Panza: Investigating the Feasibility of Fully-Local Personalized Text Generation

Armand Nicolicioiu[1*†], Eugenia Iofinova[2*], Andrej Jovanovic[3*†], Eldar Kurtic[2,4*], Mahdi Nikdan[2*], Ilia Markov[4†], Andrei Panferov[2], Nir Shavit[4,5], Dan Alistarh[2,4]
[1]ETH Zurich, [2]IST Austria, [3]University of Cambridge, [4]RedHat AI, [5] Massachusetts Institute of Technology
eugenia.iofinova@ist.ac.at, dan.alistarh@ist.ac.at

The availability of powerful open-source large language models (LLMs) opens exciting use cases, such as using personal data to fine-tune these models to imitate a user's unique writing style. Two key requirements for this functionality are *personalization*–in the sense that the output should recognizably reflect the user's own writing style—and *privacy*–users may justifiably be wary of uploading extremely personal data, such as their email archive, to a third-party service. In this paper, we demonstrate the feasibility of training and running such an assistant, which we call *Panza*, on commodity hardware, for the specific use case of email generation. Panza's personalization features are based on a combination of parameter-efficient fine-tuning using a variant of the Reverse Instructions technique [1] and Retrieval-Augmented Generation (RAG) [2]. We demonstrate that this combination allows us to fine-tune an LLM to reflect a user's writing style using limited data, while executing on extremely limited resources, e.g. on a free Google Colab instance. Our key methodological contribution is the first detailed study of *evaluation metrics* for this task, and of how different choices of system components–the use of RAG and of different fine-tuning approaches–impact the system's performance. Additionally, we demonstrate that very little data - under 100 email samples - are sufficient to create models that convincingly imitate humans, showcasing a previously unknown attack vector in language models. We are releasing the full Panza code as well as three new email datasets licensed for research use.

1. Introduction

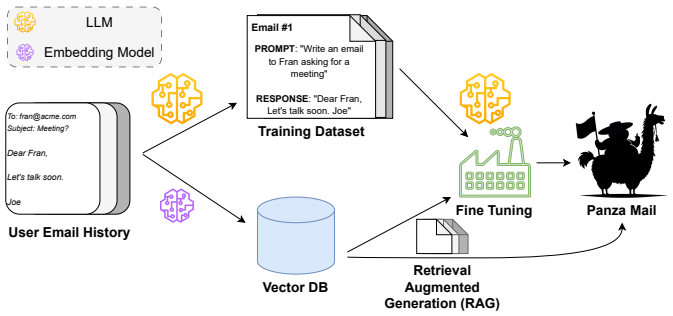
Large Language Models (LLMs) are natural candidates for implementing personal assistants, defined as software that can help a user with various repetitive tasks such as email, writing, or summarization. At the same time, many highly capable LLMs can only be accessed via an API or web interface, either because their weights are proprietary, or because the model size makes self-hosting and customized finetuning cost-prohibitive. This makes it expensive or impossible to support certain features for automated personal assistants:

1. **Personalization**: customizing the model to someone's tone, preferences, and personal history;
2. **Privacy**: allowing the model to have access to highly personal information of a caliber that—unlike corporate data—most people would not agree to share, even if promised that the data is protected by the provider's cloud.

A natural approach to addressing these constraints would be to *execute these models locally, on the user's own data and hardware*. However, this poses obvious challenges in terms of *data* and *hardware* efficiency. In this paper, we demonstrate the feasibility of designing and implementing a *fully-local personalized LLM-powered e-mail writing assistant*, which we call Panza. Panza generates e-mail messages in the user's style, given a short user prompt and, optionally, access to a set of previous emails. We demonstrate that it is possible to obtain a capable assistant starting from relatively-small existing pre-trained models such as Llama-3-8B [3] or Phi-3-Mini [4], that these models can be trained on a machine with a commodity GPU, and run on a commodity laptop, such as a MacBook, and that a small amount of user-generated text suffices for this task. Thus, we present findings from the following research perspectives:

*Equal Contribution.
†Work done while at IST Austria

Third Conference on Parsimony and Learning (CPAL 2026).



Datasets used

Name	Source	Public	Number of Emails
Tana	ENRON	✓	742
Jeff	ENRON	✓	573
Kay	ENRON	✓	481
Sara	ENRON	✓	464
David	Donated	✓(new)	166
Isabel	Donated	✓(new)	210
Marcus	Donated	✓(new)	150
Anon1	Donated		301
Anon2	Donated		669

Figure 1: (Left.) Panza's overall design. Given a set of emails produced by the user, we produce both a finetuning dataset (using Reverse Instructions) and retrieval augmented generation (RAG) database. (Right.) E-mail datasets used. The David, Isabel, and Marcus datasets are published along with this work.

1. We extend the Reverse Instructions algorithm of Köksal et al. [1] to the case of personalizing a generative LLM's output to match a user's writing style - that is, by applying it to the personalization problem, by demonstrating its effectiveness on much smaller data sizes, and by combining it with Retrieval-Augmented Generation (RAG) [2], used to retrieve relevant emails at inference time to provide additional context.
2. We perform an extensive analysis of the *text personalization task* in this context. We begin with an investigation of evaluation metrics, in conjunction with studying the impact of different design options, training, and deployment techniques on the final model performance. First, we show that the traditional BLEU / ROUGE / BERT scores for measuring text generation quality show similar trends, but *fail to fully capture performance on this task*. However, we show that the *combination of BLEU and MAUVE scores* provides very strong correlations with performance across various user studies.
3. With these metrics in place, we show that *fine-tuning on the user's data* consistently outperforms a pre-trained instruction-tuned model, even when augmented with RAG. Further, we perform one of the first in-depth studies of the impact of RAG on *personalized* model performance when applied at training time or at inference time for long-form generation.
4. We show that this entire pipeline can be executed in a *resource-constrained setup*. Both the fine-tuning and the inference / RAG components can be run efficiently and accurately on a system with a single commodity GPU, e.g. a free-tier Google Colab instance. The best parameter-efficient fine-tuning (PEFT) results are obtained using the Robust Adaptation (RoSA) method [5], which we find particularly suited for style transfer. Our main innovation here is a new and accurate adapter merging, required to combine sparse and low-rank PEFT adapters into quantized weights. Additionally, we demonstrate the practical feasibility of using Panza on a personal computer through a combination of Ollama [6] and a new Chrome plugin which integrates Panza into GMail.
5. Finally, we highlight the opportunities as well as *emerging risks* of open-source LLMs by demonstrating that as few as 50-100 emails suffice to make a convincing imitation of a person's writing style. This uncovers a novel attack vector, where malicious actors can use leaked data to create credible—and difficult-to-trace—imitations of the breach victims' writing style.

These findings are confirmed via a range of human user studies, showing that, using our techniques, fine-tuning existing open LLMs achieves high-quality personalized text generation, and that the resulting models are both sufficiently convincing and sufficiently differentiated, in the sense that humans can attribute the resulting Panza outputs to the correct (human) author.

Related Work. Personalization has seen a lot of interest for LLMs, e.g. King and Cook [7], as the recent explosion of performant models at the 4-8B parameter scale has made individualized personalization a viable goal [8–10]. To organize these efforts, the Language Personalization benchmark [11] was recently introduced for classification and summarization-style tasks. A version of this benchmark for long-form text generation has more recently become available [12]; however, as the legality and ethical use of the data in the benchmark is not clear, we do not use it here.

Our work builds upon the Reverse Instructions algorithm [1], which allows us to convert long-form user data into training data by using an LM to write synthetic prompts for each email. However, this work did not consider personalization. Sennrich et al. [13] uses generated prompts for machine translation, but this paper does not deal with long-form text generation. Instruction Backtranslation [14] contains a step in which an LM is used to construct a prompt that would result in the generation of an unlabeled document. However, the authors report that for their task, this technique was insufficient to construct useful training data, and additional filtering measures to improve quality were necessary.

We show both via examples and metrics comparisons that both finetuning and RAG are helpful for the task: while base LLMs with RAG can provide coherent outputs, including personal information obtained via retrieval, they largely fail to match a given user's writing style. A similar trend has been observed in OPPU [15], which also deals with personalized finetuning, but they investigate this problem in the context of text condensation, such as article titling and text paraphrasing, while Panza tackles the harder problem of long-form text generation from a shorter prompt. [16] investigated personalization via prompt engineering, agreeing with our results that prompt engineering without finetuning is insufficient to match style.

2. Method

The Panza design, described in Figure 1, requires a pre-trained, instruction-tuned LLM and a set of emails sent by the user. Both the LLM and the emails have dual use. First, the pre-trained LLM is used to rephrase the "raw" user emails in the form of instructions, which will be used for finetuning. Then, the LLM itself is going to be fine-tuned on these instructions, resulting in the Panza personalized model. Independently, emails are integrated into a RAG database, used at runtime.

2.1. Fine-tuning via Reverse Instructions

Finetuning is the core personalization technique behind Panza. The task data is typically a set of several hundred emails in the user's history. We convert these emails to training data by a variant of Reverse Instructions. The intuition is that we would like to "reduce" the problem of generating emails in the user's style to a specific instance of instruction-tuning, where the user query comes in the form of an instruction, e.g. "Write an email to Joanne to set up a meeting on Wednesday.", and the "correct" answer would be such an email written in the user's style. To induce this behavior from the LLM, our training works in two steps.

First, we use a pre-trained LLM to rephrase each email into a targeted instruction, containing just enough information for an assistant to write the original email. See Appendix D for the prompt we use for this task. Then, we fine-tune the backbone LLM on the (instruction, email) pairs obtained in the first step, with the standard next-token prediction training objective, which induces the LLM to reconstitute the original email as a response to the instruction. Optionally, we implement a training-time RAG component, which retrieves query-related emails (via cosine similarity of the emails and query in the embedding space of a smaller LLM), and provides them as context to the LLM at training time. This is similar to retrieval-augmented fine-tuning (RAFT) [17]. At the end of these two steps, we have obtained a personalized LLM that can respond to short user queries by writing targeted emails that should follow the user's style.

2.2. Local Fine-Tuning

In the absence of computational or memory constraints, we perform full fine-tuning (FFT) of the base model, and inference over it, possibly adding a retrieval-augmented generation (RAG) component, which retrieves similar emails sent in the past. However, fine-tuning and deploying a powerful billion-parameter model locally requires a powerful GPU with significant memory. Therefore, we investigate techniques for reducing these costs, as well as their impact in terms of accuracy metrics.

Memory efficiency is critical in our setting. For instance, full fine-tuning (FFT) of a `Mistral-7B` model [18] in half-precision using a standard Adam optimizer [19] requires more than `60GB` of GPU memory, which is not available on a consumer-grade machine. We tailor Panza to two resource-constrained settings: running on a GPU with under `24GB` RAM, and under `15GB` RAM. Throughout, we use `Mistral-7B` [18] as a running example for costs, but the techniques apply to other base LLMs of a similar size. We always consider a local

training micro-batch size of 1 to minimize memory footprint, accumulating gradients whenever necessary. See Section 4 for more details. Training usually takes under an hour.

Finetuning on a single GPU. We first assume access to a single `24GB` GPU (such as NVIDIA GeForce RTX 3090), i.e., a GPU desktop or a strong gaming laptop. In this case, we use Parameter-Efficient Fine-Tuning (PEFT) methods, which tune only a small set of parameters to enable adaptation of models to downstream tasks. We investigate the standard LoRA method [20], and the more recent Robust Adaptation (RoSA) method [5], which we find to be particularly effective for style transfer. Specifically, by training a combination of low-rank and sparse adapters on top of the base weights, RoSA allows effective fine-tuning of a half-precision `8B` model on less than `24GB` of memory, with competitive accuracy relative to full fine-tuning. For deployment, we merge the RoSA adapters into the base model weights, with the inference requiring around `15GB`. In Appendix A we demonstrate the feasibility of training Panza in under 15GB of memory by additionally incorporating quantization, including a detailed study of quantization approaches.

2.3. Efficient Inference

If the GPU is local to the user, for instance, on a gaming laptop, then the same hardware can simply be used to run inference on the model at any time. If, however, the model is trained on external hardware (for instance, on Google Colab, Lightning Studios, or another cloud provider), using the same hardware for inference becomes impractical: keeping the hardware running with the model preloaded is expensive, while loading the model onto the GPU every time Panza is called is too slow to be practical. Instead, we convert the model to a `.cpp` version that runs on CPU via Ollama [6]. Using this setup, we are able to run the model on an M-Series MacBook Pro (M3 Pro Chip with 18 GB of memory), achieving 16.09 tokens per second, or about 6.21 seconds to generate an average email. To further demonstrate the potential for a positive user experience, we built a sample Google Chrome extension that incorporates Panza into GMail (see Appendix B).

3. Evaluation Protocol

3.1. Datasets

A key challenge of this project is the lack of availability of personal e-mail datasets due to the sensitive content of the data. To our knowledge, the only available repository contains the business e-mails of 144 Enron employees, originally released by the Federal Energy Regulatory Committee; the version we use [21] is licensed for research. We use the emails of four employees (Sara Shackleton, Kay Mann, Jeff Dasovich, and Tana Jones) with over 400 (English) e-mails each, hereafter identified by their first names. To avoid influencing the model by explicitly invoking Enron, *we changed the names of the corporation and its executives*.

We use five additional datasets. The "David", "Isabel", and "Marcus" datasets were anonymized manually by the authors and donated for research use by volunteers with a clear understanding of their release and proposed use; these datasets will be released as part of this project. Two additional datasets, identified as Anon1 and Anon2, were also donated by volunteers for this project, but will not be released due to the sensitive nature of their contents. Of the nine datasets used, six contain primarily business emails, and three contain a mix of business and personal emails. The train-test split for fine-tuning / PEFT is 80-20%, and only training set emails are used to form the RAG database employed at inference time.

3.2. Metrics

Panza uses the text generation capabilities of LLMs for two tasks: to summarize a user's emails to create synthetic prompts via Reverse Instructions, and to generate new emails in the test-time prompts. These two use cases both rely on the ability of LLMs to summarize or rephrase content; the email generation task additionally requires the recall of both general and user-specific knowledge, and an imitation of a user's personal style. Thus, we divide the evaluation benchmarks into four categories: phrasing quality, user-specific knowledge, general knowledge, and style.

Phrasing quality. We measure phrasing quality with the BLEU [22] and ROUGE [23] metrics, which are standard to measure translation and summarization quality. Both metrics function by counting matching

N-grams between the LLM output and one or several 'golden' responses (for email generation, the golden response is the actual user email). The BLEU score is a weighted measure of 1-4-grams that match exactly between the output and golden text strings, normalized by the string length; we use an equal weight of 0.25 for each N-gram length. ROUGE reports 1-gram and 2-gram precision, recall, and F1-score, as well as the longest-substring precision, recall, and F1-score; in our paper, we use the longest-substring F1 score for maximum contrast with the BLEU metric. For both metrics, we use the Torchmetrics package; they are computed for each prompt/output combination, and we report the average across all prompts as the overall value. We do not compute either metric on a per-sentence basis, but rather compare N-grams in the full email text after dropping punctuation.

User-specific knowledge. As there is no database of user-specific knowledge, we restrict the evaluation of such knowledge to what is contained in the user's emails. This working assumption enables us to use RAG-assisted email generation; here, it further enables us to evaluate user-specific knowledge on the same test dataset as for the paraphrasing quality. We do not attempt to decouple knowledge content from paraphrasing quality, rather relying on the overall BLEU and ROUGE scores to reflect the correct imputed information. Note that, unlike the general knowledge desideratum, the user-specific information is better specified—a prompt requesting the user's current address is easier to evaluate than one asking for suggestions for a travel destination—and so the N-gram match is an appropriate measure of quality.

General knowledge. For world knowledge measurements, we rely on the standard six Open-LLM leaderboard evaluation suite [24]. Together, these tasks test the model's ability to perform basic reasoning and factual recall in English, and are used as a broad evaluation of a model's quality. Overall, Panza models retain over 98% of the performance of Llama-3-8B-Instruct on this benchmark. Full details are in Appendix G.

Style transfer. To measure the quality of style transfer, we use the MAUVE score [25], introduced to measure the degree of similarity between machine-generated and human-written text. Specifically, the MAUVE score relies on an estimate of the symmetric K-L divergence between the distribution of authentic text and the distribution of model outputs, which we parametrize as standard. A higher MAUVE score is correlated with higher similarity between the generated text and human text.

Discussion. A major challenge of this project is that the email generation task differs significantly from summarization or translation tasks, in that we generally expect the output email to be *longer and more detailed* than the prompt used to generate it, thus requiring some improvisation on the part of the model, and making it unlikely for the generated email to be close to the actual one. This is reflected in the BLEU/ROUGE scores, which are lower than what the reader may have observed for translation or summarization tasks. Manual review during the project development phase has shown that, nevertheless, these scores appear highly correlated with output email quality. We provide examples in Appendix F, validate BLEU and MAUVE in a human study in Appendix K, and consider BERT-score and METEOR metrics in Appendix J. As a rough guideline,

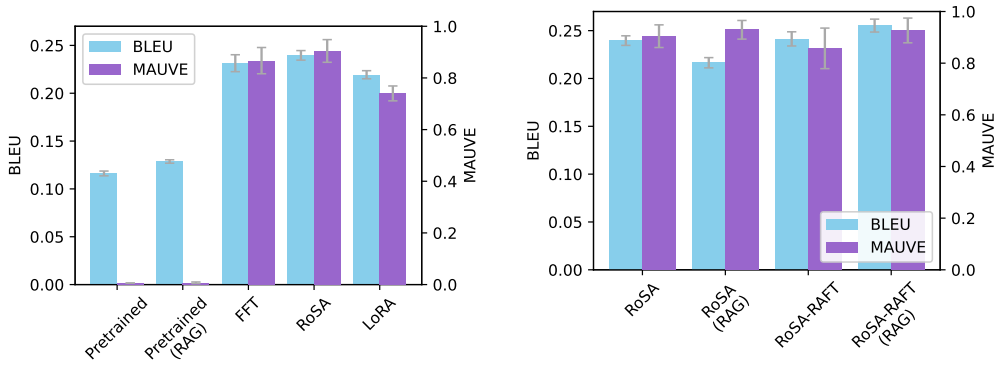


Figure 2: (Left) Average performance of finetuning-based methods compared to pretrained baselines (Llama3-8B-Instruct with and without RAG at inference). All versions of finetuning (FFT, RoSA, and LoRA) outperform the baselines; RoSA and FFT outperform LoRA. (Right) Comparison between instruction-only fine-tuning and Retrieval-Augmented Fine-Tuning (RAFT) for RoSA on Llama3-8B, averaged across users.

human subjects generally agreed that models achieving above 0.2 average BLEU and 0.75 MAUVE produced plausible emails.

4. Experimental Results

4.1. Baselines

As no baselines appear to exist for the task of *personalized* email generation, we investigate the effect of finetuning by baselining against simply using prompt engineering to elicit personalization. (We additionally compared to the results of Köksal et al. [1] in Appendix H and to GPT-4o prompted with a large subset of the user's entire email history (similar to the notion of 'profiles' in [11]) in Appendix I.) We start with publicly available instruction-finetuned LLMs: Meta-Llama-3-8B-Instruct, Mistral-7B-Instruct-v0.2, and Phi-3-mini-4k-instruct. As a first baseline, we prompt models in the following format. The system preamble sets the role of the LLM as an email writing assistant, while the user preamble contains general information about the user, such as name, occupation, and address, which has been shown to be essential to personalizing LLMs through prompt engineering [26]. Finally, we provide the actual email writing instruction, as created during the Reverse Instructions step.

This baseline (denoted **Pretrained**) writes well-structured emails, without explicit personalization. Next, we test if presenting a few samples of the user's previous emails during inference through RAG can provide enough style information. We select the closest n_{RAG} previous emails, filtered by a relevancy threshold T_{RAG}, and add them to the input as an additional preamble. We denote this baseline as **Pretrained + RAG**.

4.2. Fine-Tuning via Reverse Instructions

After generating pairs of (instruction, email), we fine-tune the pre-trained LLM to reconstruct the email given the instruction. We analyze the following regimes: full fine-tuning (**FFT**) and two PEFT methods: Robust Adaptation (**RoSA**) and Low-Rank Adaptation (**LoRA**). For both training and testing, the input is formatted with the same system and user preambles described for the baselines. We also test whether RAG can bring additional improvements.

Furthermore, we explore if the model can learn how to better leverage previous e-mails in RAG by presenting the same type of augmented prompt during fine-tuning, parameterized by the number of closest emails n_{RAG} and the relevancy threshold T_{RAG}. Additionally, to make the model robust to the absence of similar emails, we have a p_{RAG} chance of not using any similar emails for a particular instruction during fine-tuning, even if there are matches in the database. This adapts Zhang et al. [17] to our setting and is denoted as **RAFT** (Retrieval-Augmented Fine-Tuning).

Hyperparameter tuning. We found fine-tuning, and especially PEFT, to be highly sensitive to learning rate and the number of training epochs. To find suitable hyperparameters, we first used a greedy grid search approach with learning rates ranging from 10^{-6} to 10^{-4}, and epoch ranges from 1 to 9, batch sizes of 8 and 16, and using the BLEU metric as the proxy for overall model quality. We used the realistic, non-anonymized Anon1 and Anon2 datasets for hyperparameter tuning and chose values that worked well for both. Overall, we found that a learning rate of 10^{-5}, batch size of 8, and 5 epochs (3 for FFT) to work well across all base models and finetuning styles. We then did limited tuning of these parameters for the other users. The values for all users are presented in Appendix N.

4.3. Results Across Methods and Models

We find that all fine-tuning regimes outperform the Pretrained + RAG baselines by a large margin, adapting to the user's writing style. The results are illustrated in Figure 2; we present a qualitative comparison of the generated emails in Appendix F; using leading closed LLMs yielded similarly poor results. RoSA performs on par with FFT, consistently surpassing LoRA, especially in terms of the MAUVE score. The same trend is observed for all the backbones we trained, and for all users (see Appendix M for full results).

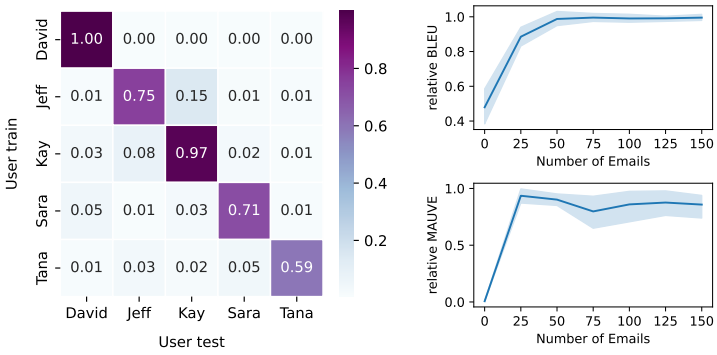


Figure 3: (Left) Style comparison between models trained for different users. Each model, trained for a particular user, is used to generate emails for unseen instructions of all the users. We measure the MAUVE score between the generations and the original emails written by the user. (Right) BLEU and MAUVE scores of Panza models trained on smaller subsets of the training data, relative to the maximum score attained for the dataset. 0 emails corresponds to the un-finetuned Llama3-8B-Instruct model, averaged across seven users.

In Figure 2, we show the effect of RAG on models fine-tuned with RoSA. We observe that, although RAG clearly improves over the pretrained baseline, it reduces the average BLEU score for the RoSA fine-tuned model (second group), but may slightly increase the average MAUVE score. This is not desirable, as BLEU score (relative to the ground-truth email) is a closer measure of content accuracy than MAUVE. Based on analyzing individual samples, we hypothesize that this may be caused by the model re-using the RAG context too aggressively at deployment time, leading to emails that are very similar to past emails–preserving style–but less accurate in terms of content. We overcome this issue by introducing RAG during fine-tuning, i.e., using RAFT: the model can then "learn" to ignore irrelevant emails present in the RAG context.

4.4. Style Evaluation

Recall that MAUVE measures the gap between machine-generated text and human-written text. Above, we reported the MAUVE scores on the test emails coming from the same user the model was trained for. Next, we do a pairwise comparison, evaluating models trained for different users on the test data of all the other users. This focuses precisely on how well style is reflected in generated emails. In Figure 3, we see that each model produces a high MAUVE score $(0.6 \text{ to } 1.0)$ *only for the test emails of the user it was trained for*, while it has a close to 0 MAUVE score on any other user. In terms of BLEU/ROUGE score, all models have essentially the same performance on the test set of any given user, suggesting all models have similar paraphrasing capacity to express the given instruction, but each does it in the style of the user it was trained for.

4.5. Impact of Training Dataset Size on Model Quality

The success of Panza on the test datasets naturally leads to the question: how little data is needed to create a convincing Panza model? While restricting data size enables further efficiency gains, this functionality also opens up avenues of attack for malicious actors, as they would need access to a small number of emails to successfully imitate a victim.

We performed an ablation study on the number of emails needed to train a Panza model. We trained individual Panza models on the users `david`, `isabel`, `marcus`, `jeff`, `kay`, `tana`, and `sara`, varying the number of emails in the training set from 25 to 150, using RoSA, and compare that to Panza performance when trained on the full available training data (without RAFT or RAG). In order to compensate for the small data size, we increase the learning rate to 0.0001 for dataset sizes of 25 and 50, and we train models of e-mail size 25 and 75 for 11 epochs and models of 50 and 100 for 9 epochs (however, the 75- and 100-email models use the standard learning rate of 1e-5).

We present the aggregate BLEU and MAUVE metrics for this study in Figure 3. In order to meaningfully aggregate between users, we normalize each user's BLEU and MAUVE scores by those obtained by the model

trained on the maximum available number of train emails (80% of the total dataset size). These results show that at a training set of 25 we already observe average BLEU scores at 90% of the maximum and MAUVE scores of about 0.85, suggesting that the resulting PANZA models are picking up many features of the user's style; by roughly 75-100 emails, the BLEU scores essentially match those of the full training sets, although MAUVE scores take longer to saturate. We therefore conclude that even quite small datasets may be sufficient to successfully impersonate a user. We provide a human study testing this conclusion in Section 5.3.

5. User Evaluation

In order to further evaluate the efficacy of Panza in generating effective email personalisation, we create three types of user studies: i) **usefulness** – to assess whether Panza models provide practical benefit for the user in their email generation, ii) **persona recognition** – to assess whether well-performing Panza models are successful in representing the authors' style, and iii) **impersonation** – to assess whether Panza produces emails that sound believably human. The details of each study's setup are presented in full in Appendix L.

5.1. Usefulness

To assess practical usefulness, we conducted a study in which six subjects were recruited and asked to train Panza models on their own data. The subjects were instructed to train two Panza models, following the standard FFT-RAG and RoSA-RAG setups with no hyperparameter tuning. They then used these models to generate responses to 16 fixed prompts and 16 prompts drawn from the subject's test set. The subjects then assigned each generated email a rating of 3 (Very useful), 2 (Moderately useful), or 1 (Not useful). Full details are available in Appendix L.1.

Table 1: Human evaluation of Panza usefulness, averaged across six subjects.

	Average usefulness score		
Model	Fixed Prompts	Own Prompts	Combined
Llama + RAG	2.11	2.10	2.10
RoSA + RAG	2.59	2.71	2.65
FFT + RAG	2.37	2.75	2.56

The results are presented in Table 1. We observe that, while on average the outputs of all models were rated as somewhat useful or better, the outputs of Panza models were generally found to be more useful than those of Llama + RAG, with average scores of 2.10 for Llama, 2.56 for FFT, and 2.65 for RoSA. We further note that for FFT and Llama, we see substantially better performance on the rater's own prompts than the fixed set, which may be seen as evidence that Panza tailors to the email *content* distribution of the user's prompts.

5.2. Persona Recognition

The intention of this study is to assess whether well-performing Panza models are successful in credibly representing the authors' style. We conducted three separate trials with four participants. Each trial is conducted with a group of four participants. In each trial, two individuals served as the authors and were asked to train Panza models on their personal emails, and two additional evaluators served as evaluators. Crucially, all participants in each trial were personally acquainted with both authors. Each participant was then asked to match the e-mails produced by four Panza models to their source data. Two of the four models were trained by the authors on their data. The third model was finetuned on the `Jeff` dataset, and the final model is an un-finetuned Llama-3-8B-Instruct model augmented with RAG. We note that none of the subjects in this study is personally familiar with `Jeff`; we included this additional control to further validate that the subjects are not just distinguishing finetuned from pretrained model output.

Table 2: Precision and recall of Panza persona attribution, averaged across three trials. Study subject precision and recall numbers are for recognizing the style of each of the two subjects individually.

Person	Precision	Recall
Study subjects (1 and 2)	0.64	0.64
Jeff (ENRON)	0.78	0.78
Llama-3-8B	0.87	0.87

For each evaluation task, evaluators were given 24 prompts and the four email responses from the four different writers in a random order, and asked to attribute these four output emails to the four model versions. The prompts to generate the responses were taken from a curated list of custom prompts.

The results are presented in Table 2, averaged over the three independent trials. We observe that the subjects were able to identify Llama and Panza-Jeff outputs with high accuracy, and were able to identify Subject 1 and Subject 2 from the other Panzas with roughly 64% precision and recall, far higher than random guessing (25%). The greatest source of error was attributing the emails of Subject 1 to Subject 2 or vice versa, which may be due to the fact that each pair of authors is mutual acquaintances, which may indicate increased stylistic similarity. Overall, this study supports our hypothesis that finetuning with Panza creates an identifiable style that can generally be attributed back to the user whose data was used to train the model.

5.3. Impersonation Capabilities

The final user study evaluates whether Panza can produce emails that have *credibility* - in that a human familiar with someone's style would find the emails to be credibly written by that author. For this study, we recruited three pairs of volunteers, one of which served as the author and the other served as the evaluator; we required the evaluator to be personally acquainted with the author and their writing style. (Two of the pairs consisted of the same two people, in alternating roles.) Each of the authors then finetuned two models from Llama3-8B using RoSA on their private data: a model trained on 80% of their training data – `Panza-L` – and a model trained on a random sample of 75 emails from their training data – `Panza-S`; the participants used the standard training hyperparameters for `Panza-L` and trained for a total of 11 epochs for `Panza-S`, matching the settings in Section 4.5.

Table 3: Emails found suspicious by acquaintances of the Panza training set authors, averaged across three trials.

Author	Suspicious
Genuine	24 %
Panza-L	28 %
Panza-S	32 %
Llama-3-8B	64 %

Then, for each user, we sampled a selection of 20 emails from their held-out test set, and evenly distributed them into four groups of five; for each group, we selected one of the following responses: i) the actual email written by the participant, ii) the output of a `Panza-L` model, iii) the output of a `Panza-S` model, and iii) the output of Llama-3-8B-Instruct model. These responses (without the prompts) were then sent to the evaluator associated with the author, who was then told that some of the emails were computer-generated, and asked to rate every response on a binary scale of whether the email is: i) suspicious – likely written by a bot, or, ii) credible: the email sounds like something the author would have written. The evaluators were not told what proportion of emails were genuine or which specific models were used to generate the responses in the study.

We present the results of this study in Table 3. We note that the experimental setup naturally makes the evaluators suspicious of the emails that they see, as they are asked explicitly to consider the genuineness of the emails they evaluate; something people likely don't do every time they see a new email. Even so, of the five emails shown to each user per category, on average, between 28-32% of the `Panza` emails were found suspicious - *in other words, even in this setting, over two-thirds of the emails generated by Panza were found credible*. By comparison, 24% of the genuine emails were found suspicious, as were 64% of the emails generated by the baseline Llama-3-8B model.

These results show that, while emails that the participant wrote themselves were found the most credible on average, emails produced by Panza were also frequently viewed as credible; this happened far less frequently with a plain Llama model. This supports our conclusion that using a Panza model makes it feasible to create messages that are perceived as written by the original author, even when individuals are inclined to be suspicious of authorship.

6. Limitations

The techniques used by Panza provide a significant improvement across a mix of metrics, with respect to on-device personalization, and, we believe, a compelling case of practically useful LLM personalization. Yet, more effort should be dedicated to 1) designing metrics for such novel personalization tasks, and also, more broadly, 2) accurately measuring LLM performance on open-ended tasks such as text generation, with regard to improving its representation of the personal data of the user. Additionally, as a proof-of-concept, Panza has currently only been tested in English and on email data, leaving open the creation of such tools for other languages and types of personalized text generation. We hope that Panza will encourage the extension of our method to such scenarios.

Impact statement

We foresee two categories of risks from presenting a project like Panza. First, a malicious user who has access to a sufficient number of a third party's emails (for instance, the ones received from that party, or available online through open mailing lists) can use a tool like Panza to create a credible malicious imitation of that individual. Second, a style-matching tool like Panza can be used to create derivative work that can credibly be misrepresented as original (for instance, for completing school assignments). Panza's low price and accessibility may aid in such misuse; however, we note that the existence of open-source text generation models and finetuning methods, as well as the text personalization tools available through service-based models such as Google Gemini, already give more sophisticated users access to these technologies. We believe that the relatively high fidelity of emails generated using our approach is a strong incentive towards consistently implementing and using cryptographic techniques such as digital signatures in order to curtail the impact of such advanced personalized text-writing tools.

References

[1] Abdullatif Köksal, Timo Schick, Anna Korhonen, and Hinrich Schütze. Longform: Effective instruction tuning with reverse instructions. *arXiv preprint arXiv:2304.08460*, 2023.

[2] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

[3] Meta. Llama 3 model card. `https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md`, 2024.

[4] Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024.

[5] Mahdi Nikdan, Soroush Tabesh, Elvir Crnčević, and Dan Alistarh. RoSA: Accurate parameter-efficient fine-tuning via robust adaptation. In *International Conference on Machine Learning (ICML)*, 2024.

[6] Ollama. Ollama. `https://github.com/ollama/ollama`, 2023.

[7] Milton King and Paul Cook. Evaluating approaches to personalizing language models. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, 2020.

[8] Jin Chen, Zheng Liu, Xu Huang, Chenwang Wu, Qi Liu, Gangwei Jiang, Yuanhao Pu, Yuxuan Lei, Xiaolong Chen, Xingmei Wang, et al. When large language models meet personalization: Perspectives of challenges and opportunities. *arXiv preprint arXiv:2307.16376*, 2023.

[9] Hannah Rose Kirk, Bertie Vidgen, Paul Röttger, and Scott A Hale. The benefits, risks and bounds of personalizing the alignment of large language models to individuals. *Nature Machine Intelligence*, 2024.

[10] 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. Personalization of large language models: A survey. *arXiv preprint arXiv:2411.00027*, 2024.

[11] Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. LaMP: When large language models meet personalization. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Association for Computational Linguistics*. Association for Computational Linguistics, 2024.

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

[13] Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 2016.

[14] Xian Li, Ping Yu, Chunting Zhou, Timo Schick, Omer Levy, Luke Zettlemoyer, Jason Weston, and Mike Lewis. Self-alignment with instruction backtranslation. *International Conference on Learning Representations (ICLR)*, 2024.

[15] Zhaoxuan Tan, Qingkai Zeng, Yijun Tian, Zheyuan Liu, Bing Yin, and Meng Jiang. Democratizing large language models via personalized parameter-efficient fine-tuning. *arXiv preprint arXiv:2402.04401*, 2024.

[16] Zhengxiang Wang, Nafis Irtiza Tripto, Solha Park, Zhenzhen Li, and Jiawei Zhou. Catch me if you can? not yet: Llms still struggle to imitate the implicit writing styles of everyday authors, 2025. URL `https://arxiv.org/abs/2509.14543`.

[17] Tianjun Zhang, Shishir G Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E Gonzalez. Raft: Adapting language model to domain specific rag. *arXiv preprint arXiv:2403.10131*, 2024.

[18] 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. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

[19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*, 2015.

[20] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[21] William W. Cohen. Enron email dataset. `https://www.cs.cmu.edu/~./enron/`, 2015.

[22] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Annual Meeting of the Association for Computational Linguistics*, 2002.

[23] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Annual Meeting of the Association for Computational Linguistics*, 2004.

[24] Edward Beeching, Clémentine Fourrier, Nathan Habib, Sheon Han, Nathan Lambert, Nazneen Rajani, Omar Sanseviero, Lewis Tunstall, and Thomas Wolf. Open llm leaderboard. `https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard`, 2023.

[25] Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaïd Harchaoui. Mauve: Measuring the gap between neural text and human text using divergence frontiers. In *Neural Information Processing Systems*, 2021.

[26] Bin Wu, Zhengyan Shi, Hossein A. Rahmani, Varsha Ramineni, and Emine Yilmaz. Understanding the role of user profile in the personalization of large language models. *arXiv preprint arXiv:2406.17803*, 2024.

[27] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*, 2023.

[28] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*, 2022.

[29] Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, Jason Phang, Laria Reynolds, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation. `https://doi.org/10.5281/zenodo.5371628`, September 2021.

[30] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[31] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations (ICLR)*, 2020.

[32] Alon Lavie and Abhaya Agarwal. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *WMT@ACL*, 2007.

[33] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The faiss library. *arXiv preprint arXiv:2401.08281*, 2025.

A. Panza under 15GB GPU memory

This setting is interesting since it allows training and deploying Panza on a free Colab instance (NVIDIA T4 GPU). In addition to the GPU memory constraint, we also assume access to only 12GB of CPU memory, as is the case for Colab. To accomplish this, we quantize multiple pipeline components to 4 bits/parameter.

1. **Quantized base weights**. We store the weights of the base model in doubly-quantized 4 bit precision [27], reducing the model size in memory by roughly 4×.

2. **Quantized accumulators**. As part of the sparse adapter's mask generation, RoSA accumulates gradients of the base weights on CPU, violating the 12GB CPU memory restriction. To remedy this issue, we change the precision of the accumulators to 4 bits using a uniform quantization with group size 128.

3. **Adapter merging**. To merge the half-precision RoSA adapters and the 4-bit base weights, we present a new accurate merging mechanism; for each layer, we dequantize the base weights, add adapters, and quantize the result back to 4 bits using the GPTQ method [28]. Our key contribution is an *adapter-aware* implemetation of the GPTQ algorithm, where the quantization is interleaved with merging per layer, without materializing the full half-precision model.

A.1. The Impact of Compression

We evaluate this memory-efficient version of Panza, which requires less than 15GB of GPU memory, and 12GB CPU RAM. We first examine how quantizing each component of the pipeline affects the results, and show that quantizing all the components at the same time can achieve reasonable results. We fine-tune the Mistral-Instruct-7b-v0.2 model using RoSA on the David dataset, and use the same three BLEU, ROUGE, and MAUVE metrics for evaluation.

Table 4: Effect of applying 4-bit quantization to different components of Panza on the David dataset without RAG. We use Mistral-Instruct-7b-v0.2 for both summarization and fine-tuning. BNB stands for bitsandbytes [27] and GPTQ indicates the quantization technique of [28]. Further, QSum: summarization with quantized model, QRoSA: RoSA with quantized base model, QAcc: quantized gradient accumulation for RoSA, QInf: Inference with quantized model, and QAll: all components quantized at the same time.

	BLEU	ROUGE	MAUVE
All BF16	0.265	0.432	0.971
QSum (BNB)	0.153	0.283	0.998
QRoSA (BNB)	0.268	0.417	0.971
QAcc (Uniform)	0.239	0.407	0.971
QInf (BNB)	0.084	0.192	0.130
QInf (GPTQ, ours)	0.251	0.414	0.971
QAll (BNB Inf)	0.076	0.119	0.070
QAll (GPTQ Inf, ours)	0.207	0.390	0.996

Quantizing components separately. As described in Appendix A, quantization can be alternatively applied to 1) RoSA base weights, 2) RoSA gradient accumulators, and 3) the final model. In addition, the summarization model can also be compressed. Table 4 shows how quantizing each component affects the final accuracy. Results show that (a) quantizing most components individually only marginally affects the results (especially in terms of MAUVE score); (b) quantizing the inference model with bitsandbytes (BNB) [27] significantly downgrades the accuracy, while (c) GPTQ quantization [28] recovers comparable accuracy to the original. Applying GPTQ to this setting is only possible due to our new implementation, described in Appendix A, which allows for merging 4-bit weights with half-precision adapters without materializing a full model.

Fully-quantized Panza. We apply quantization to every component at the same time and report the accuracy of the model in Table 4. These results show that the limitation of training using under 15GB of GPU memory results in a 5.8-point drop in BLEU score but unchanged ROUGE and MAUVE scores, suggesting a slightly worse but still useful model, relative to training in BF16 precision.

B. Panza as a Google Chrome Extension

In addition to the entire personalization framework that Panza provides via finetuning, we also focus on how the end user will make use of their personalized email agent. Specifically, we want to integrate Panza directly into the user's GMail experience such that they can use the agent directly within their typical email workflow. We address this by creating a simple, *local*, FastAPI application that is used to serve the user's Panza model. This is complimented by a *publicly available* Google Chrome Extension which will to coordinate the interaction with the email agent from the user's perspective. To use Panza within GMail, the user first describes the prompt that they wish to send to the Panza model in the typical message dialog. Then, upon clicking the Panza button in the menu bar, the Google Chrome Extension sends a request to the FastAPI service, which in turn prompts the underlying model to generate an email response. This response send back to the Google Chrome Extension which is then used to populate the email draft in the place of the prompt that had been written by the user. The workflow is illustrated in Figure 4.

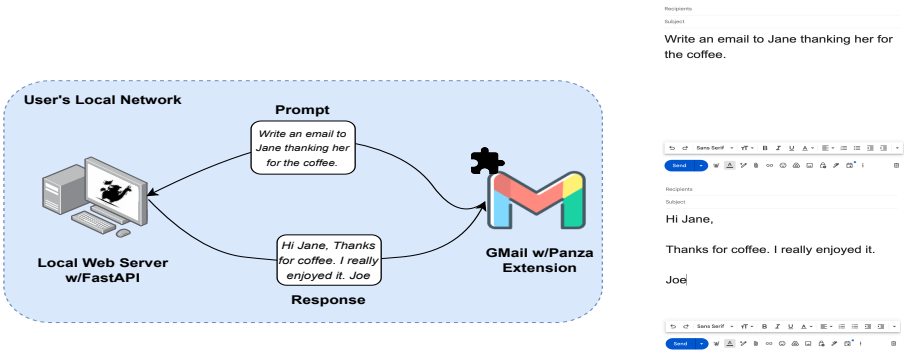


Figure 4: Panza delivered as a Google Chrome Extension

C. Base model comparison

We validate the usefulness of several popular open-source foundation models the base model for Panza, as well as our statement that fine-tuning, rather than simply RAG, is necessary to obtain a performant Panza model. In Figure 5 we show that similar performance levels can be obtained by fine-tuning various LLM backbones, when performing FFT or RoSA-RAFT across Mistral-7B-Instruct-v0.2, Llama3-8B-Instruct, and Phi-3-mini-4k-instruct. While the achieved BLEU scores are very similar across models, the only significant difference is the higher MAUVE score achieved by the Llama3 model.

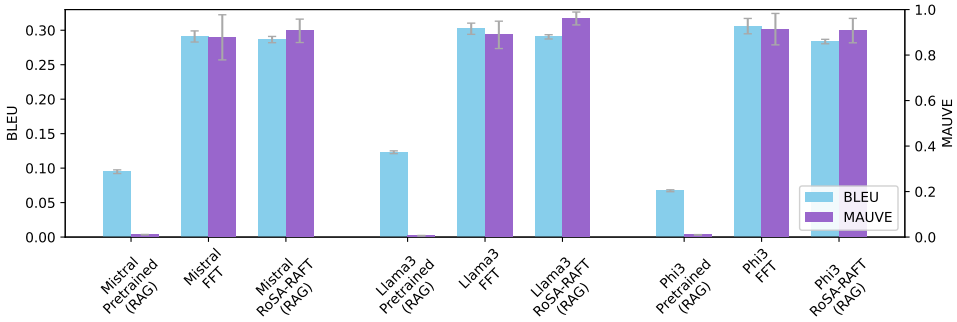


Figure 5: Comparison between models for user Anon1. All models – `Mistral-7B-Instruct-v0.2` (Mistral), `Meta-Llama-3-8B-Instruct` (Llama3), and `Phi-3-mini-4k-instruct` (Phi3) – incorporate the user's style after fine-tuning and outperform the Pretrained + RAG baseline. While BLEU scores are the same across models, MAUVE scores are higher on the Llama3 model.

D. Prompt Engineering

In the first phase of finetuning, we generate summaries with the help of the following **summarization prompt**:

```
"""Summarize the following email that I wrote, in an imperative form, in one or two
or maximum three sentences, and make sure to include relevant information, without
copying the email content itself. The summary should look like an instruction
directing someone to write the same email, and start with Instruction:
Here is the email text:
{email}"""
```

Then, to generate emails, we give the instructions back to the model using the following format:

```
"""
{system preamble}

{user preamble}

{rag prompt}    # [optional]

Instruction: {instruction}
"""
```

The **system preamble** sets the role of the LLM as follows:

```
"""Your role is that of a helpful automated email assistant. I will provide you
with a short instruction, and you have to write a well-formed email in my style
following this instruction. Be sure to follow my email writing style! In case you see
a nonsensical instruction, you should not reply with an email, but with the expression
"Sorry, but I don't get it." """
```

The **user preamble** provides optional information about the user. For the five users in our experiments, we set it to "My name is `FirstName LastName`". Generally, it can be filled with any relevant information about the user, for instance:

```
"""My name is Jane Doe. I work as a manager at Acme Corp. My address is 123
Main Street, Springfield, IL, USA. My boss's name is Alex Burns. My children's
names are Elsa, Anna, and Olaf. I am deeply committed to my hobby of underwater
basket weaving, for which we meet every Thursday at noon."""
```

Finally, for RAG we retrieve several relevant emails for the current instruction and include the following **rag prompt** to the input:

Table 5: Summarization quality relative to "golden" user-generated instructions, across different models (10 seeds).

	David		Anon1	
Method	BLEU	ROUGE	BLEU	ROUGE
Mistral-7B-Instruct-v0.2	0.219 ± 0.009	0.353 ± 0.007	0.133 ± 0.009	0.251 ± 0.012
Mistral-7B-Instruct-v0.3	0.229 ± 0.021	0.373 ± 0.021	0.164 ± 0.014	0.307 ± 0.016
Phi-3-mini-4k-instruct	0.268 ± 0.020	0.422 ± 0.018	0.180 ± 0.006	0.331 ± 0.011
Meta-Llama-3-8B-Instruct	$\mathbf{0.394} \pm 0.011$	$\mathbf{0.548} \pm 0.013$	$\mathbf{0.264} \pm 0.011$	$\mathbf{0.411} \pm 0.015$

"""Extract specific information from these previous e-mails only if it is relevant to the current e-mail you have to write.

Previous e-mails:

EMAIL CONTENT:
<email_1 content>

– – –

EMAIL CONTENT:
<email_2 content>

– – –
..."""

E. Summarization

In Table 5 we measure the summarization quality by comparing against golden summaries from the users David and Anon2. In Table 6, we inspect several generated summaries for each model. The golden summaries used in this study were performed by the data donors, without compensation.

Table 6: Summarization comparison between "golden" user-generated instructions and summaries generated by different pre-trained LLMs.

Original email	**Golden summary**	**Llama-3-8B-Instruct**	**Mistral-7B-Instruct**	**Phi-3-mini-4k-instruct**
Hi Cheryl, The proposal looks good, maybe we can discuss it in a bit more detail tomorrow. Best regards, David	Write to Cheryl saying that the proposal looks good and suggest a meeting the following day.	Write an email to Cheryl, stating that the proposal looks good and suggesting a more detailed discussion for the next day.	Write an email to Cheryl suggesting a meeting to discuss the proposal in more detail tomorrow. End with a polite closing.	Write an email to Cheryl, acknowledging the good proposal, suggesting a detailed discussion for the following day, and signing off with your name.
Hi both, I'm fairly available in the afternoons next week, except for Monday and Friday, so please feel free to suggest a slot. Best, David	Write an email to two recipients, stating that I am available in the afternoons next week, except for Monday and Friday, and ask them to suggest a time that works for them.	Write an email to both parties, stating that you are available in the afternoons next week, excluding Monday and Friday, and inviting them to suggest a suitable time slot.	Write an email informing the recipients that you are generally available in the afternoons from Tuesday to Thursday next week, except for Mondays and Fridays. Request them to suggest a suitable time.	Write an email informing the recipients that you are generally available in the afternoons next week, except for Monday and Friday, and invite them to propose a suitable time slot for a meeting.
Dear Ernie, Thank you for the invitation. I would have loved to join, but I'm trying to save some time and energy for my other admin tasks this semester :-) Best regards, David	Write an email to Ernie, thanking him for the invitation, but saying that I have to decline due to a heavy admin burden this semester.	Write an email to Ernie declining an invitation, citing a need to prioritize other administrative tasks this semester.	Inform Ernie that you're unable to attend due to prior commitments with administrative tasks, expressing gratitude for the invitation.	Write an email to Ernie, expressing gratitude for an invitation, politely declining due to prior commitments related to admin tasks, and sign off with a warm closing.
Dear David, I was wondering if I could ask you for an "emergency-mode" code review (see attached). Specifically, it would be great if you could have a look at it in the next couple of weeks. I'd really appreciate it if you can make it; please let me know. Cheers, David	Write an email to David requesting an "emergency-mode" code review, politely asking them if they can review it within the next couple of weeks.	Write an email to David requesting an "emergency-mode" code review, attaching the relevant document, and asking him to review it within the next couple of weeks.	Ask David for an urgent code review, requesting him to look at the attached document within the next couple of weeks and kindly express appreciation for his assistance.	Write an email to David requesting an emergency-mode code review, emphasizing the need for it to be completed within the next couple of weeks, and express appreciation for his potential assistance, while also asking for his confirmation of availability.

Table 7: Generation comparison for `Llama-3-8B-Instruct` model fine-tuned for different users.

Instruction	Llama-3-8B-Instruct (Pretrained)	David	Kay	Sara	Tana
Write an email to two recipients, stating that I am available in the afternoons next week, except for Monday and Friday, and ask them to suggest a time that works for them.	Dear [Recipient 1] and [Recipient 2], I hope this email finds you well. I wanted to touch base with you regarding my availability for a meeting next week. I am free in the afternoons, from Tuesday to Thursday, and would be happy to schedule a meeting with you during this time. Please note that I am not available on Monday and Friday, as I have other commitments. If you could let me know a time that works for you, I would greatly appreciate it. Best regards, [Your Name]	Hi, I am available in the afternoons next week, except for Monday and Friday. Could you please suggest a time that works for you? Best regards, David	I'm available in the afternoons next week, except for Monday and Friday. How about Tuesday or Wednesday? Please let me know what works for you. Thanks, Kay	I am available in the afternoons next week, except for Monday and Friday. Can you both let me know when you are available and we'll set something up.	I am available in the afternoons next week, except Monday and Friday. What time works for you?
Write an email to the group proposing to resume meetings tomorrow at 11:30 AM, and also mention your availability in the office on Thursday afternoon and Friday.	Dear team, I hope this email finds you well. As we've been on a brief hiatus, I'd like to propose that we resume our regular meetings tomorrow at 11:30 AM. I believe it's essential to catch up and discuss ongoing projects, and I'm confident that this will help us stay on track. Additionally, I wanted to let you know that I'll be available in the office on Thursday afternoon and Friday. If any concerns need to meet or discuss any matters, please don't hesitate to reach out to me. Looking forward to seeing you all tomorrow at 11:30 AM. Best regards, [Your Name]	"Hi all, We should probably resume the meetings tomorrow at 11:30 AM. Also, I am available in the office on Thursday afternoon and Friday. Best, David"	I'm thinking we should resume meetings tomorrow at 11:30 AM. I'm in the office tomorrow afternoon, and I'll be here Thursday afternoon, and Friday. I hope this helps. Kay	I'm going to propose that we resume our meetings tomorrow at 11:30 AM. I'll be in the office tomorrow and Thursday afternoon. I'll be out of the office on Friday. Please let me know if you have any questions. Thanks, Sara	I think we should try to resume our meetings tomorrow at 11:30 AM. I'll be in the office tomorrow afternoon, Thursday afternoon and Friday.
Write an email to Jeannie, thanking her for the monitor, and then request assistance in gaining access to the conference room after hours, as your current ID does not provide access.	Dear Jeannie, I wanted to take a moment to express my gratitude for the new monitor you provided for me. It has been a huge help in my work and I appreciate your support in keeping everything up to date. On a separate note, I was wondering if you could assist me with gaining access to the conference room after hours. As you may know, my current ID does not provide access to the room outside of regular working hours. I often find myself needing to work late [...]	Hi Jeannie, Thanks for the monitor! Unfortunately, my current ID does not provide access to the conference room after hours. Could you please help me with this? Best regards, David	Thanks very much for the monitor. I really appreciate it. I'm having some trouble getting into the conference room after hours. I've got a new ID, but it doesn't seem to work. Could you help me out? Thanks a bunch. Best, Jeff	Jeannie: Thanks for the monitor. I was wondering if you could help me with something. I need to be able to access the conference room after hours and my current ID does not provide access. Can you help me with this? Thanks, Sara	Thanks for the monitor, Jeannie! It is a great help. Now, I need your help with one more thing. My new ID doesn't seem to give me access to the conference room after hours. Can you help me out with this? Thanks!

F. Panza Generated Emails

Table 7 compares emails generated by models fine-tuned for different users, highlighting their style differences.

G. Details of general knowledge measurement

All results are obtained through the corresponding *lm-evaluation-harness* framework [29], and are reported in Table 8. Based on the results collected for `David`, we conclude that personalized Panza models retain, on average, over 98% of the baseline model's accuracy. This suggests that the Panza personalization process avoids catastrophic forgetting.

Table 8: Performance of personalized Panza models on general knowledge benchmarks.

	Llama-3-8B Instruct	FFT	LoRA	RoSA
Arc-c	60.7	60.3	60.7	58.9
MMLU	67.1	65.5	65.6	65.3
Hellaswag	78.5	78.7	78.6	78.2
Winogrande	74.5	74.3	74.9	73.2
GSM8k	68.7	70.2	69.1	68.4
TruthfulQA	51.6	50.9	51.4	50.7
Average	66.9	66.7	66.7	65.8
Recovery (%)	100	99.7	99.7	98.4

H. Comparison to LongForm

We compare Panza to the results of Köksal et al. [1], which introduces Reverse Instructions and tests this method on various forms of long-form data outputs, including emails from the Enron dataset. We note that this is not a fair comparison, as the LongForm models are trained from a non-instruction-tuned model, are finetuned on a dataset that is only about 2% emails, and make no attempt at personalization (the emails come from multiple authors). Further, we note that the specific evaluation subset of the Enron emails differs in our evaluations, and it is not possible to bridge this gap (i.e., evaluate Panza on the LongForm Enron evaluation set), as Panza requires the availability of additional emails from the same author for creating a personalized model. We therefore present these numbers only as an indication that the overall combination of instruction-tuned base model + Panza performs better than approaches that also use Enron emails and Reverse Instructions to create long-form content. Our results in Sections I and C demonstrate the superiority of Panza over base models + RAG.

Table 9: Comparison of METEOR scores of Panza emails to LongForm [1]. All Panza models are trained using the RoSA-RAFT-RAG method with default hyperparameter values.

LongForm	Panza				
Enron Avg.	Avg.	Tana	Sara	Kay	Jeff
8.9	36.3	40.4	38.6	33.8	31.3

We present the results in Table 9. The Panza-RAFT-RAG models achieve an average METEOR score of 36.3, while Köksal et al. [1] report a METEOR score of 8.9 on their best model on this task (LongForm-OPT-6.7B). We emphasize again that, due to methodological differences, this is not a fair comparison, but nevertheless believe that these results demonstrate the effectiveness of Panza on the task of email generation.

I. Comparison to GPT-4o

While the setting of uploading a user's email history to a client such as ChatGPT does not meet our privacy requirements, it is still valid to ask whether a more capable LLM equipped with the user's entire email history (i.e., the training set) would perform as well as Panza. We tested this on users Anon1 and Anon2 by using the OpenAI GPT4o [30] API to generate emails after being prompted with the e-mail generation prompt and the user's entire email history. The results are shown below for the two Anon users. For both, the BLEU score was higher when using Panza with RoSA, and the Mauve (personalization) scores of the OpenAI outputs were near 0, showing that the e-mail style was not personalized to the user. The Panza results are also presented below for easier comparison.

Table 10: Comparison of BLEU and MAUVE scores of Panza emails to those generated by GPT-4o prompted with the user's entire email history. All Panza models are trained using the RoSA-RAFT-RAG method with default hyperparameter values.

	BLEU		MAUVE	
User	GPT-4o	Panza	GPT-4o	Panza
Anon1	0.24	0.31	0.009	0.96
Anon2	0.19	0.22	0.004	0.93

J. Alternate metrics: METEOR, BERT-score

In addition to BLEU, ROUGE, and MAUVE, which we present in the main part of the paper, we considered two alternative metrics: BERT-score [31] and METEOR [32], using the Huggingface implementations for both with default hyperparameters. We use the `distilbert-base-uncased` model for computing the BERT embeddings.

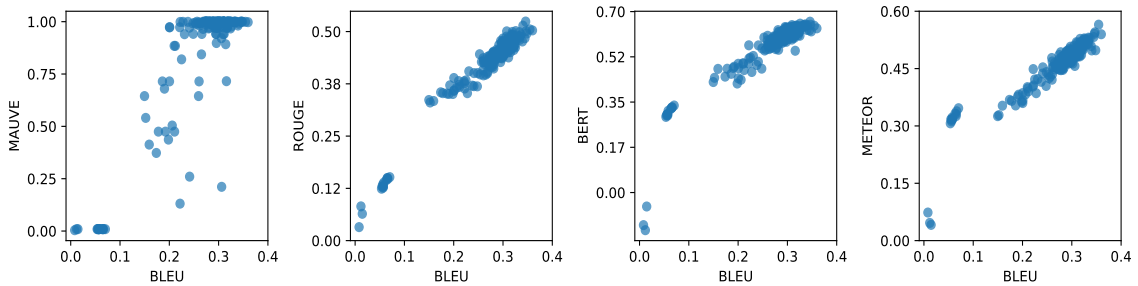


Figure 6: Similarity of BLEU to MAUVE, ROUGE, BERT-score, and METEOR for finetuned models. Note the near-perfect correlation between BLEU and ROUGE, BERT, and METEOR. By contrast, while BERT and MAUVE have some correlation, we observe models with low BERT scores and high MAUVE scores, and vice-versa. Models trained with RoSA, LoRA, and FFT are presented together; RAG is not used for inference in this evaluation.

Table 11: METEOR and BLEU scores across users and models. While METEOR scores for (un-finetuned) Llama models are similar to those for FFT and LoRA models for 4/5 users (Jeff, Tana, Kay, and Sara), we observe large differences for all users in the BLEU scores. Note that finetuned models (both RoSA and FFT) were preferred by users in our study.

	METEOR			BLEU		
User	Llama+RAG	FFT-RAFT-RAG	LoRA-RAFT-RAG	Llama+RAG	FFT-RAFT-RAG	LoRA-RAFT-RAG
Jeff	0.32	0.31	0.32	0.12	0.18	0.19
Tana	0.39	0.42	0.43	0.16	0.28	0.29
Kay	0.34	0.34	0.35	0.12	0.19	0.21
Sara	0.40	0.41	0.41	0.15	0.26	0.28
David	0.42	0.49	0.51	0.11	0.31	0.33

BERT score is frequently used to evaluate text generation quality; unlike BLEU and ROUGE, it uses embeddings in a Large Language Model to compute similarity. While BERT would have also been a good choice to measure the phrasing quality, our experiments found it to be very highly correlated with the BLEU score, and therefore redundant, as we show in Figure 6.

The METEOR score is a variant of token-matching scores such as BLEU and ROUGE, but with several changes, notably using a weighted harmonic mean of the precision and recall, and allowing word synonyms. Like BERT, it is highly redundant with BLEU when used as a finetuning quality metric (see Figure 6). Notably, however, in our experiments, we found that METEOR scores for un-finetuned Llama+RAG models were very close to Panza models in several of the datasets considered. We demonstrate this effect in Table 11. Note the contrast to BLEU, which demonstrates clear differences in non-finetuned versus finetuned models, which also agrees with our results in the human study in Section 5.1, which demonstrate that finetuned models outperform base models in the usefulness of their email generations.

We conjecture that this is due to METEOR allowing of synonyms when doing the token matching, which therefore does not take into account the specific word choices typical to someone's style, and thus prefer the BLEU score for our evaluation and hyperparameter tuning.

K. Additional Human Studies - BLEU/MAUVE Validation

We conducted a human study to validate our hypothesis that models with high BLEU and MAUVE scores are better suited for the e-mail generation task than those with both of these scores low, or with one low and the other high.

As shown in Figure 6, during the course of hyperparameter tuning, we were able to find models that performed relatively well in BLEU, MAUVE, neither, and both. We selected eight of these models, roughly corresponding to each of these four cases (Both low, High BLEU/low MAUVE, low BLEU/high MAUVE, both high) for additional human evaluation. The donor of the dataset was asked to evaluate, on the same 3-point scale described in Appendix L.1, with 3 being the highest score and 1 being the lowest. The results, superimposed

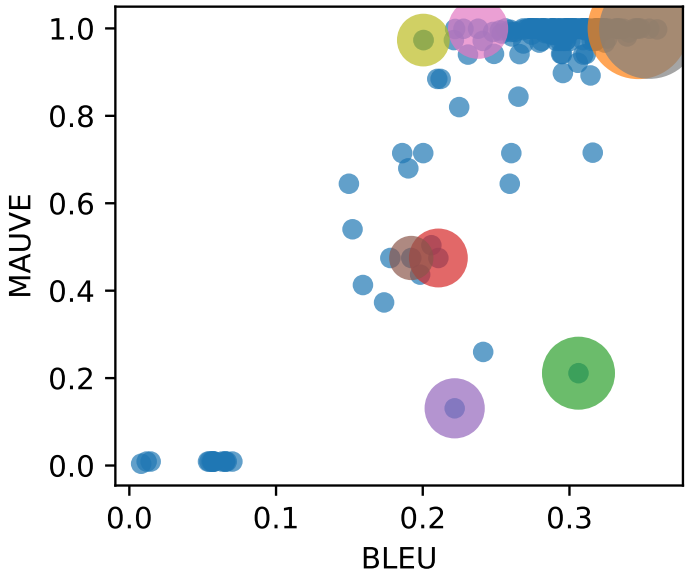


Figure 7: Human-evaluated usefulness of emails output by various Panza models, plotted against their BLEU and MAUVE score. The size of the colored circles corresponds to the human-rated usefulness of the model; bigger signifies more useful.

onto the BLEU/MAUVE scores are shown in Figure 7. We observe that the two models chosen for the highest BLEU and MAUVE scores are found to be the most useful, with average scores of 2.6. The model with high BLEU but lower MAUVE (green circle) was found to be less useful with a score of 2.125, indicating that generally the output emails required additional editing. Models with lower BLEU scores, or lower BLEU and MAUVE scores all had average ratings of 1.83 or below, with the lowest performing model (yellow circle) having a score of 1.36. We interpret these findings to suggest that, to maximize performance, both a (relatively) high BLEU and a high ROUGE score is necessary, and achievable.

This human study took the author of the David email dataset around 30 minutes, and the author was not compensated for this time.

L. Human Study Details

Subject recruitment In all user studies, due to the sensitive nature of the data, subjects were selected for their technical ability to conduct all steps of the evaluation - model training, evaluation data extraction, and the evaluation - without direct assistance from the experimenters, and therefore without exposing the experimenters to the subjects' private data. Thus, it was not possible to use any evaluation services for this task, and instead six subjects were recruited directly as volunteers. All subjects hold technical degrees, and were therefore well-suited for the task. The active time required to complete the tasks was 1-2 hours for each subject, ignoring passive tasks such as waiting for the email data export, and model training. Furthermore, each participant in the user study had between 199-979 emails in their training dataset; this was to ensure that the models had enough signal during the fine-tuning procedure to produce effective personalization, without overfitiing to the specifics of a few email samples. We note that the human rating tasks, after generating the necessary data for each task, took the subjects 30 minutes each, and the subjects were not compensated for their involvement in the user study.

L.1. Additional Details for the Usefulness Study

As Panza is intended to convert relative prompts into long-form text, scores such as BLEU, and METEOR, that seek to match Panza-generated responses to those created by actual humans misses the nuance, that there is no

one correct way to write a given email, and so we do not expect anything close to a perfect match between the Panza output and the email actually written by the user. Therefore, we conducted a human evaluation to measure the direct usefulness of the Panza models. In this evaluation, two subjects were asked to train Panza models on their own e-mail data, then evaluate the outputs of this model, as well as the outputs of Llama3-8B-Instruct augmented with RAG. Below, we detail the study and provide results from the evaluation in more detail.

Data preparation The subjects were instructed to export emails from their provider of choice, with the loose guidance that it was not necessary to obtain more than 1000 emails. The subjects then ran a script that split the data to create a held-out validation set, then finetuned Panza-FFT-RAG and Panza-RoSA-RAG models from a `Llama3-8B-Instruct` checkpoint, using default hyperparameters and no additional tuning. Once the models were prepared, the subjects ran a provided script that, in the first experiment, generated Panza responses (emails) for 16 hand-constructed prompts shown in Table 1, and, in the second experiment, randomly selected 16 prompts from the user's held-out set and likewise generated Panza responses for these emails. Users were instructed to run every script once, and only the first generation (single attempt) was allowed for each prompt per model. In addition to Panza outputs, the subjects also extracted the same responses for an un-finetuned Llama3-8B-Instruct checkpoint augmented with RAG. In all cases for each user, the RAG database was the same, and did not contain the held-out emails.

The resulting prompts and responses(emails) were output in the form of a `.csv` file, and the subjects were instructed to import the file into Google sheets and conduct the evaluation there. Some guidance was provided on ways to format the spreadsheet for easy reading.

Rating instructions The subjects were asked to rate the output emails on a 3-point scale. Specifically, they were given the following instructions:

Please rate the email outputs. The rating scale is as follows. Please use your own best judgment for what exactly minor/nontrivial changes mean to you:

- 3: Very useful: would send as-is or with minor changes
- 2: Moderately useful: would sent only after nontrivial changes
- 1: Not useful: needs a complete rewrite

The ratings were then averaged for the final score. We present the 16 hand-constructed prompts, as well as the ratings each subject gave their models on this task, in Table 1. For privacy reasons, we are not able to provide the Panza outputs for these prompts, nor any of the User-specific prompts or their outputs.

L.2. Additional Details for the Persona Recognition Study

Following the setup that was conducted for the usefulness study in Appendix L.1, each user was asked to perform inference on a further set of prompts to generate the responses necessary for the recognition study. In particular, each participant used their Panza-RoSA-RAG model and performed inference on a set of 24 *custom prompts*, which were selected by the organising team[3]. As before, users were instructed to run this script once, and only the first generation (single attempt) was allowed for each prompt per model. The resulting prompts and responses(emails) were output in the form of a `.csv` file, and the subjects were instructed to share this file

[3]It is worthwhile to note that these prompts were written such that they would not elicit any sensitive information from the user.

Table 12: Human evaluation of Panza usefulness, aggregate results for each subject.

	Subject 1		Subject 2		Subject 3		Subject 4		Subject 5		Subject 6		
	Fixed	Own	Fixed	Own	Fixed	Own	Fixed	Own	Fixed	Own	Fixed	Own	Avg.
Llama + RAG	2.00	2.13	2.25	2.25	2.38	2.69	1.38	1.19	1.69	1.50	2.94	2.81	2.10
RoSA + RAG	2.50	2.56	2.13	2.81	2.56	2.81	2.94	2.81	2.88	2.88	2.50	2.38	2.65
FFT + RAG	2.63	2.75	2.00	2.75	1.81	2.62	2.88	2.94	2.69	2.69	2.19	2.75	2.56

Table 13: Panza performance on prompts from the fixed-prompt dataset.

	Subject 1			Subject 2			Subject 3			
Prompt	**Llama**	**FFT**	**RoSA**	**Llama**	**FFT**	**RoSA**	**Llama**	**FFT**	**RoSA**	Average
Write an email to tell my two good friends Ina and George that I won't be able to make it to dinner on friday. Make sure to add an apology and say I hope to see them soon.	2	3	2	3	1	1	3	2	1	2.00
Write an email inviting a large group of people to dinner at my house. Tell them I'm excited to try out my new recipe for roasted fish.	2	2	3	1	1	1	2	2	3	1.89
Write an email to Ziqui Chen telling her that I am still interested in buying the microwave from her, and I could pick it up next Thursday or Friday.	3	3	3	3	3	3	3	2	3	2.89
Write an email to a travel agency, telling them that you and your parents would prefer Australia over Myanmar, and would like to have modern accomodations with good showers.	2	3	3	2	1	1	1	2	3	2.00
Write an email to my good friend Alice asking her if she has any recommendations for what to do in Peru, when I travel there next summer.	1	3	2	1	3	3	3	3	2	2.33
Write an email to my former coworker, apologizing for the late response and telling her that I'd be happy to grab lunch or coffee when she comes to Vienna for ICLR.	2	1	2	2	3	3	2	2	3	2.22
Write an email to James Chen to thank him for his interest in the group, but telling him that it's not clear if any interns can be hired this year.	2	3	2	2	1	2	2	1	3	2.00
Write an email to Anna Karina to tell her that you'd be happy to answer any questions she has about the group, but you will likely be slow to reply for the next few weeks, due to upcoming conference submission deadlines.	3	3	3	2	3	3	3	3	3	2.89
Write an email to the immigration authorities to tell them that I urgently need an appointment to replace my lost residence permit, but unfortunately it seems that there is no availability on their website for the last month.	3	3	2	2	1	2	1	1	2	1.89
Write an email to the university administration to complain that the shuttle bus has been over 10 minutes late every day for the last couple of weeks.	2	2	3	2	2	3	2	1	3	2.22
Write an email to former group member Alex, telling him that we can't find his laptop and asking where he left it.	2	3	3	3	3	3	2	1	2	2.44
Write an email to Katie showing her how to write a bash script that iterates over a directory and appends a string to every file.	2	2	2	3	2	3	3	1	2	2.22
Write an email to my friend Albert with a recipe for banana bread.	2	3	3	2	1	1	2	1	3	2.00
Write an email to my trainee Eric explaining the review process at NeurIPS and how to be a good reviewer.	2	2	2	3	3	1	3	2	3	2.33
Write an email to Peter to tell him my new address.	2	3	2	2	2	2	3	3	3	2.44
Write an email to Leandra to ask her to call me this evening at my phone number.	3	3	3	3	2	2	3	2	2	2.56
Average	2.19	2.63	2.50	2.25	2.00	2.13	2.38	1.81	2.56	2.27

with the experimenting team *without observing* the contents of the file. This was done to protect the integrity of the study as, as will become clear in the next paragraph, users will be asked to associate emails with users.

Once the organising team received the outputs of from each of the four participants in a group, first each of the responses were cleaned and anonymised. To ensure that no personally identifiable information was revealed that would compromise the point of the recognition task (i.e the person's name). Next, the order of responses for each prompt were shuffled randomly, and outputted to a '.csv' file that was sent to each of the four participants for the rating task. In particular, each participant was asked to, for each of the 24 prompts, to assign each response to one of the four possible individuals involved in their study. We show the three confusion matrices for each independent experiment we conducted in Figure 8.

L.3. Additional Details for the Impersonation Study

The purpose of the impersonation study is to assess whether Panza models can produce emails that are believably like those of the user in whose data it was trained. The reason behind comparing both `Panza-L` and `Panza-S` was was that given our findings in Section 4.5, we wanted to go beyond the metrics that we had selected for this work and evaluate whether the *more data efficient* model is, like the large model, able to convincingly imitate the human qualities of the authors text generation. The user's gold output and the outputs from the `Llama-3-8B` models were chosen to be controls in the experiment. Namely, our hypothesis is that all of the authors own responses should be believable, whilst all the outputs of the `Llama` model would display the converse.

Table 14: Panza performance on prompts from the fixed-prompt dataset.

	Subject 4			Subject 5			Subject 6			
Prompt	**Llama**	**FFT**	**RoSA**	**Llama**	**FFT**	**RoSA**	**Llama**	**FFT**	**RoSA**	Average
Write an email to tell my two good friends Ina and George that I won't be able to make it to dinner on friday. Make sure to add an apology and say I hope to see them soon.	1	3	3	2	3	2	3	3	3	2.56
Write an email inviting a large group of people to dinner at my house. Tell them I'm excited to try out my new recipe for roasted fish.	1	2	3	1	3	3	2	2	2	2.11
Write an email to Ziqui Chen telling her that I am still interested in buying the microwave from her, and I could pick it up next Thursday or Friday.	1	3	3	3	3	3	3	3	3	2.78
Write an email to a travel agency, telling them that you and your parents would prefer Australia over Myanmar, and would like to have modern accommodations with good showers.	1	3	3	1	3	3	3	3	3	2.56
Write an email to my good friend Alice asking her if she has any recommendations for what to do in Peru, when I travel there next summer.	1	3	3	1	3	3	3	3	3	2.56
Write an email to my former coworker, apologizing for the late response and telling her that I'd be happy to grab lunch or coffee when she comes to Vienna for ICLR.	1	3	3	2	3	3	3	2	3	2.56
Write an email to James Chen to thank him for his interest in the group, but telling him that it's not clear if any interns can be hired this year.	1	3	3	2	2	3	3	3	3	2.56
Write an email to Anna Karina to tell her that you'd be happy to answer any questions she has about the group, but you will likely be slow to reply for the next few weeks, due to upcoming conference submission deadlines.	2	3	3	3	3	3	3	3	3	2.89
Write an email to the immigration authorities to tell them that I urgently need an appointment to replace my lost residence permit, but unfortunately it seems that there is no availability on their website for the last month.	1	3	3	2	2	3	3	3	3	2.56
Write an email to the university administration to complain that the shuttle bus has been over 10 minutes late every day for the last couple of weeks.	1	3	2	2	3	3	3	1	1	2.00
Write an email to former group member Alex, telling him that we can't find his laptop and asking where he left it.	1	2	3	1	2	2	3	1	2	1.89
Write an email to Katie showing her how to write a bash script that iterates over a directory and appends a string to every file.	1	3	3	1	2	3	3	2	2	2.22
Write an email to my friend Albert with a recipe for banana bread.	2	3	3	2	3	3	3	1	3	2.56
Write an email to my trainee Eric explaining the review process at NeurIPS and how to be a good reviewer.	2	3	3	1	3	3	3	2	1	2.33
Write an email to Peter to tell him my new address.	3	3	3	2	3	3	3	2	3	2.78
Write an email to Leandra to ask her to call me this evening at my phone number.	2	3	3	1	2	3	3	1	2	2.22
Average	1.38	2.88	2.94	1.69	2.69	2.88	2.94	2.19	2.50	2.45

Table 15: Persona recognition evaluation: Precision (P) and recall (R) of Panza persona recognition, across three trials.

Trial No.	Subject 1		Subject 2		Jeff		Llama	
	P	R	P	R	P	R	P	R
1	0.76	0.77	0.79	0.77	1.00	1.00	0.97	0.98
2	0.62	0.62	0.66	0.66	0.71	0.71	0.71	0.71
3	0.54	0.54	0.45	0.45	0.67	0.67	0.94	0.94
Avg.		0.64	0.64		0.78	0.78	0.87	0.87

In addition to training the `Panza-L` model, all users were instructed to their their `Panza-S` model using the RoSA finetuning method with the default hyperparameters as per Section 4.5. As before, users were instructed to run the command for producing the outputs for the impersonation study only once, and only the first generation (single attempt) was allowed for each prompt per model. Finally, the corresponding prompts for each response were removed so that the rater only received the responses themselves. This was to ensure that we simulated a setting where users receive an email, along with no other additional context that could be used to determine the credibility of the email. The resulting responses(emails) were output in the form of a `.csv` file, where all participants were given a chance to clean the outputs before sending their model outputs to organising team, who would then coordinate the experiment with the respective partner.

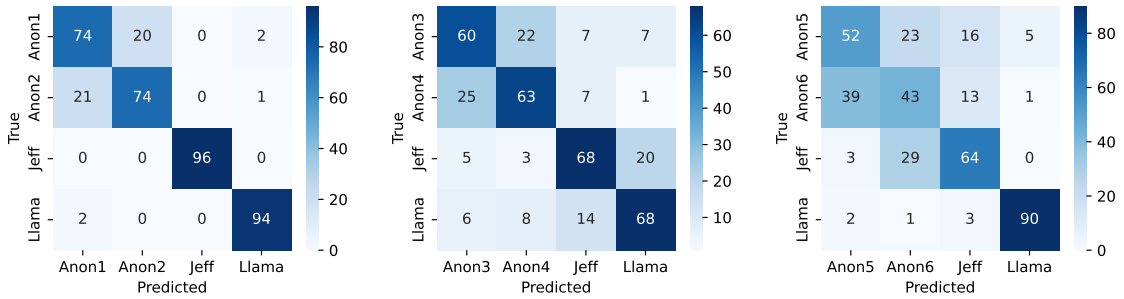


Figure 8: Confusion matrices for three independent experiments of the persona recognition study.

Table 16: Number of authentic and LLM-generated emails found suspicious by acquaintances of the Panza training set authors, results of each subject.

Author	Subject 1	Subject 2	Subject 3	Subject 4	Subject 5	Avg. (%)
Genuine	1/5	2/5	0/5	1/5	2/5	24 %
Llama-3-8B	4/5	3/5	4/5	3/5	2/5	64 %
Panza-L	5/5	1/5	0/5	0/5	1/5	28 %
Panza-S	3/5	2/5	1/5	0/5	2/5	32 %

M. Fine-Tuning Performance

We show complete results for all models, across every user in Tables 17 and 21 (Meta-Llama-3-8B-Instruct), Tables 18 and 22 (Mistral-7B-Instruct-v0.2), and Tables 19 and 23 (Phi-3-mini-4k-instruct). To compare between different models, we report the average results over all users in Table 20. We find that for all the models, finetuning successfully incorporates the user's style. In Figure 9 we perform a pairwise style comparison for model trained on different users and the test emails of all the other users. This shows MAUVE score successfully captures style differences, while BLEU/ROUGE scores are limited to measuring the paraphrasing capacity and can't distinguish style.

Table 17: Results Meta-Llama-3-8B-Instruct across all methods and users.

	David			Jeff			Kay			Sara			Tana		
Method	BLEU	Rouge	MAUVE	BLEU	Rouge	MAUVE	BLEU	Rouge	MAUVE	BLEU	Rouge	MAUVE	BLEU	Rouge	MAUVE
Pretrained	0.083	0.181	0.009	0.108	0.182	0.004	0.113	0.186	0.005	0.144	0.23	0.004	0.132	0.21	0.006
Pretrained-RAG	0.107	0.212	0.017	0.115	0.188	0.005	0.121	0.197	0.004	0.151	0.233	0.004	0.149	0.227	0.005
FFT	0.278	0.46	0.996	0.166	0.282	0.758	0.197	**0.295**	0.863	0.261	0.356	0.859	0.256	0.358	0.859
FFT-RAG	0.3	0.449	0.984	0.166	0.266	0.779	0.19	0.283	0.933	0.242	0.337	0.852	0.238	0.327	0.898
FFT-RAFT	0.299	0.476	0.997	0.164	0.279	0.715	0.192	0.285	0.891	0.253	0.357	0.914	0.266	0.363	0.903
FFT-RAFT-RAG	0.31	0.494	0.985	0.187	0.297	0.824	0.184	0.281	0.941	0.263	0.36	0.826	**0.278**	**0.372**	0.876
RoSA	0.312	0.488	0.999	0.164	0.285	0.806	**0.202**	0.293	0.898	0.26	0.355	0.872	0.26	0.352	0.948
RoSA-RAG	0.236	0.383	0.982	0.166	0.266	0.786	0.184	0.269	0.963	0.243	0.338	0.945	0.253	0.34	**0.97**
RoSA-RAFT	0.321	0.491	0.991	0.166	0.291	0.823	0.201	0.29	0.895	**0.268**	0.36	0.763	0.252	0.349	0.814
RoSA-RAFT-RAG	**0.346**	**0.509**	**1.0**	**0.192**	**0.305**	**0.867**	0.196	0.291	0.973	0.265	**0.362**	0.889	0.277	0.367	0.905
LoRA	0.26	0.401	0.68	0.167	0.27	0.224	0.184	0.269	0.979	0.247	0.351	0.921	0.239	0.329	0.897
LoRA-RAG	0.257	0.406	0.922	0.159	0.253	0.481	0.181	0.267	0.968	0.23	0.327	**0.969**	0.233	0.312	0.942
LoRA-RAFT	0.243	0.391	0.412	0.165	0.267	0.149	0.187	0.276	**0.987**	0.245	0.341	0.906	0.243	0.329	0.893
LoRA-RAFT-RAG	0.246	0.404	0.899	0.167	0.262	0.547	0.185	0.27	0.942	0.252	0.349	0.909	0.262	0.353	0.854

N. Hyperparameter Tuning

N.1. Inference

We perform generation using beam search with a temperature $T = 0.7$, number of top probability tokens to keep $top_k = 50$ and nucleus sampling parameter $top_p = 0.7$.

Table 18: Results `Mistral-7B-Instruct-v0.2` across all methods and users.

	David			Jeff			Kay			Sara			Tana		
Method	BLEU	Rouge	MAUVE	BLEU	Rouge	MAUVE	BLEU	Rouge	MAUVE	BLEU	Rouge	MAUVE	BLEU	Rouge	MAUVE
Pretrained	0.072	0.158	0.009	0.093	0.164	0.005	0.096	0.168	0.006	0.117	0.196	0.004	0.113	0.187	0.007
Pretrained-RAG	0.083	0.181	0.058	0.102	0.17	0.006	0.103	0.176	0.017	0.122	0.2	0.008	0.126	0.201	0.005
FFT	**0.335**	**0.475**	0.997	0.162	0.262	0.263	0.214	0.295	0.795	0.264	0.365	0.839	0.263	0.36	0.719
FFT-RAG	0.276	0.425	0.934	0.162	0.251	0.376	0.2	0.286	0.896	0.236	0.335	0.876	0.255	0.346	**0.872**
FFT-RAFT	0.306	0.461	0.963	0.163	0.262	0.346	0.211	**0.313**	0.696	**0.28**	**0.393**	0.792	0.254	0.35	0.759
FFT-RAFT-RAG	0.309	0.461	0.959	**0.174**	**0.277**	0.29	**0.216**	0.308	0.882	0.275	0.39	0.648	0.268	0.358	0.723
RoSA	0.306	0.459	0.988	0.169	0.258	0.482	0.209	0.302	0.797	0.253	0.358	0.955	0.24	0.33	0.74
RoSA-RAG	0.263	0.416	**1.0**	0.154	0.228	0.517	0.182	0.259	0.894	0.2	0.287	**0.971**	0.238	0.331	0.826
RoSA-RAFT	0.289	0.437	0.999	0.167	0.264	0.557	0.189	0.296	0.809	0.265	0.387	0.838	0.245	0.337	0.856
RoSA-RAFT-RAG	0.289	0.43	0.998	0.171	0.262	0.408	0.204	0.293	**0.961**	0.255	0.37	0.827	**0.273**	**0.365**	0.757
LoRA	0.294	0.444	0.976	0.157	0.247	0.56	0.145	0.229	0.44	0.214	0.318	0.413	0.214	0.318	0.288
LoRA-RAG	0.249	0.397	0.974	0.166	0.25	0.589	0.169	0.257	0.891	0.234	0.336	0.702	0.241	0.336	0.535
LoRA-RAFT	0.298	0.451	0.941	0.158	0.248	**0.681**	0.136	0.217	0.426	0.204	0.3	0.465	0.206	0.303	0.3
LoRA-RAFT-RAG	0.286	0.437	0.991	0.168	0.262	0.569	0.151	0.235	0.417	0.23	0.334	0.471	0.232	0.329	0.329

Table 19: Results for `Phi-3-mini-4k-instruct` across all methods and users.

	David			Jeff			Kay			Sara			Tana		
Method	BLEU	Rouge	MAUVE	BLEU	Rouge	MAUVE	BLEU	Rouge	MAUVE	BLEU	Rouge	MAUVE	BLEU	Rouge	MAUVE
Pretrained	0.055	0.129	0.009	0.077	0.147	0.006	0.078	0.149	0.006	0.101	0.182	0.007	0.093	0.168	0.005
Pretrained-RAG	0.054	0.126	0.009	0.082	0.151	0.007	0.082	0.151	0.012	0.107	0.185	0.009	0.101	0.176	0.009
FFT	0.33	0.488	1.0	0.169	0.281	0.86	0.199	0.299	0.866	0.27	0.379	0.884	0.265	0.364	0.869
FFT-RAG	0.31	0.474	0.994	0.179	0.28	0.726	0.196	0.283	0.858	0.231	0.322	0.959	0.254	0.353	0.911
FFT-RAFT	**0.33**	**0.508**	0.992	0.166	0.276	**0.915**	0.194	0.289	0.923	0.261	0.366	0.881	0.262	0.371	0.877
FFT-RAFT-RAG	0.327	0.495	0.998	**0.181**	0.289	0.824	0.201	0.294	0.862	0.263	0.365	0.865	**0.276**	**0.38**	0.887
RoSA	0.308	0.473	0.996	0.166	0.256	0.85	0.193	0.281	**0.956**	0.241	0.35	0.866	0.245	0.337	0.866
RoSA-RAG	0.302	0.455	1.0	0.171	0.267	0.726	0.171	0.256	0.814	0.195	0.289	**0.982**	0.251	0.339	0.948
RoSA-RAFT	0.315	0.473	**1.0**	0.157	0.257	0.875	0.18	0.278	0.916	0.234	0.343	0.939	0.245	0.331	**0.958**
RoSA-RAFT-RAG	0.319	0.472	0.996	0.173	0.274	0.869	0.188	0.278	0.896	0.238	0.345	0.812	0.265	0.354	0.945
LoRA	0.277	0.439	1.0	0.164	0.289	0.789	0.2	0.302	0.72	0.261	0.384	0.754	0.253	0.359	0.82
LoRA-RAG	0.295	0.461	0.999	0.173	0.283	0.77	0.156	0.246	0.577	0.235	0.353	0.909	0.25	0.347	0.834
LoRA-RAFT	0.288	0.443	1.0	0.16	0.286	0.846	**0.21**	**0.319**	0.852	**0.276**	**0.398**	0.767	0.261	0.366	0.802
LoRA-RAFT-RAG	0.313	0.472	0.999	0.176	**0.3**	0.65	0.209	0.311	0.919	0.269	0.389	0.923	0.274	0.376	0.757

N.2. Fine-Tuning

We perform a thorough hyperparameter tuning for every backbone, and every user over learning rate and number of epochs. For FFT we experiment with $\{1, 3, 5\}$ epochs and learning rates between $[10^{-3}, 10^{-7}]$. For PEFT methods (RoSA, LoRA), we experiment with $\{1, 3, 5, 7, 9\}$ epochs and learning rates between $[10^{-3}, 10^{-7}]$. Next, we present the best configuration found for each model, method and user.

Meta-Llama-3-8B-Instruct For users David and Jeff: FFT for 3 epochs with a learning rate of 10^{-5}; PEFT for 7 epochs with a learning rate of 10^{-5}. For users Kay, Sara and Tana: FFT for 3 epochs with a learning rate of 10^{-4} and PEFT for 7 epochs with a learning rate of 10^{-4}. For users Anon1, Anon2: FFT for 3 epochs with a learning rate of 10^{-5} and PEFT for 7 epochs with a learning rate of 10^{-4}. **Mistral-7B-Instruct-v0.2** For users David and Jeff, Anon2: FFT for 3 epochs with a learning rate of 10^{-5}; PEFT for 7 epochs with a learning rate of 10^{-4}. For users Kay, Sara and Tana: FFT for 3 epochs with a learning rate of 10^{-5} and PEFT for 7 epochs with a learning rate of 10^{-5}. For Anon1: FFT for 3 epochs with a learning rate of 10^{-5}; RoSA for 7 epochs with a learning rate of 10^{-5}, LoRA for 7 epochs with a learning rate of 10^{-4}. **Phi-3-mini-4k-instruct** Same hyperparameters for every users: FFT for 3 epochs with a learning rate of 10^{-4}; PEFT for 7 epochs wth a learning rate of 10^{-4}.

N.3. RAG and RAFT

We use the FAISS library [33] and the `sentence-transformers/all-mpnet-base-v2` model for e-mail retrieval. We set fixed values of the number of retrieval emails $n_{RAG} = 2$ during RAFT, and $n_{RAG} = 3$ at inference with RAG. We pick relatively low values, as every email retrieved increases the input length, thus the memory consumption. We experimented with larger n_{RAG} for the Pretrained baseline, without significantly different results. For RAFT, we use $p_{RAG} = 0.55$ chance to include relevant emails in the prompt. We use the same relevancy threshold $T_{RAG} = 0.2$, tuned on the private users Anon1 and Anon2 to encourage recall rather than precision. This can retrieve irrelevant emails, but prevents missing important items from the user's history. Furthermore, RAFT learns how to better filter irrelevant information in case it is added to the prompt.

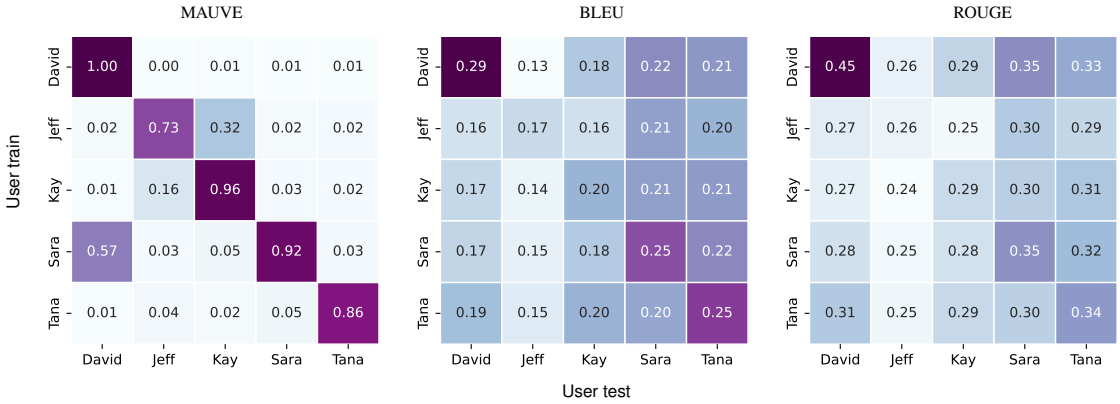


Figure 9: Style comparison between `Phi-3-mini-4k-instruct` models trained for different users.

Table 20: Comparison between models (results average over all users).

	Mistral			Llama-3			Phi-3		
Method	BLEU	Rouge	MAUVE	BLEU	Rouge	MAUVE	BLEU	Rouge	MAUVE
Pretrained	0.098	0.174	0.006	0.116	0.198	0.006	0.081	0.155	0.007
Pretrained-RAG	0.107	0.186	0.019	0.129	0.212	0.007	0.085	0.158	0.009
FFT	0.247	0.351	0.722	0.231	0.35	0.867	0.247	0.362	0.896
FFT-RAG	0.226	0.328	0.791	0.227	0.332	0.889	0.234	0.342	0.89
FFT-RAFT	0.243	0.356	0.711	0.235	0.352	0.884	0.243	0.362	0.918
FFT-RAFT-RAG	**0.248**	**0.359**	0.7	0.244	0.361	0.89	**0.25**	0.365	0.887
RoSA	0.235	0.341	0.792	0.24	0.355	0.904	0.231	0.339	0.907
RoSA-RAG	0.208	0.304	**0.841**	0.216	0.319	**0.929**	0.218	0.321	0.894
RoSA-RAFT	0.231	0.344	0.812	0.241	0.356	0.857	0.226	0.336	**0.938**
RoSA-RAFT-RAG	0.238	0.344	0.79	**0.255**	**0.367**	0.927	0.237	0.345	0.904
LoRA	0.205	0.311	0.535	0.219	0.324	0.74	0.231	0.355	0.817
LoRA-RAG	0.212	0.315	0.738	0.212	0.313	0.856	0.222	0.338	0.818
LoRA-RAFT	0.2	0.304	0.563	0.216	0.321	0.669	0.239	0.363	0.853
LoRA-RAFT-RAG	0.213	0.319	0.556	0.222	0.328	0.83	0.248	**0.37**	0.849

Table 21: Anon users results (`Meta-Llama-3-8B-Instruct`)

	Anon1			Anon2		
Method	BLEU	Rouge	Mauve	BLEU	Rouge	Mauve
Pretrained	0.11	0.207	0.006	0.094	0.18	0.005
Pretrained-RAG	0.123	0.226	0.006	0.108	0.197	0.007
FFT	0.302	0.462	0.889	0.219	0.381	0.909
FFT-RAG	0.232	0.398	0.918	0.167	0.301	**0.976**
FFT-RAFT	0.293	**0.464**	0.959	0.214	**0.383**	0.898
FFT-RAFT-RAG	0.287	0.461	0.95	0.218	0.369	0.895
RoSA	**0.306**	0.459	0.958	**0.22**	0.358	0.927
RoSA-RAG	0.278	0.435	0.975	0.198	0.327	0.934
RoSA-RAFT	0.285	0.436	0.978	0.21	0.348	0.951
RoSA-RAFT-RAG	0.29	0.438	0.961	0.217	0.353	0.939
LoRA	0.3	0.457	0.959	0.208	0.348	0.96
LoRA-RAG	0.245	0.389	0.95	0.2	0.323	0.954
LoRA-RAFT	0.289	0.45	0.965	0.206	0.349	0.956
LoRA-RAFT-RAG	0.288	0.443	**0.985**	0.214	0.353	0.957

Table 22: Anon users results (`Mistral-7B-Instruct-v0.2`)

	Anon1			Anon2		
Method	BLEU	Rouge	Mauve	BLEU	Rouge	Mauve
Pretrained	0.085	0.173	0.011	0.081	0.16	0.005
Pretrained-RAG	0.095	0.188	0.011	0.089	0.169	0.008
FFT	0.291	**0.459**	0.878	0.214	0.362	0.706
FFT-RAG	0.248	0.411	0.824	0.197	0.334	0.812
FFT-RAFT	0.295	0.457	0.877	0.217	**0.367**	0.559
FFT-RAFT-RAG	0.287	0.45	0.825	**0.222**	0.362	0.679
RoSA	0.294	0.459	0.912	0.213	0.35	0.888
RoSA-RAG	0.266	0.422	0.875	0.175	0.292	0.914
RoSA-RAFT	**0.298**	0.458	0.851	0.215	0.347	0.868
RoSA-RAFT-RAG	0.286	0.448	0.907	0.206	0.336	0.832
LoRA	0.275	0.415	0.935	0.203	0.335	**0.948**
LoRA-RAG	0.259	0.401	0.943	0.189	0.314	0.93
LoRA-RAFT	0.28	0.43	0.96	0.203	0.335	0.916
LoRA-RAFT-RAG	0.278	0.429	**0.975**	0.206	0.335	0.901

Table 23: Anon users results (`Phi-3-mini-4k-instruct`)

	Anon1			Anon2		
Method	BLEU	Rouge	Mauve	BLEU	Rouge	Mauve
Pretrained	0.067	0.144	0.009	0.069	0.139	0.006
Pretrained-RAG	0.067	0.145	0.01	0.068	0.139	0.007
FFT	**0.306**	0.468	0.914	0.23	0.383	0.942
FFT-RAG	0.302	0.462	0.888	0.219	0.359	0.939
FFT-RAFT	0.303	**0.477**	0.968	**0.233**	**0.394**	0.943
FFT-RAFT-RAG	0.3	0.473	0.923	0.229	0.374	0.943
RoSA	0.293	0.446	0.926	0.206	0.356	0.924
RoSA-RAG	0.3	0.457	0.927	0.21	0.348	0.927
RoSA-RAFT	0.288	0.449	**0.98**	0.213	0.358	0.941
RoSA-RAFT-RAG	0.284	0.445	0.907	0.215	0.35	0.926
LoRA	0.296	0.468	0.945	0.22	0.38	**0.95**
LoRA-RAG	0.292	0.454	0.886	0.223	0.373	0.935
LoRA-RAFT	0.302	0.467	0.951	0.217	0.386	0.885
LoRA-RAFT-RAG	0.301	0.466	0.945	0.226	0.379	0.924