Review-LLM: Harnessing Large Language Models for Personalized Review Generation

Anonymous ACL submission

Abstract

Product review generation is an important task in recommender systems, which could provide explanation and persuasiveness for the recommendation. Recently, Large Language Models (LLMs, e.g., ChatGPT) have shown superior text modeling and generating ability, which could be applied in review generation. However, directly applying the LLMs for generating reviews might be troubled by the "polite" phenomenon of the LLMs and could not generate personalized reviews (e.g., negative reviews). In this paper, we propose Review-LLM that customizes LLMs for personalized review generation. Firstly, we construct the prompt input by aggregating user historical behaviors, which include corresponding item titles and reviews. This enables the LLMs to capture user interest features and review writing style. Secondly, we incorporate ratings as indicators of satisfaction into the prompt, which could further improve the model's understanding of user preferences and the sentiment tendency control of generated reviews. Finally, we feed the prompt text into LLMs, and use Supervised Fine-Tuning (SFT) to make the model generate personalized reviews for the given user and target item. Experimental results on the real-world dataset show that our fine-tuned model could achieve better review generation performance than existing close-source LLMs.

1 Introduction

Online e-commerce platforms (e.g., Amazon.com) usually offer users opportunities to share reviews for items they have purchased (Sun et al., 2020). These reviews typically contain rich user preference information and detailed item attributes (McAuley and Leskovec, 2013), which can inform users about the item and improve recommendation accuracy. However, many users only provide a rating for the item but no review after purchasing the item. Therefore, review generation task has attracted more attentions (Lu et al., 2018).

Most existing methods are based on the encoder-decoder neural network framework (Li et al., 2019, 2020; Kim et al., 2020). Earlier methods utilize discrete attribute information about users and items to generate reviews (Tang et al., 2016; Dong et al., 2017; Ni et al., 2017; Zang and Wan, 2017). For example, Tang et al. (Tang et al., 2016) utilize user/item IDs, and rating as input information, and use the RNN-based decoder for generating reviews. Recent works consider using the text information to help generating reviews, such as item titles, and historical reviews of users/items, etc (Ni and McAuley, 2018; Li and Tuzhilin, 2019). Ni et al. (Ni and McAuley, 2018) propose ExpansionNet, which also integrates phrase information from item titles and review summaries into the encoder for generating reviews. Li et al. (Li and Tuzhilin, 2019) propose a RevGAN model to generate controllable and personalized reviews from item descriptions and sentiment labels.

Recently, owing to the strong reasoning and learning capabilities exhibited by Large Language Models (LLMs) (Achiam et al., 2023; Touvron et al., 2023), many researchers are extending LLMs applications in other domains, such as Recommender Systems (RS) (Xu et al., 2024). Motivated by this, in this paper, we want to preliminary explore how to extend the LLMs (e.g., Llama-3) to the review generation. Compared with other traditional generation tasks (such as poem generation), applying LLMs for the review generation in the e-commerce platforms is more challenging due to the lack of personalized information. First, most existing large language models are usually pre-trained at the corpus-level and might not capture the review style and habits of the users. This might cause the generated review to be inconsistent with user's previous reviews. Second, users are dissatisfied with many items and the corresponding reviews should be negative. However, the generated text by the LLMs is usually "polite" (Touvron et al., 2023),

which might lead to the model generating positive reviews for the user's dissatisfaction.

Hence, in this paper, we design a framework (Review-LLM) for harnessing the LLMs to generate personalized reviews. Specifically, we reconstruct the model input via aggregating the user behavior sequence, including the item titles and corresponding reviews. In this way, the model could learn user interest features and review writing styles from semantically rich text information. Furthermore, the user's rating of the item can be used to indicate the user's satisfaction with the item. We integrate this information into the prompt input accordingly. In this way, the large language model can better perceive whether users like different items, and may prevent the model from generating more "polite" reviews. Finally, we feed the input prompt text into the LLMs (Llama-3), which is subsequently fine-tuned using Supervised Fine-Tuning (SFT) to output the review for target items. For experiments, we design different difficulty levels review generation testing dataset to verify the effectiveness of different models.

2 Method

2.1 Problem Formulation

Given the user u, item v, rating r, and user's historical interaction, review generation aims to automatically generate personalized reviews for the user u towards the target item v. Especially, the user's historical interaction is a sequence of items that the user purchased, which can be denoted as $H^u = \{v_1, v_2, \cdots, v_h\}$, where h is the number of items. And corresponding rating score sequence $R^u = \{r_1, r_2, \cdots, r_h\}$, where h is the number of ratings. The i-th item title and corresponding review are denoted as: $T_i^u = \{w_1, w_2, \cdots, w_N\}$ and $E_i^u = \{w_1, w_2, \cdots, w_M\}$ respectively, where N and M are their lengths. We denote the generated review as $\hat{Y} = \{w_1, w_2, \cdots, w_L\}$ and L is the length; the reference review is denoted as $Y = \{w_1, w_2, \cdots, w_{L'}\}$ and L' is the length.

2.2 Review-LLM

In this section, we introduce Review-LLM for generating reviews. The key is to enhance the LLMs to learn more personalized user interest features and review writing styles based on the histories. Specifically, we propose to construct a prompt text for training the LLM-based model using a supervised fine-tuning approach. As shown in Figure 1, the prompt text composes of the following parts:

1) Generation Instruction: Its role is to instruct the LLMs to consider both the user's preference and historical behaviors to complete the generation task. The generation task is structured as an output of the review for the target item; **2) Input**: This contains the items the user has interacted with, including the item title, review, and rating; **3) The user purchased a new item**: This contains the target item title and the corresponding rating; **4) Response**: This is the generated review for the target item.

Then, we use the following SFT training loss to train the LLM-based review generation model:

$$\mathcal{L}_{sft} = -\sum_{i=1}^{L} logp(w_i|w_{<i}), \quad (1)$$

where w_i is the i-th word in the generated review and L is the length of that. The probability $p(w_i|w_{<i})$ is calculated by the LLM model following the next-token prediction paradigm. During the training process, we utilize the Low-Rank Adaptation (LoRA) (Hu et al., 2021) for Parameter-Efficient Fine-Tuning (PEFT), which can greatly reduce the number of trainable parameters.

During inference, we remove the review of the target item in the **4) Response**. Then we input this modified prompt into the large language model to generate the review for the target item.

3 Experiments

3.1 Experimental Setting

In this paper, we select five open-source 5-core recommendation datasets from Amazon dataset [1], including "Arts, Crafts and Sewing", "Office items", "Musical Instruments", "Toys and Games" and "Video Games". We only remain users with more than 10 historical interactions and less than 30 historical interactions. We timely sort user interactions, then employ the last review as the reference review, and treat others as historical interactions. Then, we randomly select 1000 samples from each dataset as the training set and 200 samples as simple evaluation data from the remaining data. Furthermore, we select 200 negative reviews from each dataset as hard evaluation data to test the model's ability to generate negative reviews.

[1] https://cseweb.ucsd.edu/~jmcauley/datasets/amazon_v2/

Instruction:
Given input, including the items purchased by the user historically, and corresponding reviews and ratings. Please generate a review for the new item purchased by the user, drawing from their historical reviews and ratings. Keep in mind that lower ratings typically result in poorer reviews.

Input:
Item purchased by users historically: (1) Fiskars Scallop Paper Edger Scissors. %Review: I actually bought these to use with my foam sheets. But it doesn't work. However, they work fine on regular paper. %Rating: 1.0. (2) Beadalon JW00T-1 100-Feet 7-Strand Stainless Steel Bead Stringing Wire, 0.010-Inch, Bright. %Review: Very easy to work with. %Rating: 5.0. (3) Makin's Professional Ultimate Clay Machine. %Review: Machine works fine. However, I saw it for a cheaper prize at my local craft store after I bought it from Amazon. %Rating: 5.0.

The user purchased a new item:
DIY Jewelry Making: About 24 pcs of Iron Key Chain Keychain Findings, Platinum Color. %Rating: 5.0.

Response:
A polymer clay crafter must have.

Figure 1: An example of input prompt for Review-LLM.

We conduct experiments using a cluster composed of 4*A800 80GB GPUs. We select Llama-3-8b [2] as the base model. And, we conduct the SFT training based on PyTorch and PEFT library (Mangrulkar et al., 2022) and use the LoRA (Hu et al., 2021) with a rank equal to 8. In addition, we use the Adam optimizer with learning rate of 5e-6 and batch size of 1 for SFT, and we set gradient accumulation steps as 2. We conduct each experiment independently and repeat it 5 times, and report the average results.

3.2 Baselines and Evaluation Metrics

We compare Review-LLM with: (i) closed-source models such as GPT-3.5-Turbo, GPT-4o (Achiam et al., 2023); (ii) open-source models such as, Llama-3-8b (Touvron et al., 2023).

To evaluate the performance of different models, we select ROUGE-1/L (Lin, 2004) and BERT (Kenton and Toutanova, 2019) similar score (BertScore) as evaluation metrics. ROUGE-n measures the n-gram similarity while BertScore measures the semantic similarity in the embedding space between the generated reviews and the reference reviews. We use the sentence transformers (Reimers and Gurevych, 2019) to compute the BertScore. Besides, we conduct a human evaluation experiment to test whether the generated reviews are semantically consistent with the reference reviews.

3.3 Overall Performance

Table 1 compares the performance of our method with several baselines and ablations. It is noted that the GPT-3.5-Turbo and GPT-4o are always better than Llama-3-8b, the reason is that the GPT-series models have a larger number of parameters and are pre-trained on massive data, which could learn more general knowledge. Besides, we find that some baselines without ratings perform better than with ratings, while our fine-tuning method is the opposite. We argue that this is because the user rating information is further pre-trained in our method while baselines not. Overall, our method Review-LLM outperforms all methods (including GPT-3.5-Turbo and GPT-4o) across all metrics, demonstrating the effectiveness of using the item title, review, and rating to personalized fine-tune.

Table 1: Simple evaluation. w/ rating means the prompt contains ratings and w/o rating is vice.

Method \ Metric	ROUGE-1	ROUGE-L	BertScore (mean)
GPT-3.5-turbo (w/ rating)	15.99	9.84	41.52
GPT-3.5-turbo (w/o rating)	16.00	9.81	41.37
GPT-4o (w/ rating)	12.80	8.47	40.12
GPT-4o (w/o rating)	15.41	11.22	41.73
Llama-3-8b (w/ rating)	12.23	8.23	31.30
Llama-3-8b (w/o rating)	13.82	9.59	30.46
Review-LLM (w/ rating)	**31.15**	**26.88**	**49.52**
Review-LLM (w/o rating)	30.47	26.38	48.56

Table 2: Hard evaluation. w/ rating means the prompt contains ratings and w/o rating is vice.

Method \ Metric	ROUGE-1	ROUGE-L	BertScore (mean)
GPT-3.5-turbo (w/ rating)	17.62	10.70	37.45
GPT-3.5-turbo (w/o rating)	16.07	9.89	37.25
GPT-4o (w/ rating)	16.66	9.86	39.21
GPT-4o (w/o rating)	14.51	8.73	38.64
Llama-3-8b (w/ rating)	13.47	8.05	28.38
Llama-3-8b (w/o rating)	13.11	7.89	26.96
Review-LLM (w/ rating)	**21.93**	**16.63**	**39.35**
Review-LLM (w/o rating)	17.82	13.50	35.89

3.4 Negative Review Performance

In our method, we employ user rating information to strengthen the model's understanding of user

[2] https://llama.meta.com/llama3/

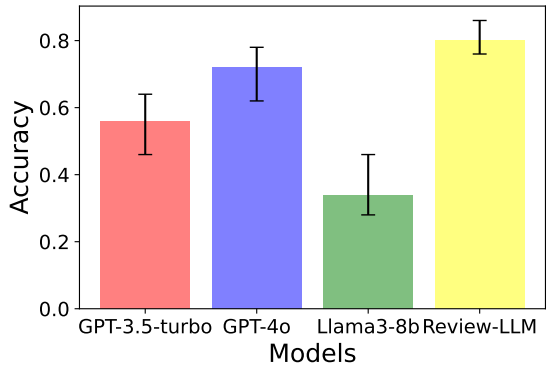

Figure 2: Human Evaluation. The bar is the mean of the model performance, and the error bar represents the max and min accuracy of the model.

preferences for different items to achieve more personalized review generation. In this part, we test the performance of the model on the constructed hard testing set. The different model performance is shown in Table 2. From the results, we can find that all model performance has decreased compared with Table 1. In particular, using Llama3-8b for inference directly, BertScore is reduced to 26.96. We argue that this is because the LLMs might be polite, resulting in insufficient negative information captured during generating reviews. Besides, methods with ratings outperform methods without ratings on semantic similarity, especially Review-LLM, which further confirms the necessity of fusing the rating information for personalized review generation.

3.5 Human Evaluation

In this part, we conduct the human evaluation to test the model performance of review generation. Considering that the generated texts with rating information usually have higher semantic similarity than those without, we only compare the models with rating information here. We randomly select 100 reference reviews and generated reviews from the simple testing set, and hire 10 Ph.D. students who are familiar with review/text generation to evaluate the similarity between generated reviews and reference reviews. If the reference review is semantically similar to the generated reviews, it is marked as 1, otherwise it is marked as 0. Figure 2 shows the percentage of generated reviews marked as 1. From the results, we can see that the designed fine-tuning data and framework could improve the quality of generated reviews and increase their semantic similarity to the reference reviews.

User histories:
(1) Tiger Tigers Eye Gem Round 6-6.5mm Beads/15.5 (Brown & Gold). (2)… (3)… The user purchased a new item: Cousin Jewelry Basics 145-Piece Starter Pack, Antique Gold. %Rating: 1.0. Response:
Reference Review: The head pins didn't match the rest of the finding. They were advertised as antique gold yet the head pins were brash gold. Cannot use them.
Review-LLM: The head pins don't match the rest of the findings. The findings are an antique gold and the head pins are a brash gold and CANNOT be used together. They shouldn't be sold as a set. The head pins are useless!
GPT-4o: Cousin Jewelry Basics 145-Piece Starter Pack, Antique Gold:I am extremely disappointed with this product. Unfortunately, it followed the same issues as previous similar purchases from this brand. The finish of the components does not match, just like the discrepancies I faced with the copper set. Some of the findings appear more tarnished gold rather than antique gold, making them unusable together in a cohesive project. ……
GPT-3.5-turbo: I am extremely disappointed with the Cousin Jewelry Basics 145-Piece Starter Pack in Antique Gold. Just like the previous items I purchased from this brand with lower ratings, this set also fell short of my expectations. The findings are advertised as antique gold, but they look more like a dull bronze color. The head pins do not match the rest of the set, just like my previous experience with the copper set. I had hoped that this time would be different, but unfortunately, it seems like the quality control issues persist. …

Figure 3: Case Study.

3.6 Case Study

To demonstrate the effect of Review-LLM on generating reviews more intuitively, we select the generated reviews (from Review-LLM, GPT-4o, GPT-3.5-Turbo) and the real review for visualization shown in Figure 3. First, we can find the review generated by our model is semantically similar to the real review and brief. In contrast, reviews derived by GPT-3.5-Turbo/GPT-4o are too long and may not be suitable for e-commerce platforms. Second, the generated review of Review-LLM better reflects review writing styles and user sentiment towards the item (we marked those in blue font). This demonstrates that our model could generate high-quality personalized reviews effectively by unifying rich user information with LLMs.

4 Conclusion

This paper presents a framework that leverages Large Language Models (LLMs) for personalized review generation in recommender systems. By aggregating user historical behaviors, including item titles, reviews, and ratings, we construct a comprehensive input prompt to capture user preferences and review writing style. In this way, the model could mitigate the generation of overly polite reviews. Then, we utilize the low-rank adaptation for parameter-efficient fine-tuning, enabling the LLMs to generate reviews for candidate items through supervised fine-tuning. Experimental results show that our fine-tuning method outperforms GPT-3.5-Turbo and GPT-4o in review generation.

5 Limitation

(1) Different individuals may focus on different aspects of a product, such as price, quality, appearance, or durability. While the proposed framework leverages user historical behaviors to capture comprehensive user interest features, it may not fully capture the diversity of individual preferences. (2) The framework primarily focuses on capturing user preferences from historical behaviors without considering the dynamics of user interactions over time. User preferences and writing styles can evolve, and incorporating temporal dynamics could potentially improve the accuracy and personalization of review generation.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Li Dong, Shaohan Huang, Furu Wei, Mirella Lapata, Ming Zhou, and Ke Xu. 2017. Learning to generate product reviews from attributes. In *ECAL*, pages 623–632.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2021. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Jihyeok Kim, Seungtaek Choi, Reinald Kim Amplayo, and Seung-won Hwang. 2020. Retrieval-augmented controllable review generation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2284–2295.

Junyi Li, Siqing Li, Wayne Xin Zhao, Gaole He, Zhicheng Wei, Nicholas Jing Yuan, and Ji-Rong Wen. 2020. Knowledge-enhanced personalized review generation with capsule graph neural network. In *CIKM*, pages 735–744.

Junyi Li, Wayne Xin Zhao, Ji-Rong Wen, and Yang Song. 2019. Generating long and informative reviews with aspect-aware coarse-to-fine decoding. In *ACL*, pages 1969–1979.

Pan Li and Alexander Tuzhilin. 2019. Towards controllable and personalized review generation. In *EMNLP*, pages 3228–3236.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Yichao Lu, Ruihai Dong, and Barry Smyth. 2018. Why i like it: multi-task learning for recommendation and explanation. In *Proceedings of the 12th ACM Conference on Recommender Systems*, pages 4–12.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. Peft: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft.

Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 165–172.

Jianmo Ni, Zachary C Lipton, Sharad Vikram, and Julian McAuley. 2017. Estimating reactions and recommending products with generative models of reviews. In *IJCNLP*, pages 783–791.

Jianmo Ni and Julian McAuley. 2018. Personalized review generation by expanding phrases and attending on aspect-aware representations. In *ACL*, pages 706–711.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Peijie Sun, Le Wu, Kun Zhang, Yanjie Fu, Richang Hong, and Meng Wang. 2020. Dual learning for explainable recommendation: Towards unifying user preference prediction and review generation. In *Proceedings of The Web Conference 2020*, pages 837–847.

Jian Tang, Yifan Yang, Sam Carton, Ming Zhang, and Qiaozhu Mei. 2016. Context-aware natural language generation with recurrent neural networks. *arXiv preprint arXiv:1611.09900*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Lanling Xu, Junjie Zhang, Bingqian Li, Jinpeng Wang, Mingchen Cai, Wayne Xin Zhao, and Ji-Rong Wen. 2024. Prompting large language models for recommender systems: A comprehensive framework and empirical analysis. *arXiv preprint arXiv:2401.04997*.

Hongyu Zang and Xiaojun Wan. 2017. Towards automatic generation of product reviews from aspect-sentiment scores. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 168–177.