

# User Embedding Model for Personalized Language Prompting

Sumanth Doddapaneni, Krishna Sayana\*, Ambarish Jash, Sukhdeep Sodhi, Dima Kuzmin

Google Research

## Abstract

Modeling long user histories plays a pivotal role in enhancing recommendation systems, allowing to capture users' evolving preferences, resulting in more precise and personalized recommendations. In this study, we tackle the challenges of modeling long user histories for preference understanding in natural language. Specifically, we introduce a new User Embedding Module (UEM) that efficiently processes user history in free-form text by compressing and representing them as embeddings, to use them as soft prompts to a LM. Our experiments demonstrate the superior capability of this approach in handling significantly longer histories compared to conventional text-based methods, yielding substantial improvements in predictive performance. Models trained using our approach exhibit substantial enhancements, with up to 0.21 and 0.25 F1 points improvement over the text-based prompting baselines. The main contribution of this research is to demonstrate the ability to bias language models via user signals.

## 1 Introduction

In recent years, Large Language Models (LLMs) have proven their versatility in various language tasks, from translation to reasoning (Bubeck et al., 2023). Scaling up models and data has played a crucial role in unlocking their potential (OpenAI, 2023; Anil et al., 2023; Touvron et al., 2023). LLMs have also been adapted for conversational tasks, instruction following, and reasoning using techniques like Instruction-Tuning (Mishra et al., 2022; Wei et al., 2022a; Sanh et al., 2022), RLHF (Ouyang et al., 2022), and Chain-of-Thought (Wei et al., 2022b). Trained on extensive internet data, these LLMs excel in generalization. They can quickly adapt to new tasks with in-context learning and are capable of not only answering questions but also reasoning about their responses.

\*Correspondence to: ksayana@google.com

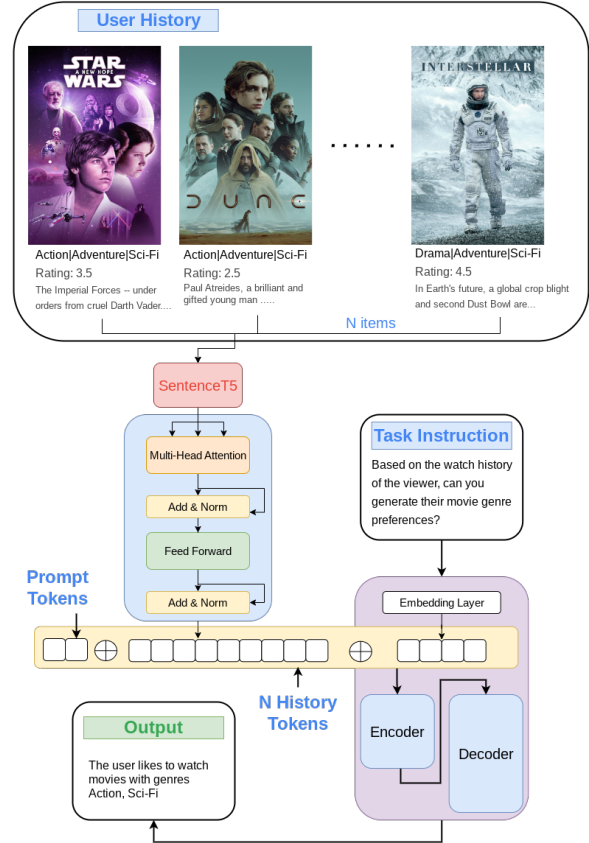


Figure 1: Overview of our User History Modeling Approach. The user history’s textual features are processed through the User Embedding module and combined with the task prompt and subsequently passed through the language model.

The usage of LLMs has evolved beyond traditional NLP tasks to encompass tasks demanding reasoning (Qiao et al., 2023), long-form generation (Ouyang et al., 2022), creativity (Kumar, 2023), and demonstrated remarkable proficiency in these areas. LLMs have been applied to search, retrieval, ranking, chat, personalization, recommendation systems, and others (Yasunaga et al., 2023; Ouyang et al., 2022; Salemi et al., 2023). One practical use case for LMs is understanding *user preferences* to *generate recommendations*, a task that extends

beyond text to encompass audio and visual modalities in real-world scenarios, as exemplified by platforms like YouTube<sup>1</sup>, Spotify<sup>2</sup> among many others.

Recent research has predominantly concentrated on examining smaller segments of user history by selecting representative samples from a users’ history (Salemi et al., 2023). Mu et al. (2023) uses learned gist tokens to compress prompts while Li et al. (2023) uses prompt rewriting based on entries retrieved from a users’ profile. This leads to the critical question *How can we effectively utilize longer user histories?* To achieve this, we employ an embedding-based technique to compress the user’s entire history, creating a sequence of representative user embedding tokens. This embedded representation enhances our ability to comprehend user preferences and subsequently generate predictions that align more closely with their interests. Further, since the User Embedding Module (UEM) module is co-trained with the LM, the representations are learned in-context for the specific tasks. Our research demonstrates the advantages of this approach, particularly in its capacity to incorporate longer user history into LMs, resulting in more robust user preference understanding. Compared to the naive approach of concatenating user history and incurring  $O(n^2)$  compute cost for self-attention, our approach demonstrates a cheap way to incorporate history metadata as an embedding thus dramatically reducing the required compute. As a result longer user histories can be easily incorporated within LMs. Our empirical findings demonstrate the ability of our approach to accommodate significantly larger histories compared to traditional text-based methods, resulting in improved predictive performance.

## 2 Approach

Following text-to-text approach of T5 (Raffel et al., 2020), we frame all tasks as text generation conditioned on the input. Formally, given a sequence of query input tokens denoted as  $X$ , we model the probability of output  $Y$  as  $Pr_\theta(Y|X)$ , where  $\theta$  represents the weights of the model. Prior studies have established two primary prompting strategies: *text-based prompting*, where textual instructions are prepended to the input (Mishra et al., 2022; Chung et al., 2022; Wei et al., 2022b), and *soft-prompting*, which adds a set of train-

able tokens as a prefix to the input tokens of the models (Lester et al., 2021; Li and Liang, 2021). Prior soft-prompting uses a fixed task-specific soft-prompt to achieve parameter-efficient fine-tuning for various language tasks, maximizing the likelihood  $Pr_\theta(Y|[K; X])$ , with  $K$  trainable tokens. We extend this idea to personalization. More specifically, using the User Embedding Module (UEM), we generate a personalised soft-prompt conditioned on the users’ history. This setup aims to maximize the likelihood of the label  $Y$ , given  $Pr_\theta(Y|[Pr_{UEM}(U); X])$ , where  $Pr_{UEM}(U)$  are the soft prompts generated by the UEM based on user history  $U$  and prefixed to the query input  $X$ . In our task definition,  $U$  corresponds to the movie metadata,  $X$  represents the task instruction, and  $Y$  represents genre preferences.

Overall, given a task instruction  $X$ , the LM embeds these tokens to a matrix  $X_e \in \mathbb{R}^{n \times e}$ , where  $n$  represents the token count and  $e$  represents the embedding dimension of the LM. The textual user history  $H = \{h_i\}_{i=1}^p$  is converted into embeddings  $U = \{u_i\}_{i=1}^p$  with SentenceT5 (Ni et al., 2022). Each history item  $u_i$  is a composite of three distinct embeddings: (i) title & genre, (ii) rating, and (iii) description. The collective history of ‘p’ items is expressed as  $U \in \mathbb{R}^{p \times 3s}$ , where  $s$  corresponds to the embedding dimension of SentenceT5. These embeddings undergo processing within a transformer network (UEM). To ensure dimension alignment with  $e$ , a linear projection layer is introduced atop the transformer, mapping the dimension  $3s$  to  $e$ , thereby yielding  $Pr_{UEM}(U) \in \mathbb{R}^{p \times e}$ .

Following Lester et al. (2021), we also incorporate ‘k’ task-level soft prompts, denoted as  $P_e \in \mathbb{R}^{k \times e}$ . Both the user and task prompts are concatenated with the input embedding, resulting in a unified embedding matrix, represented as  $[P_e; Pr_{UEM}(U); X_e] \in \mathbb{R}^{(k+p+n) \times e}$ . This composite embedding flows through the LM, maximizing the probability of  $Y$ , and concurrently updating all parameters within both model components. The model is illustrated in Figure 1.

## 3 Experiments

**Implementation.** As described in §A, we use the MovieLens dataset (Harper and Konstan, 2016) in conjunction with movie descriptions. For the embeddings  $U$  discussed in §2, we format the text in the following manner: (i) title and genre - The movie {movie\_title} is listed with genres

<sup>1</sup>youtube.com

<sup>2</sup>spotify.com

{genres}, (ii) rating - The movie is rated with {rating} stars, and (iii) description - {movie\_description}. In the case of the text-only baselines, we input the concatenated strings instead of the embeddings. The dataset is split into 117k/5k/5k for train, validation and test sets respectively. Unless specified, we use the FlanT5 (Chung et al., 2022) series of models for all experiments, training them for 10k steps with a batch size of 128. Text-history models use a learning rate of  $1e-2$ , while embedding-history models use  $5e-3$ . Our user embedding model consists of 3 transformer layers with 12 attention heads, 768d embeddings, and 2048d MLP layers, adding 65M parameters. We use 20 tokens for task-level soft prompts  $k$ .

**Evaluation.** Although the task is framed in a text-to-text format, the model’s output can be processed by a verbalizer to extract the genres. While conventional metrics like BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), and COMET (Rei et al., 2020) are used for evaluating generative text, they lack granularity in understanding the task performance. However, given the straightforward genre extraction by the verbalizer, we treat the task as a multi-label classification problem and present weighted precision, recall and F1 scores across all labels<sup>3</sup>. Our initial findings indicate that these scores offer a more interpretable assessment, both at the genre level and for the overall task evaluation, compared to token-level metrics.

### 3.1 Main Results

We present the results from our proposed approach in Table 1. The results demonstrate that incorporating a larger history significantly enhances the models’ understanding of the user preferences. Compared to the text-only models, we observe F1 improvements of 0.21 and 0.25 in performance for the base and large models, respectively. To assess performance against text-only models with a comparable history size, we train a model with only 5 history items. The results reveal slightly poorer performance, likely due to the extremely limited context window of the history (5 tokens) compared to the text-only model (over 1000 tokens). Unlike conventional language models, LongT5 (Guo et al., 2022), is trained with Transient Global Attention, allowing it to efficiently process longer text sequences. However, it’s essential to consider that this extended

capability comes at the cost of increased memory and longer training times. While FlanT5 models can be effectively trained on v3-8 TPUs, LongT5 necessitates v3-32 TPUs and requires 4x the training time of a FlanT5 model of comparable size, especially when dealing with input sequences of 16k tokens (equivalent to 50 history items). More importantly, the serving latency is also correspondingly increased, which could make these models impractical for production use.

		BASE	LARGE
Counting Baselines	precision	0.330	
	recall	0.273	
	f1	0.192	
Text Hist. 5	precision	0.276	0.257
	recall	0.287	0.273
	f1	0.273	0.261
Emb. Hist. 5	precision	0.275	0.281
	recall	0.297	0.290
	f1	0.252	0.215
LongT5 50	precision	0.541	0.568
	recall	0.523	0.558
	f1	0.529	0.557
Emb. Hist. 50	precision	0.407	0.400
	recall	0.405	0.399
	f1	0.396	0.381
Emb. Hist. 100	precision	0.416	0.459
	recall	0.413	0.441
	f1	0.404	0.444

Table 1: Model performance using proposed User Embedding Module. *Counting Baselines* refers to counting the three most frequently occurring genres across the entire user history.

### 3.2 Ablations

**Effect of History Length.** To assess the impact of the history size, we conduct a series of ablations by increasing the users’ history passed to UEM. The results are presented in Figure 2 (ref. Table 4), revealing an improvement in model performance with an increase in the number of history items. It’s worth noting that incorporating 50 history items in textual form results in an input of nearly 16k tokens. While methods like LongT5 (Guo et al., 2022), ALiBi (Press et al., 2022), and ROPE (Su et al., 2021) allow for extrapolation to longer sequences, this remains computationally intensive.

**Choice of LM.** In our experiments, we chose FlanT5 as the language model for our task. We also conducted experiments with both the base T5.1 (Raffel et al., 2020) and a LM adapted T5 model

<sup>3</sup>There are 19 genres with a high skew among the classes. We use `sklearn.metrics.classification_report`

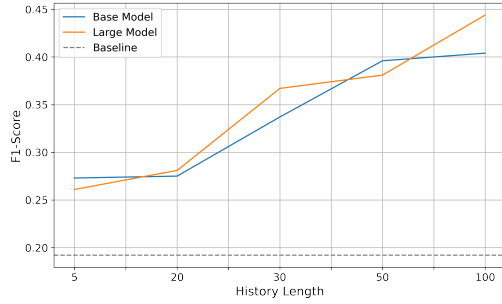


Figure 2: Comparison of model performance with increasing User History.

(Lester et al., 2021) to find the best starting point for our training. The results presented in Table 2 shows that FlanT5 has the best performance, mainly due to the instructional nature of our task prompts.

		BASE	LARGE
T5 <sub>1.1</sub>	precision	0.282	0.322
	recall	0.292	0.324
	f1	0.208	0.267
T5 <sub>LM Adapted</sub>	precision	0.353	0.398
	recall	0.374	0.397
	f1	0.338	0.378
FlanT5	precision	0.407	0.400
	recall	0.405	0.399
	f1	0.396	0.381

Table 2: Comparison of model performance with various choices of Language Models.

**Size of UEM.** For the user embedding module, we experimented with different sizes by changing the number of layers in the transformer block. We found that gradually making UEM bigger improved the performance of both the base and large models (see Table 3). However, we also recognize that the task itself may not require a very complex solution, so further increasing the size of the module may not be justified. We believe that different tasks might need different levels of complexity, and we plan to explore this in future research.

## 4 Related Work

In prior research, UserAdapter (Zhong et al., 2021) introduced a trainable token for each user, facilitating sentiment classification specialization using RoBERTa. Expanding upon this, UserIdentifier (Miresghallah et al., 2022) demonstrated that employing random userIDs effectively captures user-specific information. HuLM (Soni et al., 2022)

		BASE	LARGE
1 Layer	precision	0.391	0.395
	recall	0.381	0.384
	f1	0.346	0.347
2 Layers	precision	0.399	0.380
	recall	0.392	0.367
	f1	0.384	0.365
3 Layers	precision	0.407	0.400
	recall	0.405	0.399
	f1	0.396	0.381

Table 3: Comparison of model performance with various sizes of User Embedding Module. All the models use the same history size of 50.

pretrained a LM conditioned on higher-order data states associated with humans. Further, Salemi et al. (2023) employed retrievers like Contriver and BM25 to select representative input histories to prompt an LM to generate personalized outputs. Mu et al. (2023) utilized gist tokens to condense input prompts into a set of tokens, reducing computational overhead for recurring task instructions. Li et al. (2023) employed prompt rewriting, identifying relevant items for individual users, summarizing the information, and synthesizing key attributes to prompt the model. Our approach distinguishes itself by utilizing entire user histories, compressing them into contextually learned embeddings.

## 5 Conclusion & Future Work

In this study, we addressed several critical challenges in modeling user history for preference understanding. We introduced a User Embedding Module that processed user history as freeform text, generating token embeddings for each history item. This approach greatly simplified user history tracking and enabled the incorporation of longer user histories into the language model, and allowed their representations to be learned in context. Our empirical results demonstrated the capability of this approach to handle significantly larger histories efficiently compared to traditional text-based approaches, resulting in improved predictive performance. For future work, we would like to explore more parameter efficient approaches like LoRA (Hu et al., 2022) for finetuning LMs with UEM, which would improve both training and serving for these models. This approach can be easily extended to multimodal signals using modal specific embeddings and tying them together with UEM, and we plan to explore this direction for future work.



## Limitations

While we argue and demonstrate in this work that using a UEM is an efficient way to encode long user histories with easier extensions to multimodal inputs, we acknowledge that text prompting can be further optimized, by using text-to-text prompt compression models. These trade-offs could be further studied. The simplicity of the UEM architecture leaves a lot of headroom as demonstrated by LongT5 baselines in Table 1. Our presentations for  $U$  are using generic semantic embeddings with the use of SentenceT5 (Ni et al., 2022), these can be further improved with the use of domain specific embeddings. Our experiments are using LMs that are <1B parameters, which are usually considered smaller family of LLMs. It would be a good future direction to consider larger models with parameter efficient tuning techniques. Furthermore, our research has primarily focused on preference understanding, and hasn't been tested on tasks extending to areas such as rating prediction or item recommendation. We expect our conclusions here are likely apply to these tasks. We plan to address these limitations and pursue these avenues in our future research efforts.

## Ethics Statement

The datasets and models utilized in this study are based on publicly available and open-source resources. While we acknowledge the inherent ethical considerations associated with language models, we do not anticipate any additional ethical concerns arising from the datasets and models developed in the course of this research.

## References

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.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrmann, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott M. Lundberg, Harsha Nori, Hamid Palangi, Marco Túlio Ribeiro, and Yi Zhang. 2023. [Sparks of artificial general intelligence: Early experiments with GPT-4](#). *CoRR*, abs/2303.12712.

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.

Mandy Guo, Joshua Ainslie, David Uthus, Santiago Ontanon, Jianmo Ni, Yun-Hsuan Sung, and Yinfei Yang. 2022. [LongT5: Efficient text-to-text transformer for long sequences](#). In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 724–736, Seattle, United States. Association for Computational Linguistics.

F. Maxwell Harper and Joseph A. Konstan. 2016. [The movielens datasets: History and context](#). *ACM Trans. Interact. Intell. Syst.*, 5(4):19:1–19:19.

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.

Pratyush Kumar. 2023. [Large language models humanize technology](#). *CoRR*, abs/2305.05576.

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

Cheng Li, Mingyang Zhang, Qiaozhu Mei, Weize Kong, and Michael Bendersky. 2023. [Automatic prompt rewriting for personalized text generation](#).

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*, pages 4582–4597. Association for Computational Linguistics.

- Chin-Yew Lin. 2004. [ROUGE: A package for automatic evaluation of summaries](#). In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- 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*, pages 3449–3456. Association for Computational Linguistics.
- 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*, pages 3470–3487. Association for Computational Linguistics.
- Jesse Mu, Xiang Lisa Li, and Noah Goodman. 2023. [Learning to compress prompts with gist tokens](#).
- 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*, pages 1864–1874. Association for Computational Linguistics.
- OpenAI. 2023. [GPT-4 technical report](#). *CoRR*, abs/2303.08774.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. [Training language models to follow instructions with human feedback](#). In *NeurIPS*.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. [Bleu: a method for automatic evaluation of machine translation](#). In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL.
- Ofir Press, Noah A. Smith, and Mike Lewis. 2022. [Train short, test long: Attention with linear biases enables input length extrapolation](#). In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.
- Shuofei Qiao, Yixin Ou, Ningyu Zhang, Xiang Chen, Yunzhi Yao, Shumin Deng, Chuanqi Tan, Fei Huang, and Huajun Chen. 2023. [Reasoning with language model prompting: A survey](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5368–5393, Toronto, Canada. Association for Computational Linguistics.
- 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](#). *J. Mach. Learn. Res.*, 21:140:1–140:67.
- Ricardo Rei, Craig Stewart, Ana C. Farinha, and Alon Lavie. 2020. [COMET: A neural framework for MT evaluation](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 2685–2702. Association for Computational Linguistics.
- Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. 2023. [Lamp: When large language models meet personalization](#). *CoRR*, abs/2304.11406.
- Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Févry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. [Multi-task prompted training enables zero-shot task generalization](#). In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.
- Nikita Soni, Matthew Matero, Niranjan Balasubramanian, and H. Andrew Schwartz. 2022. [Human language modeling](#). In *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 622–636. Association for Computational Linguistics.
- Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. 2021. [Roformer: Enhanced transformer with rotary position embedding](#). *CoRR*, abs/2104.09864.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura,

Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. [Llama 2: Open foundation and fine-tuned chat models](#). *CoRR*, abs/2307.09288.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022a. [Finetuned language models are zero-shot learners](#). In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022b. [Chain-of-thought prompting elicits reasoning in large language models](#). In *NeurIPS*.

Michihiro Yasunaga, Armen Aghajanyan, Weijia Shi, Richard James, Jure Leskovec, Percy Liang, Mike Lewis, Luke Zettlemoyer, and Wen-Tau Yih. 2023. [Retrieval-augmented multimodal language modeling](#). In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 39755–39769. PMLR.

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*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 1484–1488. Association for Computational Linguistics.

## A Dataset

In the MovieLens dataset (Harper and Konstan, 2016), the available metadata for assessing a movie’s rating is confined to its title and associated genres. However, such limited information proves inadequate for both human evaluators and language models in generating predictions unless they possess prior knowledge about the movies. For instance, when considering the *Star Wars* series within the MovieLens dataset (Harper and Konstan, 2016), namely *Star Wars: Episode IV - A New Hope* (1977), *Star Wars: Episode VI - Return of the Jedi* (1983), and *Star Wars: Episode I - The Phantom Menace* (1999), all three films share identical genre classifications, namely Action, Adventure,

and Sci-Fi. Nonetheless, a closer inspection reveals noteworthy disparities in their mean ratings, with "Episode IV" and "Episode VI" accumulating ratings of 4.12 and 4.14, respectively, while "Episode I" registers a markedly lower rating of 3.06.

In reality, a viewer’s decision to watch a movie is contingent upon a multifaceted array of metadata beyond the movie genres. Variables such as the cast, crew, production studio, among others, play pivotal roles in this determination. In contemporary times, movie trailers have emerged as potent tools for piquing an individual’s interest in a movie. In the context of textual language models, we equate the concept of a *gist* as a close analogue to a movie trailer. The gist encapsulates the fundamental essence of the movie’s content while withholding explicit plot details, a characteristic akin to that of a trailer. Consequently, we propose the incorporation of such supplementary data<sup>4</sup> into the MovieLens dataset (Harper and Konstan, 2016) to facilitate more nuanced and informed predictive assessments. While it is acknowledged that this augmentation may not encompass the entirety of a viewer’s decision-making process, it represents a stride closer to the intricacies involved in real-world movie-watching choices.

After merging the MovieLens dataset (Harper and Konstan, 2016) with movie descriptions and filtering out users with fewer than 20 recorded movie views, our dataset comprises 14.4M reviews, spanning 8.2k unique movies, and involving a total of 127k users. We then divide this dataset into three subsets: 5k users for both the development and testing sets, and the remaining 117k users for the training set. To create gold labels, we aggregate genres along with their corresponding ratings across each user’s viewing history. Only genres with a minimum of three ratings are considered. Based on this aggregated information, we identify the three most preferred genres (with an average rating >3.5) and the three least preferred genres (with an average rating <3) for each user. The resulting output is structured in a text-to-text format as follows: The user likes to watch movies with genres {liked\_genres} and doesn't like to watch movies with genres {disliked\_genres}<sup>5</sup>.

<sup>4</sup>Metadata sourced from <https://www.kaggle.com/datasets/stefanoleone992/rotten-tomatoes-movies-and-critic-reviews-dataset/>

<sup>5</sup>In cases where the set of liked\_genres or disliked\_genres is empty, the text is adjusted accordingly.

## B History Length Ablation Results

		BASE	LARGE
Emb. Hist. 5	precision	0.276	0.257
	recall	0.287	0.273
	f1	0.273	0.261
Emb. Hist. 20	precision	0.319	0.321
	recall	0.328	0.326
	f1	0.275	0.281
Emb. Hist. 30	precision	0.353	0.390
	recall	0.364	0.390
	f1	0.337	0.367
Emb. Hist. 50	precision	0.407	0.400
	recall	0.405	0.399
	f1	0.396	0.381
Emb. Hist. 100	precision	0.416	0.459
	recall	0.413	0.441
	f1	0.404	0.444

Table 4: Comparison of model performance with increasing User History.