HIERARCHICAL TOKENIZATION OF MULTIMODAL MUSIC DATA FOR GENERATIVE MUSIC RETRIEVAL

Wo Jae Lee, Rifat Joyee, Zhonghao Luo, Grace Kochavi, Sudev Mukherjee, and Emanuele Coviello

Amazon Music, San Francisco, USA

ABSTRACT

Recent advances in generative retrieval allow large language models (LLMs) to recommend items by generating their identifiers token by token. This requires each item to be represented by a compact, semantically meaningful sequence of tokens that an LLM can understand. We introduce a method to generate multimodal music tokens (3MToken) that transforms rich metadata from a music database—including audio, credits, semantic tags, song and artist descriptions, musical characteristics, release dates, and consumption patterns—into discrete tokens using a Residual-Quantized Variational Autoencoder (RQ-VAE). The model learns hierarchical representations, with coarse features captured at early quantization levels and fine-grained details refined at later levels. Building on this, we propose 3MTokenRec, an instruction-tuned LLM capable of generating 3MToken sequences for retrieval while adaptively weighting modalities based on query intent. Experiments show that our approach outperforms unimodal and baseline generative methods on content-based music retrieval and text-to-music retrieval tasks.

Index Terms— Multimodal music tokenization, Generative music retrieval, Music recommendation systems

1. INTRODUCTION

With the advancement of generative AI, music streaming services are beginning to offer new user experiences, such as prompt-based music discovery and recommendation [1, 2]. One of the primary challenges in this setting is enabling LLMs to interact effectively with the data in the music database to generate high-quality recommendations [3, 4]. Traditional search and recommender systems have relied on continuous embeddings and similarity-based retrieval, where a dense vector represents each item, and retrieval is performed using nearest-neighbor search in the embedding space [5, 6]. In contrast, recent research has proposed a new paradigm known as generative retrieval, reformulating the recommendation task as a sequence generation problem [7, 8, 9, 10].

In generative retrieval, an autoregressive model directly generates the identifier of the next item token by token, similar to how a language model generates word tokens in natural language. This paradigm enables a recommender to be built on top of an LLM, which can seamlessly combine dialogue, reasoning, and recommendation in a single framework [11, 12]. One common approach is to represent items as natural language, such as movie titles or song names, which leverages the parametric knowledge of LLMs [13, 14, 15, 16, 17]. However, this free-text generation faces challenges such as mapping texts to an entity, title ambiguity, multilingual variations, and increased multi-token decoding latency in real-world applications. Due to these limitations, the preferred approach has shifted toward generative retrieval using learned discrete IDs, where the model directly predicts items [18, 19]. Rajput et al. [8] showed that representing items with Semantic IDs (sequences of discrete codes learned from item embeddings) allows a transformer-based model to generate item IDs as outputs, thereby unifying recommendation with natural language generation. Doh et al. [7] extended this concept to music recommendation by expanding LLM's vocabulary with music tokens derived from multimodal data such as audio, lyrics, playlist, metadata, and tags, which outperforms unimodal approaches based solely on text or listening history in the music recommendation task.

In this context, the emergence of LLMs has catalyzed interest in developing discrete tokenization schemes for multimodal domains [20]. Recent advances in neural discrete representation learning such as VQ-VAE and its variants, have shown strong performance in image synthesis and audio generation [21, 22]. RQ-VAE extends vector quantization by employing multiple residual codebooks. While recent studies have explored tokenization for music recommendation, most prior work emphasizes end-task performance. In contrast, systematic evaluation of tokenizer quality across modalities and the role of modality selection at retrieval time—remains underexplored. Motivated by previous studies [8, 7] that use residual quantization to create hierarchical semantic IDs for LLMs, we propose a framework to instruction fine-tune a LLM to generate multimodal music tokens and retrieve tracks by weighting modality importance. Our method extends Qwen2.5-1.5B-Instruct [23] with discrete music tokens generated by RQ-VAE and adaptively focuses on the most relevant modalities for retrieval based on user intent. The proposed retrieval mechanism enables weighted token matching, making the system more practical for real-world music recommendation applications.

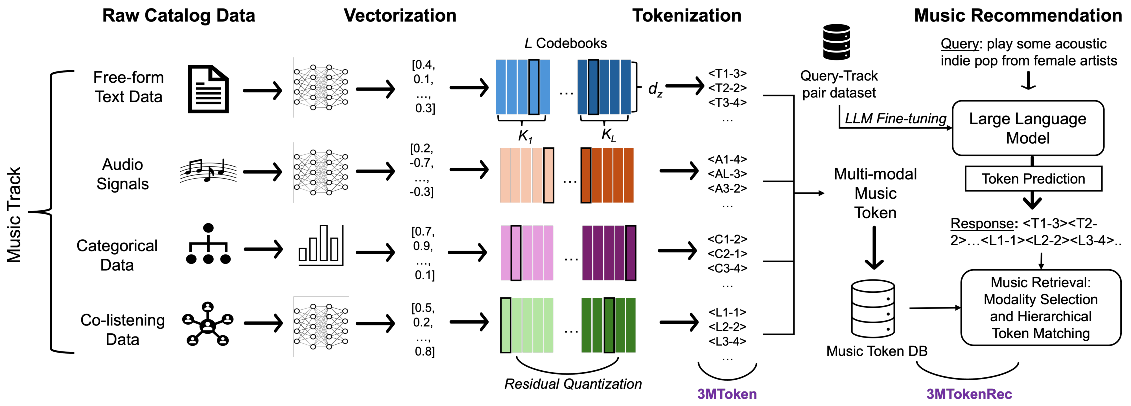


Fig. 1: Overview of the proposed system: multimodal music token (3MToken) generation and the generative music recommendation (3MTokenRec). The system ingests diverse modalities, transforms them into embeddings, and tokenizes them through a multi-stage quantization process. These tokens are then used for instruction fine-tuning with a query–track pair dataset.

2. METHOD

Fig. 1 presents an overview of the proposed model: (1) processing raw music data into multimodal embeddings, (2) transforming each track's multimodal embedding into a compact sequence of discrete tokens, (3) fine-tuning an LLM to predict multimodal music tokens from a given text prompt, and (4) retrieving top-k tracks from a pre-built database.

2.1. Music Data Vectorization

We leverage rich metadata aggregated from multiple sources to build a multimodal music tokenizer. Typically, this metadata is heterogeneous in format (e.g., categorical, scalar, and text). We categorize them into nine categories[1] (i.e. modalities) which we denote as AC, BM, ST, SC, MC, RI, SF, AB, and TC based on common user intent in music streaming services. After organizing the music data into the nine categories, the next step is to map each track-level data source into a multimodal embedding space. We generate multimodal embeddings with modality-specific encoders as follows: textual data are encoded into 4096-dimensional embeddings with a pre-trained text encoder [24]. Audio signal snippets are mapped into a 128-dimensional embedding using a CLAP-like model [25] that aligns sound recordings with textual context. Categorical metadata is transformed through binning, for example, release information is encoded with normalized year, cyclical month/day, and decade bins (18 dimensions), while musical characteristics are represented by discretized tempo (220 bins) and one-hot musical keys (25 bins). Consumption data is modeled with a session-based collaborative filtering embedding model [26, 27, 28] that learns track embeddings from co-listening and co-occurrence patterns.

2.2. Multimodal Music Token Formation with RQ-VAE

For each modality, we train a RQ-VAE model that consists of an encoder, a multi-level vector quantizer, and a decoder. Given an input embedding $\mathbf{x} \in \mathbb{R}^d$, the encoder network $f_\theta(\cdot)$ maps it to a latent representation $\mathbf{z}_e \in \mathbb{R}^{d_z}$ as follows $\mathbf{z}_e = f_\theta(\mathbf{x})$. Then, this latent space is quantized by multiple codebooks in series, which is the key to residual quantization. Instead of a single quantization step, RQ-VAE applies L sequential codebooks to iteratively refine the approximation. At the l-th level, for a given residual $\mathbf{r}_{l-1}$, the quantizer selects the closest codeword $\mathbf{e}_{k_l}$ from a learned codebook $\mathcal{E}_l = \{\mathbf{e}_1, \ldots, \mathbf{e}_{K_l}\}$ in Euclidean distance as follows: $k_l = \arg\min_k \|\mathbf{r}_{l-1} - \mathbf{e}_k\|_2^2$ where $k_l \in \{1, 2, \ldots, K_l\}$ denotes the index of the selected codeword at the l-th quantization level, K_l is the size of the codebook, $\mathbf{r}_0 = \mathbf{z}_e$ is the initial residual, and $\mathbf{r}_l = \mathbf{r}_{l-1} - \mathbf{e}_{k_l}$ is the updated residual after quantization. After multi-stage residual quantization, the final quantized representation is represented by the sum of all selected codewords: $\hat{\mathbf{z}}_q = \sum_{l=1}^{L} \mathbf{e}_{k_l}$. Then, the decoder $g_\phi(\cdot)$ reconstructs the original embedding from the quantized representation as follows: $\hat{\mathbf{x}} = g_\phi(\hat{\mathbf{z}}_q)$.

The total RQ-VAE loss can be defined by combining a reconstruction loss, a codebook loss, and a commitment loss to encourage the encoder outputs to stay close to the selected codewords as follows: $\mathcal{L}_{\text{RQ-VAE}} = \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 + \sum_{l=1}^{L} \left(\|\text{sg}[\mathbf{r}_{l-1}] - \mathbf{e}_{k_l}\|_2^2 + \beta \|\mathbf{r}_{l-1} - \text{sg}[\mathbf{e}_{k_l}]\|_2^2 \right)$ where sg$[\cdot]$ denotes the stop-gradient operator, β is the commitment weight [21]. sg$[\cdot]$ is used to control gradient flow. During inference, the discrete token sequence for each individual is obtained by concatenating the indices from all codebooks such as $(k_{mod,1}, k_{mod,2}, \ldots, k_{mod,L})$. Once we have trained an RQ-VAE model for each modality, we can encode mul-

[1] (1) *Artist Roles & Collaborations* (AC) – band members, featured artists, and vocalists, (2) *Basic Metadata* (BM) – track title and artist name, (3) *Semantic Tags* (ST) – genre and mood, (4) *Sonic Characteristics* (SC) – audio embeddings capturing acoustic properties, (5) *Musical Characteristics* (MC) – tempo and musical key, (6) *Release Information* (RI) – release date of a track, (7) *Song Facts* (SF) – recording details such as song history, (8) *Artist Biography* (AB) – artist attributes such as gender, birth year, and origin, (9) *Track Consumption* (TC) – data capturing co-listening patterns.

timodal features of a track into a set of tokens as follows: we (1) pass each modality-specific embedding through the corresponding encoder and quantize it through L codebooks to get indices and (2) map each index to a token string of the form `<{modality}{level}-{index}>`.

2.3. Instruction Fine-Tuned LLM and Music Retrieval

After tokenizing the multimodal music data into discrete tokens, we fine-tune a pre-trained LLM to understand and generate these music tokens in response to natural language queries. For the base LLM, we use Qwen2.5-1.5B-Instruct [23] with the assumption that a relatively small model is sufficient for the music retrieval task. To incorporate the multimodal music tokens, we extend the original tokenizer vocabulary $\mathcal{V}_{\text{original}}$ with our multimodal music tokens as follows $\mathcal{V}_{\text{new}} = \mathcal{V}_{\text{original}} \cup \{\text{music tokens}\} \cup \{\text{boundary tokens}\}$ where $\mathcal{V}_{\text{original}} = 151,646$ and boundary tokens are special tokens (e.g., `<begin_3MToken>` and `<end_3MToken>` which are appended at the beginning and end of multimodal music token to separate music tokens within the text in the training data. Regarding the music token, for each of the nine modalities with their respective codebook configurations, we add tokens following the format as described in Section 2.2. The generation process is constrained to produce valid token sequences in a predefined order of modalities (AC → BM → ST → SC → MC → RI → SF → AB → TC) while ensuring tokens progress from level 1 to L within each modality. The template for our training data is as follows:

```
<|im_start|>system\nYour task is recommending
music ...<|im_end|>\n<|im_start|>user\nplay some
90s rock music<|im_end|>\n<|im_start|>assistant\n
<begin_3MToken><AC1-21><AC2-73><AC3-63>...
<TC1-2><TC2-61><TC3-49><end_3MToken><|im_end|>\n
```

The vocabulary size contributed by 3MToken will be $\sum_{\text{mod}} \sum_{d=1}^{l} K_{mod,d}$ where $K_{mod,d}$ is the size of the codebook for a specific modality at quantization level d. Here, the combination of tokens across all modalities creates a large number of possible item codes. For example, if modality has codebook sizes $(K_{mod,1}, K_{mod,2}, K_{mod,3})$ where L=3, then that modality can produce $K_{mod,1} \times K_{mod,2} \times K_{mod,3}$ unique combinations of three codes. The multimodal code sequence has a capacity of $\prod_{\text{mod}} \prod_{d=1}^{3} K_{mod,d}$ unique sequences, which is large enough to cover all the items. To accommodate these new tokens, the model's embedding matrix is resized from $\mathbb{R}^{\mathcal{V}_{\text{original}} \times 1,536}$ to $\mathbb{R}^{\mathcal{V}_{\text{new}} \times 1,536}$ where 1,536 is the hidden dimension of Qwen2.5-1.5B-Instruct.

To estimate the importance of each modality for a given query, we fine-tune a pre-trained BERT-based model [29] with a custom regression head to predict relevance scores for different modalities based on free-form text. The model processes the query and outputs relevance scores $\in [0, 10]$ for each modality. These scores are used during inference, enabling the model to adaptively focus on the most relevant aspects of the music based on the user's intent. During inference, the fine-tuned LLM generates a sequence of music tokens autoregressively, and then the generated token sequence is matched against the pre-computed token database. We apply hierarchical matching, where level-2 matching is conditioned on level-1 matches, and level-3 matching is further conditioned on successful matches at levels 1 and 2. We then find top-k tracks based on token matching. We refer to the model that generates multimodal music tokens and retrieves top-k tracks as 3MTokenRec.

3. EXPERIMENTS

3.1. Dataset and Setup

We conduct experiments on a proprietary music catalog containing track- and artist-level metadata[2] covering about 1.6M tracks. We process the music data as described in Section 2 (denoted as $\mathcal{D}_c$). For instruction fine-tuning of Qwen2.5-1.5B-Instruct, we generate a synthetic query–track dataset by prompting song and artist metadata into an off-the-shelf LLM following approaches similar to [30, 31][3] (denoted as $\mathcal{D}_{qt}$). Lastly, to train a model estimating an importance of each modality, we use another off-the-shelf LLM to analyze text queries from $\mathcal{D}_{qt}$ and assign relevance scores ranging from 0 to 10 on each modality. This creates a dataset for training a predictive model that estimates the relevance of each modality given a text query (denoted as $\mathcal{D}_r$). The three datasets, $\mathcal{D}_c$, $\mathcal{D}_{qt}$, and $\mathcal{D}_r$ are used for the training of music tokenizer with RQ-VAE, the instruction fine-tuning of Qwen2.5-1.5B-Instruct, and the modality selection model, respectively.

3.2. Configuration for RQ-VAE and Fine-Tuning

In RQ-VAE,[4] so $\mathcal{V}_{\text{new}} = 153,664$. Each modality-specific RQ-VAE is trained with $\mathcal{D}_c$ for 150 epochs using AdamW optimizer [32] with a learning rate of 1e-4 and a batch size of 512. The fine-tuning of Qwen2.5-1.5B-Instruct model involves a causal language modeling objective where the model learns to predict music tokens autoregressively. Each training sequence in $\mathcal{D}_{qt}$ is constructed as a concatenation of the user query and the target 3MToken. During training, we apply causal attention masking to ensure that each position can only attend to previous positions. We fine-tune the model using AdamW with a learning rate of 1e-4, following a cosine schedule with 10% linear warmup. Training is performed for 10 epochs using distributed data parallelism across 16 NVIDIA A100 GPUs.

[2] The metadata are sourced from a combination of public databases, automated labeling systems, and expert review.

[3] We further process this synthetic dataset into the chat template [23].

[4] We adopt the following architecture for encoder $f_\theta(\cdot)$ and decoder $g_\phi(\cdot)$: a 4-layer feedforward neural network that compresses the input embedding through layers of size $512 \rightarrow 256 \rightarrow 128 \rightarrow d_z$, using ReLU activations and batch normalization at each layer and the decoder mirrors this structure symmetrically, reconstructing the original embedding.

3.3. Evaluation and Baselines

We evaluate 3MToken and 3MTokenRec on two tasks: (1) content-based retrieval (CBR) and (2) text-to-music retrieval (T2MR). For CBR, we compare quantization methods using Hit@k [7] on two datasets: 15K curated playlists (CP) and 30K co-occurrence pairs (CO) derived from listening sessions, under the assumption that tracks in the same playlist or session share semantic similarity. We use a seed track to retrieve the top-k most similar tracks based on music tokens. For T2MR, we evaluate 3MTokenRec on retrieving top-k tracks for human-curated text–track pairs. Queries span diverse aspects such as artist roles, collaborations, genres, eras, and biographies. Performance is measured with Precision@k, the proportion of relevant tracks among the retrieved results.

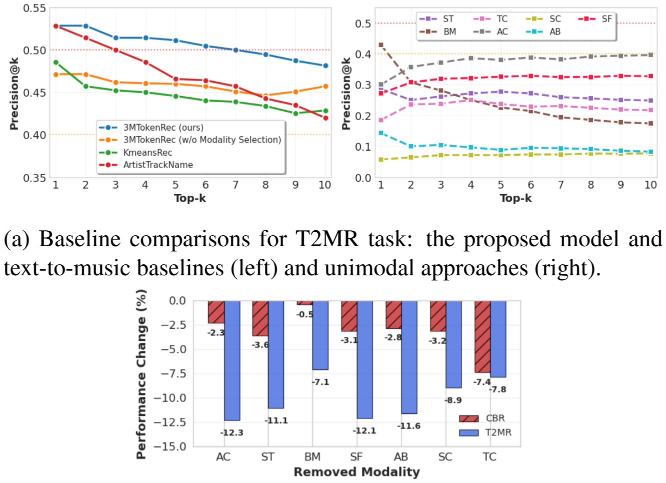
We compare 3MToken with K-means clustering, which partitions embeddings without learned representations [7], and VQ-VAE, an ablation using a single codebook. Both baselines use 1024 clusters per modality, 4.6× larger than RQ-VAE [7]. For T2MR, we evaluate 3MTokenRec against (1) an off-the-shelf LLM prompted to generate track titles and artist names (*ArtistTrackName*) [2], (2) Qwen2.5-1.5B-Instruct fine-tuned with K-means tokens (*KmeansRec*) [7], and (3) 3MTokenRec without modality selection. We also compare different unimodal variants with the proposed model.

Table 1: Hit@k on CBR task. The last two rows present relative % improvement with 3MToken to the second best models.

	k=5		k=10		k=20		k=50	
Method	**CP**	**CO**	**CP**	**CO**	**CP**	**CO**	**CP**	**CO**
Multimodal Approaches								
3MToken (ours)	**.284**	**.300**	**.352**	**.375**	**.418**	**.433**	**.513**	**.510**
K-means	.225	.228	.293	.309	.386	.387	.495	.495
VQ-VAE	.184	.178	.258	.247	.332	.322	.443	.430
Unimodal Approaches								
TC	.099	.165	.151	.239	.216	.322	.307	.426
ST	.073	.091	.112	.132	.158	.183	.233	.269
SC	.055	.078	.100	.123	.154	.179	.230	.260
AC	.084	.109	.108	.138	.130	.164	.172	.203
SF	.085	.139	.110	.180	.132	.206	.159	.229
AB	.028	.026	.046	.046	.068	.071	.110	.114
BM	.017	.024	.032	.037	.056	.064	.102	.110
vs. Multi.	**+27%**	**+32%**	**+20%**	**+21%**	**+8%**	**+12%**	**+4%**	**+3%**
vs. Uni.	**+187%**	**+82%**	**+133%**	**+57%**	**+93%**	**+34%**	**+67%**	**+20%**

3.4. Results and Discussion

We present the results of CBR task in Table 1 where 3MToken consistently outperforms the baselines across all metrics followed by K-means. K-means outperforms VQ-VAE, which indicates that the direct clustering better preserves the original embedding structure. Also, 3MToken consistently outperforms unimodal models (showing the 7 best-performing modalities for brevity in the table) on the retrieval task, demonstrating the effectiveness of our multimodal approach. The performance gap between multimodal and unimodal approaches is more significant at lower k values, indicating that the multimodal approach is particularly effective at identifying highly relevant music tracks early in the ranking process.



(a) Baseline comparisons for T2MR task: the proposed model and text-to-music baselines (left) and unimodal approaches (right).

(b) Ablation study result

Fig. 2: (a) Baseline comparisons (b) Ablation study reporting percentage changes relative to the full multimodal model.

Figure 2a shows performance from k=1 to k=10 for T2MR. The performance of most models declines as k increases, indicating that fewer relevant tracks are retrieved at larger k. *ArtistTrackName* performs better at lower k but degrades at higher k, suggesting that it captures more relevant tracks initially but lacks consistency as the candidate set expands. This baseline allows us to evaluate the effectiveness of generating Semantic IDs (i.e., 3MToken) vs. free texts. Fine-tuning with 3MToken improves average Precision@K by 13.7% over K-means, and adding a modality selection model[5] brings an additional 10.8% gain. Overall, our model consistently outperforms baselines, and our results validate that integrating information across modalities results in performance gains over unimodal approaches.

We conduct an ablation study where we remove one modality (selected in Table 1) at a time and report the average performance change in Fig. 2b. The results show that TC and AC are the most critical modalities for CBR and T2MR, respectively. Removing a modality always drops performance (-3.26% for CBR and -10.13% for T2MR on average) on both tasks, which indicates that encoding multimodal information is helpful for handling complex text queries embedded with multiple types of information as well as retrieving relevant tracks for a given seed track.

4. CONCLUSION

We present a framework for tokenizing diverse music data based on 9 categories into discrete tokens and fine-tuning an LLM for generative music retrieval. Our experiments demonstrate the effectiveness of 3MToken in the generative music retrieval task. As future work, we plan to investigate modality ordering with the goal of advancing multimodal tokenization and more generalizable generative music retrieval systems.

[5] Trained on $\mathcal{D}_r$, with 98.73% accuracy on the 20% held-out set.

5. REFERENCES

[1] Mathieu Delcluze, Antoine Khoury, Clémence Vast, Valerio Arnaudo, Léa Briand, Walid Bendada, and Thomas Bouabça, "Text2playlist: Generating personalized playlists from text on deezer," in *European Conference on Information Retrieval*. Springer, 2025, pp. 164–170.

[2] Enrico Palumbo, Gustavo Penha, Andreas Damianou, José Luis Redondo García, Timothy Christopher Heath, Alice Wang, Hugues Bouchard, and Mounia Lalmas, "Text2tracks: Prompt-based music recommendation via generative retrieval," *arXiv preprint arXiv:2503.24193*, 2025.

[3] Daeyong Kwon, SeungHeon Doh, and Juhan Nam, "Must-rag: Musical text question answering with retrieval augmented generation," 2025.

[4] Gustavo Penha, Ali Vardasbi, Enrico Palumbo, Marco De Nadai, and Hugues Bouchard, "Bridging search and recommendation in generative retrieval: Does one task help the other?," in *Proceedings of the 18th ACM Conference on Recommender Systems*, 2024, pp. 340–349.

[5] Rishabh Mehrotra, James McInerney, Hugues Bouchard, Mounia Lalmas, and Yizhou Yue, "Contextual and sequential user embeddings for large-scale music recommendation," in *Proceedings of the 14th ACM Conference on Recommender Systems (RecSys)*, 2020.

[6] Xiangyu Zhao, Maolin Wang, Xinjian Zhao, Jiansheng Li, Shucheng Zhou, Dawei Yin, Qing Li, Jiliang Tang, and Ruocheng Guo, "Embedding in recommender systems: A survey," *arXiv preprint arXiv:2310.18608*, 2023.

[7] Seungheon Doh, Keunwoo Choi, and Juhan Nam, "Talkplay: Multimodal music recommendation with large language models," *arXiv preprint arXiv:2502.13713*, 2025.

[8] Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, et al., "Recommender systems with generative retrieval," *Advances in Neural Information Processing Systems*, vol. 36, pp. 10299–10315, 2023.

[9] Weiwei Sun, Lingyong Yan, Zheng Chen, Shuaiqiang Wang, Haichao Zhu, Pengjie Ren, Zhumin Chen, Dawei Yin, Maarten Rijke, and Zhaochun Ren, "Learning to tokenize for generative retrieval," *Advances in Neural Information Processing Systems*, vol. 36, pp. 46345–46361, 2023.

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

[11] Enze Liu, Bowen Zheng, Cheng Ling, Lantao Hu, Han Li, and Wayne Xin Zhao, "Generative recommender with end-to-end learnable item tokenization," 2025.

[12] Haohao Qu, Wenqi Fan, Zihuai Zhao, and Qing Li, "Tokenrec: Learning to tokenize id for llm-based generative recommendation," 2024.

[13] Jinming Li, Wentao Zhang, Tian Wang, Guanglei Xiong, Alan Lu, and Gerard Medioni, "Gpt4rec: A generative framework for personalized recommendation and user interests interpretation," *arXiv preprint arXiv:2304.03879*, 2023.

[14] Jesse Harte, Wouter Zorgdrager, Panos Louridas, Asterios Katsifodimos, Dietmar Jannach, and Marios Fragkoulis, "Leveraging large language models for sequential recommendation," in *Proceedings of the 17th ACM Conference on Recommender Systems*, 2023, pp. 1096–1102.

[15] Zhiming Mao, Huimin Wang, Yiming Du, and Kam-Fai Wong, "Unitrec: A unified text-to-text transformer and joint contrastive learning framework for text-based recommendation," *arXiv preprint arXiv:2305.15756*, 2023.

[16] Scott Sanner, Krisztian Balog, Filip Radlinski, Ben Wedin, and Lucas Dixon, "Large language models are competitive near cold-start recommenders for language-and item-based preferences," in *Proceedings of the 17th ACM conference on recommender systems*, 2023, pp. 890–896.

[17] Damien Sileo, Wout Vossen, and Robbe Raymaekers, "Zero-shot recommendation as language modeling," in *European conference on information retrieval*. Springer, 2022, pp. 223–230.

[18] Wenjie Wang, Honghui Bao, Xinyu Lin, Jizhi Zhang, Yongqi Li, Fuli Feng, See-Kiong Ng, and Tat-Seng Chua, "Learnable item tokenization for generative recommendation," in *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, 2024, pp. 2400–2409.

[19] Bowen Zheng, Hongyu Lu, Yu Chen, Wayne Xin Zhao, and Ji-Rong Wen, "Universal item tokenization for transferable generative recommendation," *arXiv preprint arXiv:2504.04405*, 2025.

[20] Jing Zhu, Mingxuan Ju, Yozen Liu, Danai Koutra, Neil Shah, and Tong Zhao, "Beyond unimodal boundaries: Generative recommendation with multimodal semantics," 2025.

[21] Aaron Van Den Oord, Oriol Vinyals, et al., "Neural discrete representation learning," *Advances in neural information processing systems*, vol. 30, 2017.

[22] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals, "Generating diverse high-fidelity images with vq-vae-2," *Advances in neural information processing systems*, vol. 32, 2019.

[23] Qwen Team, "Qwen2.5: A party of foundation models," September 2024.

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

[25] Yusong Wu, Ke Chen, Tianyu Zhang, Yuchen Hui, Taylor Berg-Kirkpatrick, and Shlomo Dubnov, "Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[26] Greg Linden, Brent Smith, and Jeremy York, "Amazon.com recommendations: Item-to-item collaborative filtering," *IEEE Internet computing*, vol. 7, no. 1, pp. 76–80, 2003.

[27] Milos Radovanovic, Alexandros Nanopoulos, and Mirjana Ivanovic, "Hubs in space: Popular nearest neighbors in high-dimensional data," *Journal of Machine Learning Research*, vol. 11, no. sept, pp. 2487–2531, 2010.

[28] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean, "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems*, vol. 26, 2013.

[29] Nils Reimers and Iryna Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. 11 2019, Association for Computational Linguistics.

[30] SeungHeon Doh, Keunwoo Choi, Daeyong Kwon, Taesu Kim, and Juhan Nam, "Music discovery dialogue generation using human intent analysis and large language models," *arXiv preprint arXiv:2411.07439*, 2024.

[31] Megan Leszczynski, Shu Zhang, Ravi Ganti, Krisztian Balog, Filip Radlinski, Fernando Pereira, and Arun Tejasvi Chaganty, "Talk the walk: Synthetic data generation for conversational music recommendation," *arXiv preprint arXiv:2301.11489*, 2023.

[32] Ilya Loshchilov and Frank Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.