PEARL: Personalizing Large Language Model Writing Assistants with Generation-Calibrated Retrievers

Sheshera Mysore[1△†] **Zhuoran Lu**[2†] **Mengting Wan**[1] **Longqi Yang**[1]
Bahar Sarrafzadeh[1] **Steve Menezes**[1] **Tina Baghaee**[1]
Emmanuel Barajas Gonzalez[1] **Jennifer Neville**[1] **Tara Safavi**[1△]
[2]Purdue University, IN, USA [1]Microsoft, WA, USA
[△]Corresponding authors: smysore@cs.umass.edu, tarasafavi@microsoft.com

Abstract

Powerful large language models have facilitated the development of writing assistants that promise to significantly improve the quality and efficiency of composition and communication. However, a barrier to effective assistance is the lack of personalization in LLM outputs to the author's communication style, specialized knowledge, and values. In this paper, we address this challenge by proposing PEARL, a LLM writing assistant personalized with a retriever that is trained to be *generation-calibrated* for personalization. Generation calibration ensures that our retriever selects historic user authored documents to augment an LLM prompt such that they are likely to help an LLM generation better adhere to a users' preferences. We propose two key novelties for training such a retriever: (1) A training data selection method that identifies historical user requests likely to benefit from personalization *and* documents that provide that benefit; and (2) A scale-calibrating KL-divergence objective that ensures that our retriever scores remain proportional to the downstream generation quality from using the document for personalized generation. In a series of holistic evaluations, we demonstrate the effectiveness of PEARL in generating long-form texts on multiple social media datasets. Finally, we demonstrate how a generation-calibrated retriever can double as a performance predictor – detecting low quality retrieval, and improving potentially underperforming outputs via revision with LLMs.

1 Introduction

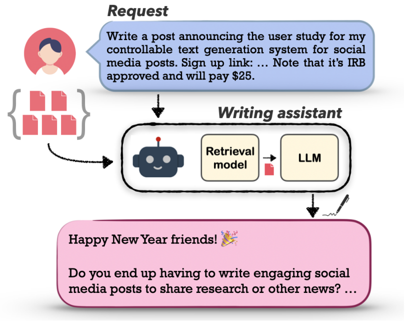

Request
Write a post announcing the user study for my controllable text generation system for social media posts. Sign up link: ... Note that it's IRB approved and will pay $25.
Writing assistant
Retrieval model
LLM
Happy New Year friends! 🎉
Do you end up having to write engaging social media posts to share research or other news? ...

Figure 1: PEARL is a request-driven generation model that personalizes LLM outputs through retrieval augmentation with a *generation calibrated* retriever.

Machine-assisted writing has seen a long history of development, progressing from providing simple syntactic checks, to revising human authored text, to recent assistants being able to fully compose texts on direction from authors (Mahlow, 2023; Dale and Viethen, 2021). The text-generation capabilities of current LLMs and has led current research to explore a new frontier of writing assistants for complex applications such as knowledge synthesis (Shen et al., 2023), peer review (Chen et al., 2023), and journalism (Wang et al., 2023c). An important element of effective writing assistants is being able to personalize generated text to retain the knowledge, style, and values of a user – an essential element of interpersonal communication (Pickering and Garrod, 2013). With current LLMs prone to generating overly generic text (Pu and Demberg, 2023), author personalization of LLMs is an important problem.

Personalizing LLM outputs may be seen as a form of alignment to individual users of the LLM (Kirk et al., 2023). However, leveraging fine-tuning for alignment in a personalization setup poses challenges to serving trained per-user models and obtaining sufficient per-user alignment training data. Therefore, we pursue in-context alignment through retrieval augmentation (Salemi et al., 2023; Li et al., 2023a). First, we assume access to a set of historic *user-authored documents* (e.g. emails, social media posts, etc.) and a user *request* for a personalized generation. To personalize LLM outputs we propose an approach to train a retrieval model that selects historic user documents to augment an LLM's prompt. Historic documents capture

[†] Work done during internship at Microsoft Research.

Proceedings of the 1st Workshop on Customizable NLP: Progress and Challenges in Customizing NLP for a Domain, Application, Group, or Individual (CustomNLP4U), pages 198–219
November 16, 2024 ©2024 Association for Computational Linguistics

users' personal style, knowledge, and values and can serve as useful context for personalized generation. While training retrievers for non-personalized applications have been explored in prior work (Gonen et al., 2022), this exploration has been limited in personalized text generation. Finally, we pursue personalization of LLMs only accessible via prompt-based APIs since this represents a common form of accessing performant large scale LLMs.

The starting point for our retriever in prior work examining effective prompts for *non-personalized* applications: Gonen et al. (2022) show the best prompts to be those with the highest conditional likelihood of generating a target text, and Rubin et al. (2022) use these likelihoods to train retrieval models for non-personalized retrieval augmentation of LLMs. While this approach performs well in non-personalized setups, *personalized* text generation presents unique challenges and opportunities: There are fewer historic documents per user ($\sim$hundreds) than common non-personalized retrieval collections, and user requests may diverge from their history as users' preferences change. A smaller retrieval corpus and shifting interests mean that all requests cannot be satisfied by retrieval from a users' historical documents – as a result, all historic requests and documents are unlikely to be useful for training a retriever. Our first contribution addresses this: We present a novel **difference of likelihoods**-based method that identifies *only* the personalizable user requests and associated documents that are likely to personalize downstream generations, and use these to train our retriever.

Next, the personalization setup offers an opportunity: Fewer historical documents per user permits the use of expressive cross-encoder retrievers instead of scalable but less expressive biencoders commonly used for non-personalized tasks (Rubin et al., 2022). However, cross-encoders produce skewed scores at the ends of their score ranges (Menon et al., 2022; Yadav et al., 2022), hampering their ability to closely track the utility of a document for personalized generation. We remedy this with our second contribution – a **personalized scale-calibrating training objective** (Yan et al., 2022). This ensures that scores from our retriever are *generation-calibrated* for personalization – i.e. the score it produces for request-document pairs is proportional to the output quality of an LLM prompted with the pair. In a case study, we show how generation calibration enables the retriever's scores to be used for *retrieval performance prediction* – detecting low-quality retrievals, and revising potentially low-quality generations.

We instantiate PEARL with multiple LLMs, `davinci-003` and `gpt-35-turbo`, at privacy compliant enterprise API endpoints and evaluate it on a private dataset of workplace communications and a public dataset of Reddit comments. For evaluation, we use a variety of evaluation methods spanning intrinsic, extrinsic, and personalized LLM-as-judge evaluations to demonstrate the value of PEARL. Further, since we train calibrated retrieval models, we present additional evaluations for calibration, ablations, and analysis in Appendices. Our evaluations demonstrate that PEARL consistently matches or outperforms strong baseline approaches.

2 Related Work

Example selection for LLMs Early work on training retrievers for augmenting LLM contexts in non-personalized applications was proposed by Rubin et al. (2022). They train retrieval models by distilling LLM likelihoods of the target completions conditioned on the prompt. Similarly Wang et al. (2023b) train retrieval models on finer-grained feedback from a trained reward model through distillation. More distantly, Zhang et al. (2022) train instances selection models on rewards from a downstream evaluation metric using reinforcement learning. Parallel with our work, Salemi et al. (2024) train bi-encoders for personalized classification and short text generation and find knowledge distillation from downstream LLMs to outperform reinforcement learning based training of retrievers. In this regard, Salemi et al. (2024) and Rubin et al. (2022) are closely related and represent closest work to ours – we compare to such an approach in ablations (Appendix C.2). Despite similarities to our work, all prior work has explored training retrievers for document selection while assuming that satisfactory predictions can be made for *all* inputs/requests. In addition to selecting documents for training, we also select training requests that benefit from retrieval augmentation – a necessity in personalization where retrieval is performed over a smaller historical document set instead of a large shared corpus. Further, no prior approaches explore calibration for retrievers and their ability to identify low-quality retrievals, and selectively revise LLM outputs – we explore this. Appendix D discusses additional work on optimizing prompts, robustness

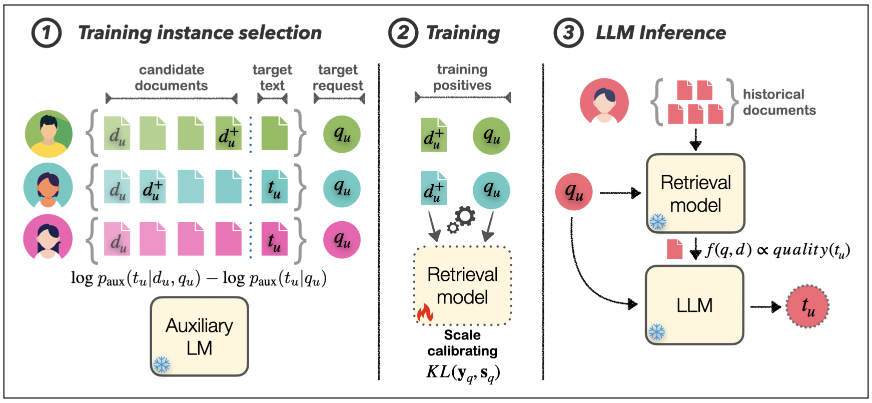


Figure 2: To train retriever, f_{retr}, an auxiliary language model is first used to identify historical *requests* that can be personalized and the best *document* to use for personalization ①. Then, f_{retr} is trained on the selected data with a scale calibrating loss function ②. Given an unseen request, f_{retr} is used to select the best instances from historical texts for augmenting an LLM prompt for personalized generation ③. Our training results in a generation calibrated retriever where scores for documents are proportional to the quality of the LLM output.

to prompt errors, and calibrated retrievers.

Personalized writing assistants While writing assistants have seen considerable exploration, only some prior work has focused on author personalization. These applications range from email (Chen et al., 2019; Trajanovski et al., 2021), to social media (Gero et al., 2022), and grammatical error correction (GEC) (Nadejde and Tetreault, 2019). These systems commonly leverage nearest-neighbor models (Chen et al., 2019; Trajanovski et al., 2021) and user-group level parameter-efficient fine-tuning for personalization (Nadejde and Tetreault, 2019). In contrast, we explore retrieval models for in-context alignment/personalization with LLMs. Parallel work has also explored personalized writing with LLMs. Li et al. (2023b) construct prompts with pre-trained retrieval and summarization models and fine-tune an LLM for personalized completion. Follow-on work has explored training a prompt-re-writer to tune prompts for a fixed LLM (Li et al., 2023a). Prompt re-writing is a complementary approach to a trained retriever, with future systems likely to benefit from both. Appendix D discusses non-personalized writing assistants and reader personalization.

3 Problem Definition

We consider a request-conditional, personalized text generation task. As input to the system, we assume a **user** u who is associated with a set of N_u **historical documents** $\mathcal{D}_u = \{d_u^{(i)}\}_{i=1}^{N_u}$, where each document d_u may be a previously-authored social media post, email, etc. The user u is further associated with a textual **request** q_u submitted to the writing assistant. The request may be authored by the user or constructed from the task context. Explicitly authored requests are increasingly common in conversational LLM interfaces (Papenmeier et al., 2021), and task contexts may be seen as implicit requests e.g. email prefixes that require completion (Chen et al., 2019). Finally, we assume access to a **large language model** f_{LLM} available via a prompt-based text generation API.

Given $\mathcal{D}_u$, q_u, and f_{LLM}, our retriever, f_{retr} is trained to select a subset of historical documents $\mathcal{D}'_u \subset \mathcal{D}_u$ as few-shot examples for the LLM. Then the LLM generates a **target text** t_u of up to 300 words: $t_u = f_{\text{LLM}}(\phi(q_u, \mathcal{D}'_u))$, where ϕ is a prompt construction function that inputs the user's request and retrieved historical documents, t_u reflects the style, knowledge, and values of u.

4 Proposed Approach

We present PEARL, an in-context aligned LLM-based model for personalized writing assistance. Our approach (Figure 2) consists of an offline retriever training stage and an online LLM inference stage. Offline, we train a **retriever** $f_{\text{retr}} : (q_u, d_u) \rightarrow \mathbb{R}$ that scores the user's historical documents for their ability to personalize the output for a user request. Further, we ensure that f_{retr} is generation calibrated i.e. the scores it produces for (q_u, d_u) pairs are proportional to the quality of the generated text from using (q_u, d_u) in a prompt. We train such a retriever through two key novelties: (1) Training data selection based on a novel difference of likelihoods from an auxiliary text generation model – we identify requests which benefit from personalization *and* documents which likely help personalize a target, and (2) A scale-calibrating training objective which ensures that retrievers closely track the benefit of request-document pairs for generation. Given a new request, our LLM is prompted to generate a target text t_u conditioned on the request and the documents retrieved by f_{retr}. Next, we describe the retriever training set construction (Algorithm 1), how we optimize the retriever, and the details of our implementation.

4.1 Training Data Setup

To optimize f_{retr} for a personalized text generation task, we carefully create a training set for f_{retr} from historical user documents by using an **auxiliary text generation model** f_{aux} to identify which requests and documents will help to personalize the generation of a target text.

Data organization We organize the training data to create a setup close to the problem defined in §3. Given a set of M users and their historical document sets $\{\mathcal{D}_u\}_{u=1}^{M}$, for each user u we partition $\mathcal{D}_u$ into two non-overlapping sets, a candidate document set $\mathcal{D}_u^c \subset \mathcal{D}_u$, and a "target" text set $\mathcal{D}_u^t \subset \mathcal{D}_u$, such that $\mathcal{D}_u^c + \mathcal{D}_u^t = \mathcal{D}_u$. The partitioning is done temporally, i.e. the target texts occur after the candidate documents, mimicking the personalization scenario where past texts are used to personalize later targets. If time data isn't available, the partitioning may be done randomly.

Next, for each target text t_u in each users $\mathcal{D}_u^t$, we pair the text with a corresponding request q_u. For training, requests may be naturally present in the data, e.g., email prefixes that require completion (Chen et al., 2019), or they may be generated synthetically (Bonifacio et al., 2022). We detail request generation in §5.1.

Auxiliary model scoring Next, we use the auxiliary text generation model f_{aux} to score each candidate document in $d_u \in \mathcal{D}_u^c$ for producing the personalized t_u corresponding to the q_u for each $(q_u, t_u) \in \mathcal{D}_u^t$. We define the score as a difference in the likelihood, per f_{aux}, of the target given the request with and without the historical document:

$$y_{q_u}^{d_u} = \log p_{\text{aux}}(t_u|d_u, q_u) - \log p_{\text{aux}}(t_u|q_u), \quad (1)$$

Importantly, Eq. (1) is highest when the request is suitable for personalization *and* the candidate document is the "right" example for personalization. That is, the request alone is not sufficient for generating the target text (i.e., the quantity defined by the second term is lower), and this candidate document is particularly beneficial to generation (i.e., the quantity defined by the first term is higher). Finally, we assume model f_{aux} to be smaller than f_{LLM} to support efficient creation of training data, and that we have access to its token likelihoods. Appendix A shows prompts used for f_{aux}.

4.2 Training Data Selection

We use the scores from Eq. 1 to identify: (1) a subset of training requests that are likely to benefit from personalization; and (2) candidate documents that are likely to benefit those requests i.e. positive training documents.

Request selection Using Eq. 1, we score all request-target pairs of a user in $\mathcal{D}_u^t$ against all of their candidate documents $d_u \in \mathcal{D}_u^c$, across all M users. After scoring, we retain the top scoring T request-target pairs. In practice, we find that setting T to the top two-thirds across the dataset works well. This step reflects the intuition that not all request-target pairs will benefit from retrieval augmentation, either due to the lack of suitable candidate documents in a user's historical document set, or due to underspecified requests making the target text simply too difficult to generate well – this is contrast with RAG setups in non-personalized scenarios where a large retrieval corpus ensures that most requests are likely to benefit from retrieval. After obtaining a high-quality set of training requests $\{q_u^*\}_{t=1}^{T}$, we discard the target texts, since they aren't used for training f_{retr} or for inference.

Candidate document selection Next, we use Eq. 1 to select the best documents for the retained requests, i.e. identify positive training documents. Given a request q_u^* selected for training, we take the P highest-scoring candidate documents $d_u \in \mathcal{D}_u^c$ as per Eq. (1) as positives, $\{d_u^+\}_{p=1}^{P}$. We sample N negative samples per positive randomly from the candidate document set for the user.

4.3 Retriever Optimization

Our f_{retr} is a cross-encoder initialized with a pretrained LM encoder and trained using data selected per Algorithm 1, through distillation of scores in Equation 1. While cross-encoders are expressive they produce scores which lie at the extremes of their score ranges (Menon et al., 2022; Yadav et al., 2022) – this hampers their ability to closely track the benefit of candidate documents for personalizing requests. We propose to remedy this through a scale calibrating training objective.

Algorithm 1 Selecting requests and positive candidate documents to train f_{retr}

```
Input: $\{\mathcal{D}_u\}_{u=1}^{M}$, $f_{\text{aux}}$  ▷ Historical documents for $M$ users and an auxiliary LM
for each user $u$ do
    $\mathcal{D}_u^c, \mathcal{D}_u^t \leftarrow \text{TemporalPartition}(\mathcal{D}_u)$  ▷ Temporally partition $\mathcal{D}_u$ into candidate and target documents
    for each target text $t_u \in \mathcal{D}_u^t$ do
        $q_u \leftarrow \text{GetRequest}(t_u)$  ▷ Obtain a synthetic or natural request
    end for
    for each $(q_u, t_u)$ pair in $\mathcal{D}_u^t$ do  ▷ Compute benefit of personalization for request-target pairs
        for each candidate $d_u$ in $\mathcal{D}_u^c$ do
            $Y[q_u, d_u] = \log p_{\text{aux}}(t_u|d_u, q_u) - \log p_{\text{aux}}(t_u|q_u)$  ▷ Equation (1)
        end for
    end for
end for
$\{q_u^*\}_{t=1}^{T} \leftarrow \text{TopK}(Y[q_u, d_u])$  ▷ Retain the top $T$ unique requests which are personalizable
for each retained request in $\{q_u^*\}_{k=1}^{T}$ do
    $\{d_u^+\}_{p=1}^{P} \leftarrow \text{TopK}(Y[q_u^*, d_u])$  ▷ Retain the top $P$ candidates that best personalize the target
end for
return $\{q_u^*, \{d_u^+\}_{p=1}^{P}\}_{t=1}^{T}$
```

Scale calibration Let $\mathbf{y}_q = [y_q^+, \ldots, y_q^-]$, where y_q^+ corresponds to the score of a positive document and y_q^- corresponds to the score of a negative document from Eq. 1. Here, $\mathbf{y}_q$ contains N negatives and 1 positive document. Similarly, let the predicted logits from $f_{\text{retr}} : (q_u, d_u) \rightarrow \mathbb{R}$ be denoted as $\mathbf{s}_q = [s_q^+, \ldots, s_q^-]$. Then, a standard KL-divergence loss is written as $KL(\mathbf{y}_q, \mathbf{s}_q) = -\sum_i \mathsf{sm}(y_{q,i}) \log \mathsf{sm}(s_{q,i})$, where sm represents the softmax function. Our proposed scale calibration modifies the KL divergence loss by adding an "anchor" example with target score y_0, which is a tunable hyperparameter, and logit s_0 set to 0, resulting in score vectors $\mathbf{y}'_q = [y_0, \mathbf{y}_q]$ and $\mathbf{s}'_q = [s_0, \mathbf{s}_q]$. The scale-calibrated KL-divergence loss is thus

$$KL(\mathbf{y}'_q, \mathbf{s}'_q) = -\sum_i \mathsf{sm}(y'_{q,i}) \log \mathsf{sm}(s'_{q,i}) \quad (2)$$

$$= -\sum_i \frac{e^{y_{q,i}}}{\sum_j e^{y_{q,j}} + e^{y_0}} \log \frac{e^{s_{q,i}}}{\sum_j e^{s_{q,j}} + 1} + \frac{e^{y_0}}{\sum_j e^{y_{q,j}} + e^{y_0}} \log \big(\sum_j e^{s_{q,j}} + 1\big). \quad (3)$$

We find that setting y_0 to the median value of scores from Eq (1) for positive candidate documents works well. This ensures that very large scores from f_{retr} are penalized (second term Eq 3) and smaller scores are prevented from being driven lower (first term Eq 3). Therefore f_{retr} scores are more evenly distributed over the score range. In practice, this ensures that predicted scores from f_{retr} more accurately reflect the distribution of f_{aux}, which in turn more closely tracks the utility of request-document pairs for personalization. We compare PEARL to baselines in §5.2 and present ablations in §C.2.

4.4 System Details

After training retriever f_{retr} offline, PEARL may be used to serve requests online. Given a unseen request, f_{retr} retrieves the top-k historical texts from $\mathcal{D}_u$, these are formatted into a prompt and input to f_{LLM} to generate a personalized target text t_u.

Our f_{retr} is initialized with a 110M parameter MPNET encoder (Song et al., 2020). For f_{LLM} we consider two performant LLMs, `davinci-003` and `gpt-3.5-turbo`. For f_{aux} we use FLANT5-XL with 3 billion parameters (Chung et al., 2022). Appendix A details our prompts and implementation.

5 Experiments

We demonstrate the effectiveness of PEARL on two personalized text generation datasets from social media platforms. For evaluation, we employ standard intrinsic evaluations, extrinsic evaluation based on downstream tasks using the generated text, and recently proposed personalized LLM-as-judge (Wang et al., 2023d). Then, in §5.3 we show how a calibrated retriever can be used for selective revision of underperforming requests. We present ablations in §C.2 and we demonstrate the calibration performance for our retriever in §C.3.

5.1 Experimental Setup

Data For evaluation, we use two open-ended long-form text generation datasets for social media: (1) Personalized post writing on WORKSM and (2) Personalized comment writing on AITA.

WORKSM WORKSM is an enterprise social network used for communication within organizations presenting a highly realistic platform for writing assistance. We obtain a random sample of ∼18k posts written by 1116 users from November 2020 to July 2023. To create an *evaluation set*, we manually examine posts greater than 50 words and receiving ≥2 comments, about 1K posts, and select 163 of the most recent posts from ∼80 users to serve as reference target texts t_u^*. These posts represent a diverse, engaging set that could benefit from personalized writing assistance and serve as high quality target references. At a high level, these posts share events, research studies, campaigns, and organizational news. Since WorkSm does not contain requests to the writing assistant, two authors not involved in model development manually wrote requests q_u per target text. Note that this was necessary given the highly regulated and private enterprise data in WorkSm preventing exposure to external crowdworkers. Our requests were authored following Guideline 1. To construct $\mathcal{D}_u$ posts created before t_u^* were used: On average, users had 31 historic posts (max of 169). To create our *training set*, we only retain posts > 10 words and users with ≥ 5 historic posts while excluding posts in our evaluation set. We generate synthetic requests with GPT-4 for training given the expense of manually authored requests – resulting in a set of $\sim$7k training requests. Enterprise contracts with API providers ensured the privacy of user data shared over the API.

AITA AITA is a Reddit subforum in which original posters (OP) describe personal moral conflicts and receive comments from other users judging them to be "the a**hole" or "not the a**hole". This dataset has been used in prior work on modeling the personal values of users (Plepi et al., 2022). We construct a personalized comment generation task from this data. We treat the OP posts as requests q_u, user comments as reference target texts t_u^*, and a user's previous comments as $\mathcal{D}_u$. Since the dataset lacks time metadata, we construct an *evaluation set* by sampling 10% of the posts as test requests, and further filter to 600 random target texts for our evaluation set to keep LLM experiments feasible. Evaluation users had 29 posts in $\mathcal{D}_u$ on average (max of 590). Our *training set* used the historical post-comment pairs from users in $\mathcal{D}_u$, resulting in $\sim$84k requests. Note that while Reddit comments are not the ideal platform for writing assistance, AITA is one of the few public datasets available for the task and resembles applications such as email response generation (Kannan et al., 2016). Appendix B details our datasets further.

Generation metrics Since personalized text generation aims to adhere to the style, knowledge, and values of *specific* users, effective evaluation for personalized generation remains an open problem (Wang et al., 2023d,a). This is in contrast to non-personalized generation, where desirable aspects of outputs can be defined uniformly across all test cases. As a result, we present evaluations using a host of standard evaluation setups aiming to demonstrate the effectiveness of Pearl from various perspectives. Our evaluations span the following standard setups (Dou et al., 2023): intrinsic evaluations based on n-gram/embedding similarity to reference texts, extrinsic evaluation through a classification accuracy based on generated text, and pairwise evaluation with personalized LLM-as-judge.

Specifically, for WorkSm we report standard evaluation measures based on n-gram and embedding similarity between generations and reference targets: ROUGE-1 (R1), ROUGE-2 (R2), and BertScore-F1 (BS-F1) (Zhang* et al., 2020). This serves as an intrinsic evaluation for WorkSm measuring the extent to which generations are similar to user authored texts. Next, since AITA users' comments primarily make a stance based on users' moral values, we measure if the stance in generated comments matches that of the user through a downstream stance prediction task – serving as an extrinsic evaluation. This evaluation may be seen as evaluating the extent to which model generations adhere to a user's values. We map generated comments to a binary "YTA" or "NTA" label based on simple high-precision rules mapping lexical variations of "you're the a**hole" and "not the a**hole" to the labels. This procedure was also found reliable for constructing ground truth labels in AITA (Plepi et al., 2022). Note that early attempts of using n-gram/embedding similarity measures for evaluation (BS-F1, R1, R2) resulted in unreliable evaluations for AITA due the large variation (length, vocabulary, emojis etc.) in AITA comments, therefore we opt for more stable extrinsic evaluations and LLM based evaluations described next.

For both AITA and WorkSm we conduct a pairwise evaluation with a recently proposed personalized LLM-as-judge (Wang et al., 2023d). Wang et al. show LLM based author identifications to be a reliable proxy task for distinguishing models of various qualities and being correlated with human quality ratings. Here, a judge LLM is presented with a reference text from a user and generations from the pair of systems being compared, then, it is prompted to select the system generation more likely to be authored by the author of the reference text. An author identification task aims to capture several aspects which distinguish individuals' writing, spanning style, knowledge and their values. In our evaluation, we compare Pearl outputs to the outputs from the best baseline as indicated by intrinsic/extrinsic evaluations and use the target reference text t_u^* in the LLM prompt as an example

LLM →	davinci-003	gpt-35-turbo
Method ↓	Macro F1(%)	Macro F1(%)
zSHOT-NP	41.97	50.43
kSHOT-NP	51.71	59.76
Random	55.52	59.47
BM25	57.26	61.66
MPNET-1B	53.72	59.23
UPR	55.76	58.15
RelevanceCE	56.85	59.59
PEARL	**61.21**	**65.34**

(a) Extrinsic classification accuracy in AITA.

LLM →	davinci-003			gpt-35-turbo		
Method ↓	BS-F1	R1	R2	BS-F1	R1	R2
zSHOT-NP	36.25	0.5029	0.2516	31.03	0.4627	0.2091
kSHOT-NP	34.08	0.4931	0.2431	32.51	0.4825	0.2258
Random	35.04	0.5036	0.2505	33.46	0.4893	0.2345
BM25	37.96	0.5287	0.2911	**36.57**	**0.5089**	0.2673
MPNET-1B	38.30	0.5281	0.2931	36.02	0.5063	0.2639
UPR	38.70	0.5337	0.3019	35.98	0.5054	0.2642
RelevanceCE	37.81	0.5288	0.2953	35.99	0.5038	0.2613
PEARL	**39.60**	**0.5419**	**0.3094**	36.49	0.5082	**0.2676**

(b) Intrinsic reference based metrics in WORKSM .

Table 1: PEARL is compared to non-personalized (NP) and LLMs personalized with retrieval on datasets of social media communication: (a) a dataset constructed from Reddit and (b) a workplace social media dataset.

of the users writing. We use GPT-4o as our judge LLM and present the judge prompt in Appendix B.4. In our evaluation we avoid rating aspects such as fluency, non-redundancy, etc. (Celikyilmaz et al., 2021) since we are primarily concerned with personalization performance and these qualities may be in conflict with specific users writing.

Baselines As baselines, we consider non-personalized models based on zero shot prompting (zSHOT-NP) and few-shot prompting with k randomly chosen example documents (kSHOT-NP). We consider retrieval-augmented personalized baselines, which selecting from a user's historical documents $\mathcal{D}_u$. They span selection at random from $\mathcal{D}_u$ (Random), with sparse retrieval by BM25, with dense retrieval by a strong MPNET model trained on 1 billion text pairs (MPNET-1B), an unsupervised crossencoder (Sachan et al., 2022) ranking documents with FLANT5-BASE likelihood: $p(q_u|d_u)$ (UPR), and a supervised crossencoder optimized on our dataset with request-document pairs, (q_u, d_u) in $\mathcal{D}_u$ (RelevanceCE). Appendix B.3 details our baselines.

5.2 Generation Evaluation

Table 1 and 2 report our evaluations. Appendix C presents ablation (C.2) and calibration (C.3) results.

Reference based evaluation Tables 1b and 1a reports automated metrics on AITA and WORKSM. First we observe that personalization through retrieval, even at Random, generally improves upon non-personalized approaches (NP), which is consistent with prior work (Salemi et al., 2023). Next, we note that the best baseline is not consistent, varying between BM25, and unsupervised crossencoder (UPR) – indicating that retrieval models designed for request-document relevance vary in performance depending on the dataset and inference LLM. Finally, we note that PEARL consistently performs at par or better than the best baselines across datasets and LLMs, indicating the effectiveness of training f_{retr} for personalized generation. For the more reliable classification metrics obtainable in AITA, PEARL outperforms all baselines with improvements of 1.5 to 5 Macro F1 points. Next, we report performance in more expressive LLM-as-judge evaluations.

	davinci-003	gpt-35-turbo
	P / B / T (%)	P / B / T (%)
AITA	**46.8** /40.3 /12.8$_{\alpha=0.56}$	**46.6** /44.9 /8.3$_{\alpha=0.55}$
WORKSM	**46.6** /42.5 /10.8$_{\alpha=0.42}$	38.9 /**42.6** /18.5$_{\alpha=0.28}$

Table 2: LLM-as-judge win-rate evaluation for AITA and WORKSM selecting a generation to be more aligned with an authors writing sample. The LLM could prefer the Proposed system (PEARL), the Baseline (BM25), or judge the outputs as Tied – denoted with P, B, and T.

Pairwise LLM-as-judge evaluation In Table 2 we report the results of personalization evaluation following the setup described in §5.1. Here, we compare against BM25-augmented as it performs within our top 2 baselines in automatic evaluations - this strong performance is consistent with prior work (Izacard et al., 2022; Thakur et al., 2021). We use GPT-4o as a judge LLM and run every pair of inputs through the judge LLM 3 times, we report average win rates over all the instances in our test set and over 3 repeated runs. Further, we randomly swap the position of the baseline and proposed method generations in the prompt to account for position biases in the judge LLM. Finally, we also report the agreement between the 3 judge LLM runs using Krippendorff's alpha (α) to ensure that

LLM judgements are consistent across runs.

In Table 2, PEARL achieves a greater win-rate than BM25 in 3 of 4 settings. In these settings we also note that the LLM judgments remain consistent across 3 repeated runs with Krippendorff's alpha between $0.41 - 0.56$ (0 indicates chance agreement). While BM25 sees a greater win-rate in WORKSM with `gpt-35-turbo`, the judgments see lower agreement ($\alpha = 0.28$) indicating the outputs to be harder to distinguish. Finally, comparing to Table 1 we see that the trends of extrinsic and intrinsic reference based evaluations are retained in LLM-as-judge evaluations – consistently indicating the benefit of PEARL across evaluation setups, inference LLMs, and datasets. In Appendix C we show an example from AITA to show the kinds of retrievals and outputs that make PEARL effective.

5.3 Selective Revision with PEARL

Having established PEARL to be an effective model for generation, we show f_{retr} to be generation calibrated in Appendix C.3. Here, we demonstrate the usefulness of a calibrated retriever in a case study using the retriever scores to selectively revise generations. Specifically, we treat the scores from f_{retr} as a predictor of retrieval performance, and in-turn text generation performance. We assume that if f_{retr} cannot find a highly scored in-context example, the generated response will be of low quality and can benefit from LLM revision (Figure 3).

Setup Given our trained retriever, we take all top-1 document scores for each request $s_1 = \max_{d_u \in \mathcal{D}_u} f_{\text{retr}}(q_u, d_u)$ and learn a threshold θ on s_1 that maximizes a downstream performance metric on a held-out development set (R2 in WORKSM and Macro-F1 in AITA). Then, given a generated target text t_u with $s_1 < \theta$, we selectively revise t_u where f_{LLM} is prompted to edit the target text. We report results of selective revision compared to a single round of generation (i.e., no revision) and full revision over the entire dataset (i.e., 100% revision). We repeat this for BM25. We provide further details and analysis in Appendix C.4.

Dataset →	AITA	WORKSM		
Method ↓ / LLM →	`gpt-35-turbo`	`gpt-35-turbo`		
$f_{\text{retr}} = \text{BM25}$	Macro F1 (%)	BS-F1	R1	R2
Stage 1 (no revision)	**59.99**	36.15	0.5052	0.2611
All (100% revision)	58.36	35.45	0.5096	0.2573
Selective revision	57.71	**37.29**	**0.5206**	**0.2738**
$f_{\text{retr}} = \text{Proposed}$	Macro F1 (%)	BS-F1	R1	R2
Stage 1 (no revision)	65.15	37.02	0.5124	0.2709
All (100% revision)	64.85	35.47	0.5045	0.2520
Selective revision	**65.36**	**37.71**	**0.5236**	**0.2818**

Table 3: Selectively revising target texts t_u based on scores from our retriever vs BM25. Also present are results of no revision and revising all outputs (100% revision) from Stage 1 outputs.

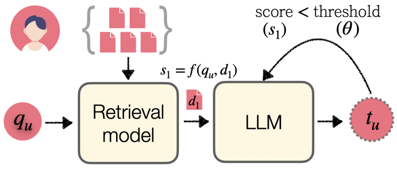


Figure 3: Generation calibration of f_{retr} allows us to use its predicted scores for performance prediction and selectively revise potentially bad generations.

Results In Table 3 we see that selective revision improves or retains performance upon a single round of generation ("Stage 1") by 2-4% in downstream performance metrics with f_{retr} =Proposed and BM25 for WORKSM. However, for AITA we see that selective revision based on BM25 shows a marked drop in performance indicating its dataset dependent calibration performance. Importantly, note that Macro F1 doesn't measure aspects of style which may have changed in revision. Finally, editing *all* outputs produced by Stage 1 generation consistently leads to degraded performance ("All"), indicating that editing is not always helpful.

We also observe that PEARL chooses 75.8% and 77.9% instances for editing in WORKSM and AITA, respectively. This indicates the potential for generation calibrated retrievers to reduce the number of expensive LLM calls made while ensuring better personalization performance. In Figure 5 (Appendix C.4) we analyze the performance of selective revision against request and user profile length. In a manual examination of requests with a low s_1 score by the PEARL f_{retr}, we find the requests to be underspecified and often require further information from a user e.g. the request "Write a post about how I like to relax after work", aims to generate a target discussing more specific forms of relaxation not present in any historical documents. This indicates that generation calibrated retrievers may be used for other forms of selective prediction and user interaction – e.g. selectively withholding predictions when satisfactory generations are unlikely or obtaining more information from users through follow-up questions. We leave such explorations to exciting future work.

6 Conclusion

In this paper we present PEARL– an LLM based writing assistant personalized with generation calibrated retrievers. We propose a method for training generation calibrated retrievers through a careful selection of training data and a scale calibrated objective. In a series of holistic evaluations, we demonstrate the effectiveness of our approach in datasets of social media communication compared to baselines (§5.2) as well as ablated models (Appendix C.2). We demonstrate the calibration performance for our retriever (Appendix C.3), and show how our retrieval model can double as a performance predictor (§5.3) and can identify outputs which can benefit from LLM revision.

7 Ethical and broader impact

Having introduced PEARL as an effective personalization strategy for writing assistance and discussed its benefits we review two implications of concern arising from better personalized text generation: challenges to factuality, and longer term influence on language use and communication.

Challenges to factuality The emergence of LLMs and their ability to generate compelling text has seen a subsequent rise in the cases of malicious use of these technologies. Augenstein et al. (2023) overview four such classes of harm: personalized attacks on individuals in the form of phishing attacks and tailored misinformation, impersonation of trusted figures (e.g. journalists or regulators), a glut of paraphrased misinformation evading detection by automatic tools often used by fact checkers, and large scale creation of fake social media profiles and plagiarized content (Brewster et al., 2023). It is possible that improvements in personalized text generation are likely to excacertabe each of these problems. To account for this, several technology and policy initiatives are under active development (Augenstein et al., 2023). These span detection of AI-generated content, cryptographic signatures intended to prove the authenticity of content, to government regulation and public education, however, their effectiveness remains under investigation.

Language use and communication Current understanding of computer mediated communication suggests that users interpersonal communication patterns are influenced by the tool/medium used for communication (Poddar et al., 2023) with a potential for these influences to have longer term influences on communication in the absence of these tools (Hancock et al., 2020). Hancock et al. outline these implications as ranging from shifts in language use (e.g a social expectation of more positive responses (Hohenstein and Jung, 2018)), issues of how individuals portray themselves and evaluate others, to long term feedback loops resulting in how we perceive ourselves. However, understanding of the implications of AI mediated communication, specially those powered by powerful LLMs, is largely developing (Hancock et al., 2020). It is likely that wide spread personalization in LLM communication agents, will necessitate further understanding of these factors and the design of systems that incorporates this understanding to ameliorate harms.

References

Nader Akoury, Shufan Wang, Josh Whiting, Stephen Hood, Nanyun Peng, and Mohit Iyyer. 2020. STORIUM: A Dataset and Evaluation Platform for Machine-in-the-Loop Story Generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6470–6484, Online. Association for Computational Linguistics.

Xiang Ao, Xiting Wang, Ling Luo, Ying Qiao, Qing He, and Xing Xie. 2021. PENS: A dataset and generic framework for personalized news headline generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 82–92, Online. Association for Computational Linguistics.

Simran Arora, Avanika Narayan, Mayee F Chen, Laurel Orr, Neel Guha, Kush Bhatia, Ines Chami, and Christopher Re. 2023. Ask me anything: A simple strategy for prompting language models. In *The Eleventh International Conference on Learning Representations*.

Isabelle Augenstein, Timothy Baldwin, Meeyoung Cha, Tanmoy Chakraborty, Giovanni Luca Ciampaglia, David Corney, Renee DiResta, Emilio Ferrara, Scott Hale, Alon Halevy, Eduard Hovy, Heng Ji, Filippo Menczer, Ruben Miguez, Preslav Nakov, Dietram Scheufele, Shivam Sharma, and Giovanni Zagni. 2023. Factuality challenges in the era of large language models. *arXiv preprint arXiv:2310.05189*.

Tal August, Katharina Reinecke, and Noah A. Smith. 2022. Generating scientific definitions with controllable complexity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8298–8317, Dublin, Ireland. Association for Computational Linguistics.

Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, and Rodrigo Nogueira. 2022. Inpars: Unsupervised dataset generation for information retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 2387–2392, New York, NY, USA. Association for Computing Machinery.

Jack Brewster, Macrina Wang, and Coalter Palmer. 2023. Plagiarism-bot? how low-quality websites are using ai to deceptively rewrite content from mainstream news outlets. NewsGaurd, The Internet Trust Tool.

Asli Celikyilmaz, Elizabeth Clark, and Jianfeng Gao. 2021. Evaluation of text generation: A survey. *arXiv preprint arXiv:2006.14799*.

Tuhin Chakrabarty, Vishakh Padmakumar, and He He. 2022. Help me write a poem: Instruction tuning as a vehicle for collaborative poetry writing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6848–6863, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Mia Xu Chen, Benjamin N. Lee, Gagan Bansal, Yuan Cao, Shuyuan Zhang, Justin Lu, Jackie Tsay, Yinan Wang, Andrew M. Dai, Zhifeng Chen, Timothy Sohn, and Yonghui Wu. 2019. Gmail smart compose: Real-time assisted writing. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 2287–2295, New York, NY, USA. Association for Computing Machinery.

Shiping Chen, Duncan P Brumby, and Anna L Cox. 2023. Using writing assistants to accelerate the peer review process. *Second Workshop on Intelligent and Interactive Writing Assistants, CHI 2023*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Elizabeth Clark, Anne Spencer Ross, Chenhao Tan, Yangfeng Ji, and Noah A. Smith. 2018. Creative writing with a machine in the loop: Case studies on slogans and stories. In *23rd International Conference on Intelligent User Interfaces*, IUI '18, page 329–340, New York, NY, USA. Association for Computing Machinery.

Daniel Cohen, Bhaskar Mitra, Oleg Lesota, Navid Rekabsaz, and Carsten Eickhoff. 2021. Not all relevance scores are equal: Efficient uncertainty and calibration modeling for deep retrieval models. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 654–664, New York, NY, USA. Association for Computing Machinery.

Antonia Creswell, Murray Shanahan, and Irina Higgins. 2023. Selection-inference: Exploiting large language models for interpretable logical reasoning. In *The Eleventh International Conference on Learning Representations*.

Robert Dale and Jette Viethen. 2021. The automated writing assistance landscape in 2021. *Natural Language Engineering*, 27(4):511–518.

Shehzaad Dhuliawala, Leonard Adolphs, Rajarshi Das, and Mrinmaya Sachan. 2022. Calibration of machine reading systems at scale. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1682–1693, Dublin, Ireland. Association for Computational Linguistics.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2023. Chain-of-verification reduces hallucination in large language models. *arXiv preprint arXiv:2309.11495*.

Yao Dou, Philippe Laban, Claire Gardent, and Wei Xu. 2023. Automatic and human-ai interactive text generation. *arXiv preprint arXiv:2310.03878*.

Asma Farajidizaji, Vatsal Raina, and Mark Gales. 2023. Is it possible to modify text to a target readability level? an initial investigation using zero-shot large language models. *arXiv preprint arXiv:2309.12551*.

Katy Ilonka Gero, Vivian Liu, and Lydia Chilton. 2022. Sparks: Inspiration for science writing using language models. In *Proceedings of the 2022 ACM Designing Interactive Systems Conference*, DIS '22, page 1002–1019, New York, NY, USA. Association for Computing Machinery.

Hugo Gonçalo Oliveira. 2017. A survey on intelligent poetry generation: Languages, features, techniques, reutilisation and evaluation. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 11–20, Santiago de Compostela, Spain. Association for Computational Linguistics.

Hila Gonen, Srini Iyer, Terra Blevins, Noah A. Smith, and Luke Zettlemoyer. 2022. Demystifying prompts in language models via perplexity estimation. *arXiv preprint arXiv:2212.04037*.

Yue Guo, Wei Qiu, Yizhong Wang, and Trevor Cohen. 2021. Automated lay language summarization of biomedical scientific reviews. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(1):160–168.

Shivanshu Gupta, Matt Gardner, and Sameer Singh. 2023. Coverage-based example selection for in-context learning. *arXiv preprint arXiv:2305.14907*.

Jeffrey T Hancock, Mor Naaman, and Karen Levy. 2020. AI-Mediated Communication: Definition, Research Agenda, and Ethical Considerations. *Journal of Computer-Mediated Communication*, 25(1):89–100.

Jess Hohenstein and Malte Jung. 2018. Ai-supported messaging: An investigation of human-human text conversation with ai support. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI EA '18, page 1–6, New York, NY, USA. Association for Computing Machinery.

Krystal Hu. 2023. Chatgpt sets record for fastest-growing user base - analyst note. National Bureau of Economic Research, Digest No. 6.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*.

Anjuli Kannan, Karol Kurach, Sujith Ravi, Tobias Kaufmann, Andrew Tomkins, Balint Miklos, Greg Corrado, Laszlo Lukacs, Marina Ganea, Peter Young, and Vivek Ramavajjala. 2016. Smart reply: Automated response suggestion for email. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 955–964, New York, NY, USA. Association for Computing Machinery.

Omar Khattab, Keshav Santhanam, Xiang Lisa Li, David Hall, Percy Liang, Christopher Potts, and Matei Zaharia. 2023. Demonstrate-search-predict: Composing retrieval and language models for knowledge-intensive nlp. *arXiv preprint arXiv:2212.14024*.

Hannah Rose Kirk, Bertie Vidgen, Paul Röttger, and Scott A Hale. 2023. Personalisation within bounds: A risk taxonomy and policy framework for the alignment of large language models with personalised feedback. *arXiv preprint arXiv:2303.05453*.

Cheng Li, Mingyang Zhang, Qiaozhu Mei, Weize Kong, and Michael Bendersky. 2023a. Automatic prompt rewriting for personalized text generation. *arXiv preprint arXiv:2310.00152*.

Cheng Li, Mingyang Zhang, Qiaozhu Mei, Yaqing Wang, Spurthi Amba Hombaiah, Yi Liang, and Michael Bendersky. 2023b. Teach llms to personalize – an approach inspired by writing education. *arXiv preprint arXiv:2308.07968*.

Daliang Li, Ankit Singh Rawat, Manzil Zaheer, Xin Wang, Michal Lukasik, Andreas Veit, Felix Yu, and Sanjiv Kumar. 2023c. Large language models with controllable working memory. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1774–1793, Toronto, Canada. Association for Computational Linguistics.

Zihao Li, Samuel Belkadi, Nicolo Micheletti, Lifeng Han, Matthew Shardlow, and Goran Nenadic. 2023d. Large language models and control mechanisms improve text readability of biomedical abstracts.

Zhiyu Lin, Upol Ehsan, Rohan Agarwal, Samihan Dani, Vidushi Vashishth, and Mark Riedl. 2023. Beyond prompts: Exploring the design space of mixed-initiative co-creativity systems. In *ICCC*.

Hongyin Luo, Yung-Sung Chuang, Yuan Gong, Tianhua Zhang, Yoon Kim, Xixin Wu, Danny Fox, Helen Meng, and James Glass. 2023. Sail: Search-augmented instruction learning. *arXiv preprint arXiv:2305.15225*.

Cerstin Mahlow. 2023. Writing tools: Looking back to look ahead. *Second Workshop on Intelligent and Interactive Writing Assistants, CHI 2023*.

Bodhisattwa Prasad Majumder, Shuyang Li, Jianmo Ni, and Julian McAuley. 2019. Generating personalized recipes from historical user preferences. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5976–5982, Hong Kong, China. Association for Computational Linguistics.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada. Association for Computational Linguistics.

Pierre-Emmanuel Mazaré, Samuel Humeau, Martin Raison, and Antoine Bordes. 2018. Training millions of personalized dialogue agents. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2775–2779, Brussels, Belgium. Association for Computational Linguistics.

Aditya Menon, Sadeep Jayasumana, Ankit Singh Rawat, Seungyeon Kim, Sashank Reddi, and Sanjiv Kumar. 2022. In defense of dual-encoders for neural ranking. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 15376–15400. PMLR.

Piotr Mirowski, Kory W. Mathewson, Jaylen Pittman, and Richard Evans. 2023. Co-writing screenplays and theatre scripts with language models: Evaluation by industry professionals. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA. Association for Computing Machinery.

Sonia K. Murthy, Kyle Lo, Daniel King, Chandra Bhagavatula, Bailey Kuehl, Sophie Johnson, Jonathan Borchardt, Daniel S. Weld, Tom Hope, and Doug Downey. 2022. Accord: A multi-document approach to generating diverse descriptions of scientific concepts.

Maria Nadejde and Joel Tetreault. 2019. Personalizing grammatical error correction: Adaptation to proficiency level and L1. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 27–33, Hong Kong, China. Association for Computational Linguistics.

Jianmo Ni, Zachary C. Lipton, Sharad Vikram, and Julian McAuley. 2017. Estimating reactions and recommending products with generative models of reviews. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 783–791, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Xiaoman Pan, Wenlin Yao, Hongming Zhang, Dian Yu, Dong Yu, and Jianshu Chen. 2023. Knowledge-in-context: Towards knowledgeable semi-parametric language models. In *The Eleventh International Conference on Learning Representations*.

Andrea Papenmeier, Dagmar Kern, Daniel Hienert, Alfred Sliwa, Ahmet Aker, and Norbert Fuhr. 2021. Starting conversations with search engines - interfaces that elicit natural language queries. In *Proceedings of the 2021 Conference on Human Information Interaction and Retrieval*, CHIIR '21, page 261–265, New York, NY, USA. Association for Computing Machinery.

Martin J Pickering and Simon Garrod. 2013. An integrated theory of language production and comprehension. *Behavioral and brain sciences*, 36(4):329–347.

Joan Plepi, Béla Neuendorf, Lucie Flek, and Charles Welch. 2022. Unifying data perspectivism and personalization: An application to social norms. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7391–7402, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ritika Poddar, Rashmi Sinha, Mor Naaman, and Maurice Jakesch. 2023. Ai writing assistants influence topic choice in self-presentation. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI EA '23, New York, NY, USA. Association for Computing Machinery.

Dongqi Pu and Vera Demberg. 2023. ChatGPT vs human-authored text: Insights into controllable text summarization and sentence style transfer. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 1–18, Toronto, Canada. Association for Computational Linguistics.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2655–2671, Seattle, United States. Association for Computational Linguistics.

Devendra Sachan, Mike Lewis, Mandar Joshi, Armen Aghajanyan, Wen-tau Yih, Joelle Pineau, and Luke Zettlemoyer. 2022. Improving passage retrieval with zero-shot question generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3781–3797, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Alireza Salemi, Surya Kallumadi, and Hamed Zamani. 2024. Optimization methods for personalizing large language models through retrieval augmentation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 752–762, New York, NY, USA. Association for Computing Machinery.

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

Zejiang Shen, Tal August, Pao Siangliulue, Kyle Lo, Jonathan Bragg, Jeff Hammerbacher, Doug Downey, Joseph Chee Chang, and David Sontag. 2023. Beyond summarization: Designing ai support for real-world expository writing tasks. *Second Workshop on Intelligent and Interactive Writing Assistants, CHI 2023*.

Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Scott Wen tau Yih. 2023. Trusting your evidence: Hallucinate less with context-aware decoding. *arXiv preprint arXiv:2305.14739*.

Kumar Shridhar, Harsh Jhamtani, Hao Fang, Benjamin Van Durme, Jason Eisner, and Patrick Xia. 2023. Screws: A modular framework for reasoning with revisions. *arXiv preprint arXiv:2309.13075*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. In *Advances in Neural Information Processing Systems*, volume 33.

Simeng Sun, Wenlong Zhao, Varun Manjunatha, Rajiv Jain, Vlad Morariu, Franck Dernoncourt, Balaji Vasan Srinivasan, and Mohit Iyyer. 2021. IGA: An intent-guided authoring assistant. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5972–5985, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Stojan Trajanovski, Chad Atalla, Kunho Kim, Vipul Agarwal, Milad Shokouhi, and Chris Quirk. 2021. When does text prediction benefit from additional context? an exploration of contextual signals for chat and email messages. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Papers*, pages 1–9, Online. Association for Computational Linguistics.

Danqing Wang, Kevin Yang, Hanlin Zhu, Xiaomeng Yang, Andrew Cohen, Lei Li, and Yuandong Tian. 2023a. Learning personalized story evaluation. *arXiv preprint arXiv:2310.03304*.

Liang Wang, Nan Yang, and Furu Wei. 2023b. Learning to retrieve in-context examples for large language models. *arXiv preprint arXiv:2307.07164*.

Sitong Wang, Lydia B Chilton, and Jeffrey V Nickerson. 2023c. Writing with generative ai: Multi-modal and multi-dimensional tools for journalists. *Second Workshop on Intelligent and Interactive Writing Assistants, CHI 2023*.

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

Tongshuang Wu, Michael Terry, and Carrie Jun Cai. 2022. Ai chains: Transparent and controllable human-ai interaction by chaining large language model prompts. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA. Association for Computing Machinery.

Nishant Yadav, Nicholas Monath, Rico Angell, Manzil Zaheer, and Andrew McCallum. 2022. Efficient nearest neighbor search for cross-encoder models using matrix factorization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2171–2194, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Le Yan, Zhen Qin, Xuanhui Wang, Michael Bendersky, and Marc Najork. 2022. Scale calibration of deep ranking models. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, page 4300–4309, New York, NY, USA. Association for Computing Machinery.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2023. Making retrieval-augmented language models robust to irrelevant context.

Hamed Zamani and Michael Bendersky. 2023. Multivariate representation learning for information retrieval. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, page 163–173, New York, NY, USA. Association for Computing Machinery.

Hanqing Zhang, Haolin Song, Shaoyu Li, Ming Zhou, and Dawei Song. 2023a. A survey of controllable text generation using transformer-based pre-trained language models. *ACM Comput. Surv.*, 56(3).

Jieyu Zhang, Ranjay Krishna, Ahmed H. Awadallah, and Chi Wang. 2023b. Ecoassistant: Using llm assistant more affordably and accurately. *arXiv preprint arXiv:2310.03046*.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213, Melbourne, Australia. Association for Computational Linguistics.

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

Yiming Zhang, Shi Feng, and Chenhao Tan. 2022. Active example selection for in-context learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9134–9148, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Prompt 1 f_{LLM} prompt used to generate a target text given historical examples retrieved by f_{retr} and a target request for AITA.

```
For a POST from the subreddit Am I The
Asshole write a COMMENT explaining if the
 author of a post is an asshole or not
the asshole as a COMMENTER.
Use the following instructions for your
response:
1. Read the below example comments by
the COMMENTER.
2. Write the comment as the COMMENTER
mimicing the length, style, reasoning,
and stances of their comments.
Here are some example comments by the
COMMENTER: {{historical_examples}}
POST: {{target_request}}
Write the COMMENT mimicing the length,
style, reasoning, and stances of the
COMMENTERS comments.
```

A Model Details

Retriever We instantiate f_{retr} with the pre-trained MPNET, which is relatively lightweight at 110M parameters (Song et al., 2020). We obtain an output score from f_{retr} as $\mathbf{w}^T \tanh\left(\mathbf{W}^T \textsc{Enc}([q_u, d_u])\right)$, where Enc represents the CLS token from the final layer of the encoder, and q_u and d_u are the text of the input

Prompt 2 f_{LLM} prompt used to generate a target text given historical examples retrieved by f_{retr} and a target request for WORKSM.

```
Given a REQUEST from a USER to author a
POST, write a POST for an enterprise
social media site mimicking the user to
satisfy the REQUEST.
Use the following instructions for your
response:
1. You should maintain consistency in
tone and style with the USER's historical
 posts.
2. You should imitate the language style
 of the USER's historical posts.
3. You should employ similar rhetorical
methods as the USER's historical posts.
Here are some historical posts by the
USER: {{historical_examples}}
REQUEST: {{target_request}}
Write the POST to satisfy the REQUEST
mimicing the tone, style, and rhetorical
 methods of the USER's historical posts.
```

request and historical document. The encoder parameters, $\mathbf{w}$, and $\mathbf{W}$ are trained.

Text generation models For f_{LLM} we consider two performant LLMs offered via API by Azure OpenAI, `davinci-003` and `gpt-3.5-turbo`. For f_{aux} we consider a smaller but still effective encoder-decoder language model, FLANT5-XL, with 3 billion parameters (Chung et al., 2022). The latter model is open-sourced, allowing us to access its token likelihoods directly, a requirement of Eq. 1. We obtain target text likelihoods by taking the average of log-probabilities of individual token likelihoods from FLANT5-XL.

LLM prompts We use Prompts 2 and 1 for LLM inference. The same prompts are used with `davinci-003` and `gpt-35-turbo`. For constructing training data in Eq 1 with a FLANT5-XL, f_{aux} we use Prompts 5, 6, 3, 4. Note that computing $p_{\text{aux}}(t_u|q_u)$ uses a set of randomly chosen few shot examples from the training set fixed across requests rather than the request alone.

PEARL implementation In constructing training data for f_{retr} we use $|\mathcal{D}_u^t| = 8$, i.e we treat the 8 most recent texts per user as their target texts. To train f_{retr}, we consider the top two canadidate documents per Eq. (1) as positive examples per request and use three negatives per positive, i.e., $P = 2$ and $N = 3$. In our LLM prompts, we use $k = 3$ retrieved examples for WORKSM and $k = 4$ for AITA, tuned on a dev set, and set generation temperature to zero.

Prompt 3 f_{aux} prompt used to compute $p_{\text{aux}}(t_u|q_u)$ in Eq (1) for AITA.

```
Here are some example posts on the Am I
The Asshole subreddit:
{{random_fewshot_examples}}. Target post:
{{target_post}}. Write a users comment for
this post:
```

Prompt 4 f_{aux} prompt used to compute $p_{\text{aux}}(t_u|d_u, q_u)$ in Eq (1) for AITA.

```
Here is an comment on a post by a user
on the Am I the Asshole subreddit:
{{candidate_comment}}. Target post:
{{target_post}}. Write a users comment for
this post:
```

We also use temperatures for target scores input to softmax functions in Eq. (2), $\mathbf{y}'_u/\tau$ with $\tau = 5$. Finally, we set $y_0 = 110$ for WORKSM and $y_0 = 5$ for AITA, which are the median values of Eq. (1) for each respective dataset on the training data. We tuned y_0 on a dev set constructed similar to our training set to 25 and 75 percentile values of Eq. (1). Our retrievers were trained on Nvidia V100 GPUs with 16GB memory or Nvidia RTX A6000 GPUs with 48GB memory. Experiments for training retrievers required about 300 hours in total.

B Experimental Details

Here we present various details of datasets, baselines, and manual evaluation.

B.1 Evaluation Requests in WORKSM

For evaluation in WORKSM two authors not involved in model development manually authored requests for each of the 163 target posts in our evaluation set. Guidelines presented to annotators for the requests are presented in Guideline 1. The requests are intended to be brief and include the salient information contained in the post. Note that annotators external to the authors weren't recruited for authoring requests due to the private and highly regulated nature of WORKSM.

B.2 Training Requests in WORKSM

Section 5.1 notes that our training set for WORKSM was constructed from synthetic requests generated by GPT4. The prompt for this is presented in Prompt 11. We follow an incremental approach

Prompt 5 f_{aux} prompt used to compute $p_{\text{aux}}(t_u|q_u)$ in Eq (1) for WORKSM.

```
Here is are some posts by a user on an
enterprise social network:
{{random_fewshot_examples}}
Here is an outline for a target post by
the user: {{target_request}}. Write the
target post:
```

Prompt 6 f_{aux} prompt used to compute $p_{\text{aux}}(t_u|d_u, q_u)$ in Eq (1) for WORKSM.

```
Here is an example post by a user on an
enterprise social network:
{{candidate_document}}. Here is an outline
for a target post by the user:
{{target_request}}. Write the target post:
```

Prompt 7 Judge LLM prompt used to select a generated post more likely to align with a reference post authored by a user for WORKSM.

```
You an an experienced linguist who helps
 people compare social media texts.
Given a REFERENCE POST and two
TARGET POSTS judge which of the TARGET
POSTs is significantly more likely to be
written by the same author as the
REFERENCE POST.
For your response use the following
instructions:
1. Make your judgement based on
stylistic patterns, ordering of
information, and tone used.
2. Output POST ONE if it is significantly
 more likely to be written by the same
author as the REFERENCE POST.
3. Output POST TWO if it is significantly
 more likely to be written by the same
author as the REFERENCE POST.
4. Output BOTH if either post could have
 been written by the same author or
neither could have been written by the
same author.
Here are the POSTS:
REFERENCE POST: {{reference_post}}
POST ONE: {{post_one}}
POST TWO: {{post_two}}
Output a justification for your
judgement, then output POST ONE, POST
TWO, or BOTH to indicate your final
decision.
```

to construct the synthetic requests: first extracting the salient aspects of the post, followed by concatenation of these aspects to result in the request. The salient aspects span: an overview of the post, proper nouns mentioned in the post, contact information, links to webpages, and any specialized knowledge or anecdotes in the post. Given the success of chain-of-thought prompting, we generate an explanation followed by salient aspects of the post – the explanations are not used elsewhere. Enterprise contracts ensure the privacy of user data shared over the API.

B.3 Baselines

We consider the following non-personalized baselines: ZSHOT-NP: This represents a non-personalized approach prompting only with the request. KSHOT-NP: A zero-shot non-personalized approach using a fixed randomly selected set of k documents for all requests. For AITA, the examples are balanced across labels.

We consider the following retrieval-augmented personalized baselines, selecting from a user's historical documents $\mathcal{D}_u$: Random: Random selection of k documents from $\mathcal{D}_u$. BM25: Represents a classic performant retrieval model based on query-document term overlap. MPNET-1B: This a strong MPNET bi-encoder trained on 1 billion text pairs from numerous domains.[1] Documents are ranked for a request using cosine similarity between embeddings. QL-FT5: An approach which ranks documents based on $p(q_u|d_u)$ with a pretrained FLANT5-BASE with 250M parameters (Sachan et al., 2022). This may be seen as an unsupervised crossencoder. RelevanceCE: A supervised crossencoder with the same architecture as f_{retr} in PEARL but differing in training. This is trained on pairs of (q_u, d_u) in $\mathcal{D}_u$ treated as positive training pairs with a crossentropy loss, with negatives selected as a random historical document from the same user not but corresponding to q_u. Note that this corresponds to a crossencoder optimized for request-document relevance, i.e. $p(\text{relevance} = 1|q_u, d_u)$, rather than personalized target text generation.

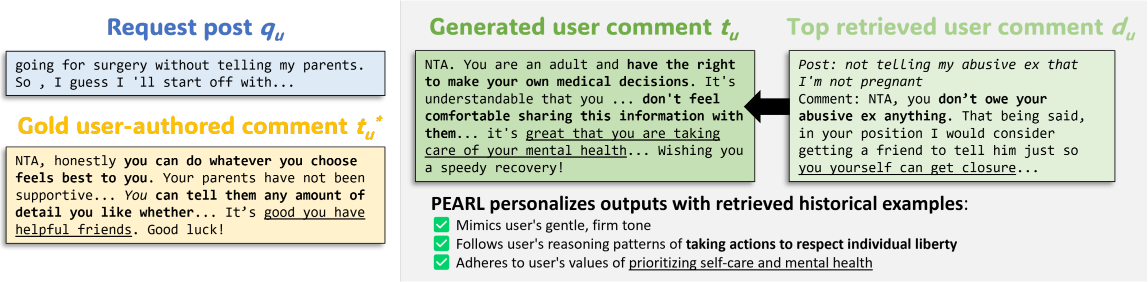
B.4 Judge LLM prompts

In Prompt 8 and 7 we present prompts for GPT-4o as a judge LLM discussed in §5.2.

C Additional Results

Here we present additional results in addition to those presented in §5.2. We present these here primarily in the interest of space.

[1] HF model: sentence-transformers/all-mpnet-base-v2


Request post q_u
Gold user-authored comment t_u^*
Generated user comment t_u
Top retrieved user comment d_u

going for surgery without telling my parents. So , I guess I 'll start off with...

NTA, honestly **you can do whatever you choose feels best to you**. Your parents have not been supportive... *You* **can tell them any amount of detail you like whether**... It’s good you have helpful friends. Good luck!

NTA. You are an adult and **have the right to make your own medical decisions**. It's understandable that you ... **don't feel comfortable sharing this information with them**... it's great that you are taking care of your mental health... Wishing you a speedy recovery!

Post: not telling my abusive ex that I'm not pregnant
Comment: NTA, you **don’t owe your abusive ex anything**. That being said, in your position I would consider getting a friend to tell him just so you yourself can get closure...

PEARL personalizes outputs with retrieved historical examples:

☑ Mimics user's gentle, firm tone
☑ Follows user's reasoning patterns of **taking actions to respect individual liberty**
☑ Adheres to user's values of prioritizing self-care and mental health

Figure 4: A qualitative example illustrating the effectiveness of PEARL on AITA: Given a request post q_u describing an ambiguous interpersonal situation regarding sharing medical information, PEARL retrieves a historical user comment d_u that demonstrates the user’s characteristic tone and values, and generates a comment t_u highly similar to the ground-truth user comment t_u^*. We **bold** qualitatively similar phrases about individual liberties and *italicize* phrases about self-care and mental health. All texts are abbreviated for space, and provided in full in Appendix C.

C.1 Qualitative example from AITA

Here we provide the full texts of the qualitative AITA examples from Figure 4, alongside the top-retrieved comment from BM25 to show the qualitative difference in retriever effectiveness.

Original post q_u: going for surgery without telling my parents. So , I guess I ’ll start off with background information .I am 21 , and trans masculine , or female to male if that term works for you

Gold user-authored comment t_u^*: “NTA, honestly you can do whatever you choose feels best to you. Your parents have not been supportive of your transition so they have elected not to be a part of the process. You can tell them any amount of detail you like whether before, after, or at different times. It’s good you have helpful friends. Good luck!”

Top PEARL-retrieved comment d_u: “NTA, you don’t owe your abusive ex anything. That being said, in your position I would consider getting a friend to tell him just so you yourself can get closure and not be wondering when he’s going to harass you about it.”

Top BM25-retrieved comment d_u: “YTA, just another lame guy who is only interested in a girl if he has a chance to sleep with her”

PEARL LLM output: “NTA. You are an adult and have the right to make your own medical decisions. It’s understandable that you don’t have a great relationship with your parents and don’t feel comfortable sharing this information with them. It’s also important to note that being transgender is not self-harm and it’s great that you are taking care of your mental health by seeing a counselor and nurse regularly. Wishing you a speedy recovery!”

Dataset →	AITA	WORKSM		
LLM →	gpt-35-turbo	gpt-35-turbo		
Method ↓	Macro F1(%)	BS-F1	R1	R2
PEARL	**65.34**	36.49	0.5082	**0.2676**
no calibrated sm	63.01	**36.69**	**0.5095**	0.2654
distill $p_{\text{aux}}(t_u \mid d_u, q_u)$	62.32	35.64	0.5057	0.2652

Table 4: PEARL compared to alternate training of f_{retr} for gpt-35-turbo.

C.2 Ablating Training Strategies

In Table 4 and 5, we compare common alternative training of f_{retr} while keeping inference steps identical for gpt-35-turbo and davinci-003 respectively. These serve to ablate our specific contributions: selection of training requests which benefit from personalization and our scale calibrating objective.

No calibrated objective Removing the scale calibration and using a standard KL divergence (–calibrated sm) degrades performance for AITA and sees comparable performance in WORKSM with gpt-35-turbo in Table 4. With davinci-003 we see scale calibration consistently improves performance (Table 5). This indicates the importance of calibration for estimating the benefit of a historical document to a request consistently across datasets and LLMs. Appendix C.3 shows scale calibration also consistently improves the correlation of retriever scores with task performance.

Distill $p_{\text{aux}}(t_u|d_u, q_u)$ **to** f_{retr}**.** The proposed f_{retr} is trained on documents which benefit personalization *and* requests which benefit from personalization. Here, we compare to an approach that only selects documents that benefit personalization by maximizing $p_{\text{aux}}(t_u|q_u, d_u)$. This assumes that *all* training requests benefit from personaliza-

Prompt 8 Judge LLM prompt used to select a generated comment more likely to align with a reference comment authored by a user for AITA.

```
You an an experienced linguist who helps
 people compare social media texts.
Given a REFERENCE POST and two
TARGET POSTS judge which of the TARGET
POSTs is significantly more likely to be
written by the same author as the
REFERENCE POST.
For your response use the following
instructions:
1. Make your judgement based on
similarity of stylistic patterns,
arguments, stances, and word choices.
2. Output POST ONE if it is significantly
 more likely to be written by the same
author as the REFERENCE POST.
3. Output POST TWO if it is significantly
 more likely to be written by the same
author as the REFERENCE POST.
4. Output BOTH if either post could have
 been written by the same author or
neither could have been written by the
same author.
Here are the POSTS:
REFERENCE POST: {{reference_post}}
POST ONE: {{post_one}}
POST TWO: {{post_two}}
Output a justification for your
judgement, then output POST ONE, POST
TWO, or BOTH to indicate your final
decision.
```

tion. We train f_{retr} with a KL-divergence objective. This approach, also, closely resembles prior work example selection in non-personalized tasks (Rubin et al., 2022) as well as personalized tasks (Salemi et al., 2024). We see in Table 4 and 5 (distill $p_{\text{aux}}(t_u|d_u, q_u)$) that this lowers performance markedly, indicating the value of our approach.

C.3 Calibration Evaluation

Since we aim to train generation calibrated retrievers, we evaluate calibration performance i.e a retrieval models scores to be predictive of downstream generation performance (Table 6). Here, Pearson r is reported between the top-1 document score for a request and the downstream generation evaluation metric – R2 for WorkSm, and Macro-F1 for AITA. To do this for AITA, we first bin evaluation requests into equal sized bins by top-1 document score, s_1, and then measure Pearson r between the bin start and the average evaluation metric per bin. Our metric is in contrast with prior work (Dhuliawala et al., 2022; Yan et al., 2022) that focuses on classification tasks, where model-predicted class probabilities can be used for measuring calibration, missing in our setup.

Dataset →	AITA	WorkSm		
LLM →	davinci-003	davinci-003		
Method ↓	Macro F1(%)	BS-F1	R1	R2
Pearl	<u>**61.21**</u>	<u>**39.60**</u>	<u>**0.5419**</u>	**0.3094**
no calibrated sm	57.27	38.88	0.5350	0.3033
distill $p_{\text{aux}}(t_u\|d_u, q_u)$	55.52	39.34	0.5359	0.3059

Table 5: Pearl compared to alternate training of f_{retr} for davinci-003.

	Method ↓ / LLM →	davinci-003	gpt-35-turbo
		Pearson r	Pearson r
AITA	BM25	0.08	-0.05
	MPNet-1B	0.07	-0.14
	UPR	-0.48	-0.02
	RelevanceCE	0.07	-0.19
	Pearl f_{retr}	**0.11**	**0.45**
	– calibrated sm	-0.48	0.12
WorkSm	BM25	0.42	0.52
	MPNet-1B	0.54	0.52
	UPR	-0.05	-0.02
	RelevanceCE	0.56	0.49
	Pearl f_{retr}	**0.64**	**0.64**
	– calibrated sm	0.58	0.55

Table 6: Calibration performance of Pearl evaluated through correlation between score for top-1 document and Macro-F1 for AITA, and R2 for WorkSm.

Among baseline methods, we see sparse and dense retrieval methods, BM25 and MPNet-1B scores to be better calibrated with downstream performance compared to likelihood-based methods like QL-FT5. Next, we see Pearl to be better correlated with downstream performance for WorkSm and AITA- indicating the effectiveness of our training strategy. Further, we also report on an ablated model, not using the scale-calibrated objective of Eq (3) (– calibrated sm). We see this approach underperform Pearl, indicating the importance of the scale-calibrated objective for a well-calibrated crossencoder. The poorer calibration of crossencoders also finds support in prior work showing their scores to lie at extremes of the score distribution (Menon et al., 2022; Yadav et al., 2022).

C.4 Selective Revision with Pearl – Extended Results

In §5.3 we demonstrate how our trained retrieval model can be used for selective revision with gpt-35-turbo. Prompt 9, 10 present the prompts

Prompt 9 f_{LLM} prompt used to for selective revision given a Stage 1 draft for AITA.

```
Given a POST from the subreddit Am I the
Asshole and a DRAFT comment from the USER
responding if the author of the POST is
an asshole or not the asshole, edit the
DRAFT comment.
Use the following instructions for your
response:
1. Maintain consistency in tone and
style with the USER's historical
comments.
2. Edit the DRAFT to use more reddit
lingo.
3. Remove statements of the POST from the
 DRAFT.
4. Output a justification for your edits
 starting with the word JUSTIFICATION.
5. Output the edited DRAFT comment
starting with the words EDITED DRAFT.
Here are some historical comments by the
 USER: {{historical_examples}}
REQUEST: {{target_request}}
DRAFT: {{target_draft}}
Output a justification for your edits,
then output the edited DRAFT starting
with the words EDITED DRAFT.
```

used for revision with both LLMs.

In Figure 5, we examine the impact of selective revision in WORKSM for requests of different lengths and users with different number of historical posts. We see that revision benefits requests of medium length and users with few posts. From Figure 5a, we hypothesize that requests that are too short may require additional user input and see no gains from revision. On the other hand requests that are too long, may be more challenging to follow and are unlikely to improve from revisions. From Figure 5b, we see that users with few posts benefit from revision indicating that these users see poorer retrievals. On the other hand users with larger profiles see a drop in performance indicating that even better calibration performance may improve performance of selective revision further.

Note that we don't report results with `davinci-003` since our procedure for learning a threshold θ for selective revision failed to find a threshold where dev set performance was improved from selective revision. Finally note that metrics reported for selective revision in Table 3 isn't directly comparable to those of Tables 1, 4, and 5 since they represent different LLM runs and exclude a dev set from WORKSM and AITA for learning θ (50 and 200 instances respectively).

Prompt 10 f_{LLM} prompt used to for selective revision given a Stage 1 draft for WORKSM.

```
Given a REQUEST and a DRAFT from a USER to
author a social media POST, edit the
DRAFT to satisfy the REQUEST.
Use the following instructions for your
response:
1. Enumerate any missing missing
information from the REQUEST in the DRAFT.
2. Enumerate any irrelevant information
for the REQUEST in the DRAFT.
3. Then output the edited DRAFT starting
with the words EDITED DRAFT.
REQUEST:{{target_request}}
DRAFT: {{target_draft}}
Output missing or irrelevant information
 for the REQUEST, then output the EDITED
DRAFT satisfying the REQUEST.
```

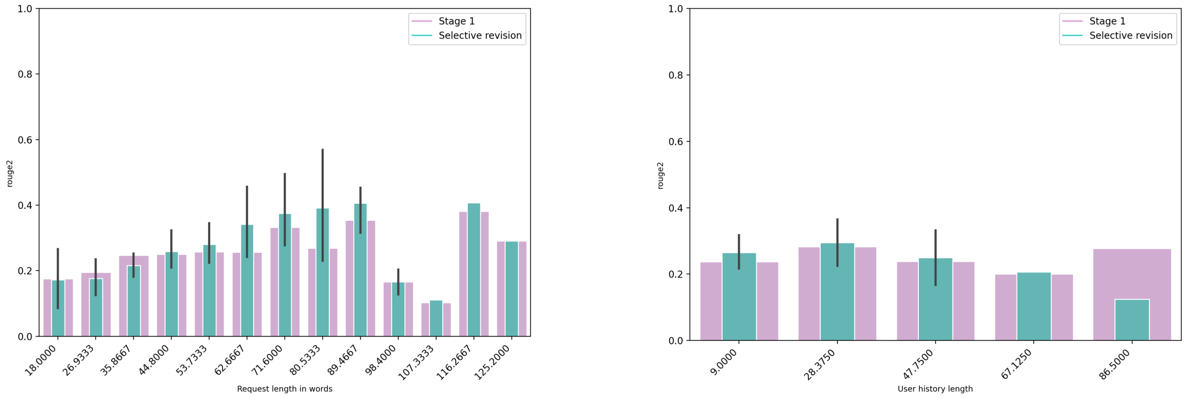
D Extended Related Work

Having discussed the closest body of related work in §2 we discuss additional related work here.

Dynamic prompts for LLMs Besides training retrievers for in-context example selection discussed in §2, other approaches have explored better use of pre-trained models for example selection. Creswell et al. (2023) select examples based on the target LLM likelihood - necessitating access to LLM likelihoods and incurring latency in retrieval. Gupta et al. (2023) explore selecting *sets* of examples with dense retrieval models, presenting a complementary approach to ours. Finally, Pan et al. (2023) use retrieval models to select examples from multiple knowledge sources and train a routing model to decide the source of knowledge to retrieve from – selective revision (§5.3) based on a retriever may be seen as a form of routing.

Prompt robustness in LLMs Simultaneous routing and retrieval also relates our approach to work ensuring that LLMs are robust to noisy retrievals. Prior approaches ensured robustness by only using retrieved documents based on simple frequency filters on entities mentioned in the input query (Mallen et al., 2023) or based on predictions from an NLI model that determines if the query entails the retrieved contexts (Yoran et al., 2023). Other approaches have sought to fine-tune the LLM to be robust to irrelevant contexts (Li et al., 2023c; Luo et al., 2023; Yoran et al., 2023) or modify the decoding procedure (Shi et al., 2023). In contrast, we determine the quality of the input context based on scale-calibrated retrieval model scores.

LLM chaining In selectively editing generations



(a) Effectiveness of selective revision for requests of different lengths (in words).

(b) Effectiveness of selective revision for users of different numbers of historical posts.

Figure 5: The impact of selective revision (§5.3) in PEARL on WORKSM compared for requests of different length and users with varying number of historical posts.

with an LLM for low-performing requests, our approach also relates to recent work on composing LLMs with other models to build more complex systems (Wu et al., 2022; Arora et al., 2023; Khattab et al., 2023). Close work is presented by approaches that leverage repeated LLM calls to verify the reasoning or factuality of previous generations (Shridhar et al., 2023; Dhuliawala et al., 2023). In contrast, our work leverages an efficient retrieval model to selectively direct low-performing generations for further revision, reducing the total number of expensive LLM calls necessary. In this respect, our approach bears similarity to Zhang et al. (2023b), who progressively run larger LLMs only when necessary for an input.

Calibrated retrievers A small body of work has explored calibrated ranking models. Yan et al. (2022) train scale-calibrated ranking models for recommendation models used for advertisement pricing systems. On the other hand, our work leverages scale-calibration for personalized writing assistance. Other work has explored joint training of retrievers and generative models to obtain calibrated retrievers (Dhuliawala et al., 2022), using Gaussian embeddings to estimate retriever uncertainty (Zamani and Bendersky, 2023), or estimating retriever confidence with monte-carlo dropout (Cohen et al., 2021). In contrast with probabilistic uncertainty estimation, PEARL minimally modifies training to result in a calibrated model and does not require extensive changes to training, model architecture, or additional inference costs.

Writing assistants A sizable body of work has explored the development of writing assistants. Compared to assistants for communication applications, these have been targeted at authors of creative texts like screenplays (Mirowski et al., 2023), stories (Akoury et al., 2020), and poems (Gonçalo Oliveira, 2017) – consequently, they focus on diverse generations and long-range coherence, rather than personalization. Further, while our work leverages a request-driven assistant, prior systems have used a variety of interaction and control methods. While text completion presents a common form of interaction (Clark et al., 2018), recent work has seen use of infilling, tag-based control (Sun et al., 2021), and instruction guided generations (Chakrabarty et al., 2022) – a deeper examination of control and interaction strategies and their trade offs are presented in related reviews (Zhang et al., 2023a; Lin et al., 2023). While our approach to personalization may be extended to some alternative interaction paradigms, other interaction techniques are likely to necessitate additional work.

Personalized text generation While we have focussed on author personalization that aims to mimic stylistic patterns, interests, and values of an author, we briefly review reader-personalized text generation – a setup aiming to generate texts that are engaging and relevant to readers' preferences. This has historically been explored for generating personalized reviews (Ni et al., 2017), recipes (Majumder et al., 2019), news headlines (Ao et al., 2021) and in dialogue agents (Mazaré et al., 2018; Zhang et al., 2018). Related work is also found in text simplification and lay summarization in the context of scientific text – this work has explored generating definitions for scientific concepts at varying levels of complexity (August et al., 2022; Murthy et al., 2022) or summarizing scientific text for lay readers (Guo et al., 2021). While recent work has explored this with modern LLMs (Li et al., 2023d; Farajidizaji et al., 2023), reader personalization remains an understudied problem and presents a rich area for future work.

E Limitations

Here, we discuss limitations of our work derive from our choice of f_{aux} and f_{LLM}, our evaluation setup, and the design of our method.

Choice of LLMs Our experiments use two closed LLMs through API access (`davinci-003`, `gpt-35-turbo`). While we show the value of PEARL with LLM's of varying performance, establishing its effectiveness with other LLMs will require further work. We also acknowledge that closed LLMs limit experimental reproducibility - however, given the widespread use of GPT models (Hu, 2023) we believe our investigation is meaningful. Finally, in constructing training data for instance selection models for an LLM, prior work has noted the best empirical performance from matching f_{aux} and f_{LLM} (Rubin et al., 2022). While we demonstrate benefits from using significantly smaller models for f_{aux}, using an open LLM will allow further validation of this result in the context of our approach. However, using a larger (open) model for f_{aux} will incur additional costs in creating training data, and smaller models for f_{LLM} are likely to see a worse generation performance - exploring this tradeoff requires future work.

Evaluation setup Next, while WORKSM represents an impactful and realistic use case for writing assistants, we acknowledge that its private nature limits reproducibility. Further, our evaluation set of WORKSM and AITA represents a limited set of scenarios that are likely to leverage writing assistants. While we believe our work represents a meaningful first step, additional future work, and online evaluations are necessary to establish the value of PEARL across the myriad of scenarios where writing assistants may be used. Finally, while we leverage several evaluation strategies to demonstrate the value of PEARL, evaluating text generations under personalization setups represents is an under-explored and a currently emerging body of work (Wang et al., 2023a,d).

Method design Finally, we note that the current design of PEARL is likely to have some drawbacks. It is possible that our proposed method for training instance selection biases system performance toward some users or requests – we leave examination of this to future work. It is also possible that formulating f_{retr} as an expressive crossencoder and the use of large LLMs will present latency limitations for interactive applications – exploration of models supporting faster retrieval and text generation inference represent important future work.

Prompt 11 GPT4 prompt used to generate synthetic requests for WorkSm posts in our training set.

```
## TASK
Given an enterprise social media post, generate a set of writing instructions that
explain how to
”reverse-engineer”; the post. Use the following steps:
- The instructions must give a high-level overview of what the post aims to
communicate. Example: [readcted]
- The instructions must include specific proper nouns (people, places, organzations)
. Example: [redacted]
- The instructions must include contact information if available. Example: [redacted
]
- The instructions must include specific links to websites or files if available.
Example: [redacted]
- The instructions must contain any knowledge that is highly specialized and is
likely to be only known to the individual who wrote the post, if available. Example:
 [redacted]
- The instructions must contain rough sketches of any personal anecdotes in the post
, if available. Example: [redacted]
- The instruction must **not** contain any formatting or ordering information from
the post.

## OUTPUT
Output the following:
<Explanation>{explanation of your reasoning for how you generated the instructions,
in 3 sentences or fewer}</Explanation>
<Instruction.Overview>{1-2 sentences overview of what the post aims to communicate
}</Instruction.Overview>
<Instruction.Names>{1-2 sentences about the people, places, or organizations
mentioned in the post, _NONE_ if not applicable}</Instruction.Names>
<Instruction.Contacts>{1-2 sentences about the contact information copied verbatim
in the post, _NONE_ if not applicable}</Instruction.Contacts>
<Instruction.Links>{1-2 sentences including the links copied verbatim from the post,
 _NONE_ if not applicable}</Instruction.Links>
<Instruction.Knowledge>{1-2 sentences paraphrasing the specialized knowledge
included in the post, _NONE_ if not applicable}</Instruction.Knowledge>
<Instruction.Anecdotes>{1-2 sentences paraphrasing the anecdotes included in the
post, _NONE_ if not applicable}</Instruction.Anecdotes>

## INPUT
{{input_post}}
```

Guideline 1 Instructions provided to annotators for authoring requests for our evaluation set in WORKSM.

Overview:
In this study, we are developing LLM-based approaches for writing social media posts on enterprise social networks. Your task is as follows: Given a social media post from an enterprise social media platform, write a short outline of the post. In writing your outline, imagine you are a manager, social media manager, or event organizer writing a rough sketch of the post with the key information you would like to share.

Data Format:
You are given a spreadsheet consisting of ~150 English posts. Each row corresponds to a single post. The spreadsheet contains the following columns: PostId, InputPost, OutputShortOutline. The first column is the ID of the post; you can ignore this column. The second column is the full text of the input post. In the third column, you will write your short outline based on the input post.

DO's for your outline:
When writing your short outline, do include the following:
- One sentence about the goal of the post: Include a brief description of what the post is trying to communicate. Example: [redacted]
- Specific proper nouns (people, places, things): Include names of specific people, places, or things in your outline. Example: [redacted]
- Specialized knowledge: If the knowledge contained in the post is highly specialized and is likely to be only known to the individual writing the post, include a rough sketch of that information in your outline. Example: [redacted]
- Personal anecdotes: If the post contains specific personal anecdotes, include a rough sketch of that information in your outline. Example: [redacted]
- Special emphasis or call to action: If the post makes a special emphasis, include a rough sketch of that emphasis or call to action in your outline. Example: [redacted]
- External website links: If the post links to an external website, include the link in your outline. Example: [redacted]

DONT's for your outline:
When writing your short outline, do not include the following:
- Anything related to the ordering of content.
- Formatting instructions.
- Any verbatim text other than specific proper nouns.