TSTRec: Tree-Structured Semantic Tokenization with Topic-Guided Session Compression for Generative Recommendation

Anonymous ACL submission

Abstract

Most existing generative recommenders construct discrete tokens from item attributes and collaborative signals without leveraging the rich world knowledge of large language models (LLMs), limiting semantic expressiveness. These methods rely on linear codebook structures further entangles multi-intent semantics. When processing user sequences, hard truncation strategy discards early interactions that contain long-term interests while retaining noise such as accidental clicks. Moreover, a high hallucination rate of generative retrieval produces many invalid candidates. To address these issues, we propose TSTRec, a framework combining Tree-Structured Residual Quantized Variational AutoEncoder (TS-RQVAE) with Topic-Guided Session Compression (TGSC). TS-RQVAE explicitly separates multi-intent item semantics with LLM topic alignment, achieving lightweight knowledge transfer from LLMs. Furthermore,TGSC expands user interest coverage via semantic clustering while simultaneously denoising the sequence. Experiments on both industrial and public datasets show that TSTRec achieves over 5% improvement in recommendation metrics compared to baseline methods, while significantly alleviating the hallucination, demonstrating the effectiveness of our method.

1 Introduction

In recommendation systems, traditional approaches typically rely on the discriminative paradigm, where items are mapped to learnable embeddings via interaction data (Klenitskiy and Vasilev, 2023; Hidasi and Karatzoglou, 2018; Hidasi et al., 2016; Zhou et al., 2020). This paradigm lacks the capacity to model fine-grained semantic structures of items. Recently, Semantic IDs have emerged as an effective method (Wu et al., 2024). This approach employs a tokenizer to map an item into a hierarchical sequence of discrete tokens, and uses a generative model to predict the token sequence of the target item in an autoregressive manner, thereby reformulating recommendation as a sequence generation problem (Geng et al., 2022; Bao et al., 2023; Geng et al., 2023; Wang et al., 2023). In this emerging paradigm, the representation quality of the semantic tokenizer and the concentration of the input sequence jointly determine recommendation performance (Deldjoo et al., 2024).

Despite rapid progress, the accuracy and reliability of generative recommenders in practice are still hindered by two core bottlenecks on the tokenizer and sequence aspects: (i) knowledge-isolated and semantically entangled tokenization, and (ii) the discarding of long-term user interests due to hard truncation strategies, coupled with vulnerability to noise interference. Crucially, both of these limitations greatly increase the risk of hallucination. We elaborate on these issues in detail below.

Most existing semantic tokenization methods suffer from knowledge isolation. They rely solely on local interaction data (Wang et al., 2024b; Zhu et al., 2024) while neglecting the rich world knowledge embedded in LLMs. Consequently, they fail to perceive the fine-grained semantic information of items. Worse, most tokenizers (Rajput et al., 2023; Zheng et al., 2024; Pang et al., 2025) adopt a linearly stacked codebook. However, items inherently exhibit multi-intent characteristics in reality, which means that different users have different intentions for clicking the same item. These tokenizers force such multi-intent items through a single residual chain. Quantization errors from heterogeneous semantic aspects interfere with each other, severely compromising the semantic purity of token representations.

Constrained by computational budgets, the prevalent hard truncation strategy simultaneously discards early interactions that encode long-term interests and retains noisy behaviors such as accidental clicks within the recent window. The prefer-

ence signal is distorted from both ends, leading the recommender to overfit to short-term noise while losing access to the user's stable intent.

Furthermore, the autoregressive generation process is susceptible to the hallucination problem, where the model decodes invalid token sequences that fail to map to any existing item. This issue is severely aggravated by the combination of semantically entangled item IDs and noise-corrupted sequence contexts, ultimately wasting valuable retrieval slots. Existing post-hoc fixes like prefix tree-constrained decoding (De Cao et al., 2021; Hua et al., 2023) incur huge memory overheads without improving the underlying representations.

To systematically address these challenges, we propose TSTRec, a tree-structured semantic tokenization framework with topic-guided session compression for generative recommendation. TSTRec overcomes the tokenizer bottleneck with a Tree-Structured Residual Quantized Variational AutoEncoder (TS-RQVAE) whose root-to-leaf paths are explicitly aligned with LLM-generated multi-granularity topics, so that distinct semantic aspects are quantized in parallel rather than entangled along a single chain. In addition, TSTRec tackles the sequence bottleneck with Topic-Guided Session Compression (TGSC), which simultaneously preserves long-term interests and suppresses noise via semantic clustering. This makes more effective use of a constrained sequence-length budget to capture richer behavioral signals while denoising the user interactions. Crucially, we find that the structural characteristics of the tree-structured codebook and the denoising capabilities of TGSC can alleviate hallucination. Experiments on a large-scale industrial dataset and the public dataset demonstrate that TSTRec consistently outperforms strong baselines on multiple recommendation metrics, and significantly reduces the hallucination rate without relying on prefix-tree masking. Our contributions are summarized as follows:

- To address knowledge isolation and semantic entanglement, we introduce TS-RQVAE to explicitly disentangle the multi-intent semantics of items and enable lightweight transfer of LLM world knowledge to the tokenizer.
- To deal with the problem of the hard truncation strategy, we design TGSC, which captures richer behavioral signals within a limited sequence length budget through semantic clustering, while denoising user interactions.
- Extensive experiments on industrial-scale and public datasets show that TSTRec improves recommendation metrics by more than 5%, consistently outperforming strong baselines. We find that TSTRec can significantly alleviate the hallucination phenomenon.

2 Related Work

Generative Recommendation via Semantic Tokenization. Generative recommendation reformulates retrieval as an autoregressive sequence generation problem. Early methods pioneered this paradigm by constructing discrete semantic IDs primarily from item content. The approaches like TIGER (Rajput et al., 2023) and RecForest (Feng et al., 2022) quantize item embeddings into hierarchical tokens using residual quantization or clustering, while IDGenRec (Tan et al., 2024) uses LLMs to generate purely textual identifiers. Despite their effectiveness, these early methods rely solely on isolated content features, limiting their ability to capture complex relational patterns among items.

Integration of Collaborative and Multimodal Signals. To overcome this limitation, subsequent research has focused on enriching the codebook with broader interactions. A major line of work injects collaborative filtering (CF) signals into the tokenization process via regularization, contrastive learning, or dual-stream architectures (e.g., LC-Rec (Zheng et al., 2024), LETTER (Wang et al., 2024a), CoST (Zhu et al., 2024), EAGER (Wang et al., 2024b), TokenRec (Qu et al., 2025)). Other advancements extend tokenization to incorporate multimodal features (Xu et al., 2026) or align the tokenizer directly with downstream recommendation objectives through unified end-to-end training frameworks (Liu et al., 2025; Deng et al., 2025).

Although collaborative or multimodal signals are injected, they are quantized along a single shared residual chain or within weakly structured clusters, leaving the multi-intent entanglement problem unresolved. Furthermore, user sequences are processed with hard truncation strategy that simultaneously discard informative long-term signals and retain noisy short-term interactions.

3 Methodology

We propose TSTRec, a tree-structured semantic tokenization framework for generative sequential recommendation. As shown in Fig. 1, the framework consists of three core modules: (1) a Multi-

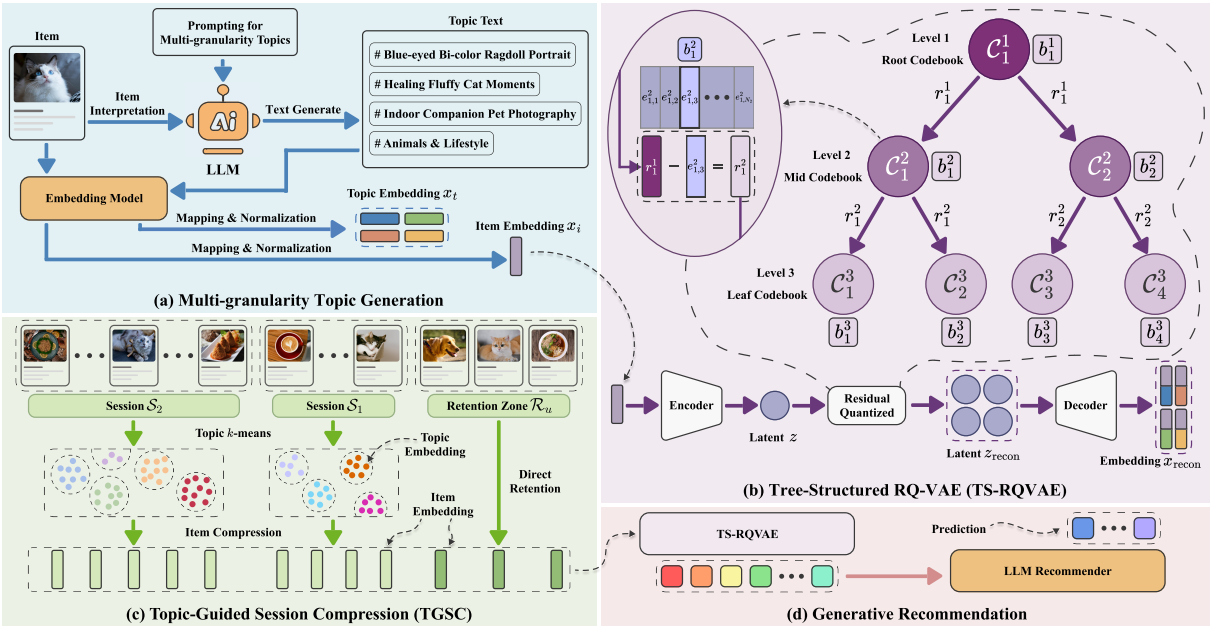

Figure 1: Overview of the TSTRec Framework. (a) Multi-granularity Topic Generation leverages LLM prompting to generate hierarchical and maps discrete tokens into continuous topic and item embeddings. (b) Tree-Structured Residual Quantized VAE (TS-RQVAE) adopts a hierarchical residual coding strategy to compress item into multi-level quantized latents for semantic representation learning. (c) Topic-Guided Session Compression (TGSC) compresses user interaction sequences through topic-guided semantic clustering to retain long-term interests and suppress noise. (d) Generative Recommendation uses the resulting sequence of discrete tokens to predict.

Granularity Topic Generator that leverages the world knowledge of LLMs to extract multi-level semantic signals for each item; (2) a Tree-Structured Residual Quantized Variational AutoEncoder (TS-RQVAE) that discretizes item embeddings into structured semantic IDs through hierarchical codebooks; and (3) Topic-Guided Session Compression (TGSC), a semantically-aware adaptive compression strategy applied during data preprocessing to condense user interaction sequences. We describe each module in detail below.

3.1 Problem Formulation

Let $\mathcal{U} = \{u_1, \ldots, u_{N_u}\}$ denote the user set and $\mathcal{I} = \{i_1, \ldots, i_{N_i}\}$ denote the item set, where N_u and N_i are the total number of users and items, respectively. Each user $u \in \mathcal{U}$ has a chronologically ordered interaction sequence $\mathcal{S}_u = (i_1^u, i_2^u, \ldots, i_T^u)$, where $i_T^u \in \mathcal{I}$ denotes the item that user u interacts with at timestamp T. Each item $i \in \mathcal{I}$ is associated with a d-dimensional embedding $\boldsymbol{x}_i \in \mathbb{R}^d$, encoded from the content information of the item.

Given a user's historical interaction sequence $\mathcal{S}_u$, sequential recommendation aims to predict the next item i_{T+1}^u that the user is likely to interact with. Formally, we aim to learn a parametric function $f(\cdot)$ that models the conditional probability:

$$P\left(i_{T+1}^u \mid \mathcal{S}_u\right) = f(\mathcal{S}_u; \Theta), \tag{1}$$

where Θ denotes the model parameters.

3.2 Multi-Granularity Topic Generator

To enrich item representations with multi-granularity semantic information, we use LLMs to generate hierarchical topic descriptions. Specifically, given the content of an item i, we prompt the LLM to produce multiple topic texts to capture diverse semantic aspects and abstraction levels.

Each generated topic text is encoded into a dense vector representation, forming a topic embedding set $\{\boldsymbol{x}_t^1, \boldsymbol{x}_t^2, \ldots, \boldsymbol{x}_t^M\}$, where M denotes the number of generated topics. We establish a direct correspondence between topics and the codebook structure: each topic maps to a specific root-to-leaf path in the tree-structured codebook, ensuring that each path explicitly corresponds to one semantic aspect, enabling a multi-path reconstruction process.

3.3 Tree-Structured Tokenizer

Items on real-world platforms inherently exhibit multi-intent characteristics. When these semantically independent aspects are forced into the same

residual chain, quantization errors from different directions interfere across layers, leading to semantic entanglement and degraded token purity. To address this problem, we propose TS-RQVAE, whose architecture explicitly mirrors the hierarchical topic structure generated by the LLM. By assigning distinct semantic aspects to separate branches of the tree-structured codebook, TS-RQVAE decouples the multi-intent semantics of items and enables each quantization path to specialize in a coherent semantic aspect.

The core of TS-RQVAE is a tree-structured hierarchical codebook that replaces the conventional linear codebook stack. We organize codebooks into a directed tree with K levels, where each node corresponds to an independent codebook. Every root-to-leaf path defines a quantization channel aligned with a specific semantic topic. We denote the i-th codebook at level k as $\mathcal{C}_i^k = \{\boldsymbol{e}_{i,1}^k, \boldsymbol{e}_{i,2}^k, \ldots, \boldsymbol{e}_{i,N_k}^k\}$, where N_k is the codebook size, and each codeword $\boldsymbol{e}_{i,j}^k \in \mathbb{R}^{d_l}$ lies in a d_l-dimensional latent space. The branching structure allows parallel paths to independently encode distinct semantic aspects without cross-contamination, while the hierarchical depth within each path preserves coarse-to-fine residual refinement.

Given an item embedding $\boldsymbol{x} \in \mathbb{R}^d$ as input, we project it into a lower-dimensional latent space via an MLP encoder:

$$\boldsymbol{z} = \text{Encoder}(\boldsymbol{x}) \in \mathbb{R}^{d_l}, \tag{2}$$

where d_l denotes the dimension of the latent space.

Residual quantization is performed in a top-down manner along the tree. The process begins at the root node ($k = 1$) with the initial residual $\boldsymbol{r}^1 = \boldsymbol{z}$. At level k, for the currently active codebook $\mathcal{C}_i^k$, the codeword closest to the residual is selected via cosine similarity:

$$b_i^k = \underset{j \in [1, N_k]}{\arg\max} \cos\left(\boldsymbol{r}_i^k, \boldsymbol{e}_{i,j}^k\right), \tag{3}$$

where $\boldsymbol{r}_i^k$ is the residual vector associated with node $\mathcal{C}_i^k$. We compute its projection component $\boldsymbol{q}_i^k$ onto the current residual direction:

$$\boldsymbol{q}_i^k = \frac{\boldsymbol{e}_{i,b_i^k}^{k\top} \boldsymbol{r}_i^k}{\|\boldsymbol{r}_i^k\|_2^2} \cdot \boldsymbol{r}_i^k. \tag{4}$$

The projection component is subtracted from the parent node's residual to produce the residual for the next level, which propagates to child nodes:

$$\boldsymbol{r}_{\text{child}(i)}^{k+1} = \boldsymbol{r}_i^k - \boldsymbol{q}_i^k, \tag{5}$$

where $\boldsymbol{r}_{\text{child}(i)}^{k+1}$ denotes the residual vector passed to the child nodes of node i at level $k+1$. At each branching point, the residual is duplicated to all child nodes, allowing parallel paths to independently quantize their respective semantic aspects from the shared residual.

For a complete root-to-leaf path p^m ($m = 1, \ldots, M$), the reconstructed latent representation is obtained by aggregating the projection components along the path, which is then mapped back to the original space by the decoder to recover both the item embedding and the fine-grained topic embedding assigned to that path:

$$\hat{\boldsymbol{y}}^m = \text{Decoder}(\sum_{k=1}^{K} \boldsymbol{q}_{n_k^m}^k) \in \mathbb{R}^{2d}, \tag{6}$$

where n_k^m is the node index traversed by path p^m at level k. The reconstruction loss measures the fidelity between the decoder output and the concatenated target representation:

$$\mathcal{L}_{\text{recon}}^m = \|\hat{\boldsymbol{y}}^m - [\boldsymbol{x}_i; \boldsymbol{x}_t^m]\|_2^2, \tag{7}$$

where $[\boldsymbol{x}_i; \boldsymbol{x}_t^m] \in \mathbb{R}^{2d}$ denotes the concatenation of the item embedding $\boldsymbol{x}_i$ and the topic embedding $\boldsymbol{x}_t^m$ associated with the m-th path. By reconstructing two embeddings simultaneously, this dual-target objective forces each path to specialize in the semantic aspect defined by its aligned topic, effectively transferring LLM knowledge into the discrete codebook structure while retaining intrinsic item information.

The quantization loss is defined as follows:

$$\mathcal{L}_{\text{rqvae}} = \sum_{i=1}^{N_c} \sum_{k=1}^{K} \Big[\|\text{sg}(\boldsymbol{r}_i^k) - \boldsymbol{e}_i^k\|_2^2 + \beta \|\boldsymbol{r}_i^k - \text{sg}(\boldsymbol{e}_i^k)\|_2^2 \Big], \tag{8}$$

where $\text{sg}(\cdot)$ denotes the stop-gradient operator and β is a hyperparameter balancing the commitment loss. The overall training objective is:

$$\mathcal{L}_{\text{total}} = \frac{1}{M} \sum_{m=1}^{M} \mathcal{L}_{\text{recon}}^m + \lambda \cdot \mathcal{L}_{\text{rqvae}}, \tag{9}$$

where λ is the weight coefficient for the quantization loss. Through this topic-aligned training strategy, TS-RQVAE effectively encodes topic information to utilize LLM world knowledge.

3.4 Topic-Guided Session Compression

To simultaneously preserve long-term interests and suppress noise within a bounded sequence budget, we propose Topic-Guided Session Compression (TGSC), a semantically-aware adaptive compression method.

Given a user interaction sequence $\mathcal{S}_u$, we retain the most recent n_r items at the end of the sequence as the recent retention zone without compression:

$$\mathcal{R}_u = (i^u_{T-n_r+1}, \ldots, i^u_T). \tag{10}$$

This design leverages the recency effect of user behavior: the most recent interactions carry the high informational value for next-item prediction, and maintaining their atomic granularity facilitates capturing the user's latest fine-grained interest signals.

For the remaining distant interactions $\mathcal{S}^{\text{far}}_u = (i^u_1, \ldots, i^u_{T-n_r})$, we split them into p sessions $\{S_1, S_2, \ldots, S_p\}$ using expanding time windows. The window size of the p-th session is defined as:

$$\Delta_p = \Delta_0 \cdot p, \quad p = 1, 2, \ldots \tag{11}$$

where Δ_0 is the base window step. The expanding window design ensures that sessions closer to the present have smaller time windows, preserving higher resolution, while distant sessions have larger windows and are aggregated at coarser granularity.

For each session S_p containing $n_p = |S_p|$ items, we retrieve all topics associated with each item and perform k-means clustering on the $M \cdot n_p$ topic embeddings in $\mathcal{S}_p$, producing $C_p = \lfloor \gamma \cdot n_p \rfloor$ topic clusters, where $\gamma \in (0, 1)$ is the compression ratio hyperparameter and $\lfloor \cdot \rfloor$ denotes rounding the value down to the nearest integer.

For each cluster $c \in \{1, \ldots, C_p\}$, we identify the set of items $\mathcal{I}^p_c$ whose topics belong to that cluster, and perform average pooling on their embeddings to obtain the compressed representation:

$$\bar{\boldsymbol{x}}^p_c = \frac{1}{|\mathcal{I}^p_c|} \sum_{i \in \mathcal{I}^p_c} \boldsymbol{x}_i. \tag{12}$$

Adopting topic-guided clustering as the compression principle offers two advantages. Items within the same topic cluster share coherent semantic aspects; thus, average pooling preserves semantic diversity while removing redundancy from repeated exposure to similar content. Furthermore, the clustering process inherently filters out isolated noise interactions: incidental behaviors that align with no coherent interest pattern are absorbed into larger, semantically coherent clusters, effectively achieving denoising as a byproduct of compression.

Let $\bar{\mathcal{X}}_p = (\bar{\boldsymbol{x}}^p_1, ..., \bar{\boldsymbol{x}}^p_{C_p})$ denote the ordered compressed representations for session S_p, and $\mathcal{X}_{\mathcal{R}_u} = (\boldsymbol{x}_{T-n_r+1}, ..., \boldsymbol{x}_T)$ denote the raw embeddings of the recent retention zone. The compressed sequence for user u is formed by concatenating these components in chronological order from distant to recent:

$$\tilde{\mathcal{S}}_u = \big(\bar{\mathcal{X}}_p, \ldots, \bar{\mathcal{X}}_1, \mathcal{X}_{\mathcal{R}_u}\big). \tag{13}$$

This design preserves the long-term interests of the user under a fixed sequence length budget while suppressing noise through semantic aggregation.

4 Experiments

4.1 Experimental Setup

Datasets and implementation. We evaluate TSTRec on a large-scale industrial dataset and the public MIND-small dataset (Wu et al., 2020). Because the industrial data contains rich text and long user sequences, we experiment with sequence lengths of $\{30, 50, 100\}$ and set $M = 4$ to capture diverse topics. We set the codebook size to 64 to maintain a fair semantic combinatorial space against the 1024^3 baseline. In contrast, since the MIND-small dataset provides sparse text and shorter sequences, we evaluate on lengths of $\{20, 30\}$ and simplify the tree to $M = 2$ paths. This adjustment prevents semantic overlap and keeps the codewords easily distinguishable. For more details on datasets and experimental implementation, please refer to Appendix A.1 and A.2. The detailed prompt template used to instruct the LLM for topic generation on the MIND dataset is provided in Appendix C.

Baselines and metrics. We compare our model against classic sequential recommenders (SASRec, BERT4Rec) and generative baselines (TIGER, LETTER). To ensure a fair structural comparison on the industrial and MIND-small dataset, we also introduce TIGER-S and LETTER-S. These are deep linear versions of TIGER and LETTER that use the same codebook configuration as TSTRec, only with different structures. We measure recommendation performance using Recall@K and NDCG@K. Importantly, we also report the hallucination rate. For more details of compared baselines and metrics, please refer to Appendix A.3 and A.4. All reported results in the table are obtained from multiple runs on a single H20 GPU.

Table 1: Comparison results on the industrial and MIND-small datasets with a sequence length of 30. The best results are **bolded**, the second best are underlined. The results for other sequence lengths are provided in Appendix B.

Dataset	Method	Recall@5	Recall@10	Recall@20	NDCG@5	NDCG@10	NDCG@20
Industrial Dataset	BERT4Rec	0.0229	0.0388	0.0603	0.0143	0.0194	0.0248
	SASRec	0.0267	0.0429	0.0687	0.0174	0.0225	0.0290
	TIGER	0.0406	0.0587	0.0758	0.0282	0.0340	0.0383
	TIGER-S	0.0420	0.0613	0.0818	0.0282	0.0344	0.0396
	LETTER	0.0419	0.0585	0.0775	0.0289	0.0340	0.0389
	LETTER-S	0.0409	0.0586	0.0754	0.0269	0.0326	0.0368
	TSTRec(w/o TGSC)	0.0448	0.0642	0.0827	0.0299	0.0362	0.0409
	TSTRec	**0.0474**	**0.0683**	**0.0869**	**0.0319**	**0.0386**	**0.0433**
MIND-small	BERT4Rec	0.0647	0.107	0.167	0.0402	0.0537	0.0689
	SASRec	0.0683	0.106	0.165	0.0416	0.0532	0.0675
	TIGER	0.0804	0.124	0.182	0.0515	0.0655	0.0800
	TIGER-S	0.0834	0.127	0.185	0.0525	0.0666	0.0811
	LETTER	0.0809	0.132	0.198	0.0521	0.0686	0.0852
	LETTER-S	0.0814	0.136	0.198	0.0514	0.0689	0.0845
	TSTRec(w/o TGSC)	0.0910	0.143	0.208	0.0588	0.0754	0.0918
	TSTRec	**0.0945**	**0.144**	**0.212**	**0.0598**	**0.0758**	**0.0928**

Table 2: Recall@5 performance comparison of different sequence lengths. The best results are **bolded**, the second best are underlined.

Method	Industrial Dataset			MIND-small	
	$L = 30$	$L = 50$	$L = 100$	$L = 20$	$L = 30$
BERT4Rec	0.0229	0.0300	0.0356	0.0628	0.0647
SASRec	0.0267	0.0312	0.0314	0.0671	0.0683
TIGER	0.0406	0.0432	0.0469	0.0757	0.0804
TIGER-S	0.0420	0.0444	0.0500	0.0778	0.0834
LETTER	0.0419	0.0430	0.0447	0.0773	0.0809
LETTER-S	0.0409	0.0445	0.0442	0.0735	0.0814
TSTRec(w/o TGSC)	0.0448	0.0456	0.0521	0.0786	0.0910
TSTRec	**0.0474**	**0.0492**	**0.0550**	**0.0810**	**0.0945**

4.2 Main Results

Table 1 presents the overall performance comparison on the large-scale industrial dataset and MIND-small dataset with sequence length of 30. Table 2 shows the Recall@5 results across different sequence lengths on both datasets. For comprehensive experimental results with other metrics and sequence lengths, please refer to Appendix B.

Tree-structured tokenization surpasses deep linear architectures. To evaluate the structural advantage of our tokenizer, we compare TSTRec against linear codebooks using the same codebook configuration. As shown in Table 1, TSTRec (w/o TGSC) consistently outperforms its deep linear counterparts, TIGER-S and LETTER-S. This superiority can be attributed to the multi-path structure of the tree codebook: each root-to-leaf path explicitly reflects a distinct semantic topic, effectively avoiding the semantic entanglement inherent in single-chain linear quantization.

TGSC effectively preserves long-Term signals and suppresses noise. Integrating Topic-Guided Session Compression (TGSC) consistently boosts performance across all sequence lengths, enabling the full TSTRec model to achieve state-of-the-art results. This improvement arises from two cooperative mechanisms: semantic aggregation allows for the effective utilization of distant historical interactions, while the semantic clustering process inherently filters out isolated noise interactions.

To further investigate the reliability of generative recommenders, we analyze the hallucination rate across different methods and sequence length.

TGSC strengthens long-sequence modeling. Compared with the hard-truncation strategy adopted by TSTRec (w/o TGSC), the compress-then-truncate design of TGSC fundamentally makes better use of a limited sequence length to model long histories, covering a longer span of user history under the same computational cost. As a result, the recommender simultaneously gains a wider behavioral horizon and a cleaner conditional token distribution. As shown in Table 2, TGSC consistently achieves the best performance across various sequence lengths, which reflects its advantage in long-sequence modeling. For complete experiments on other metrics for different sequence lengths, please refer to Appendix B.

Stable performance across sequence lengths. Beyond expanding the receptive field, the mechanism of TGSC equips TSTRec with strong stability across different sequence lengths. Standard methods often perform worse or exhibit little improvement on longer sequences, because simply adding more raw items brings in more noise. In contrast,

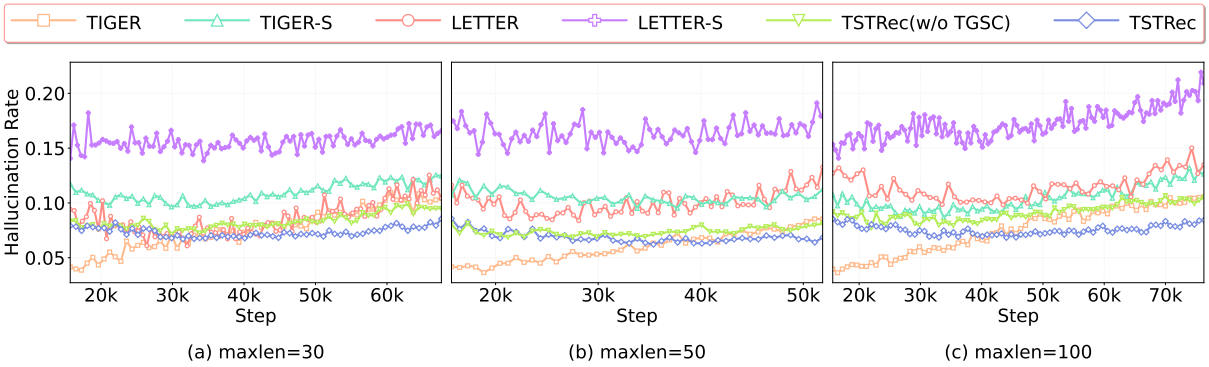

Figure 2: The hallucination rate of different methods for three sequence lengths on an industrial dataset.

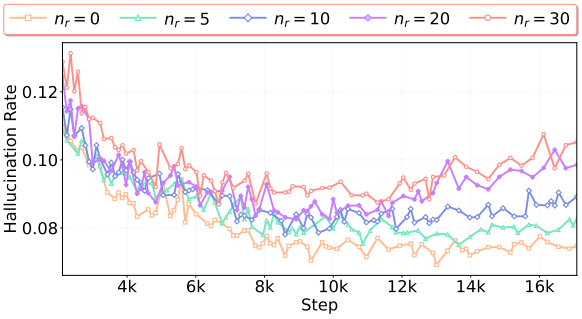

Figure 3: The hallucination rate of different value with the retention size n_r.

by inherently filtering out isolated interactions during the semantic clustering phase, TSTRec successfully protects the recommendation process from sequence-level noise. Therefore, our method consistently outperforms all baselines across different length settings, demonstrating strong robustness and adaptability to diverse user history lengths.

4.3 Hallucination Study

Tree-structured codebook intrinsically restricts hallucination. Fig. 2 illustrates the hallucination rate trajectories during training on the industrial dataset. For all methods, the hallucination rate exhibits an upward trend as training progresses, which occurs because the recommender gradually overfits to specific token co-occurrence patterns in the training data. However, TSTRec consistently maintains the lowest hallucination rate among all evaluated models. While deep linear codebooks naturally incur higher hallucination rates due to their single-chain quantization, the hierarchical architecture of our tree-structured codebook inherently bounds the generation space. This demonstrates that the multi-path semantic sharing mechanism explicitly reduces the probability of decoding invalid token combinations at the structural level.

Semantic compression denoises token distribution. The denoising capabilities of TGSC provide an additional defense against hallucination. As detailed in Fig. 3 regarding the recent retention window size (n_r), the degree of sequence compression is inversely correlated with the hallucination rate. Specifically, a smaller n_r (indicating more thorough compression and denoising) leads to a strictly lower hallucination rate, with the absolute minimum occurring at $n_r = 0$ (full compression). Conversely, configurations that retain more uncompressed raw items ($n_r = 20$ and $n_r = 30$) suffer from higher hallucination rates. This phenomenon confirms the denoising mechanism of TGSC: retaining excessive uncompressed interactions introduces individual behavioral noise, which disperses the learned conditional token distribution and thereby increases the model's likelihood of generating invalid semantic sequences.

4.4 Ablation Study

To verify the effectiveness of incorporating LLM-based world knowledge via topic information, we compare the performance of TS-RQVAE with and without topic embeddings in its reconstruction objective. As shown in Table 3 and Table 4, introducing topics yields consistent improvements on both the industrial and public datasets. This indicates that fine-grained topics significantly improve the tokenizer's representation quality, ultimately elevating recommendation performance.

4.5 Hyperparameter Analysis

As shown in Table. 5 and Table. 6, we examine the compression ratio γ and the recent retention window size n_r on the industrial dataset with sequence length of 50.

We evaluate the compression ratio γ in the val-

Table 3: Topic ablation study on the industrial dataset. The best results are **bolded**.

Sequence Length	Topic	Recall@5	Recall@20	NDCG@5	NDCG@20
30	×	0.0431	0.0804	0.0288	0.0396
	✓	**0.0448**	**0.0827**	**0.0299**	**0.0409**
50	×	0.0441	0.0869	0.0297	0.0421
	✓	**0.0456**	**0.0878**	**0.0304**	**0.0426**
100	×	0.0492	0.0918	0.0324	0.0447
	✓	**0.0521**	**0.0941**	**0.0352**	**0.0474**

Table 4: Topic ablation study on MIND dataset. The best results are **bolded**.

Sequence Length	Topic	Recall@5	Recall@20	NDCG@5	NDCG@20
20	×	0.0726	0.185	0.0456	0.0774
	✓	**0.0773**	**0.194**	**0.0488**	**0.0819**
30	×	0.0858	0.195	0.0555	0.0864
	✓	**0.0910**	**0.208**	**0.0588**	**0.0918**

ues $\gamma \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$. When γ is excessively low (0.1), the compression strategy becomes overly aggressive. It forces lots of items into a topic cluster, causing substantial information loss as items from distinct interest directions are forcibly merged, thereby erasing semantic diversity. In contrast, compression is insufficient when γ is excessively high (0.9), causing incident noise from distant interactions to persist as isolated clusters. Thus, $\gamma = 0.3$ strikes a balance between information preservation and denoising, ensuring that each topic cluster retains adequate semantic discriminability while noisy interactions are effectively aggregated and diluted.

We examine the recent retention window size n_r by evaluating values in $\{0, 5, 10, 20, 30\}$. The results show that the recent interactions are irreplaceable. Although full compression ($n_r = 0$) yields the lowest hallucination rate, its recommendation accuracy remains inferior to that of $n_r = 5$. This occurs because the latest interactions have a strong influence on predicting the next item, as the immediate intent of a user is closely related to their last few clicks. Setting $n_r = 0$ compresses these recent interactions, and the average aggregation inevitably dilutes the precision of such critical signals, causing a slight loss in the capture of immediate intent. In contrast, an excessively large retention window introduces noise. As n_r increases from 5 to 30, the performance of the recommendation gradually degrades. Retaining more uncompressed raw items preserves noise from isolated interactions, which directly interferes with the learning of conditional token distributions by the recommender. Ultimately, the optimal balance is achieved at $n_r = 5$, where only a few recent interactions are retained to capture immediate user interest, while TGSC effectively denoises the remainder of the sequence. In practice, n_r can be dynamically adjusted according to the specific application scenario and sequence length.

Table 5: Performance comparison of different compression rates (γ). The best results are **bolded**.

γ	Recall@5	Recall@20	NDCG@5	NDCG@20
0.1	0.0473	0.0905	0.0320	0.0445
0.3	**0.0492**	**0.0920**	**0.0326**	**0.0450**
0.5	0.0476	0.0877	0.0316	0.0432
0.7	0.0488	0.0906	0.0326	0.0447
0.9	0.0470	0.0885	0.0312	0.0433

Table 6: Performance comparison of different recent retention sizes (n_r). The best results are **bolded**.

m	Recall@5	Recall@20	NDCG@5	NDCG@20
0	0.0512	0.0952	0.0340	0.0467
5	**0.0518**	**0.0964**	**0.0349**	**0.0478**
10	0.0491	0.0939	0.0324	0.0453
20	0.0492	0.0920	0.0326	0.0450
30	0.0474	0.0880	0.0310	0.0428

5 Conclusion

In this work, we introduce TSTRec, a generative framework that bridges the gap between discrete item identifiers and LLM-based world knowledge. By proposing TS-RQVAE, we construct a hierarchical tree-structured codebook where branches explicitly align with multi-granularity semantic topics, effectively disentangling multi-intent item polysemy. Additionally, we design the TGSC strategy to compress user's historical sequences, optimizing the trade-off between long-term interest preservation and sequence denoising. Experiments on both a large-scale industrial dataset and a public dataset demonstrate that TSTRec significantly outperforms baselines in standard metrics. These results validate that structurally aligning tokenization with the external world knowledge of LLMs while systematically denoising interaction sequences is critical for robust generative recommendation.

6 Limitations

Although TSTRec shows strong performance in generative recommendation, we acknowledge a few limitations that provide avenues for future work.

First, our framework depends heavily on item text metadata. The Multi-Granularity Topic Gener-

ator needs descriptive text to effectively prompt LLMs for diverse semantic aspects. In scenarios with extremely sparse metadata or purely ID-based collaborative filtering settings, generating high-quality topics becomes challenging.

Second, the architecture of the Tree-Structured RQ-VAE (TS-RQVAE) is currently static. Structural hyperparameters, such as the number of root-to-leaf paths (M) and the depth of the tree (K), must be predefined based on the dataset. This structure might not adapt well to domains where items have highly variable semantic complexity.

Finally, there is a clear trade-off in the Topic-Guided Session Compression (TGSC) strategy. While clustering and average pooling reduce isolated noise and keep long-term semantics within a limited sequence length, they discard the strict chronological order of interactions inside each cluster. Although the overall time order across different sessions is maintained, fine-grained sequential patterns within older time windows are lost. Future work could explore dynamic tree-routing methods and compression techniques that better preserve sequence order.

References

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 *Proceedings of the 17th ACM Conference on Recommender Systems (RecSys)*, pages 1007–1014.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. Autoregressive entity retrieval. In *International Conference on Learning Representations*.

Yashar Deldjoo, Zhankui He, Julian McAuley, Anton Korikov, Scott Sanner, Arnau Ramisa, René Vidal, Maheswaran Sathiamoorthy, Atoosa Kasirzadeh, and Silvia Milano. 2024. A review of modern recommender systems using generative models (GenRecSys). In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 6448–6458.

Jiaxin Deng, Shiyao Wang, Kuo Cai, Lejian Ren, Qigen Hu, Weifeng Ding, Qiang Luo, and Guorui Zhou. 2025. Onerec: Unifying retrieve and rank with generative recommender and iterative preference alignment. *arXiv preprint arXiv:2502.18965*.

Chao Feng, Wuchao Li, Defu Lian, Zheng Liu, and Enhong Chen. 2022. Recommender forest for efficient retrieval. In *Advances in Neural Information Processing Systems*, volume 35, pages 38912–38924. Curran Associates, Inc.

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 (RecSys)*, pages 299–315.

Shijie Geng, Juntao Tan, Shuchang Liu, Zuohui Fu, and Yongfeng Zhang. 2023. VIP5: Towards multimodal foundation models for recommendation. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 9606–9620.

F. Maxwell Harper and Joseph A. Konstan. 2015. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 5(4).

Balázs Hidasi and Alexandros Karatzoglou. 2018. Recurrent neural networks with top-k gains for session-based recommendations. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 843–852.

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 (RecSys)*, pages 241–248.

Wenyue Hua, Shuyuan Xu, Yingqiang Ge, and Yongfeng Zhang. 2023. How to index item ids for recommendation foundation models. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, SIGIR-AP '23, page 195–204, New York, NY, USA. Association for Computing Machinery.

Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *Proceedings of the 18th IEEE International Conference on Data Mining (ICDM)*, pages 197–206.

Anton Klenitskiy and Alexey Vasilev. 2023. Turning dross into gold loss: Is BERT4Rec really better than SASRec? In *Proceedings of the 17th ACM Conference on Recommender Systems (RecSys)*, pages 1120–1125.

Enze Liu, Bowen Zheng, Cheng Ling, Lantao Hu, Han Li, and Wayne Xin Zhao. 2025. Generative recommender with end-to-end learnable item tokenization. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '25, page 729–739, New York, NY, USA. Association for Computing Machinery.

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, page 43–52, New York, NY, USA. Association for Computing Machinery.

Yunsheng Pang, Zijian Liu, Yudong Li, Shaojie Zhu, Zijian Luo, Chenyun Yu, Sikai Wu, Shichen Shen, Cong Xu, Bin Wang, and 1 others. 2025. HiGR: Efficient generative slate recommendation via hierarchical planning and multi-objective preference alignment. *arXiv preprint arXiv:2512.24787*.

Haohao Qu, Wenqi Fan, Zihuai Zhao, and Qing Li. 2025. TokenRec: Learning to tokenize ID for LLM-based generative recommendations. *IEEE Transactions on Knowledge and Data Engineering*, 37(10):6216–6231.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Q. Tran, Jonah Samost, Maciej Kula, Ed H. Chi, and Mahesh Sathiamoorthy. 2023. Recommender systems with generative retrieval. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36.

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 (CIKM)*, 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*, SIGIR '24, page 355–364, New York, NY, USA. Association for Computing Machinery.

Wenjie Wang, Honghui Bao, Xinyu Lin, Jizhi Zhang, Yongqi Li, Fuli Feng, See-Kiong Ng, and Tat-Seng Chua. 2024a. Learnable item tokenization for generative recommendation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM)*, pages 2400–2409.

Wenjie Wang, Xinyu Lin, Fuli Feng, Xiangnan He, and Tat-Seng Chua. 2023. Generative recommendation: Towards next-generation recommender paradigm. *arXiv preprint arXiv:2304.03516*.

Ye Wang, Jiahao Xun, Minjie Hong, Jieming Zhu, Tao Jin, Wang Lin, Haoyuan Li, Linjun Li, Yan Xia, Zhou Zhao, and Zhenhua Dong. 2024b. Eager: Two-stream generative recommender with behavior-semantic collaboration. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 3245–3254, New York, NY, USA. Association for Computing Machinery.

Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, and 1 others. 2020. Mind: A large-scale dataset for news recommendation. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 3597–3606.

Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, Hui Xiong, and Enhong Chen. 2024. A survey on large language models for recommendation. *World Wide Web*, 27(5):60.

Yi Xu, Moyu Zhang, Chenxuan Li, Zhihao Liao, Haibo Xing, Hao Deng, Jinxin Hu, Yu Zhang, Xiaoyi Zeng, and Jing Zhang. 2026. Mmq: Multimodal mixture-of-quantization tokenization for semantic id generation and user behavioral adaptation. In *Proceedings of the Nineteenth ACM International Conference on Web Search and Data Mining*, WSDM '26, page 788–797, New York, NY, USA. Association for Computing Machinery.

Bowen Zheng, Yupeng Hou, Hongyu Lu, Yu Chen, Wayne Xin Zhao, Ming Chen, and Ji-Rong Wen. 2024. Adapting large language models by integrating collaborative semantics for recommendation. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pages 1435–1448.

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 and Knowledge Management (CIKM)*, 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*, RecSys '24, page 969–974, New York, NY, USA. Association for Computing Machinery.

A Detail of Experimental Setup

A.1 Datasets

Industrial Dataset. This dataset is collected from a top social media platform, covering three months of user interaction data. Unlike typical public datasets such as MovieLens (Harper and Konstan, 2015) and Amazon (McAuley et al., 2015), this dataset contains rich article content, including titles, text, and tags, which provides high-quality input for multi-granularity topic generation with LLMs. Furthermore, the sequence length of each sample in this dataset exceeds 100. Since the user interaction sequences are sampled from different scenarios, we adopt a user-split strategy to construct the dataset. The Industrial dataset statistics are summarized in Table 7. We experiment with sequence lengths of 30, 50, and 100.

Table 7: Statistics of the industrial dataset.

Items	Train Users	Val Users	Test Users
3,863,853	50,000	14,395	20,000

MIND-small Dataset. MIND (Microsoft News Dataset) (Wu et al., 2020) is a news recommendation dataset released by Microsoft, and we use its small version. We follow the standard leave-one-out strategy for dataset construction. Due to the limited sequence lengths in this dataset, we experiment with sequence lengths of 20 and 30.

A.2 Hyperparameter Configurations

Industrial Dataset. Since the industrial dataset contains rich content, we set $M = 4$ to generate diverse topics. TS-RQVAE employs a complete binary tree of depth $K = 3$ (7 codebook nodes, $M = 4$ paths). To ensure a fair comparison with the 1024^3 baseline in terms of semantic combinatorial space, we set the codebook size to 64 instead of 1024. TGSC follows a compress-then-truncate strategy with $\gamma = 0.3$ and $m = 5$ and the base window step is set to $\Delta_0 = 3$ days. The test of the hallucination rate began after 20 epochs.

MIND Dataset. News articles in MIND provide sparse textual information, making it difficult to generate four distinct topics. We set $M = 2$ to avoid semantic overlap and preserve codebook discriminability. TS-RQVAE uses the left subtree (4 codebook nodes) of a complete binary tree, with codebook size 256 and embedding dimension 64. TGSC sets $\gamma = 0.3$ and $n_r = 10$. Since per-interaction timestamps are unavailable, the window step is measured in item count with $\Delta_0 = 10$ items. We use *Qwen-3.5 A3B* and *BGE-large* model to generate topics and embeddings, respectively.

Recommender. We use the T5 (Raffel et al., 2020) architecture as the generative recommender, following the configuration described in TIGER (Rajput et al., 2023). The model contains approximately 4.5 million parameters. The hyperparameters are $\beta = 0.25$ and $\lambda = 1$.

A.3 Baseline Descriptions

We compare TSTRec against both classic sequential recommenders and generative recommendation baselines. (1) **SASRec** (Kang and McAuley, 2018) uses self-attention to model user behavior sequences for next-item prediction. (2) **BERT4Rec** (Sun et al., 2019) employs bidirectional Transformer encoders with the cloze objective to capture dynamic preferences. (3) **TIGER** (Rajput et al., 2023) quantizes items into discrete semantic IDs through hierarchical linear codebooks, establishing the paradigm of semantic tokenization followed by autoregressive generation. (4) **LETTER** (Wang et al., 2024a) enhances semantic ID discriminability by injecting collaborative signals via contrastive learning. (5) **TIGER-S** is a deep linear variant of TIGER that uses the same codebook configuration as TSTRec, only with different structures. (6) **LETTER-S** is the deep linear variant of LETTER, also using the same codebook configuration as TSTRec. On the industrial dataset, both LETTER and LETTER-S are evaluated without the prefix tree to compare the hallucination rate.

A.4 Evalatioin Metrics

We evaluate performance using Recall (**Recall@K**) and Normalized Discounted Cumulative Gain (**NDCG@K**) ($K = 5, 10, 20$) for recommendation metrics. Additionally, we report the Hallucination Rate (**Hal Rate**), defined as the proportion of generated token sequences that cannot be decoded into valid item IDs. A lower HR indicates a more valid and structured semantic space. We also report the hallucination rate on the industrial dataset, which defines as the proportion of token sequences generated by the recommender that cannot be decoded into valid item IDs. All reported results in the table are obtained from multiple runs on a single H20 GPU.

B Main Experimental Results

Table 8: Comparison results on the industrial dataset. The best results are **bolded**, the second best are underlined.

Sequence Length	Method	Recall@5	Recall@10	Recall@20	NDCG@5	NDCG@10	NDCG@20
30	BERT4Rec	0.0229	0.0388	0.0603	0.0143	0.0194	0.0248
	SASRec	0.0267	0.0429	0.0687	0.0174	0.0225	0.0290
	TIGER	0.0406	0.0587	0.0758	0.0282	0.0340	0.0383
	TIGER-S	0.0420	0.0613	0.0818	0.0282	0.0344	0.0396
	LETTER	0.0419	0.0585	0.0775	0.0289	0.0340	0.0389
	LETTER-S	0.0409	0.0586	0.0754	0.0269	0.0326	0.0368
	TSTRec(w/o TGSC)	0.0448	0.0642	0.0827	0.0299	0.0362	0.0409
	TSTRec	**0.0474**	**0.0683**	**0.0869**	**0.0319**	**0.0386**	**0.0433**
50	BERT4Rec	0.0300	0.0478	0.0740	0.0196	0.0253	0.0319
	SASRec	0.0312	0.0506	0.0786	0.0202	0.0264	0.0335
	TIGER	0.0432	0.0638	0.0828	0.0296	0.0362	0.0410
	TIGER-S	0.0444	0.0656	0.0871	0.0299	0.0367	0.0422
	LETTER	0.0430	0.0618	0.0796	0.0287	0.0348	0.0394
	LETTER-S	0.0445	0.0646	0.0831	0.0298	0.0363	0.0410
	TSTRec(w/o TGSC)	0.0456	0.0674	0.0878	0.0304	0.0374	0.0426
	TSTRec	**0.0492**	**0.0713**	**0.0920**	**0.0326**	**0.0397**	**0.0450**
100	BERT4Rec	0.0356	0.0545	0.0807	0.0233	0.0294	0.0360
	SASRec	0.0314	0.0522	0.0823	0.0205	0.0271	0.0346
	TIGER	0.0469	0.0688	0.0895	0.0319	0.0390	0.0442
	TIGER-S	0.0500	0.0727	0.0937	0.0336	0.0409	0.0463
	LETTER	0.0447	0.0652	0.0888	0.0305	0.0386	0.0391
	LETTER-S	0.0442	0.0636	0.0873	0.0288	0.0357	0.0411
	TSTRec(w/o TGSC)	0.0521	0.0743	0.0941	0.0352	0.0423	0.0474
	TSTRec	**0.0550**	**0.0766**	**0.0981**	**0.0369**	**0.0439**	**0.0494**

Table 9: Performance comparison on MIND-small dataset. The best results are **bolded**, the second best are underlined.

Sequence Length	Method	Recall@5	Recall@10	Recall@20	NDCG@5	NDCG@10	NDCG@20
20	BERT4Rec	0.0628	0.105	0.162	0.0392	0.0526	0.0669
	SASRec	0.0671	0.111	0.171	0.0420	0.0561	0.0695
	TIGER	0.0757	0.121	0.178	0.0486	0.0629	0.0775
	TIGER-S	0.0778	0.122	0.180	0.0494	0.0633	0.0782
	LETTER	0.0773	0.125	0.182	0.0488	0.0658	0.0802
	LETTER-S	0.0735	0.123	0.177	0.0458	0.0631	0.0768
	TSTRec(w/o TGSC)	0.0786	0.128	0.194	0.0510	0.0652	0.0819
	TSTRec	**0.0810**	**0.136**	**0.195**	**0.0514**	**0.0723**	**0.0855**
30	BERT4Rec	0.0647	0.107	0.167	0.0402	0.0537	0.0689
	SASRec	0.0683	0.106	0.165	0.0416	0.0532	0.0675
	TIGER	0.0804	0.124	0.182	0.0515	0.0655	0.0800
	TIGER-S	0.0834	0.127	0.185	0.0525	0.0666	0.0811
	LETTER	0.0809	0.132	0.198	0.0521	0.0686	0.0852
	LETTER-S	0.0814	0.136	0.198	0.0514	0.0689	0.0845
	TSTRec(w/o TGSC)	0.0910	0.143	0.208	0.0588	0.0754	0.0918
	TSTRec	**0.0945**	**0.144**	**0.212**	**0.0598**	**0.0758**	**0.0928**

C Prompt for Topic Generation on MIND Dataset

To extract multi-level semantic signals and generate progressive topic descriptions for each item, we employ the structured prompt template detailed below. The prompt explicitly constraints the LLM to output exactly 4 core topics following a progressive semantic granularity from fine to coarse.

Prompt Template for Topic Generation

You are a top-tier recommendation algorithm and news knowledge graph expert. Your task is to deeply analyze the provided MIND news article data (focusing on 'title', 'abstract', 'category', and 'subcategory'), and extract exactly 4 core topics strictly following a "progressive semantic granularity from fine to coarse". These data will be used to train Topic Embeddings for a news recommendation system.

Please strictly adhere to the following steps and rules:

Step 1: Progressive & Self-contained Topic Extraction (CORE REQUIREMENT)
The 4 topics you extract MUST progress from extremely specific (almost like a compressed headline) to broad interest domains.
[CRITICAL RULE]: **The first two topics MUST be highly specific and narrow. The last two topics MUST be progressively broader.** Each topic must be a self-contained, meaningful noun phrase — never an isolated word or vague term.

You must extract exactly 1 topic for each of the following 4 progressive levels:

1. Level 1 — Headline-level Specificity (THIS article only): Compress the title into a topic that uniquely identifies THIS specific news story. MUST retain all proper nouns (person names, organization names, place names, brand names) and the core event/action from the title. This topic should match ONLY this one article and no other.
- Good examples: # Phil Fulmer Blasts NCAA Over Freshman Eligibility, # Prince George Royal Life Photo Collection, # Pregnant Woman Fears Telling Male Bosses
- Bad examples (TOO VAGUE): # NCAA Eligibility Dispute, # Royal Family Photos, # Workplace Pregnancy Concerns

2. Level 2 — Narrow Subtopic (a small cluster of ∼5-20 similar articles): Remove the most specific proper nouns but keep the specific angle/theme. This topic should match a small group of very similar news stories sharing the same narrow focus.
- Good examples: # NCAA Transfer Eligibility Disputes, # British Royal Children Photo Albums, # Pregnancy Discrimination Fear in Workplace
- Bad examples (TOO VAGUE): # College Sports Controversies, # Royal Family News, # Gender Issues at Work

3. Level 3 — Broad Subcategory (hundreds of articles): Generalize to the broader news subcategory or interest area. Remove specific event details, keep only the domain + angle.
- Good examples: # College Basketball Eligibility and Regulations, # British Royal Family Lifestyle Coverage, # Workplace Gender Discrimination and Motherhood
- Bad examples (TOO VAGUE): # Sports, # Lifestyle, # Gender Issues

4. Level 4 — Macro Interest Domain (thousands of articles): The broadest user interest that explains WHY someone reads this type of news. Must still be a meaningful multi-word phrase (3+ words), NOT a single category word.
- Good examples: # NCAA College Sports Fan Community, # Celebrity and Royal Entertainment, # Women Career and Family Balance
- Bad examples (TOO VAGUE): # Sports, # Entertainment, # Lifestyle

[GRANULARITY CHECK — MUST VERIFY BEFORE OUTPUT]:
- Level 1 MUST contain at least one proper noun (person/place/org name) from the title, OR a highly specific event descriptor that uniquely identifies this article.
- Level 2 MUST still be narrow enough that you can imagine only 5-20 similar articles existing.
- Level 3 and Level 4 MUST be clearly broader than Level 2 — if Level 3 looks similar to Level 2, you need to make Level 3 MORE general.
- If your Level 1 does NOT contain a proper noun or specific identifier from the title, REDO it.

Step 2: Comprehensive Filtering & Strict Output Constraints
- **Semantic Completeness**: Absolutely NO isolated adjectives or single words. Each topic must be a complete conceptual phrase.
- **Length & Formatting**: Each topic should be between 3 to 8 English words. Capitalize the First Letter of Each Word (Title Case). Each topic MUST start with a '#' followed by a space (e.g., '# Topic Name').
- **Language**: ALL topics MUST be in English. Do NOT output any non-English text under any circumstances.
- **Output Format**: Strictly output ONLY the 4 topics (one per line), absolutely DO NOT output any extra introductory remarks, analytical processes, or explanatory text.