

# Iterative Critique-Refine Framework for Enhancing LLM Personalization

Durga Prasad Maram<sup>1</sup>, Dhruvin Gandhi<sup>1</sup>, Zonghai Yao<sup>1</sup>, Gayathri Akkinapalli<sup>1</sup>,  
 Franck Dernoncourt<sup>2</sup>, Yu Wang<sup>3</sup>, Ryan A. Rossi<sup>2</sup>, Nesreen K. Ahmed<sup>4</sup>

University of Massachusetts Amherst<sup>1</sup>, Adobe Research<sup>2</sup>, University of Oregon<sup>3</sup>, Cisco AI Research<sup>4</sup>

## Abstract

Personalized text generation requires models not only to produce coherent text but also to align with a target user’s style, tone, and topical focus. Existing retrieval-augmented approaches such as LaMP and PGraphRAG enrich profiles with user and neighbor histories, but they stop at generation and often yield outputs that drift in tone, topic, or style. We present **PerFine**, a unified, training-free critique-refine framework that enhances personalization through iterative, profile-grounded feedback. In each iteration, an LLM generator produces a draft conditioned on the retrieved profile, and a critic LLM - also conditioned on the same profile - provides structured feedback on tone, vocabulary, sentence structure, and topicality. The generator then revises, while a novel knockout strategy retains the stronger draft across iterations. We further study additional inference-time strategies such as Best-of- $N$  and Topic Extraction to balance quality and efficiency. Across Yelp, Goodreads, and Amazon datasets, PerFine consistently improves personalization over PGraphRAG, with GEval gains of +7–13%, steady improvements over 3–5 refinement iterations, and scalability with increasing critic size. These results highlight that post-hoc, profile-aware feedback offers a powerful paradigm for personalized LLM generation that is both training-free and model-agnostic.

## 1 Introduction

Personalization is increasingly important for HCI, recommender systems, and natural language generation (Chen, 2023; Alhafni et al., 2024). Prior work on personalized text generation has largely relied on retrieval-augmented generation (RAG) from user profiles. However, the challenge extends beyond retrieving relevant history: generated text must also match a user’s *style*, tone, and topical focus. Benchmarks such as LaMP, LongLaMP,

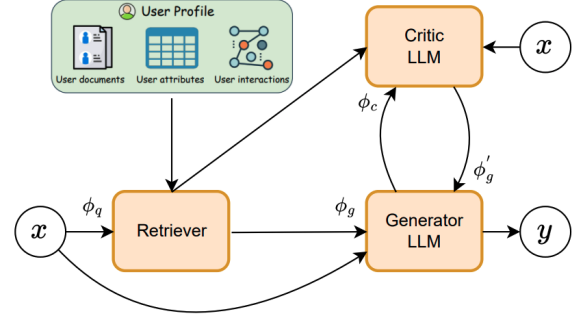


Figure 1: Overview of our framework for personalized text generation. User profile information is retrieved to guide the generator, whose outputs are iteratively critiqued and refined by PerFine, enabling multi-round personalization.

and PGraphRAG broaden the evaluation of personalization in long-text generation (Salemi et al., 2024; Au et al., 2025; Kumar et al., 2024). More recently, graph-based retrieval has been proposed to address the cold-start problem where users have sparse profiles, by leveraging neighbor user profiles in user-centric knowledge graphs (Au et al., 2025). Yet existing personalization methods primarily optimize what to retrieve and how many few-shot samples to include. They often lack a *post-hoc, profile-aware refinement* step that aligns both style and content once the full draft has been generated. We conjecture that such a refinement stage, enabled by a *critic LLM*, is crucial: it allows evaluation of the entire output against the user profile, correction of tone drift and topical gaps that retrieval cannot address, and avoidance of local heuristics from stepwise scoring. A profile-aware critic also acts as a model-agnostic oracle at inference. This makes iterative refinement a promising direction for personalization compared to pure retrieval-based or one-shot prompting, as it decouples retrieval from alignment and enables feedback to reshape the final text.

Personalized text generation is particularly chal-

lenging (Wan et al., 2025; Zhang et al., 2024a): it requires capturing nuanced writing styles and implicit preferences, retrieving relevant content, and tailoring that content to align with those preferences. A natural way to address this complexity is through **iterative refinement**, inspired by how humans revise their writing based on feedback. Automated feedback has already proven effective in addressing LLM errors such as hallucinations, unfaithful reasoning, and biased content (Pan et al., 2024), either during training, during generation, or through post-hoc correction. Post-hoc correction is especially suitable for personalization, since it evaluates the *entire* draft after generation and provides comprehensive feedback beyond stepwise scoring. Building on this paradigm, we introduce an external iterative refinement framework that generates structured feedback on **style** and **content**—the two core dimensions of personalized text generation.

We propose **PerFine**, a training-free iterative refinement framework that operates entirely at inference time. Figure 1 illustrates the architecture. We first retrieve top- $k$  entries from a user-item graph using GraphRAG. A generator LLM produces an initial draft  $y_0$ . A critic LLM, conditioned on the same profile, returns feedback along four dimensions: tone consistency, vocabulary match, sentence structure, and topic relevance. The generator revises accordingly, while a novel *knockout* strategy retains the stronger draft across iterations. The loop treats style and content as first-class constraints and is compatible with any base model that accepts natural language feedback.

Beyond this basic *PerFine* loop, we explore inference-time extensions that trace the quality-efficiency frontier (Figures 3, 4). *Topic Extraction* distills the profile into compact style and content hints. *Best-of- $N$* , layered on Knockout, samples multiple revisions per iteration and allows the critic to select the best candidate, yielding the highest quality at higher token cost. Balancing quality and efficiency, *PerFine with Knockout* is our default.

We address three questions. First, does a profile-grounded critique-refine loop improve personalized text generation compared to state-of-the-art methods such as LaMP and PGraphRAG? Second, how do critic scale and inference strategy trade off quality against efficiency? Third, how do improvements evolve across refinement iterations, and when do they plateau?

Our results show consistent gains. On

GEval, PerFine outperforms PGraphRAG by +10.25% (Yelp), +7.8% (Goodreads), and +13.41% (Amazon). METEOR also improves across datasets (0.180→0.195 on Yelp, 0.206→0.216 on Goodreads, 0.190→0.204 on Amazon). Gains accumulate over 3–5 refinement rounds before leveling off. Larger critics provide monotonic improvements (e.g., Yelp GEval climbs from 0.441 to 0.470 when scaling from 14B to 32B). Among inference-time strategies, Best-of- $N$  variant yields the strongest quality, while Knockout variant balances performance and efficiency.

## Contributions.

- We frame personalized text generation as *profile-grounded, post-hoc critique and refinement*, separating retrieval from alignment.
- We introduce **PerFine**, a training-free iterative refinement framework for personalized text generation.
- We propose inference-time strategies (Knockout, Topic Extraction, Knockout + Best-of- $N$ ) that enable flexible trade-offs between quality and efficiency.
- We provide an empirical study across three real-world datasets, showing stable gains within a few iterations, monotonic benefits from larger critics, and practical trade-offs that keep the method training-free and model-agnostic.

## 2 Proposed Framework

**Problem Definition:** Given an input  $x$  provided by a user  $u$ , and a target output generation  $y$ , the goal of the *personalized text generation* problem is to generate  $\tilde{y}$  that is aligned to the style and content of  $y$ , conditioned on the user’s personal data  $P_u$  (e.g., documents, histories, preferences). This is achieved by transforming the input  $x$  into a personalized input  $\tilde{x}$  using query ( $\phi_q$ ) and generation ( $\phi_g$ ) functions, along with a retriever module  $R$  with an optional parameter  $k$ , before passing it to a text generation module, an LLM denoted as Generator.

$$\begin{aligned}\tilde{x} &= \phi_g(x, R(\phi_q(x), P_u, k)) \\ \tilde{y} &= \text{Generator}(\tilde{x})\end{aligned}$$

**PerFine:** Our iterative refinement framework, as depicted in Figure 1, iteratively critiques and refines outputs to produce the final personalized

generation  $\tilde{y}$ . The framework comprises of the following core components: (1) Retriever, (2) Generator, and (3) Critic. The Retriever fetches the subset of user’s profile data most relevant to user query. The Generator produces the initial draft and refines the outputs based on feedback in subsequent rounds, while the Critic generates feedback. Both the Generator and Critic components are LLMs and are conditioned on the user profile. We also introduce and elaborate various inference-time strategies on top of PerFine to enhance the potential of our framework.

**Retriever:** Since a user’s profile can be large, using it in its entirety may lead to high computational costs, over-reliance on an LLM’s ability to process long contexts, and the inclusion of irrelevant profile information. To mitigate these issues, we leverage the retrieval-augmented generation (RAG) paradigm (Lewis et al., 2021), which uses only the most relevant user-profile entries, controlled by the hyperparameter  $k$ , as the context.

$$\mathcal{R}(P_u) = R(\phi_q(x), P_u, k)$$

where  $R$  is the retriever and  $\mathcal{R}(P_u)$  represents the top- $k$  subset of the user profile  $P_u$  ( $\mathcal{R}(P_u) \subseteq P_u$ ). The query function  $\phi_q(x)$  is an identity function in our setting.

**Generator:** The generator LLM produces an initial draft  $y_0$  from the user query and the top- $k$  relevant profile entries, using the generation prompt construction function  $\phi_g$ . It also refines the output in further rounds based on the personalized feedback given by critic using the refinement prompt construction function  $\phi'_g$ .

$$y_0 = \text{Generator}(\phi_g(x, \mathcal{R}(P_u)))$$

$$y_{t+1} = \text{Generator}(\phi'_g(x, y_t, f_t, \mathcal{R}(P_u))), 0 \leq t < T$$

where  $y_t$  and  $f_t$  are the output and feedback in the  $t^{\text{th}}$  iteration, while  $T$  is the maximum feedback iteration count.

**Critic:** The Critic LLM, conditioned on  $x$  and  $\mathcal{R}(P_u)$  is instructed to provide feedback  $f_t$  on  $y_t$  along the following four personalization criteria:

- **Tone Consistency:** Evaluate whether the tone and sentiment align with the user’s writing style.

- **Vocabulary Match:** Evaluate whether the vocabulary level is consistent with the user’s lexicon.
- **Sentence Structure:** Evaluate if the sentence lengths, complexity, and syntactic structures align with that of the user’s style.
- **Topic Relevance:** Evaluate if the generated content is relevant to the query, free of off-topic information, and inclusive of important aspects.

$$f_t = \text{Critic}(\phi_c(x, y_t, \mathcal{R}(P_u))), 0 \leq t \leq T$$

where  $\phi_c$  is the feedback prompt construction function.

**PerFine Setting:** The generator and critic LLMs operate in a zero-shot setting. The refinement process stops when a predefined stopping criterion is met, which here is a fixed number of iterations  $T$ . In our setup, a user’s profile consists of their own history as well as interactions with other users, represented in a user-centric bipartite graph with users and items as partitions. This results in a user profile being a collection of text samples from both the user and their neighbors. We employ Graph-based Retrieval-Augmented Generation (GraphRAG), where retrieval is performed over the graph to extract the most relevant information.

For any user  $u$ , we define the user profile  $P_u$  as the set of previous texts written by user  $u$  (i.e.,  $\{(u, j) \in E\}$ ), and the set of texts written by other users  $v$  for the same items connected to user  $u$  (i.e.,  $\{(v, j) \in E \mid (u, j) \in E\}$ ) (Edge et al., 2024).

$$P_u = \{(u, j) \in E\} \cup \{(v, j) \in E \mid (v, j) \in E\},$$

$$\forall j \in I, \quad u, v \in U, \quad u \neq v$$

where  $U$  is the set of user nodes,  $I$  is the set of item nodes and  $E$  is the set of interaction edges. The prompt construction functions  $\phi_g$ ,  $\phi_c$ ,  $\phi'_g$ , and the query  $x$  template are shared in Appendix A.1.

## Inference-time strategies

We further explore the following three inference-time variants of PerFine.

**(1) PerFine + Knockout:** In this setting, after each generation, the critic LLM compares the output  $y_t$

Dataset	Method	METEOR	GEval
Yelp	LaMP	0.156	0.361
	PGraphRAG	0.180	0.400
	PerFine + Knockout	<b>0.195</b>	<b>0.441</b>
Goodreads	LaMP	0.193	0.444
	PGraphRAG	0.206	0.445
	PerFine + Knockout	<b>0.216</b>	<b>0.480</b>
Amazon	LaMP	0.183	0.382
	PGraphRAG	0.190	0.410
	PerFine + Knockout	<b>0.204</b>	<b>0.465</b>

Table 1: **Baseline Comparison.** The table compares the performance of **PerFine+Knockout** against two baselines: **LaMP**, where the user profile contains samples only from the user’s own history, and **PGraphRAG**, which also includes samples from the profile histories of neighboring users. The output of PGraphRAG is taken as the initial generation  $y_0$ , over which PerFine+Knockout performs the feedback refinement process. Llama-3.1-8B-Instruct is used as the generator LLM, and Qwen-2.5-14B-Instruct serves as the critic LLM.

at iteration  $t$  with that of the previous round  $y_{t-1}$ , and selects the more personalized output, which then proceeds to the next round of feedback and refinement. To determine which output is more personalized, critic is instructed to evaluate the alignment in terms of style and topical relevance, referencing the profile  $\mathcal{R}(P_u)$  and the query  $x$ .

$$y_t = \text{Critic}(\phi_k(x, y_t, y_{t-1}, \mathcal{R}(P_u))), t > 0$$

where  $\phi_k$  is the knockout prompt construction function (shared in Figure 9).  $y_t$  is followed by  $f_t$  and  $y_{(t+1)}$ .

For the default setting, we operate the critic in this Best-of- $T$  kind of framework, which retains the most personalized generation across iterations. The results are shown in Table 1.

**(2) PerFine + Knockout + Best-of-N:** In this setting, for every refinement step, we do Best-of- $N$  sampling and ask the critic to compare the  $n$  sampled revisions and select the more personalized output, which is then passed on to the next round of knockout, feedback, and refinement steps.

$$y_{(t+1)}^1, \dots, y_{(t+1)}^n = \text{Generator}(\phi'_g(x, y_t, f_t, \mathcal{R}(P_u)))$$

$$y_{(t+1)} = \text{Critic}(\phi_n(x, y_{(t+1)}^1, \dots, y_{(t+1)}^n, \mathcal{R}(P_u)))$$

where  $\phi_n$  is the Best-of- $N$  prompt construction function (shared in Figure 10).  $y_{(t+1)}$  is followed by knockout, feedback and refinement steps.

**(3) PerFine + Topic Extraction:** In this setting,

instead of conditioning the critic on raw text samples from the user profile, we extract personalized aspects along the dimensions of style and content, which are then used as context. The writing style is derived from the user’s history, while the content aspects are taken from the neighbors’ profile.

$$S = \text{Topic\_Extractor}(\phi_{ts}(\mathcal{R}(P_u)_U))$$

$$C = \text{Topic\_Extractor}(\phi_{tc}(\mathcal{R}(P_u)_N))$$

$$f_t = \text{Critic}(\phi_t(x, y_t, S, C))$$

where  $\mathcal{R}(P_u)_U$  has samples only from user’s history, while  $\mathcal{R}(P_u)_N$  has samples from neighbor’s history.  $S$  and  $C$  are the extracted style and content topics by a Topic\_Extractor module, an LLM. By leveraging an explicitly summarized representation of the personalized aspects, this approach enables the critic to reference the profile information more easily during feedback generation at each iteration, while also reducing the length of the input context. The extraction prompts  $\phi_{ts}, \phi_{tc}$  are shared in Figure 11. Critic prompt  $\phi_t$  is shared in Figure 12.

### 3 Experiments Design

#### 3.1 Dataset and Graph Construction

We evaluate our approach on the product review writing task using the AgentSociety Challenge dataset (Yan et al., 2025), which is a curated collection of user–item–review triplets (user, review, item) from Yelp, Amazon (Hou et al., 2024), and Goodreads (Wan and McAuley, 2018; Wan et al., 2019). In the graph representation, one partition of the bipartite graph represents users, while the other represents items (businesses, products, or books being reviewed). An edge corresponds to a review. For each dataset, we prepare development and test splits, each containing 2,500 randomly sampled users with no overlap. For each user, we randomly take a review and add it in the split for evaluation, with the remaining reviews forming the user’s profile history. We consider only users who have at least one profile entry.

For Amazon reviews (Hou et al., 2024), we consider the domains Industrial and Scientific, Musical Instruments, and Video Games. For Goodreads reviews (Wan and McAuley, 2018; Wan et al., 2019), we consider the domains Comics, Poetry, and Children’s Books. To describe a business in Yelp data, we consider the *city*, *state*, *attributes*, and *categories* fields. To describe Amazon products, we consider the *title*, *description*, and *categories* fields.



For Goodreads, we use the *title* and *description* fields from raw data. Reviews that are not in English are filtered out.

### 3.2 Metrics and Evaluation

We use both term-based matching metrics and LLM-as-a-Judge metrics. For term-based matching, we use METEOR, while for LLM-as-a-Judge, we use G-Eval (Liu et al., 2023b), in which an LLM is prompted to generate an absolute score based on the comparison between the generation and the ground-truth reference. Further, G-Eval computes the final score by averaging over the possible scores, weighted by the probabilities assigned to each score by the backbone LLM. Evaluation prompt is shared in Appendix A.3.

### 3.3 Experimental Setup

We chose Llama-3.1-8B-Instruct as our generator LLM. For the critic, we primarily use the Qwen-2.5-Instruct models in both 14B and 32B sizes in addition to experimenting with gpt-5-mini, which is our choice of the closed-source model. We also use the same Llama-3.1-8B-Instruct model as the critic to evaluate the approach in a self-refinement setting. We used vLLM with a maximum completion token limit of 512 and a temperature of 0.6 for both LLMs. The feedback iteration count  $T$  is set to 5. For G-Eval, we used Qwen-3-32B as the backbone LLM. G-Eval was run 20 times at a temperature of 1 to obtain the probability distribution over scores. We choose Contriever (Lei et al., 2023) as the retriever with top- $k$  set to 4, where a max of  $k$  entries are retrieved from each of the user’s and neighbor’s profiles. The same top- $k$  are used for baselines as well. For PerFine + Knockout + Best-of- $N$ , we choose  $n$  to be 3.

## 4 Results

### 4.1 Baseline Comparison

We compare our method with two personalized baselines. They are (1) LaMP (Salemi et al., 2024), in which the augmented information for RAG consists of only the target user’s history. (2) PGraphRAG (Edge et al., 2024), where the retrieval is performed to fetch information from both the target user and the neighbors (from the interaction graph).

Table 1 demonstrates that PerFine+Knockout consistently outperforms the baselines, achieving improvements of 10.25% on Yelp, 7.8% on

Goodreads, and 13.41% on Amazon in the GEval scores.

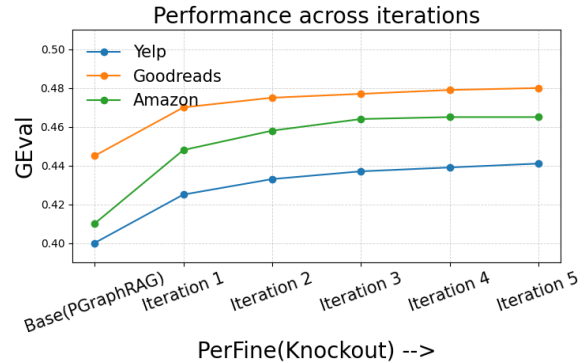


Figure 2: Performance across iterations on Yelp, Goodreads, and Amazon datasets. PerFine+Knockout starts from the PGraphRAG baseline and exhibits steady improvements, with gains plateauing after a few iterations.

The **performance trend across time** is shown in Figure 2. We observe that the scores increase with the number of iterations and converge after a few rounds of feedback, suggesting controlled and incremental alignments with the user profile over time. These evidences highlight the impact of incorporating feedback in enhancing the personalization capability of LLMs. A case study illustrating the end-to-end iterative refinement process is presented in Appendix A.4.

Dataset	Critic LLM	METEOR	GEval
Yelp	Llama-3.1-8B	0.200	0.408
	Qwen-2.5-7B	0.191	0.431
	Qwen-2.5-14B	0.195	0.441
	Qwen-2.5-32B	0.208	<b>0.470</b>
	gpt-5-mini	<b>0.211</b>	0.466
Goodreads	Llama-3.1-8B	0.211	0.451
	Qwen-2.5-7B	0.213	0.471
	Qwen-2.5-14B	0.216	0.480
	Qwen-2.5-32B	0.222	0.482
	gpt-5-mini	<b>0.230</b>	<b>0.500</b>
Amazon	Llama-3.1-8B	0.203	0.423
	Qwen-2.5-7B	0.200	0.447
	Qwen-2.5-14B	0.204	0.465
	Qwen-2.5-32B	0.209	0.480
	gpt-5-mini	<b>0.211</b>	<b>0.481</b>

Table 2: Evaluation of different critic models in the PerFine+Knockout setup (all are instruct models). Llama-3.1-8B-Instruct is the generator.

Dataset	Method	METEOR	GEval	# token (critic)	# token (gen)
Yelp	PerFine	0.185	0.402	10.77	9.54
	PerFine + Knockout	0.195	0.441	19.23	9.49
	PerFine + Knockout + Best-of- $N$	<b>0.197</b>	<b>0.449</b>	28.36	10.63
Goodreads	PerFine	0.209	0.453	11.65	10.57
	PerFine + Knockout	0.216	0.480	22	10.53
	PerFine + Knockout + Best-of- $N$	<b>0.219</b>	<b>0.483</b>	33.18	12.07
Amazon	PerFine	0.197	0.432	9.58	8.27
	PerFine + Knockout	0.204	0.465	17	8.23
	PerFine + Knockout + Best-of- $N$	<b>0.206</b>	<b>0.473</b>	25.28	9.36

Table 3: Comparison of performance and token usage (prompt+completion) across various inference strategies. We observe that the scores improve as we scale the inference time token usage. However, the additional gains from Best-of- $N$  sampling are marginal, while the associated token usage increases significantly. Token count is per query for 5 iterations in K. Llama-3.1-8B-Instruct is used as the generator, and Qwen-2.5-14B-Instruct is the critic.

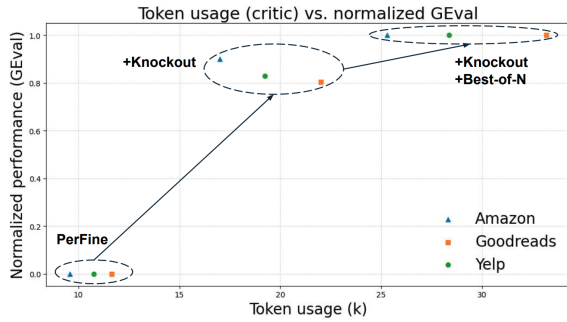


Figure 3: Visualization of the critic’s token usage (prompt + completion) vs normalized GEval performance on the Amazon, Goodreads, and Yelp datasets. Notably, PerFine+Knockout improves performance, while PerFine+Knockout+Best-of- $N$  achieves the highest scores, with increased token cost. Considering both efficiency and effectiveness, we ultimately select PerFine+Knockout.

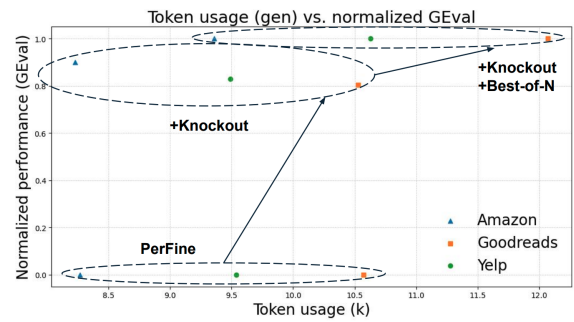


Figure 4: Figure showing generator’s token usage (prompt + completion) vs normalized GEval performance on the Amazon, Goodreads, and Yelp datasets. While the token usage for PerFine and PerFine+Knockout is similar, the token footprint increases for PerFine+Knockout+Best-of- $N$  due to the sampling of multiple revisions, while yielding only a marginal improvement in performance.

## 4.2 Ablation Studies

**Critic Model Ablation:** In Table 2, we present an ablation study by varying both the size and the family of the critic model. All experimental configurations outperform the baselines, demonstrating that the PerFine framework is compatible with a wide range of base models. For analyzing the impact of model size, we evaluate the Qwen-2.5 model in 7B, 14B, and 32B variants, and observe that performance consistently improves with larger critics. Larger critics provide more targeted feedback and refinement suggestions, thereby simplifying the generator’s incorporation of corrections. We also evaluate a self-refinement setting, in which the critic and generator are identical. This setting is resource-efficient, as it avoids the need for a sep-

arate LLM to serve as the critic, thereby reducing the overall memory footprint. Even in this configuration, PerFine achieves superior performance compared to the baselines.

The performance gain achieved with the closed-source model (gpt-5-mini) suggests that powerful proprietary models can be readily leveraged via APIs for generating feedback in PerFine framework. Furthermore, the observed gains with gpt-5-mini were obtained under a low reasoning efficiency setting, indicating that large reasoning models used as critics may be highly effective in the context of personalization and may lead to faster convergence.

**Inference-Strategies Ablation:** Table 3 compares the performance of different inference-

time strategies for the critic. The total token usage per query by the critic and generator is shown in Figure 3 and Figure 4, respectively. The PerFine+Knockout strategy, when combined with Best-of- $N$ , outperforms all other configurations. In both PerFine+Knockout and PerFine+Knockout+Best-of- $N$  settings, multiple candidate revisions are laid out for the critic to compare, either across iterations or within the same iteration, which increases the likelihood of producing an optimized output. However, this performance gain comes at the cost of higher token usage and increased latency, as shown in Figures 3 and 4. The spike in generator token usage for the PerFine+Knockout+Best-of- $N$  setting arises from sampling multiple revision outputs, while the spike in critic token usage arises from the step to select the best out of those  $N$ . The incremental improvement provided by Best-of- $N$  sampling is relatively small, and PerFine+Knockout offers a nice balance between performance and efficiency. The quality of PerFine+Topic\_Extraction setting is discussed in Appendix A.2.

## 5 Related Work

### 5.1 Personalization via RAG

RAG retrieves user-relevant segments from an external store with semantic similarity and then conditions generation on these segments (Gao et al., 2023). It is common in IR and recommendation pipelines (Zhao et al., 2024; Rajput et al., 2023; Di Palma, 2023; Wang et al., 2024). It can also reduce hallucinations by grounding outputs in factual context (Shuster et al., 2021; Li et al., 2024b). In personalization, large user profiles act as external knowledge, and a retriever picks a compact subset before decoding (Gao et al., 2023). Work in the survey groups retrievers into *sparse* and *dense* (Gao et al., 2023).

*Sparse retrieval* methods like TF-IDF and BM25 are efficient and strong baselines, yet they rely on lexical overlap and so miss semantic preference signals (Sparck Jones, 1972; Robertson et al., 1995; Salemi et al., 2023; Li et al., 2023a; Richardson et al., 2023).

*Dense retrieval* methods encode queries and documents into a continuous space for similarity search (Johnson et al., 2019), with off-the-shelf encoders such as Sentence-BERT (Reimers and Gurevych, 2019) and task-trained dual encoders like DPR (Karpukhin et al., 2020) and

Contriever (Izacard et al., 2021). In personalization, researchers also build training data and losses for user-centric retrieval, and they use fusion-at-decoder with encoder-decoder LMs; Sentence-T5 and GTR are widely used as well (Izacard and Grave, 2020; Raffel et al., 2020; Ni et al., 2021a,b; Mysore et al., 2023). Dense retrievers tend to perform better yet require careful data design and extra cost (Richardson et al., 2023). Hybrid and black-box retrieval also appear, but they add tool specificity and less transparency (Gao et al., 2023).

*Graph-based retrieval.* Traditional RAG can struggle under cold-start and fragmented histories. GraphRAG builds a user-item bipartite graph and expands the search space with neighbor profiles so it can find transferable evidence when the target user is sparse (Edge et al., 2024). This brings three advantages in our setting (Xiang et al., 2025). First, it increases topical coverage for long-form writing, since neighbors who reviewed the same item (j) supply complementary content cues. Second, it enhances style conditioning by allowing us to separate signals: we utilize the target user’s own texts for style and neighbors for query-relevant content, then merge them into the prompt. Third, it keeps control simple: we use only 1–2 hops and cap at most  $k$  texts from the user and at most  $k$  from neighbors, with Contriever to pick them (Lei et al., 2023). After generation, we fix any drift with a profile-conditioned critic and iterative revision. We take PGraphRAG’s output as the initial draft  $y_0$  and apply PerFine to turn “what was retrieved” into “how it reads,” showing stable gains under a fixed token budget (Edge et al., 2024; Salemi et al., 2023, 2024).

### 5.2 Personalization via Prompting

Personalization via prompting can be grouped into four categories: *contextual prompting*, *profile-augmented prompting*, *persona-based prompting*, and *prompt refinement* (Zhang et al., 2024b).

*Contextual prompting.* One can insert segments of a user’s demographic information, history, and item metadata into the prompt so the model performs downstream personalized tasks with context (Di Palma, 2023; Wang and Lim, 2023; Sanner et al., 2023; Li et al., 2023b; Christakopoulou et al., 2023; Tran et al., 2025b,a). This is simple and interpretable. It is also sensitive to prompt wording and scale when profiles are large or noisy, and context limits can be hit (Jin et al., 2024; Ding et al., 2024; Lin et al., 2024; Liu et al., 2023a).

*Profile-augmented prompting.* Many systems summarize, factorize, or hierarchically structure profiles, then inject distilled preferences back into the prompt to ease context bloat and cold-start. Examples include task-aware user summaries, topic/region distillation from browsing histories, and factorization prompting that turns preferences into structured attributes for downstream models (Richardson et al., 2023; Liu et al., 2024; Zheng et al., 2023; Wu et al., 2024). These improve signal density. They still run as one-shot generation with little ability to fix post-generation drift (Gao et al., 2023).

*Persona-based prompting.* A complementary line specifies an explicit persona in the prompt to guide style or behavior; representative variants span demographic, character, and individualized personas, but also introduce risks such as bias and “character hallucination” (Aher et al., 2023; Horton, 2023; Chen et al., 2024; Lim et al., 2024).

*Prompt refinement.* Beyond hand-crafted templates, some works iteratively optimize prompts to reduce manual trial-and-error in personalization, though they remain pre-generation interventions (Kim and Yang, 2024; Li et al., 2024a; Yao et al., 2024; Santurkar et al., 2023; Durmus et al., 2023).

We *do not* use persona prompts or learned prompt refiners. Instead, our method can be partially categorized as *contextual prompting* (we directly condition on profile snippets) and *profile-augmented prompting* (we distill and structure profile signals via retrieval). The key difference is that we place *natural-language feedback* at inference time and make it *profile-conditioned*: the critic returns structured feedback on tone, vocabulary, sentence structure, and topicality, the generator revises, and Knockout/Best-of- $N$  selects stronger drafts. This post-hoc alignment complements pre-generation prompting and directly corrects style/topic drift after a full draft is available.

### 5.3 Feedback for LLMs

Post-hoc feedback has proven effective in multiple domains. SELF-REFINE shows that a single model can generate, critique, and iteratively improve its own outputs without additional training (Madaan et al., 2023). REFLEXION stores verbalized reflections to improve subsequent decision making and long-horizon performance (Shinn et al., 2023). External critics decouple the target generator from the feedback provider and can be

trained to deliver actionable signals; examples include RL-based critics for feedback optimization and systems that supervise intermediate reasoning steps (REFINER) (Akyurek et al., 2023; Paul et al., 2024). Broader surveys categorize feedback channels (scalar vs. natural language), sources (self vs. external), and intervention points (during training, during generation, or post-hoc) (Pan et al., 2024); most reported gains concentrate on code, math, and QA, or other stepwise reasoning in different domains, where verification signals are available (Zhang et al., 2023a,b; Yao et al., 2025). Despite these advances, inference-time feedback for personalization has received limited attention. Prior work rarely operationalizes a profile-conditioned critic that evaluates full generations against a user’s style and content preferences and then drives iterative revision. Our framework fills this gap by (i) specifying four feedback dimensions tailored to personalization (tone, vocabulary, sentence structure, topicality), (ii) coupling them with inference-time selection strategies (Knockout, Best-of- $N$ ), and (iii) demonstrating training-free, model-agnostic improvements on profile-grounded generation beyond strong RAG baselines.

## 6 Conclusion

In this work, we introduce PerFine, an iterative refinement framework for personalized text generation. We demonstrate that a simple post-hoc personalized feedback methodology, which requires no training, can improve performance. We also introduce and experiment with various inference-time scaling mechanisms to enhance the critic’s potential, observing consistent performance gains. This work opens up promising research directions in the area of feedback for personalized LLMs.

### Limitations

One of the main limitations of our approach is the fixed number of feedback iterations. Different queries require different levels of refinement, and dynamically judging when to stop helps to avoid both overcorrection and undercorrection, while also reducing token usage and latency. In the PerFine framework, although smaller critics do improve performance, the most significant gains come from bigger LLMs. In resource-constrained deployment settings, this presents a challenge, making it crucial to train smaller LLMs as personalized critics for improved efficiency. However, obtaining



training signals for such supervision is difficult, and exploring synthetic data creation techniques may be helpful. Several possible extensions to PerFine can be explored to enhance its capabilities further. One promising direction is to make the retrieval process dynamic, where the top- $k$  profile is updated at every iteration based on feedback. In-context learning can also be effective in guiding the critic to structure its feedback in a specific way (through few-shot examples) tailored to the target task. Another important research direction is the reliable evaluation of personalized text generation, not only against ground-truth reference but also with respect to complex user preference patterns inferred from a user’s profile.

## References

- Gati V Aher, Rosa I Arriaga, and Adam Tauman Kalai. 2023. Using large language models to simulate multiple humans and replicate human subject studies. In *International conference on machine learning*, pages 337–371. PMLR.
- Afra Feyza Akyurek, Ekin Akyurek, Ashwin Kalyan, Peter Clark, Derry Tanti Wijaya, and Niket Tandon. 2023. [RL4F: Generating natural language feedback with reinforcement learning for repairing model outputs](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7716–7733, Toronto, Canada. Association for Computational Linguistics.
- Bashar Alhafni, Vivek Kulkarni, Dhruv Kumar, and Vipul Raheja. 2024. [Personalized text generation with fine-grained linguistic control](#). In *Proceedings of the 1st Workshop on Personalization of Generative AI Systems (PERSONALIZE 2024)*, pages 88–101, St. Julians, Malta. Association for Computational Linguistics.
- Steven Au, Cameron J. Dimacali, Ojasmita Pedirappagari, Namyong Park, Franck Dernoncourt, Yu Wang, Nikos Kanakaris, Hanieh Deilamsalehy, Ryan Rossi, and Nesreen K. Ahmed. 2025. [Personalized graph-based retrieval for large language models](#). *ArXiv*, abs/2501.02157.
- Jiangjie Chen, Xintao Wang, Rui Xu, Siyu Yuan, Yikai Zhang, Wei Shi, Jian Xie, Shuang Li, Ruihan Yang, Tinghui Zhu, and 1 others. 2024. From persona to personalization: A survey on role-playing language agents. *arXiv preprint arXiv:2404.18231*.
- Junyi Chen. 2023. [A survey on large language models for personalized and explainable recommendations](#). *Preprint*, arXiv:2311.12338.
- Konstantina Christakopoulou, Alberto Lalama, Cj Adams, Iris Qu, Yifat Amir, Samer Chucui, Pierce Vollucci, Fabio Soldo, Dina Bseiso, Sarah Scodel, and 1 others. 2023. Large language models for user interest journeys. *arXiv preprint arXiv:2305.15498*.
- Dario Di Palma. 2023. Retrieval-augmented recommender system: Enhancing recommender systems with large language models. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 1369–1373.
- Yiran Ding, Li Lina Zhang, Chengruidong Zhang, Yuanyuan Xu, Ning Shang, Jiahang Xu, Fan Yang, and Mao Yang. 2024. Longrope: Extending llm context window beyond 2 million tokens. *arXiv preprint arXiv:2402.13753*.
- Esin Durmus, Karina Nguyen, Thomas I Liao, Nicholas Schiefer, Amanda Askeel, Anton Bakhtin, Carol Chen, Zac Hatfield-Dodds, Danny Hernandez, Nicholas Joseph, and 1 others. 2023. Towards measuring the representation of subjective global opinions in language models. *arXiv preprint arXiv:2306.16388*.
- Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. [From local to global: A graph rag approach to query-focused summarization](#). *Preprint*, arXiv:2404.16130.
- Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2(1).
- John J Horton. 2023. Large language models as simulated economic agents: What can we learn from homo silicus? Technical report, National Bureau of Economic Research.
- Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. 2024. Bridging language and items for retrieval and recommendation. *arXiv preprint arXiv:2403.03952*.
- Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.
- Gautier Izacard and Edouard Grave. 2020. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282*.
- Hongye Jin, Xiaotian Han, Jingfeng Yang, Zhimeng Jiang, Zirui Liu, Chia-Yuan Chang, Huiyuan Chen, and Xia Hu. 2024. Llm maybe longlm: Self-extend llm context window without tuning. *arXiv preprint arXiv:2401.01325*.
- Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547.

- Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *EMNLP (1)*, pages 6769–6781.
- Jaehyung Kim and Yiming Yang. 2024. Few-shot personalization of llms with mis-aligned responses. *arXiv preprint arXiv:2406.18678*.
- Ishita Kumar, Snigdha Viswanathan, Sushrita Yerra, Alireza Salemi, Ryan Rossi, Franck Dernoncourt, Hanieh Deilamsalehy, Xiang Chen, Ruiyi Zhang, Shubham Agarwal, Nedim Lipka, and Hamed Zamani. 2024. *Longlamp: A benchmark for personalized long-form text generation*.
- Yibin Lei, Liang Ding, Yu Cao, Changtong Zan, Andrew Yates, and Dacheng Tao. 2023. *Unsupervised dense retrieval with relevance-aware contrastive pre-training*. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10932–10940, Toronto, Canada. Association for Computational Linguistics.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. *Retrieval-augmented generation for knowledge-intensive nlp tasks*. *Preprint*, arXiv:2005.11401.
- Cheng Li, Mingyang Zhang, Qiaozhu Mei, Weize Kong, and Michael Bendersky. 2024a. Learning to rewrite prompts for personalized text generation. In *Proceedings of the ACM Web Conference 2024*, pages 3367–3378.
- Cheng Li, Mingyang Zhang, Qiaozhu Mei, Yaqing Wang, Spurthi Amba Hombaiah, Yi Liang, and Michael Bendersky. 2023a. Teach llms to personalize—an approach inspired by writing education. *arXiv preprint arXiv:2308.07968*.
- Jiarui Li, Ye Yuan, and Zehua Zhang. 2024b. Enhancing llm factual accuracy with rag to counter hallucinations: A case study on domain-specific queries in private knowledge-bases. *arXiv preprint arXiv:2403.10446*.
- Xinyi Li, Yongfeng Zhang, and Edward C Malthouse. 2023b. A preliminary study of chatgpt on news recommendation: Personalization, provider fairness, fake news. *arXiv preprint arXiv:2306.10702*.
- Jung Hoon Lim, Sunjae Kwon, Zonghai Yao, John P Lalor, and Hong Yu. 2024. Large language model-based role-playing for personalized medical jargon extraction. *arXiv preprint arXiv:2408.05555*.
- Bin Lin, Chen Zhang, Tao Peng, Hanyu Zhao, Wencong Xiao, Minmin Sun, Anmin Liu, Zhipeng Zhang, Lanbo Li, Xiafei Qiu, and 1 others. 2024. Infinite-llm: Efficient llm service for long context with distillation and distributed kv-cache. *arXiv preprint arXiv:2401.02669*.
- Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023a. Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*.
- Qijiong Liu, Nuo Chen, Tetsuya Sakai, and Xiao-Ming Wu. 2024. Once: Boosting content-based recommendation with both open-and closed-source large language models. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 452–461.
- Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023b. *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.
- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. *Self-refine: Iterative refinement with self-feedback*. *Preprint*, arXiv:2303.17651.
- 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 preprint arXiv:2311.09180*.
- Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith B Hall, Daniel Cer, and Yinfei Yang. 2021a. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. *arXiv preprint arXiv:2108.08877*.
- Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Y Zhao, Yi Luan, Keith B Hall, Ming-Wei Chang, and 1 others. 2021b. Large dual encoders are generalizable retrievers. *arXiv preprint arXiv:2112.07899*.
- Liangming Pan, Michael Saxon, Wenda Xu, Deepak Nathani, Xinyi Wang, and William Yang Wang. 2024. *Automatically correcting large language models: Surveying the landscape of diverse automated correction strategies*. *Transactions of the Association for Computational Linguistics*, 12:484–506.
- Debjit Paul, Mete Ismayilzada, Maxime Peyrard, Beatriz Borges, Antoine Bosselut, Robert West, and Boi Faltings. 2024. *Refiner: Reasoning feedback on intermediate representations*. *Preprint*, arXiv:2304.01904.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text

- transformer. *Journal of machine learning research*, 21(140):1–67.
- Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, and 1 others. 2023. Recommender systems with generative retrieval. *Advances in Neural Information Processing Systems*, 36:10299–10315.
- Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.
- Chris Richardson, Yao Zhang, Kellen Gillespie, Sudipta Kar, Arshdeep Singh, Zeynab Raeesy, Omar Zia Khan, and Abhinav Sethy. 2023. Integrating summarization and retrieval for enhanced personalization via large language models. *arXiv preprint arXiv:2310.20081*.
- Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, and 1 others. 1995. *Okapi at TREC-3*. British Library Research and Development Department.
- Alireza Salemi, Julian Killingback, and Hamed Zamani. 2025. *Expert: Effective and explainable evaluation of personalized long-form text generation*. *Preprint*, arXiv:2501.14956.
- Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. 2023. *Lamp: When large language models meet personalization*.
- Alireza Salemi, Sheshera Mysore, Michael Bender-sky, and Hamed Zamani. 2024. *Lamp: When large language models meet personalization*. *Preprint*, arXiv:2304.11406.
- Scott Sanner, Krisztian Balog, Filip Radlinski, Ben Wedin, and Lucas Dixon. 2023. Large language models are competitive near cold-start recommenders for language-and item-based preferences. In *Proceedings of the 17th ACM conference on recommender systems*, pages 890–896.
- Shibani Santurkar, Esin Durmus, Faisal Ladhak, Cinoo Lee, Percy Liang, and Tatsunori Hashimoto. 2023. Whose opinions do language models reflect? In *International Conference on Machine Learning*, pages 29971–30004. PMLR.
- Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. *Reflexion: Language agents with verbal reinforcement learning*. *Preprint*, arXiv:2303.11366.
- Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. *arXiv preprint arXiv:2104.07567*.
- Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.
- Hieu Tran, Zonghai Yao, Won Seok Jang, Sharmin Sultana, Allen Chang, Yuan Zhang, and Hong Yu. 2025a. Medreadctrl: Personalizing medical text generation with readability-controlled instruction learning. *arXiv preprint arXiv:2507.07419*.
- Hieu Tran, Zonghai Yao, Lingxi Li, and Hong Yu. 2025b. Readctrl: Personalizing text generation with readability-controlled instruction learning. In *Proceedings of the Fourth Workshop on Intelligent and Interactive Writing Assistants (In2Writing 2025)*, pages 19–36.
- Kaiyang Wan, Honglin Mu, Rui Hao, Haoran Luo, Tianle Gu, and Xiuying Chen. 2025. *A cognitive writing perspective for constrained long-form text generation*. *Preprint*, arXiv:2502.12568.
- Mengting Wan and Julian J. McAuley. 2018. *Item recommendation on monotonic behavior chains*. In *Proceedings of the 12th ACM Conference on Recommender Systems, RecSys 2018, Vancouver, BC, Canada, October 2-7, 2018*, pages 86–94. ACM.
- Mengting Wan, Rishabh Misra, Ndapa Nakashole, and Julian J. McAuley. 2019. *Fine-grained spoiler detection from large-scale review corpora*. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2605–2610. Association for Computational Linguistics.
- Lei Wang and Ee-Peng Lim. 2023. Zero-shot next-item recommendation using large pretrained language models. *arXiv preprint arXiv:2304.03153*.
- Yu Wang, Nedim Lipka, Ryan A Rossi, Alexa Siu, Ruiyi Zhang, and Tyler Derr. 2024. Knowledge graph prompting for multi-document question answering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 19206–19214.
- Likang Wu, Zhaopeng Qiu, Zhi Zheng, Hengshu Zhu, and Enhong Chen. 2024. Exploring large language model for graph data understanding in online job recommendations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 9178–9186.
- Zhishang Xiang, Chuanjie Wu, Qinggang Zhang, Shengyuan Chen, Zijin Hong, Xiao Huang, and Jinsong Su. 2025. When to use graphs in rag: A comprehensive analysis for graph retrieval-augmented generation. *arXiv preprint arXiv:2506.05690*.
- Yuwei Yan, Yu Shang, Qingbin Zeng, Yu Li, Keyu Zhao, Zhiheng Zheng, Xuefei Ning, Tianji Wu, Shengen Yan, Yu Wang, and 1 others. 2025. Agentsociety challenge: Designing llm agents for user modeling and recommendation on web platforms. *arXiv preprint arXiv:2502.18754*.
- Zonghai Yao, Ahmed Jaafar, Beining Wang, Zhichao Yang, and Hong Yu. 2024. Do clinicians know how to prompt? the need for automatic prompt optimization

help in clinical note generation. In *Proceedings of the 23rd Workshop on Biomedical Natural Language Processing*, pages 182–201.

Zonghai Yao, Aditya Parashar, Huixue Zhou, Won Seok Jang, Feiyun Ouyang, Zhichao Yang, and Hong Yu. 2025. Mcqg-srefine: Multiple choice question generation and evaluation with iterative self-critique, correction, and comparison feedback. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 10728–10777.

Kechi Zhang, Zhuo Li, Jia Li, Ge Li, and Zhi Jin. 2023a. Self-edit: Fault-aware code editor for code generation. *Preprint*, arXiv:2305.04087.

Kexun Zhang, Danqing Wang, Jingtao Xia, William Yang Wang, and Lei Li. 2023b. Algo: Synthesizing algorithmic programs with llm-generated oracle verifiers. *Preprint*, arXiv:2305.14591.

Zhehao Zhang, Ryan A. Rossi, Branislav Kveton, Yijia Shao, Diyi Yang, Hamed Zamani, Franck Dernoncourt, Joe Barrow, Tong Yu, Sungchul Kim, Ruiyi Zhang, Jiuxiang Gu, Tyler Derr, Hongjie Chen, Junda Wu, Xiang Chen, Zichao Wang, Subrata Mitra, Nedim Lipka, and 2 others. 2024a. Personalization of large language models: A survey. *Preprint*, arXiv:2411.00027.

Zhehao Zhang, Ryan A Rossi, Branislav Kveton, Yijia Shao, Diyi Yang, Hamed Zamani, Franck Dernoncourt, Joe Barrow, Tong Yu, Sungchul Kim, and 1 others. 2024b. Personalization of large language models: A survey. *arXiv preprint arXiv:2411.00027*.

Zihuai Zhao, Wenqi Fan, Jiatong Li, Yunqing Liu, Xiaowei Mei, Yiqi Wang, Zhen Wen, Fei Wang, Xiangyu Zhao, Jiliang Tang, and 1 others. 2024. Recommender systems in the era of large language models (llms). *IEEE Transactions on Knowledge and Data Engineering*, 36(11):6889–6907.

Zhi Zheng, Zhaopeng Qiu, Xiao Hu, Likang Wu, Hengshu Zhu, and Hui Xiong. 2023. Generative job recommendations with large language model. *arXiv preprint arXiv:2307.02157*.



## A Appendix

### A.1 Prompts

#### Query template $x$ (Yelp)

Generate a review text written by a user who has a given an overall rating of [TARGET\_RATING] for a business with following details:  
[BUSINESS\_DETAILS]

Figure 5: Query Template

#### Generation Prompt $\phi_g$ (Yelp)

Given a target user's past reviews, a business description, and reviews from other users for the same business, your task is to generate a review that the target user might write. Below is the list of reviews from the target user on different businesses:

#User Profile:  
SAMPLE 1:  
[SAMPLE 1 Text]  
...

SAMPLE 2:  
[SAMPLE 2 Text]  
...

Consider a business with the following details:

[BUSINESS\_DETAILS]

Others have reviewed this business before. Below is a list of their reviews:

#Neighbor Profile:  
SAMPLE 1:  
[SAMPLE 1 Text]  
...

SAMPLE 2:  
[SAMPLE 2 Text]  
...

Now, generate a review from the target user who has a given an overall rating of [TARGET\_RATING] for the business mentioned above, without any additional explanation, adhering to the following instructions.

- Analyze the target user's previous reviews to capture their writing style, tone, sentiment and content preferences.
  - Use the business details to understand its features, benefits, and intended use.
  - Consider the opinions expressed in other users' reviews to identify common themes regarding various aspects of the business.
  - Generate a review that reflects how the target user would likely perceive the business, consistent with their past preferences and reviewing habits.
- Use the format: 'Review text:'.

Figure 6: Generation Prompt Template

## Critic Prompt $\phi_c$ (Yelp)

You are a Personalization Critic Module. Specifically, you will analyze user review texts written for businesses. Your task is to evaluate how well the \*Generated Output\* is personalized for the Target User and to give feedback along the specified criteria.

### # Your Inputs

Query - The user's request containing the business details.

Rating - the overall rating the target user has given for the business.

Generated Output - review text to be evaluated.

User Profile - past reviews written by the target user for other businesses.

Neighbor Profile - reviews from other users for this business.

### # Feedback Criteria

- Tone Consistency: Does the tone and sentiment in the \*\*Generated Output\*\* align with the user's writing style and the target review rating?
- Vocabulary Match: Does the vocabulary level in the \*\*Generated Output\*\* align with that of the user?
- Sentence Structure: Are the sentence lengths, complexity, and syntactic structures in the \*\*Generated Output\*\* similar to those used by the user?
- Topic Relevance: Is the output content relevant to the query, free of off-topic information, and inclusive of important aspects?

### # Analysis Strategy

- Analyze the user's writing style from the \*\*User Profile\*\*. Your feedback for the first three criteria (Tone Consistency, Vocabulary Match, Sentence Structure) should solely be on style and should not include any content-related suggestions. To evaluate tone, use profile reviews with sentiments that align closely with the target rating.
- Consider opinions expressed in other users' reviews to identify themes and aspects of the business from the \*\*Neighbor Profile\*\* relevant to the query. Use these insights to identify any off-topic or missing aspects in the \*\*Generated Output\*\*. Your feedback for the last criterion (Topic Relevance) should solely address the content and be grounded in \*\*Neighbor Profile\*\*.

You can output 'No further improvement needed' along a criteria if the performance along the criteria is exceptional.

### # Output Format

Return your output in the following JSON format:

```
{
  "Tone Consistency": **Feedback text**,
  "Vocabulary Match": **Feedback text**,
  "Sentence Structure": **Feedback text**,
  "Topic Relevance": **Feedback text**
}
```

### ## Test Inputs

#### #Query:

Generate a review text written by a user who has a given an overall rating of [TARGET\_RATING] for a business with following details:  
[BUSINESS\_DETAILS]

#### #Generated Output:

[GENERATED\_OUTPUT]

#### #User Profile:

SAMPLE 1:

[SAMPLE 1 Text]

...

SAMPLE 2:

[SAMPLE 2 Text]

...

#### #Rating:

[TARGET\_RATING]

#### #Neighbor Profile:

```
SAMPLE 1:
[SAMPLE 1 Text]
...

SAMPLE 2:
[SAMPLE 2 Text]
...

Your output should be a valid json object in ```json ``` block without any additional
explanation.
```

Figure 7: Critic Prompt Template

#### Refinement Prompt $\phi'_g$ (Yelp)

```
[GENERATION PROMPT]

Your previously generated review is:
[GENERATED_OUTPUT]

Your review has room for improvement. The feedback on your review from an expert is:
[FEEDBACK]

Based on the improvements suggested in the feedback, please improve your review, without any
additional explanation.
Use the format: 'Review text:'.
```

Figure 8: Refinement Prompt Template. The [GENERATION PROMPT] in the above template is shared in Figure 6

### Knockout Prompt $\phi_k$ (Yelp)

You are an impartial evaluator of style and content alignment. You will be evaluating a review text about a particular business written by an author. Below are samples of the author's writing containing past review texts of various businesses, review samples from other users for the business being reviewed, the input query, and two reviews.

# Author's Writing:

SAMPLE 1:

[SAMPLE 1 Text]

...

SAMPLE 2:

[SAMPLE 2 Text]

...

# Reviews from other users:

SAMPLE 1:

[SAMPLE 1 Text]

...

SAMPLE 2:

[SAMPLE 2 Text]

...

# Query:

[QUERY]

# Review A:

[GENERATED\_OUTPUT\_A]

# Review B:

[GENERATED\_OUTPUT\_B]

# Task

Analyze the samples given under \*Author's Writing\* to identify the author's writing style. Consider the opinions expressed in \*Reviews from other users\* above to understand common themes and aspects of the business. Determine which of the two reviews is more likely to have been written by the author. Consider each review's similarity with regards to (1) tone (2) vocabulary level (3) sentence structure (4) tonal alignment with the target rating (5) avoidance of off-topic information (6) inclusion of information relevant to the query based on other user reviews.

# Output Format:

You must return the winning review option along with a brief explanation for your choice. Your output should be a valid json object in ```json ``` block in following JSON format:

```
{{"answer": <either A or B>,"explanation": "....",}}
```

Figure 9: Knockout Prompt Template



### Knockout + Best-of-N Prompt $\phi_n$ (Yelp)

You are an impartial evaluator of style and content alignment. You will be evaluating a review text about a particular business written by an author. Below are samples of the author's writing containing past review texts of various businesses, review samples from other users for the business being reviewed, the input query, and three reviews.

# Author's Writing:

SAMPLE 1:

[SAMPLE 1 Text]

...

SAMPLE 2:

[SAMPLE 2 Text]

...

# Reviews from other users:

SAMPLE 1:

[SAMPLE 1 Text]

...

SAMPLE 2:

[SAMPLE 2 Text]

...

# Query:

[QUERY]

# Review A:

[GENERATED\_OUTPUT\_A]

# Review B:

[GENERATED\_OUTPUT\_B]

# Review C:

[GENERATED\_OUTPUT\_C]

# Task

Analyze the samples given under \*Author's Writing\* to identify the author's writing style. Consider the opinions expressed in \*Reviews from other users\* above to understand common themes and aspects of the business. Determine which of the three reviews is more likely to have been written by the author. Consider each review's similarity with regards to (1) tone (2) vocabulary level (3) sentence structure (4) tonal alignment with the target rating (5) avoidance of off-topic information (6) inclusion of relevant information based on other user reviews.

# Output Format:

You must return the winning review option along with a brief explanation for your choice. Your output should be a valid json object in ```json ``` block in the following JSON format:

```
{{
  "answer": <either A or B or C>,
  "explanation": "....",
}}
```

Figure 10: Knockout + Best-of-N Prompt Template

The prompt templates for the other two datasets follow the same structure, with the word ‘business’ replaced by ‘book’ for Goodreads and by ‘product’ for Amazon throughout.

## A.2 Topic Extraction

Dataset	Method	METEOR	GEval	# token (critic)	# token (gen)	# token (topics)
Yelp	PerFine	0.185	0.402	10.77	9.54	-
	PerFine + Topic_Extraction	<b>0.188</b>	<b>0.403</b>	7.73	9.48	2.60
Goodreads	PerFine	<b>0.209</b>	<b>0.453</b>	11.65	10.57	-
	PerFine + Topic_Extraction	0.209	0.450	7.73	10.48	2.81
Amazon	PerFine	0.197	<b>0.432</b>	9.58	8.27	-
	PerFine + Topic_Extraction	<b>0.199</b>	0.427	7.73	8.19	2.09

Table 4: Performance and Token usage for Topic Extraction method

The topic extraction method outperforms the baselines and achieves performance comparable to the vanilla setting, while requiring fewer critic tokens during feedback rounds. However, it introduces an initial overhead for extracting topics, which increases the overall token consumption. Qwen-2.5-32B-Instruct is used for extracting the topics.

## Topic Extraction Prompts

### Writing Style Extraction $\phi_{ts}$

You are a helpful assistant capable of analyzing a user's writing style. Your task is to evaluate a list of texts written by the user and determine the writing style. Consider elements such as tone, vocabulary, sentence structure, and overall flow.

#### # Your task

Given a list of writing samples by a user, analyze the writing style in terms of three aspects: tone, vocabulary, and sentence structure. If the samples have mixed sentiments (e.g., positive, negative, neutral), analyze and describe the writing style separately for each sentiment type. However, present your findings for each style aspect (tone, vocabulary, sentence structure) in a single, consolidated paragraph that includes distinctions based on sentiment. Avoid including specific content from the samples in your descriptions.

#### # Your inputs

User's Writing : A list of sample texts written by the user.

# Your outputs - should be a valid JSON object in ```json``` block that contains the following keys. Limit the description to two or three sentences for each aspect:.

Tone : "..."

Vocabulary style: "..."

Sentence structure: "..."

### Content Topics Extraction $\phi_{tc}$

You are a helpful assistant capable of extracting different atomic aspects discussed in a list of input texts. Your task is to analyze the texts and generate a JSON list of aspects and descriptions for each of them. Each aspect should be supported by evidence from the input texts, in the form of one or more related sentences.

#### # Your task

Given a list of review texts, identify common themes related to various characteristics of the reviewed item or experience, and extract atomic aspects. Additionally, generate a description for each identified aspect.

#### # Your inputs

Review texts : A list of review texts provided for a particular product, service, or entity.

# Your outputs - should be a valid JSON list in ```json``` block that contains the following keys.

aspect: the extracted aspect title

description: the description of the aspect.

Figure 11: Topic Extraction Prompts.

### Critic Prompt using extracted topics $\phi_t$ (Yelp)

You are a Personalization Critic Module. Specifically, you will analyze user review texts written for businesses. Your task is to evaluate how well the \*Generated Output\* is personalized for the target user and give feedback along the specified criteria.

#### # Your Inputs

Query - The user's request containing the business details.

Rating - the overall rating the target user has given for the product.

Generated Output - review text to be evaluated.

Writing style - The writing style of the target user.

Business characteristics- A list of aspects and their descriptions about the business, based on reviews from other users.

#### # Feedback Criteria

- Tone Consistency: Does the tone and sentiment in the Generated Output align with the user's writing style and the given review rating?

- Vocabulary Match: Does the vocabulary level in the Generated Output align with that of the user?

- Sentence Structure: Are the sentence lengths and structures similar to the user's writing?

- Topic Relevance: Is the output content relevant to the query, free of off-topic information, and inclusive of important aspects?

Your feedback for the first three criteria (Tone Consistency, Vocabulary Match, Sentence Structure) should be based solely on the **\*\*Writing style\*\*** and should not include any content-related suggestions.

Your feedback for the last criterion (Topic Relevance) should be based solely on the **\*\*Business characteristics\*\*** input and should address the content.

#### # Output Format

Return your output in the following JSON format:

```
{
  "Tone Consistency": ...,
  "Vocabulary Match": ...,
  "Sentence Structure": ...,
  "Topic Relevance": ...,
}
```

#### ## Test Inputs

#Query:

[QUERY]

#Generated Output:

[GENERATED\_OUTPUT]

#Writing style:

Tone: ...

Vocabulary style: ...

Sentence structure: ...

#Rating:

[TARGET\_RATING]

#Business characteristics:

aspect title: ...

aspect detail: ...

aspect title: ...

aspect detail: ...

Your output should be a valid json object in ```json ``` block without any explanation.

Figure 12: Critic prompt using extracted topics.



### A.3 Evaluation Prompt

The prompt is borrowed from (Salemi et al., 2025)

#### GEval (Pointwise scoring)

You are a helpful assistant. Please act as an impartial judge and evaluate the quality of the response to instruction of the user displayed below. Based on the scoring criteria, please provide a score to the response compared to the reference. Be as objective as possible. You should consider both content and writing style similarity to assign a score.

# Your inputs:

- instruction: the instruction provided to the AI assistant.
- reference: the correct answer to the instruction.
- response: the response generated by the AI assistant.

# Scoring Criteria:

You should assign a score to the response based on the following criteria:

- Score 0: The answer is completely unrelated to the reference.
- Score 1: The answer has minor relevance but does not align with the reference.
- Score 2: The answer has moderate relevance but contains inaccuracies.
- Score 3: The answer aligns with the reference but has minor omissions.
- Score 4: The answer is completely accurate and aligns perfectly with the reference.

# Output:

Your output should be a valid JSON object in `''json''` block that contains the following keys:  
-score: the score that you assigned to the AI assistant's answer. The score should be an integer between 0 and 4.

Figure 13 : GEval Prompt

## A.4 Case study

### QUERY

Generate a review text written by a user who has a given an overall rating of 3.0 for a business with following details:

city: Tampa  
state: FL  
attributes:  
-RestaurantsGoodForGroups: True  
-GoodForKids: True  
-RestaurantsDelivery: False  
-RestaurantsAttire: 'casual'  
-RestaurantsReservations: True  
-BYOBCorkage: 'no'  
-BusinessAcceptsCreditCards: True  
-RestaurantsTakeOut: False  
-RestaurantsPriceRange2: 2  
-Alcohol: 'full\_bar'  
-BusinessParking:  
  - garage : True  
  - street : True  
  - validated : False  
  - lot : False  
  - valet : True  
-OutdoorSeating: False  
categories: Restaurants, Asian Fusion

### GROUNDTRUTH REFERENCE

I really liked the interior. Besides being a food lover, I'm also a big fan of the ambiance. I had the lettuce wrap with chicken and shrimp, which was good. I also drank their mango mojito...tasty. My main course was the BC Spicy chicken and shrimp (their version of Kung Pao) which was also yummy. Our waitress/bartender was very nice and helpful. The food didn't blow my taste buds though, I guess I was just expecting more since it was such a nice restaurant. It is definitely worth a try and I will return for their happy hour.

### USER PROFILE

#### SAMPLE 1:

They use NY water! I really liked how they tried to capture NYC in their decorations as well. Its probably the best pizza in Tampa, but still not as good as you can get in NYC.

I would really give them 3.5 stars.

#### SAMPLE 2:

Yummy! One of the best seafood restaurants in Tampa. The fish is so fresh and it tasted delicious = double points ( I was full, but could not stop eating.), the calamari was also great (You should try it because I have never had it served the way they made it.), presentation and service = GREAT. Everything was perfect from the moment we walked in. The ambiance, service, food...you get what you pay for and I seem to have had bad luck with that lately. I can't wait to come back and be a fatty again.

#### SAMPLE 3:

This place is awesome. Bowling, bars, music, and most importantly FOOD! =) I was also very impressed with their food selection; the variety they have can satisfy any person in your groups cravings/tastebuds. The tuna in their crunchy ahi poke dip was delicious and fresh. Oh, and they also have good drinks that can be served in huge glass bowls. A little pricey, but good for sharing. You have to go there.

#### SAMPLE 4:

I've been meaning to post this update sooner, but I was extremely busy. I really appreciated that they take customer satisfaction seriously. I was contacted by one of the owners to come back in for a complimentary meal, as a result of my previous review. The manager there told me that they made changes and was very adamant in having us try the new and/or revised dishes.

From what we remembered last year (yes, it's been a year), the food has improved. We had their chips, salsa, guacamole, tacos, a fiesta bowl, and elote.

I still love their chips and salsa. Even though the chips have changed from last year, it was an improvement. They also now make homemade guacamole table-side, which was also very good. My friend usually hates guac, but decided to give it a try and loved it. The elote was good, although I wish it came out with char marks because in my opinion it tastes better that way. Service was also good. Our waitress checked up on us frequently.

Although it pains me to say what I'm going to say next, I can't help but be honest...I still was not impressed with their tacos or fiesta bowl. It was better than before, but not as good as I've had. For that reason, I would give them 3.5 stars.

I will still come here for drinks, chips, salsa and guacamole! If I was rating them just for those items and service, I'd give them 4 stars.

#### NEIGHBOR PROFILE

##### SAMPLE 1:

Rating of 3.0 with the review: We came here on the recommendation of it being "good asian food". We went in despite the instinct to run from any restaurant that tries to be more of a "production" or "show" than a fine place to eat.

The waiter was, as expected, over-friendly to the point of extreme annoyance as he tried to tell us what was "great" on the menu and how every dish was his favorite. Yeah, sure.

The food was OK, and when biased for the rest of the food I had while Tampa, I have to admit it was above average. But not by much..

##### SAMPLE 2:

Rating of 1.0 with the review: I have been spoiled by authentic Chinese restaurants in Tampa, so me going to Bamboo Club is bound to be a disappointment from the start. Do they have anything that is not deep fried? It is at least in the category of Asian Fusion to let you know that it is anything but authentic and very similar to a take-out fried meat chunks covered in sweet sauce place, except with Cheesecake Factory prices. The establishment is beautiful and it has a full bar with delicious mixed drinks. But, don't come here expecting something unique or you will be severely disappointed. If this place does so well, then maybe TGI Fridays should add stir-fry to their menu and have the same taste and success..

##### SAMPLE 3:

Rating of 4.0 with the review: I came here for lunch with my step-sister before a day of shopping at the International Plaza. To keep it light we split an order of their Mongolian Beef lunch special and I have to say that it tasted extremely good. The atmosphere of the restaurant is very upscale and relaxing and the service was perfect as well. Although you are having lunch, they still give you a large enough amount of food to feed two people and it's really a great deal for people who want the great taste, but are not in a good position to pay their dinner prices. I'll be back for sure..

##### SAMPLE 4:

Rating of 3.0 with the review: I took my wife, mother, and children there for Mother's Day. Online they offered a kid's menu, but at the restaurant, not so much. They screwed up our reservation, which they confirmed online. The server came by only twice to check our drinks, and they forgot to place 2 of our 6 orders, so while 4 out of the six ate, 2 sat there. One of the two was a child. The food, when it arrived, was excellent. The ambiance was very cool. The service was sub-par. Being a former server, I am a very good tipper, usually 25% - 30 % of the bill. This time, let's just say...not so much. Would I go back? If someone else were picking up the tab.

They have a full bar, which is great. I also like how they have a variety of food options. It's probably one of the better Asian restaurants in Tampa, but still not as good as you can get in NYC. I'd give it 3.0 stars.

PerFine (Knockout)

Feedback 0

**tone consistency:** The tone in the generated review is somewhat consistent with the target rating of 3.0, expressing a moderate level of satisfaction. However, the user's profile shows a **more playful and enthusiastic tone** in positive reviews, and a more critical tone in negative reviews, which is not fully reflected in the generated output.

**Vocabulary Match:** The vocabulary level in the generated review is generally appropriate, but it **lacks the specific and sometimes colloquial language** used in the user's profile, such as 'double points' or 'You have to go there.'

**Sentence Structure:** The sentence structure in the generated review is simple and straightforward, which is somewhat consistent with the user's profile, but the **user tends to use more complex structures and varied lengths** in their reviews.

**Topic Relevance:** The generated review covers relevant topics such as the full bar and food variety, but it misses important aspects mentioned in the neighbor profile, such as the **service quality** and the kid-friendly environment. Additionally, the review could have included more specific details about the **food quality and atmosphere**.

Generation 1

They have a full bar, which is awesome! I loved how they have a variety of food options, it's like they really try to capture the Asian vibe in their menu. You have to go there for their Asian fusion food, it's probably one of the better options in Tampa, but still not as good as you can get in NYC. The service was good, our server checked up on us frequently and was super friendly. I also appreciated that it's kid-friendly, my little ones had a blast while we were there. Although the food was good, it wasn't mind-blowing, which is why I'd give it 3.0 stars.

Feedback 1

**tone consistency:** The tone in the Generated Output is somewhat consistent with the user's writing style, but it **lacks the casual** and slightly sarcastic tone seen in SAMPLE 1 and SAMPLE 4 of the User Profile.

**Vocabulary Match:** The vocabulary used in the Generated Output is generally appropriate, but **it is more formal and less colloquial** than what the user typically uses, as seen in SAMPLE 1 and SAMPLE 4.

**Sentence Structure:** The sentence structure in the Generated Output is relatively simple and straightforward, which is somewhat similar to the user's style, but the **user tends to use more varied and complex sentence structures**, as seen in SAMPLE 4.

**Topic Relevance:** The Generated Output covers relevant aspects such as the food, service, and kid-friendliness, aligning well with the query. However, it could **include more specific details about the atmosphere** or parking, which are mentioned in the Neighbor Profile.

Generation 2

They've got a sweet setup going on with that full bar, I mean, who doesn't love a good drink selection? I dug the variety of food options, it's like they're trying to give you a taste of Asia in Tampa, but let's be real, it's still not as good as what you'd get in the city. Our server was on it, super friendly and checked in on us often, but the food, while good, wasn't blowing my mind, which is why I'm giving 'em 3.0 stars. I did appreciate that it's kid-friendly, my little ones had a blast, and the ambiance was pretty cool, I mean, it's got that upscale vibe going on. Parking's a bit of a pain, but, you know, it's a trade-off for good food, right?

Feedback 2

**tone consistency:** The tone in the generated output is relatively casual and slightly critical, which aligns with the user's writing style in the User Profile, especially in Sample 1 and Sample 4 where a casual and slightly critical tone is present.

**Vocabulary Match:** The vocabulary used in the generated output is simple and straightforward, which matches the user's vocabulary level in the User Profile.

**Sentence Structure:** The sentence structures in the generated output are relatively simple and varied, similar to the structures found in the User Profile, especially in Sample 1 and Sample 4.

**Topic Relevance:** The generated output covers relevant aspects such as food quality, service, and ambiance, but it **does not mention** the specific attributes like the **kid-friendly environment and the full bar**, which are important aspects mentioned in the query and the Neighbor Profile.



### Generation 3

They've got a nice setup going on with that full bar, offering a good selection of drinks. The variety of food options is great, it's like they're trying to give you a taste of Asia in Tampa. I liked that it's kid-friendly, my little ones had a blast, and the ambiance is pretty cool, with that upscale vibe going on. However, the food, while good, wasn't blowing my mind, which is why I'm giving 'em 3.0 stars. One thing that's worth noting is that parking can be a bit of a pain, but the good food and friendly service make up for it.

### Feedback 3

**TONE CONSISTENCY:** The tone in the generated review is somewhat consistent with the user's profile, but it is more casual and **less detailed** compared to some of the user's reviews. The user tends to be more specific and detailed in their descriptions, whereas the generated review is more general and concise.

**VOCABULARY MATCH:** The vocabulary used in the generated review matches the user's profile well, with simple and straightforward language. The user often uses casual and descriptive words which align with the generated review's vocabulary.

**SENTENCE STRUCTURE:** The sentence structures in the generated review are relatively simple and straightforward, which aligns with the user's profile. However, **the user's past reviews often contain more complex sentence structures and varied lengths.**

**TOPIC RELEVANCE:** The generated review covers important aspects of the restaurant, such as the food, drinks, ambiance, and service, which are relevant to the query. However, it lacks specific details about the Asian fusion cuisine and the kid-friendly aspects mentioned in the neighbor profile. Additionally, it does not mention the parking situation, which was a point of discussion in the neighbor profile.

### Generation 4

They've got a nice setup going on with that full bar, offering a good selection of drinks. The variety of food options is great, it's like they're trying to give you a taste of Asia in Tampa. I liked that it's kid-friendly, my little ones had a blast, and the ambiance is pretty cool, with that upscale vibe going on. However, the food, while good, wasn't blowing my mind, which is why I'm giving 'em 3.0 stars. One thing that's worth noting is that parking can be a bit of a pain, but the good food and friendly service make up for it.

### Feedback 4

**TONE CONSISTENCY:** The tone in the generated output is somewhat consistent with the user's writing style, but it **lacks the detailed and slightly critical tone** seen in the user's profile reviews. The user tends to provide more specific criticisms and less general statements.

**VOCABULARY MATCH:** The vocabulary level in the generated output matches the user's profile, using straightforward and conversational language.

**SENTENCE STRUCTURE:** The sentence structure in the generated output is simple and direct, similar to the user's profile, but the **user tends to use slightly more complex sentences with clauses.**

**TOPIC RELEVANCE:** The generated output covers relevant topics such as food, drinks, ambiance, and service, but it misses some important aspects highlighted in the neighbor profile, such as the **over-friendly waiter** and the reservation issues.

### Final Generation

I went to this Asian fusion place in Tampa and was impressed by their full bar and extensive drink selection. The variety of food options is definitely a plus, as they try to give you a taste of Asia right here in Tampa, but I have to say that the waiter was a bit too over-friendly for my taste, trying to tell us what was great on the menu and making it feel like a production rather than a genuine dining experience. The food was good, but not exceptional, and the ambiance is nice with that upscale vibe, but the parking situation was a bit of a hassle. They also seemed to have issues with reservations, which was a bit of a letdown. Overall, it's a decent spot, but nothing that would make me go out of my way for. I'd give them 3.0 stars.

### Final Feedback

**TONE CONSISTENCY:** The tone in the generated output **aligns well with the user's writing style**, expressing a balanced and slightly critical perspective. The sentiment matches the 3.0 rating.

**VOCABULARY MATCH:** The vocabulary level in the generated output **matches the user's profile, using casual and straightforward language.**

**SENTENCE STRUCTURE:** The **sentence lengths and complexity in the generated output are similar to those used by the user**, maintaining a consistent style.

**TOPIC RELEVANCE:** The **generated output covers relevant aspects such as the full bar, food options, waiter friendliness, ambiance, parking, and reservations**, aligning with the user's experience. However, it could include more specific details about the food quality and variety, as mentioned in the neighbor profile.

This case study illustrates the refinement process over five iterations. The PGraphRAG baseline does provide an appropriate response for a three-star rating. However, it could still be better personalized.

We can observe how the critic incrementally aligns the tone across iterations. In feedback 1, it suggests using a more enthusiastic and playful tone. Once this is addressed in the refined generations, it then notes in feedback 2 that the user prefers a more casual style, while in feedback 4 and feedback 5 it recommends incorporating more specific descriptions. After these aspects are integrated, the final feedback comments that the text is well aligned, balanced, and slightly critical (Highlighted in [blue](#)).

A similar pattern occurs with sentence structure, where the critic consistently suggests making it more varied and complex (inline with user's style) across all feedback iterations. This is finally corrected after five iterations, with the final feedback confirming that it is well aligned (Highlighted in [orange](#)).

When it comes to vocabulary, the critic clearly recommends adopting a more colloquial style (inline with user's style) in feedback 1 and feedback 2. Unlike tone and sentence structure, this aspect is resolved early, after two iterations, and in the remaining feedback the critic signals that it is well aligned and requires no further improvement (Highlighted in [violet](#)).

The critic is able to analyze the user's writing samples, infer the style, and compare it against the baseline generation.

On the content side, to ensure topic relevance, it provides useful suggestions regarding common aspects of restaurant reviews, such as service quality, ambiance, and friendly staff. These aspects are also present in the ground truth, leading to better topical relevance overall (Highlighted in [green](#)).

Another important observation is that generation 3 and generation 4 are identical. The critic judged the initial generation 4 as less personalized compared to generation 3, as a result of which the newly generated output was **knocked-out** and generation 3 was advanced as generation 4 (Highlighted in [navyblue](#)).