LLM2Rec: Large Language Models Are Powerful Embedding Models for Sequential Recommendation

Yingzhi He*
heyingzhi@u.nus.edu
National University of Singapore
Singapore, Singapore

Xiaohao Liu*
xiaohao.liu@u.nus.edu
National University of Singapore
Singapore, Singapore

An Zhang†
an.zhang3.14@gmail.com
University of Science and Technology of China
Hefei, China

Yunshan Ma
ysma@smu.edu.sg
Singapore Management University
Singapore, Singapore

Tat-Seng Chua
dcscts@nus.edu.sg
National University of Singapore
Singapore, Singapore

Abstract

Sequential recommendation aims to predict users' future interactions by modeling collaborative filtering (CF) signals from historical behaviors of similar users or items. Traditional sequential recommenders predominantly rely on ID-based embeddings, which capture CF signals through high-order co-occurrence patterns. However, these embeddings depend solely on past interactions, lacking transferable knowledge to generalize to unseen domains. Recent advances in large language models (LLMs) have motivated text-based recommendation approaches that derive item representations from textual descriptions. While these methods enhance generalization, they fail to encode CF signals—*i.e.,* latent item correlations and preference patterns—crucial for effective recommendation. We argue that an ideal embedding model should seamlessly integrate CF signals with rich semantic representations to improve both in-domain and out-of-domain recommendation performance.

To this end, we propose **LLM2Rec**, a novel embedding model tailored for sequential recommendation, integrating the rich semantic understanding of LLMs with CF awareness. Our approach follows a two-stage training framework: (1) Collaborative Supervised Fine-tuning, which adapts LLMs to infer item relationships based on historical interactions, and (2) Item-level Embedding Modeling, which refines these specialized LLMs into structured item embedding models that encode both semantic and collaborative information. Extensive experiments on real-world datasets demonstrate that LLM2Rec effectively improves recommendation quality across both in-domain and out-of-domain settings. Our findings highlight the potential of leveraging LLMs to build more robust, generalizable embedding models for sequential recommendation. Our codes are available at https://github.com/HappyPointer/LLM2Rec.

CCS Concepts

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

Keywords

Sequential Recommendation, Large Language Models, Embedding Models

ACM Reference Format:
Yingzhi He, Xiaohao Liu, An Zhang, Yunshan Ma, and Tat-Seng Chua. 2025. LLM2Rec: Large Language Models Are Powerful Embedding Models for Sequential Recommendation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '25), August 3–7, 2025, Toronto, ON, Canada.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3711896.3737029

1 Introduction

Sequential recommendation aims to predict users' future interactions by learning high-quality item representations that effectively capture both user preference patterns and item inherent content [41, 60]. Conventional sequential recommenders typically assign unique identifiers (IDs) to items and learn corresponding representations based on historical user interaction sequences [12, 15, 43, 44, 57, 64]. These ID-based representations primarily encode collaborative filtering (CF) signals by solely modeling multi-hop co-occurring patterns in sequential trajectories [10, 51]. While effective, such recommenders lack item content information, making them highly domain-dependent and unable to generalize to unseen items or new domains [59, 60, 65]. We argue that high-quality item representations in sequential recommender systems must simultaneously encapsulate item semantics and CF signals.

Recent advances in large language models (LLMs) have motivated extensive research into leveraging rich semantic information for improved item representation learning, including purely text-based representations [14, 22, 41] and hybrid representations that fusing semantic and CF signals [13, 30, 40]. Purely text-based recommenders extract item representations from pre-trained language models, offering strong generalization capabilities but disregarding CF signals. To mitigate this limitation, hybrid methods attempt

*Equal Contribution.
†Corresponding Author.

KDD '25, Toronto, ON, Canada

ACM ISBN 979-8-4007-1454-2/2025/08
https://doi.org/10.1145/3711896.3737029

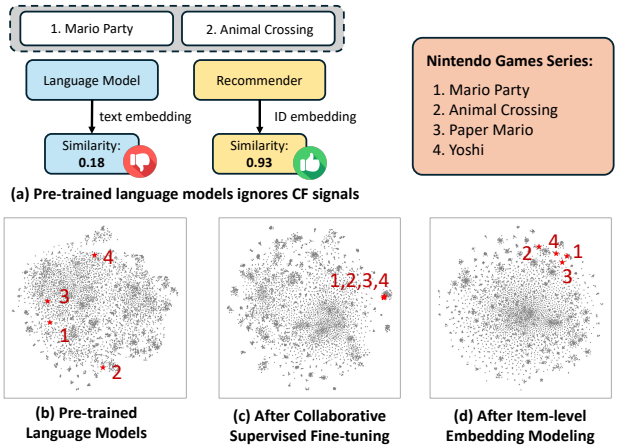

Figure 1: An illustrating example that pre-trained language models ignore CF signals and the embedding distributions in different training stages of LLM2Rec. (a) The behaviorally correlated Mario Party and Animal Crossing are close in ID embeddings but distinct in text embeddings. (b) Pre-trained language models failed to discover the behaviorally similar items. (c) Collaborative Supervised Fine-Tuning enables the LLM to capture the CF signals. (d) Item-level embedding modeling preserves CF signals while producing more distinguishable embeddings.

to integrate both semantic and CF signals through various fusion strategies, including directly concatenating ID-based and semantic representations [45, 54, 60], guiding ID-based representation learning with content features [40, 53], adopting hybrid fusion architectures [29, 30], and tuning embedding models to bridge CF and semantic spaces [13, 39]. However, fusion-based techniques, whether simple concatenation or sophisticated mechanisms like cross-attention, struggle to learn a unified representation space, leading to misalignment between item semantics and user behavior spaces. We believe that developing a general embedding model for sequential recommendation-one that inherently captures both CF and semantic knowledge-is a more promising yet underexplored research direction. While recent efforts have attempted to construct unified embedding models that align item semantics with user behavior spaces through contrastive pre-training [13, 39], these methods typically require large-scale training samples and prohibitively large batch sizes to effectively encode CF signals. More critically, they fail to fully leverage the strong semantic understanding and reasoning capabilities of state-of-the-art LLMs, such as Qwen [58] and Llama [46]. This motivates us to investigate how LLMs can be adapted to serve as generalizable embedding models for recommendation.

To develop a recommendation-specialized embedding model with strong generalization to unseen domains, we aim to integrate the powerful semantic understanding capability of LLMs with the ability to capture CF signals. Motivated by recent studies demonstrating that LLMs can effectively learn recommendation tasks through supervised fine-tuning [1, 2, 6, 25, 28, 62], we leverage this approach to make LLMs aware of the CF signals from user interaction sequences. To further facilitate the transition of the LLM from token-level prediction to item-level embedding generation [3, 18], we refine the CF-aware LLM into a structured recommendation embedding model with additional item-level embedding modeling.

To this end, we introduce LLM2Rec, a recommendation embedding model built upon the LLM that is explicitly aware of the CF signals. Specifically, our training framework includes two stages: (1) Collaborative Supervised Fine-Tuning (CSFT) and (2) Item-level Embedding Modeling (IEM). In the first stage, CSFT fine-tunes the LLM on a mixture of six real-world recommendation datasets, enforcing it to predict the next item based on the historical interaction sequence. As illustrated in Figure 1, the embeddings of several games in the Nintendo series are initially scattered. As these games are frequently co-purchased by users with similar preferences, their embeddings become closely clustered after CSFT. This shift indicates that the LLM learns to capture CF signals through CSFT. In the second stage, we enable bidirectional attention with Masked Next Token Prediction (MNTP) and apply item-level contrastive learning to further facilitate the LLM to be an embedding model. Bidirectional attention enables capturing contextualized information within item titles [3] and MNTP helps the LLM adapt to the newly introduced bidirectional attention mask. Item-level contrastive learning explicitly shifts the pre-training objective from token-level to item-level, helping to generate distinguishable item embeddings and yet preserve the CF signals. Both MNTP and item-level contrastive learning are lightweight adaptations incurring slight computational costs while remaining effective. As presented in Figure 1, the embeddings of Nintendo games remain close while becoming more differentiated, offering richer and more effective information for recommendation.

To assess the effectiveness of LLM2Rec, we conduct extensive experiments on both in-domain and out-of-domain datasets using various downstream sequential recommenders. Experimental results show that LLM2Rec consistently outperforms the existing embedding models across both in-domain and out-of-domain datasets. Additionally, the generalization ability of the embedding model benefits from training on mixed datasets spanning diverse categories. These findings underscore the potential of LLMs as inherently powerful embedding models for sequential recommendation.

2 Related Work

In this section, we briefly review the works related to this paper from two main categories: 1) sequential recommendation, and 2) embedding models.

2.1 Sequential Recommendation

Sequential recommendation learns item representations to predict items that users are likely to interact with in the future. From the item representation learning perspective, existing methods can be broadly categorized into three paradigms: ID-based methods, pure text-based methods, and hybrid ID-text methods.

ID-based methods assign each item a unique identifier and learn the corresponding representation using various sequence modeling techniques, like recurrent neural networks [12], convolutional neural networks [44], and transformer-based architectures [15, 64]. With these techniques, ID-based methods capture item correlations and user interests from user interaction sequences. Despite the

effectiveness, these methods are not capable of handling tasks from unseen domains or unseen items, lacking generalizability.

Pure text-based methods represent items with text embeddings derived from item contents, such as titles or profiles, using pre-trained language models as text encoders. Within the unified language space, these methods have the potential to generalize to unseen domains [14, 22, 41]. However, their effectiveness is largely limited by the adopted embedding model due to their heavy reliance on text embeddings. Moreover, these pre-trained language models are general for language tasks rather than specialized for recommendation, resulting in suboptimal performance as well.

Hybrid ID-text methods generate representation by incorporating both ID and textual information. Common techniques include (1) using text embeddings to guide or enhance the learning of ID representations [27, 29, 40, 52, 53, 56], (2) concatenating text and ID embeddings [31, 45, 54, 60], and (3) fusing ID and text information through attention architectures [30]. While these methods leverage textual information to improve representation learning, they still rely on ID-based embeddings, indicating that their effectiveness remains highly dependent on dataset-specific training. As a result, similar to purely ID-based methods, hybrid approaches must be trained on the target domain to learn effective representations, limiting their ability to generalize to new domains [59, 60, 65].

2.2 Embedding Models

Pre-trained embedding models play a fundamental role in various downstream tasks, including information retrieval [16, 26, 34], text similarity [4], and classification [7, 11]. Existing approaches can be broadly categorized into two main types: 1) Language encoder models with bidirectional attention and 2) LLM-based embedding models. Language encoder models have long been the dominant approach for learning text embeddings. These models leverage transformer architectures with bidirectional attention, allowing them to capture richer contextual relationships and produce more effective sentence embeddings [17, 20, 32]. Their training objectives include next sentence prediction [17], masked language modeling [17, 32], and contrastive learning techniques [8, 23, 38, 49], which has gained significant popularity for its effectiveness in producing high-quality sentence embeddings. LLM-based embedding methods have gained increasing popularity with the growing capabilities of large language models (LLMs). The straightforward approaches generate sentence embeddings directly from the last hidden states of decoder-only LLMs, either by using the hidden state of the EOS token or by average pooling the hidden states of all tokens in the sentence [35, 42, 50]. However, directly using pre-trained LLMs as embedding models without additional training leads to suboptimal performance, as they are optimized for predicting future tokens rather than generating holistic sentence representations. To address this limitation, recent approaches focus on adapting LLMs into dedicated embedding models with further adaption [3, 18, 19, 21, 24, 36]. Common techniques include sentence-level contrastive learning and attention mask modifications, both of which enhance LLMs' effectiveness as embedding models.

In recommendation tasks, the implicit relationships between items, known as CF signals, are also important alongside semantic understanding. An effective embedding model for recommendation should integrate both to maximize performance. While most sequential recommenders rely on general embedding models, some initial efforts attempt to develop recommendation-specific embeddings. Blair [13] aligns representations of user reviews with item titles using contrastive learning on over 30 million instances across 33 categories from the Amazon platform. EasyRec [39] aligns user representations with their interacted items and further incorporates diverse user and item profiling in contrastive learning. However, both methods constrain their backbone embedding model to smaller ones like BERT [17] or RoBERTa [32] due to the high computational cost of contrastive learning, which requires large batch sizes and sufficient training iterations. Recently, LLMEmb [29] extends recommendation-specific embedding models by leveraging large language models. It adopts attribute-level augmentations and aligns augmented views of the same item to enhance the generated embeddings. LLMEmb primarily treats LLMs as powerful semantic encoders and does not explicitly integrate CF signals into the learned embeddings. In contrast, our approach fuses CF signals into LLMs through collaborative supervised fine-tuning. With the following enhancements of lightweight item-level embedding adaptation, LLM2Rec achieves superior recommendation-specific embeddings while maintaining computational efficiency.

3 Methodology

In this section, we first elaborate on the problem formulation of utilizing embedding models for sequential recommendation tasks. After that, we introduce the crux of LLM2Rec, consisting of collaborative supervised fine-tuning and item-level embedding modeling. At the end, we elaborate the optimization of LLM2Rec, followed by its utilization for downstream sequential recommenders. The overall framework is depicted in Figure 2.

3.1 Problem Formulation

3.1.1 Sequential Recommendation. We have a set of items $\mathcal{I}$, and user interaction sequence X, constructing the dataset $\mathcal{D} = \{\mathcal{I}, X\}$. Wherein, each element denotes an item sequence $\{i_1, i_2, \ldots, i_{N_u}\}$; N_u is the number of the interaction items for a specific user u. In our setting, these items are typically represented by their titles. The goal of sequential recommendation is to predict the next item i_{N_u+1} given the previous interactions $i_{<N_u}$. Here we hope to obtain a sequential recommender $R(i_{<N_u})$ that is capable to capture the user intention hidden within the previous interaction and finally inferring the preferred next item. Formally, we denote it as $p(i_{N_u+1}|i_{<N_u}) = R(i_{<N_u})$.

3.1.2 Recommendation Embedding Modeling. In this work, we focus on elevating the sequential recommenders via adapting a well-trained embedding model $\mathcal{E}(\cdot)$. Given training datasets $\mathcal{D}^{\text{train}} = \{\mathcal{D}_k\}_{k=0}^{n}$, the embedding model is capable of generating effective embedding for testing datasets (*i.e.*, out-of-domain datasets) $\mathcal{D}^{\text{test}} = \{\mathcal{D}_{n+j}\}_{j=0}^{m}$. $\mathcal{E}(\cdot)$ can accept the textual description of item sequence, *i.e.*, $\{\mathbf{t}_i := [t_1^i, t_2^i, \ldots, t_{\ell_i}^i]\}_{<N_u}$), or a single item; ℓ_i the token length of item i. Accordingly, leveraging such an embedding model, we can obtain item embedding via $\mathbf{z}_i = \mathcal{E}(\mathbf{t}_i)$. Note that, in text-based recommendation, the above methods typically adopt a

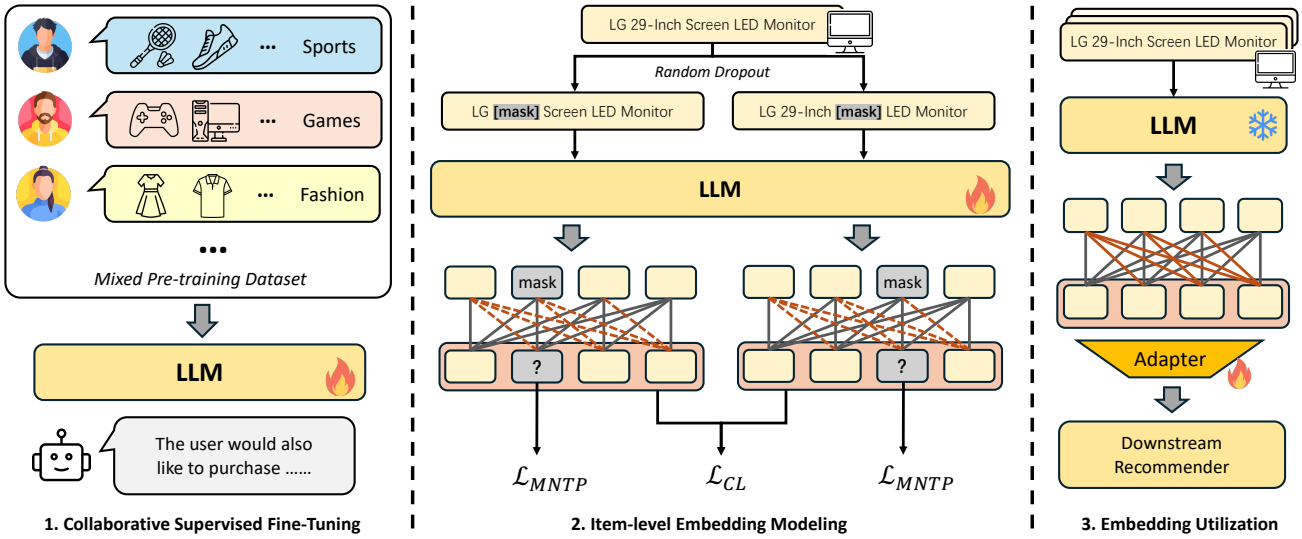

Figure 2: Illustration of the overall pre-training framework of LLM2Rec and how the generated embeddings are utilized for downstream sequential recommenders. LLM2Rec employs a two-stage training strategy: first, adapting LLMs to infer item relationships from previous interactions, namely, collaborative instruction fine-tuning (left); second, reforming specialized LLMs for item-level embeddings with two training objectives (middle). By encoding both semantic and CF information, generated embeddings bolster the exiting recommenders via a lightweight adapter (right).

pre-trained text encoder $\mathcal{E}^*(\cdot)$ to extract the semantics in a latent space. Thus, we can reformulate the sequential recommendation by explicitly involving the embedding modeling:

$$p(i_{N_u+1}|i_{<N_u}) = R(\mathbf{z}_{i<N_u}), \forall \mathbf{z}_i = \mathcal{E}(\mathbf{i}). \tag{1}$$

3.2 Collaborative Supervised Fine-tuning

We adapt LLM for sequential recommendation tasks via supervised fine-tuning with collaborative information, *i.e.,* user-item interactions. First, we construct the recommendation instructions that take the user previous interactions as input, and set the next item as the label. This formulation follows the LLM's inherent autoregressive generation manner; and our goal is to enable LLMs to perform recommendations with collaborative instructions.

Input:
Logitech G13 Gameboard,
Tamron 70-200mm Camera Lens (Nikon) ,
Pelican SD Card Case,
YONGNUO Flash Trigger (Canon) ,
TAKSTAR SGC-598 Microphone,
STK EN-EL3e Charger for Nikon Camera ,
VGA to HDMI Cable,
Allstate 2-Year Protection Plan,
BLACKRAPID Lock Star Cover.

Output: Neewer Wireless Flash Trigger for Camera

Figure 3: An example of the collaborative instruction for fine-tuning.

As aforementioned, we represent items with textual information (*e.g.*, titles), denoted as $\mathbf{t}_i := [t_1^i, t_2^i, \ldots, t_{\ell_i}^i]$, where t is the token indices; the user interactions can be represented as the concatenated item sequence, $\mathbf{t}_u := [\mathbf{i}_1, \mathbf{i}_2, \ldots, \mathbf{i}_{N_u}]$. As shown in Figure 3, the input instruction is a series of items (*i.e.*, movie titles), followed by the next item title as the desired output. The CF signals manifest by these highlighted item correlations (the darker the color, the more similar with the target items). To mitigate the influence of the template, which introduces hidden states irrelevant to the current item representation, we only retrain the item titles with some necessary separators, like commas. With such type of collaborative instructions, we employ the objective to predict the tokens within the next item autoregressively:

$$\mathcal{L}_{\text{CSFT}} = -\sum_{(u,i)\in X}\sum_{s=0}^{\ell_i} p(\mathbf{t}_{i,s}|\mathbf{t}_u, \mathbf{t}_{i,<s}), \tag{2}$$

where s denotes the current step. This step is simple yet essential to unlock the capability of LLM for recommendation and bolsters the following embedding modeling.

3.3 Item-level Embedding Modeling

Decoder-only LLMs are designed for autoregressive prediction, making them less effective at generating high-quality embeddings, a task typically suited for encoder models. However, LLMs exhibit stronger semantic understanding ability compared to traditional bidirectional encoder methods [3, 18]. Some works [5, 61, 63] propose utilizing the last hidden state of the token to represent the given text, serving as an alternative to use LLMs. However, we hope to reap both benefits of decoder-only LLM's knowledge and encoder models' architecture, thus reforming LLMs for item-level embedding modeling.

3.3.1 Reforming Decoder-only LLM to Encoder. The key differences that distinguish the encoder and decoder language models lie in

causal attention and *training objective*. Inspired by remarkable language encoders [17, 21] and recent attempts on utilizing LLM as general embedding models [3, 24], we equip the fine-tuned LLMs with *bidirectional attention* and *masked next token prediction optimization* (MNTP).

Causal → Bidirectional attention. Causal attention restricts access to information from later tokens when generating token embeddings. While this is essential for prediction tasks, embedding models require visibility of both preceding and succeeding tokens to capture comprehensive contextual details. Motivated by this, we cancel the causal attention mask, enabling prediction based on the item-level past and future context (*i.e.*, bidirectionally). To adapt model parameters for the newly introduced architecture, we impose an additional training stage with masked next token prediction (MNTP). Given an item sequence as input, we randomly mask tokens with a pre-defined fraction, then train LLMs with masked next token prediction task:

$$\mathcal{L}_{\text{MNTP}} = -\sum_{i \in \mathcal{I}} \sum_{s=0}^{\ell_i} p(\mathbf{t}_{i,s} | \mathbf{t}_{i,<s}). \tag{3}$$

The parameters of LLM that are trained with sequential recommendation tasks in causal attention, followed by MNTP tasks in bidirectional attention. Notably, this stage focuses solely on the information within a single item, aligning with the tuning of LLMs for item-level embeddings. In contrast, CSFT captures relationships between different items within user interaction sequences.

3.3.2 Item-level Contrastive Learning. Token-level embedding modeling is developed with architecture modification and MNTP optimization; however, we hope to generate item-level embedding, which is more prevailing and intuitive for downstream recommenders. The straightforward solution can be a direct average-pooling of token-level embeddings. Specifically, for item i, the item embeddings can be represented as $\mathbf{z}_i = 1/\ell_i \sum_{j \in [\ell]} \mathbf{z}_i^j$. Hence, the embedding model can be the composition of such average pooling operation and our modified LLM model π_θ, and formally denoted as $\mathcal{E} := \text{avg} \circ \pi_\theta$. In this work, we further enhance this by employing item-level contrastive learning.

Token → Item level embedding. The input item $\mathbf{t}_i$ is passed through LLM model twice with random masking independently, yielding two views of the same item (*i.e.*, $\tilde{\mathbf{t}}_\mathbf{i}^\mathbf{1}$ and $\tilde{\mathbf{t}}_\mathbf{i}^\mathbf{2}$). Following the unsupervised contrastive learning paradigm [8], we optimize the parameters via:

$$\mathcal{L}_{\text{IC}} = -\sum_{i \in \mathcal{I}} \log \frac{\left(\mathcal{E}(\tilde{\mathbf{t}}_\mathbf{i}^\mathbf{1}) \mathcal{E}(\tilde{\mathbf{t}}_\mathbf{i}^\mathbf{2})^\top / \tau\right)}{\sum_{j \in \mathcal{I}} \left(\mathcal{E}(\tilde{\mathbf{t}}_\mathbf{j}^\mathbf{1}) \mathcal{E}(\tilde{\mathbf{t}}_\mathbf{j}^\mathbf{2})^\top / \tau\right)}, \tag{4}$$

where τ is the temperature ratio. With this objective, item embeddings are learned via a holistic view and contrasted with others to enhance their distinctiveness. The differentiation induced by contrastive learning aligns well with a series of recommendation methods [55, 57], providing a stronger foundation for downstream sequential recommenders.

3.4 Optimization & Utilization

Training LLM2Rec. We train the LLMs in a sequential manner, from collaborative supervised fine-tuning ($\mathcal{L}_{\text{CSFT}}$) to item-level embedding modeling ($\mathcal{L}_{\text{MNTP}}$ and $\mathcal{L}_{\text{IC}}$). The model architecture is modified in the second stage, reforming causal attention to the bidirectional. This training strategy progressively enhances the LLMs' capabilities, featuring them to capture both semantic and CF information for recommendation.

Empowering downstream recommenders. After the training of LLM2Rec, we introduce the utilization of embeddings generated from LLM2Rec to bolster the existing sequential recommenders. Most sequential recommenders obtain their item embeddings from scratch or initialize them as trainable parameters updated along with the training [12, 15]. In this work, we provide a simple solution via an additional linear adapter: $\mathbf{z}'_i = \mathbf{W}\mathbf{z}_i + \mathbf{b}$, where $\mathbf{W}$ and $\mathbf{b}$ are the weight and bias matrices. We use these transformed embeddings as item representations. Notably, the parameters is optimized through downstream recommenders' objective for adaption. By inducing slight parameters, this linear transformation is capable of adapting our generalizable embeddings for various domains.

4 Experiments

In this section, we present the experimental results and corresponding analysis to answer the following research questions (**RQ**s).

- **RQ1:** How effective is our proposed LLM2Rec compared with other embedding models, including general and specialized ones?
- **RQ2:** How does each training stage or modification contribute to the performance of LLM2Rec?
- **RQ3:** What are the key properties of LLM2Rec?

4.1 Experiment Settings

We systematically present the details of datasets, evaluation metrics, the baselines embedding models for comparison, downstream sequential recommenders used for evaluation, as well as the implementation details.

4.1.1 Datasets and Evaluation Metrics. We elaborate on the selected datasets for pre-training embedding model and downstream sequential recommendation tasks, followed by descriptions of the evaluation metrics.

Pre-training datasets. Following prior works [13, 39], our embedding model is pre-trained on a mixture of six datasets collected from the Amazon platform [13]. These datasets span diverse categories, including *Video Games* (**Games**), *Arts, Crafts, and Sewing* (**Arts**), *Movies and TV* (**Movies**), *Home and Kitchen* (**Home**), *Electronics* (**Electronics**), and *Tools and Home Improvement* (**Tools**). The datasets consist of user interactions spanning from June 1996 to September 2023[1]. For all six training datasets, we apply 5-core filtering and limit the maximum historical interaction sequence length to 10. The datasets are partitioned into training, validation, and test sets using the leave-one-out strategy, where the last two interactions in each user sequence are reserved for validation and testing, respectively. Only the training data is used for pre-training LLM2Rec, while the validation and test sets are reserved for downstream evaluation. Detailed statistics of each pre-training dataset are listed in Table 1.

[1] https://amazon-reviews-2023.github.io/

Table 1: The statistics of the pre-training datasets.

Dataset	#Items	#Interactions
Games	9,517	153,221
Arts	12,454	132,566
Movies	13,190	136,471
Home	33,478	256,001
Electronics	20,150	197,984
Tools	19,964	159,969
Total	108,753	1,035,212

Table 2: The statistics of the in-domain and out-of-domain datasets used in downstream sequential recommendation.

Dataset	#Items	#Interactions	Train	Val&Test
Games	9,517	153,221	122,577	15,322
Arts	12,454	132,566	106,052	13,257
Movies	13,190	136,471	109,177	13,647
Sports	13,952	136,740	109,392	13,674
Baby	6,837	97,899	78,319	9,790
Goodreads	4,550	158,347	137,069	10,639

Downstream sequential recommendation datasets. For downstream sequential recommenders utilizing the generated embeddings, we train and test the downstream recommenders on the same datasets with identical data splits as used in pre-training. Specifically, we present experimental results on three datasets: **Games**, **Arts**, and **Movies**. To further evaluate the generalization ability of LLM2Rec to unseen domains, we additionally include three more out-of-domain datasets that differ significantly from the pre-training data. Specifically, we select **Sports**, **Baby** [13] from Amazon, which contain items from categories absent in the training set. We further evaluate our embedding model on cross-platform dataset: **Goodreads**[2] [47, 48]. Detailed statistics of downstream sequential recommendation datasets are presented in Table 2.

Evaluation Mertrics. We follow the prior works [9, 13, 39] and perform full ranking with all items in the dataset as potential candidates during evaluation. The performance of the recommenders is evaluated with the Recall@k and NDCG@k, where $k \in \{10, 20\}$. To ensure the reliability of experimental results and mitigate the impact of unavoidable randomness during downstream recommender training, all reported performances in this section are averaged over three runs with different random seeds.

4.1.2 Baselines and Downstream Recommenders. For baseline embedding models, we compare LLM2Rec with a diverse set of models, including both general-purpose and recommendation-specific models. The general embedding models include **BERT**, **GTE** [24], **BGE** [21], and **LLM2Vec** [3]. The recommendation-specific models include **EasyRec** [39], **BLAIR** [13], and **LLMEmb** [29].

These embedding models provide better initialization for item representation learning and can be integrated into various sequential recommenders. To evaluate their effectiveness, we test these embedding models on two different sequential recommender architectures: **GRU4Rec** [12], and **SASRec** [15]. We leave detailed introductions to each embedding model and downstream sequential recommenders in Appendix A.2.

[2]https://cseweb.ucsd.edu/~jmcauley/datasets/goodreads.html

4.1.3 Implementation Details. Our LLM2Rec is initialized with a pre-trained LLM. Unless otherwise specified, all experimental results reported for LLM2Rec in this paper are based on the Qwen2-0.5B backbone. In the first collaborative supervised fine-tuning stage, we utilize the AdamW optimizer [33] with learning rate set to $3e{-}4$. The model is fully fine-tuned with all parameters open for 10,000 steps with the effective batch size set to 128. Then for the masked next token prediction, following the established setting of prior works [3, 17], we randomly mask 20% of the input tokens and adopt the same hyperparameter settings as LLM2Vec [3]. The model is only fine-tuned for 1,000 steps with the effective batch size set to 32, which takes less than 2 hours on one single Nvidia A40 GPU. Finally, for item-level contrastive learning, item representations are augmented with dropout rate set to 0.2 and contrastive learning temperature τ set to 0.2. The model is optimized with AdamW optimizer for 1,000 steps with learning rate set to $2e{-}4$ and effective batch size set to 256.

For downstream recommenders, all models are trained using cross-entropy loss with AdamW optimizer. To ensure fair comparisons across different embedding models, we use a fixed set of hyperparameters for each recommender while varying only the text embeddings. The learning rate is set to $1e{-}3$ for SASRec and $1e-4$ for GRU4Rec. Across all three recommenders, the weight decay is fixed at $1e-4$, dropout out rate at 0.3, and the extracted text embeddings are projected to 128 dimensions using a trainable linear layer. The recommenders are trained for up to 500 epochs with an early stopping mechanism, which terminates training if the validation performance does not improve for 20 consecutive epochs. Experiments in this paper are conducted on 4 Nvidia A40 (48G) GPUs.

4.2 Performance Comparison (RQ1)

We evaluate the effectiveness of our proposed LLM2Rec using two downstream sequential recommenders on three in-domain and three out-of-domain datasets. The comprehensive results, presented in Table 3, reveal the following key observations.

LLM2Rec consistently outperforms all baseline models across both recommenders on all datasets. For the in-domain datasets, it achieves an average relative improvement of 15% on Games and Arts, and over 7% on Movies. The significant performance gain has well demonstrated the LLM2Rec's ability to effectively capture collaborative filtering signals that significantly enhance sequential recommendation performance. More importantly, LLM2Rec also excels on out-of-domain datasets, which consist of item categories absent from the pre-training data. Even on Goodreads, which differs from the training set in both item categories and source platform, LLM2Rec maintains a moderate yet consistent performance gain. The strong results on out-of-domain datasets indicate that training on a diverse set of recommendation datasets can bring both CF awareness and generalization ability to unseen domains. These findings highlight the potential of LLMs as generalizable embedding models for recommendation.

Table 3: Performance comparison of different embedding methods under in-domain and out-of-domain datasets. R is shorts for Recall, N is short for NDCG, and %Improv. indicates the relative improvement compared to the strongest baselines.

In-Domain Datasets													
Models		Games				Arts				Movies			
		R@10	N@10	R@20	N@20	R@10	N@10	R@20	N@20	R@10	N@10	R@20	N@20
GRU4Rec	BERT	0.0365	0.0184	0.0573	0.0236	0.0363	0.0191	0.0559	0.0240	0.0243	0.0126	0.0383	0.0160
	GTE	0.0540	0.0290	0.0792	0.0353	0.0348	0.0185	0.0569	0.0240	0.0396	0.0195	0.0583	0.0242
	BGE	0.0491	0.0261	0.0760	0.0329	0.0413	0.0221	0.0632	0.0276	0.0379	0.0187	0.0587	0.0239
	LLM2Vec	0.0540	0.0286	0.0784	0.0348	0.0473	0.0274	0.0678	0.0325	0.0370	0.0187	0.0557	0.0234
	BLAIR	0.0455	0.0245	0.0713	0.0309	0.0416	0.0233	0.0639	0.0289	0.0379	0.0188	0.0583	0.0239
	EasyRec	0.0450	0.0235	0.0700	0.0298	0.0436	0.0232	0.0643	0.0284	0.0356	0.0180	0.0551	0.0229
	LLMEmb	0.0544	0.0298	0.0775	0.0357	0.0480	0.0277	0.0673	0.0325	0.0377	0.0196	0.0538	0.0236
	LLM2Rec	**0.0624**	**0.0344**	**0.0874**	**0.0408**	**0.0590**	**0.0366**	**0.0802**	**0.0419**	**0.0419**	**0.0214**	**0.0595**	**0.0258**
	%Improv.	14.76%	15.46%	10.35%	14.31%	22.83%	32.32%	18.16%	29.03%	5.92%	9.46%	1.46%	6.77%
SASRec	BERT	0.0585	0.0311	0.0863	0.0381	0.0650	0.0405	0.0869	0.0460	0.0447	0.0240	0.0646	0.0290
	GTE	0.0641	0.0349	0.0911	0.0418	0.0644	0.0394	0.0880	0.0454	0.0570	0.0300	0.0817	0.0363
	BGE	0.0733	0.0410	0.1022	0.0483	0.0748	0.0475	0.1006	0.0540	0.0626	0.0350	0.0847	0.0406
	LLM2Vec	0.0740	0.0407	0.1029	0.0480	0.0770	0.0506	0.1007	0.0566	0.0662	0.0384	0.0874	0.0438
	BLAIR	0.0654	0.0361	0.0954	0.0437	0.0648	0.0379	0.0906	0.0444	0.0581	0.0315	0.0801	0.0370
	EasyRec	0.0647	0.0357	0.0926	0.0428	0.0658	0.0395	0.0929	0.0463	0.0528	0.0278	0.0739	0.0331
	LLMEmb	0.0813	0.0487	0.1085	0.0555	0.0865	0.0601	0.1086	0.0657	0.0659	0.0390	0.0837	0.0435
	LLM2Rec	**0.0865**	**0.0521**	**0.1157**	**0.0595**	**0.0925**	**0.0637**	**0.1142**	**0.0692**	**0.0705**	**0.0429**	**0.0895**	**0.0477**
	%Improv.	6.42%	6.92%	6.70%	7.04%	6.95%	5.97%	5.16%	5.29%	6.49%	10.01%	2.40%	9.13%
Out-Of-Domain Datasets													
Models		Sports				Baby				Goodreads			
		R@10	N@10	R@20	N@20	R@10	N@10	R@20	N@20	R@10	N@10	R@20	N@20
GRU4Rec	BERT	0.0335	0.0183	0.0499	0.0224	0.0111	0.0050	0.0252	0.0086	0.0851	0.0412	0.1226	0.0506
	GTE	0.0295	0.0147	0.0459	0.0188	0.0226	0.0116	0.0340	0.0145	0.1169	0.0599	0.1701	0.0733
	BGE	0.0489	0.0281	0.0685	0.0330	0.0252	0.0131	0.0364	0.0158	0.1072	0.0585	0.1517	0.0697
	LLM2Vec	0.0663	0.0464	0.0810	0.0501	0.0254	0.0138	0.0362	0.0165	0.1174	0.0655	0.1643	0.0773
	BLAIR	0.0537	0.0316	0.0735	0.0366	0.0207	0.0099	0.0316	0.0127	0.0939	0.0496	0.1339	0.0596
	EasyRec	0.0492	0.0270	0.0674	0.0315	0.0207	0.0105	0.0275	0.0122	0.0951	0.0477	0.1364	0.0581
	LLMEmb	0.0705	0.0482	0.0861	0.0521	0.0252	0.0136	0.0378	0.0168	0.1219	0.0701	0.1667	0.0814
	LLM2Rec	**0.0828**	**0.0632**	**0.0948**	**0.0662**	**0.0327**	**0.0181**	**0.0463**	**0.0216**	**0.1299**	**0.0761**	**0.1738**	**0.0872**
	%Improv.	17.50%	31.18%	10.07%	27.06%	28.55%	31.61%	22.32%	28.51%	6.58%	8.68%	2.17%	7.15%
SASRec	BERT	0.0860	0.0649	0.1017	0.0689	0.0114	0.0050	0.0232	0.0080	0.1479	0.0858	0.1929	0.0972
	GTE	0.0823	0.0584	0.1001	0.0629	0.0264	0.0142	0.0387	0.0173	0.1488	0.0851	0.1944	0.0967
	BGE	0.0974	0.0736	0.1141	0.0778	0.0428	0.0250	0.0569	0.0286	0.1445	0.0813	0.1972	0.0945
	LLM2Vec	0.1079	0.0854	0.1234	0.0893	0.0561	0.0339	0.0722	0.0379	0.1424	0.0790	0.1906	0.0911
	BLAIR	0.0893	0.0614	0.1091	0.0664	0.0332	0.0180	0.0484	0.0218	0.1508	0.0860	0.2000	0.0984
	EasyRec	0.0887	0.0627	0.1061	0.0671	0.0271	0.0154	0.0381	0.0182	0.1445	0.0825	0.1908	0.0941
	LLMEmb	0.1131	0.0936	0.1257	0.0969	0.0659	0.0439	0.0807	0.0476	0.1374	0.0778	0.1838	0.0895
	LLM2Rec	**0.1170**	**0.0976**	**0.1289**	**0.1006**	**0.0708**	**0.0503**	**0.0850**	**0.0539**	**0.1530**	**0.0897**	**0.2017**	**0.1020**
	%Improv.	3.51%	4.26%	2.56%	3.89%	7.39%	14.56%	5.32%	13.04%	1.45%	4.37%	0.83%	3.73%

For general text embedding models, the latest methods surpass earlier models like BERT due to their enhanced semantic understanding capabilities. In contrast, recommendation-specific embedding models such as BLAIR and EasyRec benefit from learning CF signals, allowing them to outperform general text embeddings like BERT. However, their language backbones with limited semantic understanding constrain their effectiveness, resulting in lower performance compared to more advanced models like BGE and LLM2Vec. LLMEmb inherits the strong semantic understanding of LLMs and further achieves performance gains by fine-tuning on recommendation-specific tasks. These results demonstrate that

Table 4: The ablation study of LLM2Rec.

Models	Games		Sports		Goodreads	
	R@10	N@10	R@10	N@10	R@10	N@10
Casual	0.0373	0.0201	0.0167	0.0091	0.0724	0.0375
Bidirectional	0.0740	0.0407	0.1079	0.0854	0.1424	0.0790
CSFT	0.0795	0.0472	0.1119	0.0935	0.1513	0.0882
IEM_1 (MNTP)	0.0801	0.0477	0.1147	0.0956	0.1564	0.0916
IEM_2 (IC)	0.0865	0.0521	0.1170	0.0976	0.1530	0.0897

both comprehensive semantic comprehension and CF awareness are crucial for improving downstream recommenders, underscoring the importance of developing a powerful embedding model integrating both aspects.

4.3 Ablation Study (RQ2)

To analyze the contribution of each training stage to the performance of LLM2Rec, we conduct an ablation study using the strongest sequential recommender, SASRec, as the fixed downstream model. The evaluation covers one in-domain dataset (Games) and two out-of-domain datasets (Sports and Goodreads). The detailed results are presented in Table 4. "Causal" and "Bidirectional" present the performance of our backbone LLM, Qwen2-0.5B, under different embedding generation strategies. In the causal setting, item embeddings are derived from the last hidden state of the [EOS] token while maintaining the causal attention mask. In the bidirectional setting, embeddings are obtained by average pooling the last hidden states of all tokens in the item title. Experimental results demonstrate that bidirectional attention consistently outperforms causal attention. While causal attention is beneficial for language generation tasks by conditioning later tokens on prior ones, it proves suboptimal for embedding generation, as it limits the model's ability to capture comprehensive contextual representations.

The bottom half of Table 4 highlights the impact of each training stage on the performance of LLM2Rec. "CSFT" represents the model's performance after Collaborative Supervised Fine-Tuning (CSFT). Item-level Embedding Modeling (IEM) consists of two steps: "IEM_1 (MNTP)" indicates the performance after additional Masked Next-Token Prediction (MNTP) training, and IEM_2 (IC) reflects the performance after item-level contrastive learning (IC), which ultimately results in LLM2Rec. Experimental results indicate that collaborative supervised fine-tuning (CSFT) contributes the most significant performance improvement across both in-domain and out-of-domain datasets, highlighting the importance of capturing CF signals in sequential recommendation and also demonstrating the effectiveness of CSFT. In comparison, masked next-token prediction (MNTP) provides a smaller yet consistent performance boost, suggesting its role in improving bidirectional contextual representations. Finally, item-level contrastive learning further enhances performance, yielding substantial gains on in-domain datasets and moderate improvements on out-of-domain datasets.

4.4 Model Study (RQ3)

4.4.1 Effect of Different LLM Backbones. As LLM2Rec builds upon pre-trained LLMs, the choice of backbone naturally affects its performance. To examine this effect, we evaluate LLM2Rec on various LLM backbones. Specifically, for each LLM backbone, we evaluate the quality of the generated embedding after two training stages of LLM2Rec, *i.e.,* the collaborative supervised fine-tuning (LLM2Rec-Stage1) and item-level embedding modeling (LLM2Rec-Stage2). To provide a comprehensive comparison, we further include two baseline methods: LLM2Vec, a state-of-the-art general-purpose embedding model, and LLMEmb, a sequential recommendation-specific embedding model. Since LLM2Rec, LLM2Vec, and LLMEmb are all fine-tuned from a pre-trained LLM, we compare their performance under the same backbone LLM. All embeddings are integrated into the same downstream recommender, SASRec, and tested on both the in-domain dataset (Games) and the out-of-domain dataset (Sports).

Table 5: Effect of mixed training dataset. ID is short for in-domain and OOD is short for out-of-domain.

Models	Games (ID)		Sports (OOD)		Goodreads (OOD)	
	R@10	N@10	R@10	N@10	R@10	N@10
Backbone	0.0740	0.0407	0.1079	0.0854	0.1424	0.0790
Single	0.0857	0.0500	0.1099	0.0921	0.1473	0.0833
%Improv.	15.81%	22.90%	1.83%	7.76%	3.41%	5.44%
Mix-2	0.0856	0.0517	0.1111	0.0936	0.1493	0.0862
%Improv.	15.66%	26.84%	2.96%	9.52%	4.82%	9.12%
Mix-6	0.0795	0.0472	0.1119	0.0935	0.1513	0.0882
%Improv.	7.38%	15.80%	3.68%	9.45%	6.20%	11.66%

As the results shown in Figure 4, models built upon stronger LLM backbones generally achieve better performance on both the in-domain (Games) and out-of-domain (Sports) datasets. This trend is consistent with expectations, as larger and more powerful LLMs tend to possess stronger semantic understanding and greater generalization capabilities. Across all evaluated backbones, both training stages of LLM2Rec contribute to consistent improvements in the effectiveness of the generated embeddings for recommendation tasks. Notably, LLM2Rec consistently outperforms the general-purpose embedding baseline, LLM2Rec, on a variety of LLM backbones. Furthermore, after completing both stages of training, LLM2Rec (LLM2Rec-Stage2) also surpasses the recommendation-specific embedding method, LLMEmb. The results on four different LLMs confirm the effectiveness of LLM2Rec and its robustness in generalizing across different LLM backbones. It is also notable that LLM2Rec achieves competitive performance when built on the lightweight Qwen2-0.5B model, offering a favorable trade-off between effectiveness and computational cost compared to larger backbones.

4.4.2 Effect of Mixed Dataset Training. Our proposed LLM2Rec is pre-trained on a mixture of six datasets spanning different categories. To examine the impact of mixed pre-training datasets, we evaluate embedding models trained on a single dataset, a mixture of two datasets, and a mixture of six datasets. For simplicity, we omit the item-level embedding modeling steps (masked next-token prediction and item-level contrastive learning), and all results are reported using SASRec as the fixed downstream recommender. We define four settings: "Backbone", which represents the base performance of the backbone LLM, Qwen2-0.5B; "Single", where the model is pre-trained only on the Games dataset; "Mix-2", pre-trained on a

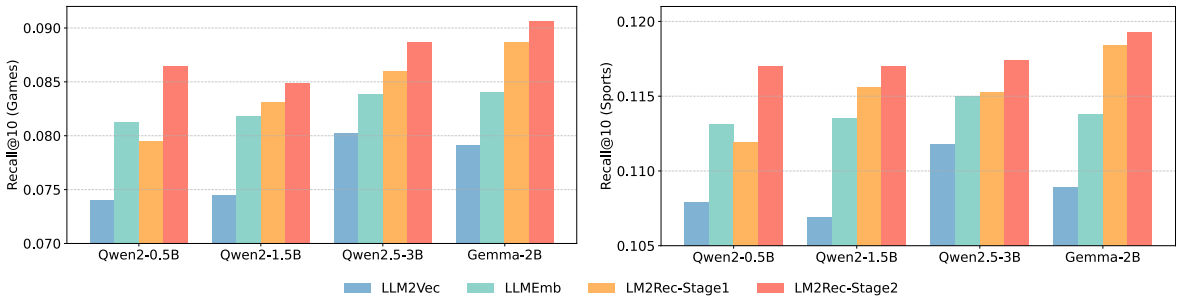

Figure 4: Performance comparison of embedding methods across different LLM backbones.

combination of Games and Arts; and "Mix-6", which corresponds to LLM2Rec, pre-trained on all six datasets. Among the three evaluated datasets, Games is in-domain for all models, whereas Sports and Goodreads are out-of-domain, as they are excluded from the pre-training datasets.

Experimental results as shown in Table 5 indicate that pre-training on a diverse set of datasets improves the model's ability to generalize to unseen domains, leading to more robust embeddings across different datasets. Meanwhile, for in-domain data, pre-training on a concentrated subset of item categories yields higher accuracy, suggesting that category-specific pre-training can be beneficial for domain-specific recommendations.

4.4.3 Efficiency Analysis. In addition to effectiveness, the efficiency of embedding models is crucial for practical deployment in sequential recommender systems. We measure the inference time required by each embedding model to encode all item titles in the Games dataset (comprising 9,517 items) on a single Nvidia A40 GPU. The results are shown in Figure 5. The vertical axis represents the total inference time, while the horizontal axis shows the Recall@10 performance of SASRec under different embedding models.

The smallest embedding models, BERT and BLAIR, take the shortest inference times, with BLAIR achieving a notable performance improvement over BERT due to its specialized fine-tuning for recommendation tasks. The latest embedding models generally deliver better performance, driven by enhanced semantic understanding, but at the cost of higher computational overhead. Larger models, such as GTE, which leverages a 7B-parameter backbone, suffer from significantly increased inference time. LLM2Vec and LLMEmb reported in Figure 5 are built on the lightweight yet powerful Qwen2-0.5B model. They achieve strong performance with relatively low computational cost, demonstrating a favorable balance between effectiveness and efficiency. Our proposed LLM2Rec inherits the efficiency of Qwen2-0.5B while enhancing performance through recommendation-specific fine-tuning. Overall, LLM2Rec generates more effective embeddings for downstream sequential recommenders while maintaining practical inference efficiency.

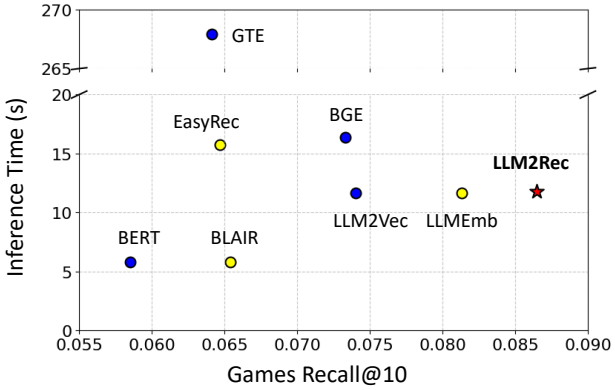

Figure 5: Comparison of inference time and performance of different embedding models.

5 Conclusion & Discussion

In this work, we introduced LLM2Rec, a specialized embedding model for sequential recommendation that incorporates both the comprehensive semantic understanding of LLMs and awareness of CF signals. General embedding models fail to capture the latent item relationships, *i.e.*, CF signals, resulting in suboptimal performance in sequential recommendation tasks. LLM2Rec bridges this gap by leveraging a two-stage training framework including collaborative supervised fine-tuning (CSFT) and item-level embedding modeling (IEM). CSFT fine-tunes the LLM to capture the CF signals with user interaction sequences and IEM further transforms the decoder-only LLM into embedding model focusing on item embedding generation for sequential recommendation. Extensive experiments on real-world datasets demonstrate that LLM2Rec consistently outperforms strong baseline embedding models across both in-domain and out-of-domain recommendation tasks. Notably, it achieves significant improvements while maintaining computational efficiency. Our results highlight the potential of LLMs as powerful embedding models for sequential recommendation.

While LLM2Rec demonstrates strong effectiveness, several promising research directions remain. First, real-world user interaction data from e-commerce platforms often contain substantial noise. Enhancing robustness through noise filtering or data augmentation could further improve LLM2Rec. Second, due to computational constraints, this study evaluates LLM2Rec on LLM backbones with up to 3B parameters. Despite its effectiveness, exploring larger-scale LLMs under the LLM2Rec framework could unlock further performance gains. Third, experimental results indicate that training on mixed datasets enhances generalization. Constructing a more diverse dataset with items from multiple platforms and categories could improve generalization, advancing toward a universal recommender system trained once and deployed widely.

acknowledgement

This research is supported by NExT++ Research Center.

References

[1] Keqin Bao, Jizhi Zhang, Wenjie Wang, Yang Zhang, Zhengyi Yang, Yancheng Luo, Chong Chen, Fuli Feng, and Qi Tian. 2023. A bi-step grounding paradigm for large language models in recommendation systems. *arXiv preprint arXiv:2308.08434* (2023).

[2] 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. In *RecSys*. 1007–1014.

[3] Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. Llm2vec: Large language models are secretly powerful text encoders. *arXiv preprint arXiv:2404.05961* (2024).

[4] Gal Chechik, Varun Sharma, Uri Shalit, and Samy Bengio. 2010. Large scale online learning of image similarity through ranking. *Journal of Machine Learning Research* 11, 3 (2010).

[5] Junyi Chen, Lu Chi, Bingyue Peng, and Zehuan Yuan. 2024. Hllm: Enhancing sequential recommendations via hierarchical large language models for item and user modeling. *arXiv preprint arXiv:2409.12740* (2024).

[6] Yuxin Chen, Junfei Tan, An Zhang, Zhengyi Yang, Leheng Sheng, Enzhi Zhang, Xiang Wang, and Tat-Seng Chua. 2024. On softmax direct preference optimization for recommendation. *arXiv preprint arXiv:2406.09215* (2024).

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 248–255.

[8] Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821* (2021).

[9] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *SIGIR*. 639–648.

[10] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *WWW*. 173–182.

[11] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. 2021. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF international conference on computer vision*. 8340–8349.

[12] Balázs Hidasi and Alexandros Karatzoglou. 2018. Recurrent neural networks with top-k gains for session-based recommendations. In *CIKM*. 843–852.

[13] Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. 2024. Bridging language and items for retrieval and recommendation. *arXiv preprint arXiv:2403.03952* (2024).

[14] Yupeng Hou, Shanlei Mu, Wayne Xin Zhao, Yaliang Li, Bolin Ding, and Ji-Rong Wen. 2022. Towards universal sequence representation learning for recommender systems. In *SIGKDD*. 585–593.

[15] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *ICDM*. IEEE, 197–206.

[16] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906* (2020).

[17] 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. Minneapolis, Minnesota.

[18] Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2024. NV-Embed: Improved Techniques for Training LLMs as Generalist Embedding Models. *arXiv preprint arXiv:2405.17428* (2024).

[19] Jinhyuk Lee, Zhuyun Dai, Xiaoqi Ren, Blair Chen, Daniel Cer, Jeremy R Cole, Kai Hui, Michael Boratko, Rajvi Kapadia, Wen Ding, et al. 2024. Gecko: Versatile text embeddings distilled from large language models. *arXiv preprint arXiv:2403.20327* (2024).

[20] Mike Lewis. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461* (2019).

[21] Chaofan Li, MingHao Qin, Shitao Xiao, Jianlyu Chen, Kun Luo, Yingxia Shao, Defu Lian, and Zheng Liu. 2024. Making text embedders few-shot learners. *arXiv preprint arXiv:2409.15700* (2024).

[22] 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. In *SIGKDD*. 1258–1267.

[23] Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281* (2023).

[24] Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281* (2023).

[25] Jianghao Lin, Rong Shan, Chenxu Zhu, Kounianhua Du, Bo Chen, Shigang Quan, Ruiming Tang, Yong Yu, and Weinan Zhang. 2024. Rella: Retrieval-enhanced large language models for lifelong sequential behavior comprehension in recommendation. In *WWW*. 3497–3508.

[26] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 740–755.

[27] Dugang Liu, Shenxian Xian, Xiaolin Lin, Xiaolian Zhang, Hong Zhu, Yuan Fang, Zhen Chen, and Zhong Ming. 2024. A Practice-Friendly Two-Stage LLM-Enhanced Paradigm in Sequential Recommendation. *arXiv preprint arXiv:2406.00333* (2024).

[28] Qijiong Liu, Nuo Chen, Tetsuya Sakai, and Xiao-Ming Wu. 2024. Once: Boosting content-based recommendation with both open-and closed-source large language models. In *SIGKDD*. 452–461.

[29] Qidong Liu, Xian Wu, Wanyu Wang, Yejing Wang, Yuanshao Zhu, Xiangyu Zhao, Feng Tian, and Yefeng Zheng. 2024. Large language model empowered embedding generator for sequential recommendation. *arXiv preprint arXiv:2409.19925* (2024).

[30] Qidong Liu, Xian Wu, Xiangyu Zhao, Yejing Wang, Zijian Zhang, Feng Tian, and Yefeng Zheng. 2024. Large Language Models Enhanced Sequential Recommendation for Long-tail User and Item. *arXiv preprint arXiv:2405.20646* (2024).

[31] Xiaohao Liu, Zhulin Tao, Jiahong Shao, Lifang Yang, and Xianglin Huang. 2022. Elimrec: Eliminating single-modal bias in multimedia recommendation. In *Proceedings of the 30th ACM International Conference on Multimedia*. 687–695.

[32] Yinhan Liu. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* 364 (2019).

[33] I Loshchilov. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101* (2017).

[34] Bhaskar Mitra, Fernando Diaz, and Nick Craswell. 2017. Learning to match using local and distributed representations of text for web search. In *Proceedings of the 26th international conference on world wide web*. 1291–1299.

[35] Niklas Muennighoff. 2022. Sgpt: Gpt sentence embeddings for semantic search. *arXiv preprint arXiv:2202.08904* (2022).

[36] Niklas Muennighoff, Hongjin Su, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. 2024. Generative representational instruction tuning. *arXiv preprint arXiv:2402.09906* (2024).

[37] Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. MTEB: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316* (2022).

[38] Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y Zhao, Yi Luan, Keith B Hall, Ming-Wei Chang, et al. 2021. Large dual encoders are generalizable retrievers. *arXiv preprint arXiv:2112.07899* (2021).

[39] Xubin Ren and Chao Huang. 2024. EasyRec: Simple yet Effective Language Models for Recommendation. *arXiv preprint arXiv:2408.08821* (2024).

[40] Xubin Ren, Wei Wei, Lianghao Xia, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. 2024. Representation learning with large language models for recommendation. In *WWW*. 3464–3475.

[41] Leheng Sheng, An Zhang, Yi Zhang, Yuxin Chen, Xiang Wang, and Tat-Seng Chua. 2024. Language Representations Can be What Recommenders Need: Findings and Potentials. *arXiv preprint arXiv:2407.05441* (2024).

[42] Jacob Mitchell Springer, Suhas Kotha, Daniel Fried, Graham Neubig, and Aditi Raghunathan. 2024. Repetition improves language model embeddings. *arXiv preprint arXiv:2402.15449* (2024).

[43] 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 *CIKM*. 1441–1450.

[44] Jiaxi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *WSDM*. 565–573.

[45] Zhulin Tao, Xiaohao Liu, Yewei Xia, Xiang Wang, Lifang Yang, Xianglin Huang, and Tat-Seng Chua. 2022. Self-supervised learning for multimedia recommendation. *IEEE Transactions on Multimedia* 25 (2022), 5107–5116.

[46] 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).

[47] Mengting Wan and Julian McAuley. 2018. Item recommendation on monotonic behavior chains. In *RecSys*. 86–94.

[48] Mengting Wan, Rishabh Misra, Ndapa Nakashole, and Julian McAuley. 2019. Fine-grained spoiler detection from large-scale review corpora. *arXiv preprint arXiv:1905.13416* (2019).

[49] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533* (2022).

[50] Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2023. Improving text embeddings with large language models. *arXiv preprint arXiv:2401.00368* (2023).

[51] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *SIGIR*. 165–174.

[52] Wei Wei, Xubin Ren, Jiabin Tang, Qinyong Wang, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. 2024. Llmrec: Large language models with graph augmentation for recommendation. In *SIGKDD*. 806–815.

[53] Yinwei Wei, Xiang Wang, Qi Li, Liqiang Nie, Yan Li, Xuanping Li, and Tat-Seng Chua. 2021. Contrastive learning for cold-start recommendation. In *MM*. 5382–5390.

[54] Yinwei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, Richang Hong, and Tat-Seng Chua. 2019. MMGCN: Multi-modal graph convolution network for personalized recommendation of micro-video. In *Proceedings of the 27th ACM international conference on multimedia*. 1437–1445.

[55] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In *SIGIR*. 726–735.

[56] Yunjia Xi, Weiwen Liu, Jianghao Lin, Xiaoling Cai, Hong Zhu, Jieming Zhu, Bo Chen, Ruiming Tang, Weinan Zhang, and Yong Yu. 2024. Towards open-world recommendation with knowledge augmentation from large language models. In *RecSys*. 12–22.

[57] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. 2022. Contrastive learning for sequential recommendation. In *ICDE*. IEEE, 1259–1273.

[58] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. 2024. Qwen2 Technical Report. *CoRR* abs/2407.10671 (2024).

[59] Fajie Yuan, Xiangnan He, Alexandros Karatzoglou, and Liguang Zhang. 2020. Parameter-efficient transfer from sequential behaviors for user modeling and recommendation. In *SIGIR*. 1469–1478.

[60] Zheng Yuan, Fajie Yuan, Yu Song, Youhua Li, Junchen Fu, Fei Yang, Yunzhu Pan, and Yongxin Ni. 2023. Where to go next for recommender systems? id-vs. modality-based recommender models revisited. In *SIGIR*. 2639–2649.

[61] Chao Zhang, Shiwei Wu, Haoxin Zhang, Tong Xu, Yan Gao, Yao Hu, and Enhong Chen. 2024. NoteLLM: A Retrievable Large Language Model for Note Recommendation. In *WWW*. 170–179.

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

[63] Xinyu Zhang, Linmei Hu, Luhao Zhang, Dandan Song, Heyan Huang, and Liqiang Nie. 2024. Laser: Parameter-Efficient LLM Bi-Tuning for Sequential Recommendation with Collaborative Information. *arXiv preprint arXiv:2409.01605* (2024).

[64] 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 *CIKM*. 1893–1902.

[65] Feng Zhu, Yan Wang, Chaochao Chen, Jun Zhou, Longfei Li, and Guanfeng Liu. 2021. Cross-domain recommendation: challenges, progress, and prospects. *arXiv preprint arXiv:2103.01696* (2021).

A Implemental Details

A.1 Training Pseudo-code.

Here we provide the training pseudo-code, indicating that LLM2Rec is sequentially optimized with different stages, as shown in Algorithm 1. The training starts from collaborative supervised fine-tuning to item-level embedding modeling. Due to the different objectives, we employ different sampling at different stages. Specifically, the user interaction sequence and the next item pairs constitute the samples for the collaborative supervised fine-tuning, while single items serve for the item-level embedding modeling. Additionally, during the item-level contrastive learning, we augment the items to two views, making it an extra sampling that differs from the training with MNTP objective.

Algorithm 1 Training Strategy for LLM2Rec

Require: Training dataset $\mathcal{D}^{\text{train}}$, pretrained LLM $\mathcal{E}$ with parameter θ, learning rate η, epochs E, temperature τ

Initialization: Load pretrained LLM $\mathcal{E}$, Initialize optimizer
Stage 1: Collaborative Supervised Fine-tuning
for epoch = 1 to E_1 **do**
  for each batch $\{(X_u, i_{N_u+1})\} \subset \mathcal{D}^{\text{train}}$ **do**
    Encode item sequence: $\mathbf{t}_u \leftarrow \text{Tokenize}(X_u)$
    Compute loss $\mathcal{L}_{\text{CSFT}}$ in Equation 2
    Update model parameters: $\theta_1 \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_{\text{CSFT}}$
  end for
end for
Stage 2: Item-level Embedding Modeling
Step 2.1: Reform LLM with Bidirectional Attention
for epoch = 1 to E_2 **do**
  for each batch $\{i\} \subset \mathcal{I}$ **do**
    Tokenize item description: $\mathbf{t}_i$
    Randomly mask tokens for MNTP
    Compute loss $\mathcal{L}_{\text{MNTP}}$ in Equation 3
    Update model parameters: $\theta' \leftarrow \theta - \eta \nabla_{\theta_1} \mathcal{L}_{\text{MNTP}}$
  end for
end for
Step 2.2: Item-level Contrastive Learning
for epoch = 1 to E_3 **do**
  for each batch $\{i\} \subset \mathcal{I}$ **do**
    Generate two masked views: $\tilde{\mathbf{t}}_i^1, \tilde{\mathbf{t}}_i^2$
    Compute item embeddings: $\mathbf{z}_i^1 = \mathcal{E}(\tilde{\mathbf{t}}_i^1)$, $\mathbf{z}_i^2 = \mathcal{E}(\tilde{\mathbf{t}}_i^2)$
    Compute loss $\mathcal{L}_{\text{IC}}$ in Equation 4
    Update model parameters: $\tilde{\theta} \leftarrow \theta' - \eta \nabla_{\theta'} \mathcal{L}_{\text{IC}}$
  end for
end for
return LLM2Rec $\mathcal{E}$ with parameter $\tilde{\theta}$.

A.2 Baselines and Sequential Recommenders.

This section provides a brief introduction to each baseline embedding model used in our experiments.

- **BERT** [17] is a milestone embedding model pre-trained using mask language modeling and next sentence prediction. Built upon the transformer architecture with bidirectional attention, BERT effectively captures contextual dependencies in text, enabling more accurate semantic representations.
- **BGE** [21] is a state-of-the-art embedding model built on a bidirectional Transformer architecture. Pre-trained on diverse datasets, it excels in retrieval and reranking tasks. In this paper, we use the pre-trained model BAAI/bge-large-en-v1.5 from the Hugging Face repository.
- **GTE** [24] transforms the LLM into an embedding model with multi-stage contrastive learning. Specifically, we select the pre-trained model Alibaba-NLP/gte-Qwen2-7B-instruct which exhibits the highest performance on Massive Text Embedding Benchmark (MTEB) [37].
- **LLM2Vec** [3] aims to effectively adapt pre-trained LLMs into embedding models through sentence-level adaptations. In this paper, we use Qwen2-0.5B as the backbone for LLM2Vec, ensuring consistency with our proposed LLM2Rec.
- **EasyRec** [39] is a recommendation-specific embedding model built on RoBERTa [32]. It leverages contrastive learning to capture collaborative filtering (CF) signals by aligning representations of user and item profiles. We load the pre-trained embedding from Hugging Face repo hkuds/easyrec-roberta-large.
- **BLAIR** [13] is a recommendation-specific embedding model, similar to EasyRec. It is pre-trained on a diverse collection of 33 Amazon datasets, comprising 3.08×10^7 data instances. In this paper, we load the pre-trained model from Hugging Face repo hyp1231/blair-roberta-base.
- **LLMEmb** [29] adopts attribute-level augmentation and uses contrastive learning as a primary training objective to improve semantic understanding of LLM embeddings, particularly for representing the long-tail items. In this paper, we implement LLMEmb using the same LLM backbone, Qwen2-0.5B, to ensure consistency with LLM2Rec.

Two downstream sequential recommenders are used as evaluation to the embedding models:

- **GRU4Rec** [12] utilizes GRU modules to capture sequential dependencies within user interaction sequences. It is trained to predict the next item in a user's sequence based on previously purchased items. We use the cross-entropy loss as optimization objective during the training process.
- **SASRec** [15] is a transformer-based recommender widely used in sequential recommendation. It leverages self-attention to capture long-range dependencies in user interaction sequences, enhancing the accuracy of future interaction predictions. Cross-entropy loss is used as the optimization objective.