Enhancing Sequential Recommenders with Augmented Knowledge from Aligned Large Language Models

Yankun Ren
Ant Group
Hangzhou, China
yankun.ryk@antgroup.com

Zhongde Chen
Ant Group
Hangzhou, China
zhongde.czd@antgroup.com

Xinxing Yang
Ant Group
Hangzhou, China
xinxing.yangxx@antfin.com

Longfei Li
Ant Group
Hangzhou, China
longyao.llf@antfin.com

Cong Jiang
Ant Group
Hangzhou, China
jiangcong.jc@antgroup.com

Lei Cheng
Ant Group
Hangzhou, China
lei.chenglei@antgroup.com

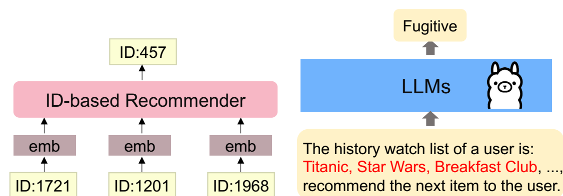
Bo Zhang
Ant Group
Hangzhou, China
aicui.zb@antgroup.com

Linjian Mo
Ant Group
Hangzhou, China
linyi01@antgroup.com

Jun Zhou*
Ant Group
Hangzhou, China
jun.zhoujun@antfin.com

ABSTRACT

Recommender systems are widely used in various online platforms. In the context of sequential recommendation, it is essential to accurately capture the chronological patterns in user activities to generate relevant recommendations. Conventional ID-based sequential recommenders have shown promise but lack comprehensive real-world knowledge about items, limiting their effectiveness. Recent advancements in Large Language Models (LLMs) offer the potential to bridge this gap by leveraging the extensive real-world knowledge encapsulated in LLMs. However, integrating LLMs into sequential recommender systems comes with its own challenges, including inadequate representation of sequential behavior patterns and long inference latency. In this paper, we propose SeRALM (Enhancing Sequential Recommenders with Augmented Knowledge from Aligned Large Language Models) to address these challenges. SeRALM integrates LLMs with conventional ID-based sequential recommenders for sequential recommendation tasks. We combine text-format knowledge generated by LLMs with item IDs and feed this enriched data into ID-based recommenders, benefitting from the strengths of both paradigms. Moreover, we develop a theoretically underpinned alignment training method to refine LLMs' generation using feedback from ID-based recommenders for better knowledge augmentation. We also present an asynchronous technique to expedite the alignment training process. Experimental results on public benchmarks demonstrate that SeRALM significantly improves the performances of ID-based sequential recommenders. Further, a series of ablation studies and analyses corroborate SeRALM's proficiency in steering LLMs to generate more pertinent and advantageous knowledge across diverse scenarios.



(a) ID-based recommenders. (b) LLM-as-recommender.

Figure 1: Comparison between (a) ID-based recommenders and (b) the LLM-as-recommender paradigm.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

Sequential Recommendation; Large Language Models; Alignment

ACM Reference Format:
Yankun Ren, Zhongde Chen, Xinxing Yang, Longfei Li, Cong Jiang, Lei Cheng, Bo Zhang, Linjian Mo, and Jun Zhou. 2024. Enhancing Sequential Recommenders with Augmented Knowledge from Aligned Large Language Models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24), July 14–18, 2024, Washington, DC, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3626772.3657782

1 INTRODUCTION

Recommender systems [16, 29, 30] are extensively utilized across various online platforms (such as Amazon, iTunes, and Netflix) to

*Corresponding author.

SIGIR '24, July 14–18, 2024, Washington, DC, USA

ACM ISBN 979-8-4007-0431-4/24/07
https://doi.org/10.1145/3626772.3657782

forecast user preferences. In practical applications, the nature of user interactions is inherently dynamic, exhibiting temporal fluctuations. Consequently, it is imperative to accurately capture the chronological patterns in user activities to generate relevant recommendations. Achieving such an understanding of user behaviors constitutes the fundamental objective of sequential recommendation algorithms, which aim to predict the next item a user may interact with based on their previously interacted items.

A multitude of methods have been proposed to improve the performance of the sequential recommendation. These methods encompass Markov Chains [8, 20], RNN/CNN models [9, 13, 23, 27] and self-attention-based mechanisms [5, 11, 33]. As depicted in Figure 1(a), conventional sequential recommendation models, which are ID-based, transform items into unique identifiers and generate a specific trainable embedding for each ID. These models utilize elaborate neural architectures to capture sequential behavior patterns from item embeddings. Although these methods show promise and possess a profound insight into behavior patterns in sequential interactions, they lack comprehensive, real-world knowledge about the items. This deficiency constrains the overall effectiveness of ID-based recommendation models. For instance, in situations where the training data is insufficient, these methods tend to underperform due to inadequately trained item embeddings [14].

Recent advancements in Large Language Models (LLMs) have substantially transformed the learning paradigm across diverse research domains, demonstrating significant promise in bridging the gap between traditional recommender systems and real-world knowledge [31]. By assimilating a comprehensive corpus of internet text, LLMs like GPT-4 [2] and LLaMA [24], have encapsulated an extensive spectrum of real-world knowledge [25].

Recent studies have been motivated by the capabilities of LLMs in providing real-world knowledge and reasoning. These studies aim to explore the application of LLMs in sequential recommendation tasks. The current research focuses on transforming sequential recommendation data into textual formats to harness LLMs as sequential recommenders [4, 7, 15]. For example, as shown in Figure 1(b), in methods such as P5 [7], a user's past interactions with items are transformed into a textual prompt: "The history watchlist of a user is: Titanic, Star Wars, Breakfast Club, ..., recommend the next item to the user." Following this prompt creation, LLMs are fine-tuned and employed to perform the sequential recommendation. This paradigm leverages the real-world knowledge of items implicitly stored in LLMs and utilizes the reasoning ability of LLMs to make recommendations. Moreover, LLMs can employ this embedded knowledge in conjunction with user interactions to deduce individual preferences. Such insights, absent from conventional recommendation datasets, enrich the original data, thereby augmenting the efficacy of the recommendation.

However, despite the advantages of accessing real-world knowledge and yielding promising results in some few-shot settings [1], these methods have yet to surpass the accuracy and efficiency of traditional ID-based recommenders. The limitations of such an LLM-as-recommender paradigm may be ascribed to several critical shortcomings, including: **(i) Inadequate representation of sequential patterns and collaborative signals contained in item IDs.** As mentioned before, each item in the prompt is encoded using its textual metadata, such as titles and descriptions [1, 3]. While effectively utilizing the linguistic capabilities of LLMs, this inadequately capture users' sequential behavior patterns and collaborative information contained in the item IDs. **(ii) Long inference latency.** The substantial number of parameters inherent to LLMs renders their direct application in recommender systems impractical for real-world industrial use.

In this paper, we aim to enhance conventional ID-based sequential recommenders by integrating the vast real-world knowledge storage and reasoning capabilities of LLMs. By doing so, we strive to achieve both effectiveness and efficiency in sequential recommendations. However, despite the appealing capabilities of LLMs, extracting relevant and beneficial knowledge from LLMs is far from straightforward. The challenge arises from the extensive and diverse information contained within LLMs, which can introduce noise detrimental to the performance of recommender systems. To mitigate this issue, it is crucial to tailor the knowledge generated by LLMs to the specific needs of the sequential recommendation task. Consider the context of movie recommendations: augmented knowledge about the intricate technical specifications of the cameras used in film production may hold little value for most users and can contribute to noise in the recommender systems. In contrast, providing a summarization of movie themes of a user's watchlist, such as "he/she likes complex relationships between people," can be significantly more impactful for modeling user preferences. Therefore, careful alignment of LLM-generated knowledge with the objective of the sequential recommendation task is essential for effective knowledge augmentation.

To address the identified challenges, we propose an approach named Enhancing Sequential Recommenders with Augmented Knowledge from Aligned Large Language Models (SeRALM). SeRALM is a model-agnostic method that aligns Large Language Models (LLMs) with conventional ID-based sequential recommenders, enabling the targeted application of sequential recommendation tasks. Through this alignment, LLMs produce contextually relevant and beneficial knowledge that enhances the recommendation process. We integrate the text-format knowledge generated by LLMs with the item IDs and feed them into ID-based recommenders. This fusion leverages the rich, real-world knowledge offered by LLMs while maintaining the ID-based sequential recommenders' proficiency in capturing sequential behavior patterns and collaborative signals. Additionally, it circumvents the inference latency problem by pregenerating LLM-derived knowledge prior to the recommendation phase, thus eliminating the need for the recommendation to wait for the time-consuming LLM generation.

Specifically, we first develop a prompt template to extract knowledge about items, which encompasses item descriptions and characteristics indicative of user preferences. The text-format augmented knowledge and the item IDs are both fed into ID-based sequential recommenders to generate recommendations. To further align LLMs with ID-based recommenders for the sequential recommendation task, we fine-tune LLMs using feedback from the ID-based recommenders' outputs and actual users' interacted items for better knowledge augmentation. Furthermore, we have developed an asynchronous technique to accelerate the aligning training of the integrated LLM and ID-based recommender, thereby improving training efficiency.

In summary, the major contributions of this paper are as follows:

- We propose SeRALM, a novel approach that combines Large Language Models (LLMs) with ID-based sequential recommenders by aligning LLMs with ID-based sequential recommenders. This integration leverages the complementary capabilities of LLMs for knowledge augmentation, enhancing the performance of existing ID-based sequential recommenders.
- We develop an alignment training method to guide LLM-generated text for improved knowledge augmentation, thereby minimizing noise information. SeRALM aligns LLMs with the sequential recommender using feedback from the ID-based recommender and the ground truth. A theoretically underpinned method is proposed , which not only addresses issues related to gradient propagation but also supports the separate deployment of LLMs and ID-based recommenders.
- We propose an asynchronous technique aimed at accelerating the alignment training of the LLM and the ID-based recommender, substantially enhancing training efficiency. During the inference stage, the knowledge from LLMs can be pre-generated to eliminate delays caused by the typically slow text generation of LLMs.
- Extensive experiments are conducted on public benchmarks, which validate SeRALM's compatibility with various state-of-the-art ID-based recommenders and its ability to boost their performances significantly. Additionally, a series of ablation studies and analyses confirm that SeRALM's alignment training method enables LLMs to generate more relevant and beneficial knowledge for sequential recommenders in settings with different training data amounts.

2 RELATED WORKS

2.1 Sequential Recommendation

Sequential recommendation is a crucial component of personalized recommender systems. Its objective is to capture users' preferences based on their historical behavior sequences. Initial approaches in this domain mainly focused on the pattern of Markov Chains [8, 19], which relies on item-to-item transition patterns to predict subsequent items based on prior selections. For instance, FPMC [20] combined Matrix Factorization (MF) to model user preferences with Markov Chains to detect and utilize sequential behavior patterns for prediction purposes.

The rise of deep learning has led to the development of numerous sequential recommendation models with deep neural networks. Caser [23] integrates a Convolutional Neural Network (CNN) to capture complex item sequence interactions, while GRU4Rec [9] employs a Recurrent Neural Network (RNN) for sequential pattern modeling. More recently, self-attention networks have demonstrated exceptional capabilities in sequential data modeling, leading to the development of several innovative models. SASRec [11] takes advantage of the Transformer architecture to attentively model sequential information in an unidirectional manner. BERT4Rec [21] enhances this approach by applying the Cloze objective, allowing for bidirectional context utilization through masked item prediction. S3-Rec [32] incorporates contrastive learning to fuse distinct item features, subsequences, and attributes. PAUP integrates self-attention with convolutions to capture both short- and long-term patterns. FEARec improves the original time domain self-attention in the frequency domain with a ramp structure to make both low-frequency and high-frequency information explicitly learned. LightSANs introduces a low-rank decomposed self-attention and decoupled position encoding, which scales linearly with respect to the sequence length and models the sequential relations between items more precisely.

These methodologies typically represent items using unique IDs and learn an ID embedding table through training data. Though they offer valuable insights into behavioral patterns in sequential interactions, they fall short in encompassing comprehensive, real-world item knowledge beyond datasets. This limitation hampers the overall effectiveness of ID-based recommendation models. In contrast, SeRALM augments ID-based models by integrating knowledge produced by large language models (LLMs) with traditional ID models.

2.2 LLMs for Sequential Recommendation

With the advent of generative LLMs such as GPTs, the focus of LLM-based recommendation has shifted towards generative recommendation strategies. These methods reframe recommendation tasks as natural language texts, employing LLMs as sequential recommenders to produce recommendation results directly [26]. At first, LLM-based methods rely on prompting [6, 22] or in-context learning [4, 15] to adapt LLMs for recommendation tasks. Nonetheless, these methods often underperformed when compared to traditional, task-specific recommendation models trained on specific data. Recent efforts have concentrated on more closely aligning LLMs with recommendation tasks through instruction tuning. The P5 framework [7] is a pioneer in this regard, offering a unified approach for integrating five recommendation tasks via fine-tuning on FLAN-T5 [18]. InstructRec [28] expanded upon this by adapting the FLAN-T5 model for various downstream recommendation tasks through instruction tuning with a broader range of text data. TALLRec [1] aligns the LLaMA model with binary recommendation tasks through two stages of instruction tuning in a few-shot scenario. Similarly, GenRec [10] directly applies instruction tuning to the LLaMA model using plain texts for the generative recommendation.

However, despite the advantages of harnessing real-world knowledge, these LLM-as-recommender methods struggle to handle the IDs and are unable to leverage sequential patterns and collaborative information contained in IDs effectively. In comparison, our proposed SeRALM method enhances the utility of LLMs by aligning them with the sequential recommendation task. It overcomes these limitations by extracting beneficial knowledge from LLMs and incorporating the extracted knowledge into ID-based recommenders. As a result, our integrated framework can fully leverage real-world knowledge, sequential behavior patterns, and collaboration information simultaneously.

3 PRELIMINARIES

In this section, we formulate the sequential recommendation task and introduce LLM's generation process based on prompts. Additionally, we establish the relevant notations for clarity and coherence in the subsequent discussion.

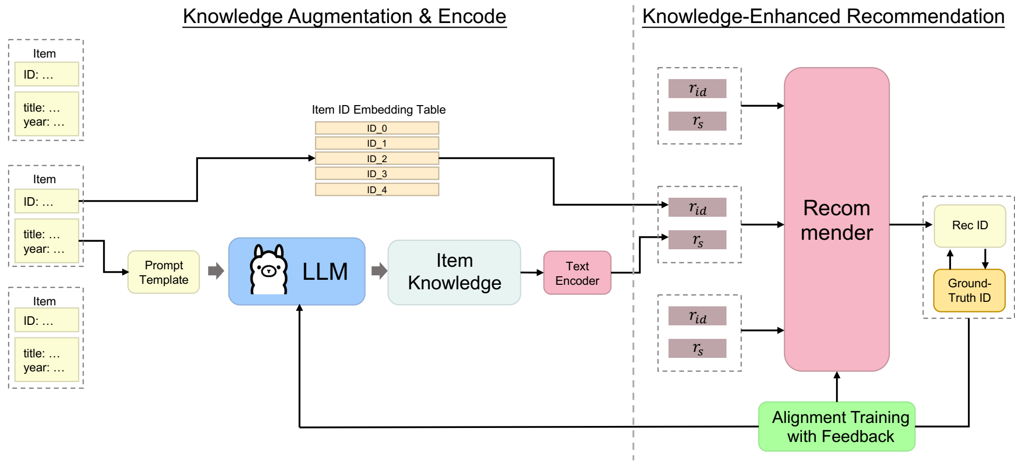


Figure 2: The overview of the SeRALM framework. The left part illustrates the Knowledge Augmentation and Encoding component, in which item knowledge is generated by the LLM, and both IDs and knowledge are encoded as vectors. The right part portrays the Knowledge-Enhanced Recommendation component. The alignment training component aligns the LLM and trains the recommender using feedback from the recommender output to improve knowledge augmentation. Once the entire system is trained, the Knowledge-Enhanced Recommendation component can be independently deployed online to produce real-time recommendations with low latency.

Sequential Recommendation. The sequential recommendation task is typically framed as a problem of predicting the next item in a sequence of data. Let us denote our dataset as $D = \{(x_1, y_1), \ldots, (x_i, y_i), \ldots, (x_n, y_n)\}$, where x_i signifies the sequence of items with which a user has previously interacted, and y_i indicates the subsequent item with which the user interacts after the sequence x_i. Specifically, a user's sequence of interacted items is represented as $x_i = [a_1, ..., a_i, ..., a_m]$, with a_i being the i-th item in that interaction sequence. The goal of sequential recommendation models is to develop a function $f(\cdot)$, parameterized by θ_r, that can accurately forecast the likelihood $P(y_i|x_i)$ that a user will interact with an item y_i, given their historical interaction sequence x_i.

LLM Generation Process. For a text prompt x related to items, a Large Language Model (LLM) produces a sequence of responses token-by-token in an autoregressive manner. Specifically, the model generates a new token t_k based on the context of previously generated tokens $t_c = (t_1, t_2, \ldots, t_{k-1})$. Let V represent the LLM's vocabulary. During the generation process, each potential token t_k in V receives a score $g(t_k|x, t_c)$ from the output layer of the LLM. The probability $p(t_k|x, t_c)$ of generating the token t_k from the LLM is then calculated using the softmax function:

$$p(t_k|x, t_c) = \frac{\exp g(t_k|x, t_c)}{\sum_{t'_k \in V} \exp g(t'_k|x, t_c)} \tag{1}$$

4 METHODOLOGY

We begin by presenting an overview of our proposed method, Enhancing Sequential Recommenders with Augmented Knowledge from Aligned Large Language Models (SeRALM). Subsequently, we delve into the specifics of each constituent element.

4.1 Overview

Our SeRALM framework is depicted in Figure 2. This model-agnostic system comprises three core components:

Knowledge Augmentation and Encoding: LLMs possess valuable real-world knowledge beneficial for the sequential recommendation task. We have devised a prompt-based technique to elicit user preference and item description knowledge from LLMs, which enriches the recommendation dataset with additional real-world insights. The acquired knowledge is then transformed into compact vectors using an efficient text encoder.

Knowledge-Enhanced Recommendation: The resultant knowledge vectors are integrated with ID-based recommendation systems, leveraging augmented knowledge from LLMs alongside traditional item IDs. This fusion allows the model to generate recommendations by analyzing sequential behavioral patterns, collaborative filtering signals, and the incorporated real-world knowledge.

Alignment Training: We introduce an alignment approach to refine the knowledge generated by LLMs, aiming to reduce irrelevant information. SeRALM aligns the LLM with the sequential recommendation task by employing feedback from the ID-based recommenders and actual user interactions. Our approach, grounded in theoretical principles, permits independent fine-tuning of LLMs and training of ID-based recommenders. This strategy not only circumvents gradient propagation issues but also supports the separate implementation of LLMs and ID-based recommenders — a prevalent industry practice.

4.2 Knowledge Augmentation and Encoding

As model size increases, LLMs increasingly encapsulate extensive real-world knowledge. This facilitates the enhancement of recommender systems through the integration of external knowledge. SeRALM harnesses LLMs to extract descriptions of items and infer

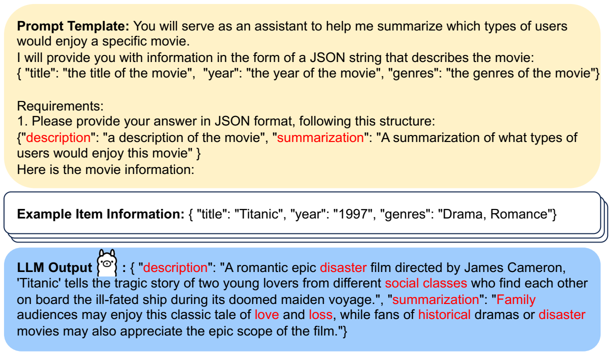


Figure 3: A Prompt example aimed at extracting descriptive knowledge and user preferences about items using LLMs. The prompt template is represented by the yellow bubble. By injecting specific information about an item shown in the white bubble into the prompt template, a prompt can be generated. This prompt then instructs the LLM to generate descriptive knowledge and deduce user preferences, as indicated in the blue bubble.

user preferences from interactions with specific items. We posit that an item sharing descriptions with one a user has interacted with previously is more likely to attract that user. On the other hand, two items with disparate descriptions may still appeal to users who share specific preferences, leading to their co-occurrence in a user's interaction history. For example, while a children's movie and a romance film may diverge in descriptions such as plot and cast, both could appeal to users with a preference for beautiful music. Thus, a user may watch both movies due to their musical appeal.

To extract descriptive and preference knowledge of items from LLMs, we design a prompt template shown in Figure 3. This prompt template instructs LLMs to generate knowledge beyond the information provided in the dataset. For instance, in Figure 3, the LLM deduces that "Titanic" appeals to "fans of historical dramas and disaster movies", which is not introduced in the original dataset.

Then, as shown in Figure 2, we encode the textual knowledge s generated by LLMs with a text encoder, such as BERT [12], to transform s into its corresponding knowledge representation r_s as follows:

$$r_s = \text{Encoder}(s) \tag{2}$$

where s is the textual knowledge generated by the LLM based on an item x.

4.3 Enhanced Recommendation with Augmented Knowledge

Once the knowledge representation r_s has been obtained, it can be incorporated into the ID-based recommender. This section explores a straightforward approach in which these knowledge representations are directly treated as additional input features. Specifically, they are used as additional feature fields and concatenated with the item ID embedding r_{id} in the ID-based recommender. During training, the text encoder is jointly optimized with the ID-based model.

Importantly, SeRALM solely alters the input fed into the ID-based recommender, maintaining independence from the recommender's architecture. This makes SeRALM flexible and compatible with various ID-based recommenders. By incorporating the knowledge, SeRALM combines both real-world knowledge and the ID-based methods' ability to capture sequential patterns and collaborate signals, providing more informed and personalized recommendations.

4.4 Alignment Training

As LLMs store a large amount of real-world knowledge, the generated knowledge contains diverse contents that can introduce noise harmful to the recommenders. To mitigate this issue, it is crucial to align the knowledge generation of LLMs with the sequential recommender to tailor the knowledge to the specific needs of the sequential recommendation task. In this section, we propose to align LLMs with recommenders and jointly train them.

In sequential recommendation, our objective is to construct a model through maximum likelihood estimation on training dataset D, aiming to maximize the objective function L:

$$L = \sum_{(x,y)\in D} \log p(y|x) \tag{3}$$

where x is an interacted item sequence and y is the next interacted item. SeRALM predicts the probability of the next interacted item y based on history item sequence x and LLMs-generated knowledge s, formalized as $p(y|x,s)$. $p(y|x,s)$ is the output of the recommender and is parameterized by the recommender parameters θ_r and the LLM parameters θ_l. Our goal is to optimize θ_r and θ_l as follows:

$$\theta_r, \theta_l = \arg\max_{\theta_r,\theta_l} \sum_{(x,y)\in D} \log p(y|x,s) \tag{4}$$

Although the recommender's parameter θ_r can be directly optimized through gradient ascent, the knowledge text s sampled from the LLM's output token distribution prevents direct propagation of gradients through s to the LLM's parameters θ_l. To address this, we model the distribution of sampling s from the LLM as $p(s|x)$, parameterized by θ_l. We consider s as a latent variable and marginalize over all possible knowledge texts S and compute $p(y|x)$ as follows:

$$p(y|x) = \sum_{s\in S} p(y|x,s)p(s|x) \tag{5}$$

Consequently, we can calculate $\nabla_{\theta_l} L$ as:

$$\begin{aligned}
\nabla_{\theta_l} \log p(y|x) &= \frac{1}{p(y|x)} \sum_{s\in S} p(y|x,s) \nabla_{\theta_l} p(s|x) \\
&= \frac{1}{p(y|x)} \sum_{s\in S} p(y|x,s) p(s|x) \nabla_{\theta_l} \log p(s|x) \\
&= \frac{1}{p(y|x)} E_{s\in p(s|x)} p(y|x,s) \nabla_{\theta_l} \log p(s|x) \\
&\approx \frac{1}{k} \sum_{i=1}^{k} \frac{p(y|x,s_k)}{p(y|x)} \nabla_{\theta_l} \log p(s_k|x)
\end{aligned} \tag{6}$$

Line 4 of Equation 6 is derived from line 3 through a Monte Carlo estimate, allowing us to approximate the gradient using a small subset of or simply one sampled knowledge texts s_k for each training instance (x,y). For simplicity, we compute $p(y|x)$ in $\nabla_{\theta_l} \log p(y|x)$

Algorithm 1 Alignment Training

Require: Training data D, the recommender Rec to be trained, the parameters θ_r of Rec, the pretrained ID-based recommender Rec_f whose parameters are freezed, the LLM to be aligned whose parameters is θ_l, the original LLM whose parameters are freezed

1: **while** not converged **do**
2: Sample $(x, y) \in D$
3: Sample one s from $p(s|x)$
4: Compute $p(y|x,s)$ with the recommender Rec, computed $p(y|x)$ with the pretrained recommender Rec_f
5: Use (x, s, y) to update θ_r with $\nabla_{\theta_r} p(y|x,s)$ as shown in Equation 9
6: Use $(x, s, p(y|x,s), p(y|x), p_{origin}(s|x))$ to update θ_l as shown in Equation 8
7: **end while**

with an ID-based recommender pretrained without augmented knowledge. Also, inspired by [17], we incorporate a per-token KL penalty from the original LLM to mitigate over-optimization of the aligned LLM as follows:

$$L_{KL} = \mathrm{E}_{s \in S} \left[\log p_{origin}(s|x) - \log p_{align}(s|x)\right] \quad (7)$$

where $p_{orig}(s|x)$ represents the probability of s of the original LLM without alignment, and $p_{align}(s|x)$ is the probability of s of the aligned LLM. The optimization of θ_l is then given by:

$$\theta_l = \theta_l + \nabla_{\theta_l} L + \alpha \nabla_{\theta_l} L_{KL} \quad (8)$$

Here, α denotes the KL penalty coefficient.

To summarize, for each sample (x, y), we sample a knowledge text s from the aligned LLM distribution $p(s|x)$. We then update the recommender's parameters θ_r as:

$$\theta_r = \theta_r + \nabla_{\theta_r} p(y|x,s) \quad (9)$$

Finally, we adjust the aligned LM's parameters θ_l as outlined above.

The whole algorithm is shown in Algorithm 1. demonstrating that we can optimize the LLM with $(s, p(y|x,s), p(y|x), p_{origin}(s|x))$ and optimize the recommender with (x, s, y). This distinct separation signifies that each model's optimization is dependent solely upon the output of the other, negating the necessity for gradient propagation between the models Consequently, this design enables independent deployment and optimization of the LLM and recommender system on disparate platforms, a common industrial requirement.

Asynchronous generation during training. The alignment training process can be divided into two jobs: (1) the *trainer* job, which updates parameters as indicated in Line 5 and Line 6 in Algorithm 1, and (2) the *generator* job, tasked with producing textual knowledge using the LLM and calculating $p(y|x,s)$, based on the recommender's output in Line 3 and Line 4 in the Algorithm 1. In the original algorithm, the *generator* job relies on the up-to-date θ_l and θ_r, constraining the *trainer* job to pause until both text generation and probability computation are complete — a factor that extends the duration of training. To expedite the alignment training, in the *generator* job, we propose to use old θ_l' and θ_r', and refresh them every few steps. As depicted in Figure 4, the *trainer* job

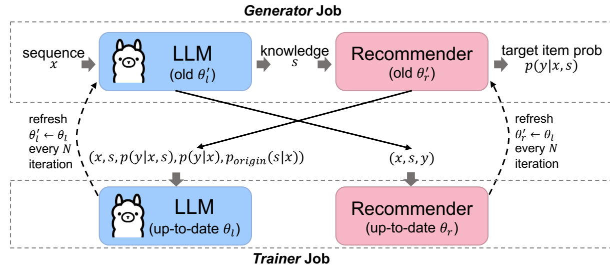


Figure 4: Asynchronous generation during alignment training.

sends the *generator* job a snapshot of θ_l and θ_r every few iterations, permitting uninterrupted training progress concurrently with the background activity of the *generator* job. This asynchronous approach allows both jobs to operate in parallel, thereby accelerating the alignment training workflow. However, it is critical to acknowledge that infrequent refreshes of the *generator* job's parameters might degrade performance due to potential distribution shifts in both the LLM and the recommender of the *generator* job.

Pre-generate and encode knowledge in the inference stage. After the alignment training, we suggest a preparatory step prior to inference—specifically, pre-generating the item knowledge s for each item x_i and pre-encoding its vector representation r_s. This measure allows for the reuse of knowledge across different users, circumventing the bottlenecks associated with the LLM's protracted generation time and the text encoder's slow processing speed. In practical industry settings, the aligned LLM can operate independently from the recommender. The LLM's sole responsibility would be to generate knowledge of recently added items on a daily basis, while the recommender handles real-time recommendation tasks. This division of labor aligns well with the immediacy demanded by online recommendation services.

What does the aligned LLM learn? Here, we examine how the alignment updating shown in Equation 8 rewards the generation of knowledge that enhances predictive accuracy

Let us consider a pair (x, y), and recall that $g(t_k|x, t_c)$ represents the score assigned to token t_k by the LLM given previously generated context t_c. To comprehend how a single step of gradient ascent in Equation 8 alters this score, we evaluate the gradient w.r.t. the LLM's parameters, θ_l:

$$\begin{aligned} \nabla_{\theta_l} \log p(y|x) &= \frac{1}{p(y|x)} \sum_{s \in S} p(y|x,s) \nabla_{\theta_l} \log p(s|x) p(s|x) \\ &= \frac{1}{p(y|x)} \sum_{s \in S} p(y|x,s) p(s|x) \nabla_{\theta_l} \sum_{k=1}^{n} \log p(t_k|x, t_c) \\ &= \sum_{k=1}^{n} \sum_{t_c} \sum_{t_k \in V} r(t_k) \nabla_{\theta_l} g(t_k|x, t_c) \end{aligned} \quad (10)$$

$$r(t_k) = p(t_c|y, x) p(t_k|t_c, x) \left[\frac{p(y|x, t_c, t_k)}{p(y|x, t_c)} - 1\right]$$

where n is the length of the generated text knowledge. For each token t_k, the gradient motivates the LLM to adjust the score $g(t_k|x, t_c)$ by $r(t_k)$. An increase is prompted if $r(t_k)$ is positive, or a decrease if it is negative. The factor $r(t_k)$ is a positive value if, and only if, $p(y|x, t_c, t_k) > p(y|x, t_c)$. Here, $p(y|x, t_c, t_k)$ signifies the probability of predicting the correct item y when augmented with knowledge token t_k given context t_c. $p(y|x, t_c)$ is the expected value of $p(y|x, t_c, t_k)$ over all potential tokens t_k. Therefore, a token t_k is positively reinforced whenever its contribution surpasses the average expectation. Overall, text knowledge s, as a sequence of tokens t_k, receives a positive update when it benefits the prediction.

Consequently, the output distribution of the LLM is refined to assign higher probabilities to content that is beneficial to the recommender. As a result, the LLM becomes increasingly likely to generate knowledge that benefits the recommender's predictions.

5 EXPERIMENTS

In this section, we will assess the performance of our proposed SeRALM on various public benchmarks and compare it with several baselines, including ID-based sequential recommenders and LLM-based sequential recommendation models. Additionally, we will conduct ablation studies, case studies, and few-shot analyses, as well as efficiency analyses to address the following questions:

- **RQ1:** How does SeRALM perform compared to conventional ID-based sequential recommenders and LLM-based sequential recommenders?
- **RQ2:** How do the alignment training and the asynchronous generation during training impact the performance of SeRALM?
- **RQ3:** How effective is SeRALM in the few-shot setting, where training data is limited?
- **RQ4:** How efficient is SeRALM in the inference stage?

5.1 Experimental Settings

5.1.1 Datasets and metrics. We experiment on three widely-used real-world datasets [5, 33], which vary significantly in domains and sparsity. (1) **MovieLens**: MovieLens is a site for recommending movies to users given their historical watched movie sequences. (2) **Amazon Books**: The online reviews and ratings of products on Amazon. We use the "Books" category in our experiments. (3) **Yelp**: A dataset for business recommendation. Following previous works [5, 11, 33], we apply the extensively used leave-one-out strategy for evaluation and employ Hit@K and NDCG@K to evaluate the performance. For each user, we rank the ground truth item in the test set with all other items of the dataset.

5.1.2 Baseline Methods. Our baselines can be divided into two types. (1) Conventional ID-based sequential recommenders. SeRALM can be integrated with these methods and enhances them with augmented knowledge from aligned LLMs: **Caser** [23] is a CNN-based method that applies horizontal and vertical convolutions for sequential recommendation; **SASRec** [11] is a unidirectional Transformer-based sequential recommendation model; **PAUP**[33] integrates self-attention with convolutions to capture both short- and long-term patterns; **LightSANs** [5] introduces a low-rank decomposed self-attention and decoupled position encoding, which scales linearly with respect to the sequence length and models the sequential relations between items more precisely. (2) LLM-based methods that follow the LLM-as-recommender paradigm and utilize LLMs as sequential recommenders to produce recommendations: **P5** [7] offers a unified approach for integrating five recommendation tasks via fine-tuning on FLAN-T5. **TALLRec** [1] aligns the LLaMA model with recommendation tasks through two stages of instruction tuning.

5.1.3 Implementation Details. We utilize Llama2-7b-chat as the LLM of SeRALM and implement TallRec with Llama2-7b-chat. Also, we refresh the parameters in the *generator* job of asynchronous generation during training every 50 training iterations, as we found such a refresh frequency rarely affects the model performances. We employ the implementation of Caser, SASRec, PAUP, and LightSANs in RecBole [30], an open-source recommender system library.

5.2 Main Results (RQ1-RQ2)

Comparisons with ID-based recommenders and Not-Aligned LLMs. We integrate SeRALM with various ID-based sequential recommenders to show the enhancement of augmented knowledge. Also, we assess the effectiveness of the aligned LLM by comparing it with a not-aligned LLM (denoted as Base+NA_LLM), which supplied knowledge to the ID-based recommender without alignment. The results, presented in Table 1, lead to the following observations:

(1) Integration of SeRALM with ID-based recommenders consistently resulted in performance gains over the original recommenders. This confirms SeRALM's ability to leverage LLMs for generating beneficial knowledge and to effectively incorporate this augmented knowledge with ID-based recommenders, thereby improving their performance without compromising their existing strengths.
(2) Although the not-aligned LLM provides some benefits to ID-based recommenders, SeRALM exhibited superior performance. This suggests that SeRALM enhances the alignment between LLMs and ID-based recommenders, enabling LLMs to produce more beneficial knowledge for these recommenders.
(3) On the Yelp dataset, the improvement with SeRALM is modest compared to other datasets. This can be attributed to the nature of Yelp's content, which consists of businesses— a domain where LLMs possess relatively limited knowledge compared to the movie and book domains in other datasets. Consequently, the impact of knowledge augmentation on Yelp is minimal.

Comparisons with LLM-based sequential recommenders. Furthermore, we evaluated several LLM-based sequential recommenders that instruction-tune LLMs and employ them to generate recommendations directly. We pick the best-performing SeRALM results from each dataset, as shown in Table 1, against these LLM-based methods. The comparative results are detailed in Table 2, and they consistently show SeRALM's superior performance. This can be attributed to SeRALM's unique capability to merge the strengths of ID-based methods—which capture sequential behavior patterns and collaborative signals—with the knowledge generated by aligned LLMs. These findings confirm SeRALM's efficacy in integrating sequential behavioral patterns with real-world knowledge in sequential recommendations.

5.3 Ablation Study (RQ2)

In this section, we investigate the efficacy of the alignment training and the impact of the parameter refresh frequency on asynchronous

Table 1: Performance improvement of ID-based sequential recommenders on public benchmarks achieved by integration with SeRALM (notated as Base+SeRALM) or with a non-aligned original LLM (notated as Base+NA_LLM). The highest-performing results are highlighted in boldface.

		MovieLens		Amazon-Books		Yelp	
ID-based Recommender	Enhancement Method	HIT@10	NDCG@10	HIT@10	NDCG@10	HIT@10	NDCG@10
PAUP	Base	22.08	11.22	9.14	4.59	4.41	2.22
	Base+NA_LLM	22.49(+1.87%)	11.45(+2.13%)	9.40 (2.90%)	4.69 (2.07%)	4.47 (+1.47%)	2.25 (+1.55%)
	Base+SeRALM	**22.75 (+3.03%)**	**11.52 (+2.63%)**	**9.52 (4.16%)**	**4.76 (+3.81%)**	**4.50 (+2.09%)**	**2.26 (+1.73%)**
LightSANs	Base	22.12	11.26	8.71	4.20	5.18	2.74
	Base+NA_LLM	22.60 (2.18%)	11.45 (1.67%)	8.89 (+2.10%)	4.31 (+2.57%)	5.25 (+1.35%)	2.78 (+1.29%)
	Base+SeRALM	**22.77 (+2.94%)**	**11.54 (2.50%)**	**9.07 (+4.14%)**	**4.37 (+4.11%)**	**5.28 (+1.85%)**	**2.80 (+2.08%)**
SASRec	Base	21.02	10.08	9.05	4.50	4.65	2.43
	Base+NA_LLM	21.43 (+1.96%)	10.26 (1.82%)	9.35 (+3.31%)	4.64 (+3.02%)	4.71 (+1.36%)	2.47 (+1.56%)
	Base+SeRALM	**21.66 (+3.03%)**	**10.37 (+2.91%)**	**9.40 (+3.90%)**	**4.68 (+4.09%)**	**4.74 (+1.96%)**	**2.48 (+1.94%)**
Caser	Base	16.63	8.08	5.37	2.65	3.17	1.57
	Base+NA_LLM	16.94 (+1.89%)	8.22 (+1.77%)	5.46 (+1.71%)	2.73 (+3.02%)	3.21 (+1.26%)	1.60 (+1.63%)
	Base+SeRALM	**17.03 (+2.45%)**	**8.29 (+2.61%)**	**5.59 (+4.10%)**	**2.76 (+4.15%)**	**3.23 (+1.79%)**	**1.61 (+2.00%)**

Table 2: Performance comparison of SeRALM with LLM-based recommenders.

	ML-1M		Books		Yelp	
Method	HIT@10	NDCG@10	HIT@10	NDCG@10	HIT@10	NDCG@10
TALLRec	21.32	10.17	9.08	4.39	4.68	2.44
P5	21.87	10.64	9.10	4.41	4.73	2.48
SeRALM	**22.77**	**11.54**	**9.52**	**4.76**	**5.28**	**2.80**

generation during training. By modifying key elements of SeRALM, we show the results in Figure 5.

Effectiveness of alignment training. We initially evaluate whether alignment training enhances both the LLM and the sequential recommender. This is achieved by reverting the aligned LLM and the recommender to their pre-alignment states, specifically to the not-aligned LLM and a recommender augmented by the not-aligned LLM's generated knowledge. The reversion, which affects both components, transforms SeRALM into a system that we denote as Base+NA_LLM in Table 1 and Section 5.2. Results in Figure 5 show that alignment training confers benefits to both the LLM and the recommender within SeRALM. Notably, optimal performance is achieved when both components operate collaboratively. This finding confirms that SeRALM's alignment training successfully aligns the LLM and the recommender, leading to enhanced performances.

Impact of asynchronous generation refresh frequency. As part of the alignment training, we employ a parallel *generator* job to generate knowledge, as described in Section 4.4. By default, we refresh the parameters in the *generator* job every 50 iterations. To ascertain the significance of frequent refreshes, we compare it with a reduced refresh rate of every 3000 iterations. Figure 5, which labels this condition as *slow_refresh*, illustrates that outdated

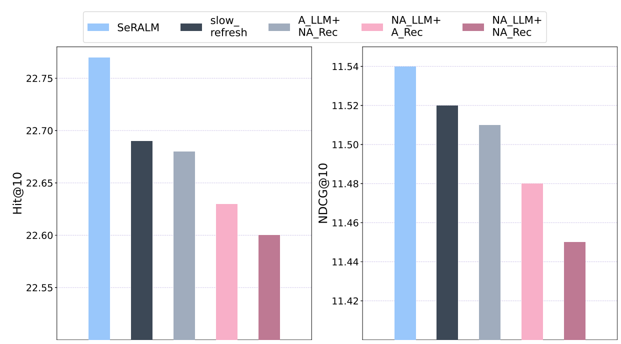


Figure 5: Ablation experiments on MovieLens. *A_LLM+NA_Rec* denotes resetting the recommender in SeRALM with a not-aligned recommender trained with knowledge generated by the not-aligned LLM. *NA_LLM+A_Rec* denotes resetting the aligned LLM in SeRALM with the not-aligned LLM. *NA_LLM+NA_Rec* denotes resetting both the aligned LLM and the recommender in SeRALM, which is denoted as *Base+NA_LLM* in Table 1 and Section 5.2. *slow_refresh* denotes refreshing parameters in the *generator* job 60x slower than the default setting.

parameters potentially undermine the alignment training process. This degradation occurs because of the distribution shift of the LLM and the recommender within the *generator* job when refreshes are infrequent. This comparison underscores the necessity of frequent parameter refreshes to maintain the efficacy of alignment training.

5.4 Case Study (RQ2)

The alignment training of SeRALM can align the LLM with the recommender to generate more beneficial knowledge. As depicted

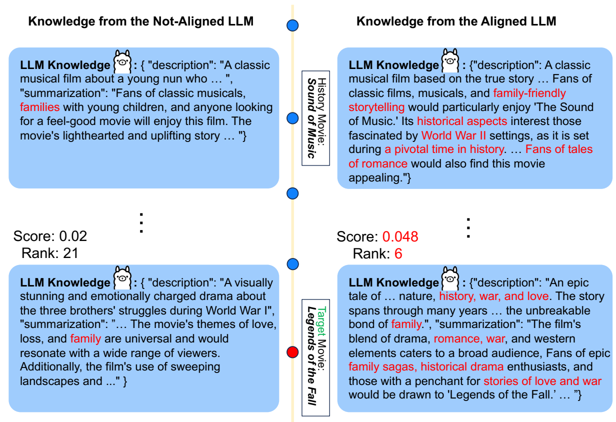


Figure 6: An example from MovieLens illustrating how SeRALM aligns the LLM to produce more beneficial knowledge. The left side shows the knowledge from the not-aligned LLM, while the right side displays the knowledge from the aligned LLM. Text highlighted in red indicates the overlapping knowledge shared by the target movie "Legends of the Fall" and a relevant historically watched movie "Sound of Music".

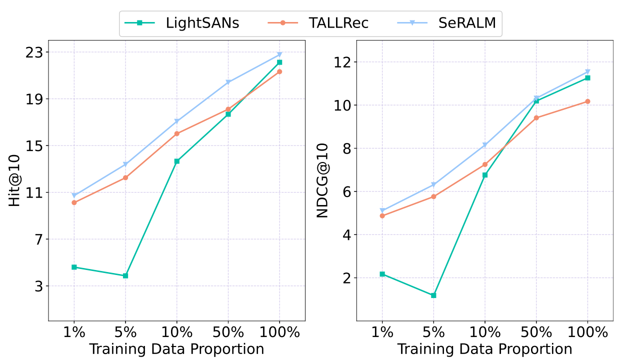


Figure 7: Low-resource performances of SeRALM, TallRec, and LightSANs on MovieLens with varying proportions of training data.

in Figure 6, we examine a case involving a user's historical movie watchlist. The movie "Legends of the Fall" is the actual subsequent movie watched, with the "Sound of Music" being a related movie in the viewing history. Employing SeRALM's alignment training enhances the LLM's output, yielding more similarity between the augmented knowledge of "Legends of the Fall" and "Sound of Music," which the not-aligned LLM fails to capture. SeRALM results in a notable improvement for the target movie "Legends of the Fall," with its relevance score increasing from 0.02 to 0.048. Consequently, this movie's ranking rises from 21st to 6th place among all candidate movies. These results underscore the efficacy of SeRALM in aligning LLMs to bolster the performance of sequential recommenders.

Table 3: Average Inference latency of different methods on MovieLens.

	TALLRec	LightSANs	SeRALM$_{rec}$	SeRALM$_{gen\&enc}$
Latency/s	4.64	2.81×10^{-3}	3.36×10^{-3}	2.57

5.5 Low-resource Analyses (RQ3)

In low-resource scenarios, we evaluated the performances of SeRALM, TallRec, and LightSANs by training each model on the MovieLens dataset with varying proportions of available training data. The results are depicted in Figure 7. It is evident that SeRALM consistently surpasses the other methods when training data is limited. This demonstrates the augmented knowledge's effectiveness from the aligned LLMs. Additionally, models incorporating LLMs outperform the ID-based approach, LightSANs, when training data is scarce. This observation can be attributed to the inadequate training of many item ID embeddings due to the lack of training data. Conversely, LLMs can provide supplementary knowledge about items, thereby improving the representations of items. When only 1% of the training data is utilized, the performance discrepancy between SeRALM and TallRec narrows, as the ID embeddings in the recommender of SeRALM become less effective with fewer training data, causing SeRALM to function similarly to a pure LLM-based method like TallRec.

5.6 Efficiency Analyses (RQ4)

This section presents the inference latency of SeRALM (w/ LightSANs), TALLRec, and ID-based LightSANs. For SeRALM, the knowledge generation and encoding can be precomputed as detailed in Section 4.4. We separately measure the latency of generating and encoding the knowledge of one item, denoted as SeRALM$_{gen\&en}$. The latency of SeRALM producing recommendations with precomputed knowledge vectors is denoted as SeRALM$_{rec}$. The evaluations are performed on a Tesla A100 (80GB) GPU, and results are summarized in Table 3. It is evident that the latency of LLM-based TALLRec is prohibitively long for industry scenarios. However, the computation-intensive SeRALM$_{gen\&en}$ is decoupled from the recommendation process, allowing SeRALM$_{rec}$ to achieve a response time comparable to that of ID-based LightSANs. This efficiency meets the real-time requirements of online recommender systems.

6 CONCLUSION

This paper introduces SeRALM, a novel framework that combines Large Language Models (LLMs) with ID-based sequential recommenders. SeRALM harnesses the real-world knowledge of LLMs to enhance ID-based sequential recommenders. SeRALM includes an alignment training method to align LLMs for improved knowledge augmentation while minimizing noise. Additionally, an asynchronous technique has been developed to expedite the alignment training. Extensive experiments on public benchmarks have validated SeRALM's compatibility with various ID-based recommenders, demonstrating its ability to significantly improve their performance. A series of analyses have confirmed that SeRALM's alignment training method enables LLMs to generate more beneficial knowledge for sequential recommenders across different training data volumes.

REFERENCES

[1] Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023. Tallrec: An effective and efficient tuning framework to align large language model with recommendation. *arXiv preprint arXiv:2305.00447* (2023).

[2] Edward Y Chang. 2023. Examining GPT-4: Capabilities, Implications and Future Directions. In *The 10th International Conference on Computational Science and Computational Intelligence*.

[3] Zeyu Cui, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. 2022. M6-rec: Generative pretrained language models are open-ended recommender systems. *arXiv preprint arXiv:2205.08084* (2022).

[4] Sunhao Dai, Ninglu Shao, Haiyuan Zhao, Weijie Yu, Zihua Si, Chen Xu, Zhongxiang Sun, Xiao Zhang, and Jun Xu. 2023. Uncovering ChatGPT's Capabilities in Recommender Systems. *arXiv preprint arXiv:2305.02182* (2023).

[5] Xinyan Fan, Zheng Liu, Jianxun Lian, Wayne Xin Zhao, Xing Xie, and Ji-Rong Wen. 2021. Lighter and better: low-rank decomposed self-attention networks for next-item recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*. 1733–1737.

[6] Yunfan Gao, Tao Sheng, Youlin Xiang, Yun Xiong, Haofen Wang, and Jiawei Zhang. 2023. Chat-rec: Towards interactive and explainable llms-augmented recommender system. *arXiv preprint arXiv:2303.14524* (2023).

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

[8] Ruining He and Julian McAuley. 2016. Fusing similarity models with markov chains for sparse sequential recommendation. In *2016 IEEE 16th international conference on data mining (ICDM)*. IEEE, 191–200.

[9] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).

[10] Jianchao Ji, Zelong Li, Shuyuan Xu, Wenyue Hua, Yingqiang Ge, Juntao Tan, and Yongfeng Zhang. 2023. Genrec: Large language model for generative recommendation. *arXiv e-prints* (2023), arXiv–2307.

[11] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*. IEEE, 197–206.

[12] 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*, Vol. 1. 2.

[13] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 1419–1428.

[14] Jiacheng Li, Ming Wang, Jin Li, Jinmiao Fu, Xin Shen, Jingbo Shang, and Julian McAuley. 2023. Text Is All You Need: Learning Language Representations for Sequential Recommendation. *arXiv preprint arXiv:2305.13731* (2023).

[15] Junling Liu, Chao Liu, Renjie Lv, Kang Zhou, and Yan Zhang. 2023. Is chatgpt a good recommender? a preliminary study. *arXiv preprint arXiv:2304.10149* (2023).

[16] Jie Lu, Dianshuang Wu, Mingsong Mao, Wei Wang, and Guangquan Zhang. 2015. Recommender system application developments: a survey. *Decision support systems* 74 (2015), 12–32.

[17] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems* 35 (2022), 27730–27744.

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

[19] Steffen Rendle. 2010. Factorization machines. In *2010 IEEE International conference on data mining*. IEEE, 995–1000.

[20] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*. 811–820.

[21] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 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.

[22] Weiwei Sun, Lingyong Yan, Xinyu Ma, Pengjie Ren, Dawei Yin, and Zhaochun Ren. 2023. Is ChatGPT Good at Search? Investigating Large Language Models as Re-Ranking Agent. *arXiv preprint arXiv:2304.09542* (2023).

[23] Jiaxi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the eleventh ACM international conference on web search and data mining*. 565–573.

[24] 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* (2023).

[25] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems* 35 (2022), 24824–24837.

[26] Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, et al. 2023. A Survey on Large Language Models for Recommendation. *arXiv preprint arXiv:2305.19860* (2023).

[27] Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M Jose, and Xiangnan He. 2019. A simple convolutional generative network for next item recommendation. In *Proceedings of the twelfth ACM international conference on web search and data mining*. 582–590.

[28] Junjie Zhang, Ruobing Xie, Yupeng Hou, Wayne Xin Zhao, Leyu Lin, and Ji-Rong Wen. 2023. Recommendation as instruction following: A large language model empowered recommendation approach. *arXiv preprint arXiv:2305.07001* (2023).

[29] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM computing surveys (CSUR)* 52, 1 (2019), 1–38.

[30] Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Yushuo Chen, Xingyu Pan, Kaiyuan Li, Yujie Lu, Hui Wang, Changxin Tian, et al. 2021. Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms. In *proceedings of the 30th acm international conference on information & knowledge management*. 4653–4664.

[31] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223* (2023).

[32] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM international conference on information & knowledge management*. 1893–1902.

[33] Yuehua Zhu, Bo Huang, Shaohua Jiang, Muli Yang, Yanhua Yang, and Wenliang Zhong. 2022. Progressive Self-Attention Network with Unsymmetrical Positional Encoding for Sequential Recommendation. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*. 2029–2033.