

# Metamorphic Evaluation of ChatGPT as a Recommender System

Madhurima Khirbat

RMIT University  
Melbourne, Australia  
madhurima.khirbat@student.rmit.edu.au

Pablo Castells

Universidad Autónoma de Madrid  
Madrid, Spain  
pablo.castells@uam.es

Yongli Ren

RMIT University  
Melbourne, Australia  
yongli.ren@rmit.edu.au

Mark Sanderson

RMIT University  
Melbourne, Australia  
mark.sanderson@rmit.edu.au

## Abstract

With the rise of Large Language Models (LLMs) such as ChatGPT, researchers have been working on how to utilize the LLMs for better recommendations. However, although LLMs exhibit black-box and probabilistic characteristics (meaning their internal working is not visible), the evaluation framework used for assessing these LLM-based recommender systems (RS) are the same as those used for traditional recommender systems. To address this gap, we introduce the metamorphic testing for the evaluation of GPT-based RS. This testing technique involves defining of metamorphic relations (MRs) between the inputs and checking if the relationship has been satisfied in the outputs. Specifically, we examined the MRs from both RS and LLMs perspectives, including rating multiplication/shifting in RS and adding spaces/randomness in the LLMs prompt via prompt perturbation. Similarity metrics (e.g. Kendall  $\tau$  and Ranking Biased Overlap(RBO)) are deployed to measure whether the relationship has been satisfied in the outputs of MRs. The experiment results on MovieLens dataset with GPT3.5 show that lower similarity are obtained in terms of Kendall  $\tau$  and RBO, which concludes that there is a need of a comprehensive evaluation of the LLM-based RS in addition to the existing evaluation metrics used for traditional recommender systems.

## Keywords

Metamorphic Evaluation, ChatGPT, Recommender Systems

### ACM Reference Format:

Madhurima Khirbat, Yongli Ren, Pablo Castells, and Mark Sanderson. 2024. Metamorphic Evaluation of ChatGPT as a Recommender System. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

## 1 Introduction

The rise and advancements of Natural Language Processing (NLP) based systems involving Large Language Models (LLMs), such as

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

*Conference acronym 'XX, June 03–05, 2024.*

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-XXXX-X/18/06

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

Generative Pre-trained Transformer (GPT), BERT, LLaMa and many more have led to the possibility of automation of tasks. Recommender systems are no exception with researchers experimenting with LLM based Recommender Systems and incorporating LLMs into existing recommended systems, such as Collaborative Filtering and Matrix Factorization [1, 2, 3, 4]. To evaluate the performance of LLM-based RS, existing research follows the same evaluation framework for traditional recommender systems [4, 5, 6, 1]. Specifically, in system-centric evaluation [7], users' rating history are used in the prompt for LLMs, and if the recommended items from this LLM-based RS match a user's preference, it is treated as a good recommendation, and traditional metrics (e.g. MAE [4], RMSE [4], NDCG [1], Hit Ratio [2], Recall [2], Precision [2] etc) are deployed accordingly.

However, LLMs are pre-trained models on extensive textual data drawn from various sources, including articles, books, websites, and other publicly accessible written materials and trained on billion parameters [8]. Thus, LLMs exhibit black-box and probabilistic characteristics, meaning their internal working is not visible, which results in different outputs for the same input across different iterations. Moreover, the evaluation becomes extremely challenging when the correct output is not known, such as in the case of top- $k$  recommendations. This leads to the emergence of test oracle problem in evaluating LLM-based recommender systems. The test oracle problem refers to a challenging problem where validating if the computed output is correct or not during the testing of data-intensive softwares, as most of the times the output is not known. So, this leads to issues in the quality of outputs and a thorough evaluation is much needed for these systems: using an LLM such as GPT for recommender systems requires more than just the evaluation of generated output (ratings or recommendations), as typically done in traditional recommender systems.

Metamorphic Testing (MT) [9] is introduced to handle the test oracle problem in the field of software testing. MT is a software testing technique based on the defined generic relations, Metamorphic Relations, between inputs rather than conventional mapping the input with the output [10]. The input is generated using any test case generation strategy and a follow up test input is generated using the defined metamorphic relations (MRs). Both the inputs are tested and the generated outputs are compared. If the defined relation exists in the outputs then the testing is said to be successful [9, 10]. While recent studies show the usage of Metamorphic Testing for the evaluation of LLMs [11], chatbots [12], and traditional RS [10],

there is a gap about the evaluation of MT in LLMs-based RS, which is the focus of this study.

This paper thoroughly evaluates the performance of GPT-based recommender systems using the metamorphic testing techniques from both RS and LLMs perspectives. Specifically, we evaluated four MRs relations: rating multiplication, rating shifting in RS; and adding spaces, randomness in the prompt for ChatGPT. We introduced a framework to control the randomness in the outputs from GPT-based RS so as to form a solid and consistent basis to evaluate the MRs. Both Kendall  $\tau$  and Ranking Biased Overlap (RBO) are selected to measure whether the relationship in MRs are satisfied in their corresponding outputs, so that both the complete ranked recommendation list and their ranking positions are taken into consideration. The contributions of this paper are:

- To the best of our knowledge, this is the first work proposing Metamorphic Evaluation for LLMs-based RS.
- A framework is proposed to control the randomness in the outputs from GPT-based RS.
- Results from MRs in both RS and LLMs indicate that there is need to evaluate LLMs-based RS differently from traditional RS.

## 2 Related Papers

### 2.1 LLM-based Recommender Systems

Due to LLMs' advanced capabilities, LLMs have the potential to significantly transform the field of recommender systems. The use of language models in recommender systems, such as LMRecSys [13], generally utilises prompt generation by providing the user history for few-shot recommendations. The prompt tuning are of two types - continuous vector embeddings as prompts [14] and discrete prompts using text tokens [15]. For example, the "Pretrain, Personalized Prompt, and Predict Paradigm" (P5) [4] combines recommendation tasks and uses instruction-based prompt design with detailed description in a natural language format. Several works indicate that the instruction based prompts are promising for NLP-related tasks as they are flexible and similar to humans communication [16]. The research on LLM-based RS majorly uses instruction-based recommendation for different LLMs such as Alpaca [17], GPT [1], Google Palm [2] which assisted in addressing the sparse user-item matrix problem.

### 2.2 Evaluation Framework in RS

For the evaluation of traditional RS, three primary evaluation framework approaches are commonly used: online evaluation, offline evaluation, and user-based evaluation [18]. Various metrics have been proposed to check the reliability and robustness of the systems. Some of the commonly used metrics for the evaluation of RS are recall, precision, f1-score, mean absolute error (MAE), root mean squared error (RMSE) [19]. To assess the scoring and ranking of a list of items, normalized discounted cumulative gain (nDCG) [20] and mean reciprocal rank (MRR) [21] are the commonly employed metrics. Currently, for the evaluation of LLM based RS, the researchers have been following the same evaluation metrics as of traditional RS, and since LLM-based RS is at its early stage, the evaluation primarily consists of offline evaluation.

## 2.3 Metamorphic Testing

The evaluation of software systems involves choosing input samples for the system execution and then comparing actual outputs with expected results to detect any failures [11, 9]. Recommender systems evaluation works in the similar way: during system-centric evaluation, the dataset is divided into training and test data; the model is trained on the training data and test data is used to evaluate the system. But during the implementation it faces a primary challenge - test oracle problem which refers to the case where validating if the output given by the software is the desired output for any given input [9]. To overcome this problem, a property-based software testing technique, metamorphic testing (MT) was introduced by Mao et. al. [10] for the evaluation of traditional recommender systems. MT works on defining generic relations (metamorphic relations (MR)) between inputs and outputs and look for violations of these relations during testing. In addition, Hyun et al. [11] proposed a metamorphic testing framework (METAL) for LLMs as a language model. Josip Bozic and Franz Wotawa [12] tested the working of chatbots using metamorphic relations by introducing unexpected user responses during the conversation.

### 2.4 Gaps

Existing research mainly works on the MT for traditional RS (e.g. Collaborative Filtering) [10], LLMs tasks [11] and on conversational chatbots [12]. This study focus on the gap of MT for LLMs-based Recommender Systems from both the recommender system perspective and the LLMs perspective to their recommendation capability. To achieve this, we developed a framework to control the randomness introduced by LLMs's probabilistic characteristics so as to examine the MRs in this context.

## 3 Methodology

### 3.1 Overview

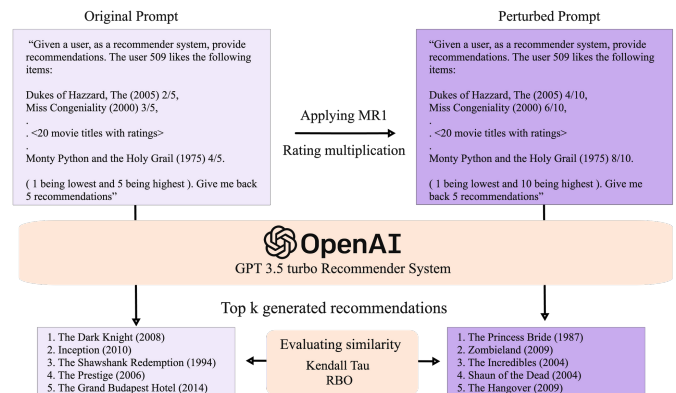


Figure 1: Overview with MR1 as an example

The metamorphic testing operates by generating source test inputs by using any test case generation strategies, and MRs are defined for these inputs on the basis of the properties of the system. The follow up test inputs are generated on the basis of the defined MRs and then the outputs for both the inputs are calculated and evaluated. If the defined MR is maintained between the outputs, the

**Table 1: Example of Prompt Perturbations**

Method	Prompt
Original Prompt	"Given a user, as a recommender system, provide recommendations. The user 509 likes the following items: Dukes of Hazzard, The (2005) 2/5, Miss Congeniality (2000) 3/5, Click (2006) 1/5, Ultraviolet (2006) 2/5, Monty Python and the Holy Grail (1975) 4/5. (1 being lowest and 5 being highest). Give me back 5 recommendations"
MR1: Rating multiplication	"Given a user, as a recommender system, provide recommendations. The user 509 likes the following items: Dukes of Hazzard, The (2005) 4/10, Miss Congeniality (2000) 6/10, Click (2006) 2/10, Ultraviolet (2006) 4/10, Monty Python and the Holy Grail (1975) 8/10. (1 being lowest and 10 being highest). Give me back 5 recommendations"
MR2: Rating shifting	"Given a user, as a recommender system, provide recommendations. The user 509 likes the following items: Dukes of Hazzard, The (2005) 3/6, Miss Congeniality (2000) 4/6, Click (2006) 2/6, Ultraviolet (2006) 3/6, Monty Python and the Holy Grail (1975) 5/6. (2 being lowest and 6 being highest). Give me back 5 recommendations"
MR3: Adding spaces	"Gi ven a u ser , as a re co m mende r syst em, prov ide r e commend ati ons. T he use r 509 lik e s t he follow i n g item s : Dukes of Ha zza rd, T h e (2005 ) 2/5, M i s s Cong e n ial it y ( 2 0 00 ) 3/ 5, Cli ck (2 006) 1/5 , Ul tr a violet ( 20 06) 2 /5 , Monty P y th o n and th e Ho ly Grail (1 975) 4/ 5. ( 1 be ing lowest and 5 be i ng hi ghest ). Giv e me back 5 recommen d a ti o ns."
MR4: Adding random words	"Given a user, as a banana recommender system, grape provide recommendations. The user pear 509 likes banana the following items: grape Dukes of banana Hazzard, The (2005) 2/5, Miss Congeniality (2000) pear 3/5, Click (2006) 1/5, Ultraviolet (2006) 2/5, Monty Python and the Holy Grail (1975) 4/5. (1 being lowest and 5 being grape highest). Give me back 5 banana recommendations, banana one movie per line and don't give any explanation"

program is said to succeed. We leverage the same testing strategy to check for the robustness of the GPT-based RS, and the overview of the proposed Metamorphic Testing framework for GPT-based RS is shown in Figure 1, which showing the overall process of one MR relation in GPT-based RS. The proposed methodology includes three main components: i) Prompt Construction; ii) Metamorphic Relations; iii) Output Refinement for randomness control.

### 3.2 Prompt construction

Here, we used the instruction based few-shot discrete prompts which is close to human language. The prompt was based on "person pattern" prompt which was proposed by White et. al. [22]. The prompt comprises both static and variable components. The static portion specifies the action GPT is tasked with, while the variable part provides the context or parameters for executing that action [2]. A generic example of the prompt is given below:

Prompt - "Given a user, as a recommender system, provide recommendations. The user {user} likes the following items: {movies}. (1 being lowest and 5 being highest). Give me back 5 recommendations"

An example of the Original Prompt and the MRs in this study are shown in Table 1.

### 3.3 Metamorphic Relations

**3.3.1 MT of the Ratings in LLM Based RS.** To evaluate the top-k generated recommendations, we considered different ratings that represent the same preference of the user. This involves applying prompt perturbation on just the ratings part of the prompt, keeping rest exactly the same as the original to check if the model is sensitive to the ratings scale and generated recommendations are consistent that represents user's preferences.

To observe the impact of changing of the rating scale on the top-k generated items, following [10], we defined two metamorphic relations in ratings from RS that were used for the evaluation:

- **MR1: Rating multiplication** - For all the items in the prompt for each user, the ratings for each item and the total rating is multiplied by a constant  $\lambda$  integer, e.g. that original ratings,  $R/5$  becomes  $\lambda[R/5]$ .

- **MR2: Rating shifting** - For all the items in the prompt for each user, the ratings for each item and the total rating is shifted, either increased or decreased, by a constant  $\lambda$  integer such that original ratings,  $R/5$  becomes  $(\lambda + R)/(\lambda + 5)$ .

**3.3.2 MT of Prompts in LLM based RS.** To evaluate the robustness of the LLM, we do metamorphic testing on the language part of

the prompt to check the impact on the performance. This involves applying linguistic variations on the prompt to see how language variations can affect's model performance. By comparing the responses to these paraphrased inputs, we can assess whether the LLM consistently produces accurate and relevant answers despite the changes in semantic structure of the prompt.

Following [11], we defined two metamorphic relations in language from LLMs perspective that were used for the evaluation. We have used semantic-preserving prompt perturbation for the linguistic manipulation of the prompt. This evaluation helps uncover potential weaknesses in the model's understanding and processing of differently formatted input using diverse linguistic contexts.

- **MR3: Adding spaces** - This relation works by inserting spaces between characters in the given prompt containing user history and rating.

- **MR4: Adding random** - This relation works by inserting random words in the given prompt containing user history and rating, such as "apple", "grape", "banana", "pear".

### 3.4 Output Refinement for Randomness Control

ChatGPT introduces an element of randomness while generating outputs to incorporate diversity. Namely, this randomness can lead to varied results for the generated recommendations during each iteration [1]. This creates difficulty in evaluating the GPT-based RS if output is very different the each time. To handle this, we controlled the randomness of the generated output through prompt engineering by considering the following two variables:

- $l$ : the number of items provided in the prompt.
- $k$ : the number of items in top- $k$  recommendation list.

To check for the impact on these variables on the recommendations and randomness during different iterations, similarity metrics were used - Kendall  $\tau$ , Ranking Biased Overlap and overlap between the lists. Finally,  $l$  and  $k$  are determined when consistent outputs are obtained in different iterations in terms of the above three similarity metrics. Detailed experiment results about this is in Section 4.

## 4 Experiment

### 4.1 Experiment Configuration

**Dataset:** For the experiments we used the MovieLens 100k dataset by Grouplens[23]. The dataset contains 100,000 ratings and 3,600 tag applications applied to 9,000 movies by 610 users. **GPT-based RS:** The recommendations are generated using GPT 3.5 turbo model. **Evaluation of MTs:** The relationship between input and outputs

of MTs are measured with the following similarity metrics: Kendall  $\tau$  [7], Ranking Biased Overlap (RBO) [24] and overlap ratio between the top- $k$  recommendation lists. The  $t$ -test with a 95% confidence level has been applied to evaluate whether the results is statistically significant.

## 4.2 Results of Randomness Control

As discussed in Sec 3.4, we control the randomness of the generated outputs from GPT-based RS with two variables:  $l$  the number of items provided in the prompt, and  $k$  the number of items in top- $k$  recommendation list.

The results for  $k$  is shown in Table 2. In the prompt construction here, we used all movies in user history if their corresponding ratings are greater than 3, which indicates a positive preference. We conducted two iterations for each user and evaluated similarity by comparing the recommendation lists for each user from these two iterations. It is observed that the bigger the  $k$ , the more randomness in the generated recommendation list. Specifically, the top 5 recommendations perform the best. The higher RBO scores suggest that the recommendations at the top of the list are more consistent across iterations compared to that of larger  $k$ , as RBO assigns greater weight to top-ranked items while Kendall  $\tau$  assesses the overall order of items in the list. So, to ensure consistency in metamorphic testing, we conduct experiments focusing on the top 5 recommendations.

**Table 2: Results of  $k$  in different top- $k$  recommendations**

$k$	Kendall $\tau$	RBO	Overlap Ratio
$k = 5$	0.8784	0.9632	0.8146
$k = 10$	0.8673	0.951	0.9445
$k = 30$	0.7679	0.9068	0.8801
$k = 50$	0.7096	0.8522	0.6870

The results for  $l$  is shown in Table 3 with various  $l$  values, ranging from 5 to 30. Specifically, we ran 10 iterations for  $l$  value and calculated the average Kendall  $\tau$ , RBO and overlap ratio. It is observed that the lists generated from different  $l$  are similar as indicated by higher similarity. We opted for a selection of  $l = 20$  in the prompt for the proposed metamorphic testing since it represents a broader range of user interests while requiring less computational time.

**Table 3: Results of  $l$  the number of items per user in the prompt for top-5 recommendations**

$l$	Kendall $\tau$	RBO	Overlap Ratio
$l = 5$	0.9120	0.9707	0.8495
$l = 10$	0.8757	0.9663	0.8088
$l = 20$	0.9116	0.9768	0.9623
$l = 30$	0.9179	0.9791	0.9665

## 4.3 Results of MRs

Following the above experiments about controlling randomness in GPT-based RS, we set  $k = 5$  and  $l = 20$  to examine the metamorphic relations: MR1, MR2, MR3 and MR4. Specifically, we ran 10

iterations of each defined MRs. The generated lists were compared against a baseline list consisting of one iteration of top-5 recommendations generated using 20 movies with no modifications. This comparison served as the reference point to assess the effectiveness of various modifications applied during the testing process.

**Table 4: Results of MRs**

Method	Kendall $\tau$ (SD)	RBO (SD)	Overlap ratio (SD)
No change (baseline)	0.9116 (0.0055)	0.9768 (0.0027)	0.9623 (0.0030)
MR1: Multiply	0.4829 (0.0082)	0.8496 (0.0021)	0.8146 (0.0025)
MR2: Addition	0.4966 (0.0057)	0.8460 (0.0014)	0.8174 (0.0028)
MR3: Spaces	0.0640 (0.0121)	0.4710 (0.0081)	0.4882 (0.0057)
MR4: Random words	0.2295 (0.0117)	0.6802 (0.0050)	0.6863 (0.0089)

Table 4 shows the results of evaluating different Metamorphic Relations (MRs) (averaged over 10 runs with standard deviation (SD)), and the unpaired  $t$ -test with 95% confidence level is deployed to examine whether the difference between each MR output and the baseline list: the corresponding  $p$ -values are  $< 0.0001$ , which means they are all statistically significantly different to the baseline list.

- **MR1:** The MR1 involves multiplying the ratings by a constant  $\lambda$  while representing the same user preferences. After introducing this MR to the prompt, it can be seen that the average Kendall  $\tau$  value dropped significantly, suggesting a reduced overlap between the lists. Interestingly, the average RBO value is still over 0.8, indicating that despite modifications the top items of the generated lists for each user are still similar as compared to the bottom. The small standard deviation indicates that there is a consistency in the results.
- **MR2:** Similar to MR1, this relation involves adding a constant  $\lambda$  in all the ratings, thereby changing the rating scale while keeping the same user preferences. The Kendall score is slightly higher than the average Kendall score of MR1 but the similarity in the order of the items generated is still poor. The RBO value reveals a similar trend, representing similarity among the top items of the list.
- **MR3:** This MR is introduced to the prompt to check for the performance under language variation. After introducing spaces in the prompt, the average Kendall and RBO values dropped very low indicating minimal correlation and overlap between the lists. The small standard deviations also show an inconsistency in the generated output.
- **MR4:** In this MR, random words are added in the prompt to check for performance. MR4 performs better than MR3 with higher average Kendall and RBO values but overall the similarity is low when compared with MR1 and MR2 which means that the generated lists are very different.

## 4.4 Discussion

It can be observed that MR1 and MR2 performed better than MR3 and MR4. This suggests that changes in the rating scale had a relatively more predictable and consistent impact on the recommendation outcomes. In contrast, altering the semantic structure of the prompt led to less desirable results, indicating a greater degree of variability or unpredictability in the system's response. Moreover, while the average Kendall  $\tau$  values showed a significant drop across all Metamorphic Relations (MRs), the average values for

Rank-Biased Overlap (RBO) didn't decrease as drastically. This suggests that despite changing the prompts after the application of MRs, the top items in the generated list remained similar across users. However, as we move down to the bottom of the list, the order of the recommendations changes, as indicated by average Kendall's  $\tau$ . In addition, there are certain limitations to the experiments which include internal validity. We repeated the experiments 10 times for each result to get an overall performance of the GPT-based RS under different situations.

## 5 Conclusion

This paper proposes metamorphic testing for LLM-based RS using metamorphic relations. Our experiments conducted on the GPT 3.5 using the MovieLens dataset revealed insights into how these metamorphic relations influence the recommendation lists generated by the system. Our findings revealed a noticeable decrease in similarity in the generated outputs using multiple MRs and prompt variations. This exploration serves as an initial step towards integrating metamorphic testing into the evaluation framework for LLM-based RS. Moving forward we intend to test multiple MRs in our future work to gain comprehensive understanding of the system.

## References

- [1] J. Liu, C. Liu, P. Zhou, R. Lv, K. Zhou, and Y. Zhang, "Is chatgpt a good recommender? a preliminary study," 2023.
- [2] D. Di Palma, G. M. Biancofiore, V. W. Anelli, F. Narducci, T. Di Noia, and E. Di Sciascio, "Evaluating ChatGPT as a recommender system: A rigorous approach." [Online]. Available: <http://arxiv.org/abs/2309.03613>
- [3] S. Dai, N. Shao, H. Zhao, W. Yu, Z. Si, C. Xu, Z. Sun, X. Zhang, and J. Xu, "Uncovering ChatGPT's capabilities in recommender systems," in *Proceedings of the 17th ACM Conference on Recommender Systems*. ACM, pp. 1126–1132. [Online]. Available: <https://dl.acm.org/doi/10.1145/3604915.3610646>
- [4] S. Geng, S. Liu, Z. Fu, Y. Ge, and Y. Zhang, "Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5)," in *Proceedings of the 16th ACM Conference on Recommender Systems*, ser. RecSys '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 299–315. [Online]. Available: <https://doi.org/10.1145/3523227.3546767>
- [5] S. Kim, H. Kang, S. Choi, D. Kim, M. Yang, and C. Park, "Large language models meet collaborative filtering: An efficient all-round llm-based recommender system," 2024.
- [6] Y. Gao, T. Sheng, Y. Xiang, Y. Xiong, H. Wang, and J. Zhang, "Chat-rec: Towards interactive and explainable llms-augmented recommender system," 2023.
- [7] A. H. Jadidinejad, C. Macdonald, and I. Ounis, "The simpson's paradox in the offline evaluation of recommendation systems," vol. 40, no. 1, pp. 4:1–4:22. [Online]. Available: <https://dl.acm.org/doi/10.1145/3458509>
- [8] X. Lin, W. Wang, Y. Li, S. Yang, F. Feng, Y. Wei, and T.-S. Chua, "Data-efficient fine-tuning for llm-based recommendation," 2024.
- [9] T. Y. Chen, F.-C. Kuo, H. Liu, P.-L. Poon, D. Towey, T. H. Tse, and Z. Q. Zhou, "Metamorphic testing: A review of challenges and opportunities," vol. 51, no. 1, pp. 1–27. [Online]. Available: <https://dl.acm.org/doi/10.1145/3143561>
- [10] C. Mao, J. Chen, X. Yi, and L. Wen, "An empirical study on metamorphic testing for recommender systems," vol. 169, p. 107410. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0950584924000156>
- [11] S. Hyun, M. Guo, and M. A. Babar, "METAL: Metamorphic testing framework for analyzing large-language model qualities." [Online]. Available: <http://arxiv.org/abs/2312.06056>
- [12] J. Bozic and F. Wotawa, "Testing chatbots using metamorphic relations," in *Testing Software and Systems*, C. Gaston, N. Kosmatov, and P. Le Gall, Eds. Springer International Publishing, pp. 41–55.
- [13] Y. Zhang, H. Ding, Z. Shui, Y. Ma, J. Zou, A. Deoras, and H. Wang, "Language models as recommender systems: Evaluations and limitations," in *NeurIPS 2021 Workshop on I (Still) Can't Believe It's Not Better*, 2021. [Online]. Available: <https://www.amazon.science/publications/language-models-as-recommender-systems-evaluations-and-limitations>
- [14] Y. Gu, X. Han, Z. Liu, and M. Huang, "Ppt: Pre-trained prompt tuning for few-shot learning," 2022.
- [15] T. Gao, A. Fisch, and D. Chen, "Making pre-trained language models better few-shot learners," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds. Online: Association for Computational Linguistics, Aug. 2021, pp. 3816–3830. [Online]. Available: <https://aclanthology.org/2021.acl-long.295>
- [16] A. Efrat and O. Levy, "The turking test: Can language models understand instructions?" 2020.
- [17] A. Acharya, B. Singh, and N. Onoe, "Llm based generation of item-description for recommendation system," in *Proceedings of the 17th ACM Conference on Recommender Systems*, ser. RecSys '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 1204–1207. [Online]. Available: <https://doi.org/10.1145/3604915.3610647>
- [18] D. Jannach, A. Manzoor, W. Cai, and L. Chen, "A survey on conversational recommender systems," vol. 54, no. 5, pp. 105:1–105:36. [Online]. Available: <https://dl.acm.org/doi/10.1145/3453154>
- [19] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, p. 5–53, jan 2004. [Online]. Available: <https://doi.org/10.1145/963770.963772>
- [20] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of ir techniques," *ACM Trans. Inf. Syst.*, vol. 20, no. 4, p. 422–446, oct 2002. [Online]. Available: <https://doi.org/10.1145/582415.582418>
- [21] D. Jannach, P. Pu, F. Ricci, and M. Zanker, "Recommender systems: Past, present, future," vol. 42, no. 3, pp. 3–6. [Online]. Available: <https://onlinelibrary.wiley.com/doi/10.1609/aimag.v42i3.18139>
- [22] J. White, Q. Fu, S. Hays, M. Sandborn, C. Olea, H. Gilbert, A. Elnashar, J. Spencer-Smith, and D. C. Schmidt, "A prompt pattern catalog to enhance prompt engineering with chatgpt," 2023.
- [23] Mar 2021. [Online]. Available: <https://grouplens.org/datasets/movielens/1m/>
- [24] W. Webber, A. Moffat, and J. Zobel, "A similarity measure for indefinite rankings," *ACM Trans. Inf. Syst.*, vol. 28, no. 4, nov 2010. [Online]. Available: <https://doi.org/10.1145/1852102.1852106>