The Thirty-Ninth AAAI Conference on Artificial Intelligence (AAAI-25)

CoT4Rec: Revealing User Preferences Through Chain of Thought for Recommender Systems

Weiqi Yue, Yuyu Yin, Xin Zhang, Binbin Shi, Tingting Liang *, Jian Wan
Hangzhou Dianzi University, Hangzhou, China
{yueweiqi, yinyuyu, zhangxin, shibinbin, liangtingting, wanjian}@hdu.edu.cn

Abstract

Large Language Models (LLMs) offer groundbreaking advancements in recommender systems through superior text analysis and decision-making support. However, integrating LLMs into recommender systems still suffers from the problems of identifier uninterpretability and lack of transparency. To address these issues and fully leverage the capabilities of LLMs, we propose a chain of thought (CoT) based recommendation framework called CoT4Rec which employs LLMs as data enhancers for user preference analysis. Initially, we design a CoT reasoning strategy that can derive more behaviorally-aligned user preference features by clustering users' historical interactions. Subsequently, we propose a two-stage recommendation model that not only makes full use of the world knowledge embedded in LLMs but also generates a logically transparent reasoning path. By integrating a user preference analyzer early in the recommendation pipeline, the model deeply analyzes users' historical interactions, helping to enhance the personalization and transparency of the recommender system. CoT4Rec demonstrates superior performance over existing state-of-the-art models in recommendation tasks across four public datasets, achieving improvements ranging from 2.2% to 12.2%.

Code — https://github.com/815382636/CoT4Rec

Introduction

Large Language Models (LLMs) have accumulated a substantial knowledge base by learning from vast amounts of text data, demonstrating excellence in understanding and generating natural language texts (Zhao et al. 2023; Li, Zhang, and Chen 2023b; Chang et al. 2024). With LLMs achieving significant advantages in handling natural language processing (NLP) tasks, researchers have begun to explore the possibility of applying these models to recommender systems. Models such as P5 (Geng et al. 2022), M6-rec (Cui et al. 2022), LC-rec (Zheng et al. 2024) and Recformer (Li et al. 2023) have injected new vitality into the field of recommender systems by integrating the extensive knowledge base of LLMs. These models have not only enhanced recommendation performance but also significantly improved the portability of recommender systems.

Current recommender systems have greatly benefited from the integration of LLMs, yet there remain two significant limitations that need to be addressed: (1) **Identifier Uninterpretability**. As shown in Figure 1 (a), most recommender systems still employ discrete identifiers (e.g., user IDs and item IDs) to represent users and items, mirroring the approach of traditional recommendation models (Geng et al. 2022; Zheng et al. 2024). These methods focus solely on collaborative filtering between users and items, failing to leverage the world knowledge integrated by LLMs and neglecting the impact at the semantic level. (2) **Lack of Transparency**. As depicted in Figure 1 (b), some approaches utilize the textual metadata (e.g., title) to represent users and items, thereby making full use of LLMs' world knowledge (Cui et al. 2022; Li et al. 2023). However, these studies rely on implicit reasoning for recommendations, which might introduce hallucination issues caused by LLMs, thereby reducing the accuracy of the recommendations and significantly impacting the user experience.

To address these limitations, some studies have considered employing generative models to capture textual information from user reviews, extracting key insights to refine the recommendation process (Lu et al. 2023; Li, Zhang, and Chen 2023a). However, it is crucial to recognize that users often struggle to summarize the underlying motivations for their actions. The explanations derived from user feedback do not truly achieve a cause-to-effect relationship, thus falling short of truly understanding the reasons behind user behaviors. Furthermore, user feedback is typically characterized by insufficiency or noise in most datasets.

Considering these issues, our goal is to develop a LLM-based recommendation architecture that has a clear recommendation logic and can fully utilize the world knowledge of LLMs. Building on this, our design introduces a user preference analyzer before the personalized recommendation. The analyzer can generate user preferences based on user's historical interactions and construct a transparent reasoning chain from user behavior to personalized recommendation. The challenge lies in ensuring that the generated user preferences are both authentic and targeted, as well as constructing a coherent recommendation model with a transparent reasoning process.

To this end, in this paper, we introduce CoT4Rec, an innovative sequential recommendation framework that breaks

---

*Corresponding author

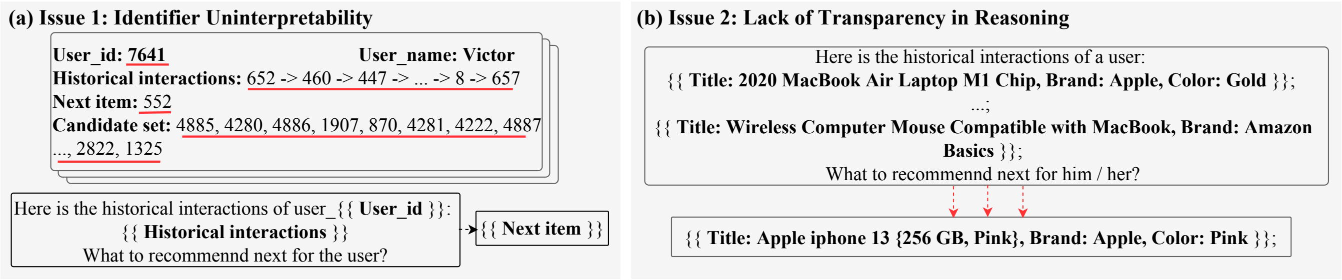


Figure 1: Limitations of existing LLM-based recommender systems. (a) Identifier Uninterpretability. (b) Lack of Transparency in Reasoning.

down the recommendation process into two logical reasoning stages: user preferences generation and personalized recommendation. This phased strategy is designed to ensure that the recommendation outcomes can accurately map the core logic of user preferences formation. In the specific implementation, our first task is to construct the user preference analyzer. To ensure that user preferences are authentic, effective, and targeted, we devise an approach for the automated generation of chain of thought (CoT) by clustering users' historical interactions, and use it to deduce users' behavioral preferences. Subsequently, during the user preference analyzer training, the model focuses on learning and summarizing users' behavioral preferences to form an in-depth understanding of users' personalized needs. Finally, the personalized recommendation model makes the final recommendation by considering both the user's historical interactions and the generated behavioral preferences. The proposed CoT4Rec establishes the recommendation logic as the generation of user behavioral preferences, enabling the enhancement of recommendation outcomes through the refinement and standardization of the user preference analyzer.

The contributions of our work can be summarized as follows:

- We design a CoT Reasoning Strategy for user preferences generation, which can effectively integrate user behavior characteristics with similar preferences by clustering users' historical interactions.
- We propose a novel sequential recommendation framework named CoT4Rec which constructs a transparent reasoning chain from user behavior to personalized recommendation.
- We demonstrate the effectiveness and superior performance of the proposed CoT4Rec through experiments on four public datasets. We further provide popularity bias analysis and case study to explore the insights of CoT4Rec.

Related Work

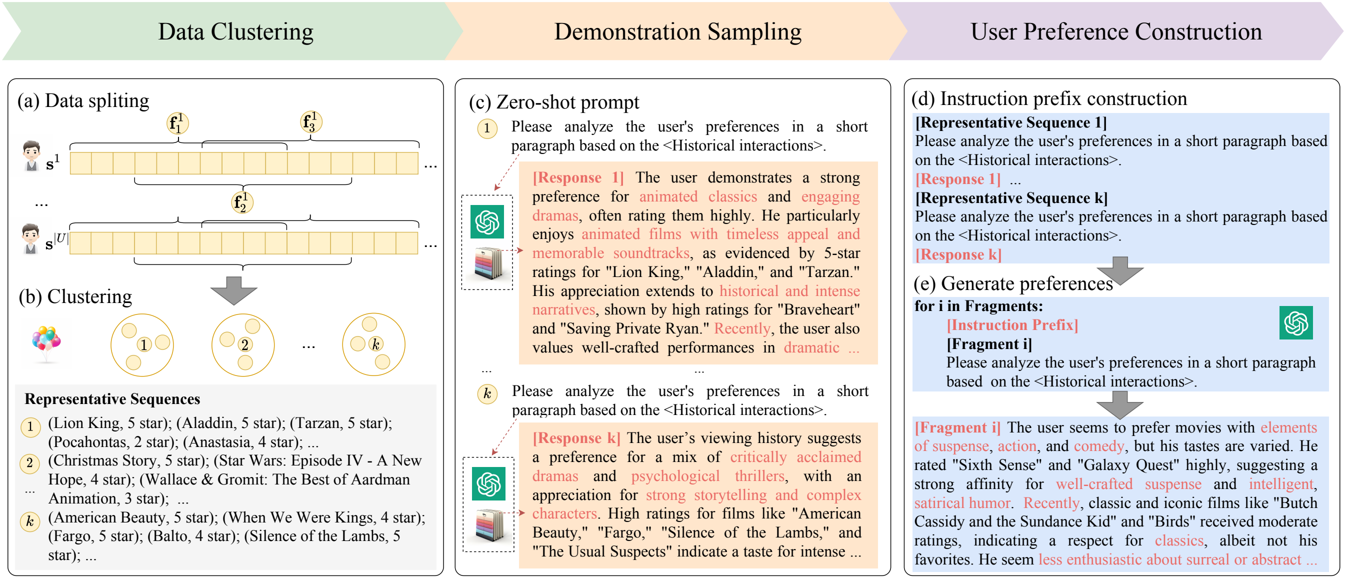
LLM for Recommendation

In recent years, recommender system based on pre-trained language model (PLMs) have garnered widespread attention from the researchers (Sun et al. 2019; Cui et al. 2022; Wang et al. 2023a). Especially with the emergence and continuous advancement of LLMs, they have not only demonstrated remarkable capabilities in various NLP tasks but also opened up possibilities for constructing more versatile and efficient recommender systems (Raffel et al. 2020; Touvron et al. 2023; Gao et al. 2023; Xi et al. 2024; Wang et al. 2023b).

LLM-based recommender systems can fine-tune LLMs to capture task-specific feature representations, which helps mitigate the issue of data sparsity and significantly enhances the interpretability of recommendation tasks. To be specific, P5 (Geng et al. 2022) converts all data (user-item interactions, user description and user comments) into natural language sequences, leveraging the inferential capabilities of generative models for recommendation. LLM-rec (Lyu et al. 2023) explores a variety of prompt strategies and enhances the performance of personalized recommendations through data augmentation. Recformer (Li et al. 2023) models user preferences and item features as language representations, effectively alleviating the data sparsity issue in recommendation tasks. TALLRec (Bao et al. 2023) introduces an efficient tuning framework for the integration of recommendation tasks with LLMs, and the fine-tuned framework has demonstrated strong cross-domain generalization capabilities. Chat-rec (Gao et al. 2023) utilizes the ChatGPT interface to replace data training. By designing prompts and connecting to external databases, it alleviates the cold start problem for new users and supporting cross-domain recommendations.

Chain of Thought (CoT)

CoT is a fine-tuning method for LLMs that enhances contextual understanding by guiding the construction of intermediate steps during the generation process. This enables the model to produce more coherent and logically text outputs (Zhang et al. 2023; Wang et al. 2022a). CoT is initially introduced to enhance the complex reasoning capabilities of LLMs, achieving remarkable effects in tasks such as mathematical word problems, commonsense reasoning, and symbolic manipulation (Wei et al. 2022). (Kojima et al. 2022) demonstrates that LLMs are decent zero-shot reasoning machines, and their generative principles already reflect CoT reasoning. However, due to the hallucination problem and industry adaptability issues of LLMs, the environmental adaptability of zero-shot CoT is relatively poor. Few-shot CoT stimulates CoT reasoning capabilities through effective



Representative Sequences

① (Lion King, 5 star); (Aladdin, 5 star); (Tarzan, 5 star); (Pocahontas, 2 star); (Anastasia, 4 star); ...

② (Christmas Story, 5 star); (Star Wars: Episode IV - A New Hope, 4 star); (Wallace & Gromit: The Best of Aardman Animation, 3 star); ...

...

k (American Beauty, 5 star); (When We Were Kings, 4 star); (Fargo, 5 star); (Balto, 4 star); (Silence of the Lambs, 5 star); ...

① Please analyze the user's preferences in a short paragraph based on the <Historical interactions>.

[Response 1] The user demonstrates a strong preference for animated classics and engaging dramas, often rating them highly. He particularly enjoys animated films with timeless appeal and memorable soundtracks, as evidenced by 5-star ratings for "Lion King," "Aladdin," and "Tarzan." His appreciation extends to historical and intense narratives, shown by high ratings for "Braveheart" and "Saving Private Ryan." Recently, the user also values well-crafted performances in dramatic ...

...

k Please analyze the user's preferences in a short paragraph based on the <Historical interactions>.

[Response k] The user's viewing history suggests a preference for a mix of critically acclaimed dramas and psychological thrillers, with an appreciation for strong storytelling and complex characters. High ratings for films like "American Beauty," "Fargo," "Silence of the Lambs," and "The Usual Suspects" indicate a taste for intense ...

[Representative Sequence 1]
Please analyze the user's preferences in a short paragraph based on the <Historical interactions>.
[Response 1] ...
[Representative Sequence k]
Please analyze the user's preferences in a short paragraph based on the <Historical interactions>.
[Response k]

for i in Fragments:
  [Instruction Prefix]
  [Fragment i]
  Please analyze the user's preferences in a short paragraph based on the <Historical interactions>.

[Fragment i] The user seems to prefer movies with elements of suspense, action, and comedy, but his tastes are varied. He rated "Sixth Sense" and "Galaxy Quest" highly, suggesting a strong affinity for well-crafted suspense and intelligent, satirical humor. Recently, classic and iconic films like "Butch Cassidy and the Sundance Kid" and "Birds" received moderate ratings, indicating a respect for classics, albeit not his favorites. He seem less enthusiastic about surreal or abstract ...

Figure 2: The construction process of the CoT reasoning strategy. User historical behavior sequences are segmented and categorized into different clusters. Representative sequences are then selected from each cluster to construct zero-shot prompts, which are employed to infer user preferences. These inferred preferences serve as a prefix for instructions to invoke LLMs to generate the core logic for all sequences.

manual demonstrations, leading to stronger performance. Nevertheless, the demonstration of the reasoning process is manually designed, which brings a great deal of manual overhead. Therefore, recent work has focused on how to use ensemble methods to lift this limitation. (Zhou et al. 2022) designs prompts by decomposing complex problems into multiple sub-problems and solving sub-problems sequentially. (Wang et al. 2022b) introduces a self-consistency decoding strategy to sample multiple outputs of LLMs. (Zhang et al. 2022) involves sampling multiple outputs from the LLMs and employs a clustering strategy to automatically construct CoT.

CoT Reasoning Strategy

A recommender system should provide personalized and accurate recommendations based on the transparent reasoning path from user historical behavior to recommendation results. Therefore, the primary task is to conduct an in-depth analysis of user behavior to generate a core logic that accurately reflects user preferences. Compared to manually constructing user preference logic, using prompt rules in conjunction with LLMs is a more efficient approach. However, current analysis indicates that LLMs are still not perfect zero-shot reasoners. Traditional zero-shot prompts suffer from messy content characteristics and hallucinations, which directly affect the core logic formation of user preferences (Kojima et al. 2022). To address these problems, inspired by (Zhang et al. 2022), we design a CoT reasoning strategy for automatically clustering users' historical interactions, aiming to generate user preferences with high-level semantics and logical coherence. As shown in Figure 2, our CoT reasoning strategy consists of three main stages: (i) Data Clustering: historical behavioral sequences of users are segmented and categorized into multiple clusters. (ii) Demonstration Sampling: a representative sequence is selected from each cluster, and user preferences are generated using simple heuristic zero-shot prompts. (iii) User Preference Construction: the sampled user preferences are employed as instruction prefix to establish the core logic that forms all user preferences.

Data Clustering

Directly applying the zero-shot prompt to invoke LLMs may result in overly broad user preferences and introduce the hallucination issues associated with LLMs. In addition, the number of historical interactions of users in recommendation tasks is variable. Users with longer histories may generate more fragments and dominate the representation learning. As a result, we employ the sliding window to segment user's historical interactions into user-agnostic fragments, and use clustering analysis on fragments to capture representative preference information in the temporal dimension. Specifically, given users' historical behavior sequences $\mathbf{S} = [\mathbf{s}^1, \mathbf{s}^2, ..., \mathbf{s}^{|U|}]$, where $\mathbf{s}^i$ denotes the historical interactions of user i and $|U|$ represents the number of users. Initially, for each element $\mathbf{s}^i$, we slice the sequential items with a window size of w and form a new set of sequence fragments $\mathbf{F}^i = [\mathbf{f}_1^i, \mathbf{f}_2^i, ..., \mathbf{f}_{L_i}^i]$, where L_i denotes the number of fragments obtained from the sequences of user i. By integrating the fragment sets of $|U|$ users, we can obtain a complete sequence fragment set $\mathbf{F} = [\mathbf{F}^1, \mathbf{F}^2, ..., \mathbf{F}^{|U|}]$. Next, we utilize the Sentence-BERT (Reimers and Gurevych 2019) to calculate the vector representation for each sequence within $\mathbf{F}$, and average the contextualized vectors to form a fixed-size sequence representation. Finally, we apply the k-means algorithm to cluster the sequence representations and generate k clusters. For each cluster j, the elements are arranged in a list $\mathbf{C}_j = [\mathbf{c}_{j,1}, \mathbf{c}_{j,2}, ...]$ in ascending order based on their distance to the centroid of cluster j.

Demonstration Sampling

In this stage, we apply zero-shot prompts to generate user preference samples based on the representative sequence within each cluster j. Subsequently, we reorganize the generated preferences with the interaction fragments to create exemplary preface statements for CoT. Specifically, for each cluster j, the sequence closest to the cluster center is selected as one of the most representative sequences for sampling. The specific prompt for the representative sequence $\mathbf{c}_{j,r}$ can be represented as follows:

> *Historical interactions:* $\{\mathbf{c}_{j,r}\}$.
> *Please analyze the user's preference in a short paragraph based on the historical interactions.*

The constructed prompt is used as the query to invoke the LLM, and the resulting outputs, after being organized and adjusted, serve as the corresponding preference $\mathbf{p}_j$.

User Preference Construction

For user preference construction, we utilize the corresponding preference samples generated as instruction prefix to construct the core logic of preferences for all user fragments. Specifically, we establish the connection between the representative sequence $\mathbf{c}_{j,r}$ and the corresponding preference $\mathbf{p}_j$, denoted as $prefix_j = (\mathbf{c}_{j,r}, \mathbf{p}_j)$. Next, the instruction prefix $Prefix = [prefix_1, prefix_2, ..., prefix_k]$ can be constructed. In the end, we generate personalized preferences for all fragments $\mathbf{F}$ using the CoT Reasoning Strategy. For user fragment $\mathbf{f}_*$, the constructed CoT prompt can be formalized as follows:

> *Prefix:* $\{Prefix\}$.
> *Historical interactions:* $\{\mathbf{f}_*\}$.
> *Please analyze the user's preference in a short paragraph based on the historical interactions.*

The generated user preferences are used as inputs for training the user preference analyzer in the subsequent recommendation model. The preference analyzer enhances the model's ability to understand and predict individual user needs.

Recommendation Model

Our CoT reasoning strategy with LLMs can generate user preference with high-level semantics and intuitive logicality. However, relying on external LLMs to generate user preference logic can lead to issues of real-time responsiveness and lack of flexibility, significantly hindering the practical application of recommender systems. To address these issues and establish the causal relationship between user preferences and recommendation outcomes, we propose a two-stage recommendation framework called CoT4Rec. CoT4Rec introduces a preference analyzer to learn directly generating user preferences without LLMs, thereby enabling the recommender system with the capability to autonomously generate preference logic.

Framework Overview

The proposed CoT4Rec consists of two components: preference analyzer and recommender. Both components utilize the flan-t5-base (Chung et al. 2024) as the backbone model which is fine-tuned through supervised learning. It is noteworthy that the inputs and outputs for each component are different. The overall framework of our CoT4Rec is shown in Figure 3.

The preference analyzer takes the user preferences generated during the CoT Reasoning Strategy stage as labels to learn the generation of preference logic. Once trained, the analyzer can independently generate preference logic with high-level semantics and strict logic, without relying on external LLMs. Specifically, the analyzer's inputs X represent prompts constructed based on the user historical behavior sequences. For user fragment $\mathbf{f}_*$, it can be described as follows:

> *Historical interactions:* $\{\mathbf{f}_*\}$.
> *Please analyze the user's preferences based on the historical interactions.*

The recommder utilizes the integrated recommder prompt as input, training its generative recommendation capabilities. To be specific, prompt template used as the input X' is formed by sequentially connecting the user fragment $\mathbf{f}_*$, the generated preference $\mathbf{p}_*$, and the candidate set $\mathbf{D}_*$. The formal representation of X' can be formalized as follows:

> *Historical interactions:* $\{\mathbf{f}_*\}$.
> *Personal preference:* $\{\mathbf{p}_*\}$.
> *Candidate set:* $\{\mathbf{D}_*\}$.
> *Based on the historical interactions and the personal preferences, please recommend an item for the user from the candidate set.*

The proposed CoT4Rec framework, with the assistance of preference analyzer and recommender, effectively integrates user preference logic to generate the top n recommended items based on user historical behavior sequences.

Training and Inference

The preference analyzer and the recommender share the same architecture but are fine-tuned independently. During the training phase, the model is fine-tuned for specific recommendation tasks through supervised learning to enhance its predictive accuracy and adaptability. In the inference phase, the preference analyzer first generates the user preference logic based on their historical behavior sequences. Subsequently, the recommender integrates the logic with users' historical interactions to generate personalized recommendations.

Training We employ the flan-t5-base as the backbone model for preference analyzer component and recommender component, and format the training data in a sequence-to-sequence manner to accommodate language generation

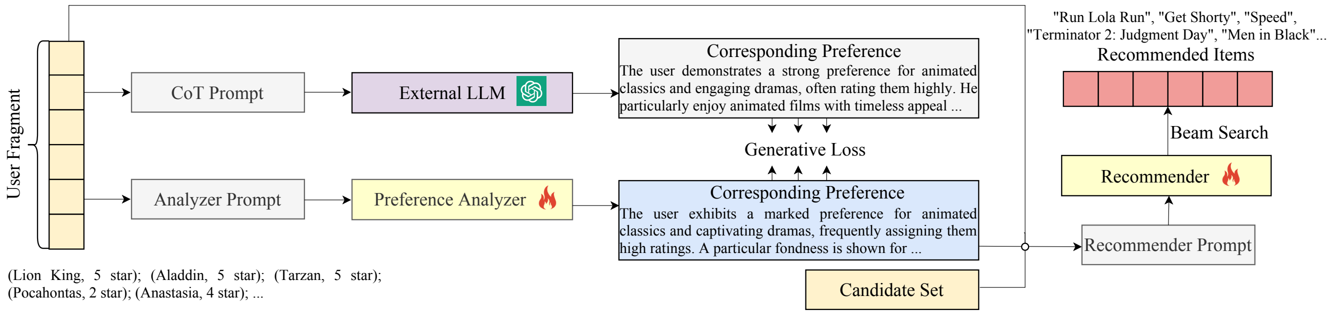


Figure 3: Overview of the CoT4Rec framework. CoT4Rec consists of two components: preference analyzer and recommender. Both components share the same model architecture but differ in inputs and outputs. The preference analyzer takes the user preferences generated in CoT Reasoning Strategy stage as labels, autonomously learns to generate preference logic based on users' historical interaction fragments. The recommder utilizes the integrated prompt formed with users' historical interaction fragments, preference logic, and candidate set as input, training its generative recommendation capabilities.

tasks. Specifically, given the language input X or X', the probability of generating target Y of length N can be written as follows (take X for example):

$$p(Y|X) = \prod_{i=1}^{N} p_\theta(Y_i|X, Y_{<i}) \tag{1}$$

where Y_i is the i^{th} token of Y, $Y_{<i}$ denotes the token before Y_i and $p_\theta(Y_i|X, Y_{<i})$ is implemented with a Transformer-based network.

Furthermore, the fine-tuned model learns the model parameters θ by minimizing the negative log-likelihood of the generation target in an end-to-end manner as follows:

$$\mathcal{L}_\theta = \sum_{(X,Y)\in\mathcal{B}} \sum_{i=1}^{|Y|} \log P(Y_i|X, Y_{<i}) \tag{2}$$

where $(X, Y) \in \mathcal{B}$ represents a pair of instruction and response (label) in one batch.

Inference The goal of the preference analyzer is to generate user preferences based on their historical behavior sequences, while the objective of the recommender is to select the top n items from the entire candidate set that best match the given user preferences. Both objectives are achieved through a sequence-to-sequence inference format. After fine-tuning, the preference analyzer and the recommender can independently generate their respective targets based on the instruction prompts. In addition, unlike the preference analyzer, the recommender requires an item list as target output, so its decoder module applies beam search to generate a list of potential next items and evaluate them within the context of the candidate set. The decoding processes can be formulated as follows:

$$R = \text{Beam\_Search}(\mathcal{D}, X', B) \tag{3}$$

where $\mathcal{D}$ represents the decoder function, and B denotes the beam size.

Furthermore, to ensure the quality of generation, the probabilities of illegal item indices are set to 0 when calculating logits.

Dataset	#Users	#Items	#Inters	#Sparsity
ML100K	943	1,349	99,287	92.20%
ML1M	6,040	3,416	999,611	95.16%
Movies	123,960	50,052	1,697,533	99.97%
Electronics	192,403	63,001	1,689,188	99.99%

Table 1: Statistics of the preprocessed datasets.

Experiment

Experimental Setup

Datasets To evaluate the effectiveness of CoT4Rec, comprehensive experiments are conducted on four public datasets from various sources: 1) **MovieLens**[1]: MovieLens is a widely used public dataset focusing on movie ratings, with data sourced provided by users from IMDB and movie databases. ML100K and ML1M, two benchmark datasets are selected, each containing approximately 100,000 and 1,000,000 user interactions, respectively. 2) **Amazon**[2]: The Amazon dataset includes user reviews and ratings from the Amazon platform across 24 different categories. Movies and Electronics , the two million-level categories are chosen as the benchmark datasets. The statistical information of the four datasets is summarized in Table **??**.

Baselines The following representative sequence recommendation models are adopted as baselines for comparison with the proposed CoT4Rec.

- **BPR-MF** (Rendle et al. 2012) is a pairwise ranking model that uses Biased Matrix Factorization (Bias-MF) as the prediction function for recommendations.
- **SVD++** (Koren 2008) is a matrix factorization-based method, which utilizes both explicit and implicit feedback to enhance the accuracy of recommendations.
- **DMF** (Xue et al. 2017) can simultaneously consider both explicit ratings and implicit feedback to achieve better

---

[1] https://grouplens.org/datasets/movielens

[2] http://jmcauley.ucsd.edu/data/amazon

Methods	ML100K				ML1M				Movies				Electronics			
	N@5	N@10	HR@5	HR@10	N@5	N@10	HR@5	HR@10	N@5	N@10	HR@5	HR@10	N@5	N@10	HR@5	HR@10
BPR-MF	0.3024	0.3659	0.4501	0.6486	0.3526	0.4073	0.5137	0.6826	0.3962	0.4392	0.5346	0.6676	0.3092	0.3472	0.4179	0.5354
SVD++	0.3087	0.3685	0.4586	0.6433	0.4857	0.5293	0.6399	0.7745	0.3918	0.4335	0.5224	0.6512	0.2775	0.3172	0.3848	0.5077
DMF	0.3023	0.3661	0.4480	0.6450	0.3043	0.3589	0.4500	0.6182	0.4006	0.4455	0.5455	0.6843	0.2775	0.3143	0.3783	0.4922
NeuMF	0.3002	0.3592	0.4490	0.6316	0.3322	0.3903	0.4899	0.6694	0.3791	0.4211	0.5134	0.6429	0.3026	0.3358	0.4031	0.5123
GRU4Rec	0.3564	0.4122	0.5134	0.6856	0.4923	0.5267	0.6571	0.7626	0.4038	0.4459	0.5287	0.6688	0.3154	0.3551	0.4284	0.5511
STAMP	0.3560	0.4070	0.5159	0.6730	0.4701	0.5098	0.6273	0.7498	0.3935	0.4366	0.5246	0.6577	0.3095	0.3489	0.4196	0.5430
DuoRec	0.3184	0.3735	0.4655	0.6373	0.5127	0.5458	0.6531	0.7551	0.4003	0.4338	0.5113	0.6149	0.3337	0.3682	0.4374	0.5442
NCR	0.3760	0.4240	0.5456	0.6943	0.4783	0.5229	0.6495	0.7866	0.4255	0.4670	0.5611	0.6891	0.3499	0.3878	0.4639	0.5812
OpenP5	0.3542	0.4157	0.5174	0.7090	0.5139	0.5479	0.6576	0.7932	0.4033	0.4269	0.5375	0.6499	0.3746	0.4054	0.4803	0.5943
CoT4Rec	**0.3857**	**0.4411**	**0.5643**	**0.7295**	**0.5252**	**0.5676**	**0.6880**	**0.8191**	**0.4659**	**0.5158**	**0.6178**	**0.7730**	**0.3941**	**0.4329**	**0.5152**	**0.6346**
Improv.	2.6%	4.0%	3.4%	2.9%	2.2%	3.6%	4.6%	3.3%	9.5%	10.4%	10.1%	12.2%	5.2%	6.8%	7.3%	6.9%

Table 2: The performance comparison between CoT4Rec and baselines. **Bold** represents the best result. Underline represents the second best result.

recommendations.

- **NeuMF** (He et al. 2017) is a network-based collaborative filtering model that employs a multi-layer perceptron to model the interactions between users and items.
- **GRU4Rec** (Hidasi et al. 2015) is an RNN model for session-based recommendations, using extensive user history for suggestions.
- **STAMP** (Liu et al. 2018) captures both user's general interests from long-term sessions and current interests from short-term memory for recommendations.
- **DuoRec** (Qiu et al. 2022) leverages contrastive learning to reshapes the distribution of sequences for recommendations.
- **NCR** (Chen et al. 2021) combines representation learning and logical reasoning, bridging differentiable neural networks with symbolic inference for reasoning.
- **OpenP5** (Xu, Hua, and Zhang 2024) extracts the sequence recommendation component from P5 (a unified recommendation sharing framework), and optimizes the construction of item indexing for sequential recommendations.

Evaluation Metrics We employ two widely used metrics for evaluation: Normalized Discounted Cumulative Gain (N@K) and Hit Rate (HR@K), where the value of K is set to 5 and 10. Following previous works (Shi et al. 2020), we conduct data preprocessing on four datasets, and apply the leave-one-out strategy for evaluation. Specifically, for each user's historical behavior sequence, the most recent item and the second most recent item are used as the test data and validation data, respectively. The remaining historical behavior sequences are used for training. For each correct interaction item in the validation and test sets, 100 unrelated items are randomly selected and shuffled with it. The shuffled set of 101 items is then ranked for evaluation.

Implementation Details For CoT Reasoning Strategy, we employ GPT-3.5-turbo-1106 as the interface for LLM calls. For data clustering, we employ the sliding window to segment user's historical interactions into user fragments. The sliding window size is set to 5, with a stride of 3 for each slide. Then, we apply k-means algorithm to cluster these fragments into 6 clusters. Representative sequences from each cluster are used with the zero-shot prompt to construct the exemplary preface statements for CoT. For the reference analyzer and the recommender training, we utilize the flan-t5-base as backbone and conduct independent full-parameter fine-tuning at each component. We employ the optimizer AdamW (Loshchilov, Hutter et al. 2017) to adaptively adjust the model, setting the maximum number of epochs to 10, the learning rate to 5e-5, and the weight decay to 0.01. The experiments are conducted on 2 NVIDIA V6000 48G GPUs.

Overall Performance

Table 2 presents a comparison of CoT4Rec and baseline models on four datasets. The results are highlighted with bold for the best performance and underline for the second-best performance. Overall, CoT4Rec outperforms all baseline models on four datasets. The improvements compared with the best baselines range from 2.2% to 12.2%. Notably, the performance gains in the "Movies" and "Electronics" datasets are particularly significant, exceeding 5% in both cases. These substantial performance improvements are attributed to the use of LLMs as data enhancers to generate more accurate user preferences, allowing the model to leverage the rich world knowledge within LLMs more effectively.

In addition, several points are noteworthy. Traditional MF-based methods (BPR-MF and SVD++) and MLP-based methods (DMF and NeuMF) perform relatively poorly due to the omission of the impact of users' historical behaviors on recommendations. Sequential recommendation mod-

Methods	ML100K				ML1M				Movies				Electronics			
	N@5	N@10	HR@5	HR@10	N@5	N@10	HR@5	HR@10	N@5	N@10	HR@5	HR@10	N@5	N@10	HR@5	HR@10
A: **CoT4Rec**	**0.3857**	**0.4411**	**0.5643**	**0.7295**	**0.5252**	**0.5676**	**0.6880**	**0.8191**	**0.4659**	**0.5158**	**0.6178**	**0.7730**	**0.3941**	**0.4329**	**0.5152**	**0.6346**
B: w/o PA	0.3603	0.4283	0.5315	0.6992	0.4990	0.5244	0.6486	0.7811	0.4132	0.4529	0.5553	0.6643	0.3588	0.3831	0.4547	0.5855
C: w/o TS	0.3823	0.4369	0.5569	0.7187	0.5198	0.5534	0.6783	0.8134	0.4573	0.5122	0.6017	0.7685	0.3852	0.4211	0.5097	0.6249

Table 3: The ablation study of CoT4Rec. **Bold** represents the best result.

els, such as GRU4Rec, STAMP and DuoRec, consider the influence of time series and show significant performance improvement over traditional methods. Reasoning-based model NCR and LLM-based model OpenP5 have achieved excellent and stable performance in all baselines. NCR stands out because it introduces a logical reasoning module into the sequential recommendation process, enhancing the model's cognitive capabilities. OpenP5's success is mainly due to its use of collaborative indexing to integrate the set of item IDs and the use of multiple classic instruction prompts for model training. However, although NCR and OpenP5 achieve remarkable performance, they still fall behind the proposed CoT4Rec. CoT4Rec's advantage lies in its construction of a reasoning path from user historical behavior to personalized recommendations, which is more targeted and transparent than NCR's symbolic reasoning path. Furthermore, by using item metadata instead of discrete identifiers (OpenP5), CoT4Rec effectively leverages the world knowledge integrated by LLMs.

Ablation Study

CoT4Rec introduces a user preference analyzer and structures the recommender system into a two-stage reasoning framework, effectively enhancing the quality of recommendations. To verify the effectiveness of the user preference analyzer (PA) and the two-stage recommendation strategy (TS), we conduct an ablation study on four datasets. Table 3 provides the results, comparison between (A: CoT4Rec) and (B: w/o PA) shows that the LLMs with strong reasoning capabilities can serve as an effective data augmentation tool to generate user preferences, significantly improving the performance of subsequent recommendations. Furthermore, the comparison between (A: CoT4Rec) and (C: w/o TS) demonstrates the effectiveness of the user preference analyzer constructed in CoT4Rec. Once the user preference analyzer is fully trained, CoT4Rec can independently generating user preferences and establishing a relatively transparent reasoning path from user historical behavior to personalized recommendations, without relying on external LLMs.

Further Analysis

Prompt Construction Approaches for LLM Table 4 describes the comparative performance of different prompt construction approaches in enhancing the usability of user preference generated by GPT-3.5. Zero-Shot-CoT refers to the approach that does not rely on pre-defined prompt examples but directly utilizes GPT-3.5 to construct user preferences. Random-Manual-CoT involves manually constructing CoT prompts by randomly selecting sequences of user's historical behavior. The proposed CoT Reasoning strategy employs a clustering mechanism on sequences of user's historical behavior, thereby establishing a CoT framework. Table 4 illustrates that the proposed method demonstrates superior performance and enhanced stability. Compared with Zero-Shot-CoT method, the proposed method standardizes the reasoning process, effectively leveraging the emergent capabilities of LLMs while significantly reducing the occurrence of hallucinations. Furthermore, in contrast to the Random-Manual-CoT method, the proposed method not only has stronger stability but also significantly reduces the manual labor costs.

Methods	ML100K		Electronics	
	N@10	HR@10	N@10	HR@10
Zero-Shot-CoT	0.4311	0.7002	0.4027	0.6151
Random-Manual-CoT	0.4337	0.7189	0.4287	0.6290
Ours	**0.4411**	**0.7295**	**0.4329**	**0.6346**

Table 4: The comparative recommendation performance of user preferences generated by different prompt construction approaches.

Impact of Sequence Length Figure 4 illustrates the impact of varying sequence lengths on NCR, OpenP5, and CoT4Rec. Overall, CoT4Rec demonstrates a significant improvement in performance across different sequence lengths compared to the baseline methods. In addition, it is observed that NCR performs well when the sequence length is short, but its performance noticeably deteriorates in longer sequence environments. In contrast, the OpenP5 method is more adept at handling longer sequences. This phenomenon arises because NCR incorporates logical operations to summarize user preferences, which makes it more suitable for scenarios with shorter sequences. As an LLM-based model, OpenP5's self-attention sequence encoding and positional encoding enable it to excel in processing long sequences. CoT4Rec is adaptable to environments with varying sequence lengths because the incorporation of user preferences enhances its feature representation in shorter sequences, while its modeling process based on LLM enables it to adapt to long sequence environments.

Popularity Bias Popularity analysis in recommender systems is crucial as it reveals the tendency of recommendation algorithms to favor items that are already widely popular.

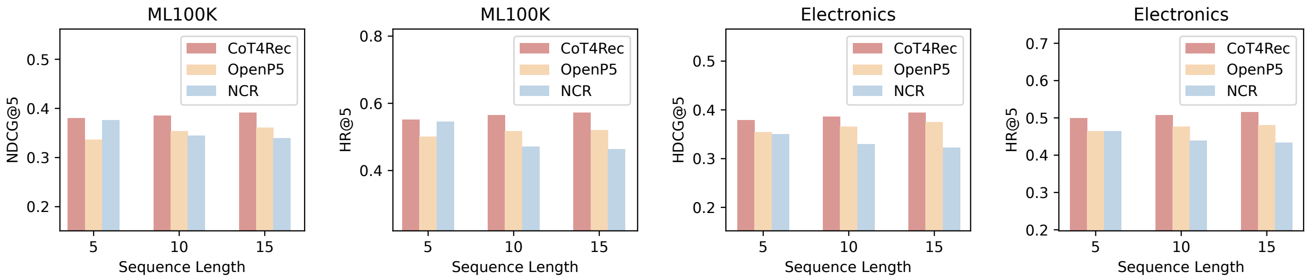


Figure 4: Performance comparison w.r.t. sequence length.

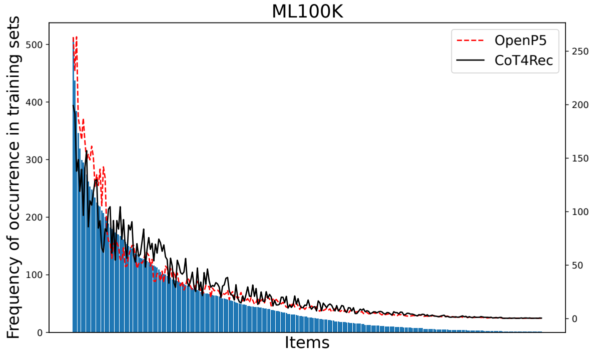
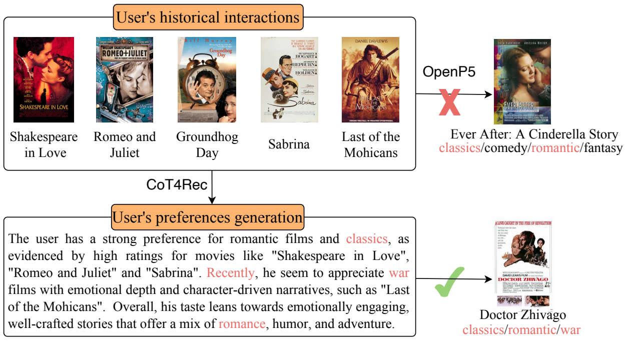


Figure 5: Analysis of popularity bias. The blue bars illustrate the frequency of items in the train set, while the red and black lines represent the cumulative count of the top 5 alternatives in inference for OpenP5 and CoT4Rec.

Popularity bias can lead to homogenization of recommendation results, affecting personalized recommendations and diversity. Moreover, this bias can exacerbate the long-tail effect in the real-world recommendation domain, where a few hot items receive excessive attention while a large number of niche items are neglected. To address this challenge, the popularity bias analysis in the CoT4Rec method is conducted through experiments. As shown in Figure 5, compared to the OpenP5 method, CoT4Rec can more evenly recommend items of different frequencies of occurrence. In contrast, the OpenP5 method tends to focus on recommending popular head items. CoT4Rec significantly alleviates popularity bias, promoting fairness and diversity in the recommender system.

Case Study

CoT4Rec integrates a user preference analyzer, establishing a clearer reasoning path from the user historical behaviors to personalized recommendations. Figure 6 provides a specific sample where CoT4Rec successfully generates the correct recommendations, while OpenP5 produces incorrect ones. This phenomenon is mainly due to CoT4Rec's ability to summarize the general patterns of user historical behaviors and capture the user's most recent behavioral patterns (the user recently watched "The Last of the Mohicans" which is a war film). CoT4Rec not only provides a clear path for subsequent recommendations, improving the recommendation effect, but also increases the interpretability of the recommendation process.



The user has a strong preference for romantic films and classics, as evidenced by high ratings for movies like "Shakespeare in Love", "Romeo and Juliet" and "Sabrina". Recently, he seem to appreciate war films with emotional depth and character-driven narratives, such as "Last of the Mohicans". Overall, his taste leans towards emotionally engaging, well-crafted stories that offer a mix of romance, humor, and adventure.

Figure 6: Different ability to recommend subsequent items between OpenP5 and CoT4Rec.

Conclusion

In this work, we design a CoT Reasoning Strategy for user preference generation, which can integrate user behavior characteristics with similar preferences. In addition, we introduce an innovative sequence recommendation framework named CoT4Rec, which divides the recommendation process into two logical reasoning stages: user preference analysis and personalized recommendation. This separation strategy ensures that the recommendation results can accurately map the core logic of the generated user preference formation. The experimental evaluation demonstrates the effectiveness of our proposed CoT4Rec.

Acknowledgments

This work is supported by National Science and Technology Major Project under No. 2022ZD0116700, the National Science Foundation of China under No. U20A20173 and No. 62402151, "Pionee" and "Leading Goose" R&D Program of Zhejiang Province under No. 2024C01166, the Foundation of Yunnan Key Laboratory of Service Computing under No. YNSC24124.

References

Bao, K.; Zhang, J.; Zhang, Y.; Wang, W.; Feng, F.; and He, X. 2023. Tallrec: An effective and efficient tuning framework to align large language model with recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, 1007–1014.

Chang, Y.; Wang, X.; Wang, J.; Wu, Y.; Yang, L.; Zhu, K.; Chen, H.; Yi, X.; Wang, C.; Wang, Y.; et al. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3): 1–45.

Chen, H.; Shi, S.; Li, Y.; and Zhang, Y. 2021. Neural collaborative reasoning. In *Proceedings of the Web Conference 2021*, 1516–1527.

Chung, H. W.; Hou, L.; Longpre, S.; Zoph, B.; Tay, Y.; Fedus, W.; Li, Y.; Wang, X.; Dehghani, M.; Brahma, S.; et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70): 1–53.

Cui, Z.; Ma, J.; Zhou, C.; Zhou, J.; and Yang, H. 2022. M6-rec: Generative pretrained language models are open-ended recommender systems. *arXiv preprint arXiv:2205.08084*.

Gao, Y.; Sheng, T.; Xiang, Y.; Xiong, Y.; Wang, H.; and Zhang, J. 2023. Chat-rec: Towards interactive and explainable llms-augmented recommender system. *arXiv preprint arXiv:2303.14524*.

Geng, S.; Liu, S.; Fu, Z.; Ge, Y.; and Zhang, Y. 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.

He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; and Chua, T.-S. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, 173–182.

Hidasi, B.; Karatzoglou, A.; Baltrunas, L.; and Tikk, D. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*.

Kojima, T.; Gu, S. S.; Reid, M.; Matsuo, Y.; and Iwasawa, Y. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35: 22199–22213.

Koren, Y. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 426–434.

Li, J.; Wang, M.; Li, J.; Fu, J.; Shen, X.; Shang, J.; and McAuley, J. 2023. Text is all you need: Learning language representations for sequential recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1258–1267.

Li, L.; Zhang, Y.; and Chen, L. 2023a. On the relationship between explanation and recommendation: Learning to rank explanations for improved performance. *ACM Transactions on Intelligent Systems and Technology*, 14(2): 1–24.

Li, L.; Zhang, Y.; and Chen, L. 2023b. Prompt distillation for efficient llm-based recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 1348–1357.

Liu, Q.; Zeng, Y.; Mokhosi, R.; and Zhang, H. 2018. STAMP: short-term attention/memory priority model for session-based recommendation. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 1831–1839.

Loshchilov, I.; Hutter, F.; et al. 2017. Fixing weight decay regularization in adam. *arXiv preprint arXiv:1711.05101*, 5.

Lu, H.; Ma, W.; Wang, Y.; Zhang, M.; Wang, X.; Liu, Y.; Chua, T.-S.; and Ma, S. 2023. User perception of recommendation explanation: Are your explanations what users need? *ACM Transactions on Information Systems*, 41(2): 1–31.

Lyu, H.; Jiang, S.; Zeng, H.; Xia, Y.; and Luo, J. 2023. Llm-rec: Personalized recommendation via prompting large language models. *arXiv preprint arXiv:2307.15780*.

Qiu, R.; Huang, Z.; Yin, H.; and Wang, Z. 2022. Contrastive learning for representation degeneration problem in sequential recommendation. In *Proceedings of the fifteenth ACM international conference on web search and data mining*, 813–823.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140): 1–67.

Reimers, N.; and Gurevych, I. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Rendle, S.; Freudenthaler, C.; Gantner, Z.; and Schmidt-Thieme, L. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*.

Shi, S.; Chen, H.; Ma, W.; Mao, J.; Zhang, M.; and Zhang, Y. 2020. Neural logic reasoning. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 1365–1374.

Sun, F.; Liu, J.; Wu, J.; Pei, C.; Lin, X.; Ou, W.; and Jiang, P. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, 1441–1450.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Wang, W.; Xu, Y.; Feng, F.; Lin, X.; He, X.; and Chua, T.-S. 2023a. Diffusion recommender model. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 832–841.

Wang, X.; Wei, J.; Schuurmans, D.; Le, Q.; Chi, E.; Narang, S.; Chowdhery, A.; and Zhou, D. 2022a. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Wang, X.; Wei, J.; Schuurmans, D.; Le, Q.; Chi, E.; and Zhou, D. 2022b. Rationale-augmented ensembles in language models. *arXiv preprint arXiv:2207.00747*.

Wang, Y.; Jiang, Z.; Chen, Z.; Yang, F.; Zhou, Y.; Cho, E.; Fan, X.; Huang, X.; Lu, Y.; and Yang, Y. 2023b. Recmind:

Large language model powered agent for recommendation. *arXiv preprint arXiv:2308.14296*.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35: 24824–24837.

Xi, Y.; Liu, W.; Lin, J.; Cai, X.; Zhu, H.; Zhu, J.; Chen, B.; Tang, R.; Zhang, W.; and Yu, Y. 2024. Towards open-world recommendation with knowledge augmentation from large language models. In *Proceedings of the 18th ACM Conference on Recommender Systems*, 12–22.

Xu, S.; Hua, W.; and Zhang, Y. 2024. Openp5: An open-source platform for developing, training, and evaluating llm-based recommender systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 386–394.

Xue, H.-J.; Dai, X.; Zhang, J.; Huang, S.; and Chen, J. 2017. Deep matrix factorization models for recommender systems. In *IJCAI*, volume 17, 3203–3209. Melbourne, Australia.

Zhang, Z.; Zhang, A.; Li, M.; and Smola, A. 2022. Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*.

Zhang, Z.; Zhang, A.; Li, M.; Zhao, H.; Karypis, G.; and Smola, A. 2023. Multimodal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923*.

Zhao, W. X.; Zhou, K.; Li, J.; Tang, T.; Wang, X.; Hou, Y.; Min, Y.; Zhang, B.; Zhang, J.; Dong, Z.; et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Zheng, B.; Hou, Y.; Lu, H.; Chen, Y.; Zhao, W. X.; Chen, M.; and Wen, J.-R. 2024. Adapting large language models by integrating collaborative semantics for recommendation. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, 1435–1448. IEEE.

Zhou, D.; Schärli, N.; Hou, L.; Wei, J.; Scales, N.; Wang, X.; Schuurmans, D.; Cui, C.; Bousquet, O.; Le, Q.; et al. 2022. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.