PERSOMA: PERsonalized SOft ProMpt Adapter Architecture for Personalized Language Prompting

Liam Hebert*
liam.hebert@uwaterloo.ca
Google Research
Mountain View, California, USA

Krishna Sayana
ksayana@google.com
Google Research
Mountain View, California, USA

Ambarish Jash
ajash@google.com
Google Research
Mountain View, California, USA

Alexandros Karatzoglou
Google Research
Mountain View, California, USA

Sukhdeep Sodhi
Google Research
Mountain View, California, USA

Sumanth Doddapaneni*
Google Research
Mountain View, California, USA

Yanli Cai
Google Research
Mountain View, California, USA

Dima Kuzmin
Google Research
Mountain View, California, USA

ABSTRACT

Understanding the nuances of a user's extensive interaction history is key to building accurate and personalized natural language systems that can adapt to evolving user preferences. To address this, we introduce PERSOMA, **Per**sonalized **So**ft Pro**m**pt **A**dapter architecture. Unlike previous personalized prompting methods for large language models, PERSOMA offers a novel approach to efficiently capture user history. It achieves this by resampling and compressing interactions as free form text into expressive soft prompt embeddings, building upon recent research utilizing embedding representations as input for LLMs. We rigorously validate our approach by evaluating various adapter architectures, first-stage sampling strategies, parameter-efficient tuning techniques like LoRA, and other personalization methods. Our results demonstrate PERSOMA's superior ability to handle large and complex user histories compared to existing embedding-based and text-prompt-based techniques.

CCS CONCEPTS

• **Computing methodologies** → **Natural language processing**; *Information extraction*; *Natural language generation.*

KEYWORDS

Personalization, Natural Language Processing, User Understanding, Soft Prompting, Large Language Models

ACM Reference Format:
Liam Hebert, Krishna Sayana, Ambarish Jash, Alexandros Karatzoglou, Sukhdeep Sodhi, Sumanth Doddapaneni, Yanli Cai, and Dima Kuzmin. 2024. PERSOMA: PERsonalized SOft ProMpt Adapter Architecture for Personalized Language Prompting. In *Proceedings of Workshop on Generative AI in Recommendation and Personalization at KDD '24 (GenAIRecP @ KDD'24).* ACM, New York, NY, USA, 8 pages. https://doi.org/XXXXXXX.XXXXXXX

*Work done while an intern at Google Research.

GenAIRecP @ KDD'24, August 25–29, 2024, Barcelona, Spain

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Personalized systems have become increasingly essential across various applications in today's connected digital landscape. These systems leverage insights from past interactions to tailor experiences to each user's unique preferences and needs. Personalized systems are found in diverse domains, from content recommendations like music [18] and movies [8], to personalized medicine [12, 26] and customized educational learning pathways [3, 23].

Amid these advancements in personalization, large language models (LLMs) applications have emerged in many domains. When trained on an extensive training corpus of natural language tokens and billions of parameters, large language models have displayed exceptional emergent capabilities to tackle multiple tasks without re-training. In other words, LLMs have emerged as a "one-size-fits-all" solution to many tasks in various domains. To achieve this goal, LLMs are often trained to reproduce text from large generalist and task-agnostic datasets. However, this emphasis on generalist capabilities can limit the effectiveness of LLMs in personalizing outputs towards the needs and characteristics of specific users.

Research to bring personalization to LLMs has primarily explored the development of sophisticated text prompt templates, categorized into simple personalized text prompts, retrieval-augmented prompts, and profile-augmented prompts. Given a list of natural language descriptions of prior actions and a desired task prompt, the simple approach appends the entire set of historical descriptions to the task prompt, leveraging the LLM's inherent in-context learning abilities to personalize the output. However, given the potential for extensive user histories which may exceed the LLM's context window, some studies have proposed retrieval methods to distill this history into the most relevant segments for personalization. Other approaches have also explored using an LLM to synthesize the user's history into brief natural language profiles of user preferences before concatenating them to the desired task prompt.

However, text-prompting approaches to personalization rely on representing the user's history in a lengthy natural language paragraph. Beyond constraints with limited context lengths, several recent works have also found that LLMs often forget information when faced with long prompts. In Liu et al. [17], the authors observed that LLMs tend to leave out information in the middle of the prompt. This problem is further compounded with findings by Shi et al. [27], which find that LLMs are also often distracted by irrelevant context when faced with large prompts. Together, these findings raise the fundamental question of this research: *Is natural language prompting an effective strategy for personalization at scale?*

Recent research [5] has explored using user embeddings as prompts to enhance the personalization capabilities of Large Language Models (LLMs). Building on this approach, we introduce PERSOMA: the **Per**sonalized **So**ft Pro**m**pt **A**dapter Architecture. PERSOMA goes beyond traditional text-based personalization by leveraging parameter-efficient finetuning and history resampling techniques. Instead of relying solely on text, PERSOMA utilizes a soft prompt adapter to condense a user's historical interactions into a compact set of expressive prompt embeddings. These embeddings are further compressed using a perceiver and mapped to the LLM's vocabulary space, ensuring prompt efficiency. This innovative approach allows PERSOMA to steer the output of a frozen LLM toward user preferences without sacrificing the model's capacity for in-context learning or further finetuning. By training the soft prompt adapter on specific tasks, PERSOMA guarantees that the prompt embeddings are contextually relevant, incorporating the necessary information from the user's history to optimize for the desired task.

Our empirical studies, utilizing the PaLM 2 model, demonstrate that PERSOMA effectively harnesses extensive user history for personalization while maintaining computational efficiency. On the MovieLens user preferences dataset [5], PERSOMA outperforms previous embedding-based techniques by 0.18 in F1 score and matches the performance of a fully finetuned text prompting baseline, all while using significantly less computational resources. We investigate PERSOMA's capabilities by comprehensively evaluating various history sampling strategies, input formats, and parameter-efficient training techniques.

2 RELATED WORK

Various text-based prompting methods have been proposed to achieve personalization in language models. These approaches range from concatenating historical interactions into text prompts [7] to using first-stage retrievers for sampling history items [25] and prompt rewriting to summarize critical information [14]. However, text-based prompting suffers from computational constraints due to large inputs and finetuning requirements.

Soft prompting, a parameter-efficient technique introduced by [13], addresses these challenges by adapting frozen language models to specific tasks. This approach has been extended to various domains, including multilingual tasks [33], transfer learning [29], and vision models [2]. Recently, [21] proposed using the language model itself to create more expressive task soft prompt tokens.

Soft prompting has also been explored for personalization. Methods like [34] and [19] create trainable user-specific tokens, while [32] uses MLP networks to create personalized soft prompts from tabular user data. UEM [5] explores using dense user-item embeddings as personalized soft prompts. Recent work has also explored adapting other parameter-efficient techniques for personalization. [6] explore injecting adapter layers in the transformer stack to adapt models towards a specific topic domain.

Building upon UEM and previous work, our research introduces novel methods for resampling, diverse encoder architectures, parameter-efficient training, and comprehensive evaluation of various history sampling techniques. This expands the capabilities and understanding of soft prompting for personalization.

3 METHOD

3.1 PERSOMA Architecture

Our method is grounded in the unified text-to-text approach proposed by T5 [24], where tasks are conceptualized as text generation. Formally, for an input sequence of tokens X, the output sequence Y is modelled probabilistically as $Pr_{\theta_{LM}}(Y|X)$ parameterized by the language model's learned weights θ_{LM}. Prior research in text-to-text tasks can be divided into two predominant prompting methodologies: *text-based prompting*, which integrates textual directives into the input [4, 20, 31], and *soft-prompting*, introducing a series of fixed trained tokens preceding the model's input tokens to capture information about a specific task [13, 15].

Conventionally, soft-prompting employs a static, task-specific soft-prompt to enhance parameter efficiency across various linguistic tasks, optimizing the likelihood $Pr_{\theta_{LM}}(Y|[T;X])$, where T represents a fixed set of trainable task tokens (called task "soft prompts") and X is the task input. Given a set of historical interactions H_u described in natural language for user u, we further train a soft prompt adapter network to jointly *compress and resample* these interactions into a set of personalized user=specific soft-prompt tokens P_u where $|P_u| \leq |H_u|$. The generated soft-prompt tokens can then be utilized to optimize $Pr_{\theta_{LM}}(Y_u|[P_u;T;X])$, where Y_u is a personalized text output conditioned on user u's history.

In practice, PERSOMA (depicted in Figure 1) consists of three primary components: the history encoder, soft prompt adapter and resampler and a large language model decoder. We first encode each natural language user interaction $h_i \in H_u$ using a SentenceT5 text embedding model [22], our history encoder, creating H'_u. We then feed the set of H'_u into our soft prompt adapter network to generate the set of personalized soft prompt tokens P_u. In our results, we experiment with architectures which jointly encode the sequence of H'_u embeddings (Perceiver [11], Transformer [28]) as well as those that encode embeddings individually (MLP projection). Finally, the combined soft prompt $[T;P_u]$ is fed into a frozen PaLM 2 Language Model to generate personalized responses to the given task parameterized by T. To ensure our work is comparable to the prior art, all mentions of PalM 2 refer to the smallest XXS version.

3.2 MovieLens Personalized Genre Prediction Task

To train and evaluate PERSOMA, we utilize the personalized genre prediction task proposed by [5] using the MovieLens dataset [9]. We briefly describe it here for completeness. The MovieLens dataset contains the viewing habits of users and metadata about the movies

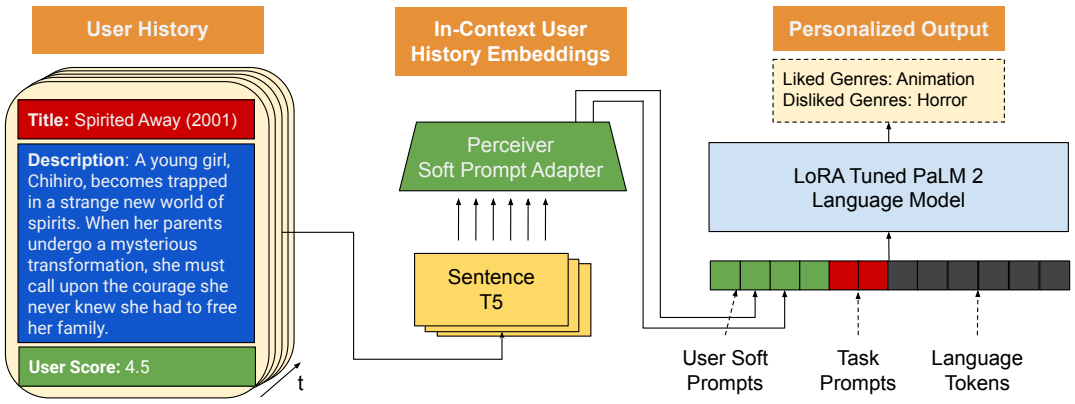

Figure 1: Overview of the PERSOMA architecture. Sentence T5 encodes natural language descriptions of the user's history and is then jointly resampled by a Perceiver adapter to create in-context user soft prompts. These soft prompts are then concatenated with task embeddings to prompt a PaLM 2 language model towards personalized output.

they are watching. We formulate the genre prediction task on this dataset as follows: given a set of natural language titles m^t, descriptions m^d and review scores m^r for each movie $m \in H_u$ user u watched, predict the set of genres the user likes G_u^+ and dislikes G_u^-. The task is suitable for evaluating personalization as the model must understand the user's preferences to predict personalized genre suggestions. We removed all films with less than 20 reviews to ensure that each movie was high quality and could be seen sufficiently across user sequences in the dataset. After pruning, the final dataset includes 14.4 M reviews from 127 K users spanning 8.2 K unique movies and 19 genres.

To apply PERSOMA and other embedding-based personalization baselines on this task, we construct our set of item embeddings H'_u by concatenating the history encoder embeddings for m^t, m^d and m^r. For text-only baselines, we instead represent the input history by concatenating the direct text description of each movie, formatted as

$$U_{text} = \{[\texttt{Title:}m^t, \texttt{Description:}m^d, \texttt{Rating:}m^r] \mid \forall m \in H_u\}$$

Finally, to frame this task as a text-to-text generation task, we formulate the target genre prediction output Y as

$$Y = \texttt{Liked Genres:}\ G_u^+\ \texttt{Not Liked Genres:}\ G_u^-$$

where a comma delimits each genre in G_u^+ and G_u^- (ex: "Liked Genres: Action, Comedy. Not Liked Genres: Romance").

Rather than use conventional text generation metrics such as BLEU and ROUGE, the predicted genre names are extracted from the generated string to evaluate class-weighted Recall, Precision, and F1. This ensures that we explicitly consider the model's performance on the target task, irrespective of the order in which the genres are predicted. Given that users can like or dislike each of the 19 genres, our evaluation metrics evaluate the problem as a 38-class multi-classification problem[1]. We divide the MovieLens dataset into sets of 114K/5K/5K users for training, validation and test respectively, following [5].

[1]The resulting multi-label distribution can be seen in the Appendix Table 5

4 RESULTS

Using the MovieLens Personalized Genre Prediction Task, we evaluate our method against other state-of-the-art personalization baselines. We also conduct an extensive ablation study assessing the impact of various history sampling techniques and item input formats. Unless otherwise noted, we utilize the most recent N movies the user watched to construct the user's history, where N is either 5, 10 or 50 (denoted as "History N"). We also include a simple "Counting" baseline that selects the top three and bottom three watched genres from the user's history as the predicted liked and disliked genres, respectively.

In each experiment, we use 20 learned tokens as our task prompt T. We also use a batch size of 32 and a learning rate of 0.001, training for 300k steps with early stopping to optimize validation F1. When comparing various soft prompt adapter networks, we utilize a 3-layer MLP, 1-layer Transformer network, and 4-layer Perceiver network for each variant. For resampling with Perceiver, the output prompt size is resampled to 20 regardless of history size. We use PaLM2 XXS as the LLM in all our PERSOMA experiments.

4.1 Effect of PERSOMA Soft Prompt Adapter Architecture

A vital component of the PERSOMA architecture is how we process, resample, and represent the user's historical interactions for personalization. We evaluate PERSOMA's performance towards this capability by comparing it against various personalization methods when faced with a user history of the latest 5, 10, and 50 movies watched, respectively. We compare against end-to-end finetuned models UEM Base and UEM Large from [5], which use T5, and PaLM 2 [1] XXS model following the text input formulation described in section 3.2 as embedding-based and text prompt-based baselines respectively. We also include a zero-shot Gemini 1.5 Pro model baseline to compare against PERSOMA methods without end-to-end finetuning.

Our results, shown in Table 1, comparing MLP, Perceiver, and Transformer variants of PERSOMA, demonstrate that both MLP

Table 1: Performance of various adapter architectures and other personalization methods on the MovieLens Genre Prediction Task with Recency Sampling

	History 5			History 10			History 50		
Encoder Architecture	F1	Recall	Precision	F1	Recall	Precision	F1	Recall	Precision
Counting	0.177	0.223	0.246	0.181	0.251	0.249	0.168	0.248	0.271
UEM Large	0.215	0.290	0.281	-	-	-	0.381	0.399	0.400
UEM Base	0.252	0.297	0.275	-	-	-	0.396	0.405	0.407
PERSOMA MLP	0.278	0.274	0.324	0.316	0.355	0.312	0.569	0.563	0.588
PERSOMA Perceiver	0.256	0.305	0.244	0.281	0.327	0.275	0.371	0.411	0.371
PERSOMA Transformer	0.268	0.301	0.257	0.319	0.348	0.312	0.545	0.561	0.545
Fine-tuned PaLM 2 (XXS) Text Prompting	0.285	0.316	0.279	0.335	0.360	0.327	Out of context length		
PERSOMA MLP with Frozen LM	0.280	0.316	0.271	0.323	0.359	0.319	0.541	0.562	0.532
PERSOMA Perceiver with Frozen LM	0.260	0.311	0.265	0.297	0.331	0.292	0.339	0.378	0.360
PERSOMA Transformer with Frozen LM	0.263	0.308	0.256	0.302	0.339	0.295	0.544	0.564	0.546
Gemini 1.5 Pro Zero-Shot Text Prompting	0.187	0.196	0.279	0.221	0.249	0.308	0.261	0.318	0.346

Table 2: Average text token input length for Embedding-based versus Text-based methods. More tokens incur greater computational complexity.

Method/History	5	10	50
Embedding-based	75	80	130
Embedding-based w Perceiver	75	80	100
Text-based (99%)	900	1600	16000
Relative Difference	**12x**	**20x**	**123x-160x**

and Transformer variants outperform UEM Base and UEM Large, particularly with larger history sizes. For example, PERSOMA MLP achieves a 0.19 higher F1 score than UEM Large when processing 50 items. Even with a frozen language model, PERSOMA MLP with Frozen LM still outperforms UEM Large by 0.16 F1 with the same history size.

Since PERSOMA MLP with Frozen LM trains far fewer parameters than UEM Large, our results indicate that finetuning the entire model may be unnecessary with a stronger base LLM (PaLM2 vs T5). Instead, training the adapter to create expressive user embeddings within the LLM language space is sufficient to create significant personalization gains, even at smaller history sizes. This result is further reinforced when resampling with a Perceiver model, unlocking further improvements in token efficiency.

Next, we inspect the performance of PERSOMA against text-prompting baselines. Here, it is essential to note that text prompting requires significantly more computing, scaling according to text token length rather than having a single token per history item. This is best seen in Table 2, where we note that text prompting would require $\approx$ 16 000 input tokens to model a user history of 50 items in MovieLens, whereas PERSOMA would only need 130, further reduced to 100 with Perceiver resampling. Further, it is worth noting that PaLM 2 is finetuned end-to-end toward the target task, tuning many more parameters than our Frozen variants.

Despite utilizing several orders of magnitude less computing, the gap between PERSOMA and PaLM 2 text prompting is only a marginal 0.007 F1 and 0.019 F1 difference when using a history size of 5 and 10, respectively. Comparing PERSOMA Frozen and Gemini, we see that PERSOMA MLP achieves 0.308 higher F1 at a history length of 50. This demonstrates the effectiveness of using in-domain soft prompts for personalization over text prompts.

Finally, we compare the performance of various PERSOMA soft prompt adapter architectures. Surprisingly, we find that PERSOMA MLP, a smaller and simpler architecture, outperforms PERSOMA Perceiver and PERSOMA Transformer. This is especially the case when utilizing a large history size of 50, where PERSOMA MLP outperforms PERSOMA Transformer and PERSOMA Perceiver by 0.024 F1 and 0.198 F1, respectively.

There are a few possible reasons for this loss in optimal performance with sequential adapter models. First, it is essential to note that while MovieLens does provide an ordering of watched films, the order is self-reported by users on the platform via a survey [9]. Given that ordering is not strict, architectures incorporating positional encodings, such as Transformer and Perceiver, may be better suited for tasks that heavily depend on interactions' temporal order. Second, the lack of performance of the Perceiver may be attributed to the reduced number of prompt tokens, which may not have sufficient expressivity as input to the LLM. However, we note that the Perceiver result resampled to a history length of 20 is better than MLP with a history length of 10.

4.2 Parameter Efficient Training with LoRA

To assess the effectiveness of parameter-efficient finetuning techniques for PERSOMA, we conducted experiments using Low-Rank Adaptation (LoRA) [10] and frozen language model weights, comparing them to a fully finetuned PERSOMA model. For LoRA experiments, we use rank four adaptation for both attention and transformer feedforward layers.

As demonstrated in Table 4, when using recency sampling with a history size of 50, both LoRA and frozen PERSOMA variants

Table 3: Performance of various history sampling techniques and sizes. We evaluate PERSOMA MLP against PaLM 2 text prompting and naive counting.

Sampling	Method	History 5			History 10			History 50		
		F1	Recall	Precision	F1	Recall	Precision	F1	Recall	Precision
Recency	PERSOMA	0.278	0.274	0.324	0.316	0.355	0.312	0.569	0.563	0.588
	PaLM 2	0.285	0.279	0.316	0.335	0.360	0.327	Out of context length		
	Counting	0.177	0.223	0.246	0.181	0.251	0.249	0.168	0.248	0.271
Random	PERSOMA	0.265	0.319	0.254	0.320	0.363	0.312	0.574	0.591	0.568
	PaLM 2	0.320	0.361	0.325	0.388	0.419	0.382	Out of context length		
	Counting	0.177	0.246	0.223	0.189	0.265	0.267	0.168	0.248	0.271
Long Tail	PERSOMA	0.259	0.316	0.246	0.283	0.317	0.274	0.329	0.373	0.318
	PaLM 2	0.290	0.322	0.279	0.312	0.342	0.303	Out of context length		
	Counting	0.164	0.207	0.201	0.163	0.209	0.204	0.158	0.206	0.203
Top-K Popularity	PERSOMA	0.294	0.358	0.325	0.351	0.389	0.355	0.593	0.613	0.588
	PaLM 2	0.324	0.374	0.325	0.393	0.433	0.396	Out of context length		
	Counting	0.183	0.254	0.281	0.189	0.267	0.314	0.166	0.249	0.296
Genre Sample	PERSOMA	0.271	0.336	0.259	0.313	0.367	0.321	0.587	0.607	0.580
	PaLM 2	0.313	0.349	0.313	0.370	0.388	0.360	Out of context length		
	Counting	0.189	0.257	0.243	0.187	0.267	0.237	0.166	0.251	0.269

Table 4: Performance of Parameter Efficient Techniques for Training PERSOMA MLP with a History Size of 50 Movies

Method	F1	Precision	Recall
PERSOMA with LoRA LM	0.533	0.524	0.557
PERSOMA with Frozen LM	0.541	0.562	0.532
PERSOMA End-to-End	0.569	0.563	0.588

achieved strong F1 scores of 0.533 and 0.541 respectively, closely matching the performance of the end-to-end finetuned model (0.569 F1). This indicates that parameter-efficient techniques can be effectively applied to PERSOMA, offering a promising avenue for reducing computational costs without sacrificing performance.

4.3 Effect of History Sampling Strategies

Finally, inspired by the LAMP personalization benchmark [25], we extensively evaluate the impact of various methods sampling strategies on the performance of PERSOMA, PaLM 2 and Counting. For a given history size H, we evaluate the following sampling strategies:

- **Recency:** Select the latest H watched movies
- **Random:** Uniformly sample H watched movies
- **Long Tail:** Uniformly sample H watched movies that are below the global 90th quartile of popularity
- **Top-K Popularity:** Select the top H globally popular movies the user watched
- **Genre Sample:** Sample H movies the user watched according to the users genre density

The results of each sampling strategy can be seen in Table 3. We can see that PERSOMA performance improves significantly as the history size increases regardless of sampling strategy. PERSOMA achieved the highest performance with Top-K popularity, achieving 0.593 F1 and 0.294 F1 at history 50 and 5, respectively. Top-K sampling bested Recency sampling, which achieved 0.569 F1 and 0.278 F1 with a history of 50 and 5.

One notable outlier sampling strategy was long tail sampling, achieving a lower F1 across PERSOMA and PaLM 2. This performance gap is likely due to many esoteric films not being in the popular zeitgeist and, therefore, not being well represented in the training data for the language model (unlike Top-K sampling, which features only popular movies). Besides long-tail sampling, all other strategies perform within ± 0.03 F1 of each other, demonstrating PERSOMA's robustness.

5 CONCLUSIONS & FUTURE WORK

This paper introduces PERSOMA, an architecture designed to tackle the challenges of effectively modelling user history for personalization tasks. PERSOMA's core strength lies in its ability to compress and resample historical user interactions into informative in-context soft prompt tokens while employing parameter-efficient techniques for finetuning the language model.

Through extensive experimentation with various encoder architectures, sampling methods, and parameter-efficient techniques, we demonstrate the versatility and robustness of PERSOMA. Notably, PERSOMA matches the performance of text-based prompting even when restricted to the same history size and surpasses these baselines with longer histories, all while requiring significantly less computational power. This makes PERSOMA a valuable tool for practitioners seeking efficient natural language personalization without compromising performance.

We also believe that embedding representations in LLMs is a promising avenue for further work in personalized prompting. We have investigated the effectiveness of our resampling strategies with

Perceiver on the MovieLens dataset, which contains user sequences of O(100) history items. However, real-world production datasets often contain much more extensive interaction histories. Exploring the benefits of resampling with such datasets would be valuable.

Additionally, incorporating sparse first-stage retrievers to pre-filter history items for PERSOMA could prove beneficial, mainly when dealing with histories exceeding 500 interactions. Our sampling experiments, specifically the positive impact of task-targeted sampling (Top-K and Genre Sampling), highlight the potential of this approach. Combining these techniques with Perceiver offers a promising way to efficiently represent long user journeys.

Finally, while we employed PaLM 2 as our large language model decoder, future studies could examine the performance of various LLM decoders of different sizes, such as replacing PaLM 2 with T5 small [24] Phi-2 [16], or MiniLM [30]. Such research could shed light on the applicability of PERSOMA in low-latency and resource-constrained production environments.

ACKNOWLEDGMENTS

The authors would like to thank Santiago Ontanon, who graciously offered expert advice and feedback on initial versions of this work.

REFERENCES

[1] Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernández Ábrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan A. Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vladimir Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, and et al. 2023. PaLM 2 Technical Report. *CoRR* abs/2305.10403 (2023). https://doi.org/10.48550/arXiv.2305.10403 arXiv:2305.10403

[2] Adrian Bulat and Georgios Tzimiropoulos. 2022. Language-aware soft prompting for vision & language foundation models. *arXiv preprint arXiv:2210.01115* (2022).

[3] Lijia Chen, Pingping Chen, and Zhijian Lin. 2020. Artificial intelligence in education: A review. *Ieee Access* 8 (2020), 75264–75278.

[4] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. 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. *CoRR* abs/2210.11416 (2022). https://doi.org/10.48550/arXiv.2210.11416 arXiv:2210.11416

[5] Sumanth Doddapaneni, Krishna Sayana, Ambarish Jash, Sukhdeep Sodhi, and Dima Kuzmin. 2024. User Embedding Model for Personalized Language Prompting. In *Proceedings of the 1st Workshop on Personalization of Generative AI Systems (PERSONALIZE 2024)*, Ameet Deshpande, EunJeong Hwang, Vishvak Murahari, Joon Sung Park, Diyi Yang, Ashish Sabharwal, Karthik Narasimhan, and Ashwin Kalyan (Eds.). Association for Computational Linguistics, St. Julians, Malta, 124–131. https://aclanthology.org/2024.personalize-1.12

[6] Junchen Fu, Fajie Yuan, Yu Song, Zheng Yuan, Mingyue Cheng, Shenghui Cheng, Jiaqi Zhang, Jie Wang, and Yunzhu Pan. 2024. Exploring adapter-based transfer learning for recommender systems: Empirical studies and practical insights. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*. 208–217.

[7] Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2022. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). In *Proceedings of the 16th ACM Conference on Recommender Systems*. 299–315.

[8] Carlos A Gomez-Uribe and Neil Hunt. 2015. The netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems (TMIS)* 6, 4 (2015), 1–19.

[9] F. Maxwell Harper and Joseph A. Konstan. 2016. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.* 5, 4 (2016), 19:1–19:19. https://doi.org/10.1145/2827872

[10] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net. https://openreview.net/forum?id=nZeVKeeFYf9

[11] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. 2021. Perceiver: General perception with iterative attention. In *International conference on machine learning*. PMLR, 4651–4664.

[12] Kevin B Johnson, Wei-Qi Wei, Dilhan Weeraratne, Mark E Frisse, Karl Misulis, Kyu Rhee, Juan Zhao, and Jane L Snowdon. 2021. Precision medicine, AI, and the future of personalized health care. *Clinical and translational science* 14, 1 (2021), 86–93.

[13] Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. arXiv:2104.08691 [cs.CL]

[14] Cheng Li, Mingyang Zhang, Qiaozhu Mei, Weize Kong, and Michael Bendersky. 2023. Automatic Prompt Rewriting for Personalized Text Generation. arXiv:2310.00152 [cs.CL]

[15] Xiang Lisa Li and Percy Liang. 2021. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.). Association for Computational Linguistics, 4582–4597. https://doi.org/10.18653/v1/2021.acl-long.353

[16] Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. 2023. Textbooks are all you need ii: phi-1.5 technical report. *arXiv preprint arXiv:2309.05463* (2023).

[17] Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics* 12 (2024), 157–173.

[18] Martijn Millecamp, Nyi Nyi Htun, Yucheng Jin, and Katrien Verbert. 2018. Controlling spotify recommendations: effects of personal characteristics on music recommender user interfaces. In *Proceedings of the 26th Conference on user modeling, adaptation and personalization*. 101–109.

[19] 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, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, Marine Carpuat, Marie-Catherine de Marneffe, and Iván Vladimir Meza Ruíz (Eds.). Association for Computational Linguistics, 3449–3456. https://doi.org/10.18653/v1/2022.naacl-main.252

[20] Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-Task Generalization via Natural Language Crowdsourcing Instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.). Association for Computational Linguistics, 3470–3487. https://doi.org/10.18653/v1/2022.acl-long.244

[21] Jesse Mu, Xiang Lisa Li, and Noah Goodman. 2023. Learning to Compress Prompts with Gist Tokens. arXiv:2304.08467 [cs.CL]

[22] Jianmo Ni, Gustavo Hernández Ábrego, Noah Constant, Ji Ma, Keith B. Hall, Daniel Cer, and Yinfei Yang. 2022. Sentence-T5: Scalable Sentence Encoders from Pre-trained Text-to-Text Models. In *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.). Association for Computational Linguistics, 1864–1874. https://doi.org/10.18653/v1/2022.findings-acl.146

[23] Muh Putra Pratama, Rigel Sampelolo, and Hans Lura. 2023. Revolutionizing education: harnessing the power of artificial intelligence for personalized learning. *Klasikal: Journal of Education, Language Teaching and Science* 5, 2 (2023), 350–357.

[24] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research* 21, 1 (2020), 5485–5551.

[25] Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. 2023. LaMP: When Large Language Models Meet Personalization. *CoRR* abs/2304.11406 (2023). https://doi.org/10.48550/arXiv.2304.11406 arXiv:2304.11406

[26] Nicholas J Schork. 2019. Artificial intelligence and personalized medicine. *Precision medicine in Cancer therapy* (2019), 265–283.

[27] Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*. PMLR, 31210–31227.

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all

Table 5: Label Distribution of the Movie Lens Genre Prediction Task. In this task, training examples consist of sets of "liked" and "disliked" genres

Genre	Like	Dislike	Total
Drama	41 404	1 747	43 151
Crime	30 243	2 249	32 489
War	30 708	1 204	31 912
Romance	23 830	3 879	27 709
Thriller	20 239	3 711	23 950
Comedy	16 972	5 369	22 341
Mystery	20 106	1 872	21 978
Adventure	14 589	4 894	19 483
Action	12 665	5 649	18 314
Fantasy	12 653	4 154	16 807
Animation	14 389	2 293	16 682
Children	10 870	5 195	16 065
IMAX	9 277	1 892	11 169
Horror	5 972	5 078	11 050
Musical	7 510	2 133	9 643
Western	4 689	757	5 446
Documentary	3 353	178	3 531
Sci-Fi	1 388	498	1 886
Film-Noir	681	16	697
All	281 538	52 765	334 303

you need. *Advances in neural information processing systems* 30 (2017).

[29] Tu Vu, Brian Lester, Noah Constant, Rami Al-Rfou, and Daniel Cer. 2022. SPoT: Better Frozen Model Adaptation through Soft Prompt Transfer. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 5039–5059.

[30] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems* 33 (2020), 5776–5788.

[31] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *NeurIPS*. http://papers.nips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html

[32] Yiqing Wu, Ruobing Xie, Yongchun Zhu, Fuzhen Zhuang, Xu Zhang, Leyu Lin, and Qing He. 2024. Personalized Prompt for Sequential Recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2024).

[33] Mengjie Zhao and Hinrich Schütze. 2021. Discrete and soft prompting for multilingual models. *arXiv preprint arXiv:2109.03630* (2021).

[34] 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, Online Event, August 1-6, 2021 (Findings of ACL, Vol. ACL/IJCNLP 2021)*, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.). Association for Computational Linguistics, 1484–1488. https://doi.org/10.18653/v1/2021.findings-acl.129

A LABEL DISTRIBUTION

B EFFECT OF ITEM INPUT FORMAT

A key aspect of PERSOMA is its utilization of semantic content within historical items. To assess the sensitivity of our architecture and the personalization task to different representations, we conducted an ablation study examining the impact of removing either the movie title or description (Table 5).

Our findings reveal that, for shorter history lengths, PERSOMA performs better using only the movie title than the complete description. Interestingly, omitting the title affects PERSOMA more significantly than omitting the description, while the opposite effect is observed with text prompting using PaLM 2. We hypothesize that this is because movie titles often succinctly capture the essence of a film, unlike a detailed description. Consequently, it is more challenging to learn the compression from a descriptive embedding into a limited number of tokens (5 or 10).

In scenarios with short history inputs, text prompting techniques may be preferable. Alternatively, increasing the size of output tokens independently of the input history length (i.e., resampling to a larger token length) could be considered. We leave this exploration for future work.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009

Table 6: Ablation study of various item input formats on PERSOMA and PaLM 2 text prompting

Input Format	Method	History 5			History 10			History 50		
		F1	Recall	Precision	F1	Recall	Precision	F1	Recall	Precision
Complete	PERSOMA	0.250	0.321	0.246	0.315	0.355	0.312	0.563	0.576	0.560
	PaLM 2	0.285	0.279	0.316	0.335	0.360	0.327	Out of context length		
Title + Score	PERSOMA	0.303	0.330	0.302	0.355	0.376	0.353	0.558	0.598	0.591
	PaLM 2	0.280	0.294	0.272	0.339	0.354	0.328	Out of context length		
Desc. + Score	PERSOMA	0.275	0.321	0.286	0.326	0.365	0.328	0.523	0.536	0.521
	PaLM 2	0.307	0.343	0.299	0.373	0.392	0.377	Out of context length		