CoGenesis: A Framework Collaborating Large and Small Language Models for Secure Context-Aware Instruction Following

Kaiyan Zhang[1], **Jianyu Wang**[2], **Ermo Hua**[1], **Biqing Qi**[1,4]
Ning Ding[1,3], **Bowen Zhou**[1*]
[1] Tsinghua University, [2] Beijing Institute of Technology
[3] Frontis.AI, [4] Harbin Institute of Technology
zhang-ky22@mails.tsinghua.edu.cn, zhoubowen@tsinghua.edu.cn

Abstract

With the advancement of language models (LMs), their exposure to private data is increasingly inevitable, and their deployment (especially for smaller ones) on personal devices, such as PCs and smartphones, has become a prevailing trend. In contexts laden with user information, enabling models to both safeguard user privacy and execute commands efficiently emerges as an essential research imperative. In this paper, we propose CoGenesis, a collaborative generation framework integrating large (hosted on cloud infrastructure) and small models (deployed on local devices) to address privacy concerns logically. Initially, we design a pipeline to create personalized writing instruction datasets enriched with extensive context details as the testbed of this research issue. Subsequently, we introduce two variants of CoGenesis based on sketch and logits respectively. Our experimental findings, based on our synthesized dataset and two additional open-source datasets, indicate that: 1) Large-scale models perform well when provided with user context but struggle in the absence of such context. 2) While specialized smaller models fine-tuned on the synthetic dataset show promise, they still lag behind their larger counterparts. 3) Our CoGenesis framework, utilizing mixed-scale models, showcases competitive performance, providing a feasible solution to privacy issues.

1 Introduction

Large Language Models (LLMs)[1] have demonstrated significant potential in advancing artificial intelligence, exhibiting exceptional ability in instruction following and achieving superior performance in various tasks such as writing, coding, and other text-based activities (Achiam et al., 2023; Bubeck et al., 2023; Touvron et al., 2023a,b). LLM-based AI assistants play a crucial role in executing instructions, aiding in writing tasks, and accelerating work processes, thereby fostering content innovation (Zhang et al., 2023a; Haase and Hanel, 2023). LLMs often require extensive context information for generating more personalized and effective content, owing to their in-context learning abilities (Brown et al., 2020). Retrieval Augmented Generation (RAG) (Gao et al., 2023) has proven beneficial in incorporating additional context to enhance the informativeness and personalization of LLM outputs. However, current instruction-following tasks often fail to consider rich user context in their design (Wang et al., 2023c; Xu et al., 2023b; Ding et al., 2023). Incorporating personal

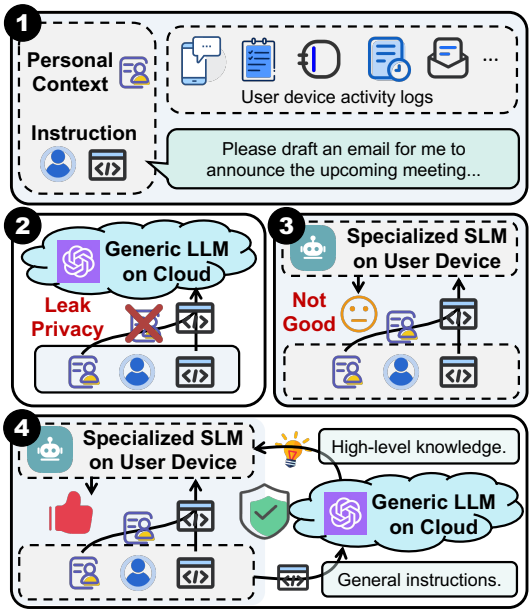

Figure 1: ❶ Context-aware instruction following example. ❷ LLMs excel with context but risk privacy. ❸ Specialized and smaller LMs (SLMs) on device are privacy-friendly but underperform. ❹ Collaborating LLMs and SLMs enhances privacy and performance.

[*] Corresponding author

[1]This paper defines large LMs (LLMs) as both closed and open-source models, designed for universal application and advanced performance, and intended for cloud deployment. Conversely, small LMs (SLMs) refer to models tailored for specific tasks and deployed on local devices.

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 4295–4312

experiences and activity logs could significantly augment the effectiveness of these instructions, called context-aware instruction following.

Despite their advancements, the most sophisticated LLMs, including GPT-4, Claude, and Gemini, are primarily commercialized and deployed on cloud services. This API-based deployment ensures the privacy of the LLMs but potentially compromises user privacy (Xiao et al., 2023; Zhang et al., 2023c). Although more powerful LLMs are being open-sourced, like Llama-2 (Touvron et al., 2023b), Qwen (Bai et al., 2023), and Mistral (Jiang et al., 2024), their stable deployment on local devices with limited resource remains challenging. Recent advancements in smaller LMs (SLMs) equipped with billions of parameters now enable their deployment on consumer desktops and smartphones, achieving satisfactory performance (Bai et al., 2023; Zhang et al., 2024; Singer et al., 2024; Hu et al., 2024). The balance between performance and privacy in LLMs and SLMs raises three critical questions: *(1) How effectively can LLMs operate without stringent user privacy contexts? (2) To what extent can specialized models, boasting billions of parameters, excel in context-aware instruction following? (3) Is it possible to navigate the trade-off between performance and privacy through collaborations between large and small models?*

As indicated in Figure 1, considering the following scenario in ❹: smaller, personalized LMs are deployed on user devices with limited resources. These SLMs can access private data and activity logs on the devices while processing instructions. In contrast, the more advanced general LLMs operate on cloud services and receive only general instructions. In this setup, the LLMs provide high-level knowledge like deeper planning, superior outlines, and even "dark knowledge". Meanwhile, the SLMs utilize the context information and knowledge provided by the LLMs to collaboratively generate personalized content. Current privacy protection methods for API-based services are limited (Cummings et al., 2023); they are either capable of handling only simple classification tasks with text sanitization (Kan et al., 2023; Chen et al., 2023b) or require encryption or noise addition (Zhou et al., 2022; Wu et al., 2023), which still pose a risk of data leakage. In contrast, the collaborative generation approach involving SLMs and LLMs can logically prevent privacy breaches without the need to upload context information.

Overall, we highlight our contributions as follows: (1) We introduce the context-aware instruction-following task that incorporates extensive user privacy context information. To support this, we design a four-step data construction process and synthesize a modest amount of instructional data for experimental validation. (2) We investigate the performance of LLMs and SLMs on this task. Our findings indicate that SLMs, when provided with context, can outperform LLMs lacking context but lag behind the performance of LLMs equipped with context. (3) For context-aware instruction generation, we present the CoGenesis framework. CoGenesis comprises sketch-based and logit-based variants to facilitate collaboration between large and small language models. This approach not only safeguards context privacy but also ensures performance gains.

2 Context-aware Instruction Following

2.1 Task Definition

Current instruction formats either consist of standalone instructions or instructions accompanied by additional inputs (Wang et al., 2023c). While these instructions typically cover generic tasks such as writing, searching, and coding, the inputs often contain specific task information, such as tables and coding problems. We classify the context-aware instruction-following task within the domain of controllable conditional text generation (Zhang et al., 2023b), enriching standard instructions with additional personal context. The task $\mathcal{T}$, focused on generating a response r relevant to user-specific data marked by privacy and personal style, is directed by instructions t and contextual information $\mathcal{C}$. It underscores the significance of contextual integration for enhancing output personalization and relevance, formally expressed as:

$$\mathcal{T}(t, \mathcal{C}) : r = g(t, \mathcal{C}; \theta) \tag{1}$$

This generative model g with parameters θ aims to adaptively respond to user instructions within the nuanced context of individual data attributes.

2.2 Data Construction

As illustrated on the left side of Figure 2, we delineate a novel four-step pipeline for crafting context-aware instructions aimed at generating personalized and creative text with AI assistants for diverse user groups. Our methodology begins with the

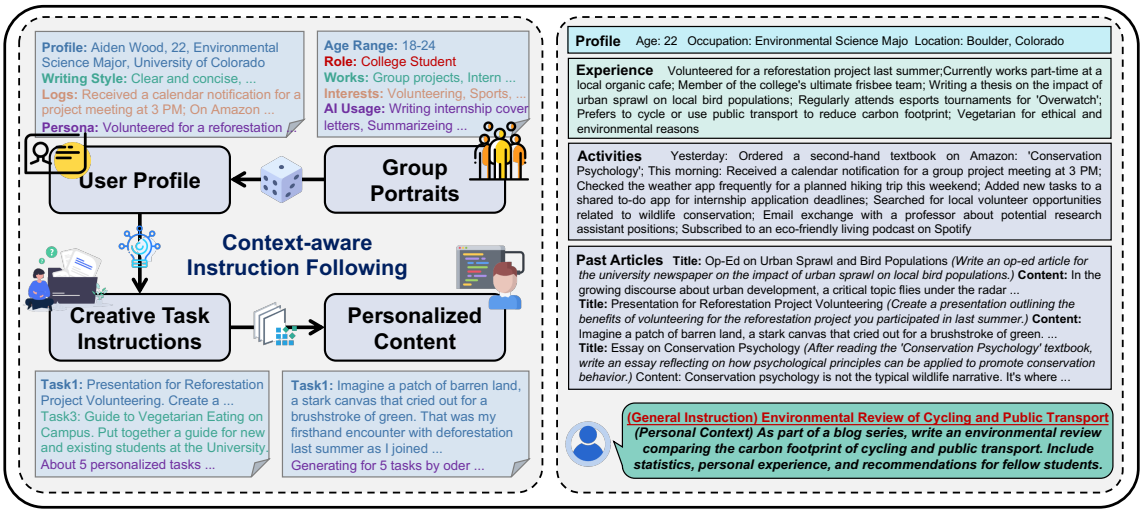

Figure 2: This illustration demonstrates construction process and example of context-aware instructions.

creation of detailed user group portraits, capturing demographics, professional backgrounds, and interests to identify specific AI application scenarios. Individual user profiles are then elaborated, incorporating unique writing styles, fictional personal details, and smart device usage to construct nuanced characters for AI writing tasks. These profiles inform the design of writing tasks that resonate with each character's lifestyle and digital interactions, ensuring task realism and relevance. Finally, we generate personalized texts that reflect the characters' professional and personal narratives with stylistic accuracy, demonstrating our approach's efficacy in producing coherent, context-specific content for AI-facilitated text generation.

An example is illustrated on the right side of Figure 2, where the user instruction comprises a general section and a personal section. The latter, in conjunction with the provided profile, experience, activities, and previous articles, constitutes the privacy-sensitive context information.

3 Collaborative Generation Framework

3.1 Overview of CoGenesis

We present the CoGenesis framework that capitalizes on the strengths of two differently scaled models: a LLM with parameters θ_l and a SLM with parameters θ_s. This framework is centered around the fusion strategy, denoted as $f(\cdot)$, which intelligently combines the outputs from both models. Specifically, θ_l generates replies solely based on the general instruction t, while θ_s considers both user instruction t and additional personal context $\mathcal{C}$ for its output generation. The fusion strategy $f(\cdot)$ aims to synergistically blend the outputs of $\theta_l(r|t)$ and $\theta_s(r|t,\mathcal{C})$. Intuitively, the combined performance is expected to not only surpass that of the individual models but also closely match the performance of θ_l had it processed both r and $\mathcal{C}$.

In our collaborative framework, sketches (or outlines) of content and next token logits from LLMs are considered forms of high-level knowledge. The two approaches to the function $f(\cdot)$ are identified as sketch-based and logit-based, respectively. The sketch-based approach is model-agnostic, whereas the logit-based method requires LLMs and SLMs to share the same tokenizer in our present configuration. In the following sections, we will detail these two implementations of $f(\cdot)$ sequentially.

3.2 Sketch-based CoGenesis

Recognizing the strengths of LLMs in planning and SLMs in crafting contextualized responses, we introduce a "sketch-then-fill" approach to synergize their capabilities for personalized content generation. As depicted on the left side of Figure 3, this approach consists of two crucial steps:

Step1: Sketch Generation by LLMs. Given the substantial cost and complexity, especially with API-dependent LLMs, we simplify the process by directly prompting LLMs with a general instruction t. The sketch r_{sketch} of content is derived through text decoding from the LLMs using a sampling strategy, succinctly represented as:

$$r_{\text{sketch}} = \text{Decoding}_{\text{LLM}}(t; \theta_l) \quad (2)$$

Step2: Content Personalization by SLMs. After acquiring the sketch r_{sketch}, the SLM utilizes it,

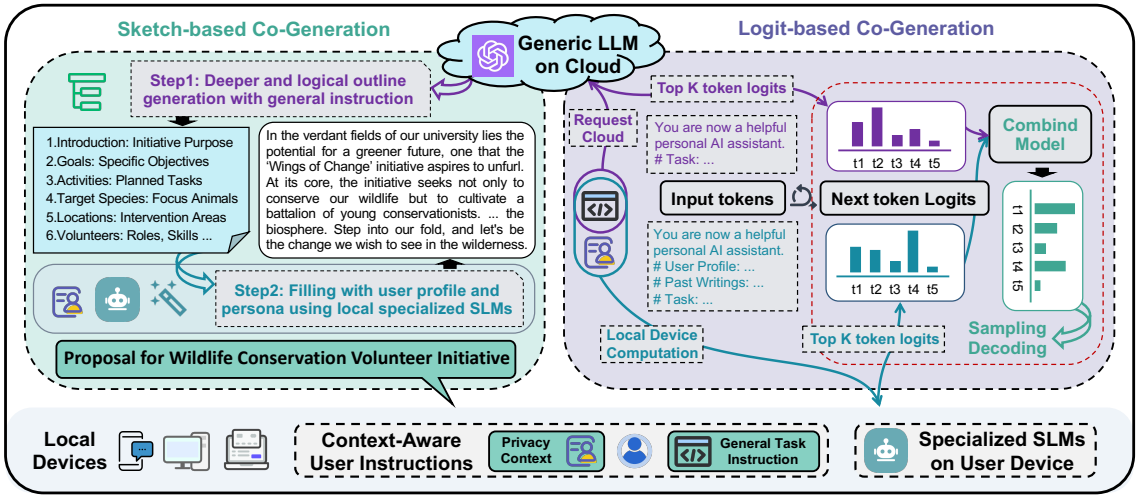

Figure 3: This figure demonstrates two collaborative generation variants of CoGenesis framework.

along with the initial instruction t and personal context $\mathcal{C}$, to tailor personalized content r. This phase focuses on fine-tuning the SLM's parameters θ_s to optimize content relevance and personalization:

$$\hat{r} = \arg\max_{r} P(r|t, \mathcal{C}, r_{\text{sketch}}; \theta_s) \quad (3)$$

In this formula, r denotes the final, customized content, and P is the likelihood of generating r given instruction t, context $\mathcal{C}$, and sketch r_{sketch} with parameters θ_s. This approach delineates the use of LLMs for foundational text sketching based on user prompts, without necessitating parameter adjustments, and SLMs for further content refinement. This ensures the final output aligns with user specifications and their interaction history, highlighting the distinct yet complementary roles of LLMs and SLMs in personalized content creation.

3.3 Logit-based CoGenesis

The logits produced in the final layer of language models encapsulate a wealth of information, reflecting the models' internal dark knowledge. Previous efforts, such as contrastive decoding (Li et al., 2022) and emulator tuning (Mitchell et al., 2023), have explored the synergistic use of logits from both LLMs and SLMs to diminish hallucinations (Sennrich et al., 2023), augment reasoning capabilities (O'Brien and Lewis, 2023), and streamline the fine-tuning process of LLMs (Liu et al., 2024). Motivated by these works, our logit-based strategy involves integrating the logits of LLMs and SLMs under different inputs, ensuring collaborative determination of the subsequent token. A notable aspect of our method is the differential context exposure for the models: SLMs access the full privacy context, while LLMs are provided with only broad instructions, as shown in Figure 3.

Defining the response sequence up to the kth token as $r_{<k}$, and denoting the kth token probabilities over vocabulary generated by LLMs and SLMs as p_k^l and p_k^s respectively, we leverage a lightweight combined model, denoted as CombModel, with parameters θ_c, to derive fusion weights w for final combined probabilities p_k^c. The computation of the logit-based method proceeds as follows:

$$p_k^s = \theta_s(r_{<k}, t, \mathcal{C}),\ p_k^l = \theta_l(r_{<k}, t) \quad (4)$$

$$w = \text{CombModel}(p_k^l, p_k^s) \quad (5)$$

$$p_k^c = w \cdot p_k^s + (1 - w) \cdot p_k^l \quad (6)$$

4 Experiments

4.1 Datasets

Synthetic Dataset. Following the construction process outlined in § 2.2, we synthesize a context-aware instruction-following dataset with GPT-4. Due to cost considerations, we construct a dataset representing thousands of fictitious users, based on hundreds of group portraits. After conducting quality and format filtering, we obtained a total of 1,500 users for training and validation. Additionally, we selected approximately 200 users from diverse group portraits to serve as the test set.

Open-source Datasets. In addition to our synthesized context-aware instruction datasets, we also utilize publicly accessible, personalized context writing datasets, although they are limited to specific tasks in domains such as email and academic

papers. Specifically, we employ the processed Avocado Research Email and Citation Network Papers datasets in LaMP (Salemi et al., 2023). Furthermore, we refine these datasets to facilitate the generation of email bodies and paper abstracts, considering previous emails and papers from the same users as contextual information.

Further details about the synthesized and processed datasets can be found in Appendix A.

4.2 Baselines

Settings. We primarily evaluate four configurations in our experiments: **1) *LLM with context.*** We engage LLMs with additional context to facilitate personalized generation. It's an *upper bound* that may compromise context privacy. **2) *LLM w/o context.*** Given the importance of privacy in contextual data, it is advisable to limit requests to cloud-based LLMs to general instructions only. This setting serves as a *lower bound* that preserves user privacy. **3) *SLM with context.*** This setting establishes the baseline for privacy-protected, on-device personalized generation. **4) *SLM + LLM with context.*** This is our proposed collaborative generation between large- and small-scale models. Within this framework, we evaluate sketch-based and logit-based methods.

Models. Our selection of LLMs encompasses both commercial API-based and open-source models. For the commercial segment, we concentrate on GPT-3.5-turbo and GPT-4-turbo. In the realm of open-source, we opt for the largest models within the Llama-2 (Touvron et al., 2023b), Qwen (72B in versions 1 and 1.5 [2]) (Bai et al., 2023), and Mistral (Jiang et al., 2024) series. Regarding SLMs, we prioritize the most recently released models with 1∼2 billions of parameters. This includes TinyLlama (Zhang et al., 2024), Qwen (1.8B in versions 1 and 1.5), StableLM [3], and H2O-Danube (Singer et al., 2024). For both LLMs and SLMs, our focus is on the chat versions, employing the default template for each model for consistency.

Additionally, zero-shot LLMs are used in both *with context* and *w/o context* settings. For *SLM with context*, we include both zero-shot and fine-tuned models, whereas only fine-tuned SLMs are utilized in *SLM + LLM with context*. Further information on prompts for LLMs and fine-tuning details for SLMs is presented in Appendix B.

[2] https://github.com/QwenLM/Qwen1.5

[3] https://huggingface.co/stabilityai

4.3 Evaluation Metrics

For instruction-following evaluation, LLM-based evaluators like GPT-4 have shown high consistency and effectiveness (Chang et al., 2023), particularly in the generation of personalized content, outperforming human evaluators in consistency (Wang et al., 2023b). Therefore, we employ GPT-4-turbo as a judge to assess generated content from multiple perspectives. This evaluation encompasses several criteria: personalization, alignment with the user profile, helpfulness, relevance, depth, creativity, and the level of detail, collectively contributing to the overall score (**Ovl.**). To further analyze the performance, we introduce two distinct prompts: one incorporating user context (**Ovl.(w)**) and the other excluding it (**Ovl.(w/o)**), enabling a comparative assessment of LLMs in both personalized content generation and broad instruction adherence. Additionally, the personalized scores of the responses are assessed independently, indicated as **Per.**

4.4 Main Results

As illustrated in Table 1, we analyze the experimental results considering the following aspects:

Results on LLMs. In the *with context* setting, `GPT-4-turbo` outperformed all other models across all metrics and datasets, followed by `Qwen-Chat(v1.5)` with the second-best performance. Similar outcomes were observed in the *w/o context* setting. However, a comparison between the two settings reveals that all LLMs exhibit diminished performance in terms of personalization and overall scores in the absence of context information, underscoring the value of context. Interestingly, while context greatly influences the scores in *with context* setting, it has a negligible effect on scores in *w/o context* setting, reflecting advanced capability of LLMs in adhering to general instructions. Moreover, when comparing our synthesized context-aware dataset and the email dataset to the paper abstract dataset, the latter demonstrates limited personalization factors resulting in high performance in *w/o context* setting.

Results on SLMs. Zero-shot SLMs exhibit highly varied performances, influenced by pre-training and supervised fine-tuning factors. Notably, `StableLM-Zephyr` achieves the highest performance, with `H2O-Danube-Chat` and `Qwen-Chat(v1.5)` closely competing for second place. Interestingly, with the advantage of context, zero-shot SLMs can outperform LLMs in scenarios

Model	Params	Context-aware Instructions			Avocado Emails			Academic Paper Abstracts		
		Ovl.(w)	Per.	Ovl.(w/o)	Ovl.(w)	Per.	Ovl.(w/o)	Ovl.(w)	Per.	Ovl.(w/o)
zero-shot LLM with context (upper bound)										
L1 GPT-4-turbo	N/A	**8.85**	**8.90**	**8.54**	**8.31**	**7.71**	**8.05**	**8.64**	**8.47**	**8.68**
L2 GPT-3.5-turbo	N/A	8.30	8.33	7.58	7.70	7.45	7.57	7.97	7.88	8.29
L3 Llama-2-Chat	70B	7.78	7.98	7.96	7.00	7.18	8.05	7.48	7.12	8.32
L4 Qwen-Chat(v1)	72B	8.38	8.38	8.14	7.62	7.10	7.70	7.90	7.62	8.24
L5 Qwen-Chat(v1.5)	72B	8.70	8.67	8.26	8.20	7.69	7.80	8.52	8.16	8.60
L6 Mixtral-8x7b	47B	8.12	8.22	7.96	7.35	6.88	6.92	7.92	7.62	8.08
zero-shot LLM w/o context (lower bound)										
L1 GPT-4-turbo	N/A	**6.10**	**6.46**	**8.75**	5.10	3.79	**8.58**	**8.42**	**7.94**	**8.73**
L2 GPT-3.5-turbo	N/A	4.34	3.76	7.47	3.72	3.03	7.61	3.37	3.59	6.04
L3 Llama-2-Chat	70B	4.74	4.60	8.12	3.38	3.52	8.04	6.74	6.18	7.90
L4 Qwen-Chat(v1)	72B	3.70	3.38	7.72	3.28	2.38	7.50	6.42	5.54	7.90
L5 Qwen-Chat(v1.5)	72B	5.86	5.98	8.52	**5.83**	**4.40**	8.54	7.92	7.16	8.28
L6 Mixtral-8x7b	47B	5.32	5.08	8.14	3.38	3.52	8.04	6.74	6.18	7.90
zero-shot SLM with context										
S1 StableLM-Zephyr	1.6B	**6.88**	6.82	6.68	**6.03**	**5.49**	6.51	**7.32**	**6.94**	**8.14**
S2 H2O-danube-chat	1.8B	6.56	**6.94**	6.60	4.77	5.00	**6.57**	5.98	5.54	6.94
S3 TinyLlama-Chat	1.1B	1.72	1.84	2.14	4.00	3.54	5.40	1.96	1.88	4.24
S4 Qwen-Chat(v1)	1.8B	5.78	5.50	6.00	4.91	4.54	6.49	4.46	5.04	6.74
S5 Qwen-Chat(v1.5)	1.8B	6.86	6.86	**7.20**	5.51	4.89	6.14	6.42	6.14	7.78
finetuned SLM (+ LLM) with context										
S1 StableLM-Zephyr	1.6B	8.30	8.56	7.73	7.58	6.70	7.20	7.96	7.64	8.18
+ *L1 sketch*	*mixed*	8.48$_{\uparrow 0.18}$	8.56$_{\uparrow 0.00}$	7.98$_{\uparrow 0.25}$	7.68$_{\uparrow 0.10}$	6.62$_{\downarrow 0.08}$	7.48$_{\uparrow 0.28}$	8.28$_{\uparrow 0.32}$	7.48$_{\downarrow 0.16}$	8.38$_{\uparrow 0.20}$
S2 H2O-danube-chat	1.8B	7.64	7.58	7.00	6.50	6.16	6.34	7.70	7.30	8.06
+ *L1 sketch*	*mixed*	7.84$_{\uparrow 0.20}$	7.78$_{\uparrow 0.20}$	7.14$_{\uparrow 0.14}$	7.14$_{\uparrow 0.64}$	6.72$_{\uparrow 0.56}$	7.52$_{\uparrow 1.18}$	8.10$_{\uparrow 0.40}$	7.28$_{\downarrow 0.02}$	8.18$_{\uparrow 0.12}$
S3 TinyLlama-Chat	1.1B	7.42	7.66	6.78	6.12	5.92	6.20	7.66	7.32	8.18
+ *L1 sketch*	*mixed*	7.66$_{\uparrow 0.24}$	7.14$_{\downarrow 0.52}$	6.82$_{\uparrow 0.04}$	6.58$_{\uparrow 0.46}$	6.02$_{\uparrow 0.10}$	6.60$_{\uparrow 0.40}$	7.72$_{\uparrow 0.06}$	7.36$_{\uparrow 0.04}$	8.10$_{\downarrow 0.08}$
+ *L3 logits*	*mixed*	7.76$_{\uparrow 0.34}$	7.74$_{\uparrow 0.08}$	7.06$_{\uparrow 0.28}$	6.06$_{\downarrow 0.06}$	6.16$_{\uparrow 0.24}$	6.94$_{\uparrow 0.74}$	8.14$_{\uparrow 0.48}$	7.34$_{\uparrow 0.02}$	8.04$_{\downarrow 0.14}$
S4 Qwen-Chat(v1)	1.8B	7.44	7.76	7.02	7.00	6.71	7.06	7.84	7.36	8.18
+ *L1 sketch*	*mixed*	7.80$_{\uparrow 0.36}$	7.82$_{\uparrow 0.06}$	7.64$_{\uparrow 0.62}$	7.18$_{\uparrow 0.18}$	6.44$_{\downarrow 0.27}$	7.28$_{\uparrow 0.22}$	8.02$_{\uparrow 0.18}$	7.70$_{\uparrow 0.34}$	8.34$_{\uparrow 0.16}$
+ *L4 logits*	*mixed*	8.12$_{\uparrow 0.68}$	8.20$_{\uparrow 0.44}$	7.86$_{\uparrow 0.84}$	7.48$_{\uparrow 0.48}$	6.44$_{\downarrow 0.27}$	7.46$_{\uparrow 0.40}$	7.92$_{\uparrow 0.08}$	7.16$_{\downarrow 0.20}$	8.30$_{\uparrow 0.12}$
S5 Qwen-Chat(v1.5)	1.8B	8.08	8.12	7.40	6.34	5.56	6.54	7.68	7.30	8.18
+ *L1 sketch*	*mixed*	8.18$_{\uparrow 0.10}$	7.98$_{\downarrow 0.14}$	7.62$_{\uparrow 0.22}$	6.54$_{\uparrow 0.20}$	5.84$_{\uparrow 0.28}$	6.74$_{\uparrow 0.20}$	8.10$_{\uparrow 0.42}$	7.24$_{\downarrow 0.06}$	8.32$_{\uparrow 0.14}$
+ *L5 logits*	*mixed*	8.28$_{\uparrow 0.20}$	8.22$_{\uparrow 0.10}$	7.80$_{\uparrow 0.40}$	6.66$_{\uparrow 0.32}$	5.70$_{\uparrow 0.14}$	7.12$_{\uparrow 0.58}$	8.14$_{\uparrow 0.46}$	7.28$_{\downarrow 0.02}$	8.40$_{\uparrow 0.22}$

Table 1: The table displays the performance of LLMs and SLMs in both *with context* and *w/o context* settings. We highlight the best result in **bold**, the second in underline and indicate variations in each SLM using ↑ and ↓.

lacking context. However, LLMs consistently outperform SLMs in overall scores without context, underscoring their superior capabilities in general instruction generation. After fine-tuning, SLMs surpass the performance of many LLMs with context, demonstrating the benefits of specialization. Yet, SLMs do not reach the performance levels of the most powerful LLMs, such as `GPT-4-turbo` and `Qwen-Chat(v1.5)`. These results highlight the promise of collaboration between specialized SLMs and LLMs for achieving better personalized scores, deeper writing, and enhanced instruction generalization.

Results on Mixed-Scale Models Collaborations. In our exploration of mixed-scale model collaboration, we utilize the sketch-based method for all SLMs and the logit-based method exclusively within the Llama and Qwen model families. This comparison reveals that collaborations between mixed-scale models achieve results comparable to those of LLMs alone, while also safeguarding privacy. Collaborative efforts generally enhance the overall scores of SLMs, both in evaluations with and without user context. Nevertheless, incorporating sketch-based collaboration, in particular, might slightly detract from personalization scores due to the reliance on LLMs that inference without context. Between the logit-based and sketch-based approaches, the former proves more efficacious, contingent upon the SLMs and LLMs utilizing a common tokenizer. Overall, this collaborative strategy between mixed-scale models offers a promising avenue for balancing efficiency and privacy considerations, though it still necessitates further refinements to optimize performance.

4.5 Ablation Study

4.5.1 Sketch-based CoGenesis

Sketch vs. Full Content. Figure 4 contrasts the efficacy of employing merely the sketch versus the entire content provided by LLMs. The findings indicate that incorporating full content generally

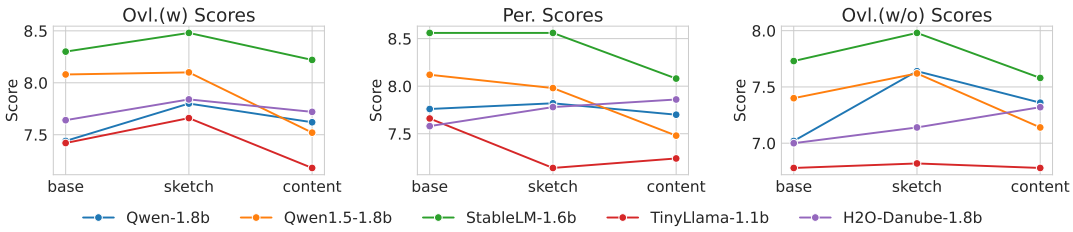

Figure 4: Comparative Performance of SLMs Utilizing Sketch versus Full Content.

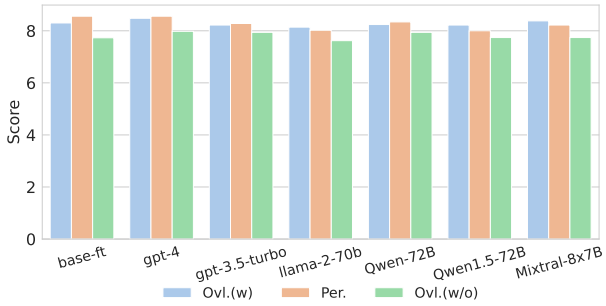

Figure 5: Performance of StableLM Using Sketches Generated by Various Models.

Models \ Metric	Ovl.(w)	Per.	Ovl.(w/o)
Qwen-72B-Chat (*with context*)	**8.38**	**8.38**	**8.14**
Qwen-72B-Chat (*w/o context*)	3.70	3.38	7.72
Qwen-1.8B-Chat (*finetuned*)	7.44	7.76	7.02
Mean Pooling Fusing	7.76	7.84	7.42
Max Pooling Fusing	7.90	7.94	7.52
Learnable Weights Fusing	**8.12**	**8.20**	**7.86**

Table 2: Comparison of Mean and Max Fusion Strategies Against our Learnable Fusion Model.

Model and Setting	Win/Tie/Lose (%)	BLEU	ROUGE-L
SLM finetuned	-/50/-	2.07	13.95
LLM w/ context	38/2/10	2.61	14.66
LLM w/o context	3/0/47	1.51	13.54
Sketch-based CoGen	27/3/20	1.81	12.98
Logits-based CoGen	32/5/13	2.30	14.18

Table 3: Human Assessments and Automated Evaluation Results (Qwen-72B/1.8B as LLM/SLM)

detracts from model performance, in contrast to utilizing no content or only the sketch. This discrepancy can be attributed to the potential overload of redundant information. As explored in (Weston and Sukhbaatar, 2023), an excess of content does not invariably enhance performance and risks the inclusion of extraneous details.

Generalization Capabilities of Sketch. For small models fine-tuned with sketches generated by GPT-4, we explore their generalization potential by utilizing sketches from a variety of LLMs during testing. Specifically, we focus on `StableLM-Zephyr` for detailed ablation analysis. Figure 5 demonstrates that employing sketches from alternative LLMs marginally affects the overall and personalized scores negatively but enhances the overall score in scenarios lacking context, relative to `GPT-4`. This suggests that sketches generated by different models vary and exhibit limited generalization capabilities.

4.5.2 Logit-based CoGenesis

Logits Fusing Strategy. We implement a learnable model designed for merging logits from context-free LLMs and context-inclusive SLMs in § 3.3. Additionally, we explore straightforward max and mean pooling strategies, acknowledged as robust baselines in (Ormazabal et al., 2023). According to Table 2, both max and mean pooling methods enhance the performance of SLMs, with max pooling proving to be superior. Nonetheless, given the distinct input conditions for LLMs and SLMs, a learnable fusion model becomes essential in context-aware environments, outperforming simple pooling techniques significantly.

Logits of LLMs and SLMs. Within the fusion model, we integrate logits from LLMs and SLMs using self-adjusting weights. Figure 6 showcases the output generated by both `Qwen-72B` and `Qwen-1.8B`, employing color-coded weights to delineate their respective contributions: red signifies SLMs, blue denotes LLMs, and white represents an evenly balanced weight of 0.5. It is observed that LLMs predominantly influence the sketch of the generated content, whereas SLMs play a more significant role across the majority of tokens, underscoring the importance of collaboration.

4.6 Human Evaluation

Evaluations by LLMs have not yet been demonstrated to be infallible; consequently, we conducted human evaluations. To streamline the complexity of this assessment, annotators were instructed

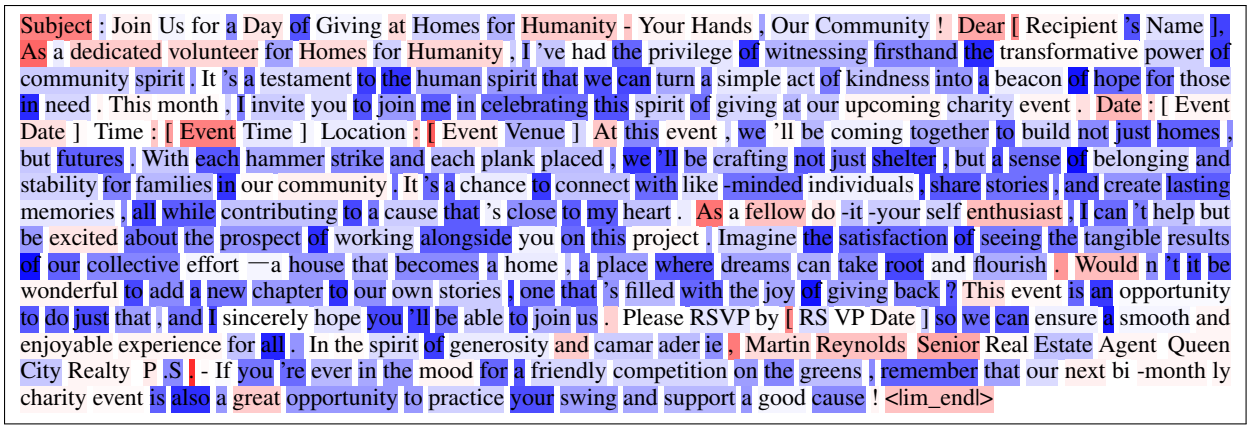

Figure 6: Visualization of LLM and SLM Logits Weights per Token During Generation. Blue signifies SLM contributions, red indicates LLM contributions, with darker shades representing higher weights.

to compare outcomes under various settings to those of the finetuned SLM, recording results as win/tie/lose, mirroring the methodology utilized in lmsys/chatbot-arena-leaderboard [4]. Additionally, we employed traditional word overlap metrics such as BLEU and ROUGE-L, calculated using the evaluate library[5].

As shown in Table 3, the results of human evaluation are consistent with GPT-4 evaluations, where CoGenesis performs better than finetuned SLMs and LLMs without context separately, and performs closely to LLMs with context.

5 Discussion

Although our experiments have been limited to models of the same family using identical tokenizers, these methods could potentially be expanded through a tokenizer alignment strategy (Fu et al., 2023; Wan et al., 2024). This principle aligns with other logits-based decoding techniques such as speculative decoding, contrastive decoding. By aligning the tokens and probabilities of models with different tokenizers, it is feasible to facilitate knowledge transfer across various LLMs and SLMs.

Recent studies have demonstrated the potential to reconstruct prompts based on the distribution of next token tokens (Morris et al., 2023, 2024). However, the accuracy of extraction, particularly the exact match scores, remains discouragingly low. Furthermore, since only the top-k logits for each token are utilized in our experimental, the cost of reconstruction is prohibitively high. Therefore, logits-based collaboration remains sufficiently secure, and can be further enhanced with the implementation of encryption and noise addition algorithms. As shown in Table 4, we have investigated the logit-based CoGenesis approach, which enhances privacy by uploading only the previous few tokens instead of the entire response. This approach, inspired by the principle that a good start leads to effective completion (Jain et al., 2024; Wang and Zhou, 2024), suggests that initial guidance from LLMs on the first few tokens can direct SLMs to independently generate the remainder of the response. Performance data indicate improvements when transferring just 8, 16, or 32 tokens to cloud LLMs, compared to a fine-tuned SLM alone, with the entire response typically exceeding 500 tokens. Increasing the number of transferred tokens boosts the LLM + SLM performance, allowing us to balance enhanced performance against reduced privacy risks.

Settings	Ovl.(w/)	Per.	Ovl.(w/o)
LLM w/ context	8.38	8.38	8.14
LLM w/o context	3.70	3.38	7.72
FT SLM (rk=0 toks)	7.44	7.76	7.02
LLM+SLM (rk=8 toks)	7.72	7.96	7.38
LLM+SLM (rk=16 toks)	7.78	7.84	7.22
LLM+SLM (rk=32 toks)	7.94	7.98	7.30
LLM+SLM (rk=64 toks)	7.94	8.04	7.54
LLM+SLM (rk=128 toks)	7.98	8.06	7.44
Logit-based CoGenesis	8.12	8.20	7.86

Table 4: Comparative Analysis of First Token Quantities used in Logit-based CoGenesis.

6 Related Works

The advent of LLMs has revolutionized the field of instruction following, with models being trained

[4] https://huggingface.co/spaces/lmsys/chatbot-arena-leaderboard

[5] https://github.com/huggingface/evaluate

on diverse and complex instruction sets, enabling them to perform a wide array of tasks from creative writing (Franceschelli and Musolesi, 2023) to coding (Qian et al., 2023) and debugging (Jimenez et al., 2023). The push towards collecting high-quality instruction data (Wang et al., 2023c; Ding et al., 2023; Xu et al., 2023a) has allowed for the development of both proprietary and open-source medium-scale language models adept at following instructions. However, the reliance on cloud-based proprietary models like ChatGPT and GPT-4 for instruction execution raises significant privacy concerns due to the potential risks associated with uploading sensitive data (Achiam et al., 2023; Team et al., 2023; Liu et al., 2023). To mitigate these risks, various privacy-preserving techniques have been employed, albeit with limitations in completely securing user data (Cummings et al., 2023; Kan et al., 2023; Chen et al., 2023b; Wu et al., 2023). Furthermore, there is a notable gap in instruction datasets, particularly in the inclusion of contextual information (Salemi et al., 2023; Wang et al., 2023b), highlighting an area for further exploration in context-aware instruction formulation to enhance privacy and personalization.

In parallel, the exploration of mixed-scale model collaboration emerges as a promising avenue to address the scalability, efficiency (Xia et al., 2024), and privacy (Yao et al., 2023) challenges inherent to LLMs. While larger models benefit from increased capabilities, their high inference costs and privacy concerns contrast with the lower costs and greater accessibility of smaller models (Bai et al., 2023; Gunasekar et al., 2023; Zhang et al., 2024; Grangier et al., 2024; Singer et al., 2024). Research in this domain is bifurcated into collaborative training and inference strategies, including offsite-tuning (Xiao et al., 2023) and speculative decoding (Leviathan et al., 2023), aiming to leverage the strengths of both large and small models (Mitchell et al., 2023; Liu et al., 2024). This paper specifically investigates collaborative inference methods, including sketch-based and logit-based approaches, to enhance the efficiency and privacy of LLMs, suggesting a promising direction in utilizing mixed-scale models for instruction following tasks.

7 Conclusion

This paper investigates context-aware instruction following, enriching prompts with detailed user privacy information. We outline a pipeline for creating context-aware instructions. To mitigate privacy issues, we introduce CoGenesis, a collaborative framework between SLMs and LLMs utilizing sketches and logits. Our results highlight the advantages of mixed-scale model collaboration, suggesting fruitful directions for future research.

Limitations

This study explores context-aware instruction following, introducing strategies for collaboration between large and small models to address privacy concerns. We developed a synthetic dataset for context-aware instruction following to empirically test our approaches. Our findings suggest that this model collaboration can significantly mitigate privacy risks associated with using public API-based LLMs. However, our dataset is limited in size and was specifically crafted for preliminary validation. Future work will focus on expanding this dataset to enhance its quality, realism, and diversity. Additionally, our proposed methods, particularly the logits-based approach, are currently restricted to models sharing the same tokenizer. Further research on tokenizer alignment is necessary to broaden the applicability of our strategies.

Ethics Statement

The advent of LLMs has underscored the urgent need to address privacy and security concerns within the realm of artificial intelligence. This paper concentrates on the privacy challenges posed by context-aware instruction-following applications of LLMs, proposing methods to mitigate these concerns without compromising the models' effectiveness. We emphasize the ethical imperative of protecting user data, adopting a strategy that involves generating synthetic datasets using GPT-4 for training and testing. This approach ensures that our research does not compromise real-world user privacy by preventing any potential data leakage. In essence, our work not only seeks to advance the technological capabilities of LLMs but also to uphold the highest standards of ethical responsibility by safeguarding user privacy through innovative and secure data handling practices.

Acknowledgements

This work is supported by the National Key R&D Program of China (No. 2022ZD0119101). We extend our gratitude to the anonymous reviewers for their insightful feedback.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. arXiv preprint arXiv:2303.08774.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. arXiv preprint arXiv:2309.16609.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. Advances in neural information processing systems, 33:1877–1901.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. arXiv preprint arXiv:2303.12712.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2023. A survey on evaluation of large language models. ACM Transactions on Intelligent Systems and Technology.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, et al. 2023a. Alpagasus: Training a better alpaca with fewer data. arXiv preprint arXiv:2307.08701.

Sai Chen, Fengran Mo, Yanhao Wang, Cen Chen, Jian-Yun Nie, Chengyu Wang, and Jamie Cui. 2023b. A customized text sanitization mechanism with differential privacy. In Findings of the Association for Computational Linguistics: ACL 2023, pages 5747–5758.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. 2023. Ultrafeedback: Boosting language models with high-quality feedback. arXiv preprint arXiv:2310.01377.

Rachel Cummings, Damien Desfontaines, David Evans, Roxana Geambasu, Matthew Jagielski, Yangsibo Huang, Peter Kairouz, Gautam Kamath, Sewoong Oh, Olga Ohrimenko, et al. 2023. Challenges towards the next frontier in privacy. arXiv preprint arXiv:2304.06929.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 3029–3051, Singapore. Association for Computational Linguistics.

Giorgio Franceschelli and Mirco Musolesi. 2023. On the creativity of large language models. arXiv preprint arXiv:2304.00008.

Yao Fu, Hao Peng, Litu Ou, Ashish Sabharwal, and Tushar Khot. 2023. Specializing smaller language models towards multi-step reasoning. In Proceedings of the 40th International Conference on Machine Learning, volume 202 of Proceedings of Machine Learning Research, pages 10421–10430. PMLR.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. arXiv preprint arXiv:2312.10997.

David Grangier, Angelos Katharopoulos, Pierre Ablin, and Awni Hannun. 2024. Specialized language models with cheap inference from limited domain data. arXiv preprint arXiv:2402.01093.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, et al. 2023. Textbooks are all you need. arXiv preprint arXiv:2306.11644.

Jennifer Haase and Paul HP Hanel. 2023. Artificial muses: Generative artificial intelligence chatbots have risen to human-level creativity. arXiv preprint arXiv:2303.12003.

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, Xinrong Zhang, Zheng Leng Thai, Kaihuo Zhang, Chongyi Wang, Yuan Yao, Chenyang Zhao, Jie Zhou, Jie Cai, Zhongwu Zhai, Ning Ding, Chao Jia, Guoyang Zeng, Dahai Li, Zhiyuan Liu, and Maosong Sun. 2024. Minicpm: Unveiling the potential of small language models with scalable training strategies.

Kushal Jain, Niket Tandon, and Kumar Shridhar. 2024. Well begun is half done: Importance of starting right in multi-step math reasoning.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. arXiv preprint arXiv:2310.06825.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. arXiv preprint arXiv:2401.04088.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik

Narasimhan. 2023. Swe-bench: Can language models resolve real-world github issues? arXiv preprint arXiv:2310.06770.

Zhigang Kan, Linbo Qiao, Hao Yu, Liwen Peng, Yifu Gao, and Dongsheng Li. 2023. Protecting user privacy in remote conversational systems: A privacy-preserving framework based on text sanitization. arXiv preprint arXiv:2306.08223.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. arXiv preprint arXiv:2001.08361.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2023. Fast inference from transformers via speculative decoding. In International Conference on Machine Learning, pages 19274–19286. PMLR.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2022. Contrastive decoding: Open-ended text generation as optimization. arXiv preprint arXiv:2210.15097.

Alisa Liu, Xiaochuang Han, Yizhong Wang, Yulia Tsvetkov, Yejin Choi, and Noah A Smith. 2024. Tuning language models by proxy. arXiv preprint arXiv:2401.08565.

Yang Liu, Yuanshun Yao, Jean-Francois Ton, Xiaoying Zhang, Ruocheng Guo Hao Cheng, Yegor Klochkov, Muhammad Faaiz Taufiq, and Hang Li. 2023. Trustworthy llms: a survey and guideline for evaluating large language models' alignment. arXiv preprint arXiv:2308.05374.

Eric Mitchell, Rafael Rafailov, Archit Sharma, Chelsea Finn, and Christopher D Manning. 2023. An emulator for fine-tuning large language models using small language models. arXiv preprint arXiv:2310.12962.

MLC team. 2023. MLC-LLM.

John Xavier Morris, Volodymyr Kuleshov, Vitaly Shmatikov, and Alexander M Rush. 2023. Text embeddings reveal (almost) as much as text. In The 2023 Conference on Empirical Methods in Natural Language Processing.

John Xavier Morris, Wenting Zhao, Justin T Chiu, Vitaly Shmatikov, and Alexander M Rush. 2024. Language model inversion. In The Twelfth International Conference on Learning Representations.

Sean O'Brien and Mike Lewis. 2023. Contrastive decoding improves reasoning in large language models. arXiv preprint arXiv:2309.09117.

Aitor Ormazabal, Mikel Artetxe, and Eneko Agirre. 2023. CombLM: Adapting black-box language models through small fine-tuned models. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 2961–2974, Singapore. Association for Computational Linguistics.

Ajay Patel, Delip Rao, Ansh Kothary, Kathleen McKeown, and Chris Callison-Burch. 2023. Learning interpretable style embeddings via prompting LLMs. In Findings of the Association for Computational Linguistics: EMNLP 2023, pages 15270–15290, Singapore. Association for Computational Linguistics.

Chen Qian, Xin Cong, Cheng Yang, Weize Chen, Yusheng Su, Juyuan Xu, Zhiyuan Liu, and Maosong Sun. 2023. Communicative agents for software development. arXiv preprint arXiv:2307.07924.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. arXiv preprint arXiv:2305.18290.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. 2023. Lamp: When large language models meet personalization. arXiv preprint arXiv:2304.11406.

Rico Sennrich, Jannis Vamvas, and Alireza Mohammadshahi. 2023. Mitigating hallucinations and off-target machine translation with source-contrastive and language-contrastive decoding. arXiv preprint arXiv:2309.07098.

Philipp Singer, Pascal Pfeiffer, Yauhen Babakhin, Maximilian Jeblick, Nischay Dhankhar, Gabor Fodor, and Sri Satish Ambati. 2024. H2o-danube-1.8 b technical report. arXiv preprint arXiv:2401.16818.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. arXiv preprint arXiv:2312.11805.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, et al. 2023. Zephyr: Direct distillation of lm alignment. arXiv preprint arXiv:2310.16944.

Fanqi Wan, Xinting Huang, Deng Cai, Xiaojun Quan, Wei Bi, and Shuming Shi. 2024. Knowledge fusion of large language models.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2023a. Improving text embeddings with large language models. arXiv preprint arXiv:2401.00368.

Xuezhi Wang and Denny Zhou. 2024. Chain-of-thought reasoning without prompting.

Yaqing Wang, Jiepu Jiang, Mingyang Zhang, Cheng Li, Yi Liang, Qiaozhu Mei, and Michael Bendersky. 2023b. Automated evaluation of personalized text generation using large language models. arXiv preprint arXiv:2310.11593.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023c. Self-instruct: Aligning language models with self-generated instructions. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Anna Wegmann, Marijn Schraagen, and Dong Nguyen. 2022. Same author or just same topic? towards content-independent style representations. In Proceedings of the 7th Workshop on Representation Learning for NLP, pages 249–268, Dublin, Ireland. Association for Computational Linguistics.

Lai Wei, Zihao Jiang, Weiran Huang, and Lichao Sun. 2023. Instructiongpt-4: A 200-instruction paradigm for fine-tuning minigpt-4. arXiv preprint arXiv:2308.12067.

Jason Weston and Sainbayar Sukhbaatar. 2023. System 2 attention (is something you might need too). arXiv preprint arXiv:2311.11829.

Tong Wu, Ashwinee Panda, Jiachen T Wang, and Prateek Mittal. 2023. Privacy-preserving in-context learning for large language models. arXiv e-prints, pages arXiv–2305.

Heming Xia, Zhe Yang, Qingxiu Dong, Peiyi Wang, Yongqi Li, Tao Ge, Tianyu Liu, Wenjie Li, and Zhifang Sui. 2024. Unlocking efficiency in large language model inference: A comprehensive survey of speculative decoding. arXiv preprint arXiv:2401.07851.

Guangxuan Xiao, Ji Lin, and Song Han. 2023. Offsite-tuning: Transfer learning without full model. arXiv preprint arXiv:2302.04870.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023a. Wizardlm: Empowering large language models to follow complex instructions. arXiv preprint arXiv:2304.12244.

Canwen Xu, Daya Guo, Nan Duan, and Julian McAuley. 2023b. Baize: An open-source chat model with parameter-efficient tuning on self-chat data. arXiv preprint arXiv:2304.01196.

Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Eric Sun, and Yue Zhang. 2023. A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. arXiv preprint arXiv:2312.02003.

Chaoning Zhang, Chenshuang Zhang, Sheng Zheng, Yu Qiao, Chenghao Li, Mengchun Zhang, Sumit Kumar Dam, Chu Myaet Thwal, Ye Lin Tun, Le Luang Huy, et al. 2023a. A complete survey on generative ai (aigc): Is chatgpt from gpt-4 to gpt-5 all you need? arXiv preprint arXiv:2303.11717.

Hanqing Zhang, Haolin Song, Shaoyu Li, Ming Zhou, and Dawei Song. 2023b. A survey of controllable text generation using transformer-based pre-trained language models. ACM Computing Surveys, 56(3):1–37.

Kaiyan Zhang, Ning Ding, Biqing Qi, Xuekai Zhu, Xinwei Long, and Bowen Zhou. 2023c. CRaSh: Clustering, removing, and sharing enhance fine-tuning without full large language model. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 9612–9637, Singapore. Association for Computational Linguistics.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024. Tinyllama: An open-source small language model. arXiv preprint arXiv:2401.02385.

Xin Zhou, Jinzhu Lu, Tao Gui, Ruotian Ma, Zichu Fei, Yuran Wang, Yong Ding, Yibo Cheung, Qi Zhang, and Xuanjing Huang. 2022. TextFusion: Privacy-preserving pre-trained model inference via token fusion. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 8360–8371, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Xin Zhou, Yi Lu, Ruotian Ma, Tao Gui, Yuran Wang, Yong Ding, Yibo Zhang, Qi Zhang, and Xuanjing Huang. 2023. TextObfuscator: Making pre-trained language model a privacy protector via obfuscating word representations. In Findings of the Association for Computational Linguistics: ACL 2023, pages 5459–5473, Toronto, Canada. Association for Computational Linguistics.

A Dataset Details

A.1 Context-Aware Instructions

The four steps for constructing context-aware instructions are as follows.

Group Portraits. We begin by constructing highly diverse user group portraits from the real world, encompassing a wide range of groups such as college students, programmers, and various other professions. For each group, we define their age range, identify their professional field or occupation, and enumerate typical activities and hobbies to capture the group's unique interests. Additionally, we delineate specific scenarios in which these groups might utilize an AI assistant for personalized and creative text generation in both their professional and personal lives. Examples of such use cases include drafting business emails, writing creative blogs, composing academic papers, and crafting extended tweets.

User Profile. Building upon our diverse user group portraits, we next develop individual user profiles with rich detail. Emphasizing consistency and realism, our process involves four steps: 1) Personal Writing Style: Tailoring language use and expression unique to each character. 2) Private Information: Creating 5-10 fictional details for each profile, including life events and technology interactions. 3) Smart Device Usage: Generating 5-10 fictional activity logs per profile, covering messages, purchases, schedules, and more. Our aim is to shape distinct, multi-dimensional characters for a variety of AI writing applications.

Writing Task Instructions. In this phase, we craft text creation tasks tailored to our user characters, ensuring alignment with their professions, hobbies, and lifestyles. These tasks are intricately linked to their mobile phone activity logs and personal details, weaving the characters' experiences, social media activities, and AI assistant interactions into the narratives. For each character, we develop K tasks that are both realistic and contextually relevant.

Personalized Generations. Here, we produce personalized texts adhering to specific guidelines. Our focus is on crafting authentic and coherent narratives that vividly reflect each character's professional and personal life. By adapting to the user's unique writing style, we aim to create personalized and stylistically distinctive content. To ensure coherence and relevance across various tasks, content for each task is generated sequentially.

Following these four steps, we utilize GPT-4 to create instructions with user context and responses for subsequent experiments.

A.2 LaMP Dataset Processing

To minimize the personalization gap between the target content and user profiles in the Avocado Research Email and Citation Network Papers datasets in LaMP [6], we have implemented a two-step processing approach. Initially, we retrieve content that most closely aligns with the user profile using `intfloat/e5-mistral-7b-instruct` embeddings (Wang et al., 2023a). Subsequently, we encode the target content and the entire profile using style embeddings (Wegmann et al., 2022; Patel et al., 2023) , selecting only the most personalized samples. Furthermore, the content lengths and profiles of samples in LaMP vary significantly, ranging from 10 to 100,000 tokens [7], exceeding the context length capabilities of contemporary models. Consequently, we have selected samples with a minimum length of 128 characters for paper abstracts and 64 for email bodies, and a maximum length of 1024 characters for both. Owing to the unavailability of a test dataset in LaMP, we repurposed the dev dataset as our test set. Additionally, we divided the filtered training data into actual training and validation datasets, using a 9:1 split ratio.

A.3 Dataset Statistics

The statistical results of the final processed dataset are presented in Table 1.

Dataset	Context-aware	Avocado Emails	Paper Abstracts
Total Users	1736	N/A	N/A
Avg Profile Length	1182	618	1000
Output Length	155	178	158
Train Samples	1346	1,137	1,448
Dev Samples	150	127	161
Test Samples	240	346	357

Table 5: The table presents the statistics of constructed dataset and public datasets.

Our synthetic instruction datasets include a variety of instruction types, such as preparing a speech, designing a plan, and more. All of these tasks require models to utilize personal context information, including previous activities and schedules. To illustrate, we display the top 10 most common root verbs and top 10 direct noun objects in our constructed datasets as shown in Figure 6

[6] https://lamp-benchmark.github.io/download

[7] https://github.com/openai/tiktoken/

Verb	Percent (%)	Verb	Percent (%)
write	19.4	post	24.0
draft	16.0	article	13.8
compose	13.9	speech	12.5
create	11.5	proposal	10.2
develop	8.7	guide	7.2
prepare	8.2	series	4.4
craft	6.3	piece	4.7
curate	4.5	plan	3.2
design	2.1	outline	3.0
script	2.0	email	2.8

Table 6: The top 10 most common root verbs and top 10 direct noun objects in our constructed datasets.

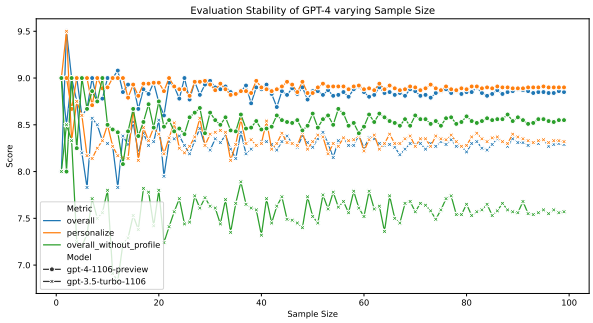

Figure 7: This illustration demonstrates evaluation consistency and stability of GPT-4 as a judge.

B Model Details

For all open-source Large Language Models, we utilize vLLM [8] for efficient inference, setting the temperature to 0.7, top-p to 0.9, and the maximum number of new tokens to 1024.

Regarding the specialization of Small Language Models (SLMs), we apply a range of learning rates {5e-6, 8e-6, 1e-5, 2e-5, 5e-5} across different models and datasets. Furthermore, we implement an early stopping strategy to identify the optimal model based on validation performance as the specialized models. We fine-tune each model using a batch size of 8, max sequence length of 4096, across four A6000 48GB GPUs.

For the combined model, we employ a three-layer neural network featuring ReLU activation, with sigmoid activation applied to determine the final weights. At each generation step, only the top 10 logits from both LLMs and SLMs are utilized. The intermediate hidden layers are configured with sizes of 512 and 16, respectively. The model is trained using a learning rate of 2e-3 and a batch size of 2. The combined model is trained on the training dataset, with both LLMs and SLMs assigned the same target response. Additionally, an early stopping strategy based on the validation set performance is employed to select the optimal combined model. For all the aforementioned experiments, we calculate the mean scores using three distinct random seeds.

C Evaluation Details

Owing to the costs associated with evaluation, we assess only a portion of the test samples. To account for GPT-4's evaluation stability, we plot a curve illustrating the relationship between scores and the number of evaluated samples. As depicted in Figure 7, the results stabilize once the number of samples reaches 100. Therefore, we randomly select 100 samples from the test set, of which only 80 samples are utilized in the widely recognized benchmark, MT-Bench [9].

D Prompt Details

Prompts designed for querying Large Language Models (LLMs) both with and without context are outlined in Table 7. Prompts intended for extracting outlines are illustrated in Table 8. Prompts used for GPT-4 based evaluation are depicted in Table 9.

E Related Works

E.1 Instruction Following and Privacy

Large language models (LLMs), after being trained on high-quality instruction data and calibrated to align with human intentions, have acquired the capability to execute instructions across a range of activities such as creative writing (Franceschelli and Musolesi, 2023), coding (Qian et al., 2023), debugging (Jimenez et al., 2023), and various other text-based tasks (Bubeck et al., 2023). Contemporary research in instruction following prioritizes the acquisition of high-quality data (Wang et al., 2023c), which encompasses instructions of varied complexity (Xu et al., 2023a) and diversity (Ding et al., 2023; Cui et al., 2023), as well as ensuring a minimal dataset size for effective generalization (Chen et al., 2023a; Wei et al., 2023). Thanks to these technological advancements along with alignment algorithms (Rafailov et al., 2023), medium-scale language models with around ten billion parameters have been made open-source and perform adeptly at following instructions (Tunstall et al., 2023; Jiang et al., 2023; Zhang et al., 2024).

[8] https://github.com/vllm-project/vllm

[9] https://huggingface.co/spaces/lmsys/mt-bench

Nevertheless, premier chat models like ChatGPT, GPT-4, Claude, and Gemini remain proprietary, largely due to commercial considerations, necessitating that our instruction data be uploaded to the cloud (Achiam et al., 2023; Team et al., 2023). While these models serve as potent AI assistants in daily professional and personal endeavors, they also pose significant privacy risks (Liu et al., 2023). To address privacy concerns, LLMs have employed various techniques (Cummings et al., 2023), including text sanitization (Kan et al., 2023), differential privacy (Chen et al., 2023b; Wu et al., 2023), and hidden representations (Zhou et al., 2022, 2023). However, these methods still involve uploading potentially sensitive data to the cloud, which inherently cannot eliminate the risk of privacy breaches.

Moreover, current instruction datasets predominantly cover general domains, with insufficient focus on contextual information modeling. The most closely related works involve personalized response generation, evolving from traditional benchmarks (Salemi et al., 2023; Wang et al., 2023b). The integration of extensive context information into general open-domain instructions remains an area of ongoing exploration. This paper aims to delve into context-aware instruction formulation as a means to advance research on privacy considerations within the realm of instruction following.

E.2 Mixed-Scale Models Collaboration

The "scaling law" in language modeling posits that models with a greater number of parameters exhibit enhanced capabilities (Kaplan et al., 2020). However, these more robust models also encounter challenges related to higher inference costs, efficiency (Xia et al., 2024), and privacy concerns (Yao et al., 2023). Conversely, smaller models, ranging from 1 to 2 billion parameters, are gaining popularity due to their increasingly impressive performance (Bai et al., 2023; Gunasekar et al., 2023; Zhang et al., 2024; Grangier et al., 2024; Singer et al., 2024). These specialized models are coupled with lower inference costs and the feasibility of deployment on consumer-grade desktops and smartphones (MLC team, 2023). Collaborations between mixed-scale models represent a promising research avenue. The body of current research in this area primarily falls into two categories: training and inference. For collaborative training, Offsite-tuning has been introduced as a method to protect both user data and the privacy of large models (Xiao et al., 2023; Zhang et al., 2023c). This approach involves using an emulator derived from LLMs, fine-tuning it on specific downstream data, and subsequently integrating the learned parameters back into the LLMs. On the inference front, techniques like speculative decoding (Leviathan et al., 2023; Xia et al., 2024) and contrastive decoding (Li et al., 2022; O'Brien and Lewis, 2023) aim to enhance and expedite LLMs' inference processes by leveraging smaller draft or expert models. Additionally, emulator tuning (Mitchell et al., 2023) and proxy tuning (Liu et al., 2024) have been devised to economize on fine-tuning large models; however, they can also be considered forms of collaborative decoding during inference. This paper focuses on examining collaboration during inference, specifically investigating sketch-based and logit-based methods.

SYSTEM PROMPT FOR REQUEST WITHOUT CONTEXT You are now a helpful personal AI assistant. Aim for insightful and high-quality solutions that make users satisfied.
SYSTEM PROMPT FOR REQUEST WITH CONTEXT IN CONTEXT-AWARE You are now a helpful personal AI assistant. You should emulate the author's style and tone based on provided history content. Your responses should be detailed and informative, using the personal information reasonably in the user's profile. Aim for insightful and high-quality solutions that make users satisfied.
SYSTEM PROMPT FOR REQUEST WITH CONTEXT IN PERSONALIZED EMAILS AND PAPERS You are now a helpful personal AI assistant. You should emulate the author's style and tone based on provided history content. Your responses should be detailed and informative, matching the author's unique writing approach. Aim for insightful and high-quality solutions that make users satisfied.
FEW-SHOT INSTRUCTION FOR REQUEST WITH CONTEXT IN CONTEXT-AWARE ## User Profile {profile} ## User Writing History {history} ## Task {task}
FEW-SHOT INSTRUCTION FOR REQUEST WITHOUT CONTEXT IN CONTEXT-AWARE {task}
FEW-SHOT INSTRUCTION FOR REQUEST WITH CONTEXT IN PERSONALIZED EMAILS ## History Emails {examples} ## Task Compose an email for the subject '{task}' that matches the author's unique style and tone.
FEW-SHOT INSTRUCTION FOR REQUEST WITHOUT CONTEXT IN PERSONALIZED EMAILS Compose an email for the subject '{task}'
FEW-SHOT INSTRUCTION FOR REQUEST WITH CONTEXT IN PERSONALIZED PAPERS ## History Paper Abstracts {examples} ## Task Compose an abstract for the title '{task}' that matches the author's unique content, style and tone.
FEW-SHOT INSTRUCTION FOR REQUEST WITHOUT CONTEXT IN PERSONALIZED PAPERS Compose an abstract for the title '{task}'

Table 7: Prompts for querying LLMs with and without context.

PROMPT FOR EXTRACTING OUTLINE OF CONTEXT-AWARE.

You're an organizer responsible for only giving the skeleton (not the full content) for answering the question. Provide the skeleton in a list of points (numbered 1., 2., 3., etc.) to answer the question. Instead of writing a full sentence, each skeleton point should be very short with only 3-5 words. Generally, the skeleton should have 8-15 points. You can refer to the following examples:

[Task1]: Develop a Marketing Script for Your Monthly Dinner Party: Create a script that highlights your monthly dinner party as a networking platform.
[Skeleton1]: 1. Warmly lit dining room\n2. Fine china and gourmet dishes\n3. Soft music background\n4. Invitation opening\n5. Guests arriving and networking\n6. Host's welcoming toast\n7. Expertly paired courses and wine\n8. Animated guest discussions\n9. Guest speaker's address\n10. Post-dinner networking lounge\n11. Online community continuation\n12. Next event date highlighted\n13. Closing with logo and contact info

[Task2]: Compose a reflective essay on the evolution of bridge design: Thomas, with his patent in bridge design, can discuss the evolution of bridge engineering, modern challenges, and future perspectives.
[Skeleton2]: 1. Introduction to bridges\n2. Early bridges: materials, principles\n3. Roman arches, concrete use\n4. Industrial Revolution: iron, steel\n5. Brooklyn Bridge: design icon\n6. 20th-century advances: materials, techniques\n7. Modern challenges: sustainability, climate\n8. Future technologies: smart materials, sensors\n9. Ethical considerations, safety\n10. Conclusion: adaptation, advancement

Now, please provide the skeleton for the following question.
{question}

PROMPT FOR EXTRACTING OUTLINE OF EMAIL.

You're an organizer responsible for only giving the skeleton (not the full content) for answering the question. Provide the skeleton in a list of points (numbered 1., 2., 3., etc.) to answer the question. Instead of writing a full sentence, each skeleton point should be very short with only 3-5 words. Generally, the skeleton should have 8-15 points. You can refer to the following examples:

[Task1]: Compose an email for the subject 'T-Mobile Sidekick debuts, FileMaker launches mobile DB, and more!'
[Skeleton1]: 1. JavaWorld techno-tidbits intro\n2. T-Mobile Sidekick debut\n3. FileMaker mobile DB launch\n4. Palm OS 5 devices release\n5. Mobile security advancements\n6. Newsletter system update\n7. Customer service instructions\n8. JavaWorld team sign-off\n9. Editorial and advertising contacts\n10. Privacy policy reminder\n11. Copyright notice

[Task2]: Compose an email for the subject 'tomcat4, where servlet.jar is set ???'
[Skeleton2]: 1. Tomcat 4 servlet.jar location?\n2. Navigating Tomcat directory.\n3. Specifics for Tomcat 4.\n4. Setting up web application.\n5. Importance of servlets.\n6. Documentation exploration.\n7. Request for expert advice.\n8. Configuration file settings?\n9. Thanks and anticipation.\n10. P.S. Collaboration value.

Now, please provide the skeleton for the following question.
{question}

PROMPT FOR EXTRACTING OUTLINE OF PAPER.

You're an organizer responsible for only giving the skeleton (not the full content) for answering the question from high-level perspective. Provide the skeleton in a list of points (numbered 1., 2., 3., etc.) to answer the question. Instead of writing a full sentence, each skeleton point should be very short with only few words. Generally, the skeleton should have 8-15 points. You can refer to the following examples:

[Task1]: Compose an abstract for the title 'Ensemble of Anchor Adapters for Transfer Learning'
[Skeleton1]: 1. Transfer learning importance\n2. Traditional approaches limitations\n3. Ensemble of Anchor Adapters introduction\n4. Anchor adapters concept\n5. Ensemble strategy for robustness\n6. Hybrid loss function formulation\n7. Experiments on heterogeneous domains\n8. EAA outperforms state-of-the-art\n9. Novel transferability metric introduction\n10. Contribution: ensemble and domain adaptation integration

[Task2]: Compose an abstract for the title 'Variability in software architecture: the road ahead'
[Skeleton2]: 1. Software architecture evolution\n2. VARSA symposium introduction\n3. Previous work foundation\n4. Challenges and opportunities\n5. Keynote speeches, research, collaboration\n6. Capturing and leveraging variability\n7. Cognitive and technical burdens\n8. Variability's impact on quality\n9. Lifecycle integration\n10. Research agenda proposal\n11. Interdisciplinary dialogue\n12. Tools, techniques, theory advancements\n13. Roadmap for strategic directions\n14. Conference essence and goals

Now, please provide the skeleton for the following question.
{question}

Table 8: Prompts for extracting outlines.

EVALUATION INSTRUCTION FOR OVERALL QUALITY OF GENERATED CONTENT.

[Instruction]
Please act as an impartial evaluator and assess the quality of the AI assistant's response to the user question shown below. Your assessment should focus on how well the response aligns with the user's personalized profile and writing history. Evaluate factors such as the response's adherence to the user's personal style, consistency with their profile, helpfulness, relevance, accuracy, depth, creativity, and level of detail. Begin your evaluation by providing a short explanation. Be as objective as possible. After providing your explanation, you must rate the response on a scale of 1 to 10 by strictly following this format: "[[rating]]", for example: "Rating: [[5]]".

[User Profile and Writing History]
{profile_info}
{writing_history}

[Question]
{question}

[The Start of Assistant's Answer]
{answer}
[The End of Assistant's Answer]

EVALUATION INSTRUCTION FOR OVERALL QUALITY OF GENERATED CONTENT WITHOUT PROFILE.

[Instruction]
Please act as an impartial judge and evaluate the quality of the response provided by an AI assistant to the user question displayed below. Your evaluation should consider factors such as the helpfulness, relevance, accuracy, depth, creativity, and level of detail of the response. Begin your evaluation by providing a short explanation. Be as objective as possible. After providing your explanation, you must rate the response on a scale of 1 to 10 by strictly following this format: "[[rating]]", for example: "Rating: [[5]]".

[Question]
{question}

[The Start of Assistant's Answer]
{answer}
[The End of Assistant's Answer]

EVALUATION INSTRUCTION FOR CONSISTENCY BETWEEN GENERATED CONTENT AND PERSONAL PROFILE.

[Instruction]
Please act as an impartial judge and evaluate the AI assistant's response based on its alignment with the user's personal profile and writing history. Focus your assessment on the personalization aspects of the response, including its adherence to the user's unique style, preferences, and consistency with their profile. Consider how well the response addresses the user's individual needs and interests. Begin your evaluation by providing a short explanation. Be as objective as possible. After providing your explanation, you must rate the response on a scale of 1 to 10 by strictly following this format: "[[rating]]", for example: "Rating: [[5]]".

[User Profile and Writing History]
{profile_info}
{writing_history}

[Question]
{question}

[The Start of Assistant's Answer]
{answer}
[The End of Assistant's Answer]

Table 9: Prompts for GPT-4 based evaluation.