Personalized Graph-Based Retrieval for Large Language Models

Steven Au[1], Cameron J. Dimacali[1], Ojasmitha Pedirappagari[1], Namyong Park, Franck Dernoncourt[2], Yu Wang[3], Nikos Kanakaris[4]*, Hanieh Deilamsalehy[2] Ryan A. Rossi[2], Nesreen K. Ahmed[5]

[1]University of California Santa Cruz, [2]Adobe Research,
[3]University of Oregon, [4]Amazon Web Services, [5]Cisco AI Research

https://pgraphrag-benchmark.github.io

Abstract

As large language models (LLMs) continue to evolve, their ability to deliver personalized, context-aware responses holds significant promise for enhancing user experiences. However, most existing personalization approaches rely solely on user history, limiting their effectiveness in cold-start and sparse-data scenarios. We introduce Personalized Graph-based Retrieval-Augmented Generation (PGraphRAG), a framework that enhances personalization by leveraging user-centric knowledge graphs. By integrating structured user information into the retrieval process and augmenting prompts with graph-based context, PGraphRAG improves both relevance and generation quality. We also present the Personalized Graph-based Benchmark for Text Generation, designed to evaluate personalized generation in real-world settings where user history is minimal. Experimental results show that PGraphRAG consistently outperforms state-of-the-art methods across diverse tasks, achieving average ROUGE-1 gains of 14.8% on long-text and 4.6% on short-text generation—highlighting the unique advantages of graph-based retrieval for personalization.

1 Introduction

The rapid advancement of large language models (LLMs) has enabled a wide range of NLP applications, including conversational agents, content generation, and code synthesis. Models like GPT-4 (OpenAI, 2024) now power virtual assistants capable of answering complex queries and engaging in multi-turn dialogue (Brown et al., 2020). As these models continue to evolve, their ability to generate personalized, context-aware responses offers new opportunities to enhance user experiences (Salemi et al., 2024b; Huang et al., 2022). Personalization enables LLMs to adapt outputs to individual preferences and goals, resulting in richer, more relevant interactions (Zhang et al., 2024). While personalization has been studied in areas such as information retrieval and recommender systems (Xue et al., 2009; Naumov et al., 2019), its integration into LLMs for generation tasks remains relatively underexplored.

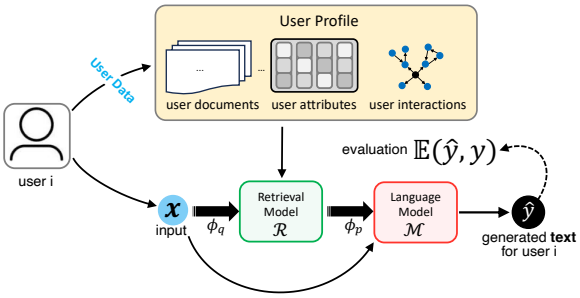

Figure 1: Overview of the proposed PGraphRAG framework. We construct user-centric graphs from user profile and interaction data, then retrieve structured, user-relevant information from the graph. This context is used to condition the language model's generation, producing personalized outputs for user i.

One of the key challenges in advancing personalized LLMs is the lack of benchmarks that adequately capture the complexities of personalization tasks. Popular natural language processing (NLP) benchmarks (e.g., (Wang et al., 2019b), (Wang et al., 2019a), (Gehrmann et al., 2021)) primarily focus on general language understanding and generation, with limited emphasis on personalization. As a result, researchers and practitioners lack standardized datasets and evaluation metrics for developing and assessing models designed for personalized text generation. Recently, efforts such as LaMP (Salemi et al., 2024b) and LongLaMP (Kumar et al., 2024) have begun addressing this gap. LaMP evaluates personalization for tasks like email subject and news headline generation, while LongLaMP extends this to long-text tasks such as email and abstract generation. However, both benchmarks rely exclusively on user his-

*The work does not relate to the author's position at Amazon.

tory to model personalization. Here, user history typically refers to a set of previously written texts by the same user—such as past reviews, messages, or profile-specific documents—which are used as context to condition the generation.

Challenges with Cold-Start Users. While leveraging user history is valuable for capturing individual style and preferences, it presents a cold-start challenge: many users have little or no prior data. In fact, as shown in Figure 2, over 99.99% of users in the Amazon Reviews dataset have fewer than three interactions. Benchmarks like LaMP and LongLaMP filter out these users by imposing a minimum user profile size threshold to ensure sufficient data for personalization. As a result, they exclude the vast majority of users, making their evaluations less representative of real-world deployment. This design choice leads to model failures when prompts lack sufficient context, often resulting in generic outputs.

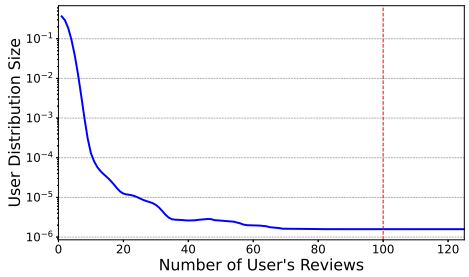

Figure 2: Distribution of user profile sizes in the Amazon user-product dataset. The vast majority of users have only a few reviews, highlighting the prevalence of sparse profiles. The red vertical line indicates the minimum profile size threshold used in prior benchmarks such as LaMP and LongLaMP.

Proposed Approach. To address these challenges, we propose *Personalized Graph-based Retrieval-Augmented Generation* (PGraphRAG), a novel framework that enhances personalized text generation by leveraging user-centric knowledge graphs. These structured graphs represent user information — such as interests, preferences, and prior interactions — in an interconnected graph structure. During inference, PGraphRAG retrieves semantically relevant context from both the user's own profile and neighboring profiles extracted from the graph, and augments the prompt with this information to guide generation. This graph-based approach enables the model to produce contextually appropriate and personalized outputs, even when user history is sparse or unavailable (see Figure 1).

Formally, the target task of PGraphRAG is personalized text generation conditioned on user-specific context retrieved from a structured knowledge graph. Given a user query (e.g., a product title or review prompt), the system retrieves relevant entries from the graph-based profile and generates an output tailored to the user's preferences. This setup generalizes personalization beyond pure user text history, enabling context-rich generation even in sparse or cold-start settings.

Proposed Benchmark. To evaluate our approach, we introduce the *Personalized Graph-based Benchmark for Text Generation*, a novel evaluation benchmark designed to fine-tune and assess LLMs on twelve personalized text generation tasks, including long- and short-form generation as well as classification. This benchmark addresses the limitations of existing personalized LLM benchmarks by providing datasets that specifically target personalization capabilities in real-world settings where user history is sparse. In addition, it enables a more comprehensive assessment of a model's ability to personalize outputs based on structured user information.

Our benchmark supports evaluation in sparse-profile settings, and PGraphRAG is designed to retrieve semantically relevant context not only from the user's own profile but also from neighboring profiles extracted from the graph — enabling effective personalization even when the user has only a single input (e.g., one review in their profile). Empirically, PGraphRAG significantly outperforms LaMP in these low-profile scenarios, demonstrating the advantages of graph-based reasoning over strict reliance on user history.

Our contributions are summarized as follows:

1. **Benchmark.** We introduce the *Personalized Graph-based Benchmark for Text Generation*, consisting of 12 tasks spanning long-form generation, summarization, and classification. To support further research, we release the benchmark publicly. [1]
2. **Method.** We propose *PGraphRAG*, a retrieval-augmented generation framework that addresses the cold-start problem by augmenting generation with structured, user-specific information from a knowledge graph.
3. **Effectiveness.** We show that PGraphRAG achieves state-of-the-art performance across all tasks in our benchmark, demonstrating the value of graph-based reasoning for personal-

[1] https://pgraphrag-benchmark.github.io

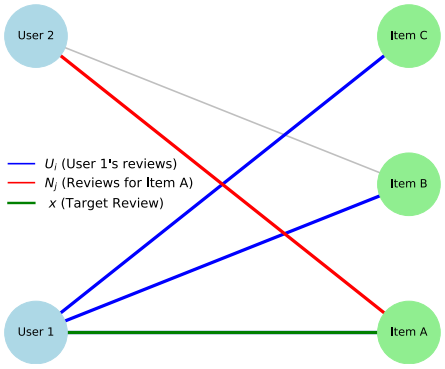

Figure 3: Example of a bipartite user-centric graph $G = (U, V, E)$ showing users, items, and interaction edges (e.g., reviews).

ized text generation.

2 Personalized Graph-based Benchmark for LLMs

We introduce the *Personalized Graph-Based Benchmark* to evaluate LLMs on their ability to generate personalized outputs across twelve tasks, spanning long-form generation, short-form generation, and ordinal classification. The benchmark is constructed from real-world datasets across multiple domains.

2.1 Personalized Text Generation: Problem Definition

Each benchmark instance includes: (1) an input sequence x to the LLM, (2) a target output y the model is expected to generate, and (3) a user profile P_i derived from a structured user-centric graph. Given an input-output pair (x, y) associated with user i, the goal is to generate a personalized output $\hat{y}$ that aligns with the semantics and style of y, conditioned on the user profile P_i.

We assume user context is represented using a bipartite user-centric graph that captures user-item interactions (see Figure 3 for an illustration). The profile P_i is constructed from this graph and includes both interactions authored by the user and related signals from similar items or neighboring users. The full construction of P_i is detailed in Section 3.

Formally, the personalized generation task is defined as:

$$\hat{y} = \arg\max_{y'} \Pr(y' \mid x, P_i) \tag{1}$$

where x is the input query, y is the target output, and P_i denotes the profile of user i derived from a user-item interaction graph. The model generates an output $\hat{y}$ that maximizes the likelihood of personalized text conditioned on the input and user profile. This formulation enables generalization beyond user history by leveraging structured, graph-derived context.

In practice, our framework retrieves a personalized context $\mathcal{R}(P_i) \subseteq P_i$ from the graph to condition generation, yielding the operational objective:

$$\hat{y} = \arg\max_{y'} \Pr(y' \mid x, \mathcal{R}(P_i)) \tag{2}$$

where $\mathcal{R}(P_i)$ represents the retrieved subset of user- and item-level interactions used as context during generation.

Finally, statistics for all benchmark tasks and their associated graphs are summarized in Table 1 and Table 2. Additional dataset split details are provided in the appendix.

2.2 Task Definitions

Task 1: User Product Review Generation. Personalized review text generation has progressed from incorporating user-specific context to utilizing LLMs for producing fluent and contextually relevant reviews and titles (Ni and McAuley, 2018). This task aims to generate a product review i_{text} for a target user, conditioned on their own review title i_{title} and a set of additional reviews P_i from their user profile. We construct this dataset from the Amazon Reviews 2023 corpus (Hou et al., 2024), spanning multiple product categories and used to define a bipartite user-item graph.

Task 2: Hotel Experience Generation. Hotel reviews often contain rich narratives reflecting personal experiences, making personalization essential to capturing individual preferences and expectations (Kanouchi et al., 2020). This task focuses on generating a personalized hotel experience story i_{text}, using the target user's review summary i_{title} and contextual reviews P_i. We use the Hotel Reviews dataset, a subset of Datafiniti's Business Database (Datafiniti, 2017), to construct a user-hotel bipartite graph.

Task 3: Stylized Feedback Generation. Writing style — influenced by grammar, punctuation, and expression — is deeply personal and often shaped by geographic and cultural factors (Alhafni et al., 2024). This task involves generating personalized

Task	Type	Avg. Input Length	Avg. Output Length	Avg. Profile Size	# Classes
User-Product Review Generation	Long Text Generation	3.754 ± 2.71	47.90 ± 19.28	1.05 ± 0.31	-
Hotel Experiences Generation	Long Text Generation	4.29 ± 2.57	76.26 ± 22.39	1.14 ± 0.61	-
Stylized Feedback Generation	Long Text Generation	3.35 ± 2.02	51.80 ± 20.07	1.09 ± 0.47	-
Multilingual Product Review Generation	Long Text Generation	2.9 ± 2.40	34.52 ± 12.55	1.08 ± 0.33	-
User-Product Review Title Generation	Short Text Generation	30.34 ± 37.95	7.02 ± 1.14	1.05 ± 0.31	-
Hotel Experiences Summary Generation	Short Text Generation	90.40 ± 99.17	7.64 ± 0.92	1.14 ± 0.61	-
Stylized Feedback Title Generation	Short Text Generation	37.42 ± 38.17	7.16 ± 1.11	1.09 ± 0.47	-
Multilingual Product Review Title Generation	Short Text Generation	22.17 ± 20.15	7.15 ± 1.09	1.08 ± 0.33	-
User-Product Review Ratings	Ordinal Classification	34.10 ± 38.66	-	1.05 ± 0.31	5
Hotel Experiences Ratings	Ordinal Classification	94.69 ± 99.62	-	1.14 ± 0.61	5
Stylized Feedback Ratings	Ordinal Classification	40.77 ± 38.69	-	1.09 ± 0.47	5
Multilingual Product Ratings	Ordinal Classification	25.15 ± 20.75	-	1.08 ± 0.33	5

Table 1: Data statistics for the PGraphRAG Benchmark across the four datasets. For each task, we report the average input and output lengths (in words), measured on the test set using BM25-based retrieval with GPT. The average profile size indicates the number of reviews per user used for personalization.

Dataset	Users	Items	Edges/Reviews	Average Degree
User-Product Review Graph	184,771	51,376	198,668	1.68
Hotel Experiences Graph	15,587	2,975	19,698	2.12
Stylized Feedback Graph	58,087	600	71,041	2.42
Multilingual Product Review Graph	112,993	55,930	131,075	1.55

Table 2: Graph statistics for the datasets used in the personalized tasks. Each row reports the number of users, items, and edges (i.e., reviews), as well as the average degree of the resulting user-centric bipartite graph. The four graphs correspond to: User-Product, Multilingual Product, Stylized Feedback, and Hotel Experiences.

product feedback i_{text}, based on the user's feedback title i_{title} and additional feedback samples P_i from their profile. We utilize the Grammar and Online Product dataset, a subset of the Datafiniti Business corpus (Datafiniti, 2018), which reflects stylistic variation across multiple platforms and domains.

Task 4: Multi-lingual Review Generation. Personalization in multilingual review generation presents unique challenges due to differences in linguistic structures, cultural norms, and stylistic conventions (Cortes et al., 2024). This task focuses on generating product reviews i_{text} in Brazilian Portuguese, using the target user's review title i_{title} and additional reviews P_i from their profile. We construct this dataset using B2W-Reviews (Real et al., 2019), sourced from Brazil's largest e-commerce platform.

Task 5: User Product Review Title Generation. Short text generation for personalized review titles is particularly challenging, requiring the model to summarize sentiment and reflect user-specific phrasing preferences. This task generates a review title i_{title} for a given user, using their review text i_{text} and additional profile reviews P_i, without relying on parametric user embeddings (Xu et al., 2023). The dataset is derived from Amazon Reviews (Hou et al., 2024).

Task 6: Hotel Experience Summary Generation. Helping users write summaries of hotel experiences requires distilling detailed narratives into concise summaries that reflect individual preferences (Kamath et al., 2024). This task generates a hotel experience summary i_{title} based on the user's full experience text i_{text} and additional hotel reviews P_i. We use the Hotel Reviews dataset from the Datafiniti Business Database (Datafiniti, 2017).

Task 7: Stylized Feedback Title Generation. Stylized feedback summarization aims to capture individual voice and tone in generating short-form feedback. This task benchmarks stylized opinion generation across domains such as music, groceries, and household items (Iso et al., 2024). The model generates the target user's feedback title i_{title} based on their full feedback text i_{text} and additional feedback P_i from similar users. The dataset is built from the Datafiniti Products dataset (Datafiniti, 2018).

Task 8: Multi-lingual Review Title Generation. Multilingual short-text personalization adds further complexity, particularly in Brazilian Portuguese, where style and syntax vary significantly across users (Scalercio et al., 2024). This task gen-

erates a personalized review title i_{title} using the user's full review text i_{text} and contextual examples P_i from their graph neighborhood. Data: B2W-Reviews (Real et al., 2019).

Task 9: User Product Review Ratings. Predicting personalized product ratings involves understanding sentiment, user bias, and historical feedback. This task formulates rating prediction as an ordinal classification problem, where the model predicts $i_{\text{rating}} \in \{1, 2, 3, 4, 5\}$ based on the user's review text i_{text}, title i_{title}, and additional profile context P_i. The dataset is constructed from Amazon Reviews (Hou et al., 2024).

Task 10: Hotel Experience Ratings. Hotel ratings often reflect nuanced factors such as location, cleanliness, and service. This task models hotel experience rating i_{rating} prediction as a classification problem based on the user's review story i_{text}, summary i_{title}, and surrounding review context P_i. Data: Datafiniti Hotel Reviews (Datafiniti, 2017).

Task 11: Stylized Feedback Ratings. Cross-domain sentiment prediction explores how writing quality and sentiment expression vary across platforms (Yu et al., 2021). This task assigns a numerical feedback rating i_{rating} to a stylized user review using the input review text i_{text}, review title i_{title}, and personalized context P_i. The dataset is taken from the Datafiniti Product Database on Grammar and Online Product Reviews (Datafiniti, 2018).

Task 12: Multi-lingual Product Ratings. While sentence-level sentiment classification in Portuguese has seen success (de Araujo et al., 2024), this task extends to full review-level sentiment modeling in a multilingual setting. The model predicts a Portuguese user-product rating i_{rating} using both the review text i_{text}, the title i_{title}, and additional user-item interactions P_i. We construct this dataset using B2W-Reviews (Real et al., 2019).

3 The PGraphRAG Framework

Personalizing LLMs in real-world settings requires addressing two key challenges: (1) user profiles are often sparse or unavailable, and (2) incorporating additional user-related context must remain relevant, efficient, and scalable. To tackle these issues, PGraphRAG leverages structured user-centric knowledge graphs for context construction, and combines this with retrieval-augmented prompting. This design enables the model to generalize beyond parametric user embeddings or history-based filtering by dynamically retrieving relevant signals from graph-based user profiles that extend beyond the user's direct history.

Here, we present *PGraphRAG*, our proposed framework for personalizing large language models (LLMs) through graph-based retrieval augmentation. PGraphRAG enhances generation by conditioning a shared LLM on structured, user-specific context extracted from a user-centric knowledge graph. This enables tailored and context-aware outputs, especially in sparse or cold-start scenarios.

PGraphRAG leverages a bipartite user-centric graph $G = (U, V, E)$ to incorporate contextual signals beyond direct user history. We represent user context as a bipartite graph, where U is the set of user nodes, V the set of item nodes, and E the set of interaction edges (see Figure 3 for an illustration). An edge $(i, j) \in E$ corresponds to an interaction between user i and item j, such as a review that includes metadata like text, title, and rating. The user profile P_i consists of the set of reviews written by user i, along with reviews for the same items j written by other users $k \neq i$. For a given user $i \in U$, we define the profile P_i as the union of:

- the set of interactions authored by user i: $\{(i, j) \in E\}$,
- the set of interactions for the same items j written by other users $k \neq i$: $\{(k, j) \in E \mid (i, j) \in E\}$.

$$P_i = \{(i, j) \in E\} \cup \{(k, j) \in E \mid (i, j) \in E\} \quad \forall j \in V,\ k \in U,\ k \neq i \tag{3}$$

Due to context window limitations and efficiency considerations, we apply retrieval augmentation to select only the most relevant entries from P_i for conditioning the model. Given an input sample (x, y) for user i, the PGraphRAG workflow proceeds in three steps: a query function, a graph-based retrieval module, and a prompt construction function, as illustrated in Figure 1:

1. **Query Function** (ϕ_q): The query function transforms the input x into a query q for retrieval.
2. **Graph-Based Retrieval** ($\mathcal{R}$): The retrieval function $\mathcal{R}(q, G, k)$ takes as input the query q, the bipartite graph G, and a threshold k.

It first constructs the user profile P_i from G as defined above, and then retrieves the top-k most relevant entries from the user profile P_i with respect to q.

3. **Prompt Construction** (ϕ_p): The prompt construction assembles a personalized prompt for user i by combining the input x with the retrieved entries.

The final input to the LLM is a personalized, context-augmented prompt $\tilde{x}$ defined as:

$$\tilde{x} = \phi_p(x, \mathcal{R}(\phi_q(x), G, k)) \tag{4}$$

The pair $(\tilde{x}, y)$ is then used for inference or fine-tuning. This modular pipeline enables efficient, graph-aware personalization across diverse tasks and user sparsity levels.

Modularity and Extensibility. While we define P_i as a hybrid of user-authored and neighbor-authored interactions, PGraphRAG is modular by design. The underlying graph can be leveraged in alternative ways depending on the application: for example, practitioners may define P_i using only user-specific data, only neighbor interactions, or other graph-based traversal strategies (e.g., multi-hop reasoning or community-based filtering). Each component of the framework—query formulation, retrieval logic, and prompt construction—can be adapted independently, making PGraphRAG extensible to a wide range of personalized retrieval scenarios. In addition, the retrieval module supports plug-and-play compatibility with a variety of retrievers, such as BM25, or Contriever, allowing flexibility in balancing speed, semantic relevance, and computational cost.

4 Experiments

Setup. We evaluate our methods using two LLM backbones. The first is the LLaMA 3.1 8B Instruct model (Touvron et al., 2023), implemented with the Huggingface `transformers` library and configured to generate up to 512 tokens. The second is the GPT-4o-mini model (OpenAI, 2024), accessed via the Azure OpenAI Service (Services, 2023) using the `AzureOpenAI` interface, with a decoding temperature of 0.4. All experiments are conducted on an NVIDIA A100 GPU with 80GB of memory.

Dataset Splits and Graph Construction We construct bipartite user-entity graphs and split users into training, development, and test sets while preserving connectivity. Full details on data construction, neighbor filtering, and stratification are provided in Appendix A.

Graph Construction. We construct a bipartite user-entity graph from the selected user profiles in the validation and test splits. Each user node is connected to entity nodes (e.g., products, hotels, feedback targets) based on authored content, with edges representing user interactions such as reviews, summaries, or ratings. This graph supports two retrieval configurations: (1) *user-only*, which retrieves content authored solely by the target user (i.e., from their personal profile), and (2) *user+neighbor*, which additionally includes content from neighboring users who have interacted with the shared target entity. In both modes, the retrieved content defines the personalized context passed to the language model.

Ranking and Retrieval. The query used for retrieval varies by task type: for *Long Text Generation*, we use the review title; for *Short Text Generation*, the review text; and for *Ordinal Classification*, a combination of title and text. We apply two retrieval models—BM25 (Robertson and Zaragoza, 2009) and Contriever (Lei et al., 2023) to select the top-k most relevant entries from either the user-only or user+neighbor profiles. To enforce consistency between users with high activity and cold-start users, we cap retrieval at k, even if more candidate entries are available (see Table 7 and Figure 2). All textual inputs are tokenized using NLTK's `word_tokenize`. We use the default settings for both retrieval models; for Contriever, mean pooling is applied over token embeddings.

LLM Prompt Generation. Once the top-k entries are retrieved, we construct a *template-based prompt* that includes both the user's query (e.g., a request for a full review, a title, or a rating) and the contextual information from the graph. This prompt is passed to the LLM for generation. An illustration of task-specific prompt formatting is shown in Figure 4.

Baseline Methods. We compare PGraphRAG against both non-personalized and personalized baselines. (1) *No-Retrieval* constructs the prompt without any retrieval augmentation; the LLM generates the output solely from the query. (2) *Random-Retrieval* augments the prompt with content randomly sampled from all user profiles, introducing unrelated context. (3) *LaMP* (Salemi et al., 2024b) is a personalized baseline that augments the prompt

using content from the target user's own history (e.g., previously written reviews).

Evaluation. We evaluate each method by providing task-specific inputs and comparing generated outputs against reference labels. For generation tasks (long and short text), we report ROUGE-1, ROUGE-L (Lin, 2004), and METEOR (Banerjee and Lavie, 2005) scores. For rating prediction tasks, we measure mean absolute error (MAE) and root mean squared error (RMSE).

4.1 Baseline Comparison

We compare PGraphRAG against baselines on the three task types in our benchmark — long-text generation, short-text generation, and rating prediction.

Long Text Generation. Tables 3 and 16 show that PGraphRAG consistently outperforms all baseline methods—including No-Retrieval, Random-Retrieval, and LaMP—across ROUGE-1, ROUGE-L, and METEOR metrics. The largest performance gains are observed in Task Hotel Experience Generation, where PGraphRAG achieves +32.1% in ROUGE-1, +21.7% in ROUGE-L, and +25.7% in METEOR over the LaMP baseline using the LLaMA-3.1-8B-Instruct model. These improvements highlight the benefits of incorporating structured, graph-based context beyond user history.

Short Text Generation. Tables 4 and 17 show that PGraphRAG outperforms the baselines in most cases. In Task User Product Review Title Generation, PGraphRAG achieves consistent gains over LaMP in the LLaMA-3.1-8B-Instruct model: ROUGE-1 (+5.6%), ROUGE-L (+5.9%), and METEOR (+6.8%). These improvements, while smaller than those in long-form tasks, reflect the limited headroom for personalization in very short text generation tasks such as review title. Because the target texts are extremely brief, minor lexical differences can significantly affect overlap-based metrics, and there are fewer opportunities for retrieved context to meaningfully influence generation.

Ordinal Classification. Tables 8 and 18 show that PGraphRAG yields modest improvements over LaMP in rating prediction tasks. It outperforms LaMP in 1 out of 4 tasks with LLaMA-3.1-8B-Instruct and in 2 out of 4 tasks with GPT. The largest gains are observed on the Multilingual Product Ratings task, with improvements in MAE (+1.75%) and RMSE (+1.12%) for LLaMA-3.1-8B-Instruct , and MAE (+2.16%) and RMSE (+3.17%) for GPT. These gains, while small, suggest that user profiles can aid numerical prediction when meaningful variability exists across user preferences. In domains like hotel experiences or digital products, where user expectations tend to be homogeneous, graph-based personalization may offer limited additional signal.

4.2 Ablation Studies

We conduct ablation experiments to assess the impact of different retrieval configurations on PGraphRAG's performance. Specifically, we vary the retrieval depth (i.e., top-k), the retrieval scope (user-only vs. user+neighbors), and the retriever model (BM25 vs. Contriever). Full results and analysis are provided in Appendix A.

5 Conclusion

We presented PGraphRAG, a framework that enhances personalized text generation by integrating user-centric knowledge graphs into retrieval-augmented generation. Unlike prior methods that rely solely on user history, PGraphRAG enriches generation with structured user profiles, enabling adaptive personalization even in sparse data settings. Our experiments show that graph-based retrieval significantly improves performance across diverse tasks, outperforming state-of-the-art baselines. Beyond improved metrics, PGraphRAG introduces a scalable design that generalizes user preferences and adapts to new users through structural retrieval. This work lays a foundation for future personalized LLM systems, particularly in applications requiring robustness to data sparsity, cold starts, and context adaptation.

References

Oshin Agarwal, Heming Ge, Siamak Shakeri, and Rami Al-Rfou. 2021. Knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3554–3565, Online. 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*

Long Text Generation	Metric	PGraphRAG	LaMP	No-Retrieval	Random-Retrieval
LLaMA-3.1-8B-Instruct					
Task 1: User-Product Review Generation	ROUGE-1	**0.178**	0.173	0.172	0.124
	ROUGE-L	**0.129**	0.129	0.123	0.094
	METEOR	0.151	0.138	**0.154**	0.099
Task 2: Hotel Experiences Generation	ROUGE-1	**0.263**	0.199	0.231	0.216
	ROUGE-L	**0.157**	0.129	0.145	0.132
	METEOR	**0.191**	0.152	0.153	0.152
Task 3: Stylized Feedback Generation	ROUGE-1	**0.217**	0.186	0.190	0.184
	ROUGE-L	**0.158**	0.134	0.131	0.108
	METEOR	**0.178**	0.177	0.167	0.122
Task 4: Multilingual Product Review Generation	ROUGE-1	**0.188**	0.176	0.174	0.146
	ROUGE-L	**0.147**	0.141	0.136	0.116
	METEOR	**0.145**	0.125	0.131	0.109
GPT-4o-mini					
Task 1: User-Product Review Generation	ROUGE-1	**0.189**	0.171	0.169	0.159
	ROUGE-L	**0.130**	0.117	0.116	0.114
	METEOR	**0.196**	0.176	0.177	0.153
Task 2: Hotel Experiences Generation	ROUGE-1	**0.263**	0.221	0.223	0.234
	ROUGE-L	**0.152**	0.135	0.135	0.139
	METEOR	**0.206**	0.164	0.166	0.181
Task 3: Stylized Feedback Generation	ROUGE-1	**0.211**	0.185	0.187	0.177
	ROUGE-L	**0.140**	0.123	0.123	0.121
	METEOR	**0.202**	0.183	0.189	0.165
Task 4: Multilingual Product Review Generation	ROUGE-1	**0.194**	0.168	0.170	0.175
	ROUGE-L	**0.144**	0.125	0.128	0.133
	METEOR	**0.171**	0.154	0.152	0.149

Table 3: Zero-shot performance on the test set for the Long Text Generation tasks using *LLaMA-3.1-8B-Instruct* and *GPT-4o-mini*. For each model, the best retriever configuration was selected based on validation performance.

Short Text Generation	Metric	PGraphRAG	LaMP	No-Retrieval	Random-Retrieval
LLaMA-3.1-8B-Instruct					
Task 5: User Product Review Title Generation	ROUGE-1	**0.131**	0.124	0.121	0.103
	ROUGE-L	**0.125**	0.118	0.115	0.098
	METEOR	**0.125**	0.117	0.112	0.096
Task 6: Hotel Experience Summary Generation	ROUGE-1	**0.127**	0.126	0.122	0.118
	ROUGE-L	**0.118**	0.117	0.114	0.110
	METEOR	0.102	**0.106**	0.101	0.093
Task 7: Stylized Feedback Title Generation	ROUGE-1	**0.149**	0.140	0.136	0.133
	ROUGE-L	**0.142**	0.134	0.131	0.123
	METEOR	**0.142**	0.136	0.129	0.121
Task 8: Multi-lingual Review Title Generation	ROUGE-1	0.124	0.121	**0.125**	0.120
	ROUGE-L	0.116	**0.122**	0.117	0.110
	METEOR	**0.108**	0.094	0.092	0.103
GPT-4o-mini					
Task 5: User Product Review Title Generation	ROUGE-1	**0.115**	0.108	0.113	0.102
	ROUGE-L	**0.112**	0.105	0.110	0.099
	METEOR	**0.099**	0.091	0.093	0.085
Task 6: Hotel Experience Summary Generation	ROUGE-1	**0.116**	0.108	0.114	0.112
	ROUGE-L	**0.111**	0.104	0.109	0.107
	METEOR	**0.081**	0.075	0.079	0.076
Task 7: Stylized Feedback Title Generation	ROUGE-1	**0.122**	0.113	0.114	0.115
	ROUGE-L	**0.118**	0.109	0.110	0.111
	METEOR	**0.104**	0.096	0.097	0.093
Task 8: Multi-lingual Review Title Generation	ROUGE-1	0.111	0.115	**0.118**	0.108
	ROUGE-L	0.105	0.107	**0.110**	0.102
	METEOR	0.083	0.088	**0.089**	0.078

Table 4: Zero-shot performance on the test set for the Short Text Generation tasks using *LLaMA-3.1-8B-Instruct* and *GPT-4o-mini*. For each model, the best retriever configuration was selected based on validation performance.

AI Systems (PERSONALIZE 2024), pages 88–101, St. Julians, Malta. Association for Computational Linguistics.

Reinald Kim Amplayo, Jihyeok Kim, Sua Sung, and Seung-won Hwang. 2018. Cold-start aware user and product attention for sentiment classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2535–2544, Melbourne, Australia. Association for Computational Linguistics.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2023. Benchmarking large language models in retrieval-augmented generation.

Eduardo G. Cortes, Ana Luiza Vianna, Mikaela Martins, Sandro Rigo, and Rafael Kunst. 2024. LLMs and translation: different approaches to localization between Brazilian Portuguese and European Portuguese. In *Proceedings of the 16th International Conference on Computational Processing of Portuguese - Vol. 1*, pages 45–55, Santiago de Compostela, Galicia/Spain. Association for Computational Linguistics.

Datafiniti. 2017. Hotel reviews, version 5. Retrieved September 15, 2024 from `https://www.kaggle.com/datasets/datafiniti/hotel-reviews/data`.

Datafiniti. 2018. Grammar and online product reviews, version 1. Retrieved September 15, 2024 from `https://www.kaggle.com/datasets/datafiniti/grammar-and-online-product-reviews`.

Gladson de Araujo, Tiago de Melo, and Carlos Maurício S. Figueiredo. 2024. Is ChatGPT an effective solver of sentiment analysis tasks in Portuguese? a preliminary study. In *Proceedings of the 16th International Conference on Computational Processing of Portuguese - Vol. 1*, pages 13–21, Santiago de Compostela, Galicia/Spain. Association for Computational Linguistics.

Li Dong, Shaohan Huang, Furu Wei, Mirella Lapata, Ming Zhou, and Ke Xu. 2017. Learning to generate product reviews from attributes. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 623–632, Valencia, Spain. Association for Computational Linguistics.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey.

Sebastian Gehrmann, Tosin Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Anuoluwapo Aremu, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna-Adriana Clinciu, Dipanjan Das, Kaustubh Dhole, Wanyu Du, Esin Durmus, Ondřej Dušek, Chris Chinenye Emezue, Varun Gangal, Cristina Garbacea, Tatsunori Hashimoto, Yufang Hou, Yacine Jernite, Harsh Jhamtani, Yangfeng Ji, Shailza Jolly, Mihir Kale, Dhruv Kumar, Faisal Ladhak, Aman Madaan, Mounica Maddela, Khyati Mahajan, Saad Mahamood, Bodhisattwa Prasad Majumder, Pedro Henrique Martins, Angelina McMillan-Major, Simon Mille, Emiel van Miltenburg, Moin Nadeem, Shashi Narayan, Vitaly Nikolaev, Andre Niyongabo Rubungo, Salomey Osei, Ankur Parikh, Laura Perez-Beltrachini, Niranjan Ramesh Rao, Vikas Raunak, Juan Diego Rodriguez, Sashank Santhanam, João Sedoc, Thibault Sellam, Samira Shaikh, Anastasia Shimorina, Marco Antonio Sobrevilla Cabezudo, Hendrik Strobelt, Nishant Subramani, Wei Xu, Diyi Yang, Akhila Yerukola, and Jiawei Zhou. 2021. The GEM benchmark: Natural language generation, its evaluation and metrics. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 96–120, Online. Association for Computational Linguistics.

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*.

Xiaolei Huang, Lucie Flek, Franck Dernoncourt, Charles Welch, Silvio Amir, Ramit Sawhney, and Diyi Yang. 2022. Usernlp'22: 2022 international workshop on user-centered natural language processing. In *Companion Proceedings of the Web Conference 2022*, WWW '22, page 1176–1177, New York, NY, USA. Association for Computing Machinery.

Yu-Yang Huang, Rui Yan, Tsung-Ting Kuo, and Shou-De Lin. 2014. Enriching cold start personalized language model using social network information. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 611–617, Baltimore, Maryland. Association for Computational Linguistics.

Hayate Iso, Xiaolan Wang, and Yoshi Suhara. 2024. Noisy pairing and partial supervision for stylized opinion summarization. In *Proceedings of the 17th International Natural Language Generation Conference*, pages 13–23, Tokyo, Japan. Association for Computational Linguistics.

Gautier Izacard and Edouard Grave. 2020. Leveraging passage retrieval with generative models for open domain question answering. *CoRR*, abs/2007.01282.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. 2022. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems*, 33(2):494–514.

Srinivas Ramesh Kamath, Fahime Same, and Saad Mahamood. 2024. Generating hotel highlights from unstructured text using LLMs. In *Proceedings of the 17th International Natural Language Generation Conference*, pages 280–288, Tokyo, Japan. Association for Computational Linguistics.

Minki Kang, Jin Myung Kwak, Jinheon Baek, and Sung Ju Hwang. 2023. Knowledge graph-augmented language models for knowledge-grounded dialogue generation.

Shin Kanouchi, Masato Neishi, Yuta Hayashibe, Hiroki Ouchi, and Naoaki Okazaki. 2020. You may like this hotel because ...: Identifying evidence for explainable recommendations. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 890–899, Suzhou, China. Association for Computational Linguistics.

Jihyeok Kim, Seungtaek Choi, Reinald Kim Amplayo, and Seung-won Hwang. 2020. Retrieval-augmented controllable review generation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2284–2295, Barcelona, Spain (Online). International Committee on Computational Linguistics.

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.

Yibin Lei, Liang Ding, Yu Cao, Changtong Zan, Andrew Yates, and Dacheng Tao. 2023. Unsupervised dense retrieval with relevance-aware contrastive pretraining. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10932–10940, Toronto, Canada. Association for Computational Linguistics.

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

Ziqing Liu, Enwei Peng, Shixing Yan, Guozheng Li, and Tianyong Hao. 2018. T-know: a knowledge graph-based question answering and infor-mation retrieval system for traditional Chinese medicine. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 15–19, Santa Fe, New Mexico. Association for Computational Linguistics.

Hanjia Lyu, Song Jiang, Hanqing Zeng, Yinglong Xia, Qifan Wang, Si Zhang, Ren Chen, Christopher Leung, Jiajie Tang, and Jiebo Luo. 2024. Llm-rec: Personalized recommendation via prompting large language models.

Puneet Mathur, Zhe Liu, Ke Li, Yingyi Ma, Gil Karen, Zeeshan Ahmed, Dinesh Manocha, and Xuedong Zhang. 2024. DOC-RAG: ASR language model personalization with domain-distributed co-occurrence retrieval augmentation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 5132–5139, Torino, Italia. ELRA and ICCL.

Maxim Naumov, Dheevatsa Mudigere, Hao-Jun Michael Shi, Jianyu Huang, Narayanan Sundaraman, Jongsoo Park, Xiaodong Wang, Udit Gupta, Carole-Jean Wu, Alisson G. Azzolini, Dmytro Dzhulgakov, Andrey Mallevich, Ilia Cherniavskii, Yinghai Lu, Raghuraman Krishnamoorthi, Ansha Yu, Volodymyr Kondratenko, Stephanie Pereira, Xianjie Chen, Wenlin Chen, Vijay Rao, Bill Jia, Liang Xiong, and Misha Smelyanskiy. 2019. Deep learning recommendation model for personalization and recommendation systems. *CoRR*, abs/1906.00091.

Jianmo Ni and Julian McAuley. 2018. Personalized review generation by expanding phrases and attending on aspect-aware representations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 706–711, Melbourne, Australia. Association for Computational Linguistics.

OpenAI. 2024. Gpt-4o system card.

Heiko Paulheim. 2017. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic web*, 8(3):489–508.

Livy Real, Marcio Oshiro, and Alexandre Mafra. 2019. B2w-reviews01: an open product reviews corpus. In *STIL-Symposium in Information and Human Language Technology*.

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.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*, 3:333–389.

Ahmmad O. M. Saleh, Gokhan Tur, and Yücel Saygín. 2024. Sg-rag: Multi-hop question answering with large language models through knowledge graphs. In *International Conference on Natural Language and Speech Processing*.

Alireza Salemi, Surya Kallumadi, and Hamed Zamani. 2024a. Optimization methods for personalizing large language models through retrieval augmentation.

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

Mikhail Salnikov, Hai Le, Prateek Rajput, Irina Nikishina, Pavel Braslavski, Valentin Malykh, and Alexander Panchenko. 2023. Large language models meet knowledge graphs to answer factoid questions. In *Proceedings of the 37th Pacific Asia Conference on Language, Information and Computation*, pages 635–644, Hong Kong, China. Association for Computational Linguistics.

Arthur Scalercio, Maria Finatto, and Aline Paes. 2024. Enhancing sentence simplification in Portuguese: Leveraging paraphrases, context, and linguistic features. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 15076–15091, Bangkok, Thailand. Association for Computational Linguistics.

Phillip Schneider, Tim Schopf, Juraj Vladika, Mikhail Galkin, Elena Simperl, and Florian Matthes. 2022. A decade of knowledge graphs in natural language processing: A survey. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 601–614, Online only. Association for Computational Linguistics.

Azure AI Services. 2023. Openai (gpt-4o-mini-20240718) [large language model]. https://learn.microsoft.com/en-us/azure/ai-services/openai.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019a. *SuperGLUE: a stickier benchmark for general-purpose language understanding systems*. Curran Associates Inc., Red Hook, NY, USA.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019b. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*.

Hongyan Xu, Hongtao Liu, Zhepeng Lv, Qing Yang, and Wenjun Wang. 2023. Pre-trained personalized review summarization with effective salience estimation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10743–10754, Toronto, Canada. Association for Computational Linguistics.

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

Jianfei Yu, Chenggong Gong, and Rui Xia. 2021. Cross-domain review generation for aspect-based sentiment analysis. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4767–4777, Online. Association for Computational Linguistics.

Hongyu Zang and Xiaojun Wan. 2017. Towards automatic generation of product reviews from aspect-sentiment scores. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 168–177, Santiago de Compostela, Spain. Association for Computational Linguistics.

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, Nesreen Ahmed, and Yu Wang. 2024. Personalization of large language models: A survey.

A Appendix

A.1 Data Construction and Splitting

To construct the user–item interaction graph, we represent users and domain-specific entities (e.g., products, hotels, feedback targets) as nodes, with edges corresponding to user-generated content (e.g., reviews, summaries, ratings). To support graph-based personalization, we require that each selected user has at least one interaction with an entity that is also associated with another user — i.e., a shared neighbor in the bipartite graph. If a randomly selected user interaction does not meet this criterion, we instead sample a different interaction from the same profile. Users without any neighbor-compatible interactions remain in the dataset but are excluded from gold-label selection, since sampling is performed at the edge level rather than over full profiles. This filtering ensures that the graph remains connected and supports comparative evaluation and cold-start scenarios, where even users with minimal history share contextually linked entities with others.

After identifying each user's valid neighbor-linked interaction(s), we divide users into training, development, and test sets while preserving graph connectivity across splits. To ensure that personalization signals remain intact, we apply two levels of neighbor preservation:

1. **Global Neighbor Preservation:** Entities with multiple associated users are grouped so that at least one other user in the same split has interacted with the same entity.

2. **Local Neighbor Preservation:** Once a user is assigned to a split, any other users who interacted with the same entity are also placed in that split to maintain graph connectivity.

We further stratify each split based on user profile size to match the original distribution of user activity while preserving both global and local connectivity. This joint control over profile stratification and neighbor assignment ensures that the resulting graphs in each split maintain realistic interaction patterns and structural properties. Graph statistics are shown in Table 2, task-level data statistics in Table 1, and dataset splits in Table 5.

A.2 Performance Gains

Table 6 shows the relative percent gains of PGraphRAG compared to LaMP across Tasks 1–7.

Dataset	Train Size	Validation Size	Test Size
User-Product Review	20,000	2,500	2,500
Multilingual Product Review	20,000	2,500	2,500
Stylized Feedback	20,000	2,500	2,500
Hotel Experiences	9,000	2,500	2,500

Table 5: Dataset split sizes across training, validation, and test sets for the four domains.

Notably, Task 8 (Multi-lingual Review Title Generation) shows reduced gains, which we attribute to cultural differences in review conventions—for example, the frequent use of the generic phrase Muito bom" (Very good") in Brazilian Portuguese titles. In long-text generation with GPT-4o-mini, PGraphRAG achieves improvements of approximately 15% in ROUGE-1, 13% in ROUGE-L, and 15% in METEOR. Similar trends are seen with LLaMA-3.1-8B, with improvements of 15%, 11%, and 13% respectively. In short-text generation, GPT shows improvements of 5% across all metrics, while LLaMA gains range from 2–6%.

In addition, Table 7 shows the review density per product, where sparsity is balanced from the original graph for both product and user nodes.

A.3 Prompt and Output Examples

Figure 4 shows the prompt template across task types. Below, we show the output for Task 2 comparing PGraphRAG and LaMP against the gold label. PGraphRAG captures specific contextual cues from the graph (e.g., correct location and hotel amenities), while LaMP's output often relies on generic context from the target user's own reviews, which leads to generating irrelevant content (e.g., wrong hotel location).

Gold Review

Title: Amazing stay! **Review:** Had a great stay, our room was very clean and very peaceful. It is in the heart of DT Seattle. We took the train to the hotel and it was pretty easy. We enjoyed walking all over downtown Seattle—no need to rent a car. Major shopping and amazing restaurants right around the corner! We will stay here again!

Model	Metric	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8
	ROUGE-1	10.53	18.96	14.05	15.48	6.48	7.41	7.96	-3.48
GPT-4o-mini	ROUGE-L	11.11	12.59	13.82	15.20	6.67	6.73	8.26	-1.87
	METEOR	11.36	25.61	10.38	11.04	8.79	8.00	8.33	-5.68
	ROUGE-1	2.89	32.16	16.67	6.82	5.65	0.79	6.43	2.48
LLaMA-3.1-8B-Instruct	ROUGE-L	0.00	21.71	17.91	4.26	5.93	0.85	5.97	-4.92
	METEOR	9.42	25.66	0.56	16.00	6.84	-3.77	4.41	14.89

Table 6: Relative percentage gains of PGraphRAG over LaMP across Tasks 1–8 using *GPT-4o-mini* and *LLaMA-3.1-8B-Instruct*.

Reviews (n)	**Exact Count (Pct.)**	**Cumulative Count (Pct.)**
1	25,530 (49.69%)	25,530 (49.69%)
2	9,488 (18.47%)	35,018 (68.16%)
3	4,784 (9.31%)	39,802 (77.47%)
4	2,639 (5.14%)	42,441 (82.61%)
5	1,836 (3.57%)	44,277 (86.18%)

Table 7: Distribution of review counts per product in the Amazon Review dataset (Tasks 1, 5, and 9). Most products have five or fewer reviews.

LaMP

Title: Amazing stay! **Review:** I recently had an amazing stay at the Newark Garden Hotel. Despite the circumstances that brought my family together, the hotel provided a comforting and welcoming environment. The rooms were immaculate, and the staff went above and beyond to ensure we felt at home. The hot breakfast each morning was a delightful touch that made our mornings brighter. I highly recommend this hotel for anyone looking for a pleasant and accommodating stay!

PGraphRAG

Title: Amazing stay! **Review:** I had an amazing stay at the Grand Hyatt Seattle! From the moment I checked in, the staff was incredibly welcoming and attentive. The room was spacious, clean, and beautifully appointed, with a stunning view of the city. The bathroom was a highlight, featuring a luxurious soaking tub and a large walk-in shower. The location was perfect, just a short walk to Pike Place Market and other attractions. I also enjoyed the fantastic amenities, including the well-equipped gym and the delicious food options available at the hotel. Overall, it was a wonderful experience, and I can't wait to return!

A.4 PGraphRAG Ablation Details

To assess the contributions of user-specific and neighbor-derived context in our retrieval framework, we conduct an ablation study comparing three variants of PGraphRAG:

- **PGraphRAG**: The full method, which retrieves context from both the target user's profile and neighboring users who share entities (e.g., items or experiences).
- **PGraphRAG-N**: A neighbor-only variant that excludes the target user's own interactions and relies solely on neighboring users for context.
- **PGraphRAG-U**: A user-only variant that restricts retrieval to the target user's own history, ignoring all neighbor signals.

Table 9 shows the results for long-text generation (Tasks 1–4) using GPT-4o-mini and LLaMA-3.1-8B. Both PGraphRAG and PGraphRAG-N consistently outperform PGraphRAG-U across datasets, highlighting the value of graph-based retrieval. Notably, PGraphRAG-N performs on par with or slightly below the full PGraphRAG method, suggesting that neighboring-user context alone is often sufficient for high-quality personalization — especially in low-profile or cold-start scenarios where the target user's history is sparse.

Results for short-text generation tasks (Tasks 5–8) are shown in Table 10. Similar patterns hold, with PGraphRAG and PGraphRAG-N outperforming PGraphRAG-U across most tasks. One exception is Task Hotel Experience Summary Generation, where PGraphRAG-U slightly outperforms all graph-based variants, possibly due to limited variation in the data or a mismatch between neighbor context and task-specific semantics.

A.5 Impact of the Retrieved Items k

To understand how the size of the retrieved context affects performance, we conduct an ablation study varying the number of retrieved entries $k \in 1, 2, 4$. Table 11 reports results for long-text generation (Tasks 1–4), using *GPT-4o-mini* and *LLaMA-3.1-8B-Instruct*. Corresponding results for short-text generation (Tasks 5–8) appear in Table 12.

Ordinal Classfication	Metric	PGraphRAG	LaMP	No-retrieval	Random-retrieval
LLaMA-3.1-8B-Instruct					
Task 9: User Product Review Ratings	MAE ↓	0.3400	**0.3132**	0.3212	0.3272
	RMSE ↓	0.7668	**0.7230**	0.7313	0.7616
Task 10: Hotel Experience Ratings	MAE ↓	0.3688	0.3492	**0.3340**	0.3804
	RMSE ↓	0.6771	0.6527	**0.6372**	0.6971
Task 11: Stylized Feedback Ratings	MAE ↓	0.3476	**0.3268**	0.3256	0.3704
	RMSE ↓	0.7247	**0.6803**	0.6806	0.7849
Task 12: Multi-lingual Product Ratings	MAE ↓	**0.4928**	0.5016	0.5084	0.5096
	RMSE ↓	**0.8367**	0.8462	0.8628	0.8542
GPT-4o-mini					
Task 9: User Product Review Ratings	MAE ↓	0.3832	0.3480	**0.3448**	0.4188
	RMSE ↓	0.7392	**0.7065**	**0.7065**	0.8082
Task 10: Hotel Experience Ratings	MAE ↓	**0.3284**	0.3336	0.3336	0.3524
	RMSE ↓	**0.6083**	0.6197	0.6197	0.6384
Task 11: Stylized Feedback Ratings	MAE ↓	0.3476	**0.3448**	0.3416	0.4080
	RMSE ↓	0.6738	**0.6669**	0.6711	0.7370
Task 12: Multi-lingual Product Ratings	MAE ↓	**0.4348**	0.4444	0.4564	0.4700
	RMSE ↓	**0.7367**	0.7608	0.7718	0.8112

Table 8: Performance comparison on rating prediction tasks (Tasks 9-12) using *GPT-4o-mini* and *LLaMA-3.1-8B*.

Long Text Generation	Metric	PGraphRAG	PGraphRAG-N	PGraphRAG-U
LLaMA-3.1-8B-Instruct				
Task 1: User-Product Review Generation	ROUGE-1	0.173	**0.177**	0.168
	ROUGE-L	0.124	**0.127**	0.125
	METEOR	0.150	**0.154**	0.134
Task 2: Hotel Experiences Generation	ROUGE-1	0.263	**0.272**	0.197
	ROUGE-L	0.156	**0.162**	0.128
	METEOR	0.191	**0.195**	0.121
Task 3: Stylized Feedback Generation	ROUGE-1	**0.226**	0.222	0.181
	ROUGE-L	**0.171**	0.165	0.134
	METEOR	**0.192**	0.186	0.147
Task 4: Multilingual Product Review Generation	ROUGE-1	**0.174**	0.172	0.174
	ROUGE-L	0.139	0.137	**0.141**
	METEOR	**0.133**	0.126	0.125
GPT-4o-mini				
Task 1: User-Product Review Generation	ROUGE-1	**0.186**	0.185	0.169
	ROUGE-L	**0.126**	0.125	0.114
	METEOR	**0.187**	0.185	0.170
Task 2: Hotel Experiences Generation	ROUGE-1	0.265	**0.268**	0.217
	ROUGE-L	0.152	**0.153**	0.132
	METEOR	0.206	**0.209**	0.161
Task 3: Stylized Feedback Generation	ROUGE-1	**0.205**	0.204	0.178
	ROUGE-L	**0.139**	0.138	0.121
	METEOR	**0.203**	0.198	0.178
Task 4: Multilingual Product Review Generation	ROUGE-1	**0.191**	0.190	0.164
	ROUGE-L	**0.142**	0.140	0.123
	METEOR	**0.173**	0.169	0.155

Table 9: Ablation study results for long text generation tasks using *LLaMA-3.1-8B-Instruct* and *GPT-4o-mini*. PGraphRAG-N represents Neighbors-only context retrieval and PGraphRAG-U represents User-only context retrieval.

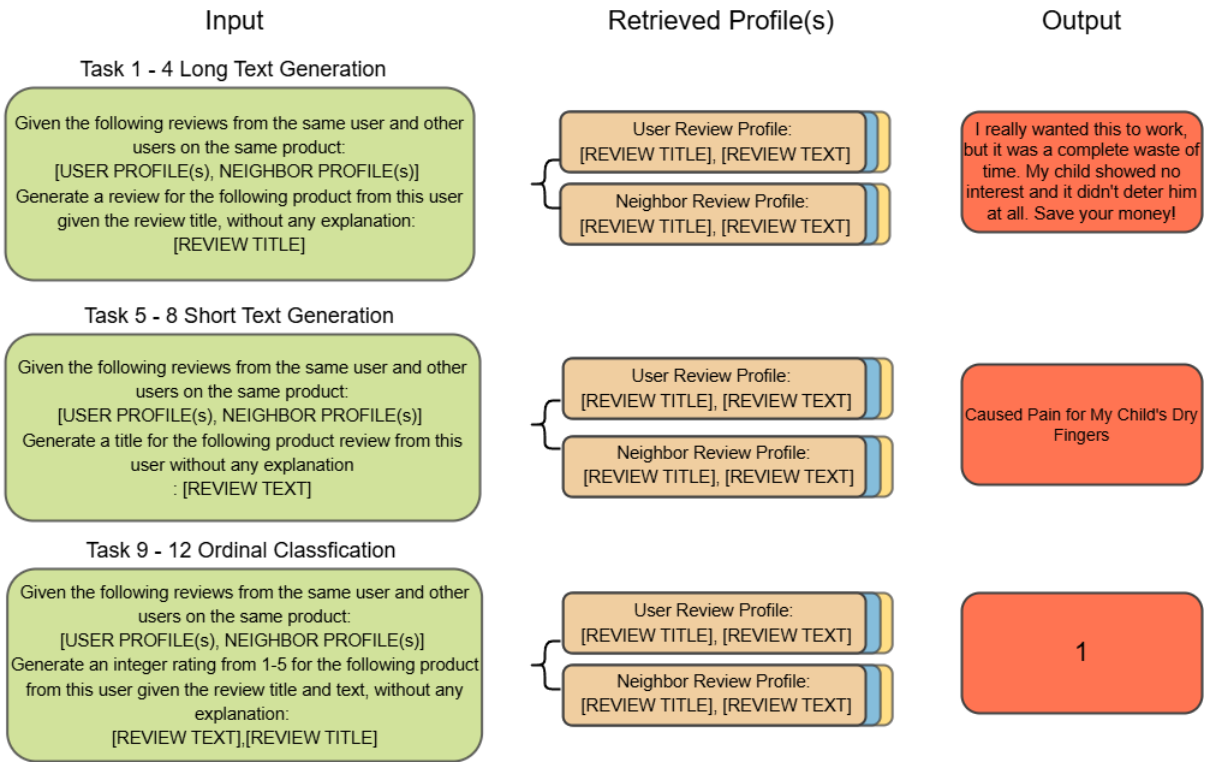

Figure 4: Prompt configurations used for each task type. Teletype placeholders (e.g., `{{title}}`) are replaced with task-specific input and retrieved context at inference time.

Short Text Generation	**Metric**	**PGraphRAG**	**PGraphRAG-N**	**PGraphRAG-U**
LLaMA-3.1-8B-Instruct				
Task 5: User Product Review Title Generation	ROUGE-1	0.125	**0.129**	0.115
	ROUGE-L	0.119	**0.123**	0.109
	METEOR	0.117	**0.120**	0.111
Task 6: Hotel Experience Summary Generation	ROUGE-1	0.121	**0.124**	0.119
	ROUGE-L	0.113	**0.115**	0.111
	METEOR	0.099	0.103	**0.105**
Task 7: Stylized Feedback Title Generation	ROUGE-1	0.132	**0.135**	0.128
	ROUGE-L	0.128	**0.130**	0.124
	METEOR	0.129	**0.132**	0.124
Task 8: Multi-lingual Product Review Title Generation	ROUGE-1	**0.131**	0.131	0.124
	ROUGE-L	**0.123**	0.122	0.114
	METEOR	**0.118**	0.110	0.098
GPT-4o-mini				
Task 5: User Product Review Title Generation	ROUGE-1	0.111	**0.116**	0.112
	ROUGE-L	0.106	**0.111**	0.108
	METEOR	0.097	**0.099**	0.095
Task 6: Hotel Experience Summary Generation	ROUGE-1	0.118	**0.119**	0.109
	ROUGE-L	0.112	**0.113**	0.104
	METEOR	**0.085**	**0.085**	0.077
Task 7: Stylized Feedback Title Generation	ROUGE-1	**0.109**	0.107	0.108
	ROUGE-L	**0.107**	0.105	0.104
	METEOR	**0.096**	0.094	0.091
Task 8: Multi-lingual Product Review Title Generation	ROUGE-1	0.108	0.109	**0.116**
	ROUGE-L	0.104	0.104	**0.109**
	METEOR	0.082	0.089	**0.091**

Table 10: Ablation study results for short text generation tasks using *LLaMA-3.1-8B-Instruct* and *GPT-4o-mini*. PGraphRAG-N represents Neighbors-only context retrieval and PGraphRAG-U represents User-only context retrieval.

Overall, increasing k generally leads to improved generation performance across tasks and models. This trend highlights the value of larger retrieved contexts, which provide richer signals about user preferences and item semantics. The gains are especially evident when moving from $k = 1$ to $k = 2$, though marginal returns diminish between $k = 2$ and $k = 4$ in some cases.

That said, the benefit of higher k values is constrained by data sparsity. Many user profiles contain fewer than four qualifying interactions—especially in cold-start settings. In such cases, the retriever returns all available entries, even if they are fewer than the specified k. As a result, the effective retrieved context size varies across users, especially in the low-profile regime. This behavior reflects the practical limitations of personalization at scale and underscores the importance of designing retrieval-aware systems that can operate under sparse supervision.

Long Text Generation	**Metric**	$k = 1$	$k = 2$	$k = 4$
LLaMA-3.1-8B-Instruct				
Task 1: User-Product Review Generation	ROUGE-1	0.160	0.169	**0.173**
	ROUGE-L	0.121	**0.125**	0.124
	METEOR	0.125	0.138	**0.150**
Task 2: Hotel Experiences Generation	ROUGE-1	0.230	0.251	**0.263**
	ROUGE-L	0.141	0.151	**0.156**
	METEOR	0.152	0.174	**0.191**
Task 3: Stylized Feedback Generation	ROUGE-1	0.200	0.214	**0.226**
	ROUGE-L	0.158	0.165	**0.171**
	METEOR	0.154	0.171	**0.192**
Task 4: Multilingual Product Review Generation	ROUGE-1	0.163	0.169	**0.174**
	ROUGE-L	0.134	0.137	**0.139**
	METEOR	0.113	0.122	**0.133**
GPT-4o-mini				
Task 1: User-Product Review Generation	ROUGE-1	0.176	0.184	**0.186**
	ROUGE-L	0.121	0.125	**0.126**
	METEOR	0.168	0.180	**0.187**
Task 2: Hotel Experiences Generation	ROUGE-1	0.250	0.260	**0.265**
	ROUGE-L	0.146	0.150	**0.152**
	METEOR	0.188	0.198	**0.206**
Task 3: Stylized Feedback Generation	ROUGE-1	0.196	0.200	**0.205**
	ROUGE-L	0.136	0.136	**0.139**
	METEOR	0.186	0.192	**0.203**
Task 4: Multilingual Product Review Generation	ROUGE-1	0.163	0.169	**0.174**
	ROUGE-L	0.134	0.137	**0.139**
	METEOR	0.113	0.122	**0.133**

Table 11: Ablation study results showing the impact of varying k (number of retrieved neighbors) on PGraphRAG's performance. Results are reported for *LLaMA-3.1-8B-Instruct* and *GPT-4o-mini* on long-text generation tasks (Tasks 1 - 4).

A.6 Impact of Retriever Method $\mathcal{R}$

We evaluate how the choice of retriever affects the performance of PGraphRAG by comparing two retrieval backends: BM25, a sparse keyword-based retriever, and Contriever, a dense unsupervised retriever based on sentence embeddings.

Short Text Generation	**Metric**	$k = 1$	$k = 2$	$k = 4$
LLaMA-3.1-8B-Instruct				
Task 5: User Product Review Title Generation	ROUGE-1	**0.128**	0.123	0.125
	ROUGE-L	**0.121**	0.118	0.119
	METEOR	**0.123**	0.118	0.117
Task 6: Hotel Experience Summary Generation	ROUGE-1	**0.122**	0.121	0.121
	ROUGE-L	0.112	**0.114**	0.113
	METEOR	**0.104**	0.102	0.099
Task 7: Stylized Feedback Title Generation	ROUGE-1	0.129	**0.132**	**0.132**
	ROUGE-L	0.124	0.126	**0.128**
	METEOR	0.129	**0.130**	0.129
Task 8: Multi-lingual Product Review Title Generation	ROUGE-1	0.129	0.126	**0.131**
	ROUGE-L	0.120	0.119	**0.123**
	METEOR	0.117	0.116	**0.118**
GPT-4o-mini				
Task 5: User Product Review Title Generation	ROUGE-1	**0.111**	0.110	**0.111**
	ROUGE-L	**0.106**	0.105	**0.106**
	METEOR	0.093	0.094	**0.097**
Task 6: Hotel Experience Summary Generation	ROUGE-1	0.114	0.114	**0.118**
	ROUGE-L	0.109	0.109	**0.112**
	METEOR	0.082	0.082	**0.085**
Task 7: Stylized Feedback Title Generation	ROUGE-1	0.100	0.103	**0.109**
	ROUGE-L	0.098	0.101	**0.107**
	METEOR	0.087	0.090	**0.096**
Task 8: Multi-lingual Product Review Title Generation	ROUGE-1	0.104	0.104	**0.108**
	ROUGE-L	0.098	0.098	**0.104**
	METEOR	0.077	0.078	**0.082**

Table 12: Ablation study results showing the impact of varying k (number of retrieved neighbors) on PGraphRAG's performance. Results are reported for *LLaMA-3.1-8B-Instruct* and *GPT-4o-mini* on short-text generation tasks (Tasks 5-8).

Table 13 reports results for long-text generation (Tasks 1–4), and Table 14 provides results for short-text generation (Tasks 5–8). Across both GPT-4o-mini and LLaMA-3.1-8B-Instruct models, we observe that PGraphRAG performs consistently well regardless of the retrieval method. The differences between BM25 and Contriever are minor, and no retriever dominates across all datasets or metrics.

These findings indicate that PGraphRAG is robust to the choice of retriever and does not rely on fine-tuned or heavily engineered retrieval strategies. While BM25 sometimes yields slightly higher scores, the overall parity suggests that our graph-based retrieval and prompting framework can effectively integrate contextual signals from either sparse or dense retrieval methods.

A.7 Impact of Ranked Retrieval

Table 15 evaluates the role of ranking in PGraphRAG by comparing the following retrieval variants:

Long Text Generation	Metric	Contriever	BM25
LLaMA-3.1-8B-Instruct			
Task 1: User-Product Review Generation	ROUGE-1	0.172	**0.173**
	ROUGE-L	0.122	**0.124**
	METEOR	**0.153**	0.150
Task 2: Hotel Experiences Generation	ROUGE-1	0.262	**0.263**
	ROUGE-L	0.155	**0.156**
	METEOR	0.190	**0.191**
Task 3: Stylized Feedback Generation	ROUGE-1	0.195	**0.226**
	ROUGE-L	0.138	**0.171**
	METEOR	0.180	**0.192**
Task 4: Multilingual Product Review Generation	ROUGE-1	0.172	**0.174**
	ROUGE-L	0.134	**0.139**
	METEOR	**0.135**	0.133
GPT-4o-mini			
Task 1: User-Product Review Generation	ROUGE-1	0.182	**0.186**
	ROUGE-L	0.122	**0.126**
	METEOR	0.184	**0.187**
Task 2: Hotel Experiences Generation	ROUGE-1	0.264	**0.265**
	ROUGE-L	**0.152**	**0.152**
	METEOR	**0.207**	0.206
Task 3: Stylized Feedback Generation	ROUGE-1	0.194	**0.205**
	ROUGE-L	0.128	**0.139**
	METEOR	0.201	**0.203**
Task 4: Multilingual Product Review Generation	ROUGE-1	0.190	**0.191**
	ROUGE-L	0.141	**0.142**
	METEOR	**0.174**	0.173

Table 13: Ablation study results showing the effect of retriever choice on PGraphRAG performance. Results are reported for *LLaMA-3.1-8B-Instruct* and *GPT-4o-mini* on the long-text generation task (Tasks 1-4).

Short Text Generation	Metric	Contriever	BM25
LLaMA-3.1-8B-Instruct			
Task 5: User Product Review Title Generation	ROUGE-1	0.122	**0.125**
	ROUGE-L	0.116	**0.119**
	METEOR	0.115	**0.117**
Task 6: Hotel Experience Summary Generation	ROUGE-1	0.117	**0.121**
	ROUGE-L	0.110	**0.113**
	METEOR	0.095	**0.099**
Task 7: Stylized Feedback Title Generation	ROUGE-1	0.125	**0.132**
	ROUGE-L	0.121	**0.128**
	METEOR	0.122	**0.129**
Task 8: Multi-lingual Product Review Title Generation	ROUGE-1	0.126	**0.131**
	ROUGE-L	0.118	**0.123**
	METEOR	0.112	**0.118**
GPT-4o-mini			
Task 5: User Product Review Title Generation	ROUGE-1	**0.113**	0.111
	ROUGE-L	**0.108**	0.106
	METEOR	**0.097**	**0.097**
Task 6: Hotel Experience Summary Generation	ROUGE-1	0.113	**0.118**
	ROUGE-L	0.107	**0.112**
	METEOR	0.080	**0.085**
Task 7: Stylized Feedback Title Generation	ROUGE-1	0.108	**0.109**
	ROUGE-L	0.106	**0.107**
	METEOR	0.094	**0.096**
Task 8: Multi-lingual Product Review Title Generation	ROUGE-1	**0.108**	**0.108**
	ROUGE-L	0.103	**0.104**
	METEOR	**0.082**	**0.082**

Table 14: Ablation study results showing the effect of retriever choice on PGraphRAG performance. Results are reported for *LLaMA-3.1-8B-Instruct* and *GPT-4o-mini* on the short-text generation task (Tasks 5-8).

1. PGraphRAG*: retrieves $k = 4$ randomly sampled entries from the profile without ranking.
2. PGraphRAG**: retrieves and includes all available context within the model's input limit (i.e., $k \to \infty$).

As expected, PGraphRAG** performs best due to its access to a larger and more diverse context. However, our focus is on the impact of removing ranking while keeping k fixed.

Removing ranking (PGraphRAG $\to$ PGraphRAG*) leads to a drop in ROUGE-1 of 2.29% for long-text generation and 3.18% for short-text tasks. The effect is also visible in user-only retrieval (PGraphRAG-U $\to$ PGraphRAG-U*), with decreases of 0.92% and 1.98% for long- and short-text tasks, respectively. These consistent declines underscore the importance of ranking in identifying relevant context.

While PGraphRAG** demonstrates the upper bound of performance, its scalability is limited due to cost and context length constraints. In contrast, ranked retrieval with a fixed k (as in PGraphRAG) offers a strong balance between performance and efficiency, making it more suitable for real-world deployment.

A.8 Evaluating Different GPT Variants

To compare the performance of different GPT variants, we evaluate PGraphRAG using a fixed retrieval configuration (BM25, $k = 4$) across two OpenAI models: GPT-4o-mini and GPT-o1. Among these, GPT-4o-mini demonstrated the best trade-off between accuracy, cost, and consistency on long-text generation tasks.

A.9 Impact of Length Constraints in GPT Model

In short-text generation tasks, controlling output length is essential to balance informativeness and conciseness. We evaluate the effect of fixed output constraints of 3, 5, and 10 words. Empirically, a 5-word constraint offers the best trade-off across evaluation metrics, yielding higher-quality outputs with minimal verbosity. We therefore adopt 5-word

Task	Metric	PGraphRAG	PGraphRAG*	PGraphRAG**	PGraphRAG-U	PGraphRAG-U*	PGraphRAG-U**
Long Text Generation							
Task 1: User-Product Review Generation	ROUGE-1	0.189	0.186	**0.191**	0.171	0.169	0.170
	ROUGE-L	**0.130**	0.125	**0.130**	0.117	0.114	0.117
	METEOR	0.196	0.188	**0.205**	0.176	0.173	0.180
Task 2: Hotel Experiences Generation	ROUGE-1	0.263	0.266	**0.267**	0.221	0.223	0.225
	ROUGE-L	0.152	0.152	**0.153**	0.135	0.134	0.135
	METEOR	0.206	0.209	**0.216**	0.164	0.168	0.171
Task 3: Stylized Feedback Generation	ROUGE-1	**0.211**	0.200	0.210	0.185	0.180	0.186
	ROUGE-L	**0.140**	0.133	0.136	0.123	0.122	0.123
	METEOR	0.202	0.206	**0.225**	0.183	0.184	0.189
Task 4: Multilingual Product Review Generation	ROUGE-1	0.194	0.188	**0.196**	0.168	0.167	0.171
	ROUGE-L	**0.144**	0.138	0.141	0.125	0.125	0.128
	METEOR	0.171	0.176	**0.188**	0.154	0.155	0.155
Short Text Generation							
Task 5: User Product Review Title Generation	ROUGE-1	0.115	0.114	**0.119**	0.108	0.108	0.111
	ROUGE-L	0.112	0.109	**0.114**	0.105	0.102	0.105
	METEOR	0.099	0.121	**0.128**	0.091	0.116	0.119
Task 6: Hotel Experience Summary Generation	ROUGE-1	0.116	0.117	**0.121**	0.108	**0.121**	0.119
	ROUGE-L	0.111	0.107	**0.112**	0.104	0.111	0.110
	METEOR	0.081	0.104	**0.109**	0.075	**0.109**	0.107
Task 7: Stylized Feedback Title Generation	ROUGE-1	**0.122**	0.111	0.120	0.113	0.115	0.114
	ROUGE-L	**0.118**	0.105	0.114	0.109	0.109	0.108
	METEOR	0.104	0.117	**0.126**	0.096	0.124	0.123
Task 8: Multi-lingual Product Review Title Generation	ROUGE-1	0.111	0.108	0.112	**0.115**	0.110	0.110
	ROUGE-L	0.105	0.100	0.104	**0.107**	0.103	0.101
	METEOR	0.083	0.101	0.105	0.088	**0.108**	0.107

Table 15: Zero-shot test set results for text generation using *GPT-4o-mini*. **PGraphRAG*** denotes retrieval of $k = 4$ randomly selected entries without ranking, while **PGraphRAG**** represents unbounded retrieval up to the model's context limit ($k \to \infty$).

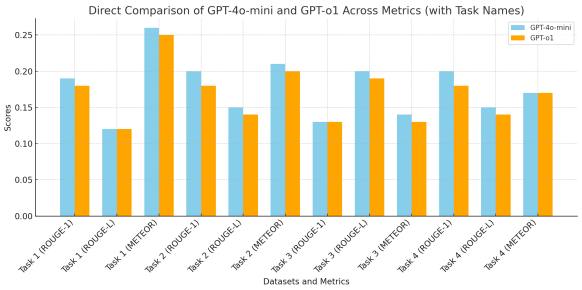

Figure 5: Comparison of *GPT-4o-mini* and *GPT-o1-preview* on the test set across Tasks 1–4 using BM25 retriever with $k = 4$.

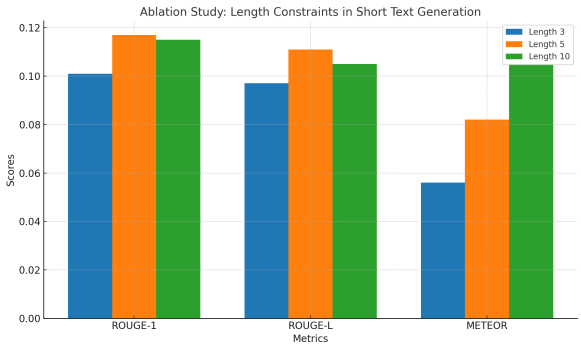

Figure 6: Effect of different output length constraints (3, 5, and 10 words) on short-text generation performance using PGraphRAG, measured on the validation set.

outputs as the default setting for all short-text generation experiments.

A.10 Validation Results

We conduct extensive validation experiments across all representative tasks, evaluating all combinations of language models, retrieval strategies, and top-k settings. The goal is to identify the most effective configuration for each task prior to test-time evaluation.

Results are reported in Tables 16, 17, and 18, corresponding to long-text generation, short-text generation, and ordinal classification tasks, respectively.

For each task, we select the best-performing configuration based on validation performance. These selected settings are then used in the test set evaluation. Notably, trends observed in the validation phase remain consistent in the test results, reinforcing the robustness of our setup.

B Related Work

Personalization in NLP

Personalization in natural language processing (NLP) focuses on tailoring responses to user-specific preferences, behaviors, and contexts, improving user experience and task performance. Early work in personalized generation relied on neural encoder-decoder models and incorporated attributes such as sentiment (Zang and Wan, 2017),

Long Text Generation	Metric	PGraphRAG	LaMP	No-retrieval	Random-retrieval
LLaMA-3.1-8B-Instruct					
Task 1: User-Product Review Generation	ROUGE-1	**0.173**	0.168	0.172	0.126
	ROUGE-L	0.124	**0.125**	0.121	0.095
	METEOR	0.150	0.134	**0.152**	0.101
Task 2: Hotel Experiences Generation	ROUGE-1	**0.263**	0.197	0.224	0.211
	ROUGE-L	**0.156**	0.128	0.141	0.130
	METEOR	**0.191**	0.121	0.148	0.147
Task 3: Stylized Feedback Generation	ROUGE-1	**0.226**	0.181	0.177	0.142
	ROUGE-L	**0.171**	0.134	0.125	0.104
	METEOR	**0.192**	0.147	0.168	0.119
Task 4: Multilingual Product Review Generation	ROUGE-1	**0.174**	0.174	0.173	0.146
	ROUGE-L	0.139	**0.141**	0.134	0.117
	METEOR	**0.133**	0.125	0.130	0.110
GPT-4o-mini					
Task 1: User-Product Review Generation	ROUGE-1	**0.186**	0.169	0.168	0.157
	ROUGE-L	**0.126**	0.114	0.113	0.112
	METEOR	**0.187**	0.170	0.173	0.148
Task 2: Hotel Experiences Generation	ROUGE-1	**0.265**	0.217	0.222	0.233
	ROUGE-L	**0.152**	0.132	0.133	0.138
	METEOR	**0.206**	0.161	0.164	0.164
Task 3: Stylized Feedback Generation	ROUGE-1	**0.205**	0.178	0.177	0.168
	ROUGE-L	**0.139**	0.121	0.119	0.117
	METEOR	**0.203**	0.178	0.184	0.160
Task 4: Multilingual Product Review Generation	ROUGE-1	**0.191**	0.164	0.167	0.171
	ROUGE-L	**0.142**	0.123	0.125	0.131
	METEOR	**0.173**	0.155	0.153	0.150

Table 16: Zero-shot Validation set results for long text generation using *LLaMA-3.1-8B-Instruct* and *GPT-4o-mini* on Tasks 1-4.

stylistic cues (Dong et al., 2017), and demographic metadata (Huang et al., 2014). To address data sparsity, approaches such as warm-start attention (Amplayo et al., 2018) and user embeddings were developed.

Recent efforts have expanded personalization using retrieval-augmented generation (RAG) strategies. Methods like in-context prompting (Lyu et al., 2024), retrieval-enhanced summarization (Richardson et al., 2023), and optimization via reinforcement learning or distillation (Salemi et al., 2024a) have improved output fluency and relevance. Benchmarking frameworks such as LaMP (Salemi et al., 2024b) and LongLaMP (Kumar et al., 2024) have standardized evaluation of personalized tasks (e.g., email writing, abstract generation). Meanwhile, retrieval-enhanced generation pipelines (Kim et al., 2020) improve long-form text by incorporating relevant user history.

However, most prior work assumes dense, high-coverage user history, limiting effectiveness in cold-start or sparse-profile scenarios. Few approaches leverage structured representations (e.g., knowledge graphs) to generalize beyond individual user traces. This gap highlights a need for models that can retrieve personalized yet diverse context using structured user-item relationships.

Knowledge Graphs and Retrieval-Augmented Generation (RAG)

Knowledge graphs (KGs) provide structured, relational context useful in a variety of NLP tasks such as question answering, entity linking, and reasoning (Liu et al., 2018; Schneider et al., 2022). By leveraging graph traversal and multi-hop paths, KGs enable precise contextualization in tasks that require reasoning over entity relationships (Salnikov et al., 2023). Recent techniques such as data synthesis and subgraph construction have improved KG scalability and coverage (Agarwal et al., 2021).

In parallel, retrieval-augmented generation (RAG) frameworks enhance LLMs by incorporating external memory or document retrieval into the generation process (Izacard and Grave, 2020).

Short Text Generation	Metric	PGraphRAG	LaMP	No-retrieval	Random-retrieval
LLaMA-3.1-8B-Instruct					
Task 5: User Product Review Title Generation	ROUGE-1	**0.125**	0.114	0.111	0.101
	ROUGE-L	**0.119**	0.108	0.105	0.095
	METEOR	**0.117**	0.111	0.104	0.094
Task 6: Hotel Experience Summary Generation	ROUGE-1	**0.121**	0.119	0.115	0.115
	ROUGE-L	**0.113**	0.111	0.108	0.107
	METEOR	**0.105**	**0.105**	0.100	0.094
Task 7: Stylized Feedback Title Generation	ROUGE-1	**0.132**	0.128	0.127	0.108
	ROUGE-L	**0.128**	0.124	0.122	0.104
	METEOR	**0.129**	0.124	0.118	0.103
Task 8: Multi-lingual Product Review Title Generation	ROUGE-1	**0.132**	0.128	0.108	0.127
	ROUGE-L	**0.128**	0.124	0.104	0.122
	METEOR	**0.129**	0.124	0.103	0.118
GPT-4o-mini					
Task 5: User Product Review Title Generation	ROUGE-1	**0.114**	0.106	0.109	0.107
	ROUGE-L	**0.107**	0.100	0.103	0.102
	METEOR	**0.119**	0.115	0.116	0.109
Task 6: Hotel Experience Summary Generation	ROUGE-1	**0.115**	**0.115**	0.114	0.112
	ROUGE-L	0.105	**0.106**	**0.106**	0.103
	METEOR	0.105	**0.106**	**0.106**	0.099
Task 7: Stylized Feedback Title Generation	ROUGE-1	**0.105**	0.101	**0.105**	0.098
	ROUGE-L	**0.102**	0.097	0.101	0.093
	METEOR	**0.118**	0.111	0.118	0.105
Task 8: Multi-lingual Product Review Title Generation	ROUGE-1	**0.108**	0.106	**0.108**	0.103
	ROUGE-L	0.099	0.098	**0.099**	0.095
	METEOR	0.101	0.102	**0.103**	0.095

Table 17: Zero-shot Validation set results for short text generation using *LLaMA-3.1-8B* and *GPT-4o-mini* on Tasks 5-8.

When integrated with KGs, RAG enables structured multi-hop reasoning (Saleh et al., 2024), rare entity recognition (Mathur et al., 2024), and hallucination reduction in generative outputs (Kang et al., 2023; Chen et al., 2023).

Despite these gains, scaling KGs in real-world systems (e.g., personalized recommendation) remains challenging (Ji et al., 2022). Graph construction, update, and refinement require sophisticated methods to ensure correctness and completeness (Paulheim, 2017). Moreover, traditional RAG pipelines using dense vector retrieval may struggle to integrate symbolic signals from structured graphs or handle noisy or misaligned data sources (Gao et al., 2024).

Toward Structured Personalization via Graph-Augmented RAG

The intersection of personalization, knowledge graphs, and RAG presents a promising research direction. Recent surveys (Zhang et al., 2024) emphasize the importance of personalization in LLMs but call for approaches that generalize across users with limited history and incorporate structured context. Our work addresses this by using user-centric bipartite graphs to retrieve not only user-authored content but also related interactions from similar users, enabling robust personalization under sparse conditions.

Unlike conventional user-history-based personalization, graph-augmented RAG offers a principled way to incorporate both individual and community signals—supporting generalization, diversity, and data efficiency at inference time.

C Limitations

While PGraphRAG demonstrates strong performance across personalized generation tasks, there are several considerations that present opportunities for future enhancement.

Scalability considerations. Although personalization approaches can raise scalability concerns, PGraphRAG is designed for efficient large-scale deployment. It constructs a unified, sparse user-item bipartite graph offline — i.e., graph construction is a one-time cost, similar to those used in scalable

Ordinal Classfication	Metric	PGraphRAG	LaMP	No-retrieval	Random-retrieval
LLaMA-3.1-8B-Instruct					
Task 9: User Product Review Ratings	MAE ↓	0.3272	0.3220	**0.3200**	0.3516
	RMSE ↓	0.7531	**0.7280**	0.7294	0.7972
Task 10: Hotel Experience Ratings	MAE ↓	0.3868	0.3685	**0.3614**	0.4008
	RMSE ↓	0.6989	0.6750	**0.6643**	0.7178
Task 11: Stylized Feedback Ratings	MAE ↓	**0.3356**	0.3368	0.3372	0.3812
	RMSE ↓	0.6856	0.6859	**0.6826**	0.7759
Task 12: Multi-lingual Product Ratings	MAE ↓	0.5228	**0.5216**	0.5282	0.5392
	RMSE ↓	0.8483	**0.8395**	0.8519	0.8704
GPT-4o-mini					
Task 9: User Product Review Ratings	MAE ↓	0.3652	0.3508	**0.3484**	0.4176
	RMSE ↓	0.7125	0.6943	**0.6925**	0.7792
Task 10: Hotel Experience Ratings	MAE ↓	**0.3308**	0.3472	0.3528	0.3640
	RMSE ↓	**0.6056**	0.6394	0.6475	0.6627
Task 11: Stylized Feedback Ratings	MAE ↓	**0.3340**	0.3364	0.3356	0.3972
	RMSE ↓	0.6515	0.6545	**0.6484**	0.7158
Task 12: Multi-lingual Product Ratings	MAE ↓	**0.4568**	0.4832	0.4908	0.4820
	RMSE ↓	**0.7414**	0.7808	0.7897	0.7917

Table 18: Performance comparison on rating prediction tasks (Tasks 9-12) using *GPT-4o-mini* and *LLaMA-3.1-8B-Instruct* on the validation set. Results are reported using MAE and RMSE metrics across retrieval methods.

recommender systems. As shown in Table 2, the graph is inherently sparse, enabling efficient storage and indexing. At inference time, rather than retrieving over the entire corpus as in traditional RAG settings, PGraphRAG scopes retrieval to a localized subgraph centered on the input user. This subgraph includes both the user's own interactions and those of neighboring users who share items. Standard retrievers (e.g., BM25 or Contriever) are then applied over this constrained set, significantly reducing search overhead while retaining personalized context. This design keeps runtime and memory usage low and supports scalable deployment across large user bases. In future work, we plan to explore compression techniques and real-time profile updates to further enhance scalability in dynamic environments.

Graph completeness and data sparsity. While the quality of retrieval can be influenced by the completeness of the user-centric graph, PGraphRAG is explicitly designed to operate under sparse and noisy conditions. Our benchmark includes users with minimal interaction history, yet results show strong performance across tasks compared to baseline methods. This robustness arises from PGraphRAG's graph-based retrieval strategy, which leverages neighboring nodes to provide relevant contextual signals even when direct user data is limited. Nonetheless, integrating implicit signals (e.g., click rate or engagement time) and developing more resilient retrieval methods for incomplete graphs remains a promising direction for future work.

Generalization vs. user adaptation. A core challenge lies in developing training strategies that balance individual personalization with generalization across user populations. While our approach augments prompts with structured context, future work may explore personalized fine-tuning or adapter layers to enhance this tradeoff further.

Static user profiles. Currently, user profiles are treated as static during evaluation. In real-world scenarios, preferences evolve over time. Extending the framework to model temporal dynamics and support profile updates is a promising direction for improving long-term personalization.