LaMP: When Large Language Models Meet Personalization

Alireza Salemi
University of Massachusetts Amherst
United States
asalemi@cs.umass.edu

Sheshera Mysore
University of Massachusetts Amherst
United States
smysore@cs.umass.edu

Michael Bendersky
Google Research
United States
bemike@google.com

Hamed Zamani
University of Massachusetts Amherst
United States
zamani@cs.umass.edu

ABSTRACT

This paper highlights the importance of personalization in the current state of natural language understanding and generation 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 classification and four text generation tasks. We also propose a retrieval augmentation approach that retrieves personalized items from user profiles to construct personalized prompts for large language models. Our baseline zero-shot and fine-tuned model results indicate that LMs utilizing profile augmentation outperform their counterparts that do not factor in profile information.

1 INTRODUCTION

As natural language processing (NLP) systems evolve, personalization has emerged as a key factor in meeting the user's expectations for tailored experiences that align with their unique needs and preferences. While 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 [10, 29, 50] – its exploration in NLP has been limited. However, the importance of personalization for text classification and generation tasks has been highlighted in recent work of Flek [9] and Dudy et al. [7]. This work also notes the potential of personalization for centering users and creating accessible and inclusive systems. This optimism has also been reflected in the recent UserNLP'22 workshop [15].

In tandem, the introduction of large language models (LLMs), such as GPT4 [32], has revolutionized NLP. Recent work has shown the benefits (and harms) of personalizing LLMs [19]. Despite this and the importance of personalization in real-world problems, developing and evaluating LLMs for producing personalized responses remain understudied. Therefore, in this paper, we underscore the importance of personalization in shaping the future of NLP and take the first step towards developing and evaluating personalization in the context of LLMs by proposing the LaMP benchmark.[1]

While existing NLP benchmarks, such as (Super)GLUE [44, 45], KILT [34], and GEM [12] have significantly progressed the NLP frontier, 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 News Categorization (categorical classification with 15 categories), and (3) Personalized Product Rating (ordinal classification from 1 to 5-star rating for e-commerce products). Moreover, 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. Therefore, LaMP provides a rich environment for developing personalized NLP models.

For personalizing the LM outputs, a solution is to incorporate the user profile into a LM prompt. However, user profiles are often large and exceed the length limitations of LMs. Even if we relax such limitations as the technology evolves, the cost of processing large input sequences is considerable. Therefore, we propose a personalized retrieval augmentation solution, where for each test input, we retrieve personalized items from the user profile to be included in the LM prompt. We show that using this approach, the performance of LMs 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.

2 PERSONALIZING LLM OUTPUTS

2.1 Problem Formulation

The LaMP benchmark considers every data sample as an individual user, with a collection of user records associated with each sample for all tasks. These user records can facilitate the personalization of language models based on user-specific data. Consequently, each data sample can be partitioned into three components: an input sequence that serves as the model's input, a target output that

[1]LaMP stands for Language Model Personalization.

REML '23, July 27, 2023, Taipei, Taiwan

the model is expected to produce, and a profile that encapsulates any auxiliary information that can be employed to personalize the model according to the user's specific preferences or requirements. Therefore, all tasks within the LaMP benchmark are formalized as follows: for a given textual input x, the goal is to develop a model M that generates personalized output y for user u. One can model this task as $\arg\max_y p(y|x,u)$. For each user u, the model M can take advantage of $P_u = \{(x_{u1}, y_{u1}), (x_{u2}, y_{u2}), \cdots, (x_{um_u}, y_{um_u})\}$ where each (x_{ui}, y_{ui}) denotes a pair of input and personalized output for user u in the same format as x and y. Each task is accompanied by a training set that does not share any user with the test set.

2.2 A Retrieval Augmentation Approach for Personalizing LLMs

In the tasks at hand, each user profile consists of a (potentially large) collection of data points pertaining to the user. Given the inherent context length constraint of language models in addition to their efficiency and cost, it is only practical to incorporate a subset of these data points as input prompts. Moreover, it is important to note that not all entries within a user profile are necessarily relevant to the specific task the user aims to accomplish. Consequently, we propose the development of solutions through retrieval augmentation. This framework selectively extracts pertinent information from the user profile that are relevant to the current unseen test case. An overview of the method is shown in Figure 1.

To achieve personalization for a given sample (x_i, y_i) associated with 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 prompt for user u based on input x_i and the retrieved entries. Consequently, the input $\bar{x_i}$ for the language model is derived using the following formulation:

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

where we use $(\bar{x_i}, y_i)$ to train or evaluate the language models. In this paper, we use Contriever [16], BM25 [39], and a random selection from user's profile to model $\mathcal{R}$. The implementation of the ϕ_p function is depicted in Table 3 in Appendix C. For the ϕ_q function, we concatenate the non-template parts of each input, which can be seen in Figure 3 in Appendix B to obtain the query.

3 THE LAMP BENCHMARK

The LaMP benchmark aim at assessing the efficacy of language models in producing personalized outputs based on user-specific information for seven diverse tasks. These tasks can be categorized as either personalized text classification or personalized text generation tasks:

- **Personalized Text Classification**
  (1) Personalized Citation Identification
  (2) Personalized News Categorization
  (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

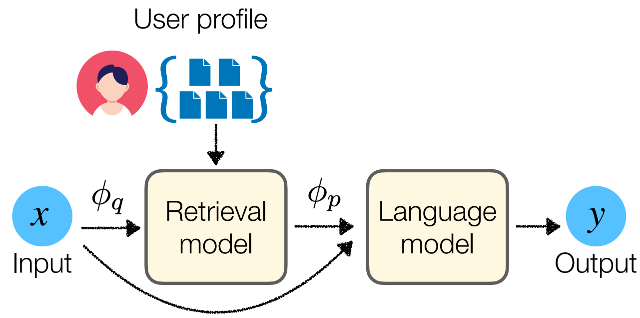

Figure 1: An overview of the retrieval-augmented method for personalizing LLMs. ϕ_q and ϕ_p represent query and prompt construction functions.

In the following sections, we introduce our data collection approach for each of these tasks.

3.1 Task 1: Personalized Citation Identification

This task assesses the capacity of a language model to establish a connection between two papers' titles based on the user profile. In more detail, if the user u writes a paper on a topic x, the language model should determine what papers they cite. Figure 3 in Appendix B illustrates an instance of this task, wherein the model is required to predict the association between a paper's title written by u and two candidate papers, which may serve as references. Therefore, this is a binary classification task.

Data Collection. To generate data samples, we leverage the Citation Network Dataset (V14) [42], which comprises information on scientific papers, authors, and citations. 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.

Profile Specification. 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. We exclude the paper selected to generate the input sequence for this task from the user's profile.

Evaluation. To evaluate performance on this task, we partition the data into train, validation, and test sets. A summary of the dataset statistics is provided in Table 4. As this is a balanced binary classification task, we adopt accuracy as the sole evaluation metric.

3.2 Task 2: Personalized News Categorization

This task aims to assess the capability of a language model to classify news articles written by a user (journalist) u. An illustration of this task is presented in Figure 3 in Appendix B. Given a news article x written by u, the language model must predict its category from the set of available categories based on the user's past articles.

Data Collection. To construct our dataset for this task, we leverage the news categorization dataset [27, 28] obtained from the HuffPost website[2]. However, we filter out some categories and merge similar ones to form a more concise set of categories, as described in Appendix A. Next, we group articles by their author, taking only the first author in cases where there are multiple authors, and retain only authors with a minimum of four articles who have published in at least three different categories. Then, we partition the authors into training, validation, and test sets. For each article of an author, we use the article as input, the article's category as output, and the remaining articles of the same author 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.

Profile Specification. For this task, the user profile consists of the articles written by the user along with their respective categories. However, we exclude the article that is selected as the input for the task from the user profile.

Evaluation. For this task, we create three sets: training, validation, and testing, and their statistics are reported in Table 4. Given that this task involves multi-class classification, we employ accuracy and macro-averaged F1-score as the evaluation metrics to measure the model's performance.

3.3 Task 3: Personalized Product Rating

This task evaluates the language model's ability to predict the rating that a user u has given to a product based on the review written by u for the product. The user's profile, as illustrated in Figure 3 in Appendix B, serves as a basis for predicting an ordinal score with a range from 1 to 5, and exclusively consisting of integer values.

Data Collection. In this task, we create our dataset by leveraging the Amazon Reviews Dataset [30]. 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.

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.

[2]https://www.huffpost.com/

Profile Specification. In this task, the profile refers to a user's other reviews and their respective assigned ratings. It is worth noting that the article selected as the input for the task does not contribute to the user's profile. Instead, the profile is solely derived from the remaining reviews authored by the user.

Evaluation. To evaluate the performance of our model, we randomly split our dataset into training, validation, and test sets. The relevant statistics of our dataset are presented in Table 4. Given that our task is an ordinal multi-class classification problem, we employ RMSE and MAE as the primary evaluation metrics.

3.4 Task 4: Personalized News Headline Generation

This task aims to evaluate the language model's capability to generate a headline for a news article written by a user u. An example of this task is presented in Figure 3 in Appendix B. Specifically, the task involves providing the language model with a news article and requesting it to generate a headline that accurately reflects the user's interests and writing style, as captured in their profile. This task assesses the model's ability to produce headlines that are informative and personalized, based on the user's profile.

Data Collection. To construct our dataset for this task, we leverage the News Categorization dataset [27, 28] from the HuffPost website[3]. The dataset provides author information for each article and is used to group articles by their respective authors. Similar to Section 3.2, 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, we randomly select 50% of the created samples for each author and add them to the user's corresponding set.

Profile Specification. For this task, we define the user profile as the collection of previous articles and their corresponding headlines written by the same author. We exclude the selected article for the sample input from the user's profile.

Evaluation. To evaluate the performance of the model on the headline generation task, we create a train, validation, and test set. The dataset statistics are presented in Table 4. To measure the quality of the generated headlines, we adopt Rouge-1 and Rouge-L as evaluation metrics, which have been commonly used in previous work for headline generation tasks [33, 51]. These metrics capture the overlap between the generated headline and the ground-truth reference headlines based on n-gram overlap and longest common subsequence (LCS) matching.

[3]https://www.huffpost.com/

3.5 Task 5: Personalized Scholarly Title Generation

This task evaluates the language model's capability to generate a title for a research paper, taking into account the other papers authored by the user. An example of this task is illustrated in Figure 3 in Appendix B, where the language model is presented with an abstract of a paper and is required to generate a title that aligns with the user's profile.

Data Collection. Similar to Section 3.1, we leverage the Citation Network Dataset (V14) [42] that includes information about scientific papers, authors, and citations to construct our dataset. 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 shows the input format for this task.

Profile Specification. In this task, the user profile comprises all the papers authored by the user. We extract only the title and abstract of each paper to form the user's profile. Notably, we exclude the paper chosen to create the task input from the user profile.

Evaluation. For this task, we generate a training, validation, and test sets. We report the statistics of this task in Table 4. Similar to the Task 4, we adopt Rouge-1/Rouge-L [22] as evaluation metrics.

3.6 Task 6: Personalized Email Subject Generation

The primary objective of this task is to evaluate the language model's proficiency in generating an appropriate email subject based on the user's writing style. Figure 3 in Appendix B provides an example of the task, which involves providing an email as input to the language model and requesting it to generate a corresponding subject that accurately reflects the content of the email while aligning with the user's writing style.

Data Collection. In this study, we adopt the Avocado Research Email Collection [31] as the primary dataset for our task. 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 article 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.

Profile Specification. For this task, we define a user's profile as the entire set of emails authored by that user. To construct the profile, we extract both the subject and content of each email, excluding the email selected to serve as the input for the given task.

Evaluation. In this study, we create distinct training, validation, and test datasets to facilitate model development and evaluation. We provide an overview of these datasets' key statistics in Table 4. We adopt the Rouge-1 and Rouge-L metrics [22] for evaluation.

3.7 Task 7: Personalized Tweet Paraphrasing

The proposed task aims to evaluate the language model's capacity to generate a paraphrased version of a tweet, considering the writing style of the user. An illustration of the task is presented in Figure 3 in Appendix B. Specifically, the language model is presented with a tweet and instructed to generate a corresponding paraphrase that captures the essence of the original tweet while aligning with the writing style of the user.

Data Collection. In this task, we utilize the Sentiment140 dataset [13] as our tweet collection set. 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`)[4] 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.

Profile Specification. We construct user profiles using all the tweets that a user has posted, excluding the tweet that was selected to form the input to the task. To this end, we only retain the tweet text and disregard other metadata associated with each tweet.

Evaluation. We partition the collected dataset into three distinct subsets: train, validation, and test. We report the key statistics of this dataset in Table 4. In line with prior research [56], we utilize Rouge-1 and Rouge-L [22] as the evaluation metrics.

4 EXPERIMENTS

This section describes results of experiments on LaMP. We refer the reader to Appendix D for details about experimental setup.

4.1 Personalized Results for Fine-Tuned Language Models

This section establishes baseline results for fine-tuned language models. We also investigate the impact of employing various retrieval techniques and the effect of retrieving different quantities of entries from a user's profile. This analysis aims to provide insights into the efficacy of diverse retrieval methods and the benefits of adjusting the number of retrieved entries for personalization tasks.

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

Table 1: The personalized results for a fine-tuned LM (i.e., FlanT5-base). For all metrics the higher the better, except for RMSE/MAE which are used for the LaMP-3 dataset. k shows the number of documents retrieved for personalizing LM.

			FlanT5-base (fine-tuned)				
			Non-Personalized	Untuned profile, $k = 1$			Tuned profile
Dataset		Metric		Random	BM25	Contriever	
LaMP-1: Personalized Citation Identification	dev	Accuracy	0.522	0.597	0.623	0.695	**0.731**
	test	Accuracy	0.518	0.598	0.649	0.688	**0.734**
LaMP-2: Personalized News Categorization	dev	Accuracy	0.730	0.771	0.784	0.811	**0.835**
		F1	0.504	0.529	0.555	0.595	**0.637**
	test	Accuracy	0.674	0.699	0.718	0.729	**0.763**
		F1	0.499	0.522	0.546	0.555	**0.614**
LaMP-3: Personalized Product Rating	dev	MAE	0.314	0.312	0.282	0.275	**0.258**
		RMSE	0.624	0.633	0.609	0.589	**0.572**
	test	MAE	0.275	0.284	0.258	0.248	**0.245**
		RMSE	0.581	0.602	0.573	0.563	**0.560**
LaMP-4: Personalized News Headline Generation	dev	ROUGE-1	0.158	0.167	0.176	0.188	**0.201**
		ROUGE-L	0.144	0.152	0.161	0.172	**0.185**
	test	ROUGE-1	0.153	0.162	0.167	0.173	**0.186**
		ROUGE-L	0.140	0.148	0.153	0.159	**0.171**
LaMP-5: Personalized Scholarly Title Generation	dev	ROUGE-1	0.424	0.389	0.441	0.405	**0.453**
		ROUGE-L	0.382	0.352	0.401	0.367	**0.414**
	test	ROUGE-1	0.418	0.409	0.440	0.431	**0.450**
		ROUGE-L	0.378	0.371	0.399	0.393	**0.409**
LaMP-6: Personalized Email Subject Generation	dev	ROUGE-1	0.392	0.469	0.575	0.567	**0.583**
		ROUGE-L	0.374	0.454	0.563	0.553	**0.570**
	test	ROUGE-1	0.379	0.486	0.586	0.572	**0.587**
		ROUGE-L	0.358	0.470	0.570	0.558	**0.575**
LaMP-7: Personalized Tweet Paraphrasing	dev	ROUGE-1	0.511	0.512	0.520	0.522	**0.526**
		ROUGE-L	0.456	0.457	0.465	0.467	**0.471**
	test	ROUGE-1	0.509	0.514	0.521	0.524	**0.528**
		ROUGE-L	0.455	0.460	0.468	0.471	**0.475**

The Impact of Retrievers on the End-to-End Performance. We employ three retrieval approaches for implementing $\mathcal{R}$: 1) a baseline random selector from the user profile, 2) BM25 [39], and 3) Contriever [16]. BM25 is considered as a robust and strong term-matching retrieval model and Contriever is a pretrained dense retrieval model. FlanT5-base is fine-tuned in all the experiments in this section as the LM that generates personalized output. The results are shown in Table 1. The results suggest that personalization improves the performance for all the tasks within the LaMP benchmark. In majority of cases, even a random selection of documents from the user profile leads to performance improvements.

When retrieving one document per user for personalizing the LM's output, Contriever demonstrates the best performance in all classification datasets (i.e., LaMP-1, LaMP-2, and LaMP-3). For text generation, Contriever performs best for Personalization News Headline Generation (LaMP-4) and Personalized Tweet Paraphrasing (LaMP-7). For Email Generation and Scholarly Title Generation tasks (LaMP-5 and LaMP-6), BM25 demonstrates superior performance. Both BM25 and Contriever outperform a random profile selector in all LaMP datasets. This indicates that merely incorporating information from the user profile into the input is not sufficient, but rather selecting the most relevant information is crucial. This underscores the importance of careful consideration in selecting and incorporating pertinent user profile elements in LM prompts.

The Impact of k (the Number of Retrieved Items from each User Profile) on the End-to-End Performance. 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. Figure 2 depicts the outcome of applying the optimal retriever from Table 1 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, certain tasks experience a decline in performance under these conditions. Given the finite context size of language models, exploring approaches to generate a unified prompt from multiple user entries appears to be a promising avenue for future investigation.

Table 2: The zero-shot personalized results. For all metrics the higher the better, except for RMSE/MAE which are used for the LaMP-3 dataset. For personalized models, the tuned retriever based on the validation performance was selected.

			Non-Personalized		Personalized	
Dataset		Metric	FlanT5-XXL	GPT-3.5	FlanT5-XXL	GPT-3.5
LaMP-1: Personalized Citation Identification	dev	Accuracy	0.522	0.510	0.675	**0.701**
	test	Accuracy	0.520	0.541	**0.699**	0.695
LaMP-2: Personalized News Categorization	dev	Accuracy	0.591	0.610	0.598	**0.693**
		F1	0.463	0.455	**0.471**	0.455
	test	Accuracy	0.581	0.594	0.617	**0.643**
		F1	0.475	0.488	0.512	**0.563**
LaMP-3: Personalized Product Rating	dev	MAE	0.357	0.699	**0.282**	0.658
		RMSE	0.666	0.977	**0.5841**	1.102
	test	MAE	0.344	0.706	**0.267**	0.620
		RMSE	0.650	0.972	**0.552**	1.049
LaMP-4: Personalized News Headline Generation	dev	ROUGE-1	0.164	0.133	**0.192**	0.160
		ROUGE-L	0.149	0.118	**0.178**	0.142
	test	ROUGE-1	0.163	0.136	**0.182**	0.150
		ROUGE-L	0.147	0.119	**0.167**	0.133
LaMP-5: Personalized Scholarly Title Generation	dev	ROUGE-1	0.455	0.395	**0.467**	0.398
		ROUGE-L	0.410	0.334	**0.424**	0.336
	test	ROUGE-1	0.442	0.387	**0.450**	0.390
		ROUGE-L	0.400	0.329	**0.411**	0.329
LaMP-6: Personalized Email Subject Generation	dev	ROUGE-1	0.332	-	**0.466**	-
		ROUGE-L	0.320	-	**0.453**	-
	test	ROUGE-1	0.362	-	**0.482**	-
		ROUGE-L	0.343	-	**0.471**	-
LaMP-7: Personalized Tweet Paraphrasing	dev	ROUGE-1	**0.459**	0.396	0.448	0.391
		ROUGE-L	**0.404**	0.337	0.396	0.324
	test	ROUGE-1	**0.453**	0.399	0.448	0.390
		ROUGE-L	**0.395**	0.336	0.394	0.322

The Impact of Tuning Retriever Parameters on the End-to-End Performance. Based on the performance on the validation set for each dataset, we tuned two parameters for each dataset: (1) the retrieval model (BM25 vs. Contriever), and (2) the number of retrieved items (k). For parameter tuning, we used the following metrics: 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 column of Table 1. As expected, the tuned model outperforms the other models on all datasets.

4.2 Zero-Shot Personalized Results

Due to the widespread of employing large-scale language models with no fine-tuning in contemporary research, we conduct an evaluation of two such models on our benchmark. In particular, we leverage GPT 3.5 (alias `gpt-3.5-turbo` or ChatGPT[5] and FlanT5-XXL [3]. 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

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

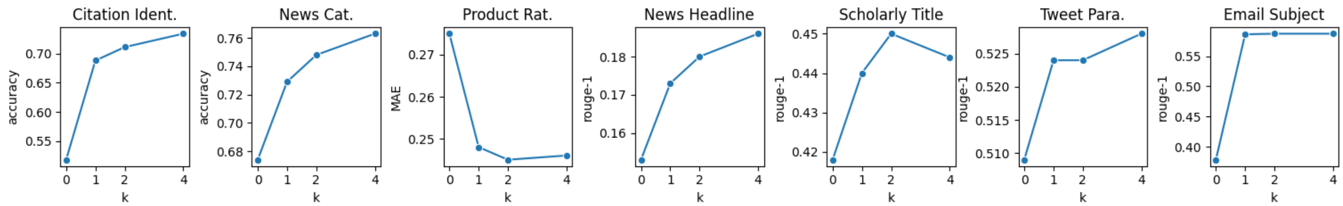

Figure 2: The performance on downstream tasks using the best retriever for each task from Table 1 with different numbers of retrieved entries from the user profile.

to individual tasks and assess their performance based on the generated outputs. In the context of 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 [53]. Subsequently, we assign the most similar label to the generated output as the corresponding output for the given input. To shed light on this, GPT-3.5 generated out-of-the-label predictions 8%, 6%, and 2% of the time for the LaMP-1, LaMP-2, and LaMP-3 tasks, respectively. On the other hand, FlanT5-XXL predictions are consistently among the questioned labels.

Table 2 shows the result of this benchmark on LLMs in a zero-shot scenario. The results show that, except for the Personalized Tweet Paraphrasing task, using the user's profile with LLMs improves their performance on this benchmark in a zero-shot setting. The outcomes in Table 1 show the results for FlabT5-base, a 250M parameter model, fine-tuned on each task. Table 2 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 employed. It is worth noting that prompt engineering is not the central objective of this study. Consequently, any disparities in performance must be evaluated in light of this contextual information.

5 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 [20]. It also represents an important element of large-scale industry recommender systems [5, 6, 49] and has also been studied for search applications [2, 4, 8], in contexts ranging from query auto-completion [17] to collaborative search [50]. We refer readers to [38] for an overview of this line of work.

Personalization has been examined extensively for dialogue agents [25, 47, 52]. Given the lack of real conversational data, this work has constructed dialogue data for users by promoting crowd-workers to author dialogues based on specific personas [52], and through extracting user attributes and utterances from Reddit [25, 47] and Weibo [37, 54]. To leverage more realistic conversational data, Vincent et al. [43] annotates a dataset of movie dialogues with narrative character personas and posit the potential for using LLMs for dialogue generation conditioned on these personas. Besides exploring text generation for dialogues, other work has also leveraged publicly available reviews and recipes to explore personalization for review [21] and recipe generation tasks [24]. Wuebker et al. [48] explore parameter efficient models for personalized translation models with a non-public dataset. Finally, Ao et al. [1] presents a personalized headline generation dataset constructed from realistic user interaction data on Microsoft News. This is closely related to the LaMP-4 task of LaMP, which focuses on personalization for *authors* rather than readers. More broadly, LaMP presents resources for the tasks which have seen lesser attention than those based on dialogue – expanding the underexplored space of personalizing text generation systems [7].

While a body of work has focused on user-facing applications, others have explored personalization for more fundamental problems in language modeling. This body of work has leveraged openly available user data on Reddit [46], Facebook, Twitter [41], and other blogging websites [18]. Besides pre-training LMs for personalization, Soni et al. [41] also explore applying a personalized LM for downstream tasks in stance classification and demographic inference. Similarly, other work has explored personalized sentiment prediction tasks on publicly available Yelp and IMDB data [26, 55] – this work bears a resemblance to the LaMP-3 task in LaMP and ties back to rating prediction tasks explored in recommendation tasks. Finally, Plepi et al. [36] 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 research [14, 35, 40].

6 CONCLUSION

This paper presents a novel benchmark named LaMP for training and evaluating LMs for personalized text classification and generation tasks. LaMP consists of seven datasets: three classification (including two categorical and one ordinal) and four generation datasets. To establish baseline performance, we perform extensive experiments using various LMs and retrieval techniques for selecting user profile entries for producing personalized prompts. Lastly, we report the performance of prominent LLMs on this benchmark.

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, and in part by 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.

REFERENCES

[1] 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)*. Association for Computational Linguistics, Online, 82–92. https://doi.org/10.18653/v1/2021.acl-long.7

[2] 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* (Portland, Oregon, USA) *(SIGIR '12)*. Association for Computing Machinery, New York, NY, USA, 185–194. https://doi.org/10.1145/2348283.2348312

[3] 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. arXiv:2210.11416 [cs.LG]

[4] 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*. http://citeseer.ist.psu.edu/453602.html

[5] 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* (Banff, Alberta, Canada) *(WWW '07)*. Association for Computing Machinery, New York, NY, USA, 271–280. https://doi.org/10.1145/1242572.1242610

[6] 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* (Barcelona, Spain) *(RecSys '10)*. Association for Computing Machinery, New York, NY, USA, 293–296. https://doi.org/10.1145/1864708.1864770

[7] 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*. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 5190–5202. https://doi.org/10.18653/v1/2021.emnlp-main.421

[8] Susan T. Dumais. 2016. Personalized Search: Potential and Pitfalls. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management* (Indianapolis, Indiana, USA) *(CIKM '16)*. Association for Computing Machinery, New York, NY, USA, 689. https://doi.org/10.1145/2983323.2983367

[9] 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*. Association for Computational Linguistics, Online, 7828–7838. https://doi.org/10.18653/v1/2020.acl-main.700

[10] 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* (Seoul, Republic of Korea) *(CHI '15)*. Association for Computing Machinery, New York, NY, USA, 649–658. https://doi.org/10.1145/2702123.2702503

[11] Markus Freitag and Yaser Al-Onaizan. 2017. Beam Search Strategies for Neural Machine Translation. In *Proceedings of the First Workshop on Neural Machine Translation*. Association for Computational Linguistics, Vancouver, 56–60. https://doi.org/10.18653/v1/W17-3207

[12] 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)*. Association for Computational Linguistics, Online, 96–120. https://doi.org/10.18653/v1/2021.gem-1.10

[13] Alec Go, Richa Bhayani, and Lei Huang. 2009. *Twitter Sentiment Classification using Distant Supervision*. http://help.sentiment140.com/home

[14] 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* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 115, 19 pages. https://doi.org/10.1145/3491102.3502004

[15] 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* (Virtual Event, Lyon, France) *(WWW '22)*. Association for Computing Machinery, New York, NY, USA, 1176–1177. https://doi.org/10.1145/3487553.3524879

[16] 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* (2022). https://openreview.net/forum?id=jKN1pXi7b0

[17] 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)*. Association for Computational Linguistics, Melbourne, Australia, 700–705. https://doi.org/10.18653/v1/P18-2111

[18] Milton King and Paul Cook. 2020. Evaluating Approaches to Personalizing Language Models. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*. European Language Resources Association, Marseille, France, 2461–2469. https://aclanthology.org/2020.lrec-1.299

[19] 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* (2023).

[20] Joseph Konstan and Loren Terveen. 2021. Human-Centered Recommender Systems: Origins, Advances, Challenges, and Opportunities. *AI Magazine* 42, 3 (Nov. 2021), 31–42. https://doi.org/10.1609/aimag.v42i3.18142

[21] 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)*. Association for Computational Linguistics, Hong Kong, China, 3237–3245. https://doi.org/10.18653/v1/D19-1319

[22] Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*. Association for Computational Linguistics, Barcelona, Spain, 74–81. https://aclanthology.org/W04-1013

[23] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*. https://openreview.net/forum?id=Bkg6RiCqY7

[24] 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)*. Association for Computational Linguistics, Hong Kong, China, 5976–5982. https://doi.org/10.18653/v1/D19-1613

[25] 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*. Association for Computational Linguistics, Brussels, Belgium, 2775–2779. https://doi.org/10.18653/v1/D18-1298

[26] 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*. Association for Computational Linguistics, Seattle, United States, 3449–3456. https://doi.org/10.18653/v1/2022.naacl-main.252

[27] Rishabh Misra. 2022. News Category Dataset. *arXiv preprint arXiv:2209.11429* (2022).

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

[29] 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. arXiv:1906.00091 [cs.IR]

[30] 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)*. Association for Computational Linguistics, Hong Kong, China, 188–197.

https://doi.org/10.18653/v1/D19-1018
[31] Oard, Douglas, Webber, William, Kirsch, David A., and Golitsynskiy, Sergey. 2015. Avocado Research Email Collection. https://doi.org/10.35111/WQT6-JG60
[32] OpenAI. 2023. GPT-4 Technical Report. arXiv:2303.08774 [cs.CL]
[33] 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)*. Association for Computational Linguistics, Dublin, Ireland, 6438–6461. https://doi.org/10.18653/v1/2022.acl-long.446
[34] 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*. Association for Computational Linguistics, Online, 2523–2544. https://doi.org/10.18653/v1/2021.naacl-main.200
[35] 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*. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 10671–10682. https://aclanthology.org/2022.emnlp-main.731
[36] 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*. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 7391–7402. https://aclanthology.org/2022.emnlp-main.500
[37] 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* (Virtual Event, Canada) *(SIGIR '21)*. Association for Computing Machinery, New York, NY, USA, 2470–2477. https://doi.org/10.1145/3404835.3463239
[38] Omid Rafieian and Hema Yoganarasimhan. 2023. AI and Personalization. *Artificial Intelligence in Marketing* (2023), 77–102.
[39] 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)*. Gaithersburg, MD: NIST, 109–126.
[40] 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*. Association for Computational Linguistics, Seattle, United States, 175–190. https://doi.org/10.18653/v1/2022.naacl-main.13
[41] Nikita Soni, Matthew Matero, Niranjan Balasubramanian, and H. Andrew Schwartz. 2022. Human Language Modeling. In *Findings of the Association for Computational Linguistics: ACL 2022*. Association for Computational Linguistics, Dublin, Ireland, 622–636. https://doi.org/10.18653/v1/2022.findings-acl.52
[42] 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*. 990–998.
[43] 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* (2023).
[44] 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*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2019/file/4496bf24afe7fab6f046bf4923da8de6-Paper.pdf
[45] 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*. Association for Computational Linguistics, Brussels, Belgium, 353–355. https://doi.org/10.18653/v1/W18-5446
[46] 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)*. Association for Computational Linguistics, Dublin, Ireland, 1742–1752. https://doi.org/10.18653/v1/2022.acl-long.122
[47] 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*. Association for Computational Linguistics, Online, 1956–1970. https://doi.org/10.18653/v1/2021.naacl-main.157
[48] 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*. Association for Computational Linguistics, Brussels, Belgium, 881–886. https://doi.org/10.18653/v1/D18-1104
[49] 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* (Seattle, WA, USA) *(RecSys '22)*. Association for Computing Machinery, New York, NY, USA, 502–505. https://doi.org/10.1145/3523227.3547394
[50] Gui-Rong Xue, Jie Han, Yong Yu, and Qiang Yang. 2009. User Language Model for Collaborative Personalized Search. *ACM Trans. Inf. Syst.* 27, 2, Article 11 (mar 2009), 28 pages. https://doi.org/10.1145/1462198.1462203
[51] Kosuke Yamada, Yuta Hitomi, Hideaki Tamori, Ryohei Sasano, Naoaki Okazaki, Kentaro Inui, and Koichi Takeda. 2021. Transformer-based Lexically Constrained Headline Generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 4085–4090. https://doi.org/10.18653/v1/2021.emnlp-main.335
[52] 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)*. Association for Computational Linguistics, Melbourne, Australia, 2204–2213. https://doi.org/10.18653/v1/P18-1205
[53] 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*. https://openreview.net/forum?id=SkeHuCVFDr
[54] 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*. Association for Computational Linguistics, Seattle, United States, 5808–5820. https://doi.org/10.18653/v1/2022.naacl-main.426
[55] 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*. Association for Computational Linguistics, Online, 1484–1488. https://doi.org/10.18653/v1/2021.findings-acl.129
[56] 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*. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 5075–5086. https://doi.org/10.18653/v1/2021.emnlp-main.414

A NEWS CATEGORIZATION DATASET CATEGORY MERGING STRATEGY

The original news categorization dataset contains the following categories: POLITICS, ENTERTAINMENT, HEALTHY LIVING, BUSINESS, SPORTS, COMEDY, PARENTS, WOMEN, CRIME, RELIGION, STYLE, TRAVEL, ARTS, SCIENCE, ARTS & CULTURE, TECH, COLLEGE, EDUCATION, STYLE & BEAUTY, CULTURE & ARTS, FOOD & DRINK. We decided only to keep the following categories based on their distribution: POLITICS, ENTERTAINMENT, HEALTHY LIVING, BUSINESS, SPORTS, COMEDY, PARENTS, WOMEN, CRIME, RELIGION, STYLE, TRAVEL, ARTS, SCIENCE, ARTS & CULTURE, TECH, COLLEGE, EDUCATION, STYLE & BEAUTY, CULTURE & ARTS, FOOD & DRINK.

Next, based on the similarity of different categories, we merge the following categories into the same category:

- "ENTERTAINMENT" and "COMEDY" are merged as "ENTERTAINMENT".
- "STYLE" and "STYLE & BEAUTY" are merged as "STYLE & BEAUTY".
- "ARTS", "CULTURE & ARTS", and "ARTS & CULTURE" are merged as "CULTURE & ARTS".
- "COLLEGE" and "EDUCATION" are merged as "EDUCATION"..

- "SCIENCE" and "TECH" are merged as "SCIENCE & TECHNOLOGY".

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 3.

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 3, we create a prompt for each profile entry. In the second stage, following the instructions in Table 3, 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 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

Experimental Setup To train models, we leverage the AdamW [23] with a learning rate of 5×10^{-5}. We set 5% of the total training steps as warmup steps using a linear warmup scheduler. We also incorporate a weight decay of 10^{-4} to prevent overfitting during training. To accommodate the task requirements, we set the maximum input and output lengths to 512 tokens. We train our generation and classification models for 20 and 10 epochs, respectively. We utilize a FlanT5-base [3] model for all experiments, unless explicitly stated otherwise. We employ beam search [11] with a beam size of 4 in all experiments to improve the model's ability to generate high-quality predictions.

E DATASETS STATISTICS FOR LAMP TASKS

Table 4 provides the statistics, such as type of the task, number of samples, input and output length, and profile size of the datasets available in LaMP benchmark.

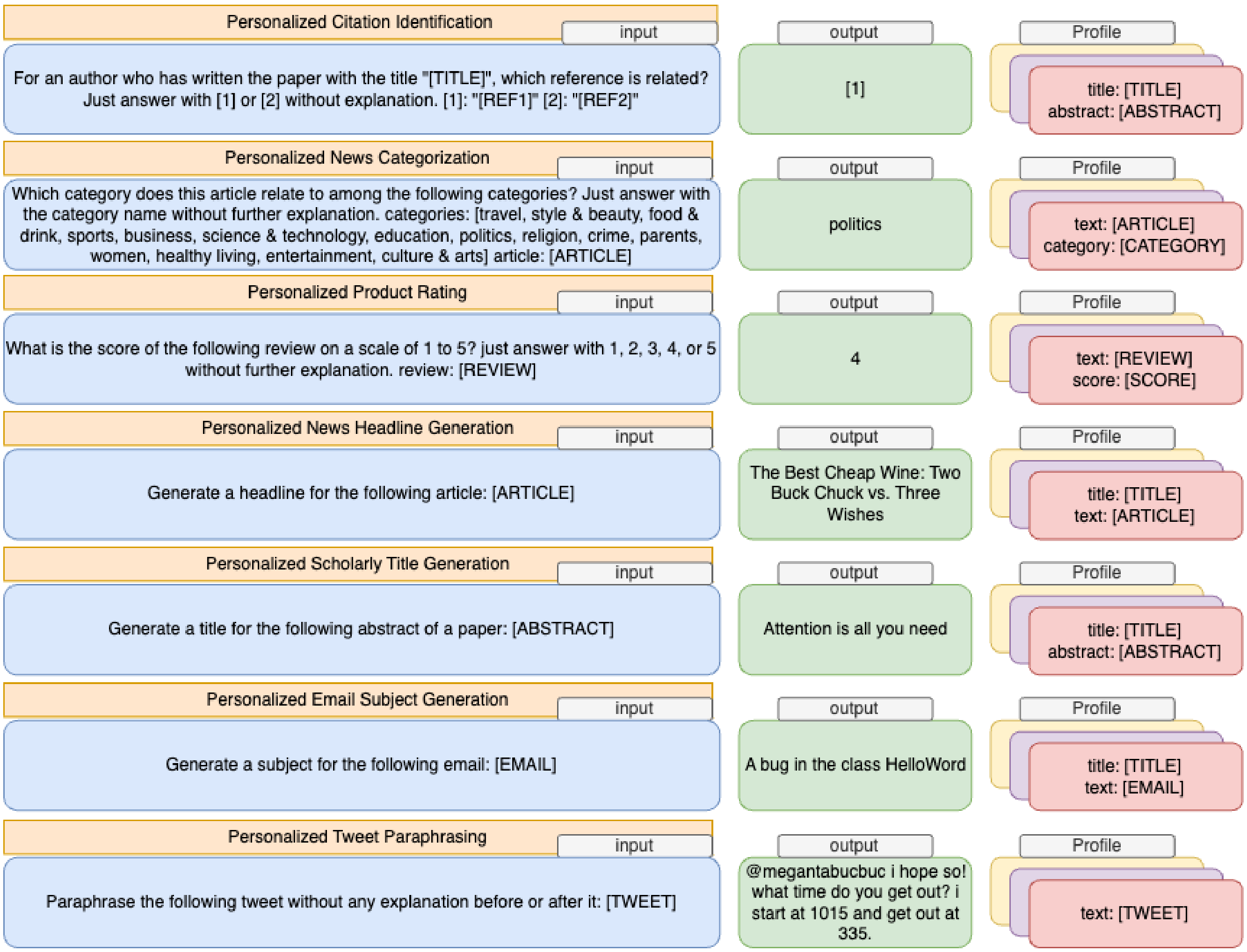

Figure 3: An overview of the templates used for creating data samples for each task in LaMP.

Table 3: 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.

Task	Per Profile Entry Prompt (PPEP)	Aggregated Input Prompt(AIP)
LaMP-1: Citation Ident.	"P_i[title]"	add_to_paper_title(concat([PPEP(P_1), ..., PPEP(P_n)], ", and "), [INPUT])
LaMP-2: News Cat.	the category for the article: "P_i[text]" is "P_i[category]"	concat([PPEP(P_1), ..., PPEP(P_n)], ", and "). [INPUT]
LaMP-3: Product Rat.	P_i[score] is the score for "P_i[text]"	concat([PPEP(P_1), ..., PPEP(P_n)], ", and "). [INPUT]
LaMP-4: News Headline	"P_i[title]" is the title for "P_i[text]"	concat([PPEP(P_1), ..., PPEP(P_n)], ", and "). [INPUT]
LaMP-5: Scholarly Title	"P_i[title]" is the title for "P_i[abstract]"	concat([PPEP(P_1), ..., PPEP(P_n)], ", and "). Following the given patterns [INPUT]
LaMP-6: Email Subject	"P_i[title]" is the title for "P_i[text]"	concat([PPEP(P_1), ..., PPEP(P_n)], ", and "). [INPUT]
LaMP-7: Tweet Para.	"P_i[text]"	concat([PPEP(P_1), ..., PPEP(P_n)], ", and ") are written by a person. Following the given patterns [INPUT]

Table 4: Statistics of the datasets in the LaMP benchmark.

Task	Type	#train	#dev	#test	Input Length	Output Length	#Profiles	#classes
Citation Ident.	binary classification	9682	2500	2500	51.40 ± 5.72	-	90.61 ± 53.87	2
News Cat.	categorical classification	5914	1052	1274	65.93 ± 12.29	-	306.42 ± 286.65	15
Product Rat.	ordinal classification	20000	2500	2500	145.14 ± 157.96	-	188.10 ± 129.42	5
News Headline	text generation	12527	1925	2376	30.53 ± 12.67	9.78 ± 3.10	287.16 ± 360.62	-
Scholarly Title	text generation	9682	2500	2500	152.81 ± 86.60	9.26 ± 3.13	89.61 ± 53.87	-
Email Subject	text generation	4840	1353	1246	436.15 ± 805.54	7.34 ± 2.83	80.72 ± 51.73	-
Tweet Para.	text generation	10437	1500	1496	29.76 ± 6.94	16.93 ± 5.65	17.74 ± 15.10	-