a unique factual response and does not need any additional information about the user.

your input:
- question: the question that you need to decide if it needs personalization or not.

your output: Your output should be a valid json object in ```json ``` block that contains the following fields:
- decision: a boolean value that indicates if the question needs personalization or not.
- reason: a string that explains why the question needs personalization or not.

question: {question}
output: ```json

Figure 6: Prompt used with Gemini 1.5 Pro to evaluate whether the question can benefit from personalization.

3: The generated aspects contain some of the important information needs mentioned in the post detail but missed a few of them.

2: The generated aspects contain some of the important information needs mentioned in the post detail but missed some of them.

Prompt for Filtering Evaluable Questions

```
You are a highly skilled assistant with excellent judgment and analytical capabilities. Your task is to evaluate whether the given detailed post contains sufficient information about the aspects that the user expect to see in a response to their question. If the post contains the required information, extract those aspects and provide a clear explanation of why each aspect is significant for the user and what specific details they would expect to see in the response.

# your input:
    - post: the detailed post that the user who asked the question provided the information about the question
    - question: the question that the user asked

# your output: Your output should be a valid json object in ```json ``` block that each object contains the following fields:
    - enough_details: A boolean that is true if the detailed post has information about the aspects that are important to the user or if these aspects can be inferred.
    - aspects: If enough_details is false, return an empty list. If enough_details is true, return a list of aspects that are important for the user to be included in the response to their question. Each aspect should have the following fields:
        - aspect: a string that is the name of the aspect that is important to be present in the response
        - evidence: a string that shows the points to the section from the detailed post that mentions this aspect
        - reason: a string that explains why the aspect is important for the user
question: {question}
post: {post}
output: ``` json
```

Figure 7: Prompt used with Gemini 1.5 Pro to assess whether sufficient information is available for evaluation and extract key rubric aspects from the question narrative.

```
1: The generated aspects contain a few of the important information needs mentioned in the post detail and missed most of them.
```

In the second experiment, annotators are provided with two generated responses for a given question, along with the corresponding question narrative that outlines the user's information need. Based on the information specified in the question narrative, annotators are asked to determine which of the two responses better addresses the user's stated requirements and to select the preferred response accordingly:

```
Given a question, its corresponding post detail, and two generated responses, please evaluate which response best addresses the user's information need as described in the post detail. You may select:
- Response 1 if it better satisfies the user's stated requirements,
- Response 2 if it better satisfies the user's stated requirements, or
- Tie if both responses are equally satisfactory in addressing the question based on the post detail.

Question: [question]
Post Detail: [details]

Response 1: [response 1]
Response 2: [response 2]
```

C Evaluation Metric Details

In this paper, we explore three distinct evaluation strategies for assessing the quality of generated responses to user questions. We compare the outcomes of these methods against human preference judgments to identify the most effective evaluation approach for the LaMP-QA benchmark.

LLM-Based Directly Scoring Using User-Stated Information Needs: In this method, we provide the LLM with the question, the corresponding ques-

Figure 8: Example 1 of a question, the question narrative (i.e., post details), and the extracted important aspects that the user expects to be addressed in the response from the LaMP-QA benchmark.

tion narrative that specify the user's information needs, and the generated response, along with a set of evaluation criteria defined in the prompt shown in Figure 10. The LLM evaluates the generated output based on how well it aligns with the stated information needs and assigns a score on a 5-point scale: 0, 0.25, 0.5, 0.75, 1. This approach relies on the LLM's ability to interpret and reason over the user's expectations as expressed in the question narrative.

LLM-Based Pairwise Preference Using User-Stated Information Needs: In this method, the LLM is presented with the question, the corresponding question narrative that articulate the user's information needs, and two candidate responses. The model is then asked to select the response that better addresses the user's question, taking into account the specified information needs. However, consistent with prior findings by Salemi et al. (2025a), we observe that this approach suffers from significant position bias. Specifically, using

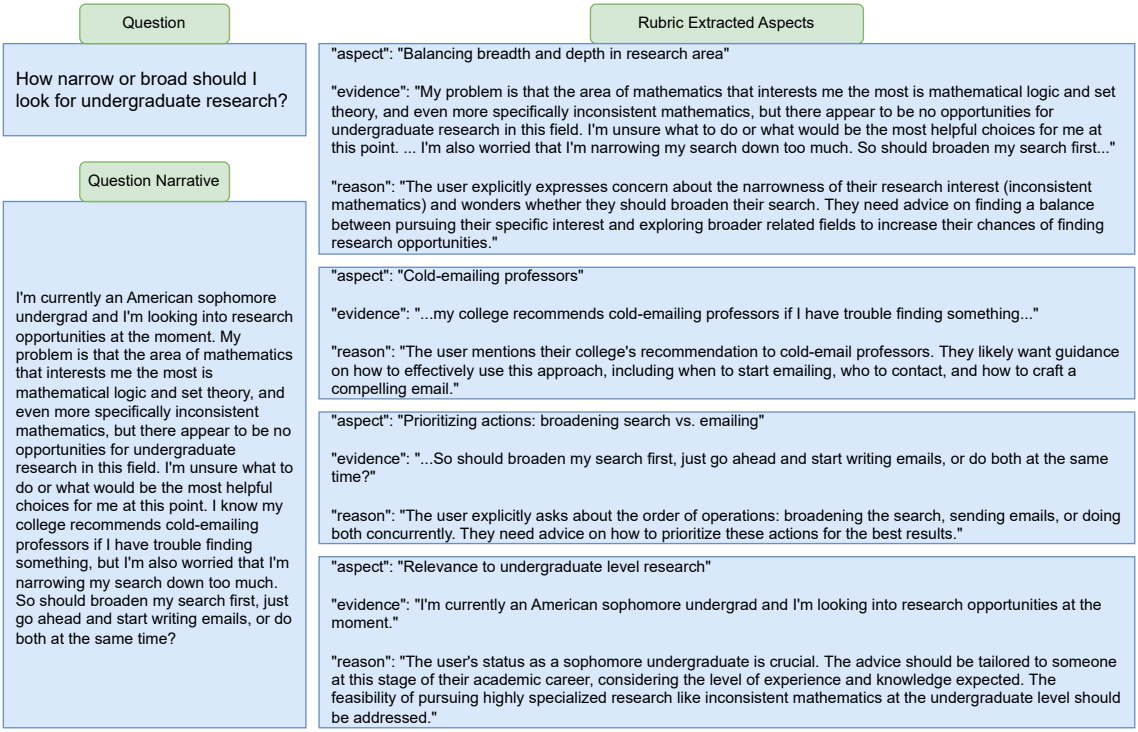

Figure 9: Example 2 of a question, the question narrative (i.e., post detail), and the extracted important aspects that the user expects to be addressed in the response from the LaMP-QA benchmark.

instruction-tuned Qwen 2.5 (32B parameters), we find that simply reversing the order of the responses alters the LLM's preference in 78% of cases. This high sensitivity to response position highlights the unreliability of this strategy for robust assessment.

Aspect-Based Evaluation Using User Information Needs: This evaluation strategy, described in Section 3.2, leverages the set of extracted important aspects derived from the user's stated information needs. For each aspect, the LLM assesses how well the generated response addresses it, assigning a score between 0 and 2, which is then normalized by dividing by 2. The prompt used for this evaluation is shown in Figure 12. The final score for the response is computed as the average of the normalized scores across all aspects. The method is formally defined in Algorithm 1. The key distinction between this method and direct scoring using an LLM is that it evaluates each aspect individually, assigning a score to each user-relevant aspect rather than providing a single overall score for the entire response. This aspect-level evaluation enables a fine-grained assessment of how well the response aligns with the user's information needs.

D Prompt Templates

As described in Section 4, this paper proposes three baseline approaches for the LaMP-QA benchmark. The first baseline, which generates responses without incorporating any personalized context, uses the prompt provided in Figure 13. The second baseline leverages RAG to incorporate personalized information retrieved from the user profile; it uses the prompt illustrated in Figure 14. The third baseline further extends RAG by introducing an intermediate planning step that identifies the aspects the user expects in a response. These aspects are then used to guide the final response generation, following the prompt in Figure 15.

E Experimental Setup Details

In this paper, we use a combination of open and proprietary models for the generator LLM M. Specifically, we employ instruction-tuned Gemma 2 (9B parameters[5]) (Gemma-Team, 2024), instruction-tuned Qwen 2.5 (7B parameters[6]) (Qwen et al.,

[5]Available at: https://hf.co/google/gemma-2-9b-it
[6]Available at: https://hf.co/Qwen/Qwen2.5-7B-Instruct

Evaluation Prompt (Without Aspects)

```
You are a fair and insightful judge with exceptional reasoning and analytical abilities. Your task is to evaluate a user's question, a generated response to that question, and assign an score to generated response based on how well it addresses the described information needs in the detailed explanation of the question from the user.
# your input:
        - question: the question asked by the user.
        - details: the detailed explanation of the question from the user.
        - response: a generated response to the user's question.

# criteria:
Score 0: The generated response does not answer the user's question based on the provided details.
Score 0.25: The response is minimally relevant but fails to adequately address the core information needs expressed in the user's detailed explanation. It may contain significant factual inaccuracies, overlook key aspects, or provide only a superficial answer.
Score 0.5: The response is partially correct and addresses some important aspects of the user's detailed question. However, it lacks completeness, precision, or depth, and may include minor inaccuracies or omissions that affect its overall usefulness.
Score 0.75: The response is mostly correct and responds to the user's detailed question in a largely satisfactory way. It may not cover all aspects fully or with optimal clarity, but it shows clear understanding and provides helpful, mostly accurate information.
Score 1: The response is fully correct, clear, and complete. It addresses all or nearly all important elements of the user's detailed question with accuracy, depth, and clarity. It demonstrates thoughtful reasoning and directly satisfies the user's information needs.

# your output: Your output should be a valid json object in ```json ``` contains the following fields:
  - score: the score assigned to generated output based on the given criteria and the detailed explanation of the question from the user.

# question:
{question}
# details:
{details}
# response:
{response}

output: ``` json
```

Figure 10: Evaluation prompt used for assessing the quality of generated personalized responses without the extracted aspects.

2025), and GPT-4o-mini[7] (OpenAI, 2024) as the proprietary model. Throughout all experiments, the generator model remains frozen and is not fine-tuned, allowing us to isolate the effects of personalization methods without altering the underlying LLM. For the planner model M_{plan}, we use Qwen 2.5 with 7B parameters. Training is performed using the Adam optimizer (Kingma and Ba, 2014) with a learning rate of 5×10^{-5} and a batch size of 32. We apply gradient clipping with a maximum norm of 1 to ensure stability. Training is conducted for up to 2000 steps, with a warmup phase spanning the first 10% of steps, followed by a linear decay of the learning rate. We fine-tune the model using LoRA (Hu et al., 2022) with rank $r = 16$, scaling factor $\alpha = 16$, and a dropout rate of 0.05, applied without modifying bias parameters. LoRA is implemented via the PEFT library.[8] Model checkpoints are evaluated every 250 steps using the validation set to monitor performance and select the best checkpoint.

All experiments are conducted using 4 NVIDIA A100 GPUs with 80GB of VRAM and 128GB of system RAM. All models are configured with a maximum input-output token limit of 8192 tokens. Response generation is performed using nucleus sampling (Holtzman et al., 2020) with a temperature of 0.1. For efficient inference and deploy-

[7]Available at: https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/

[8]Available at: https://github.com/huggingface/peft

Evaluation Prompt (Pairwise)

```
You are a fair and insightful judge with exceptional reasoning and analytical abilities. Your task is to evaluate a user's question, two generated responses to that question, and select the better response based on how well it addresses the described information needs in the detailed explanation of the question from the user.

# your input:
        - question: the question asked by the user.
        - details: the detailed explanation of the question from the user.
        - response 1: the first generated response to the user's question.
        - response 2: the second generated response to the user's question.

# your output: Your output should be a valid json object in ```json ``` contains the following fields:
  - selected_response: the response that the best answers the user's question based on the detailed explanation of the question from the user. This should be "response_1", "response_2", or "tie".

# question:
{question}
# details:
{details}
# response 1:
{response_1}
# response 2:
{response_2}

output: ``` json
```

Figure 11: Evaluation prompt used for assessing the quality of generated personalized responses with pairwise evaluation.

Evaluation Prompt

```
You are a fair and insightful judge with exceptional reasoning and analytical abilities. Your task is to evaluate a user's question, a generated response to that question, and an aspect that is important to the user. Based on this information, identify if the aspect is addressed in the generated response. Provide a clear and accurate assessment.
# your input:
        - question: the question asked by the user.
        - details: the detailed explanation of the question from the user.
        - response: a generated response to the user's question
        - aspect: the aspect that is important to the user, consisting of the following fields:
                - aspect: the title for the aspect.
                - reason: the reason that this aspect is important for the user.
                - evidence: the evidence from the user detailed explanation that the aspect extracted from.
# your output: Your output should be a valid json object in ```json ``` contains the following fields:
  - match_score: A score between 0 to 2 that indicates how well the generated response addresses this aspect, where: 0 means the response does not cover this aspect, 1 means the response somewhat covers this aspect, and 2 means the response covers this aspect very well.

# question:
{question}
# details:
{details}
# response:
{response}
# aspect:
- aspect: {aspect.aspect}
- reason: {aspect.reason}
- evidence: {aspect. evidence}
output: ``` json
```

Figure 12: Evaluation prompt used for assessing the quality of generated personalized responses using the extracted aspects.

Algorithm 1 Implementation of evaluation metric for the LaMP-QA benchmark.

Input: prompt x_u, requirement details r_u, important aspects E_{x_u}, generated response $\hat{y}_{x_u}$, Evaluator LLM π

Output: score $s_{\hat{y}_{x_u}}$

$s^t_{\hat{y}_{x_u}} = 0$ ▷ Score initialization with zero

for $e_i \in E_{x_u}$ **do** ▷ Scoring output based on each aspect

  $s^t_{e_i} = \pi(x_u, r_u, e_i, \hat{y}_{x_u})$ ▷ Scoring output based on aspect using prompt in Figure 12

  $s_{e_i} = \frac{s^t_{e_i}}{2}$ ▷ Normalizing the aspect score for aspect by division by 2

  $s^t_{\hat{y}_{x_u}} = s^t_{\hat{y}_{x_u}} + s_{e_i}$ ▷ Score accumulation for averaging

end for

$s_{\hat{y}_{x_u}} = \frac{s^t_{\hat{y}_{x_u}}}{|E_{x_u}|}$ ▷ Averaging the output score using division by the number of aspects

return $s_{\hat{y}_{x_u}}$ ▷ Returning score for output for user u

No-Personalization

```
You are a helpful assistant designed to generate personalized responses to user questions.  Your output should be a valid json object in ```json ``` block that contains the following fields:\n   - personalized_answer: contains the personalized answer to the user's current question.

# question:
{question}

output: ```json
```

Figure 13: Prompt used with LLMs that do not incorporate personalization.

RAG-Personalization

```
You are a helpful assistant designed to generate personalized responses to user questions. Your task is to answer a user's question from a post in a personalized way by considering this user's past post questions and detailed descriptions of these questions.
# Your input:
        - The user's current question from a post.
        - The user's past post questions and detailed descriptions of these questions.

# Your task: Answer the user's current question in a personalized way by considering this user's past post questions and detailed descriptions of these questions, to learn about the user's preferences.

# Your output: You should generate personalized answer to the user's current question by considering this user's past post questions and detailed descriptions of these questions to learn about user's preferences. Your output should be a valid json object in ```json ``` block that contains the following fields:
        - personalized_answer: contains the personalized answer to the user's current question considering the this user's past post questions and detailed descriptions of these questions to learn about user's preferences.

# Past post questions and detailed descriptions of these questions:
{profile}
# Current post question:
{question}

output: ``` json
```

Figure 14: Prompt used with LLMs that personalize output using RAG.

ment of LLMs, we leverage the VLLM library.[9] For the retriever, we use Contriever (Izacard et al.,

[9]Available at: https://github.com/vllm-project/vllm

Planning Prompt

```
You are a helpful assistant designed to generate the topics that user might expect to see in a response to their question. Your task is to extract the important aspects that the user expects to see in a response to their question considering the previous questions asked by the same user and the detailed information need they provided in the post.

# Your input:
    - The user's current question.
    - The user's past post questions and detailed descriptions of these questions.

# Your task: Extract the important aspects that the user expects to see in a response to their question considering the previous questions asked by the same user and the detailed information need they provided in the post.

# Your output: You should generate a list of aspects that are important for the user to be included in the response to their question.

# Past post questions and detailed descriptions of these questions:
{profile}
# Current post question:
{question}

output:
```

Generation with Plan

```
You are a helpful assistant designed to generate personalized responses to user questions. Your task is to answer a user's question from a post in a personalized way by considering this user's past post questions and detailed descriptions of these questions. Additionally, you are provided with the aspects that the user expects to see in the response to their question, which you can use to generate a personalized answer.

# Your input:
    - The user's current question from a post.
    - The user's past post questions and detailed descriptions of these questions.
    - The aspects that the user expects to see in the response to their question.

# Your task: Answer the user's current question in a personalized way by considering this user's past post questions and detailed descriptions of these questions, to learn about the user's preferences. Additionally, you should consider the aspects that the user expects to see in the response to their question.

# Your output: You should generate personalized answer to the user's current question by considering this user's past post questions and detailed descriptions of these questions to learn about user's preferences. Additionally, you should consider the aspects that the user expects to see in the response to their question. Your output should be a valid json object in ```json ``` block that contains the following fields:
    - personalized_answer: contains the personalized answer to the user's current question considering the this user's past post questions and detailed descriptions of these questions to learn about user's preferences.

# Past post questions and detailed descriptions of these questions:
{profile}
# Current post question:
{question}
# Aspects expected in the response:
{plan}

output: ``` json
```

Figure 15: Prompt used with LLMs that personalize output using `PlanPers`.

2022), a dense retrieval model fine-tuned on the MS MARCO dataset (Bajaj et al., 2018), to re-

Method	Arts & Entertainment	Lifestyle & Personal Development	Society & Culture	Average (macro)
Gemma 2 Instruct (9B)				
No Personalization	0.2025	0.3874	0.3973	0.3290
RAG-Personalization (Random P)	0.1960	0.3330	0.3340	0.2876
RAG-Personalization (Asker's P_u)	0.3260	0.4569	0.4765	0.4198
PlanPers	**0.3768**†	**0.4857**†	**0.5408**†	**0.4677**†
Qwen 2.5 Instruct (7B)				
No Personalization	0.3419	0.4687	0.4566	0.4224
RAG-Personalization (Random P)	0.2547	0.3789	0.3829	0.3388
RAG-Personalization (Asker's P_u)	0.3822	0.4679	0.4909	0.4470
PlanPers	**0.3890**	**0.5051**†	**0.5181**†	**0.4707**†
GPT 4o-mini				
No Personalization	0.3923	0.5175	0.5072	0.4723
RAG-Personalization (Random P)	0.3016	0.4205	0.4044	0.3743
RAG-Personalization (Asker's P_u)	0.4357	0.4960	0.5179	0.4832
PlanPers	**0.4789**†	**0.5684**†	**0.5885**†	**0.5452**†

Table 3: Performance on the validation set. † shows a statistically significant difference between the best-performing baseline and the others using t-test ($p < 0.05$).

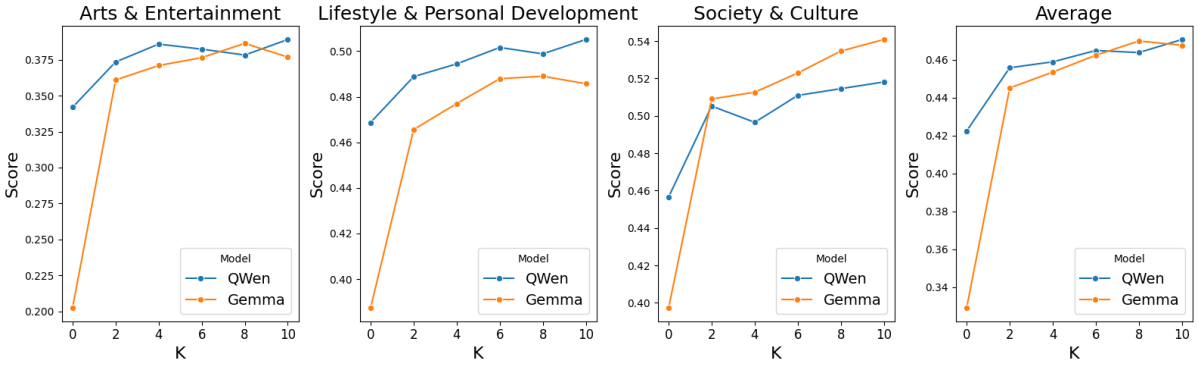

Figure 16: Effect of number of retrieved items from the user's profile on the performance of Plan-RAG Personalization on the validation set.

trieve $k = 10$ relevant items from the user profile P_u, unless otherwise specified.

F Experiments on validation set

The results of the baselines on the validation set of the LaMP-QA benchmark are reported in Table 3. These results confirm the findings discussed in Section 5.2 on the test set, demonstrating consistent trends across both the test and validation sets. Additionally, the effect of varying the number of retrieved items from the user profile on the performance of Plan-RAG Personalization is shown in Figure 16. These results mirror the findings discussed in Section 5.2 for the test set, reinforcing that incorporating more personalized context from the user profile leads to improved performance on the validation set.

G AI Assistance Usage

We used ChatGPT[10] as a writing assistant. Specifically, initial drafts of certain paragraphs were paraphrased using ChatGPT, after which manual revisions were applied before inclusion in the paper.

[10] https://chat.openai.com/