RecGPT: A Foundation Model for Sequential Recommendation

Yangqin Jiang
University of Hong Kong
Hong Kong, China
mrjiangyq99@gmail.com

Xubin Ren
University of Hong Kong
Hong Kong, China
xubinrencs@gmail.com

Lianghao Xia
University of Hong Kong
Hong Kong, China
aka_xia@foxmail.com

Da Luo
Wechat, Tencent
Guang Zhou, China
lodaluo@tencent.com

Kangyi Lin
Wechat, Tencent
Guang Zhou, China
plancklin@tencent.com

Chao Huang[*]
University of Hong Kong
Hong Kong, China
chaohuang75@gmail.com

Abstract

This work addresses a fundamental barrier in recommender systems: the inability to generalize across domains without extensive retraining. Traditional ID-based approaches fail entirely in cold-start and cross-domain scenarios where new users or items lack sufficient interaction history. Inspired by foundation models' cross-domain success, we develop a foundation model for sequential recommendation that achieves genuine zero-shot generalization capabilities. Our approach fundamentally departs from existing ID-based methods by deriving item representations exclusively from textual features. This enables immediate embedding of any new item without model retraining. We introduce unified item tokenization with Finite Scalar Quantization that transforms heterogeneous textual descriptions into standardized discrete tokens. This eliminates domain barriers that plague existing systems. Additionally, the framework features hybrid bidirectional-causal attention that captures both intra-item token coherence and inter-item sequential dependencies. An efficient catalog-aware beam search decoder enables real-time token-to-item mapping. Unlike conventional approaches confined to their training domains, RecGPT naturally bridges diverse recommendation contexts through its domain-invariant tokenization mechanism. Comprehensive evaluations across six datasets and industrial scenarios demonstrate consistent performance advantages. Our model is open-source and available at: `https://github.com/HKUDS/RecGPT`.

1 Introduction

Recommender systems (RSs) have emerged as indispensable tools for navigating the overwhelming sea of digital content, offering personalized guidance across diverse platforms including e-commerce marketplaces (Wang et al., 2020), multimedia streaming services (Jiang et al., 2024a), and social networking sites (Jamali and Ester, 2010). By intelligently filtering vast information landscapes, these systems not only enhance user satisfaction but also drive engagement and retention for platform operators. Within this ecosystem, sequential recommendation approaches have gained particular prominence for their ability to capture temporal dynamics and evolving preferences, enabling more accurate predictions of users' future interactions by modeling the intricate patterns within their historical behavior sequences (Fang et al., 2020).

In the sequential recommendation, existing frameworks hit fundamental barriers when confronted with new contexts, data-sparse environments, or cold-start scenarios (Zhao et al., 2023; Zhang et al., 2025), invariably demanding resource-intensive retraining cycles. This critical limitation creates an urgent need for a transformative recommendation paradigm with robust **Zero-Shot Generalization** capabilities that can adapt seamlessly across diverse scenarios without prior exposure to specific users or items, effectively making recommendations in previously unseen contexts while maintaining recommendation quality.

Inspired by the remarkable success of foundation models in visual and language domains (Liu et al., 2023; DeepSeek-AI, 2024), which achieve exceptional cross-domain generalization through large-scale pre-training, a compelling question emerges: **Can we develop foundation models for sequential recommenders with effective pre-training paradigms?** Such models would enable efficient knowledge transfer across diverse recommendation scenarios without the computational burden of extensive retraining. This capability would be particularly valuable in challenging sparse data conditions, cold-start situations, and zero-shot recommendation contexts that current recommender systems struggle to address effectively. However, realizing this vision requires addressing several fundamental challenges:

[*] Chao Huang is the corresponding author.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 10129–10143
November 4-9, 2025 ©2025 Association for Computational Linguistics

t

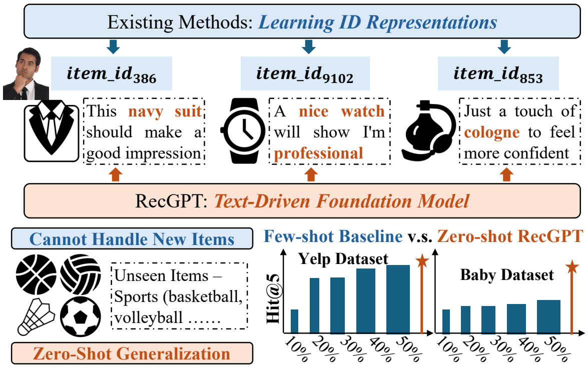

Existing Methods: Learning ID Representations
item_id386
item_id9102
item_id853
This navy suit should make a good impression
A nice watch will show I'm professional
Just a touch of cologne to feel more confident
RecGPT: Text-Driven Foundation Model
Cannot Handle New Items
Unseen Items – Sports (basketball, volleyball ……
Zero-Shot Generalization
Few-shot Baseline v.s. Zero-shot RecGPT
Yelp Dataset
Baby Dataset
Hit@5
10%
20%
30%
40%
50%
10%
20%
30%
40%
50%

Figure 1: Our RecGPT model demonstrates strong cross-domain zero-shot generalization capabilities, consistently outperforming existing recommender systems in few-shot scenarios (even when those systems incorporate 10%-50% of downstream unseen data) without requiring any domain-specific training data.

The Semantic Heterogeneity of Item Representations. Inspired by the success of autoregressive LLMs in achieving cross-domain generalization through next-token prediction, we recognize a natural fit for sequential recommendation, which inherently models temporal patterns in user interactions. However, directly applying this paradigm faces a critical challenge: the semantic heterogeneity of item representations across domains. Unlike language tokens that share universal linguistic structures, items from different platforms (*e.g.*, e-commerce products, videos, news) exhibit vastly different descriptive formats and attribute spaces. This heterogeneity creates insurmountable barriers for traditional ID-based systems, which fail to transfer knowledge across domains. To address this limitation, recommendation foundation models must develop a domain-invariant tokenization mechanism that unifies diverse textual descriptions into a standardized discrete token space–preserving semantic richness while enabling the powerful generalization capabilities of autoregressive modeling.

Hierarchical Dependencies in Item Tokenization. Language tokenization enables straightforward autoregressive modeling—once words are mapped to tokens, the model simply predicts the next token without explicit word-level considerations. Item tokenization, however, presents a fundamentally different challenge: when an item is represented by multiple tokens, the model must simultaneously capture two levels of dependencies. At the macro level, it must model sequential relationships between items to understand user preference evolution. At the micro level, it must maintain coherence among tokens belonging to the same item to preserve its semantic integrity. This dual-dependency structure demands a more sophisticated attention mechanism that can distinguish between intra-item and inter-item relationships while maintaining the autoregressive framework necessary for generalization.

The Decoding Efficiency Challenge. The computational complexity of converting token predictions to item recommendations creates a critical bottleneck for real-world deployment. The vast space of possible token combinations far exceeds actual catalog sizes, making naive search approaches computationally prohibitive—a vocabulary of 50,000 tokens with 4-token items yields $6.25 \times 10^{1}8$ theoretical combinations versus millions of actual items. Therefore, efficient decoding mechanisms must exploit semantic structure and approximate search techniques to map token predictions to valid items at inference speeds compatible with real-time recommendation systems.

To address these challenges, we present RecGPT, a text-driven foundation model that reimagines sequential recommendation through the lens of autoregressive generation. Our framework introduces three architectural designs that collectively enable zero-shot generalization capabilities. First, we develop a Unified Item Tokenization mechanism that employs Finite Scalar Quantization (FSQ) to transform heterogeneous textual descriptions into a standardized discrete token space. This domain-invariant representation solves the semantic heterogeneity problem by creating a universal vocabulary that bridges diverse recommendation contexts, without requiring platform-specific adaptations. Unlike traditional ID-based systems confined to their training domains, our tokenization preserves rich semantic information while enabling cross-domain knowledge transfer.

Second, we propose a Universal Recommendation Modeling architecture featuring hybrid bidirectional-causal attention that elegantly resolves the hierarchical dependency challenge inherent in multi-token item representations. This innovative attention mechanism allows comprehensive information exchange among tokens representing the same item through bidirectional processing, while maintaining strict causal relationships between sequential items for accurate temporal modeling. We further introduce auxiliary semantic pathways that complement discrete tokens with continuous embeddings, effectively counteracting information loss during quantization. Third, our Efficient

Item Token Decoder employs catalog-aware beam search with Trie-based prefix constraints, transforming the computationally intractable token-to-item mapping into a practical real-time operation. By exploiting the sparse nature of valid item combinations within the vast token space, our decoder achieves optimal complexity while maintaining superior recommendation quality.

Through experiments on several datasets and industrial deployment serving millions of users, we show that RecGPT achieves superior zero-shot performance–consistently outperforming traditional recommenders. Our comprehensive analysis reveals exceptional cold-start capabilities, robust power-law scaling properties, and architectural advantages confirmed through systematic ablations.

2 Methodology

We propose a **Text-Driven Foundation Model** for sequential recommendation that fundamentally departs from traditional ID-based approaches. While conventional recommenders learn separate embeddings for each unique item ID–creating inherent limitations in generalization capabilities and domain adaptability (Jiang et al., 2024b)–our approach derives item representations exclusively from textual features (*e.g.*, titles, descriptions, and categories) through a specialized text encoder. This paradigm shift offers three substantial advantages: (1) **Zero-shot Transferability**, as any new item can be immediately embedded via its textual description without model retraining; (2) **Cross-domain Compatibility**, since textual semantics naturally bridge diverse recommendation contexts ranging from e-commerce products to video content and news articles; and (3) **Enhanced Robustness** in sparse-data and cold-start scenarios, where ID-based methods typically fail due to insufficient interaction histories but text-based representations remain informative via their rich semantic content.

2.1 Unified Item Tokenization

To unify item representations and bridge semantic gaps across recommendation scenarios, we employ a unified item tokenizer (Fig. 2 (i)). Items are first embedded into a consistent text representation space using their descriptions and then quantized into multiple discrete tokens applicable for various domains. This quantization paradigm uses a unified codebook, effectively tokenizing item embeddings and enabling efficient autoregressive pre-training.

2.1.1 Text-based Item Representation

We leverage MPNet (Song et al., 2020) as our foundational text encoder to derive item representations from textual descriptions. This strategic choice capitalizes on MPNet's innovative architecture that seamlessly integrates Masked Language Modeling (MLM) and Permuted Language Modeling (PLM), addressing fundamental limitations in contextual understanding that plague earlier transformer models (Devlin, 2018; Yang, 2019). Unlike ID-based methods that fail to generalize to new items, our text-driven approach enables zero-shot recommendations and eliminates the need for extensive item-specific training data, particularly advantageous for cold-start scenarios and long-tail items.

For a text sequence $X = (x_1, \ldots, x_n)$, our text encoder optimizes the following objective function:

$$\mathbb{E}_{z\in\mathcal{Z}_n} \sum_{t=c+1}^{n} \log P(x_{z_t} | x_{z_{<t}}, \Phi_{z_{>c}}; \theta), \qquad (1)$$

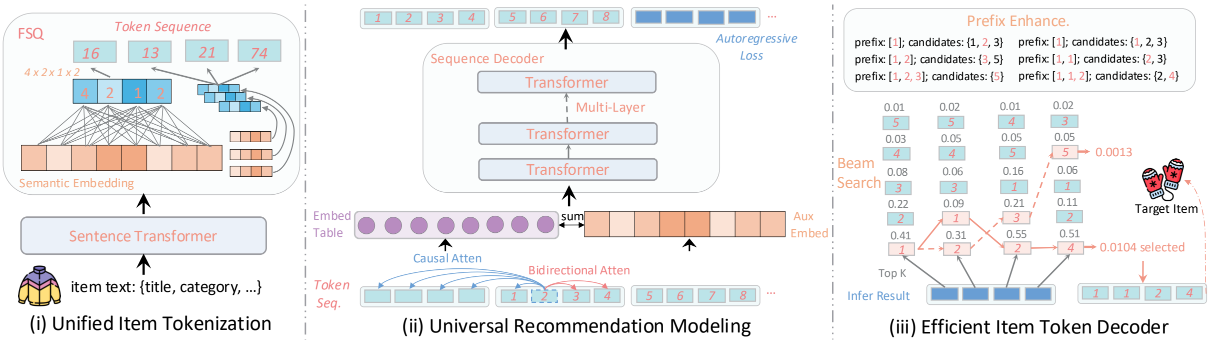
where $\mathcal{Z}_n$ represents permutations of indices $(1, \cdots, n)$, z_t and $z< t$ denote the t-th index and first $t-1$ elements in permutation z, and $\Phi_{z_{>c}}$ indicates masks in positions $z_{>c}$. After training with this objective, RecGPT generates d_L-dimensional embeddings e_i for item i with textual features X_i as:

$$e_i = \text{MPNet}(X_i), e_i \in \mathbb{R}^{d_L}. \qquad (2)$$

This approach produces semantically rich, domain-agnostic representations that maintain consistent meaning across diverse recommendation contexts. By grounding recommendations in language encoding, our model naturally bridges domain gaps and supports cross-domain transfer without requiring explicit adaptation mechanisms or specialized embeddings for recommendation contexts.

2.1.2 Quantizing Item Embeddings

To bridge the gap between continuous semantic spaces and discrete token-based processing, we introduce a novel embedding quantization mechanism inspired by advances in image processing (Esser et al., 2021). This critical component transforms our continuous item representations into discrete tokens—a representation format ideally suited for transformer architectures while maintaining semantic fidelity. We adopt Finite Scalar Quantization (Mentzer et al., 2023) (FSQ), which elegantly resolves the codebook collapse challenges that have limited previous quantization approaches.



(i) Unified Item Tokenization

(ii) Universal Recommendation Modeling

(iii) Efficient Item Token Decoder

Figure 2: Overall framework of the proposed RecGPT.

For each item i, our quantization strategy maps its embedding to a fixed-length token sequence $s_i = s_i^0, s_i^1, \cdots, s_i^{K-1}$, where each token $s_i^k \in \mathcal{C}$ belongs to a finite codebook. This transformation begins by partitioning the d_L-dimensional semantic embedding e_i into K sub-vectors $e_i^k \in \mathbb{R}^{d_L/K}$. This partitioning preserves local semantic structures while enabling more granular quantization.

The quantization process employs a carefully designed pipeline: first applying a sigmoid function $\sigma(\cdot)$ to normalize values to $(0, 1)$, then using a rounding function $R[\cdot]$ with hyperparameter L to map each component to one of L distinct integers. To achieve sufficient representational capacity, we divide each sub-vector e_i^k into d_{fsq} segments, creating a vast discrete space of $L^{d_{fsq}}$ possible representations. This exponential expansion of the codebook size ($|\mathcal{C}| = L^{d_{fsq}}$) enables near-lossless mapping of semantic information while maintaining discrete token structure. Formally:

$$\text{FSQ}(e_i^k) = R[(L-1)\sigma(\mathcal{T}_{in}(e_i^k))], \tag{3}$$

where the output $\text{FSQ}(e_i^k) \in 0, \cdots, L - 1^{d_{fsq}}$ constrains each dimension to integers between 0 and $L - 1$. The transformation $\mathcal{T}\text{in}(e_i^k) = W\text{in}e_i^k + b$ reduces dimensionality from d_L/K to d_fsq. This mapping requires optimizing only the linear transformation parameters, ensuring computational efficiency. To address the gradient-blocking nature of the rounding function, we implement the Straight-Through Estimator (STE) technique:

$$\begin{aligned}\text{FSQ}(e_i^k) &= (L-1)\sigma(\mathcal{T}_{in}(e_i^k)) + \\ &\text{sg}[R[(L-1)\sigma(\mathcal{T}_{in}(e_i^k))] - (L-1)\sigma(\mathcal{T}_{in}(e_i^k))],\end{aligned} \tag{4}$$

where sg[·] denotes the "stop gradient" operation that allows gradient flow through the non-differentiable rounding function. This quantization mechanism transforms user interaction histories into universal token sequences rather than dataset-specific item IDs—a crucial advantage that enables cross-domain generalization. By operating in a shared, discrete token space, our model can seamlessly transfer knowledge across recommendation domains, significantly enhancing zero-shot capabilities and cold-start performance compared to traditional ID-based approaches.

Optimization for Item Quantization. Training our quantization function requires a reconstruction mechanism capable of restoring the original semantic fidelity from discrete tokens. We implement a sophisticated multi-layer transformer decoder that leverages bidirectional attention to capture complex dependencies between quantized sub-vectors. This architecture choice is critical—unlike simpler feed-forward approaches, transformers excel at contextualizing relationships across the entire token sequence $[\hat{e}_i^0, \cdots, \hat{e}_i^{K-1}]$, enabling high-fidelity reconstruction of the original embedding $\hat{e}_i$.

Our optimization strategy employs L_1 loss, which prioritizes sparse, robust reconstructions while being less sensitive to outliers than squared-error alternatives. This enhances preservation of distinguishing semantic features during the quantization-reconstruction process. Formally, we minimize:

$$\begin{aligned}\mathcal{L}_{fsq} = \sum_{i=0}^{N-1} \|e_i - \text{Decoder}(\\ [\mathcal{T}_{out}(\hat{e}_i^0), \cdots, \mathcal{T}_{out}(\hat{e}_i^{K-1})])\|_1,\end{aligned} \tag{5}$$

where $\hat{e}_i^k$ represents the quantized representation derived from e_i^k (specifically, $\hat{e}i^k = \mathcal{T}_{in}(e_i^k)$), and $\mathcal{T}_{out}$ functions as a dimensionality-expansion transform that maps each compact token from the lower-dimensional space d_{fsq} back to the richer representational capacity of d_L/K.

2.2 Universal Recommendation Modeling

Our approach harnesses the transformative capabilities of autoregressive (AR) transformer architectures—models that have revolutionized zero-shot generalization across language (Brown, 2020), vision (Tian et al., 2024), and multimodal domains (Lu et al., 2022). By reformulating sequential recommendation as next-token prediction, we unlock the remarkable generalization potential of the autoregressive paradigm for cross-domain recommendation with minimal adaptation requirements while addressing two fundamental challenges:

i) Limitations of unidirectional attention. Since each item in our framework is represented by a token sequence, standard unidirectional attention constrains information flow between tokens belonging to the same item. This artificial barrier significantly hampers the model's ability to form coherent item representations and accurately model user preferences for next-item prediction.

ii) The information compression inherent in quantization. While our quantization approach creates a sufficiently expressive discrete space to uniquely represent items, the compression process inevitably sacrifices some semantic nuances from the original item descriptions. This information bottleneck, if left unaddressed, could degrade user preference modeling accuracy.

As depicted in Fig. 2 (ii), RecGPT implements two innovative architectural solutions - specifically designed to maximize information retention and flow - to overcome these fundamental challenges:

Bidirectional Attention for Item Tokens. To address the first challenge, RecGPT implements a hybrid attention mechanism combining unidirectional causal attention for modeling sequential item relationships with bidirectional attention for tokens within each item. This architectural innovation enables comprehensive information sharing among tokens representing the same item while preserving temporal causality necessary for next-item prediction by restricting cross-item attention to previously encountered items in the sequence.

Auxiliary Item Semantic Features. To avoid information loss from representation quantization, RecGPT integrates original textual features alongside learnable token embeddings within the transformer network. For each item i, there exists a feature sequence $\{e_i^0, \ldots, e_i^{K-1}\}$ where $e_i^k \in \mathbb{R}^{d_L/K}$. A linear layer aligns the dimension of e_i^k with the AR model's dimension d_{ar}. Since the feature and token sequences are equal in length, we derive two embeddings—auxiliary embedding $E_{\text{aux}} \in \mathbb{R}^{T \times d_{\text{ar}}}$ and token embedding $E_{\text{wte}} \in \mathbb{R}^{T \times d_{\text{ar}}}$—where T is the AR model's maximum input sequence length.

To maintain consistency between E_{aux} and E_{wte} and address optimization challenges, we apply Layer Normalization (LNorm$(\cdot)$) to both embeddings. The normalized outputs are then added to the positional embeddings $E_{\text{wpe}} \in \mathbb{R}^{T \times d_{\text{ar}}}$, producing the final representations $X \in \mathbb{R}^{T \times d_{\text{ar}}}$ fed into the AR model's transformer layers. This process is formally described as follows:

$$X = \text{LNorm}(E_{aux}) + \text{LNorm}(E_{wte}) + E_{wpe}. \quad (6)$$

2.2.1 Training Objective

During training, the goal of the autoregressive (AR) model is to maximize the likelihood of predicting the next token, specifically aiming to forecast the probability of the next token based on the preceding token sequence. The loss function is expressed as the negative log-likelihood loss:

$$\mathcal{L}_{\text{ar}} = -\sum_{t=0}^{T-1} \log P\left(Y_t \mid X_{<\lfloor \frac{t}{K} \rfloor \times K}\right), \quad (7)$$

where Y_t denotes the token at position t, and $X_{<\lfloor \frac{t}{K} \rfloor \times K}$ represents all embeddings preceding position $\lfloor \frac{t}{K} \rfloor \times K$. Here, $\lfloor \cdot \rfloor$ denotes the floor function. By incorporating the bidirectional attention, our model can compute the probabilities of the next K tokens during the training phase.

2.3 Efficient Item Token Decoder

To forecast the next item based on the predicted token sequence, we employ efficient search algorithms within the item token decoder (Fig. 2 (iii)).

Efficient Next-item Prediction. Firstly, RecGPT utilizes beam search to determine the top-n most likely token sequences. This strategy strikes a balance between efficiency and the size of the search space. Specially, as RecGPT is able to directly predict the next K tokens during the inference, we only need a single inference pass for each item's token sequence. Furthermore, beam search can be executed directly on the probabilities of these K tokens, thus eliminating the need to occupy excessive search space states that would otherwise consume memory allocated for inference.

Prefix Enhancement. Given that the number of candidate items is typically much smaller than the number of potential token sequences (*i.e.*, $L^{d_{\text{fsq}}}$), a straightforward approach is to further limit the token search space by leveraging the candidate item search space. To achieve this, we construct a Trie to store all token sequences associated with the candidate items. Consequently, when conducting beam

Table 1: Zero-shot performance of RecGPT v.s. Few-shot performance of baselines. Best and second-best are in bold and underlined. "*" indicates the improvement is statistically significant (*i.e.,* p-value < 0.05).

Data	Metric	GRU4Rec		GRU4RecF		Caser		BERT4Rec		FDSA		CL4SRec		DuoRec		ICLRec		MAERec		**RecGPT**	
		@3	@5	@3	@5	@3	@5	@3	@5	@3	@5	@3	@5	@3	@5	@3	@5	@3	@5	@3	@5
Cross-Domain in Amazon Dataset																					
Baby	Hit@1	.0019		.0019		.0025		.0024		.0025		.0024		.0025		.0010		.0016		**.0273***	
	Hit	.0051	.0090	.0060	.0091	.0062	.0099	.0061	.0096	.0064	.0099	.0060	.0093	.0061	.0103	.0046	.0082	.0055	.0090	**.0281***	**.0283***
	NDCG	.0036	.0052	.0043	.0056	.0046	.0061	.0045	.0060	.0047	.0061	.0045	.0059	.0046	.0063	.0031	.0045	.0038	.0052	**.0277***	**.0279***
Games	Hit@1	.0031		.0032		.0032		.0032		.0026		.0045		.0031		.0026		.0018		**.0364***	
	Hit	.0079	.0133	.0093	.0133	.0093	.0139	.0082	.0135	.0084	.0135	.0112	.0163	.0078	.0117	.0060	.0107	.0078	.0119	**.0374***	**.0376***
	NDCG	.0058	.0080	.0066	.0083	.0066	.0084	.0061	.0083	.0059	.0080	.0082	.0103	.0058	.0074	.0046	.0065	.0052	.0069	**.0370***	**.0371***
Office	Hit@1	.0021		.0019		.0021		.0021		.0021		.0021		.0022		.0021		.0020		**.0280***	
	Hit	.0042	.0059	.0043	.0053	.0040	.0059	.0031	.0049	.0042	.0061	.0043	.0058	.0040	.0056	.0038	.0049	.0036	.0047	**.0293***	**.0299***
	NDCG	.0033	.0040	.0033	.0037	.0032	.0040	.0027	.0034	.0033	.0041	.0034	.0040	.0032	.0039	.0031	.0035	.0029	.0034	**.0287***	**.0290***
Cross-Domain on Different Platforms																					
Yelp	Hit@1	.0011		.0008		.0007		.0013		.0009		.0021		.0019		.0003		.0029		**.0161***	
	Hit	.0023	.0036	.0022	.0039	.0026	.0040	.0037	.0054	.0025	.0038	.0056	.0083	.0043	.0065	.0027	.0045	.0079	.0119	**.0163***	**.0166***
	NDCG	.0018	.0023	.0016	.0023	.0018	.0024	.0026	.0033	.0018	.0023	.0040	.0051	.0033	.0042	.0016	.0024	.0057	.0074	**.0162***	**.0163***
Washington	Hit@1	.0012		.0016		.0010		.0027		.0016		.0032		.0024		.0017		.0041		**.0122***	
	Hit	.0030	.0043	.0037	.0056	.0033	.0046	.0073	.0114	.0038	.0053	.0085	.0127	.0067	.0105	.0043	.0069	.0103	**.0154**	**.0128***	.0130
	NDCG	.0022	.0028	.0028	.0036	.0023	.0028	.0053	.0070	.0029	.0035	.0063	.0079	.0049	.0064	.0032	.0042	.0076	.0097	**.0126***	**.0127***
Steam	Hit@1	.0396		.0484		.0121		.0757		.1022		.0681		.1011		.0014		.0727		**.1237***	
	Hit	.0675	.0830	.0737	.0881	.0350	.0538	.1002	.1161	.1231	**.1366**	.0986	.1177	.1223	.1351	.0433	.0678	.1018	.1172	**.1246***	.1253
	NDCG	.0557	.0621	.0630	.0689	.0254	.0331	.0898	.0964	.1143	.1232	.0858	.0936	.1134	.1187	.0257	.0357	.0897	.0960	**.1242***	**.1245***

search on tokens, we utilize the currently searched tokens as prefixes and employ the Trie for efficient prefix matching to identify all candidate items that begin with the existing tokens. This strategy significantly reduces the search space, thereby enhancing both the efficiency and accuracy of the search.

3 Evaluation

3.1 Experimental Setup

Datasets. For a comprehensive evaluation of RecGPT, we collect six public datasets from different recommendation scenarios for experimentation. The statistics and detailed descriptions of these datasets are presented in Appendix A.1.

Baselines & Implementation Details. RecGPT is compared with a diverse set of 9 baselines from different research streams. Appendix A.2 gives a thorough introduction to these methods. And Appendix A.3 elaborates the implementation details for our RecGPT framework and the baselines.

Evaluation Settings. We follow previous recommendation works (Qiu et al., 2022) to set our metrics, dataset division method, and other evaluation protocols. See Appendix A.4 for more details.

3.2 Cross-domain Zero-shot Recommendation

To demonstrate RecGPT's cross-domain zero-shot prediction capability, we evaluate it on six datasets and compare its performance against traditional sequence recommenders trained on 10% of the data, as traditional methods relying on ID representations lack zero-shot capabilities. Based on the results (Table 1), we have the following observations:

i) Superior Zero-shot Performance. RecGPT exhibits outstanding zero-shot prediction across diverse datasets, significantly surpassing traditional sequential recommenders trained on 10% of the data in nearly all metrics without fine-tuning on target test sets. This superior generalizability stems from our unified item tokenizer, which compresses semantic embeddings from multiple domains into unified token representations, enabling the model to learn universal sequence patterns. Additionally, extensive pre-training on large-scale data enhances the model's robustness and generative capabilities, effectively addressing zero-shot challenges.

ii) Enhanced Cross-Domain Adaptability. RecGPT excels in cross-domain scenarios by effectively generalizing from extensive pre-training without relying on ID representations. In contrast, traditional sequence recommenders depend heavily on ID embeddings and sufficient training data, resulting in limited effectiveness and suboptimal performance even with only 10% of the data. Although recent approaches incorporate additional text features (*e.g.*, GRU4RecF, FDSA) or self-supervised learning techniques (*e.g.*, CL4SRec, DuoRec, ICLRec, MAERec), they still fall short of RecGPT's superior performance. This underscores RecGPT's advantages in overcoming challenges inherent in cross-domain recommendations.

Additionally, the offline feature generation pipeline of RecGPT, which incorporates daily updates for active users and cold items, has been suc-

Table 2: Cold-start performance comparison of different recommendation models. Best and second-best results are in bold and underlined. "*" indicates the improvement is statistically significant (*i.e.,* p-value < 0.05).

Data	Metric	GRU4Rec		GRU4RecF		Caser		BERT4Rec		FDSA		CL4SRec		DuoRec		ICLRec		MAERec		**RecGPT**	
		@3	@5	@3	@5	@3	@5	@3	@5	@3	@5	@3	@5	@3	@5	@3	@5	@3	@5	@3	@5
Baby	Hit@1	.0030		.0030		.0029		.0030		.0064		.0092		.0099		.0011		.0062		**.0165***	
	Hit	.0090	.0123	.0080	.0122	.0086	.0124	.0085	.0120	.0130	.0168	.0159	.0193	.0164	**.0204**	.0057	.0092	.0124	.0160	**.0171***	.0172
	NDCG	.0063	.0077	.0058	.0075	.0061	.0077	.0061	.0075	.0101	.0117	.0130	.0144	.0136	.0153	.0038	.0052	.0098	.0112	**.0168***	**.0169***
Office	Hit@1	.0020		.0028		.0024		.0022		.0088		.0083		.0096		.0049		.0085		**.0188***	
	Hit	.0053	.0070	.0060	.0086	.0045	.0063	.0041	.0058	.0149	.0189	.0150	.0186	.0172	**.0207**	.0080	.0096	.0157	.0189	**.0201***	.0204
	NDCG	.0039	.0046	.0046	.0057	.0036	.0043	.0033	.0040	.0123	.0140	.0122	.0137	.0140	.0154	.0067	.0074	.0127	.0140	**.0195***	**.0197***
Yelp	Hit@1	.0050		.0049		.0032		.0023		.0061		.0057		.0063		.0003		.0057		**.0126***	
	Hit	.0120	.0164	.0125	.0180	.0084	.0129	.0069	.0107	.0142	.0219	**.0149**	**.0221**	.0147	.0220	.0059	.0102	.0133	.0196	.0128	.0131
	NDCG	.0090	.0108	.0092	.0115	.0061	.0080	.0049	.0065	.0107	.0139	.0109	.0138	.0111	**.0141**	.0035	.0053	.0101	.0127	**.0127***	.0128

cessfully implemented in a real-world industrial recommendation system, demonstrating its practical efficiency. The performance of RecGPT in industrial scenarios is detailed in Appendix A.5.

3.3 Cold-start Recommendation

The performance of RecGPT in cold-start scenarios, where users have minimal historical interaction records, is also examined. To realistically simulate these, we randomly truncate each user interaction sequence in the dataset to an item sequence of length 1 to 3, using the subsequent item as the prediction target. Baseline recommendation methods are allowed full training on the entire dataset with cold-start strategies, and are evaluated on the processed test set alongside the unfine-tuned RecGPT.

As shown in Table 2, RecGPT outperforms well-trained baselines across most metrics and datasets in cold-start settings. In these scenarios, traditional methods struggle to effectively capture historical information for modeling user preferences and behavior patterns due to limited interaction history. Consequently, even when trained on complete datasets, they fail to optimize adequately, negatively impacting their performance during testing. In contrast, RecGPT leverages pre-training on large-scale data, providing it with strong generative capabilities, enabling it to predict the next item a user is likely to engage with, even with minimal interaction history.

3.4 Ablation Study

In this section, we examine the effectiveness of RecGPT's key components by comparing ablated variants regarding their cross-domain zero-shot performance. Details of variants are as follows:

- **w/o FSQ** replaces the unified item tokenizer with randomly assigned tokens for downstream items.
- **w/o Bidir** removes the bidirectional attention for item token sequences and simply applies the global causal attention mechanism.
- **w/o Aux** removes the auxiliary semantic features.
- **w/o Prefix** removes the prefix enhancement that is applied during the inference phase.

Table 3: Ablation study results of RecGPT.

Variants	**Baby**		**Office**		**Yelp**	
	Hit@5	NDCG@5	Hit@5	NDCG@5	Hit@5	NDCG@5
w/o FSQ	0.0178	0.0177	0.0167	0.0166	0.0139	0.0138
w/o Bidir	0.0279	0.0275	0.0288	0.0283	0.0162	0.0158
w/o Aux	0.0191	0.0189	0.0205	0.0200	0.0075	0.0065
w/o Prefix	0.0282	0.0274	0.0281	0.0280	0.0162	0.0161
RecGPT	**0.0283**	**0.0279**	**0.0299**	**0.0290**	**0.0166**	**0.0163**

The results of the ablation study are presented in Table 3. The findings indicate the following:

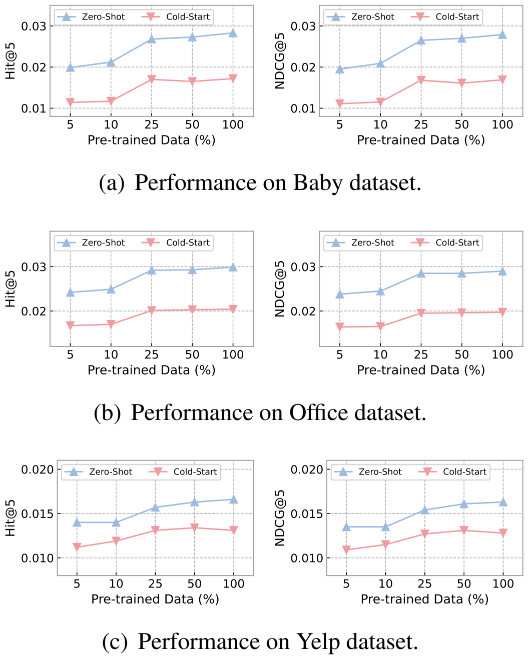
i) Component Contributions: Each key component in RecGPT significantly enhances the model's performance, with unified item tokenization playing a crucial role in enabling zero-shot prediction capability by compressing semantic item embeddings into token sequences, thereby improving generalization and robustness.

ii) FSQ Rewards: FSQ is crucial for the model's zero-shot prediction capability, as it compresses semantic embeddings into token sequences. This process significantly improves the model's generalization ability and robustness.

iii) Auxiliary Representation Benefits: The auxiliary item continuous representation allows for more effective utilization of semantic embeddings, minimizing information loss from vector quantization and further enhancing overall model performance.

3.5 Scaling Law Investigation

In this section, we investigate the applicability of the scaling law to RecGPT. Given that recommender systems operate in an efficiency-sensitive domain, it is particularly pertinent to examine how the volume of training data influences model performance while maintaining a relatively compact model size. To this end, we conduct experiments using five distinct versions of RecGPT, each differing in the amount of training data utilized (*i.e.,* 5%, 10%, 25%, 50%, and 100%). For each version,



(a) Performance on Baby dataset.

(b) Performance on Office dataset.

(c) Performance on Yelp dataset.

Figure 3: Performance *w.r.t.* the volume of training data.

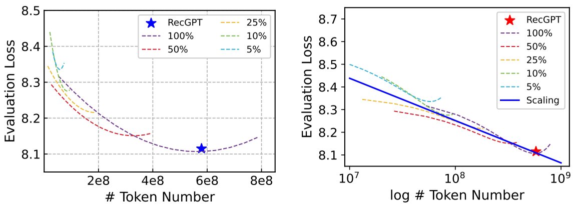


Figure 4: The scaling law of RecGPT.

training is performed on the respective training data until the loss on the evaluation set stabilizes, indicating the convergence. The evaluation results are presented in Figure 3, showcasing the zero-shot performance of the various model versions. Additionally, Figure 4 illustrates the correlation between the number of tokens and the evaluation loss for each model version throughout the training process. Key findings are summarized as follows:

i) Scalability from a Data Perspective. As illustrated in Figure 3, as the volume of data used for pre-training increases, RecGPT's zero-shot and cold-start performance progressively improves across various datasets, underscoring its scalability. It is noteworthy that the performance enhancement from RecGPT-10% to RecGPT-25% is significantly greater than the differences observed between other versions. This suggests the emergent ability (Wei et al., 2022) of RecGPT, highlighting the effectiveness of scaling up in enhancing its generalization capabilities.

ii) Adherence of RecGPT to the Scaling Law in Data Dimension. Recent studies (Touvron et al., 2023) suggest that, for achieving target performance, training a relatively smaller model on a larger dataset can be more advantageous, as this approach reduces inference costs, which is particularly relevant in recommendation. In Figure 4, we apply a power-law term to the losses associated with token counts. By fixing the model's parameter size, we can predict the losses after training on additional tokens. Notably, fitting the optimal losses of models trained on various data subsets yields a line that accurately predicts the final losses of RecGPT, indicated by a ⋆. Given that RecGPT utilizes a decoder-only architecture similar to LLMs, it demonstrates comparable scaling law characteristics while maintaining strong generalization capabilities. This suggests that access to more training data could further enhance its performance.

3.6 Comparison with Pre-trained Methods

In this section, we conduct a comparison of RecGPT with other pre-trained sequential recommenders, including S^3-Rec (Zhou et al., 2020), UniSRec (Hou et al., 2022), VQ-Rec (Hou et al., 2023), TIGER (Rajput et al., 2024), IDGenRec (Tan et al., 2024), and RecFormer (Li et al., 2023). Since S^3-Rec is ID-based, we train it using 10% of the training data, following the few-shot approach mentioned in Section 3.2.

Comparison with Text-Based Pretrained Models. The results in Figure 5 show that RecGPT outperforms other pre-trained sequential recommenders. Notably, S^3-Rec significantly surpasses traditional methods presented in Table 1, indicating that its pre-training tasks effectively enhance performance and mitigate the negative effects of limited training data. VQ-Rec ranks just below RecGPT, benefiting from quantization techniques that improve the generation of token sequences and enhance generalization capability. In contrast, our method employs a decoder-only architecture that follows the scaling law, supported by an order of magnitude more pre-training data (*i.e.*, ten million for RecGPT compared to one million for both UniSRec and VQ-Rec, and three million for RecFormer), resulting in superior performance.

In-Depth Comparison with VQ-Rec. To enable a more thorough comparison with existing vector quantization-based generative recommenders, we perform an in-depth analysis using VQ-Rec as a case study. Specifically, we retrain two variants based on RecGPT and VQ-Rec, referred to

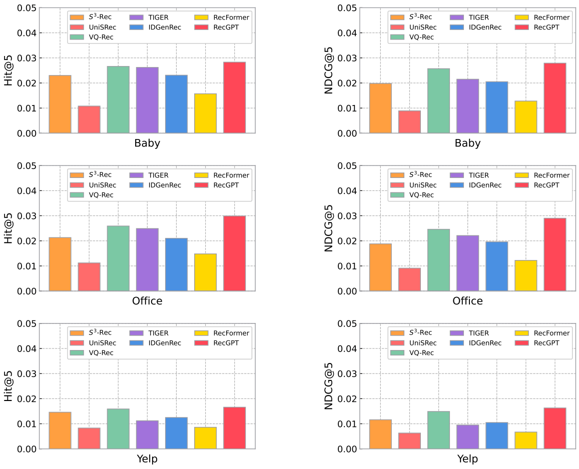


Figure 5: Comparison of pre-trained sequential models.

as RecGPT-10% and VQ-Rec (Re-trained). For RecGPT-10%, we reduce RecGPT's parameter size and randomly sample 10% of the original training dataset for retraining, yielding a training data scale comparable to VQ-Rec. For VQ-Rec (Re-trained), we train it on RecGPT's complete training dataset. As shown in Fig. 6, our results indicate that, despite a similar amount of training data, RecGPT-10% outperforms VQ-Rec. Moreover, the performance of VQ-Rec (Re-trained), trained on a larger dataset, declines, underscoring the advantages of RecGPT's decoder-only architecture and the generalization benefits from its autoregressive training objective.

4 Related Work

Sequential Recommendation. Sequential recommendation aims to predict users' next interacted items based on their historical interactions. Recent methods are based on recurrent neural networks (Li et al., 2017), convolutional neural networks (Tang and Wang, 2018), and graph neural networks (Ye et al., 2023). Notably, SASRec (Kang and McAuley, 2018) and BERT4Rec (Sun et al., 2019) introduced self-attention and Transformer architectures, significantly improving performance and scalability. To enhance generalization beyond ID-based embeddings (Yuan et al., 2023), recent works like UniSRec (Hou et al., 2022) and VQ-Rec (Hou et al., 2023) leverage multi-modal features for transferable recommendations. Additionally, large language models have been employed to create unified recommender systems based on text (Geng et al., 2022; Zhang et al., 2023). However, these methods lack zero-shot capabilities and depend heavily on sufficient training data. In contrast, our approach addresses these limitations, enabling effective predictions on new datasets.

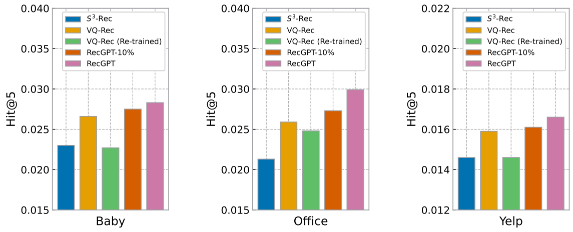


Figure 6: In-Depth Comparison with VQ-Rec.

Generative Recommendation. Generative recommendation is an emerging paradigm where each item is represented as discrete tokens, known as semantic IDs (Rajput et al., 2024). This tokenization enables nuanced and memory-efficient item representations by combining various discrete tokens to form a comprehensive feature space (Van Den Oord et al., 2017). Generative recommenders process sequences of item tokens as input and predict subsequent tokens in an autoregressive manner (Radford et al., 2019; Brown, 2020), which are then mapped back to recommended items for actionable insights. Pioneering work such as TIGER (Rajput et al., 2024) utilizes residual quantization to compute semantic codes, while later studies have enhanced quantization mechanisms (Petrov and Macdonald, 2023; Wang et al., 2024b) through methods like contrastive quantization (Zhu et al., 2024) and the development of learnable tokenizers (Wang et al., 2024a; Liu et al., 2024). In contrast, our work focuses on scaling laws within the generative recommendation framework to improve generalization.

5 Conclusion

This paper presents RecGPT, a foundation model designed to enhance the generalization of sequential recommendation systems, particularly in zero-shot scenarios. By utilizing a unified item tokenizer based on vector quantization, it effectively tackles challenges related to feature semantic diversity and interaction behavior variability across different domains. Its architecture, which combines a decoder-only transformer with bidirectional attention, allows the model to capture complex user behaviors and improve generalization capabilities. Experimental evaluations show RecGPT outperforms traditional ID-based recommendation methods across various datasets, highlighting its exceptional adaptability and generalization potential.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grants 624B2122.

6 Limitations

In real-world scenarios, items commonly have abundant modal information, including text, images, audio, and more. However, this work focuses on utilizing the textual information of items as their semantic representations and does not fully exploit the multimodal information available. While our approach can handle multimodal information by extending encoders corresponding to different modalities, the autoregressive modeling of multiple features may introduce new challenges. This represents a promising avenue for future exploration.

References

Tom B Brown. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Yongjun Chen, Zhiwei Liu, Jia Li, Julian McAuley, and Caiming Xiong. 2022. Intent contrastive learning for sequential recommendation. In *Proceedings of the ACM Web Conference 2022*, pages 2172–2182.

DeepSeek-AI. 2024. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Patrick Esser, Robin Rombach, and Bjorn Ommer. 2021. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883.

Hui Fang, Danning Zhang, Yiheng Shu, and Guibing Guo. 2020. Deep learning for sequential recommendation: Algorithms, influential factors, and evaluations. *ACM Transactions on Information Systems (TOIS)*, 39(1):1–42.

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*, pages 299–315.

Balázs Hidasi, Massimo Quadrana, Alexandros Karatzoglou, and Domonkos Tikk. 2016. Parallel recurrent neural network architectures for feature-rich session-based recommendations. In *Proceedings of the 10th ACM conference on recommender systems*, pages 241–248.

Yupeng Hou, Zhankui He, Julian McAuley, and Wayne Xin Zhao. 2023. Learning vector-quantized item representation for transferable sequential recommenders. In *Proceedings of the ACM Web Conference 2023*, pages 1162–1171.

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*.

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 *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 585–593.

Mohsen Jamali and Martin Ester. 2010. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 135–142.

Yangqin Jiang, Lianghao Xia, Wei Wei, Da Luo, Kangyi Lin, and Chao Huang. 2024a. Diffmm: Multi-modal diffusion model for recommendation. *arXiv preprint arXiv:2406.11781*.

Yangqin Jiang, Yuhao Yang, Lianghao Xia, Da Luo, Kangyi Lin, and Chao Huang. 2024b. Reclm: Recommendation instruction tuning. *arXiv preprint arXiv:2412.19302*.

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

Jiacheng Li, Jingbo Shang, and Julian McAuley. 2022. Uctopic: Unsupervised contrastive learning for phrase representations and topic mining. *arXiv preprint arXiv:2202.13469*.

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 *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1258–1267.

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*, pages 1419–1428.

Enze Liu, Bowen Zheng, Cheng Ling, Lantao Hu, Han Li, and Wayne Xin Zhao. 2024. End-to-end learnable item tokenization for generative recommendation. *arXiv preprint arXiv:2409.05546*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. In *NeurIPS*.

Jiasen Lu, Christopher Clark, Rowan Zellers, Roozbeh Mottaghi, and Aniruddha Kembhavi. 2022. Unified-io: A unified model for vision, language, and multi-modal tasks. In *The Eleventh International Conference on Learning Representations*.

Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. 2023. Finite scalar quantization: Vq-vae made simple. *arXiv preprint arXiv:2309.15505*.

Aleksandr V Petrov and Craig Macdonald. 2023. Generative sequential recommendation with gptrec. *arXiv preprint arXiv:2306.11114*.

Ruihong Qiu, Zi Huang, Hongzhi Yin, and Zijian Wang. 2022. Contrastive learning for representation degeneration problem in sequential recommendation. In *Proceedings of the fifteenth ACM international conference on web search and data mining*, pages 813–823.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, and 1 others. 2024. Recommender systems with generative retrieval. *Advances in Neural Information Processing Systems*, 36.

Xubin Ren, Lianghao Xia, Yuhao Yang, Wei Wei, Tianle Wang, Xuheng Cai, and Chao Huang. 2023. Sslrec: A self-supervised learning library for recommendation. *arXiv preprint arXiv:2308.05697*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pretraining for language understanding. *Advances in neural information processing systems*, 33:16857–16867.

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*, pages 1441–1450.

Juntao Tan, Shuyuan Xu, Wenyue Hua, Yingqiang Ge, Zelong Li, and Yongfeng Zhang. 2024. Idgenrec: Llm-recsys alignment with textual id learning. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 355–364.

Yong Kiam Tan, Xinxing Xu, and Yong Liu. 2016. Improved recurrent neural networks for session-based recommendations. In *Proceedings of the 1st workshop on deep learning for recommender systems*, pages 17–22.

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*, pages 565–573.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. 2024. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *arXiv preprint arXiv:2404.02905*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Aaron Van Den Oord, Oriol Vinyals, and 1 others. 2017. Neural discrete representation learning. *Advances in neural information processing systems*, 30.

Jianling Wang, Raphael Louca, Diane Hu, Caitlin Cellier, James Caverlee, and Liangjie Hong. 2020. Time to shop for valentine's day: Shopping occasions and sequential recommendation in e-commerce. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 645–653.

Wenjie Wang, Honghui Bao, Xinyu Lin, Jizhi Zhang, Yongqi Li, Fuli Feng, See-Kiong Ng, and Tat-Seng Chua. 2024a. Learnable tokenizer for llm-based generative recommendation. *arXiv preprint arXiv:2405.07314*.

Yidan Wang, Zhaochun Ren, Weiwei Sun, Jiyuan Yang, Zhixiang Liang, Xin Chen, Ruobing Xie, Su Yan, Xu Zhang, Pengjie Ren, and 1 others. 2024b. Enhanced generative recommendation via content and collaboration integration. *arXiv preprint arXiv:2403.18480*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, and 1 others. 2022. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. 2022. Contrastive learning for sequential recommendation. In *2022 IEEE 38th international conference on data engineering (ICDE)*, pages 1259–1273. IEEE.

An Yan, Zhankui He, Jiacheng Li, Tianyang Zhang, and Julian McAuley. 2023. Personalized showcases: Generating multi-modal explanations for recommendations. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2251–2255.

Zhilin Yang. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

Yaowen Ye, Lianghao Xia, and Chao Huang. 2023. Graph masked autoencoder for sequential recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 321–330.

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 *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2639–2649.

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*.

Tingting Zhang, Pengpeng Zhao, Yanchi Liu, Victor S Sheng, Jiajie Xu, Deqing Wang, Guanfeng Liu, Xiaofang Zhou, and 1 others. 2019. Feature-level deeper self-attention network for sequential recommendation. In *IJCAI*, pages 4320–4326.

Weizhi Zhang, Yuanchen Bei, Liangwei Yang, Henry Peng Zou, Peilin Zhou, Aiwei Liu, Yinghui Li, Hao Chen, Wang, and 1 others. 2025. Cold-start recommendation towards the era of large language models (llms): A comprehensive survey and roadmap. *arXiv preprint arXiv:2501.01945*.

Chuang Zhao, Hongke Zhao, Ming He, Jian Zhang, and Jianping Fan. 2023. Cross-domain recommendation via user interest alignment. In *WWW*, pages 887–896.

Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Yushuo Chen, Xingyu Pan, Kaiyuan Li, Yujie Lu, Hui Wang, Changxin Tian, and 1 others. 2021. Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms. In *proceedings of the 30th acm international conference on information & knowledge management*, pages 4653–4664.

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*, pages 1893–1902.

Jieming Zhu, Mengqun Jin, Qijiong Liu, Zexuan Qiu, Zhenhua Dong, and Xiu Li. 2024. Cost: Contrastive quantization based semantic tokenization for generative recommendation. In *Proceedings of the 18th ACM Conference on Recommender Systems*, pages 969–974.

A Appendix

A.1 Datasets

The statistics of the experimental datasets are shown in Table 4. Below presents the detailed description for these datasets.

(1) **Pre-trained datasets.** We select eleven categories from Amazon review datasets (Hou et al., 2024), *"All Beauty"*, *"Books"*, *"Clothing Shoes and Jewelry"*, *"Electronics"*, *"Health and Household"*, *"Kindle Store"*, *"Home and Kitchen"*, *"Magazine Subscriptions"*, *"Movies and TV"*, *"Cell Phones and Accessories"*, and *"Sports and Outdoors"*, as the datasets for pre-training.

(2) **Evaluation datasets.** We select three other categories from Amazon review datasets (Hou et al., 2024), *"Amazon Fashion"*, *"Musical Instruments"*, and *"Industrial and Scientific"*, as the datasets for evaluation during the pre-training.

(3) **Test datasets.** To thoroughly evaluate the cross-domain zero-shot capability of RecGPT, we select three categories (*i.e., "Baby Products"*, *"Video Games"*, and *"Office Products"*) from Amazon review dataset (Hou et al., 2024) with the same recommendation scenarios of the pre-training data. Furthermore, we incorporate three additional datasets from different recommendation scenarios. **Yelp**[1] is a commonly-used dataset contains user ratings on business venues. **Washington** (Li et al., 2022; Yan et al., 2023) is a dataset that includes review information from Google Maps (*e.g.,* ratings, text, and images) along with relevant business metadata from locations across Washington, the United States. **Steam** (Kang and McAuley, 2018) is a dataset crawled from *Steam*, a large online video game distribution platform.

A.2 Compared Methods

For a comprehensive evaluation, we thoroughly compare our RecGPT with a diverse set of baselines derived from different research streams.

- **GRU4Rec** (Tan et al., 2016): It utilizes a GRU model to encode interaction sequences and integrates a ranking-based loss function specifically for session-based recommendation.

- **GRU4RecF** (Hidasi et al., 2016): It proposes a session model based on RNN, leveraging deep learning to combine user clicks and item features to enhance the performance of recommenders.

Table 4: Statistics of the experimental datasets.

Datasets	#Users	#Items	#Interactions	#Avg.Length
Pre-trained	12,472,073	15,491,643	131,657,450	10.56
- Beauty	1,464	6,570	13,679	9.34
- Books	1,091,587	2,978,216	13,859,969	12.69
- Clothing	3,088,673	4,875,707	30,245,204	9.79
- Electronics	1,692,840	1,128,480	16,248,100	9.59
- Health	828,935	518,044	7,598,392	9.16
- Kindle	902,107	1,221,419	16,242,839	18.01
- Kitchen	3,096,330	2,742,128	30,758,013	9.93
- Magazine	380	865	2,679	7.05
- Movies	653,846	569,357	7,622,246	11.65
- Phones	547,327	628,147	4,103,048	7.49
- Sports	568,584	822,710	4,963,281	8.73
Evaluation	184,674	377,186	1,615,405	8.75
- Fashion	13,942	76,873	104,115	7.46
- Instruments	76,012	116,447	711,607	9.36
- Scientific	94,720	183,866	799,683	8.44
Baby	184,851	123,537	1,551,060	8.39
Games	117,742	83,137	1,030,529	8.75
Office	333,744	363,786	2,735,472	8.19
Yelp	287,116	148,523	4,392,168	15.29
Washington	625,428	120,080	12,382,314	19.79
Steam	334,594	15,066	4,214,640	12.59

- **Caser** (Tang and Wang, 2018): This approach converts users' interacted item sequences into image-like 2D representations within temporal and latent dimensions, and utilizes convolutional filters to capture sequential patterns.

- **BERT4Rec** (Sun et al., 2019): The Cloze task has been incorporated into sequential recommendation, utilizing a bidirectional attentive encoder.

- **FDSA** (Zhang et al., 2019): It aims to identify patterns of item and feature transitions through the use of self-attentive networks.

- **CL4SRec** (Xie et al., 2022): This approach empowers recommendation through various sequence-level augmentation methods, such as item cropping, masking, and reordering.

- **DuoRec** (Qiu et al., 2022): This work explores the problem of representation degeneration in sequential recommendation and proposes solutions grounded in contrastive learning techniques.

- **ICLRec** (Chen et al., 2022): It enhances sequential recommendation by performing clustering and contrastive learning on user intentions to improve the quality of recommendations.

- **MAERec** (Ye et al., 2023): This work improves recommenders by dynamically distilling global item transitional information for self-supervised augmentation, overcoming label scarcity and data noise in sequential recommendation tasks.

[1] https://www.yelp.com/dataset

A.3 Implementation Details

We implement our approach utilizing the Transformer library from Hugging Face (https://huggingface.co). Each item's semantic embedding is segmented into four parts, meaning that each item is represented by four tokens. The maximum input length T of the model is 1,024, with a hidden dimension d_{ar} of 768, meaning the size of the position embedding E_{wpe} is $\mathbb{R}^{1025\times768}$. For the FSQ quantizer, $d_{fsq} = 5$, with each corresponding dimension L being 8, 8, 8, 6, and 5. This allows it to discretize each sub-vector of the itme embedding into a space of 15,360 tokens (*i.e.*, $8 \times 8 \times 8 \times 6 \times 5 = 15360$). Therefore, the model's token embedding table is with the shape of $\mathbb{R}^{15360\times768}$. For the multi-layer decoder used in the model, we implemented the transformer from GPT-2 with a total of 3 layers.

For the baseline methods, we employ two open-source recommendation libraries, RecBole (Zhao et al., 2021) and SSLRec (Ren et al., 2023). To ensure a fair comparison, we optimize all methods using the Adam optimizer and conduct a thorough search for the hyperparameters of each compared method. We also implement early stopping with a patience of 10 epochs to mitigate overfitting.

Regarding the resources required for model training, we primarily conducted experiments on four A100 40G GPUs; however, it was also tested that a single 3090 24G GPU can complete all experiments (with the trade-off of longer training times and smaller batch size settings).

A.4 Evaluation Settings

To evaluate the performance of the next item prediction task, we utilize two commonly employed metrics: Hit@N and NDCG@N, with N set to 1, 3, and 5. For the six test datasets, we partition them in a 9:1 ratio, meaning that 90% of the user interaction sequences in each dataset are designated as training data (for training traditional recommenders), while the remaining 10% serve as test data. We rank the ground-truth item of each sequence against all other items for evaluation on the test data, and report the average score across all test users.

A.5 Performance in Industrial Scenarios

To evaluate the efficacy of our methodology in practical environments, we further systematically evaluate RecGPT alongside several competitive baselines on a large-scale dataset derived from actual industrial data, referred to here as "Industrial". The configurations of the baseline approaches are aligned with those outlined in Section 3.2.

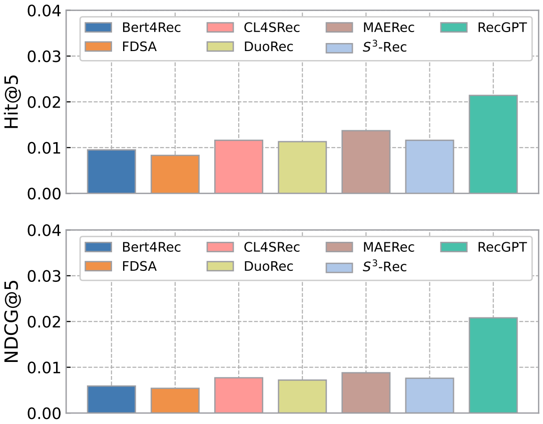


Figure 7: Performance on Industrial Dataset.

This "Industrial" dataset is sourced from a prominent online content platform (name omitted for anonymity), serving millions of users. It pertains to news content and encompasses various data points, including user identities, accessed news articles, article titles, timestamps, and additional relevant information. The dataset comprises 163,385 users, 455,372 items, and 999,140 interaction records.

As demonstrated in Figure 7, the proposed RecGPT outperforms all baseline methods on the Industrial dataset, demonstrating its robust generalization capability in addressing cross-domain tasks within real-world settings. This superior performance can be attributed to the extensive data accumulated during the training phase, which enables the model to learn diverse user preference patterns and effectively adapt to previously unexplored domains.

A.6 In-Depth Comparison with VQ-Rec (Continued)

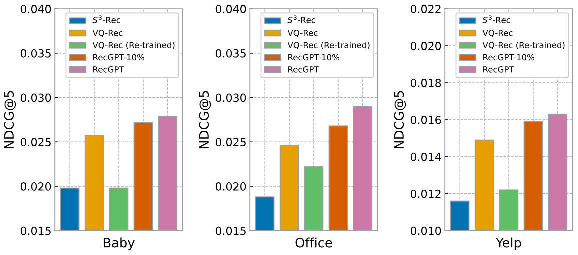


Figure 8: In-Depth Comparison with VQ-Rec (Continued).

A.7 Insights from Industrial Application

In our application of RecGPT within real industrial contexts, we discovered that foundation models for recommendation systems have not yet emerged as the ultimate solution for a wide range of challenges, as large language models (LLMs) have in natural language processing (NLP). The inherent complexity of human preferences, coupled with the information explosion in modern recommendation contexts, necessitates a diverse array of input features that exhibit a high degree of heterogeneity, including numerical data, text, images, and audio. From a technical standpoint, while LLMs have primarily unified foundational models at the text level, the processing and integration of information from various modalities remains an active area of research.

The features generated by RecGPT can serve as valuable auxiliary information to enhance the performance of recommendation systems in specific areas, such as addressing user cold-start situations. However, certain features, such as fundamental user profile information, remain indispensable. As related technologies continue to advance, foundation models for recommendation systems hold considerable potential for further development, aiming to process all relevant features and leverage the scaling laws and generalization capabilities inherent in foundation models. This approach aspires to provide a comprehensive solution within the recommendation system domain, with our RecGPT representing an initial effort inspired by the success of LLMs.