LaMP: When Large Language Models Meet Personalization

Alireza Salemi[1], Sheshera Mysore[1], Michael Bendersky[2], Hamed Zamani[1]
[1]University of Massachusetts Amherst
[2]Google Research
{asalemi,smysore,zamani}@cs.umass.edu
bemike@google.com

The LaMP Benchmark: http://lamp-benchmark.github.io/

Abstract

This paper highlights the importance of personalization in large language models and introduces the LaMP benchmark — a novel benchmark for training and evaluating language models for producing personalized outputs. LaMP offers a comprehensive evaluation framework with diverse language tasks and multiple entries for each user profile. It consists of seven personalized tasks, spanning three text classification and four text generation tasks. We additionally propose two retrieval augmentation approaches that retrieve personal items from each user profile for personalizing language model outputs. To this aim, we study various retrieval models, including term matching, semantic matching, and time-aware methods. Extensive experiments on LaMP for zero-shot and fine-tuned language models demonstrate the efficacy of the proposed retrieval augmentation approach and highlight the impact of personalization in various natural language tasks.

1 Introduction

The recent development of large language models (LLMs) has revolutionized natural language processing (NLP) applications. As the use of LLMs, such as GPT-4 (OpenAI, 2023), in real-world applications evolves, personalization emerges as a key factor in meeting the user's expectations for tailored experiences that align with their unique needs and preferences (Huang et al., 2022). Personalization has been widely studied by various communities, including the information retrieval (IR) and human-computer interaction (HCI) communities, often with applications to search engines and recommender systems (Fowler et al., 2015; Xue et al., 2009; Naumov et al., 2019). Recent work has also highlighted the impact and concerns associated with personalizing LLMs and tying it to ongoing work on alignment (Kirk et al., 2023). Despite this and the importance of personalization in many real-world problems, developing and evaluating LLMs for producing personalized responses remain relatively understudied. To bridge this gap, this paper underscores the importance of personalization in shaping the future of NLP systems and takes a first step towards developing and evaluating personalization in the context of large language models by introducing the LaMP benchmark[1] — a comprehensive and diverse benchmarks of personalized text classification and generation tasks.

While many existing well-known NLP benchmarks, such as GLUE (Wang et al., 2018), SuperGLUE (Wang et al., 2019), KILT (Petroni et al., 2021), and GEM (Gehrmann et al., 2021) have led to significant progress in various NLP tasks, they have often taken the dominant NLP approach of "one-size-fits-all" to modeling and evaluation, and do not allow the development of models that adapt to the specific needs of end users – limiting extensive research on personalization in NLP tasks. In contrast, LaMP offers a comprehensive evaluation framework incorporating diverse language tasks that require personalization. LaMP consists of three personalized text classification tasks: (1) Personalized Citation Identification (binary classification), (2) Personalized Movie Tagging (categorical classification with 15 tags), and (3) Personalized Product Rating (ordinal classification from 1 to 5-star rating for e-commerce products). Further, LaMP includes four text generation datasets: (4) Personalized News Headline Generation, (5) Personalized Scholarly Title Generation, (6) Personalized Email Subject Generation, and (7) Personalized Tweet Paraphrasing. For these seven tasks, we explore the two dominant settings in personalization: (a) personalization for new users with a user-based data split and (b) personalization for future interactions of existing users with a time-based data split. Therefore, LaMP provides a rich environment

[1]LaMP stands for Language Model Personalization.

for developing personalized NLP models. To foster research in this area, we release the LaMP benchmark, the data construction, evaluation scripts, and a leaderboard.

For personalizing the language model outputs, a straightforward solution is to incorporate the user profile into a language model prompt. However, user profiles are often large and exceed the length limitations of large language models. Even as such limitations are relaxed with evolving technology, the cost of processing large input sequences is considerable. Therefore, we propose two retrieval augmentation solutions for LLM personalization, in which for each test input, we retrieve items from the user profile to be included in the LLM prompt for personalization. The first approach uses in-prompt augmentation (IPA) for personalization, and the second approach encodes each personal item separately and integrate them later in the decoder using the fusion-in-decoder model of Izacard and Grave (2021). We demonstrate that using this approach, the performance of language models improves on all datasets in the LaMP benchmark. Based on this retrieval augmentation solution, we evaluate different retrievers for personalized prompt construction and establish benchmark results for fine-tuned and zero-shot language models. The empirical findings of our research reveal that the process of fine-tuning a language model utilizing our personalized augmentation technique yields a noteworthy relative average enhancement of 23.5% across the benchmark. Even in zero-shot settings where an off-the-shelf LLM without fine-tuning (e.g., FlanT5-XXL) is used, utilizing our proposed method results in a relative average improvement of 12.2% across the tasks. Finally, this paper smooths the path towards developing advanced user-centric NLP systems.

2 The LaMP Benchmark

Problem Formulation. Generative language models often take an input x and predict the most probable sequence tokens y that follows x. Personalizing language models can be defined as conditioning the model's output on a user u, represented by a user profile. In LaMP, we define user profile as the user's historical data, i.e., the past input and personalized outputs produced by or approved by the user, $P_u = \{(x_{u1}, y_{u1}), (x_{u2}, y_{u2}), \cdots, (x_{um_u}, y_{um_u})\}$. Therefore, each data entry in the LaMP benchmark consists of three components: an input sequence x that serves as the model's input, a target output y that the model is expected to produce, and a profile P_u that encapsulates any auxiliary information that can be used to personalize the model for the user.

Overview of LaMP. Given the above problem formulation, we develop the LaMP benchmark that aims to assess the efficacy of LLMs in producing personalized outputs y, based on inputs x and user-specific information P_u. Outputs y of different types result in seven diverse tasks spanning personalized text classification and generation:

- **Personalized Text Classification**
  - (1) Personalized Citation Identification
  - (2) Personalized Movie Tagging
  - (3) Personalized Product Rating
- **Personalized Text Generation**
  - (4) Personalized News Headline Generation
  - (5) Personalized Scholarly Title Generation
  - (6) Personalized Email Subject Generation
  - (7) Personalized Tweet Paraphrasing

2.1 Tasks Definitions

Next, we overview of each task used in LaMP and detail data construction in Appendix A.

LaMP-1: Personalized Citation Identification

The citation behavior of researchers is dependent on their interests and is commonly used to evaluate and develop personalized systems for recommending papers (Färber and Jatowt, 2020). This task recasts citation recommendation as a binary classification task and assesses the ability of a language model to identify user preferences for citations. Specifically, if the user u writes a paper x, a language model must determine which of two candidate papers u will cite in x (see Figure 3).

To generate data samples, we leverage the Citation Network Dataset (V14) (Tang et al., 2008), which comprises information on scientific papers, authors, and citations. For this task, the profile of each user encompasses all the papers they have authored. We retain only the title and abstract of each paper in the user's profile.

LaMP-2: Personalized Movie Tagging

Users tagging behavior for media such as movies and books are known to be idiosyncratic and depends on their understanding of the tag and the aspects of the item they focus on. This has motivated a large body of work on personalized tagging

Task	Type	Separation	#Train	#Dev	#Test	Input Length	Output Length	Profile Size	#Classes
Citation Ident.	binary classification	user	9682	2500	2500	51.40 ± 5.72	-	90.61 ± 53.87	2
		time	6542	1500	1500	51.43 ± 5.70		84.15 ± 47.54	
Movie Tag.	categorical classification	user	3820	692	870	92.27 ± 20.83	-	159.29 ± 330.81	15
		time	5073	1410	1557	92.39 ± 21.95		86.76 ± 189.52	
Product Rat.	ordinal classification	user	20000	2500	2500	145.14 ± 157.96	-	188.10 ± 129.42	5
		time	20000	2500	2500	128.18 ± 146.25		185.40 ± 129.30	
News Headline	text generation	user	12527	1925	2376	30.53 ± 12.67	9.78 ± 3.10	287.16 ± 360.62	-
		time	12500	1500	1800	29.97 ± 12.09	10.07 ± 3.10	204.59 ± 250.75	
Scholarly Title	text generation	user	9682	2500	2500	152.81 ± 86.60	9.26 ± 3.13	89.61 ± 53.87	-
		time	14682	1500	1500	162.34 ± 65.63	9.71 ± 3.21	87.88 ± 53.63	
Email Subject	text generation	user	4840	1353	1246	436.15 ± 805.54	7.34 ± 2.83	80.72 ± 51.73	-
		time	4821	1250	1250	454.87 ± 889.41	7.37 ± 2.78	55.67 ± 36.32	
Tweet Para.	text generation	user	10437	1500	1496	29.76 ± 6.94	16.93 ± 5.65	17.74 ± 15.10	-
		time	13437	1498	1500	29.72 ± 7.01	16.96 ± 5.67	15.71 ± 14.86	

Table 1: Data statistics of the tasks in the LaMP benchmark. Each dataset in the LaMP benchmark has two evaluation settings: (a) user-based data split to test personalization for new users and (b) a time-based data split to test personalization for future interactions of existing users.

(Gupta et al., 2010). We use this task to evaluate the ability of language models to make tag assignments for a movie contingent on the user's historical tagging behavior. Specifically, given a movie description x and a user's historical movie-tag pairs, a language model must predict one of 15 tags for x. We obtain tag assignments from the MovieLens dataset (Harper and Konstan, 2015). Additionally, we obtain movie descriptions from MovieDB.[2]

LaMP-3: Personalized Product Rating

Product reviews commonly express a nuanced set of user preferences for a product and which in turn determine their rating for the product. Predicting ratings based on user reviews has been studied extensively in personalized sentiment prediction tasks (Mireshghallah et al., 2022). While this is commonly treated as a regression task, to use autoregressive language models, we frame it as a multi-class classification task. Specifically, given the user u's historical review and rating pairs and an input review x, the model must predict an integer rating from $1-5$. We construct our dataset from a dataset of Amazon reviews (Ni et al., 2019).

LaMP-4: Personalized News Headline Generation

Authors writing displays distinct stylistic elements influenced by both personal and social factors (Zhu and Jurgens, 2021). Journalists authoring headlines are likely to balance between faithfully representing an article, appealing to their readers, and maintaining their own identity. This offers a useful testbed for personalized text generation. Here, we evaluate the ability of a language model to capture the stylistic patterns of an author by requiring it to generate a headline for an input news article, x, given a user profile of the authors' historical article-title pairs. To create a dataset, we use a collection of Huffington Post articles (Misra, 2022; Misra and Grover, 2021).

LaMP-5: Personalized Scholarly Title Generation

As with LaMP-4, the generation of titles for research articles offers a test bed for personalized text generation but varies in text domain. In this task, we require language models to generate titles for an input article x, given a user profile of historical article-title pairs for an author. Here, only use article abstracts. We create our dataset from the Citation Network Dataset (V14) (Tang et al., 2008) also used for LaMP-1.

LaMP-6: Personalized Email Subject Generation

Similar to LaMP-4 and 5, generating email subjects also provides a valuable test bed for personalized text generation. Email assistance is also known to be a task that significantly benefits from personalization (Trajanovski et al., 2021). Here, we require language models to generate an email subject for an input email message x, given historical email-subject pairs authored by a user. For this task, we leverage a private dataset of emails the Avocado Research Email Collection (Oard, Douglas et al., 2015). Given its private nature this is unlikely to be contained in pre-training data providing a meaningful challenge for language models.

[2] https://www.themoviedb.org/

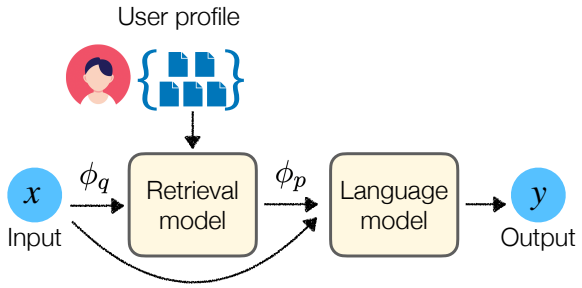

Figure 1: An overview of the retrieval-augmented method for personalizing LLMs. ϕ_q and ϕ_p represent query and prompt construction functions.

LaMP-7: Personalized Tweet Paraphrasing

Social media posts adhere strongly to various personal stylistic patterns of authors (Zhu and Jurgens, 2021). Here, we construct a personalized tweet paraphrasing task and require models to generate a tweet in the style of a user given an input tweet x, and a user profile of historical tweets by the user. To construct this task we use data from the Sentiment140 dataset (Go et al., 2009). Figure 3 provides examples of tasks 1-7 in LaMP.

2.2 Data Splits

To enable evaluation in common personalization settings, LaMP offers two different data splitting settings: (1) user-based separation and (2) time-based separation. In user-based separation (denoted as LaMP-iU for Task i), train/validation/test splits are made by partitioning across users, ensuring that no shared users appear across splits. This strategy measures personalization for new users.

In time-based separation (denoted as LaMP-iT for Task i), train/validation/test splits are made by partitioning user items ordered by time. The most recent user items are chosen to create the input-output pairs, with older items serving as user profiles. Appendix A contains additional details, and Table 1 reports dataset sizes.

2.3 Evaluation

For evaluating classification tasks, we use Accuracy for LaMP-1 (balanced binary classification), Accuracy/F1 for LaMP-2 (multi-class classification), and MAE/RMSE for LaMP-3 (ordinal multi-class classification). Following previous works (Zhou and Bhat, 2021; Panthaplackel et al., 2022) on text generation, we use Rouge-1/Rouge-L (Lin, 2004) as evaluation metrics for the text generation generation tasks (LaMP-4 to LaMP-7).

3 Retrieval Augmentation for Personalizing LLMs

To personalize a language model two broad strategies may be explored: (1) fine-tuning the LM for each user and (2) prompting a shared LM with user specific input or context. The former approach necessitates substantial computational resources, especially for fine-tuning larger LLMs. Moreover, accommodating personalized LLMs for each user in industry-scale systems encompassing millions or billions of users necessitates a significant storage and serving capacity. Therefore, we focus on developing strategies for training models personalized via user-specific inputs.

Each task in LaMP, each user profile consists of a potentially large collection of data points. Given the inherent context length constraint of many LLMs and the cost of processing long sequences, we incorporate a subset of these data points as input prompts. Further, not all entries within a user profile are necessarily relevant to the specific input at hand. To do this, we propose solutions based on retrieval augmentation (See Figure 1). This framework selectively extracts pertinent information from the user profile that is relevant to the current unseen test case and generates model predictions conditioned on this information.

Specifically, for a given sample (x_i, y_i) for user u, we employ three primary components: (1) a query generation function ϕ_q that transforms the input x_i into a query q for retrieving from the user u's profile, (2) a retrieval model $\mathcal{R}(q, P_u, k)$ that accepts a query q, a user profile P_u and retrieves k most pertinent entries from the user profile, and (3) a prompt construction function ϕ_p that assembles a personalized prompt for the user u based on input x_i and the retrieved entries. For retrieval augmentation, we explore two strategies: **(1) In-Prompt Augmentation (IPA)** and **(2) Fusion-in-Decoder (FiD)** (Izacard and Grave, 2021). The input to both approaches constructs inputs, $\bar{x}_i$, using $\mathcal{R}$ to select k items from the user profile P_u:

$$\bar{x}_i = \phi_p(x_i, \mathcal{R}(\phi_q(x_i), P_u, k)) \quad (1)$$

where we use $(\bar{x}_i, y_i)$ to train or evaluate the language models. With FiD, LLMs receive multiple inputs, each of which is encoded separately within its encoder. These separate encodings are then merged together in the decoder. Here, the inputs $\{\bar{x}_{i1}, ..., \bar{x}_{ik}\}$ for the encoder are derived as:

$$\bar{x}_{ij} = \phi_p(x_i, d_{ij}) \quad (2)$$

where d_{ij} is the j'th retrieved item using retrieval model from the user profile (i.e, $\mathcal{R}(\phi_q(x_i), P_u, k)$). Note that, IPA and FiD offer different tradeoffs. FiD necessitates training of the language model while IPA may be applied without training. Further, while FiD can only be used with encoder-decoder models, IPA can be used across architectures. However, FiD allows us to incorporate more items from the user profile into the LLM's input.

We explore various choices for the retrieval model $\mathcal{R}$. In our experiments, we study a strong term matching model, BM25 (Robertson et al., 1995), a state-of-the-art pre-trained dense retrieval model, Contriever (Izacard et al., 2022), a retrieval model that returns most recent profile entries in descending order (i.e., Recency), and a Random document selector from the user profile. The prompt construction function ϕ_p `concatenates` the instruction for each task, the input sequence, and the user profile. The specific prompts are presented in Table 5. For the ϕ_q function, we use the target input for each task as the query (see Figure 3).

4 Experiments

This section describes our experiments, results, and findings on the LaMP benchmark.

4.1 Experimental Setup

For training FiD and the generative model in IPA we leverage AdamW (Loshchilov and Hutter, 2019) with a learning rate of 5×10^{-5} and a batch size 64 and set 5% of the total training steps as warmup using a linear scheduler. A weight decay of 10^{-4} is incorporated to prevent overfitting. The maximum input and output lengths is set to 512 and 128 tokens, respectively. We train both classification and generation models for 10 and 20 epochs, respectively. A FlanT5-base[3] (Chung et al., 2022) model is used for all experiments, unless explicitly stated otherwise (in experiments with LLMs, we use FlanT5-XXL). We employ beam search (Freitag and Al-Onaizan, 2017) with a beam size of 4. All models are implemented with Huggingface `transformers` and evaluations are conducted using the `evaluate` library. All the experiments are conducted on a single Nvidia RTX8000 GPU with 49GB of GPU memory and 128GB of CPU memory for maximum 3 days on each experiment. All the results reported are based on a single run.

[3] https://huggingface.co/google/flan-t5-base

4.2 Fine-Tuning Retrieval Augmented LMs for Personalization

In the first sets of experiments, we establish baseline personalization results for a fine-tuned language model. We also investigate the impact of employing various retrieval techniques and the effect of retrieving different quantities of entries from a user profile. This analysis aims to provide insights into the efficacy of diverse retrieval methods and the potential benefits of adjusting the number of retrieved entries for personalization tasks.

Impact of Retrievers on Retrieval-Augmented Personalization Models. Here, we study different implementations of $\mathcal{R}$ with a fine-tuned FlanT5-base model for generating personalized output: (1) a baseline random selector from the user profile, (2) BM25 (Robertson et al., 1995), (3) Contriever[4] (Izacard et al., 2022), and (4) Recency, in which we select the latest item in the user profile based on time (only for time-based separation setting). BM25 is considered as a robust and strong term-matching retrieval model and Contriever is a pre-trained dense retrieval model.

The results of this experiment are shown in Table 2 for user-based separation and in Table 3 for time-based separation. The results suggest that personalization improves the performance for all tasks within the LaMP benchmark. In most cases, even a random selection of documents from the user profile and the creation of personalized prompts leads to performance improvements compared to non-personalized prompts given to the LM. Note that non-personalized prompt can be achieved with no retrieval augmentation (No-Retrieval) or with augmentation with a random item from all user profiles. Results for more non-personalized baselines are presented in Appendix E.

When retrieving one document per user for personalizing the language model's output, Contriever demonstrates the best performance for most classification tasks (i.e., LaMP-1U, LaMP-2U, LaMP-3U, LaMP-1T, and LaMP-2T). Recency only outperforms Contriever in the LaMP-3T. Note that recency is considered as a simple yet strong personalization signal in search and recommendation (Fader et al., 2005; Reinartz and Kumar, 2000, 2003). For text generation, Contriever performs best for Personalization News Headline Generation (LaMP-4U) and Personalized Tweet Paraphrasing

[4] https://huggingface.co/facebook/contriever

		FlanT5-base (fine-tuned)						
		Non-Personalized		Untuned profile, $k = 1$			Tuned profile	
Dataset	Metric	No-Retrieval	Random	Random	BM25	Contriever	IPA	FiD($k = 16$)
LaMP-1U: Personalized Citation Identification	Accuracy ↑	0.518	0.539	0.598	0.649	0.688	0.734	**0.754**
LaMP-2U: Personalized Movie Tagging	Accuracy ↑	0.468	0.442	0.497	0.524	0.536	0.556	**0.642**
	F1 ↑	0.435	0.403	0.459	0.480	0.506	0.519	**0.607**
LaMP-3U: Personalized Product Rating	MAE ↓	0.275	0.286	0.284	0.258	0.248	0.246	**0.236**
	RMSE ↓	0.581	0.607	0.602	0.573	0.563	0.565	**0.539**
LaMP-4U: Personalized News Headline Generation	ROUGE-1 ↑	0.153	0.159	0.162	0.167	0.173	**0.186**	0.180
	ROUGE-L ↑	0.140	0.147	0.148	0.153	0.159	**0.171**	0.166
LaMP-5U: Personalized Scholarly Title Generation	ROUGE-1 ↑	0.418	0.408	0.409	0.440	0.431	**0.450**	0.431
	ROUGE-L ↑	0.378	0.370	0.371	0.399	0.393	**0.409**	0.392
LaMP-6U: Personalized Email Subject Generation	ROUGE-1 ↑	0.379	0.473	0.486	0.586	0.572	**0.587**	0.567
	ROUGE-L ↑	0.358	0.457	0.470	0.570	0.558	**0.575**	0.555
LaMP-7U: Personalized Tweet Paraphrasing	ROUGE-1 ↑	0.509	0.510	0.514	0.521	0.524	**0.528**	0.517
	ROUGE-L ↑	0.455	0.457	0.460	0.468	0.471	**0.475**	0.464

Table 2: The results for a fine-tuned LM on the test set of the user-based setting. The number of retrieved document for personalizing LM is denoted by k. Details for tuning the profile on validation sets is in Table 6 in Appendix D.

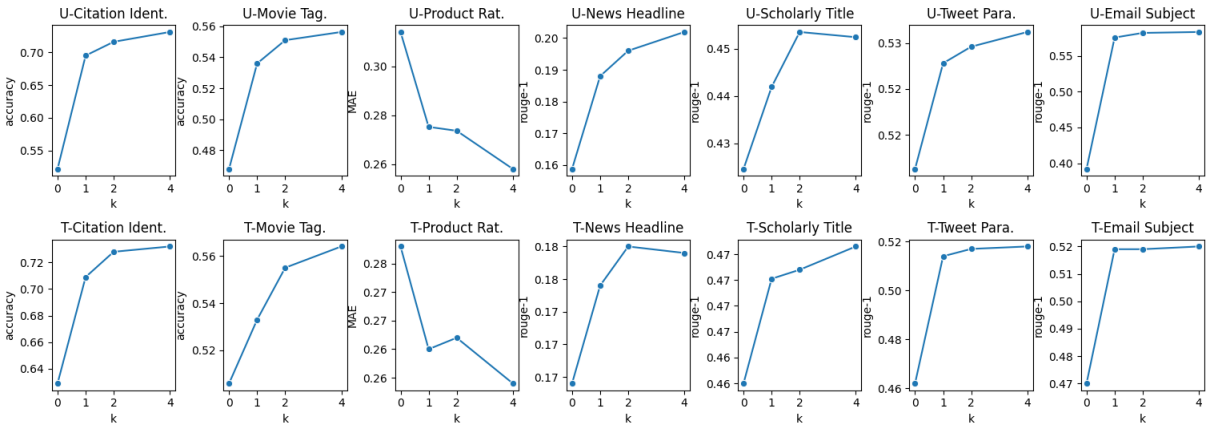

Figure 2: The performance on downstream tasks using the best retriever for each task from Tables 2 and 3 with different numbers of retrieved entries, k, from the user profile. The results indicate that increasing the number of retrieved documents for most datasets results in a better personalized performance.

(LaMP-7U) in user-based separation setting. For Email Generation and Scholarly Title Generation tasks (LaMP-5U and LaMP-6U), BM25 demonstrates superior performance. Both BM25 and Contriever outperform a random profile selector in all LaMP datasets. For the time-based separation setting, Contriever outperforms other methods in all generation tasks except News Headline Generation (LaMP-4T), where recency performs better.

Generally, the results indicate that incorporating any information from the user profile into the input is not sufficient, but rather selecting the most relevant and/or recent information is crucial. This underscores the importance of careful consideration in selecting and incorporating pertinent user profile elements in LLM prompts. There is no clear winner among the retrieval models we study and an ensemble of relevance and temporal signals for personalization should be studied in the future.

Impact of the Number of Retrieved Items, k, on LLM Personalization. Each sample within this benchmark consists of a substantial number of user profile entries. As such, exploring the impact of incorporating multiple entries to augment the input of the language model can provide valuable insights into addressing the unresolved challenges posed by this benchmark. For the sake of space, we focus on our In-Prompt Augmentation (IPA) ap-

		FlanT5-base (fine-tuned)							
		Non-Personalized		Untuned profile, $k = 1$				Tuned profile	
Dataset	Metric	No-Retrieval	Random	Random	BM25	Contriever	Recency	IPA	FiD($k = 16$)
LaMP-1T: Personalized Citation Identification	Accuracy ↑	0.628	0.625	0.657	0.682	0.688	0.691	**0.714**	0.698
LaMP-2T: Personalized Movie Tagging	Accuracy ↑	0.506	0.513	0.518	0.539	0.533	0.549	0.564	**0.661**
	F1 ↑	0.443	0.449	0.456	0.472	0.475	0.492	0.519	**0.624**
LaMP-3T: Personalized Product Rating	MAE ↓	0.280	0.280	0.279	0.278	0.281	0.279	0.266	**0.250**
	RMSE ↓	0.615	0.616	0.612	0.614	0.606	0.608	**0.598**	**0.598**
LaMP-4T: Personalized News Headline Generation	ROUGE-1 ↑	0.159	0.160	0.169	0.171	0.176	0.173	**0.177**	0.170
	ROUGE-L ↑	0.145	0.147	0.155	0.157	0.162	0.158	**0.162**	0.157
LaMP-5T: Personalized Scholarly Title Generation	ROUGE-1 ↑	0.462	0.459	0.460	0.471	0.472	0.466	**0.479**	0.456
	ROUGE-L ↑	0.416	0.412	0.414	0.423	0.426	0.420	**0.431**	0.414
LaMP-6T: Personalized Email Subject Generation	ROUGE-1 ↑	0.479	0.500	0.525	0.537	0.545	0.532	**0.547**	0.540
	ROUGE-L ↑	0.463	0.452	0.507	0.522	0.530	0.518	**0.533**	0.525
LaMP-7T: Personalized Tweet Paraphrasing	ROUGE-1 ↑	0.462	0.474	0.505	0.508	0.505	0.503	**0.516**	0.502
	ROUGE-L ↑	0.416	0.457	0.456	0.457	0.455	0.453	**0.465**	0.450

Table 3: The results for a fine-tuned LM on the test set of the time-based setting. The number of retrieved document for personalizing LM is denoted by k. Details for tuning the profile on validation sets is in Table 8 in Appendix D.

proach for personalization and depict the model's performance w.r.t different profile sizes in Figure 2. This experiment uses the best retriever from Tables 2 and 3 across various tasks, while varying the number of retrieved entries from user profiles. The results suggest that increasing the number of retrieved items leads to improved performance in downstream tasks. However, some tasks experience a decline in performance. Given the finite context size of language models, exploring approaches to generate a unified prompt from multiple user entries may represent promising future work.

Impact of Tuning Retriever Hyperparameters. Based on the performance on the validation set for each dataset, we tuned two parameters: (1) the retrieval model (BM25 vs. Contriever vs. Recency) for IPA and FiD, and (2) retrieved items count (k) for IPA. We consistently utilize 16 documents for FiD, as we do not observe much variance in the results. Both IPA and FiD approaches use FlanT5-base. For hyperparameter tuning, we used the following metrics on the development sets: Accuracy for LaMP-1 and LaMP-2, MAE for LaMP-3, and ROUGE-1 for all text generation tasks. The results for this tuned model are presented in the last two columns of Table 2 and Table 3. As expected, the tuned model outperforms the other models on all datasets. For text classification tasks, FiD surpasses the performance of IPA in all datasets, with the exception of LaMP-1T. Conversely, IPA exhibits superior performance across all text generation datasets.

4.3 Zero-Shot Personalized Results for LLMs

With the widespread adoption of employing LLMs with no fine-tuning in contemporary research, we conduct an evaluation of two such models on our benchmark.[5] Particularly, we leverage GPT 3.5 (alias `gpt-3.5-turbo` or ChatGPT[6]) and FlanT5-XXL (Chung et al., 2022). FlanT5-XXL comprises 11B parameters, however, the size of GPT-3.5 is unknown (GPT3 consists of 175B parameters). For evaluation, we provide each model with the inputs corresponding to individual tasks and assess their performance based on the generated outputs. In classification tasks, if the produced output does not correspond to a valid class, we resort to calculating the similarity between each class label and the generated output utilizing BERTScore (Zhang* et al., 2020). Thus, we assign the most similar label to the generated output as the output for the given input. GPT-3.5 generated out-of-the-label predictions 8%, 4%, 6%, 4%, 2%, and 4% of the time for the LaMP-1U, LaMP-1T, LaMP-2U, LaMP-2T, LaMP-3U, and LaMP-3T tasks, respectively. On the other hand, FlanT5-XXL predictions are consistently among the questioned labels.

Table 4 shows the result of LLMs on this benchmark in a zero-shot scenario. The results show that, except for the Personalized Tweet Paraphrasing task, using the user's profile with LLMs im-

[5] As previously stated, FiD approach cannot be utilized with untrained models. Consequently, the experiments conducted in this section pertain solely to IPA method.

[6] https://openai.com/blog/chatgpt

		User-based Separation				Time-based Separation			
		Non-Personalized		Personalized		Non-Personalized		Personalized	
Dataset	Metric	FlanT5-XXL	GPT-3.5	FlanT5-XXL	GPT-3.5	FlanT5-XXL	GPT-3.5	FlanT5-XXL	GPT-3.5
LaMP-1: Personalized Citation Identification	Accuracy ↑	0.520	0.541	**0.699**	0.695	0.502	0.508	**0.636**	0.634
LaMP-2: Personalized Movie Tagging	Accuracy ↑	0.365	0.408	0.414	**0.508**	0.360	0.382	0.396	**0.466**
	F1 ↑	0.308	0.314	0.364	**0.457**	0.276	0.299	0.304	**0.418**
LaMP-3: Personalized Product Rating	MAE ↓	0.344	0.706	**0.267**	0.620	0.333	0.677	**0.299**	0.603
	RMSE ↓	0.650	0.972	**0.552**	1.049	0.650	0.948	**0.616**	1.002
LaMP-4: Personalized News Headline Generation	ROUGE-1 ↑	0.163	0.136	**0.182**	0.150	0.176	0.146	**0.188**	0.158
	ROUGE-L ↑	0.147	0.119	**0.167**	0.133	0.160	0.128	**0.172**	0.140
LaMP-5: Personalized Scholarly Title Generation	ROUGE-1 ↑	0.442	0.387	**0.450**	0.390	0.471	0.424	**0.483**	0.425
	ROUGE-L ↑	0.400	0.329	**0.411**	0.329	0.422	0.355	**0.433**	0.351
LaMP-6: Personalized Email Subject Generation	ROUGE-1 ↑	0.362	-	**0.482**	-	0.335	-	**0.401**	-
	ROUGE-L ↑	0.343	-	**0.471**	-	0.319	-	**0.387**	-
LaMP-7: Personalized Tweet Paraphrasing	ROUGE-1 ↑	**0.453**	0.399	0.448	0.390	**0.448**	0.390	0.440	0.382
	ROUGE-L ↑	**0.395**	0.336	0.394	0.322	**0.396**	0.330	0.389	0.318

Table 4: The zero-shot personalized results on the test set of user- and time-based separation settings. The tuned retriever was selected based on the validation performance as reported in Tables 7 and 9 in Appendix D. The results show that personalizing LLMs with the proposed approach improves all datasets in zero-shot setting except LaMP-7.

proves their performance on this benchmark in a zero-shot setting. The outcomes in Tables 2 and 3 show the results for FlanT5-base, a 250M parameter model, fine-tuned on each task. Table 4 presents the zero-shot application of LLMs. These findings indicate that fine-tuning smaller models on downstream tasks leads to enhanced performance in comparison to zero-shot performance of LLMs.

Finally, it is crucial to highlight that the observed outcomes, which indicate superior performance of FlanT5-XXL over GPT-3.5, should not be construed as an inherent deficiency of the latter model. The efficacy of LLMs is extensively contingent upon the caliber and configuration of the input prompts. It is worth noting that prompt engineering, which plays a significant role in performance of LLMs, is not the central objective of this study. Consequently, any disparities in performance must be evaluated in light of this contextual information.

5 Research Problems Enabled by LaMP

LaMP can facilitate research in several areas, including but not limited to:

Prompting Language Models for Personalization. The integration of user profiles into language models can be approached using hard prompts, but their limited context size makes it difficult to include lengthy user profile entries. Exploring different prompts for personalization could be interesting. An alternative solution is generating personalization prompts based on the user profile, instead of relying on retrieved entries. Furthermore, the use of soft prompts (Lester et al., 2021) can be helpful for personalizing language models.

Evaluation of Personalized Text Generation. The commonly used evaluation metrics for text generation, whether syntactical (Lin, 2004; Banerjee and Lavie, 2005; Papineni et al., 2002) or semantical (Zhang* et al., 2020), do not incorporate the user into their evaluation process. Consequently, such metrics may not be entirely suitable for evaluating personalized text generation problems. Exploring new evaluation metrics that account for the user's preferences can benefit this area of research.

Learning to Retrieve from User Profiles. Learning to rank has been widely explored in various retrieval scenarios. Optimizing ranking models that select personalized entries for the sake of personalized text classification and/or generation would be a potentially impactful research direction.

6 Related Work

Personalization has been well studied for information access problems, with the organization of the Netflix Challenge and its associated datasets representing an important driver of academic focus on personalization (Konstan and Terveen, 2021). It also represents an important element of large-scale industry recommender systems (Davidson et al., 2010; Das et al., 2007; Xu et al., 2022) and has also been extensively studied for search applications (Bennett et al., 2012; Dumais, 2016; Croft et al., 2001; Tabrizi et al., 2018; Zeng et al., 2023),

in contexts ranging from query auto-completion (Jaech and Ostendorf, 2018) to collaborative personalized search (Xue et al., 2009). We refer readers to Rafieian and Yoganarasimhan (2023) for an overview of this line of work. Here, we cover personalization in NLP, focusing on research datasets.

Personalization has been examined extensively for dialogue agents (Wu et al., 2021; Zhang et al., 2018; Mazaré et al., 2018). Compared to other NLP tasks. This focus likely stems from the importance of tailoring dialogue to users and conditioning generated utterances on specific personas. Given the lack of real conversational data, some work has constructed dialogue data for users by promoting crowd-workers to author dialogues based on specific personas (Zhang et al., 2018), and through extracting user attributes and utterances from Reddit (Mazaré et al., 2018; Wu et al., 2021) and Weibo (Zhong et al., 2022; Qian et al., 2021). To leverage more realistic conversational data, recent work of Vincent et al. (2023) annotate a dataset of movie dialogues with narrative character personas and posit the potential for using LLMs for dialogue generation conditioned on these personas. Other work has also leveraged publicly available reviews and recipes to explore personalization for review (Li and Tuzhilin, 2019) and recipe generation (Majumder et al., 2019). Wuebker et al. (2018) explore parameter efficient models for personalized translation models. Finally, Ao et al. (2021) presents a personalized headline generation dataset constructed from realistic user interaction data on Microsoft News. This is closely related to the LaMP-4 task, which focuses on personalization for *authors* rather than readers. LaMP presents resources for the tasks which have seen lesser attention than those based on dialogue – expanding the underexplored space of personalizing text classification/generation systems (Flek, 2020; Dudy et al., 2021).

While a body of work has focused on user-facing applications, others have explored personalization for more fundamental problems in language modeling. They have used openly available user data on Reddit (Welch et al., 2022), Facebook, Twitter (Soni et al., 2022), and other blogging websites (King and Cook, 2020). Besides pre-training LMs for personalization, Soni et al. (2022) explores applying a personalized LM for downstream tasks in stance classification and demographic inference. Similarly, other work has explored personalized sentiment prediction on publicly available Yelp and IMDB data (Mireshghallah et al., 2022; Zhong et al., 2021) – this work bears a resemblance to the LaMP-3 task and ties back to rating prediction explored in recommendation tasks. Finally, Plepi et al. (2022) examines the application of personalization methods to modeling annotators in a classification task reliant on modeling social norms – making an important connection between personalization and an emerging body of work on accommodating human label variation in NLP (Rottger et al., 2022; Gordon et al., 2022; Plank, 2022).

7 Conclusion

This paper presented a novel benchmark named LaMP for training and evaluating language models for personalized text classification and generation. LaMP consists of seven datasets: three classification and four generation datasets. We proposed retrieval augmentation solutions for personalizing LLMs. Notably, we studied two augmentation approaches: in-prompt augmentation (IPA) and fusion-in-decoder (FiD). We performed extensive experiments using various LLMs and retrieval techniques for selecting user profile entries for producing personalized prompts. We demonstrated that our LLM personalization approaches can lead to 12.2% average performance improvements across datasets on zero-shot setting, and 23.5% with fine-tuning. Finally, we underscore the paramount importance of personalization in the current era dominated by large language models. We firmly believe that the future of natural language processing systems lies in a user-centric approach, tailoring solutions to individual needs for optimal effectiveness.

Acknowledgments

This work was supported in part by the Center for Intelligent Information Retrieval, in part by NSF grant #2143434, in part by the Office of Naval Research contract number N000142212688, in part by Google, Microsoft, and Lowe's. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsors.

Limitations

LaMP comes with certain limitations arising from task definitions, data leakage into LLM pre-training collections, effective evaluations for personalized generations and broader privacy concerns associated with personalization.

Task definitions Although all tasks are designed to assess language models' proficiency in personalization, certain tasks could be better grounded in realistic scenarios and real-world applications. For instance, framing the Personalized Citation Identification task as a binary classification problem might not accurately represent real-world situations, where individuals generally need to interact with a more extensive array of articles. Additionally, while Personalized Product Rating is intrinsically linked to predicting user satisfaction, the approach may not be entirely realistic, as reviews in real-world contexts are often accompanied by a numerical rating, rendering direct score prediction less relevant. That being said, LaMP creates an environment for evaluating the abilities of LMs in producing personalized outputs.

Leakage to LLM pretraining data The data used for creating the LaMP benchmark mostly consists of publicly available data on the Web, e.g., public tweets, scholarly articles, news articles, and product reviews. We should take this into consideration that some of this data may have been observed by the large language models during their pretraining process. Therefore, they may even perform poorer in unseen cases compared to what we observe from the results on most the LaMP datasets. For this reason, we included the Avocado dataset for Personalized Email Subject Generation as this is not publicly available on the Web and we expect that language models do not use this dataset for pretraining given the restrictions on the data usage agreement.

Evaluating personalized generation The majority of text generation tasks addressed in this research employ short text generation as it offers greater convenience for evaluation purposes. Well-defined metrics, such as ROUGE and BLEU scores, are readily available to assess the quality of short text generation. Conversely, evaluating long text generation poses significant challenges due to its subjective nature, absence of a definitive reference, the necessity for coherence and consistency, contextual comprehension, varied output, semantic and factual accuracy, as well as the limitations of conventional metrics. Evaluators must account for multiple factors, encompassing structural integrity, clarity, effective employment of context, creativity, and subjectivity. Attaining consistent and objective evaluations proves arduous, as it heavily relies on human judgment, which can introduce biases.

Privacy and personalization Finally, we urge the readers to be mindful of privacy implication of LLM personalization. Several studies have shown successful membership attacks against deep learning models (including LLMs) (Mattern et al., 2023; Shokri et al., 2017), thus using personal and private data in fine-tuning may put the privacy of some users at risk. Despite its importance, this paper does not study privacy issues and further analyses are required to ensure how fine-tuned personalized models can preserve the privacy of users. Note that we do not have privacy concerns for the proposed retrieval augmentation approach when a zero-shot language model is used, assuming that the zero-shot language model is hosted in-house or at a trusted third party.

References

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.

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

Paul N. Bennett, Ryen W. White, Wei Chu, Susan T. Dumais, Peter Bailey, Fedor Borisyuk, and Xiaoyuan Cui. 2012. Modeling the impact of short- and long-term behavior on search personalization. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, page 185–194, New York, NY, USA. Association for Computing Machinery.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Bruce W. Croft, Stephen Cronen-Townsend, and Victor Lavrenko. 2001. Relevance feedback and personalization: A language modeling perspective. In *DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries*.

Abhinandan S. Das, Mayur Datar, Ashutosh Garg, and Shyam Rajaram. 2007. Google news personalization: Scalable online collaborative filtering. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, page 271–280, New York, NY, USA. Association for Computing Machinery.

James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, and Dasarathi Sampath. 2010. The youtube video recommendation system. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, page 293–296, New York, NY, USA. Association for Computing Machinery.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*.

Shiran Dudy, Steven Bedrick, and Bonnie Webber. 2021. Refocusing on relevance: Personalization in NLG. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5190–5202, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Susan T. Dumais. 2016. Personalized search: Potential and pitfalls. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, page 689, New York, NY, USA. Association for Computing Machinery.

Peter S. Fader, Bruce G.S. Hardie, and Ka Lok Lee. 2005. Rfm and clv: Using iso-value curves for customer base analysis. *Journal of Marketing Research*, 42(4):415–430.

Michael Färber and Adam Jatowt. 2020. Citation recommendation: Approaches and datasets. *Int. J. Digit. Libr.*

Lucie Flek. 2020. Returning the N to NLP: Towards contextually personalized classification models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7828–7838, Online. Association for Computational Linguistics.

Andrew Fowler, Kurt Partridge, Ciprian Chelba, Xiaojun Bi, Tom Ouyang, and Shumin Zhai. 2015. Effects of language modeling and its personalization on touchscreen typing performance. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, page 649–658, New York, NY, USA. Association for Computing Machinery.

Markus Freitag and Yaser Al-Onaizan. 2017. Beam search strategies for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 56–60, Vancouver. Association for Computational Linguistics.

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

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision.

Mitchell L. Gordon, Michelle S. Lam, Joon Sung Park, Kayur Patel, Jeff Hancock, Tatsunori Hashimoto, and Michael S. Bernstein. 2022. Jury learning: Integrating dissenting voices into machine learning models. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA. Association for Computing Machinery.

Manish Gupta, Rui Li, Zhijun Yin, and Jiawei Han. 2010. Survey on social tagging techniques. *SIGKDD Explor. Newsl.*

F. Maxwell Harper and Joseph A. Konstan. 2015. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 5(4).

M.A. Hearst, S.T. Dumais, E. Osuna, J. Platt, and B. Scholkopf. 1998. Support vector machines. *IEEE Intelligent Systems and their Applications*, 13(4):18–28.

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

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*.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.

Aaron Jaech and Mari Ostendorf. 2018. Personalized language model for query auto-completion. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 700–705, Melbourne, Australia. Association for Computational Linguistics.

Milton King and Paul Cook. 2020. Evaluating approaches to personalizing language models. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2461–2469, Marseille, France. European Language Resources Association.

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*.

Joseph Konstan and Loren Terveen. 2021. Human-centered recommender systems: Origins, advances, challenges, and opportunities. *AI Magazine*, 42(3):31–42.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Pan Li and Alexander Tuzhilin. 2019. Towards controllable and personalized review generation. 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 3237–3245, Hong Kong, China. Association for Computational Linguistics.

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

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

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.

Justus Mattern, Fatemehsadat Mireshghallah, Zhijing Jin, Bernhard Schölkopf, Mrinmaya Sachan, and Taylor Berg-Kirkpatrick. 2023. Membership inference attacks against language models via neighbourhood comparison.

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.

Fatemehsadat Mireshghallah, Vaishnavi Shrivastava, Milad Shokouhi, Taylor Berg-Kirkpatrick, Robert Sim, and Dimitrios Dimitriadis. 2022. UserIdentifier: Implicit user representations for simple and effective personalized sentiment analysis. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3449–3456, Seattle, United States. Association for Computational Linguistics.

Rishabh Misra. 2022. News category dataset. *arXiv preprint arXiv:2209.11429*.

Rishabh Misra and Jigyasa Grover. 2021. *Sculpting Data for ML: The first act of Machine Learning*.

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

Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. 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 188–197, Hong Kong, China. Association for Computational Linguistics.

Oard, Douglas, Webber, William, Kirsch, David A., and Golitsynskiy, Sergey. 2015. Avocado research email collection.

OpenAI. 2023. Gpt-4 technical report.

Sheena Panthaplackel, Adrian Benton, and Mark Dredze. 2022. Updated headline generation: Creating updated summaries for evolving news stories. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6438–6461, Dublin, Ireland. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online. Association for Computational Linguistics.

Barbara Plank. 2022. The "problem" of human label variation: On ground truth in data, modeling and evaluation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10671–10682, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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.

Hongjin Qian, Xiaohe Li, Hanxun Zhong, Yu Guo, Yueyuan Ma, Yutao Zhu, Zhanliang Liu, Zhicheng Dou, and Ji-Rong Wen. 2021. Pchatbot: A large-scale dataset for personalized chatbot. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 2470–2477, New York, NY, USA. Association for Computing Machinery.

Omid Rafieian and Hema Yoganarasimhan. 2023. Ai and personalization. *Artificial Intelligence in Marketing*, pages 77–102.

Werner J. Reinartz and V. Kumar. 2000. On the profitability of long-life customers in a noncontractual setting: An empirical investigation and implications for marketing. *Journal of Marketing*, 64(4):17–35.

Werner J. Reinartz and V. Kumar. 2003. The impact of customer relationship characteristics on profitable lifetime duration. *Journal of Marketing*, 67(1):77–99.

Stephen Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. 1995. Okapi at trec-3. In *Proceedings of the Third Text REtrieval Conference*, TREC-3, pages 109–126. Gaithersburg, MD: NIST.

Paul Rottger, Bertie Vidgen, Dirk Hovy, and Janet Pierrehumbert. 2022. Two contrasting data annotation paradigms for subjective NLP tasks. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 175–190, Seattle, United States. Association for Computational Linguistics.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18.

Nikita Soni, Matthew Matero, Niranjan Balasubramanian, and H. Andrew Schwartz. 2022. Human language modeling. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 622–636, Dublin, Ireland. Association for Computational Linguistics.

Shayan A. Tabrizi, Azadeh Shakery, Hamed Zamani, and Mohammad Ali Tavallaei. 2018. Person: Personalized information retrieval evaluation based on citation networks. *Information Processing & Management*, 54(4):630–656.

Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. 2008. Arnetminer: Extraction and mining of academic social networks. In *KDD'08*, pages 990–998.

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.

Sebastian Vincent, Rowanne Sumner, Alice Dowek, Charlotte Blundell, Emily Preston, Chris Bayliss, Chris Oakley, and Carolina Scarton. 2023. Personalised language modelling of screen characters using rich metadata annotations. *arXiv preprint arXiv:2303.16618*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy,

and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Charles Welch, Chenxi Gu, Jonathan K. Kummerfeld, Veronica Perez-Rosas, and Rada Mihalcea. 2022. Leveraging similar users for personalized language modeling with limited data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1742–1752, Dublin, Ireland. Association for Computational Linguistics.

Yuwei Wu, Xuezhe Ma, and Diyi Yang. 2021. Personalized response generation via generative split memory network. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1956–1970, Online. Association for Computational Linguistics.

Joern Wuebker, Patrick Simianer, and John DeNero. 2018. Compact personalized models for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 881–886, Brussels, Belgium. Association for Computational Linguistics.

Jiajing Xu, Andrew Zhai, and Charles Rosenberg. 2022. Rethinking personalized ranking at pinterest: An end-to-end approach. In *Proceedings of the 16th ACM Conference on Recommender Systems*, RecSys '22, page 502–505, New York, NY, USA. Association for Computing Machinery.

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

Hansi Zeng, Surya Kallumadi, Zaid Alibadi, Rodrigo Nogueira, and Hamed Zamani. 2023. A personalized dense retrieval framework for unified information access. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, New York, NY, USA. Association for Computing Machinery.

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*.

Hanxun Zhong, Zhicheng Dou, Yutao Zhu, Hongjin Qian, and Ji-Rong Wen. 2022. Less is more: Learning to refine dialogue history for personalized dialogue generation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5808–5820, Seattle, United States. Association for Computational Linguistics.

Wanjun Zhong, Duyu Tang, Jiahai Wang, Jian Yin, and Nan Duan. 2021. UserAdapter: Few-shot user learning in sentiment analysis. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1484–1488, Online. Association for Computational Linguistics.

Jianing Zhou and Suma Bhat. 2021. Paraphrase generation: A survey of the state of the art. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5075–5086, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jian Zhu and David Jurgens. 2021. Idiosyncratic but not arbitrary: Learning idiolects in online registers reveals distinctive yet consistent individual styles. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

A Data Creation Details for Tasks in the LaMP benchmark

This section explains the details behind creating inputs, outputs, and profile entries for each task in the LaMP benchmark. Note that all datasets are in English language.

A.1 User-based Separation Setting

Personalized Citation Identification. To generate data samples, we leverage the Citation Network Dataset (V14) (Tang et al., 2008), which comprises information on scientific papers, authors, and citations. The dataset does not provide demographic details of the users in the data. We select all papers from this dataset that meet the following criteria: 1) they are written in English, 2) they contain at least one reference and one author, and 3) they include an abstract. Subsequently, we group papers based on their authors and only consider authors who have written at least 50 papers. For each author, we randomly select one of their papers and one of its cited references. For negative document selection, we randomly choose one of the first author's

co-authors and one of the papers they have cited in one of their papers, which has not been cited by the first author. If no such author exists, we randomly select an author and repeat this process. Finally, we construct the input, output, and profile of the generated samples for this task, employing the template depicted in Figure 3 in Appendix B. After creating the samples for all users, we divide users into the train, validation, and test sets for this task.

Personalized Movie Tagging. To construct our dataset for this task, we leverage the MovieLense dataset (Harper and Konstan, 2015) obtained from the MovieLense website[7]. This dataset includes the tags that individual users have attributed to each film. However, it does not contain the descriptions or summaries of the films. To acquire these, we retrieve the overviews of each film in this dataset from the Movie Database website[8]. The MovieLense dataset comprises over a thousand tags. However, for our dataset, we retain only the top 15 most frequently selected tags as labels. The dataset does not provide demographic details of the users in the data. Furthermore, for the sake of simplicity, we only include users who have assigned a single tag to a film. We only retain users with a minimum of five tagged movies. Then, we partition the users into training, validation, and test sets. For each movie tagged by a user, we use the movie description as input, the movie's tag as the output, and the remaining movies tagged by the same user and their tags as the user profile for that sample, following the template shown in Figure 3 in Appendix B. Finally, we randomly select 50% of the generated samples for each user in training, validation, and test sets, and add them to the samples of the corresponding set.

Personalized Product Rating. In this task, we create our dataset by leveraging the Amazon Reviews Dataset (Ni et al., 2019). We filtered out users (i.e., amazon customers who have written reviews) who have written less than 100 and the 1% users with the most reviews as outliers. Since the Amazon Reviews dataset is quite extensive, we randomly sampled a subset of users from the dataset, which we then split into training, validation, and testing sets. Note that the dataset does not provide demographic details of the users in the data.

[7] https://movielens.org/

[8] https://www.themoviedb.org/?language=en-US

To construct the input-output pairs for our task, for each user, we randomly select one of their reviews as the input to the task and use their other reviews as their profile. Specifically, we use the profile to capture the author's writing style, preferences, and tendencies. In this setup, the user's score for the input review serves as the ground truth output for our task. To gain a better understanding of the input, output, and profile, refer to Figure 3 in Appendix B.

Personalized News Headline Generation. To construct our dataset for this task, we leverage the News Categorization dataset (Misra, 2022; Misra and Grover, 2021) from the HuffPost website. The dataset provides author information for each article and is used to group articles by their respective authors. The dataset does not provide demographic details of the users in the data. We filtered out the authors with less than four articles. In cases where an article has multiple authors, we assign it only to the first author.

We then randomly split the authors into training, validation, and test sets. For each author in each set, we create input-output pairs by selecting each article as the input, the headline of the article as the output, and the remaining articles written by the same author as their profile. This setup aims to capture the author's writing style, preferences, and tendencies, which can be leveraged to generate headlines that align with their interests. An example of this setup is presented in Figure 3 in Appendix B. Finally, to ensure a diverse and representative dataset, we randomly select 50% of the created samples for each author and add them to the user's corresponding set.

Personalized Scholarly Title Generation. Similar to Section LaMP-1, we leverage the Citation Network Dataset (V14) (Tang et al., 2008) that includes information about scientific papers, authors, and citations to construct our dataset. The dataset does not provide demographic details of the users in the data. We only kept the papers that meet the following criteria: 1) written in English, 2) have at least one reference and one author, and 3) have an abstract. Then, we group papers by their authors and only consider authors who have published at least 50 papers. For each author, we randomly choose one of their papers and use its abstract as input, its title as output, and the remaining papers as the author's profile. Figure 3 in Appendix B illustrates the input format for this task. After cre-

ating the samples for all users, we divide users into the train, validation, and test sets for this task.

Personalized Email Subject Generation. In this study, we adopt the Avocado Research Email Collection (Oard, Douglas et al., 2015) as the primary dataset for our task. The dataset does not provide demographic details of the users in the data. To curate the dataset, we first perform a filtering step where we exclude emails with subject lengths of fewer than five words and content lengths of fewer than 30 words. Next, we group the emails based on their sender's email address, retaining only those from users with email frequencies ranging between 10 to 200 emails. We further divide the users into distinct training, validation, and test sets to ensure that our model generalizes well to unseen data. To generate training examples for each user, we create input-output pairs by considering each email as the input and the corresponding email subject as the output. We supplement these pairs with other emails written by the same user as their profile, as shown in Figure 3 in Appendix B. We ensure that our dataset is diverse and representative by randomly selecting 50% of the curated samples for each user and adding them to their respective sets.

Personalized Tweet Paraphrasing. In this task, we utilize the Sentiment140 dataset (Go et al., 2009) as our tweet collection set. The dataset does not provide demographic details of the users in the data. To ensure that the collected tweets are of adequate length, we only retain tweets containing at least 10 words. We then group the tweets based on the user ID and filter out users with fewer than 10 tweets. Subsequently, we randomly select one tweet from each user profile and use it as input to ChatGPT (i.e., `gpt3.5-turbo`) to generate a paraphrased version. The generated paraphrase is then utilized as the input to our NLP task, with the original tweet serving as the corresponding output. The remaining tweets of the user constitute the user's profile, excluding the one selected as input. Figure 3 in Appendix B provides an overview of the input-output-profile template for our proposed task. After creating the samples for all users, we divide users into the train, validation, and test sets for this task.

A.2 Time-based Separation Setting

Personalized Citation Identification. To generate data samples, we leverage the Citation Network Dataset (V14) (Tang et al., 2008), which comprises information on scientific papers, authors, and citations. The dataset does not provide demographic details of the users in the data. We select all papers from this dataset that meet the following criteria: 1) they are written in English, 2) they contain at least one reference and one author, and 3) they include an abstract. Subsequently, we group papers based on their authors and only consider authors who have written at least 50 papers. We divide each author's papers based on the publication year into three groups chronologically: 1) profile papers, 2) train papers, 3) validation papers, and 4) test papers, where the order of groups shows the flow of time. Each train, validation, and test paper set in this task consists of only one paper. Then, for each paper in the train, validation, and test sets for the user, we select each paper and one of its cited references. For negative document selection, we randomly choose one of the first author's co-authors and one of the papers they have cited in one of their papers, which has not been cited by the first author. If no such author exists, we randomly select an author and repeat this process. Finally, we construct the input, output, and profile of the generated samples for this task, employing the template depicted in Figure 3 in Appendix B. It should be noted that after creating all the samples in the train, validation, and test sets for all the users and aggregating them, we randomly select a subset of validation and test sets to create the final sets for the task.

Personalized Movie Tagging. To construct our dataset for this task, we leverage the MovieLense dataset (Harper and Konstan, 2015) obtained from the MovieLense website[9]. This dataset includes the tags that individual users have attributed to each film. The dataset does not provide demographic details of the users in the data. However, it does not contain the descriptions or summaries of the films. To acquire these, we retrieve the overviews of each film in this dataset from the Movie Database website[10]. The MovieLense dataset comprises over a thousand tags. However, for our dataset, we retain only the top 15 most frequently selected tags as labels. Furthermore, for the sake of simplicity, we only include users who have assigned a single tag to a film. We only retain users with a minimum of five tagged movies. We divide each user's tagged movies based on the date they tagged into three

[9] https://movielens.org/

[10] https://www.themoviedb.org/?language=en-US

groups, chronologically: 1) profile movies, 2) train movies, 3) validation movies, and 4) test movies, where the order of groups shows the flow of time. Each train, validation, and test movies set in this task consists of 20%, 10%, and 10% of movies, respectively. Then, for each movie in the train, validation, and test sets for the user, we use the movie's description as the input, the movie's tag as the output, and the profile movies and their tags as the profile for that sample, following the template shown in Figure 3 in Appendix B. It should be noted that after creating all the samples in the train, validation, and test sets for all the users and aggregating them, we randomly select a subset of validation and test sets to create the final sets for the task.

Personalized Product Rating. In this task, we create our dataset by leveraging the Amazon Reviews Dataset (Ni et al., 2019). The dataset does not provide demographic details of the users in the data. We filtered out users (i.e., amazon customers who have written reviews) who have written less than 100 and the 1% users with the most reviews as outliers. Since the Amazon Reviews dataset is quite extensive, we randomly sampled a subset of users from the dataset. We divide each user's reviews based on the review date into three groups chronologically: 1) profile reviews, 2) train reviews, 3) validation reviews, and 4) test reviews, where the order of groups shows the flow of time. Each train, validation, and test reviews set in this task consists of only one review.

To construct the input-output pairs for our task, for each user, we select their reviews in each of train, validation, and test sets as the input to the task and use the profile reviews as their profile. Specifically, we use the profile to capture the author's writing style, preferences, and tendencies. Additionally, the user's score for the input review serves as the ground truth output for our task. To gain a better understanding of the input, output, and profile, refer to Figure 3 in Appendix B. It should be noted that after creating all the samples in the train, validation, and test sets for all the users and aggregating them, we randomly select a subset of validation and test sets to create the final sets for the task.

Personalized News Headline Generation. To construct our dataset for this task, we leverage the News Categorization dataset (Misra, 2022; Misra and Grover, 2021) from the HuffPost website. The dataset provides author information for each article and is used to group articles by their respective authors. The dataset does not provide demographic details of the users in the data. We filtered out the authors with less than ten articles. In cases where an article has multiple authors, we assign it only to the first author. We divide each author's articles based on the publishing date into three groups chronologically: 1) profile articles, 2) train articles, 3) validation articles, and 4) test articles, where the order of groups shows the flow of time. Each train, validation, and test articles set in this task consists of 20%, 10%, and 10% articles, respectively. Then, for each article in the train, validation, and test sets for the user, we create input-output pairs by selecting each article as the input, the headline of the article as the output, and the profile articles written by the same author as their profile. This setup aims to capture the author's writing style, preferences, and tendencies, which can be leveraged to generate headlines that align with their interests. An example of this setup is presented in Figure 3 in Appendix B. It should be noted that after creating all the samples in the train, validation, and test sets for all the users and aggregating them, we randomly select a subset of validation and test sets to create the final sets for the task.

Personalized Scholarly Title Generation. We leverage the Citation Network Dataset (V14) (Tang et al., 2008) that includes information about scientific papers, authors, and citations to construct our dataset. The dataset does not provide demographic details of the users in the data. We only kept the papers that meet the following criteria: 1) written in English, 2) have at least one reference and one author, and 3) have an abstract. Then, we group papers by their authors and only consider authors who have published at least 50 papers. We divide each author's papers based on the publication year into three groups chronologically: 1) profile papers, 2) train papers, 3) validation papers, and 4) test papers, where the order of groups shows the flow of time. Each train, validation, and test paper set in this task consists of only one paper. Then, for each paper in the train, validation, and test sets for the user, we use its abstract as input, its title as output, and the profile papers as the author's profile. Figure 3 in Appendix B illustrates the input format for this task. After creating the samples for all users, we divide users into the train, validation, and test sets for this task. It should be noted that

after creating all the samples in the train, validation, and test sets for all the users and aggregating them, we randomly select a subset of validation and test sets to create the final sets for the task.

Personalized Email Subject Generation. In this study, we adopt the Avocado Research Email Collection (Oard, Douglas et al., 2015) as the primary dataset for our task. The dataset does not provide demographic details of the users in the data. To curate the dataset, we first perform a filtering step where we exclude emails with subject lengths of fewer than five words and content lengths of fewer than 30 words. Next, we group the emails based on their sender's email address, retaining only those from users with email frequencies ranging between 10 to 200 emails. We divide each user's emails based on the publishing date into three groups chronologically: 1) profile emails, 2) train emails, 3) validation emails, and 4) test emails, where the order of groups shows the flow of time. Each train, validation, and test emails set in this task consists of 20%, 10%, and 10% articles, respectively. Then, for each article in the train, validation, and test sets for the user, we create input-output pairs by considering each email as the input and the corresponding email subject as the output. We supplement these pairs with profile emails written by the same user as their profile, as shown in Figure 3 in Appendix B. It should be noted that after creating all the samples in the train, validation, and test sets for all the users and aggregating them, we randomly select a subset of validation and test sets to create the final sets for the task.

Personalized Tweet Paraphrasing. In this task, we utilize the Sentiment140 dataset (Go et al., 2009) as our tweet collection set. The dataset does not provide demographic details of the users in the data. To ensure that the collected tweets are of adequate length, we only retain tweets containing at least 10 words. We then group the tweets based on the user ID and filter out users with fewer than 10 tweets. We divide each user's tweets based on the publication year into three groups chronologically: 1) profile tweets, 2) train tweets, 3) validation tweets, and 4) test tweets, where the order of groups shows the flow of time. Each train, validation, and test tweet set in this task consists of only one paper. Then, for each tweet in the train, validation, and test sets for the user, we use it as input to ChatGPT (i.e., `gpt3.5-turbo`) to generate a paraphrased version. The generated paraphrase is then utilized as the input to our NLP task, with the original tweet serving as the corresponding output. The profile tweets of the user constitute the user's profile. Figure 3 in Appendix B provides an overview of the input-output-profile template for our proposed task. It should be noted that after creating all the samples in the train, validation, and test sets for all the users and aggregating them, we randomly select a subset of validation and test sets to create the final sets for the task.

B Samples of the Tasks Introduced in the LaMP Benchmark

As mentioned earlier, LaMP proposes seven tasks to evaluate language model personalization. In order to create the data points, we use just a carefully designed template for each task. Figure 3 depicts a sample and template for each task in LaMP. Generally, each sample in each task has an input and output accompanied by a profile consisting of several entries about the user, helping the model to produce personalized results for the user. While the profile entries in the same task have a similar structure, the structure varies between tasks. For example, Figure 3 shows that the profile for Personalized Product Rating comprised of documents with text and score sections, while the profile entries in Personalized Scholarly Title Generation have abstract and title attributes.

C Prompts Used for Adding User Profile to the Language Model's Input

In order to use multiple entries from the user profile to personalize the language model's input, we construct task-specific prompts using the templates and instructions in Table 5.

The prompt creation consists of two stages: 1) Per Profile Entry Prompt (PPEP) creation and 2) Aggregated Input Prompt (AIP) creation. In the first stage, following the instructions in Table 5, we create a prompt for each profile entry. In the second stage, following the instructions in Table 5, we combine the PPEP prompts into a single prompt to be fed to the language model. It should be noted that due to the limited context size of language models, we need to trim the PPEP prompts. More accurately, considering k prompts need to be merged and that the maximum capacity for the task input is $\bar{L}$ and the maximum context size of the language model is L, we let each PPEP occupy $\frac{L-\bar{L}}{k}$ tokens in the language model's input. For PPEPs

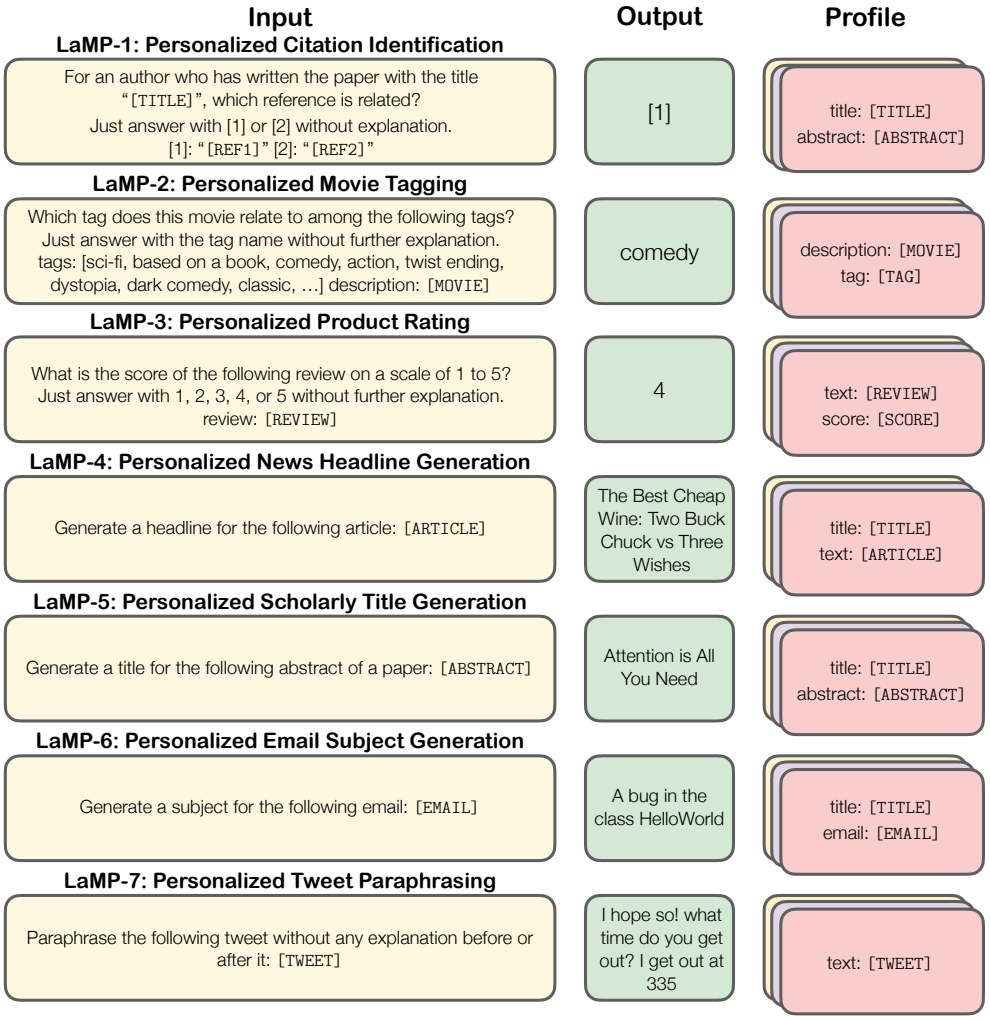

Figure 3: An overview of the templates used for creating data samples for each task in LaMP. Teletype text is replaced with realistic data for each task.

Task	Per Profile Entry Prompt (PPEP)	Aggregated Input Prompt(AIP)
1: Citation Identification	"P_i[title]"	add_to_paper_title(concat([PPEP(P_1), ..., PPEP(P_n)], ", and "), [INPUT])
2: Movie Tagging	the tag for the movie: "P_i[description]" is "P_i[tag]"	concat([PPEP(P_1), ..., PPEP(P_n)], ", and "). [INPUT]
3: Product Rating	P_i[score] is the score for "P_i[text]"	concat([PPEP(P_1), ..., PPEP(P_n)], ", and "). [INPUT]
4: News Headline Generation	"P_i[title]" is the title for "P_i[text]"	concat([PPEP(P_1), ..., PPEP(P_n)], ", and "). [INPUT]
5: Scholarly Title Generation	"P_i[title]" is the title for "P_i[abstract]"	concat([PPEP(P_1), ..., PPEP(P_n)], ", and ""). Following the given patterns [INPUT]
6: Email Subject Generation	"P_i[title]" is the title for "P_i[text]"	concat([PPEP(P_1), ..., PPEP(P_n)], ", and "). [INPUT]
7: Tweet Paraphrasing	"P_i[text]"	concat([PPEP(P_1), ..., PPEP(P_n)], ", and ") are written by a person. Following the given patterns [INPUT]

Table 5: Prompts template used to augment the input of the LM with the user profile. concat is a function that concatenates the strings in its first argument by placing the string in the second argument between them. add_to_paper_title is a function designed to add the string in its first argument to the paper's title in the Personalized Citation Identification task. PPEP is a function that create the prompt for each entry in the retrieved profile entries. [INPUT] is the task's input.

that are longer than the calculated number, we trim the non-template parts that have less importance in the final performance of the model – the parts that do not provide category, score, or title. We select

$\bar{L} = 256$ in this paper.

D Performance of the Models on the Validation Set

This section reports the results of experiments on the validation set. Table 6 reports the results of fine-tuning the language model on the user-based separation setting on the validation set. Table 8 shows the results of fine-tuning the language model on the time-based separation setting on the validation set. Table 7 shows the results of zero-shot evaluation of large language models on the user-based separation setting on the validation set. Table 9 depicts the results of zero-shot evaluation of large language models on the time-based separation setting on the validation set.

E Performance of Some Other Non-Personalized Baselines on the LaMP Benchmark

To explore the performance of additional baselines, we present the performance of the Support Vector Machine (SVM) (Hearst et al., 1998) as a conventional classifier, BERT (Devlin et al., 2019) as a neural transformer-based encoder, and BART (Lewis et al., 2020) as a generative model. SVM and BERT are evaluated on classification tasks, and BART is evaluated on generation tasks within the LaMP benchmark. The results for the user-based and time-based separation configurations are documented in Table 10 and Table 11, respectively.

F Dataset Licenses

This section specifies the licences and terms of use for each task, which is the same as the original dataset's license:

1. Personalized Citation Identification: CC BY-NC-SA 4.0
2. Personalized Movie Tagging: Educational or academic research, NON COMMERCIAL USE
3. Personalized Product Rating: CC BY-NC-SA 4.0
4. Personalized News Headline Generation: CC BY-NC-SA 4.0
5. Personalized Scholarly Title Generation: CC BY-NC-SA 4.0
6. Personalized Email Subject Generation: Avocado Collection End User Agreement LDC2015T03
7. Personalized Tweet Paraphrasing: CC BY-NC-SA 4.0

G AI Assistance Usage

In this paper, ChatGPT[11] has been used as a writing assistant. In more detail, an initial paragraph is given to the ChatGPT and asked to paraphrase the given text. Further edits were also applied to the generated text and then used.

[11] https://chat.openai.com/

		FlanT5-base (fine-tuned)							
		Non-Personalized		Untuned profile, $k=1$			Tuned retriever, k	Tuned profile	
Dataset	Metric	No-Retrieval	Random	Random	BM25	Contriever		IPA	FiD($k=16$)
LaMP-1U: Personalized Citation Identification	Accuracy ↑	0.522	0.526	0.597	0.623	0.695	Contriever, 4	0.731	**0.743**
LaMP-2U: Personalized Movie Tagging	Accuracy ↑	0.449	0.447	0.513	0.498	0.524	Contriever, 4	0.560	**0.640**
	F1 ↑	0.403	0.405	0.462	0.442	0.472		0.512	**0.613**
LaMP-3U: Personalized Product Rating	MAE ↓	0.314	0.324	0.312	0.282	0.275	Contriever, 4	**0.258**	0.259
	RMSE ↓	0.624	0.650	0.633	0.609	0.589		**0.572**	0.577
LaMP-4U: Personalized News Headline Generation	ROUGE-1 ↑	0.158	0.163	0.167	0.176	0.188	Contriever, 4	**0.201**	0.194
	ROUGE-L ↑	0.144	0.151	0.152	0.161	0.172		**0.185**	0.180
LaMP-5U: Personalized Scholarly Title Generation	ROUGE-1 ↑	0.424	0.387	0.389	0.441	0.405	BM25, 2	**0.453**	0.445
	ROUGE-L ↑	0.382	0.350	0.352	0.401	0.367		**0.414**	0.405
LaMP-6U: Personalized Email Subject Generation	ROUGE-1 ↑	0.392	0.466	0.469	0.575	0.567	BM25, 4	**0.583**	0.559
	ROUGE-L ↑	0.374	0.452	0.454	0.563	0.553		**0.570**	0.547
LaMP-7U: Personalized Tweet Paraphrasing	ROUGE-1 ↑	0.511	0.512	0.512	0.520	0.522	Contriever, 4	**0.526**	0.511
	ROUGE-L ↑	0.456	0.456	0.457	0.465	0.467		**0.471**	0.457

Table 6: The personalized text classification and generation results for a fine-tuned language model (i.e., FlanT5-base) on the validation set of user-based separation setting. k denotes the number of documents retrieved for personalizing language model outputs.

		Non-Personalized		Personalized	
Dataset	Metric	FlanT5-XXL	GPT-3.5	FlanT5-XXL	GPT-3.5
LaMP-1U: Personalized Citation Identification	Accuracy ↑	0.522	0.510	0.675	**0.701**
LaMP-2U: Personalized Movie Tagging	Accuracy ↑	0.348	0.372	0.369	**0.466**
	F1 ↑	0.268	0.290	0.294	**0.424**
LaMP-3U: Personalized Product Rating	MAE ↓	0.357	0.699	**0.282**	0.658
	RMSE ↓	0.666	0.977	**0.5841**	1.102
LaMP-4U: Personalized News Headline Generation	ROUGE-1 ↑	0.164	0.133	**0.192**	0.160
	ROUGE-L ↑	0.149	0.118	**0.178**	0.142
LaMP-5U: Personalized Scholarly Title Generation	ROUGE-1 ↑	0.455	0.395	**0.467**	0.398
	ROUGE-L ↑	0.410	0.334	**0.424**	0.336
LaMP-6U: Personalized Email Subject Generation	ROUGE-1 ↑	0.332	-	**0.466**	-
	ROUGE-L ↑	0.320	-	**0.453**	-
LaMP-7U: Personalized Tweet Paraphrasing	ROUGE-1 ↑	**0.459**	0.396	0.448	0.391
	ROUGE-L ↑	**0.404**	0.337	0.396	0.324

Table 7: The zero-shot personalized text classification and generation results on the validation set of user-based separation setting. For personalized models, the tuned retriever based on the validation performance was selected.

		FlanT5-base (fine-tuned)								
		Non-Personalized		Untuned profile, $k = 1$				Tuned retriever, k	Tuned profile	
Dataset	Metric	No-Retrieval	Random	Random	BM25	Contriever	Recency		IPA	FiD($k = 16$)
LaMP-1T: Personalized Citation Identification	Accuracy ↑	0.629	0.630	0.662	0.695	0.709	0.681	Contriever, 4	**0.732**	0.694
LaMP-2T: Personalized Movie Tagging	Accuracy ↑	0.512	0.506	0.531	0.538	0.551	0.546	Contriever, 4	0.570	**0.658**
	F1 ↑	0.460	0.453	0.480	0.485	0.505	0.493		0.522	**0.615**
LaMP-3T: Personalized Product Rating	MAE ↓	0.278	0.276	0.273	0.269	0.272	0.260	Recency, 4	0.259	**0.252**
	RMSE ↓	0.595	0.598	0.590	0.583	0.589	0.576		**0.568**	0.586
LaMP-4T: Personalized News Headline Generation	ROUGE-1 ↑	0.164	0.166	0.176	0.176	0.177	0.179	Recency, 2	**0.185**	0.178
	ROUGE-L ↑	0.149	0.151	0.160	0.161	0.163	0.165		**0.169**	0.164
LaMP-5T: Personalized Scholarly Title Generation	ROUGE-1 ↑	0.462	0.458	0.459	0.473	0.470	0.462	Contriever, 4	**0.472**	0.454
	ROUGE-L ↑	0.414	0.410	0.412	0.425	0.423	0.416		**0.423**	0.411
LaMP-6T: Personalized Email Subject Generation	ROUGE-1 ↑	0.470	0.503	0.504	0.509	0.519	0.510	Contriever, 4	**0.520**	0.513
	ROUGE-L ↑	0.455	0.450	0.489	0.496	0.507	0.497		**0.509**	0.500
LaMP-7T: Personalized Tweet Paraphrasing	ROUGE-1 ↑	0.462	0.462	0.507	0.509	0.514	0.510	Contriever, 4	**0.518**	0.505
	ROUGE-L ↑	0.414	0.448	0.457	0.460	0.464	0.459		**0.467**	0.455

Table 8: The personalized text classification and generation results for a fine-tuned language model (i.e., FlanT5-base) on the validation set of time-based separation setting. k denotes the number of documents retrieved for personalizing language model outputs.

		Non-Personalized		Personalized	
Dataset	Metric	FlanT5-XXL	GPT-3.5	FlanT5-XXL	GPT-3.5
LaMP-1T: Personalized Citation Identification	Accuracy ↑	0.498	0.478	**0.656**	0.640
LaMP-2T: Personalized Movie Tagging	Accuracy ↑	0.326	0.333	0.361	**0.439**
	F1 ↑	0.255	0.273	0.283	**0.403**
LaMP-3T: Personalized Product Rating	MAE ↓	0.335	0.720	**0.294**	0.608
	RMSE ↓	0.639	1.000	**0.586**	1.022
LaMP-4T: Personalized News Headline Generation	ROUGE-1 ↑	0.173	0.146	**0.192**	0.159
	ROUGE-L ↑	0.157	0.128	**0.175**	0.138
LaMP-5T: Personalized Scholarly Title Generation	ROUGE-1 ↑	**0.472**	0.413	**0.472**	0.421
	ROUGE-L ↑	0.419	0.348	**0.422**	0.352
LaMP-6T: Personalized Email Subject Generation	ROUGE-1 ↑	0.316	-	**0.382**	-
	ROUGE-L ↑	0.302	-	**0.369**	-
LaMP-7T: Personalized Tweet Paraphrasing	ROUGE-1 ↑	**0.454**	0.390	0.440	0.392
	ROUGE-L ↑	**0.401**	0.331	0.391	0.325

Table 9: The zero-shot personalized text classification and generation results on the validation set of time-based separation setting. For personalized models, the tuned retriever based on the validation performance was selected.

Dataset	Metric	SVM		BERT		BART	
		Validation	Test	Validation	Test	Validation	Test
LaMP-1T: Personalized Citation Identification	Accuracy ↑	0.512	0.523	0.520	0.483	-	-
LaMP-2T: Personalized Movie Tagging	Accuracy ↑	0.836	0.609	0.609	0.593	-	-
	F1 ↑	0.810	0.580	0.588	0.559	-	-
LaMP-3T: Personalized Product Rating	MAE ↓	0.227	0.515	0.395	0.364	-	-
	RMSE ↓	0.446	0.992	0.758	0.718	-	-
LaMP-4T: Personalized News Headline Generation	ROUGE-1 ↑	-	-	-	-	0.166	0.159
	ROUGE-L ↑	-	-	-	-	0.151	0.145
LaMP-5T: Personalized Scholarly Title Generation	ROUGE-1 ↑	-	-	-	-	0.418	0.409
	ROUGE-L ↑	-	-	-	-	0.380	0.375
LaMP-6T: Personalized Email Subject Generation	ROUGE-1 ↑	-	-	-	-	0.405	0.379
	ROUGE-L ↑	-	-	-	-	0.387	0.361
LaMP-7T: Personalized Tweet Paraphrasing	ROUGE-1 ↑	-	-	-	-	0.515	0.512
	ROUGE-L ↑	-	-	-	-	0.460	0.456

Table 10: The results of non-personalized text classification and generation results on the validation and test set of user-based separation setting of SVM (Hearst et al., 1998), BERT (Devlin et al., 2019), and BART (Lewis et al., 2020).

Dataset	Metric	SVM		BERT		BART	
		Validation	Test	Validation	Test	Validation	Test
LaMP-1T: Personalized Citation Identification	Accuracy ↑	0.540	0.500	0.584	0.582	-	-
LaMP-2T: Personalized Movie Tagging	Accuracy ↑	0.842	0.647	0.639	0.637	-	-
	F1 ↑	0.810	0.610	0.602	0.598	-	-
LaMP-3T: Personalized Product Rating	MAE ↓	0.221	0.556	0.337	0.342	-	-
	RMSE ↓	0.448	1.203	0.706	0.725	-	-
LaMP-4T: Personalized News Headline Generation	ROUGE-1 ↑	-	-	-	-	0.177	0.171
	ROUGE-L ↑	-	-	-	-	0.161	0.157
LaMP-5T: Personalized Scholarly Title Generation	ROUGE-1 ↑	-	-	-	-	0.458	0.456
	ROUGE-L ↑	-	-	-	-	0.414	0.416
LaMP-6T: Personalized Email Subject Generation	ROUGE-1 ↑	-	-	-	-	0.498	0.532
	ROUGE-L ↑	-	-	-	-	0.486	0.518
LaMP-7T: Personalized Tweet Paraphrasing	ROUGE-1 ↑	-	-	-	-	0.507	0.504
	ROUGE-L ↑	-	-	-	-	0.455	0.453

Table 11: The results of non-personalized text classification and generation results on the validation and test set of time-based separation setting of SVM (Hearst et al., 1998), BERT (Devlin et al., 2019), and BART (Lewis et al., 2020).