

Learning from Natural Language Feedback for Personalized Question Answering

Alireza Salemi
University of Massachusetts Amherst
Amherst, MA, USA
asalemi@cs.umass.edu

Hamed Zamani
University of Massachusetts Amherst
Amherst, MA, USA
zamani@cs.umass.edu

Abstract

Personalization is crucial for enhancing both the effectiveness and user satisfaction of language technologies, particularly in information-seeking tasks like question answering. Current approaches for personalizing large language models (LLMs) often rely on retrieval-augmented generation (RAG), followed by reinforcement learning with scalar reward signals to teach models how to use retrieved personal context. We believe that these scalar rewards sometimes provide weak, non-instructive feedback, limiting learning efficiency and personalization quality. We introduce VAC, a novel framework for personalized response generation that replaces scalar rewards with natural language feedback (NLF) that are generated conditioned on the user profiles and the question narratives. NLF serves as a rich and actionable supervision signal, allowing the policy model to iteratively refine its outputs and internalize effective personalization strategies. Training alternates between optimizing the feedback model and fine-tuning the policy model on the improved responses, resulting in a policy model that no longer requires feedback at inference. Evaluation on the LaMP-QA benchmark that consists of three diverse domains demonstrates consistent and significant improvements over the state-of-the-art results. Human evaluations further confirm the superior quality of the generated responses. These results demonstrate that NLF provides more effective signals for optimizing personalized question answering.

CCS Concepts

• Information systems → Personalization; Users and interactive retrieval; • Computing methodologies → Natural language generation; Learning from implicit feedback.

Keywords

Personalized Question Answering, Natural Language Feedback, Retrieval-Augmented Generation, Personalization, User Modeling, Iterative Training, Feedback-based Learning

ACM Reference Format:

Alireza Salemi and Hamed Zamani. 2026. Learning from Natural Language Feedback for Personalized Question Answering. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '26)*, July 20–24, 2026, Melbourne, VIC, Australia. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3805712.3809747>



This work is licensed under a Creative Commons Attribution International 4.0 License.

SIGIR '26, Melbourne, VIC, Australia

© 2026 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-2599-9/2026/07
<https://doi.org/10.1145/3805712.3809747>

1 Introduction

Personalization has become a critical component in human-centered systems such as search [4, 6, 16, 36], recommendation [18, 19, 21], and text generation [12, 25, 28, 30], as it enhances user satisfaction, increases engagement, and improves overall system efficiency [35]. By tailoring outputs to individual user preferences and contexts, personalized systems can deliver more relevant and effective interactions [12, 27]. Previous work on personalized text generation has primarily focused on content generation [12, 27–29], which is fundamentally different from information seeking. In content generation, the objective is to mimic the user’s writing style and preferences, whereas in information seeking, the primary goal is to deliver personalized relevant information to the user. In the context of information seeking, e.g., question answering, personalization is particularly valuable, as it enables generation of responses that align with the user’s intent, background, and preferences, resulting in more accurate, relevant, and user-specific responses [30].

Previous work on personalizing LLMs has mainly relied on retrieval-augmented generation (RAG) [28, 30], where personalized information is retrieved from a user profile and appended to the prompt to guide the model’s output. To optimize this RAG pipeline for personalization, various approaches have been explored. For instance, when ground-truth outputs are available for a user, the model can be trained to generate these outputs conditioned on the retrieved personalized context [28]. However, such labeled data is often unavailable across many users and tasks. To address this, reinforcement learning has been employed to further enhance the model’s ability to incorporate personalized information, typically by using personalized scalar rewards—learned through a personalized reward model or derived from user-provided explanations or rubrics—that reflect the quality of generated responses [27].

These optimization methods face several limitations. First, the ground-truth output provided for a user represents only one of many potentially acceptable responses, making supervised training on a single target suboptimal and prone to local minima [27]. Additionally, in the case of reinforcement learning, scalar rewards provide relatively weak supervision—they indicate whether an output is good or bad but lack actionable feedback on how to improve. Consequently, the model must infer effective adjustments without explicit guidance. Moreover, optimizing with scalar rewards often requires exploration across a wide range of outputs, leading to slow convergence and increased training cost.

To address these challenges, we introduce VAC,¹ a framework that replaces the scalar reward model with a feedback model that generates natural language feedback (NLF) on the policy model’s

¹In Sanskrit, Vāc is the personified goddess of speech, language, and wisdom. VAC also stands for Verbal-Alignment for Customization (or personalization).

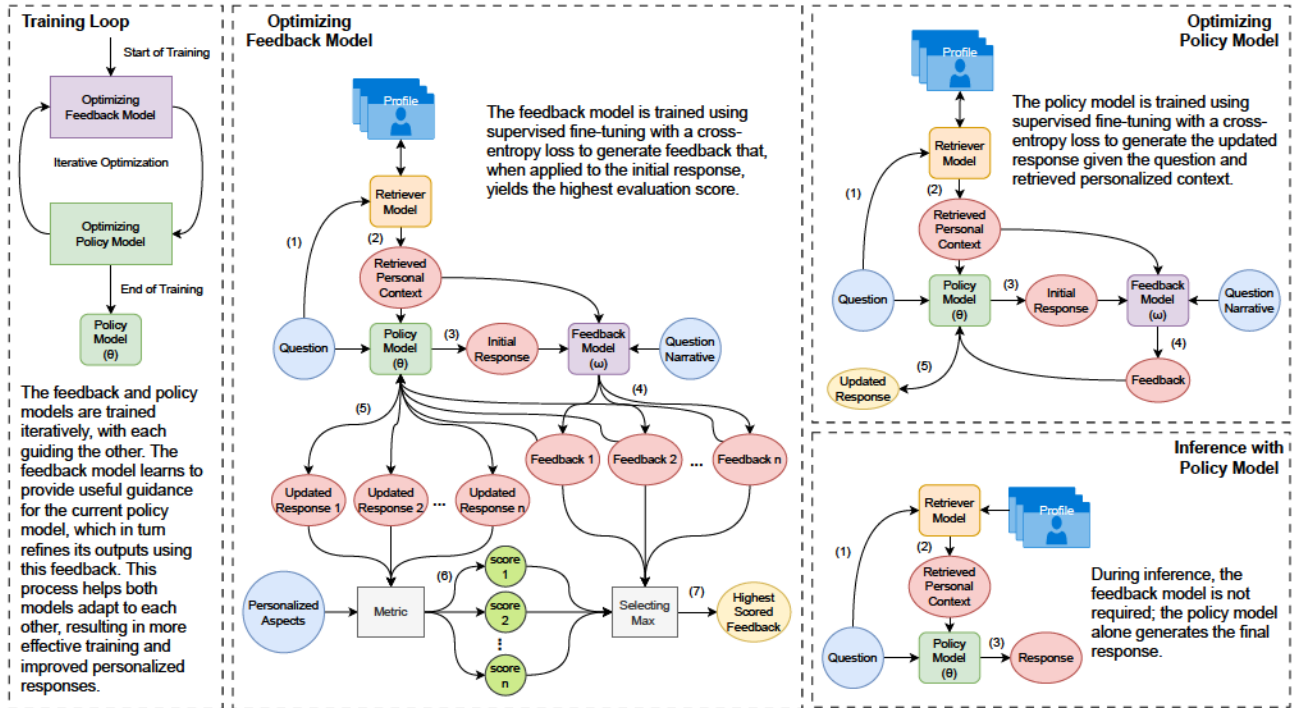


Figure 1: Overview of the training loop and inference process in the VAC framework, illustrating the interaction between the feedback model and policy model during training, and the use of the policy model at inference time.

responses during training. The NLF is generated based on personalized user preferences derived from the user profile and a user-authored question narrative. This feedback provides a richer and more interpretable form of supervision, guiding the policy model toward producing more personalized outputs. Training in VAC, as depicted in Figure 1, follows an iterative process over several rounds. In each iteration, the feedback model is first optimized to generate feedback that, when used to revise a response by the current policy model, leads to measurable improvements according to a task-specific evaluation metric. Once trained, the feedback model generates feedback for the responses produced by the current policy model. This feedback is then used to guide the policy model in editing its initial responses to improve them for the user. Finally, the policy model is fine-tuned via supervised learning to generate the improved responses directly from the input, eliminating the need for feedback at inference time. Compared to scalar rewards, which offer only coarse and indirect supervision signals, natural language feedback provides explicit and actionable guidance. Training the feedback model on outputs generated by the current policy model allows it to adapt to the model’s evolving behavior and capabilities, resulting in more targeted and effective feedback. Conversely, training the policy model on feedback-refined responses enables it to internalize effective personalization patterns, improving its generation quality without relying on feedback at test time.

To evaluate VAC, we conduct experiments on the recent Language Model Personalized Question Answering benchmark (LaMP-QA)

[30], which includes three diverse domains. Our results demonstrate that VAC consistently outperforms all baselines, achieving a 13.6% relative improvement over the non-personalized baseline, a 3.6% improvement over the best-performing personalized baseline while being 1.9× more efficient in terms of inference time, and a 6.0% improvement over reinforcement learning with scalar rewards. Additionally, human evaluation shows that VAC is preferred in 44% of the cases, ties in 33%, and is less preferred in only 23% of comparisons against the state-of-the-art method. Furthermore, we provide detailed ablation studies analyzing various aspects of the proposed method, including the impact of different optimization strategies and the size of the feedback model on overall performance. To facilitate future research, we publicly release our code and data.²

2 Related Work

Personalization. Personalization plays a central role in search, recommendation, and text generation [5, 12, 21, 27, 28, 36], as it has been shown to improve user satisfaction, efficiency, and long-term engagement [26, 30]. Personalization is particularly beneficial for question answering, as it enables models to generate responses that are better aligned with the user’s preferences, background, and prior knowledge, ultimately leading to more relevant and effective answers [30]. In this work, we focus on personalized question answering, a setting for which, to the best of our knowledge, LaMP-QA [30] is the only publicly available benchmark.

²Available at: <https://github.com/alirezasaalemi7/VAC>

To personalize an LLM, Salemi et al. [28] proposed a RAG framework that retrieves information from the user profile and incorporates it into the prompt provided to the LLM. Furthermore, Salemi et al. [27] extend this approach by optimizing the LLM with reinforcement learning to better incorporate retrieved personal context. Beyond this line of work, existing methods for personalization span a range of strategies, including training retrievers with personalized relevance feedback [25], fine-tuning LLMs with user-specific supervision [10], and designing prompts tailored to individual users [13]. Parameter-efficient fine-tuning has also been explored for personalized generation [33], with recent efforts integrating these techniques into RAG pipelines [29]. In addition, reasoning and self-training have shown promise in improving long-form personalized generation [27]. Personalized assistants have been studied across a variety of domains, including education and enterprise applications [14, 17, 20, 40]. Despite interest in personalized generation, personalized question answering remains relatively underexplored.

Learning from Natural Language Feedback. Due to the expressive nature of language, NLF has been explored as a training signal in tasks such as mathematical reasoning and code generation, where ground truth answers are well-defined [2, 3]. These studies demonstrate that human-written NLF can substantially improve model performance, while feedback generated by other LLMs tends to be less effective. At inference time, NLF has also been used in collaborative setups, where two models jointly solve a task through iterative feedback [15, 22, 34, 37, 38]. Another line of work leverages NLF at inference to optimize prompts rather than model parameters [39], though such methods typically introduce latency at test time.

This paper provides the first attempt on using NLF for personalization. Our work departs from priors in several key ways. First, unlike domains such as math and code where answers are either correct or incorrect, personalization requires learning subjective user preferences, where a response may be suitable for one user but not for another. Second, we automatically generate feedback conditioned on the information about the user, removing the need for human supervision during training. Third, our method operates within a RAG framework, where both the retrieval and generation components contribute to the policy and feedback models' performance. Finally, we propose a joint training procedure that alternates between optimizing the feedback and policy model, allowing them to co-adapt and resulting in more effective learning.

3 Problem Formulation

We consider a setting in which a user u is associated with a profile $P_u = \{d_i^u\}_{i=1}^{|P_u|}$, consisting of their previous questions and corresponding detailed descriptions. Given a new query x_u , an LLM π_θ generates a personalized response $y_{x_u} = \pi_\theta(P_u, x_u)$ by conditioning on both P_u and x_u . To evaluate the quality of the generated response, we assume access to a set of n_{x_u} user-specific aspects $E_{x_u} = \{e_i\}_{i=1}^{n_{x_u}}$, which are extracted from a personalized question narrative r_{x_u} provided by the user. These aspects are used exclusively for evaluation and are not accessible to the policy model during generation. A metric $\mu(x_u, \hat{y}_u, E_{x_u}, r_{x_u})$ quantifies the quality of the response based on the extent to which the expected aspects are addressed in the generated output. Since the aspects are explicitly derived from user-provided requirements, this evaluation

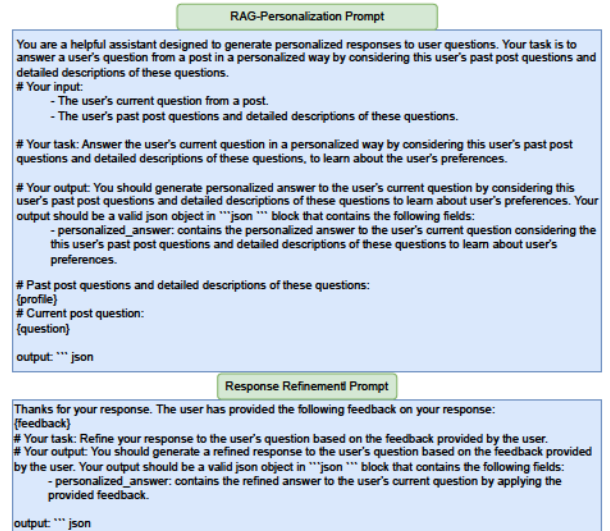


Figure 2: Prompts used for response generation in RAG-Personalization, Offline RL Personalization, and VAC (top), and for response refinement in VAC (bottom).

framework enables a targeted assessment of how well the response aligns with the user's personalized information needs. Finally, we assume the existence of a training dataset $D = \{(x_i, P_i, E_{x_i}, r_{x_i})\}_{i=1}^{|D|}$ that our method can learn from. In the test set, the structure is identical, but the aspects and narratives are used exclusively for evaluation and are never provided to the policy model.

4 The VAC Framework

As discussed in Section 1, optimizing the policy model π_θ with scalar rewards is limited in effectiveness, as these rewards offer only coarse feedback indicating overall output quality, without specifying how the output should be improved. Consequently, the model must explore and infer appropriate adjustments on its own, which slows convergence and increases the cost of training.

To address these challenges, we propose the VAC framework which replaces the scalar reward model with a feedback model ϕ_ω that generates natural language feedback (NLF) on the outputs of the policy model to guide it toward more personalized responses during training. The framework follows an iterative training procedure in which the feedback and policy models are alternately optimized over multiple rounds. This enables the feedback model to improve its ability to generate effective, personalization-oriented feedback to guide the policy model to generate more personalized responses, while the policy model progressively learns to produce more personalized responses without relying on feedback during inference. The remainder of this section details our proposed method.

Overview of the Training Pipeline: The overview of the VAC's iterative training loop is illustrated in Figure 1 and Algorithm 1. Over T iterations, each iteration t begins by optimizing the feedback model ϕ_{ω_t} using offline reinforcement learning to learn an effective feedback strategy that improves the personalized responses of the policy model $\pi_{\theta^{t-1}}$, based on the response distribution generated

by $\pi_{\theta^{t-1}}$. This optimization process can be formalized as:

$$\omega^t = \operatorname{argmax}_{\omega} \frac{1}{|D|} \sum_{d \in D} U_{\text{feedback}}(\phi_{\omega}; d; \pi_{\theta^{t-1}}) \quad (1)$$

where D is the training dataset and U_{feedback} is a objective function that measures the utilization of the feedback generated by the feedback model for the policy model $\pi_{\theta^{t-1}}$ based on the how it improves the evaluation metric μ , as will be described in Section 4.2.

Once the feedback model trained, in the same iteration, the feedback model ϕ_{ω^t} is used to produce feedback for outputs generated by $\pi_{\theta^{t-1}}$, and the policy model is asked to revise its responses accordingly. The updated responses are then used to fine-tune a new policy model π_{θ^t} using supervised learning, allowing it to better personalize its responses for users in subsequent iterations without relying on feedback at inference time. This is formalized as:

$$\theta^t = \operatorname{argmax}_{\theta} \frac{1}{|D|} \sum_{d \in D} U_{\text{policy}}(\pi_{\theta}; d; \phi_{\omega^t}) \quad (2)$$

where D is the training dataset and U_{policy} is a objective function for the policy model that encourages the model using supervised learning to generated the updated response after applying feedback from feedback model ϕ_{ω^t} without relying on feedback during inference, as will be described in Section 4.2.

The details of the feedback generation and response refinement process are provided in Section 4.1 and the optimization procedures for both the policy and feedback models are described in Section 4.2.

Overview of the Inference Pipeline: For inference, as shown in Figure 1 (Inference w/ policy model), we adopt the same RAG pipeline as Salemi and Zamani [30], where for a question x_u from user u with profile P_u , a retriever R selects K relevant documents from P_u . These retrieved documents are then appended to the question to form the prompt and fed into the trained policy model π_{θ^t} to generate the response using the prompt shown in Figure 2 (top), formalized as $y = \pi_{\theta^t}(x_u; R(x_u, P_u, K))$.

4.1 Feedback Generation & Output Refinement

Given an initial personalized response $y_{t-1} = \pi_{\theta^{t-1}}(x_u; R(x_u, P_u, K))$ to a question x_u for user u in iteration t generated by the previous iteration policy model $\pi_{\theta^{t-1}}$, this section outlines the procedure for generating NLF and refining the response accordingly.

Feedback Generation. To generate feedback for the initial personalized response y_{t-1} in iteration t , we first retrieve K documents from the user profile P_u using the retriever R , conditioned on the question x_u . Then, given the question, the retrieved documents, the initial response, and the question narrative r_{x_u} for the user u , we prompt the feedback model ϕ_{ω^t} to analyze the response and produce NLF aimed at improving personalization of the response, guided by the narrative. The prompt used for this process is illustrated in Figure 3. Formally, the feedback is defined as: $f = \phi_{\omega^t}(x_u; R(x_u, P_u, K); r_{x_u}; y_{t-1})$.

Output Refinement. To refine the initial output of iteration t denoted as y_{t-1} using feedback f , we append the initially retrieved documents from the user profile with the initial response y_{t-1} in iteration t and the feedback, and prompt the policy model $\pi_{\theta^{t-1}}$ to revise its response based on this information. This process is

Feedback Model Prompt

```
You are a helpful assistant designed to generate feedback for the generated response to a user's question, considering the user's detailed information need and the aspects that the user expects to see in the response to their question. Your task is to provide actionable feedback on how to improve the generated response based on the user's detailed information need and the aspects that the user expects to see in the response to their question.
# Your input:
- The user's current question.
- The user's past post questions and detailed descriptions of these questions.
- The generated response to the user's question.
- The detailed information need that the user provided in the post for this question.
# Your task: Provide actionable feedback on how to improve the generated response based on the user's detailed information need and the aspects that the user expects to see in the response to their question.
# Your output: Your output should be a valid json object in ""json"" block that contains the following fields:
- feedback: contains the feedback on how to improve the generated response based on the user's detailed information need and the aspects that the user expects to see in the response to their question.
# Past post questions and detailed descriptions of these questions:
(profile)
# Current post question:
(question)
# Generated response:
(generated_response)
# Detailed information need:
(question_narrative)
output: "" json
```

Figure 3: Prompt used with the feedback model in the VAC framework to generate NLF on the initial output.

illustrated in the prompt shown in Figure 2 (bottom) and is formally defined as: $y_t = \pi_{\theta^{t-1}}(x_u; R(x_u, P_u, K); y_{t-1}; f)$.

4.2 Feedback & Policy Model Optimization

This section describes training objectives for the feedback and policy model, which are optimized iteratively as shown in Algorithm 1.

Optimizing the Feedback Model. This section defines the U_{feedback} objective function. The primary goal of the feedback model ϕ_{ω^t} at iteration t is to generate feedback that effectively improves the personalized responses produced by the previous policy model $\pi_{\theta^{t-1}}$, so that the improved responses can provide useful supervision for training the next policy model π_{θ^t} . Accordingly, the optimization objective of the feedback model should be designed to encourage the generation of feedback that, when used by the policy model to revise its initial output, leads to a measurable improvement in the quality and personalization of the response.

To optimize the feedback model ϕ_{ω^t} , we follow a self-training approach as outlined in Algorithm 1 (lines 2–12). For each input example in the dataset, we first generate an initial personalized response using the policy model from the previous iteration, $\pi_{\theta^{t-1}}$. Then, using the previous iteration feedback model $\phi_{\omega^{t-1}}$, we generate N diverse feedback candidates by sampling at a high temperature (temperature = 1). Each candidate feedback is then applied independently to the response y_{t-1} using $\pi_{\theta^{t-1}}$ to produce a revised response. These revised responses are subsequently evaluated using the personalized rubrics provided in the dataset, based on the downstream task-specific metric (as explained in Section 5.1). The feedback that leads to the highest evaluation score is selected as the most effective one (line 10 in Algorithm 1). This selected feedback is then used as a supervision signal to train the current feedback model ϕ_{ω^t} using supervised fine-tuning with a cross-entropy loss [32] (line 12 in Algorithm 1). The model is trained to generate this effective feedback given the question, the initial response, the retrieved documents, and the question narrative. This optimization encourages ϕ_{ω^t} to generate feedback that, when used to revise the policy model's output, results in measurable improvements according to the task-specific evaluation metric. In this way, the objective

Algorithm 1 Implementation of the training loop in the VAC framework.

Input: policy model π_{θ^0} , feedback model ϕ_{ω^0} , retriever R , dataset D , metric μ , number of training iterations T , number of retrieved documents K , number of generated feedback N

Output: trained policy model π_{θ^T} , trained feedback model ϕ_{ω^T}

```

1: for  $t = 1$  until  $T$  do
2:   // training the feedback model  $\phi_{\omega^t}$  for round  $t$ 
3:    $D_{\phi_{\omega^t}} = \{\}$  ▷ This round's training data for feedback model
4:   for  $(x_u, P_u, E_{x_u}, r_{x_u}) \in D_{\text{train}}$  do ▷ For each input in training dataset
5:      $y_{t-1} = \pi_{\theta^{t-1}}(x_u; R(x_u, P_u, K))$  ▷ Generate initial output
6:      $F = \{\}$  ▷ Set of feedbacks for this specific output
7:     for  $j = 1$  until  $N$  do ▷ For  $N$  times
8:        $F = F \cup \{\phi_{\omega^{t-1}}(x_u; R(x_u, P_u, K); r_{x_u}; y_{t-1})\}$  ▷ Generate a feedback using feedback model with a high temperature for the
generated output
9:     end for
10:     $D_{\phi_{\omega^t}} = D_{\phi_{\omega^t}} \cup \{(x_u, y_{t-1}, P_u, r_{x_u}, f) | \underset{f \in F}{\operatorname{argmax}} \mu(x_u, \pi_{\theta^{t-1}}(x_u; R(x_u, P_u, K)); y_{t-1}; f), E_{x_u}, r_{x_u})\}$  ▷ Find the feedback that
maximizes the metric when applied to the previous generated output and add it to the training set
11:   end for
12:    $\omega^t = \underset{\omega}{\operatorname{argmax}} \sum_{(x_u, y_{t-1}, r_{x_u}, f) \in D_{\phi_{\omega^t}}} \log p_{\omega}(f | x_u; R(x_u, P_u, K); r_{x_u}; y_{t-1})$  ▷ Maximize the probability of generating feedback given
the generated output, input, and personalized aspects
13:   // training the policy model  $\pi_{\theta^t}$  for round  $t$ 
14:    $D_{\pi_{\theta^t}} = \{\}$  ▷ This round's training data for policy model
15:   for  $(x_u, P_u, E_{x_u}, r_{x_u}) \in D_{\text{train}}$  do ▷ For each input in training dataset
16:      $y_{t-1} = \pi_{\theta^{t-1}}(x_u; R(x_u, P_u, K))$  ▷ Generate initial output
17:      $f = \phi_{\omega^t}(x_u; R(x_u, P_u, K); r_{x_u}; y_{t-1})$  ▷ Generate a feedback using the optimized feedback model
18:      $y_t = \pi_{\theta^{t-1}}(x_u; R(x_u, P_u, K); y_{t-1}; f)$  ▷ Apply the generated feedback to the previous generated output
19:      $D_{\pi_{\theta^t}} = D_{\pi_{\theta^t}} \cup \{(x_u, P_u, y_t)\}$  ▷ Add the updated output to the training set
20:   end for
21:    $\theta^t = \underset{\theta}{\operatorname{argmax}} \sum_{(x_u, P_u, y_t) \in D_{\pi_{\theta^t}}} \log p_{\theta}(y_t | x_u; R(x_u, P_u, K))$  ▷ Maximize the probability of generating the updated output given the
inputs and retrieved personal documents
22: end for
23: return  $\pi_{\theta^T}, \phi_{\omega^T}$  ▷ Return the fully trained policy and feedback model

```

function U_{feedback} is formulated such that optimizing it results in feedback that is effective in improving the policy model's outputs.

Optimizing the Policy Model. This section defines the U_{policy} objective function. As shown in Algorithm 1 (lines 13–21), the optimization of the policy model proceeds as follows: For each input in the training dataset, the policy model from the previous iteration, $\pi_{\theta^{t-1}}$, first generates an initial response y_{t-1} to the input query. Next, the feedback model ϕ_{ω^t} , trained in the current iteration to align itself with the policy model $\pi_{\theta^{t-1}}$, produces a feedback signal for each initial response. This feedback is then used by $\pi_{\theta^{t-1}}$ to revise its initial output, producing an improved response (line 18 in Algorithm 1). Finally, the updated policy model π_{θ^t} is trained using supervised fine-tuning with a cross-entropy loss [32], learning to generate the improved response directly from the input query (line 21 in Algorithm 1). This design of the objective function U_{policy} assumes that the policy model's updated response after applying feedback is a better personalized response for the user. Consequently, the model is trained to reproduce this response directly, without requiring feedback. This training procedure helps the policy model learn to produce higher-quality, personalized responses without relying on feedback at inference time.

5 Experiments

5.1 Experimental Setup

Datasets & Evaluation: We conduct our experiments on the only publicly available dataset for personalized question answering, the LaMP-QA benchmark [30], which comprises three diverse domains: (1) Art & Entertainment, (2) Lifestyle & Personal Development, and (3) Society & Culture.³ Each example in these datasets includes a user query, the user's question history serving as their profile, a question narrative that reflects the user's perspective and intent, and a set of personalized rubrics that specify the aspects that an ideal response should address. The statistics of the datasets used in our experiments are shown in Table 1. To evaluate responses, following Salemi and Zamani [30], we employ the instruction-tuned

³LaMP [28] and LongLaMP [12] are other available datasets for personalized text generation; however, they are not well-suited for our setting for two main reasons. First, their primary focus is on content generation tasks, which differs fundamentally from our focus on question answering as explained in Section 1. Second, unlike LaMP-QA—which provides explicit annotations of user expectations as ground-truth—they include only a single reference response. This design limits their ability to represent the diversity of valid personalized outputs and risks misalignment with real user expectations, as noted by Salemi and Zamani [30]. Such limitations are especially restrictive for our feedback-provision method, where converging to a single response is both infeasible and suboptimal for robust training and fair evaluation.

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

Attribute	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

Qwen 2.5 model with 32 billion parameters⁴ [23]. For each question, the LLM assesses whether each personalized aspect is addressed in the response, assigning a score in the range [0, 2]. The scores are then normalized to the range [0, 1]. The final score for a generated response is computed as the average normalized score across all personalized aspects. For more information about this we refer the reader to Salemi and Zamani [30]. Additionally, in one experiment, we confirm our main experimental result through side-by-side human evaluation between VAC and the best baseline.

Training Configuration: We use the instruction-tuned Qwen 2.5 model with 7 billion parameters⁵ [23] as the policy LLM. Extending our approach to larger or alternative LLMs would require over 750 GPU hours, which is beyond our computational budget; thus, we limit our experiments to a single LLM, which constitutes a limitation of this work. Unless otherwise specified, the feedback model uses the same backbone LLM (instruction-tuned Qwen 2.5 model with 7 billion parameters). We conduct training for $T = 3$ iterations. For training feedback provider, we generate $N = 16$ feedback per output. Each iteration uses the best checkpoint from the previous iteration to initialize the models' weights. Training is performed using the Adam optimizer [11] with a learning rate of 5×10^{-5} , a batch size of 32, and gradient clipping with a maximum norm of 1 for stability. Models are trained for up to 5000 steps with a warmup phase over the first 10% of steps, followed by linear learning rate decay. We fine-tune the models using Low-Rank Adaptation (LoRA) [8], with rank $r = 16$, scaling factor $\alpha = 16$, and a dropout rate of 0.05, applied without modifying bias parameters. LoRA is implemented using the PEFT library.⁶ Model checkpoints are evaluated every 250 steps on the validation set, and the best-performing checkpoint is selected for evaluation. All experiments are conducted using 4 NVIDIA A100 GPUs with 80GB of VRAM and 128GB of RAM, completed in around 750 GPU hours.

Inference Configuration: All models are configured with a maximum input-output token limit of 8192 tokens. Response generation is performed using nucleus sampling [7] with a temperature of 0.1. To enable efficient inference and deployment of LLMs, we utilize the VLLM library.⁷ For retrieval, we employ Contriever [9], a dense retriever fine-tuned on the MS MARCO dataset [1], to retrieve $k = 10$ relevant documents from the user profile.

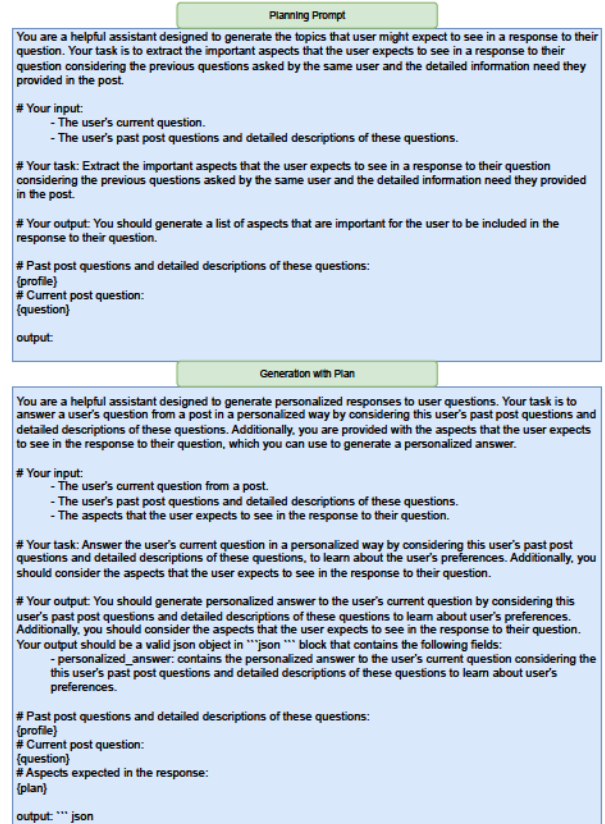


Figure 4: Prompt used for PlanPers [30] baseline.

Baselines: We compare VAC against a set of personalized and non-personalized baselines. For the non-personalized baseline, we directly provide the question to the LLM without any user context. For personalized baselines, we include the following:

- RAG-Personalization [28, 30]: The question is used to retrieve relevant documents from the user profile, which then the LLM generates a response using both the query and the retrieved personal context with the prompt shown in Figure 2 (top).
- RAG with Random User Profiles [30]: Similar to the previous method, but retrieval is performed on randomly sampled user profiles instead of the actual user profile. This baseline assesses the impact of using mismatched user information.

⁴Available at: <https://hf.co/Qwen/Qwen2.5-32B-Instruct>

⁵Available at: <https://hf.co/Qwen/Qwen2.5-7B-Instruct>

⁶Available at: <https://github.com/huggingface/peft>

⁷Available at: <https://github.com/vllm-project/vllm>

Table 2: Performance on the test set of the LaMP-QA benchmark. The superscript \dagger shows a statistically significant difference between the best-performing baseline and our method using t-test ($p < 0.05$).

Method	Runtime (second / query)	Arts & Entertainment	Lifestyle & Personal Development	Society & Culture	Average (macro)
No-Personalization	0.78	0.3129	0.4582	0.4769	0.416
RAG-Personalization (Random P)	1.65	0.2547	0.3829	0.4037	0.3471
RAG-Personalization	1.71	0.3397	0.4481	0.4967	0.4281
PlanPers	3.12	0.3518	0.4818	0.5240	0.4525
Offline RL RAG-Personalization	1.67	0.3579	0.4621	0.5070	0.4423
VAC	1.63	0.3454	0.5288\dagger	0.5331\dagger	0.4691\dagger

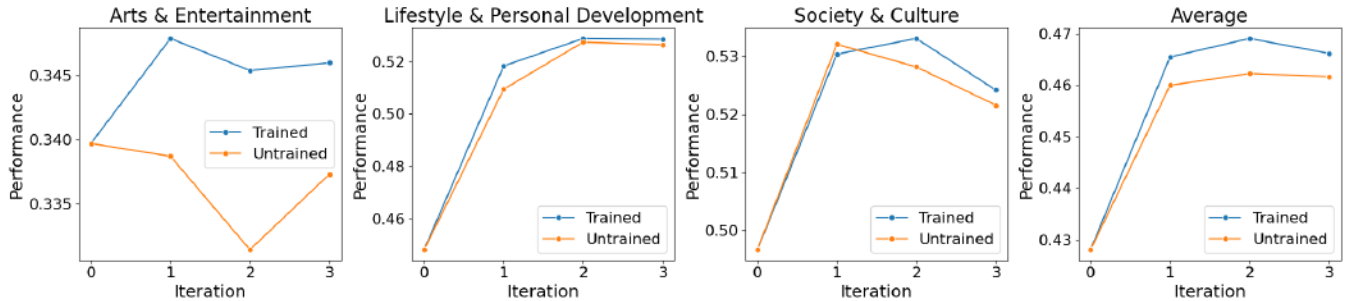


Figure 5: Performance of VAC in different training iterations with trained and untrained feedback model.

- PlanPers [30]: This method first retrieves information from the user profile using the question, then generates a high-level response plan based on the documents and the question. Conditioned on the plan, the retrieved documents, and the question, the LLM generates the final personalized response. This method uses the prompts shown in Figure 4 for plan and response generation. We refer the reader to Salemi and Zamani [30] for details.
- Offline RL RAG-Personalization: To compare with an approach that leverages scalar reward signals for training personalized LLMs, we implement Expectation-Maximization Offline Reinforcement Learning [27, 31], using the downstream evaluation metric as the scalar reward. This algorithm is used for training due to the similarity of its training loop to that of our proposed method, making it a fair basis for comparison between learning from NLF and scalar reward. Similar to RAG-Personalization baseline, this method begins by retrieving a set of documents from the user profile. Based on the retrieved content, a set of 16 candidate responses is generated for each question using the prompt shown in Figure 2 (top). The downstream task metric that scores outputs based on the question narrative and personalized rubrics is then applied to these responses, and the one with the highest score is selected to supervise the next iteration of training. The model is trained for three iterations under the same configuration as our method, with the best checkpoint from all iterations used for evaluation. This comparison enables an empirical assessment of the efficiency and effectiveness of NLF-based optimization versus scalar reward-based optimization.

All baselines are evaluated under the same setup and conditions as VAC, including identical configurations for maximum input and

output lengths, training budget, number of retrieved documents, retrieval model, and generation temperature.

5.2 Empirical Results

Comparison with the Baselines: The results of our method and the baselines on the LaMP-QA benchmark datasets are presented in Table 2. As shown, VAC statistically significantly outperforms all baselines in terms of average performance across the benchmark. More specifically, VAC achieves statistically significant improvements over all baselines in 2 out of the 3 personalized question answering tasks. The only task where VAC does not outperform the baselines is Art & Entertainment. These results highlight the effectiveness of learning from natural language feedback for improving personalization in response generation based on user preferences.

Table 2 reports the runtime for each method. Among them, the non-personalized LLM yields the lowest runtime, primarily because it processes shorter inputs and incurs no retrieval overhead. In contrast, all RAG-based personalization methods—including VAC—have higher runtime due to the added cost of retrieving relevant user profile documents. The highest runtime is observed for the PlanPers baseline [30], which is nearly twice as slow as VAC due to two step generation method used in this method, yet yields significantly lower performance. Overall, these results demonstrate that VAC provides superior personalization performance with runtime costs comparable to the most efficient personalized baselines.

Effect of Optimizing Feedback Model: To examine the effect of training the feedback model on the performance of VAC, we conduct two sets of experiments: one in which both the policy and feedback models are updated after each iteration, and another in which the feedback model remains frozen while only the policy

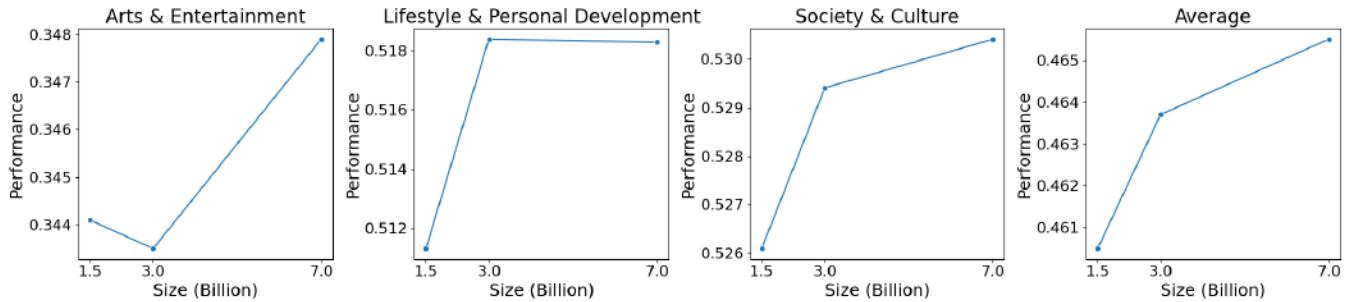


Figure 6: Effect of feedback model size on the performance of VAC.

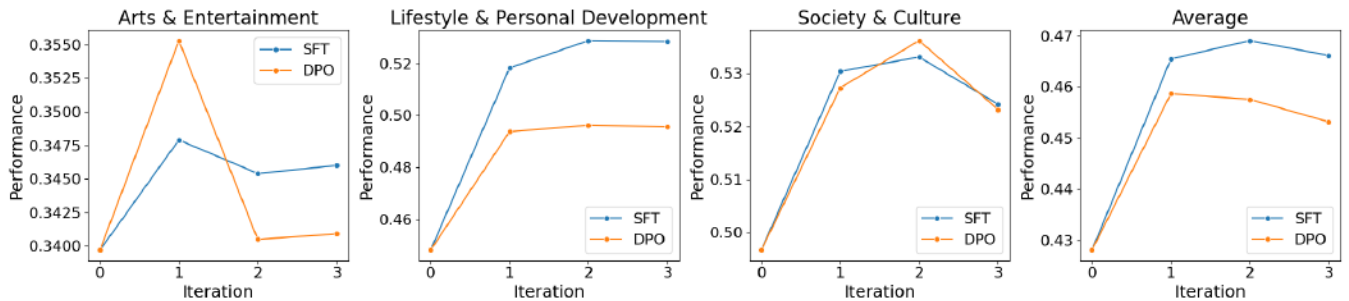


Figure 7: Effect of optimization method for policy model in VAC in different training iterations.

model is trained. The results are reported in Figure 5. As shown, jointly training the feedback model to align with the evolving policy model consistently outperforms the frozen-feedback setup across all datasets. These findings highlight the importance of optimizing the feedback model in each iteration to match the updated capabilities of the policy model, thereby enabling the generation of more effective and informative feedback.

Effect of Number of Training Iterations (T): To investigate the impact of the number of training iterations (T) on the performance of VAC, we train the model for up to three iterations and evaluate it after each iteration. The results in Figure 5 show that the performance improves during the first two iterations but plateaus in the third. It also indicates that this plateau effect is more pronounced when the feedback model is untrained, highlighting the importance of optimizing the feedback model on the performance. These observations suggest that continued training with VAC yields diminishing returns after a few iterations, and that joint optimization of both the policy and feedback models is crucial for maximizing effectiveness.

Effect of Feedback Model Size: To examine how the size and capabilities of the feedback model—which plays a central role in guiding the policy model during training—affect its effectiveness in helping the policy model to learn user preferences, we conduct experiments using instruction-tuned Qwen 2.5 of different parameter sizes: 1.5 billion, 3 billion, and 7 billion. These experiments are carried out over a single training iteration, and the results are presented in Figure 6. As shown, larger feedback models consistently lead to better performance of the policy model, suggesting that increased capacity enables the feedback model to generate more informative and actionable feedback. These findings underscore the

importance of using strong feedback providers to more effectively supervise and guide the learning process of the policy model.

Effect of Optimization Method for Policy Model: As described in Section 4.2, once the updated output is generated using feedback, we train the policy model to reproduce this updated output via supervised fine-tuning (SFT) using a cross-entropy loss. An alternative approach is to optimize the model using methods such as Direct Preference Optimization (DPO) [24], which maximizes the likelihood of the updated output while minimizing the likelihood of the initial output. To investigate the effectiveness of this alternative, we perform a comparative experiment and report the results in Figure 7. The results show that, with the exception of the first iteration on the Art & Entertainment dataset, SFT performs on par with or significantly better than DPO. We hypothesize that this outcome stems from a key assumption in DPO: it treats the initial and updated outputs as distinctly contrasting pairs. However, as training progresses, the policy model begins to generate outputs that are already close to the updated ones. In such cases, enforcing a strong separation between the two—as DPO does—may unnecessarily distort the model’s output distribution. In contrast, SFT more naturally accommodates these subtler improvements, allowing the model to refine its outputs without overcorrecting.

Human Evaluation: We conduct a human evaluation to compare VAC with the best-performing baseline, PlanPers [30], from human perspective. We randomly sampled 100 examples from the LaMP-QA benchmark. Each example was independently evaluated by two human annotators, who were asked to compare the responses based on the rubric aspects and question narrative, and to determine which response better addressed the question—or if the responses



Figure 8: Case study illustrating the initial response, feedback, and updated response (top row), and a comparison between VAC and the best-performing baseline, PlanPers [30] (bottom row).

were equally good. The inter-annotator agreement, measured using Cohen’s kappa, is 0.7832, indicating a high level of consistency between annotators. The results of this evaluation are presented in Figure 9. In 44% of the cases, VAC was preferred by the annotators

for better addressing the rubric aspects. In 23% of the cases, the PlanPers baseline was favored. The remaining 33% were judged as ties. These demonstrate that VAC produces responses that are more

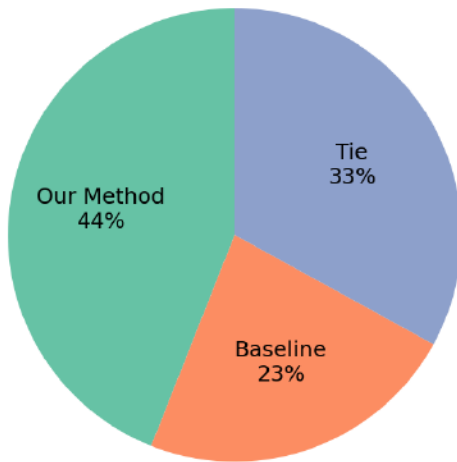


Figure 9: Results of human evaluation between VAC and the best performing baseline, PlanPers [30].

aligned with user-specific rubric aspects from human perspective, indicating its effectiveness in personalized question answering.

5.3 Case Study

This section presents case studies of initial responses, feedback, and updates during training, along with post-training outputs from the policy model compared to the top baseline.

Initial response, feedback, and updated response during training: As shown in Figure 8 (top row), the initial response to the user’s question about determining when turkey stock is done cooking was generally informative but lacked structure and omitted key contextual details from the user’s narrative—particularly the focus on slow stovetop simmering and the influence of different bone types on cooking duration. The feedback addressed these shortcomings by recommending the inclusion of concrete doneness indicators (e.g., flavor, consistency, clarity), specific advice for stovetop cooking, and considerations for bone variability. It also suggested organizing the response into clear, distinct sections to enhance clarity. The updated response effectively incorporates these recommendations by outlining clear signs of doneness, offering practical monitoring strategies, and providing estimated cooking times based on bone type. Presented in a well-structured format, the revised response better reflects the user’s context and yields a more personalized, informative, and user-aligned response.

Comparing VAC and the best-performing baseline after training: As shown in Figure 8 (bottom row), VAC delivers a more tailored response than PlanPers by directly addressing the user’s specific context and aligning more closely with the personalized evaluation rubrics. While PlanPers provides general guidance on refinishing exterior trim, VAC goes further by incorporating details that reflect the user’s concerns—such as previous issues with paint durability, uncertainty about the trim material, and questions about the suitability of Sherwin-Williams Emerald paint. The response includes specific advice on surface preparation, compatibility with

composite materials, and environmental factors like local climate. This leads to higher scores across rubric dimensions including Material Compatibility, Paint Recommendation, and Durability/Longevity, demonstrating that VAC more effectively captures user intent and provides context-aware, actionable guidance.

6 Conclusions and Future Work

We introduced VAC, a new framework for personalized response generation that replaces scalar rewards with natural language feedback as the primary supervision signal. By leveraging user-specific feedback grounded in both the user profile and question narrative, VAC provides informative and actionable guidance for training personalized LLMs. In contrast to conventional reward modeling pipelines that compress rich user preferences into a single scalar signal, our approach preserves the structure and semantics of feedback, enabling more fine-grained alignment between model behavior and user intent. Our iterative training loop—alternating between feedback generation and policy refinement—enables the policy model to internalize personalization strategies without requiring feedback at inference time, effectively amortizing the cost of feedback into the training phase. This design not only improves sample efficiency but also allows the model to generalize personalization patterns across users and contexts. Experimental results on LaMP-QA showed that VAC consistently outperforms existing personalized and non-personalized baselines and is also preferred by human evaluators, indicating that natural language feedback can serve as a scalable and expressive alternative to traditional supervision signals.

For future work, we plan to extend this feedback-based framework beyond response-level generation to include feedback over reasoning traces, enabling more personalized and transparent multi-step reasoning. Such an extension could allow the model to align not only its final outputs but also its intermediate decision-making process with user-specific preferences, potentially improving both interpretability and controllability. Additionally, we aim to apply this method to a broader range of personalization tasks beyond question answering, including dialogue systems, recommendation-oriented generation, and long-form content creation, where user intent is often more diffuse and evolving. We also plan to investigate its effectiveness on different classes of LLMs, including reasoning-focused models, as well as its interaction with emerging training paradigms such as tool use and agentic workflows. Finally, exploring the robustness of natural language feedback under noisy or partially observed user profiles, and understanding its role in mitigating biases in personalized systems, remain important open directions. These efforts will help assess the generality and adaptability of natural language feedback as a supervision mechanism for personalized text generation.

Acknowledgments

This work was supported in part by the Center for Intelligent Information Retrieval, in part by an award from Adobe Systems, Inc., in part by an award from Google, in part by NSF grant number #2143434, and with support from Google.org. Any opinions, findings and conclusions or recommendations in this material are those of the authors and do not necessarily reflect those of the sponsor.

References

- [1] 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. arXiv:1611.09268 [cs.CL] <https://arxiv.org/abs/1611.09268>
- [2] Angelica Chen, Jérémy Scheurer, Jon Ander Campos, Tomasz Korbak, Jun Shern Chan, Samuel R. Bowman, Kyunghyun Cho, and Ethan Perez. 2024. Learning from Natural Language Feedback. *Transactions on Machine Learning Research* (2024). <https://openreview.net/forum?id=xo3h15MwvU>
- [3] Angelica Chen, Jérémy Scheurer, Tomasz Korbak, Jon Ander Campos, Jun Shern Chan, Samuel R. Bowman, Kyunghyun Cho, and Ethan Perez. 2024. Improving Code Generation by Training with Natural Language Feedback. arXiv:2303.16749 [cs.SE] <https://arxiv.org/abs/2303.16749>
- [4] Zhicheng Dou, Ruihua Song, and Ji-Rong Wen. 2007. A large-scale evaluation and analysis of personalized search strategies. In *Proceedings of the 16th International Conference on World Wide Web (Banff, Alberta, Canada) (WWW '07)*. Association for Computing Machinery, New York, NY, USA, 581–590. doi:10.1145/1242572.1242651
- [5] 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 (Seoul, Republic of Korea) (CHI '15)*. Association for Computing Machinery, New York, NY, USA, 649–658. doi:10.1145/2702123.2702503
- [6] Qian Guo, Wei Chen, and Huaiyu Wan. 2021. AOLAPS: A Large-scale Data Set for Personalized Search. *Data Intelligence* 3, 4 (10 2021), 548–567. arXiv:https://direct.mit.edu/dint/article-pdf/3/4/548/1968580/dint_a_00104.pdf doi:10.1162/dint_a_00104
- [7] 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*. <https://openreview.net/forum?id=rygGQyrFvH>
- [8] 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*. <https://openreview.net/forum?id=nZevKeeFYf9>
- [9] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised Dense Information Retrieval with Contrastive Learning. *Transactions on Machine Learning Research* (2022). <https://openreview.net/forum?id=jKN1pX17b0>
- [10] 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. arXiv:2310.11564
- [11] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR* abs/1412.6980 (2014). <https://api.semanticscholar.org/CorpusID:6628106>
- [12] 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. arXiv:2407.11016 [cs.CL] <https://arxiv.org/abs/2407.11016>
- [13] 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, doi:10.1145/3589334.3645408
- [14] 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. arXiv:2308.07968
- [15] Yanyang Li, Michael R. Lyu, and Liwei Wang. 2025. Learning to Reason from Feedback at Test-Time. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (Eds.). Association for Computational Linguistics, Vienna, Austria, 5241–5253. doi:10.18653/v1/2025.acl-long.262
- [16] Jiongnan Liu, Zhicheng Dou, Guoyu Tang, and Sulong Xu. 2023. JDsearch: A Personalized Product Search Dataset with Real Queries and Full Interactions. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (Taipei, Taiwan) (SIGIR '23)*. Association for Computing Machinery, New York, NY, USA, 2945–2952. doi:10.1145/3539618.3591900
- [17] 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. arXiv:2405.04656
- [18] Hanjia Lyu, Song Jiang, Hanqing Zeng, Yinglong Xia, Qifan Wang, Si Zhang, Ren Chen, Chris Leung, Jiajie Tang, and Jiebo Luo. 2024. LLM-Rec: Personalized Recommendation via Prompting Large Language Models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, Kevin Duh, Helena Gomez, and Steven Bethard (Eds.). Association for Computational Linguistics, Mexico City, Mexico, 583–612. doi:10.18653/v1/2024.findings-naacl.39
- [19] Chunyan Mao, Shuaishuai Huang, Mingxiu Sui, Haowei Yang, and Xueshe Xue. 2024. Analysis and Design of a Personalized Recommendation System Based on a Dynamic User Interest Model. arXiv:2410.09923 [cs.IR] <https://arxiv.org/abs/2410.09923>
- [20] 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. arXiv:2311.09180
- [21] 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 Dzhalgakov, Andrey Mallevich, Ilya Cherniavskii, Yinghai Lu, Raghuraman Krishnamoorthi, Ansha Yu, Volodymyr Kondratenko, Stephanie Pereira, Xianjie Chen, Wenlin Chen, Vijay Rao, Bill Jia, Liang Xiong, and Misha Smelyanskiy. 2019. Deep Learning Recommendation Model for Personalization and Recommendation Systems. arXiv:1906.00091 [cs.IR]
- [22] Debjit Paul, Mete Ismayilzada, Maxime Peyrard, Beatriz Borges, Antoine Bosselut, Robert West, and Boi Faltings. 2024. REFINER: Reasoning Feedback on Intermediate Representations. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, Yvette Graham and Matthew Purver (Eds.). Association for Computational Linguistics, St. Julian's, Malta, 1100–1126. doi:10.18653/v1/2024.eacl-long.67
- [23] Qwen, , An Yang, Baosong Yang, Beichen Zhang, Binuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiayi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuyang Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 Technical Report. arXiv:2412.15115 [cs.CL] <https://arxiv.org/abs/2412.15115>
- [24] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. In *Thirty-seventh Conference on Neural Information Processing Systems*. <https://openreview.net/forum?id=HPuSDXJaa9>
- [25] Alireza Salemi, Surya Kallumadi, and Hamed Zamani. 2024. 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 (Washington DC, USA) (SIGIR '24)*. Association for Computing Machinery, New York, NY, USA, 752–762. doi:10.1145/3626772.3657783
- [26] Alireza Salemi, Julian Killingback, and Hamed Zamani. 2025. ExPerT: Effective and Explainable Evaluation of Personalized Long-Form Text Generation. In *Findings of the Association for Computational Linguistics: ACL 2025*, Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (Eds.). Association for Computational Linguistics, Vienna, Austria, 17516–17532. <https://aclanthology.org/2025.findings-acl.900/>
- [27] Alireza Salemi, Cheng Li, Mingyang Zhang, Qiaozhu Mei, Weize Kong, Tao Chen, Zhuowan Li, Michael Bendersky, and Hamed Zamani. 2025. Reasoning-Enhanced Self-Training for Long-Form Personalized Text Generation. arXiv:2501.04167 [cs.CL] <https://arxiv.org/abs/2501.04167>
- [28] Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. 2024. LaMP: When Large Language Models Meet Personalization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 7370–7392. doi:10.18653/v1/2024.acl-long.399
- [29] Alireza Salemi and Hamed Zamani. 2025. Comparing Retrieval-Augmentation and Parameter-Efficient Fine-Tuning for Privacy-Preserving Personalization of Large Language Models. In *Proceedings of the 2025 International ACM SIGIR Conference on Innovative Concepts and Theories in Information Retrieval (ICTIR) (Padua, Italy) (ICTIR '25)*. Association for Computing Machinery, New York, NY, USA, 286–296. doi:10.1145/3731120.3744595
- [30] Alireza Salemi and Hamed Zamani. 2025. LaMP-QA: A Benchmark for Personalized Long-form Question Answering. In *Proceedings of the The 2025 Conference on Empirical Methods in Natural Language Processing (to appear)*. Association for Computational Linguistics, Suzhou, China.
- [31] Avi Singh, John D Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J Liu, James Harrison, Jaehoon Lee, Kelvin Xu, Aaron T Parisi, Abhishek Kumar, Alexander A Alemi, Alex Rizkowsky, Azade Nova, Ben Adlam, Bernd Bohnet, Gamaleldin Fathy Elsayed, Hanie Sedghi, Igor Mordatch, Isabelle Simpson, Izzeddin Gur, Jasper Snoek, Jeffrey Pennington, Jiri Hron, Kathleen Kenalee, Kevin Swersky, Kshiteej Mahajan, Laura A Culp, Lechao Xiao, Maxwell Bileschi, Noah Constant, Roman Novak, Rosanne Liu, Tris Warkentin, Yamini Bansal, Ethan Dyer, Behnam Neyshabur, Jascha Sohl-Dickstein, and Noah Fiedel. 2024. Beyond Human Data: Scaling Self-Training for Problem-Solving

- with Language Models. *Transactions on Machine Learning Research* (2024). <https://openreview.net/forum?id=INAYUngGFK> Expert Certification.
- [32] 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* (Montreal, Canada) (*NIPS'14*). MIT Press, Cambridge, MA, USA, 3104–3112.
- [33] Zhaoxuan Tan, Zheyuan Liu, and Meng Jiang. 2024. Personalized Pieces: Efficient Personalized Large Language Models through Collaborative Efforts. arXiv:2406.10471
- [34] Zhiheng Xi, Dingwen Yang, Jixuan Huang, Jiafu Tang, Guanyu Li, Yiwen Ding, Wei He, Boyang Hong, Shihan Do, Wenyu Zhan, Xiao Wang, Rui Zheng, Tao Ji, Xiaowei Shi, Yitao Zhai, Rongxiang Weng, Jingang Wang, Xunliang Cai, Tao Gui, Zuxuan Wu, Qi Zhang, Xipeng Qiu, Xuanjing Huang, and Yu-Gang Jiang. 2024. Enhancing LLM Reasoning via Critique Models with Test-Time and Training-Time Supervision. arXiv:2411.16579 [cs.CL]. <https://arxiv.org/abs/2411.16579>
- [35] Yiyan Xu, Jinghao Zhang, Alireza Salemi, Xinting Hu, Wenjie Wang, Fuli Feng, Hamed Zamani, Xiangnan He, and Tat-Seng Chua. 2025. Personalized Generation In Large Model Era: A Survey. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (Eds.). Association for Computational Linguistics, Vienna, Austria, 24607–24649. <https://aclanthology.org/2025.acl-long.1201/>
- [36] Gui-Rong Xue, Jie Han, Yong Yu, and Qiang Yang. 2009. User Language Model for Collaborative Personalized Search. *ACM Trans. Inf. Syst.* 27, 2, Article 11 (mar 2009), 28 pages. doi:10.1145/1462198.1462203
- [37] Hao Yan, Saurabh Srivastava, Yintao Tai, Sida I. Wang, Wen-tau Yih, and Ziyu Yao. 2023. Learning to Simulate Natural Language Feedback for Interactive Semantic Parsing. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 3149–3170. doi:10.18653/v1/2023.acl-long.177
- [38] Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. 2024. Large Language Models as Optimizers. In *The Twelfth International Conference on Learning Representations*. <https://openreview.net/forum?id=Bb4VGOWELI>
- [39] Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Pan Lu, Zhi Huang, Carlos Guestrin, and James Zou. 2025. Optimizing generative AI by backpropagating language model feedback. *Nature* 639 (2025), 609–616.
- [40] 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)*, Kevin Duh, Helena Gomez, and Steven Bethard (Eds.). Association for Computational Linguistics, Mexico City, Mexico, 2386–2398. <https://aclanthology.org/2024.naacl-long.132>