04

Semantic ID

-based Generative Recommendation

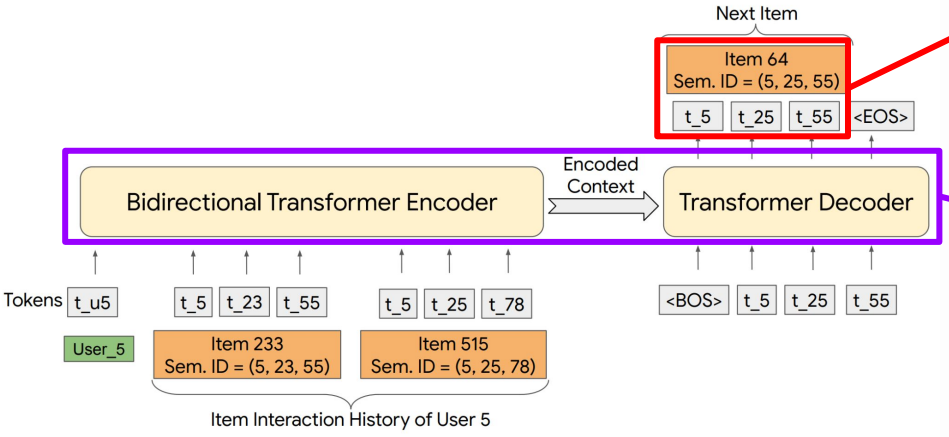
Semantic IDs

(also called: SemID or SID)

A few tokens that jointly index one item.

t3, t321, t643, t1011

SemID-based Generative Recommendation



Part 1:

How to construct SIDs

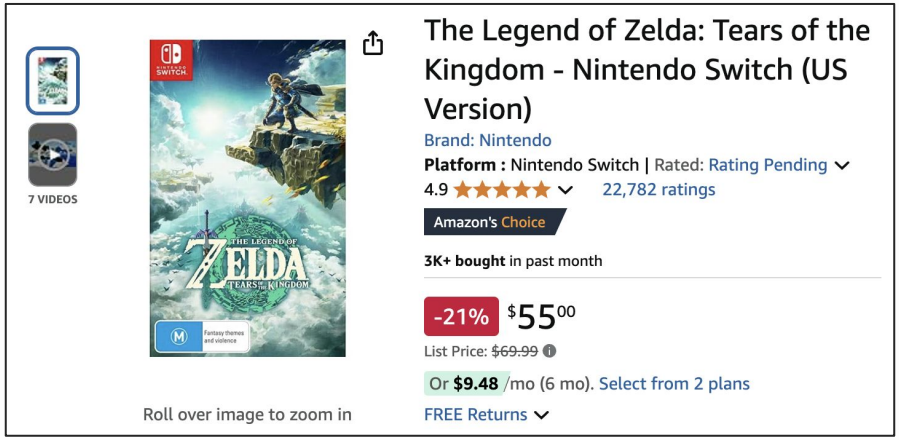
Part 2:

How to build SID-based Generative Rec Models

Part 1: Semantic ID Construction

Semantic ID Construction

Input: all data associated with the item (description, title, interactions, features, …)



Output: mapping between **items** ⇔ **Semantic IDs**

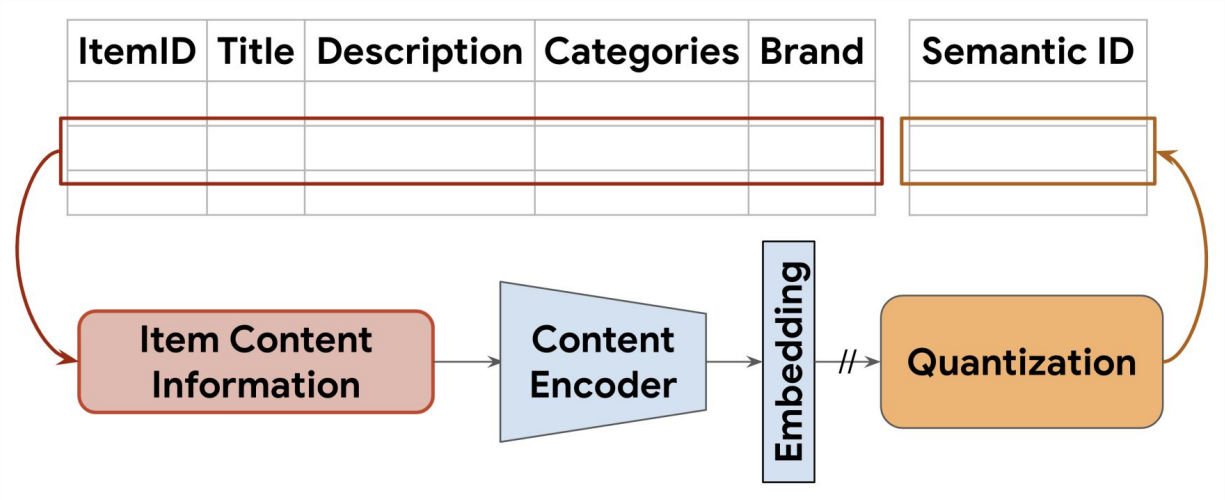
B097B2YWFX ⇔ {t3, t321, t643, t1011}

Part 1: Semantic ID Construction

(i) **First example**: TIGER and RQ-VAE-based SemIDs;

SemID Construction – First Example: TIGER

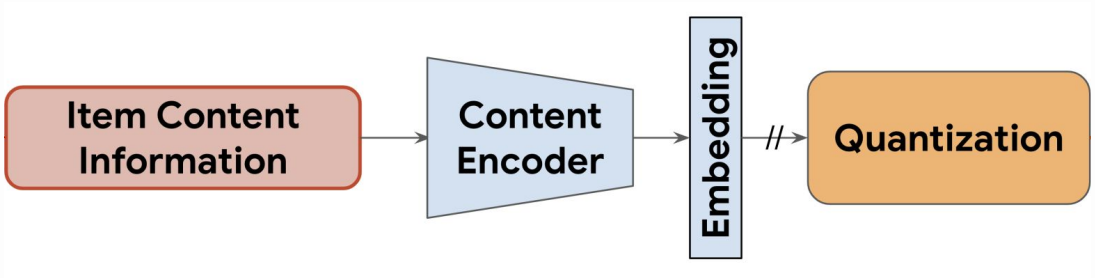
Input: concatenated text features



Output: mapping between **items** ⇔ **Semantic IDs**

SemID Construction – First Example: TIGER

Text ➤ Vector ➤ IDs



Rajput et al. Recommender Systems with Generative Retrieval. NeurIPS 2023.

SemID Construction – First Example: TIGER

1. Item Content Information (Text)

ItemID **Title** **Description**

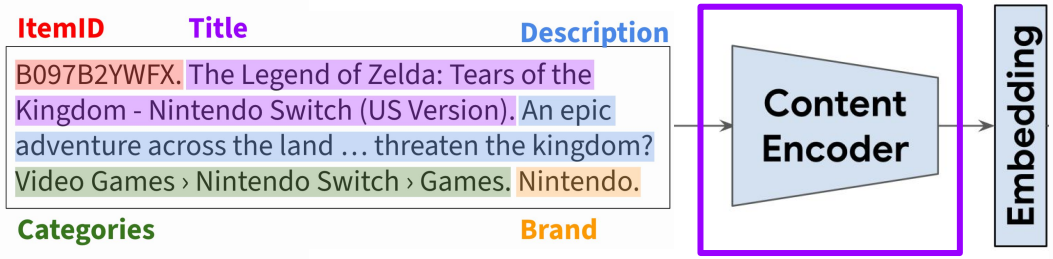
B097B2YWFX. The Legend of Zelda: Tears of the Kingdom - Nintendo Switch (US Version). An epic adventure across the land … threaten the kingdom? Video Games › Nintendo Switch › Games. Nintendo.

Categories **Brand**

SemID Construction – First Example: TIGER

2. Content Encoder + Embedding (Text ➤ Vector)

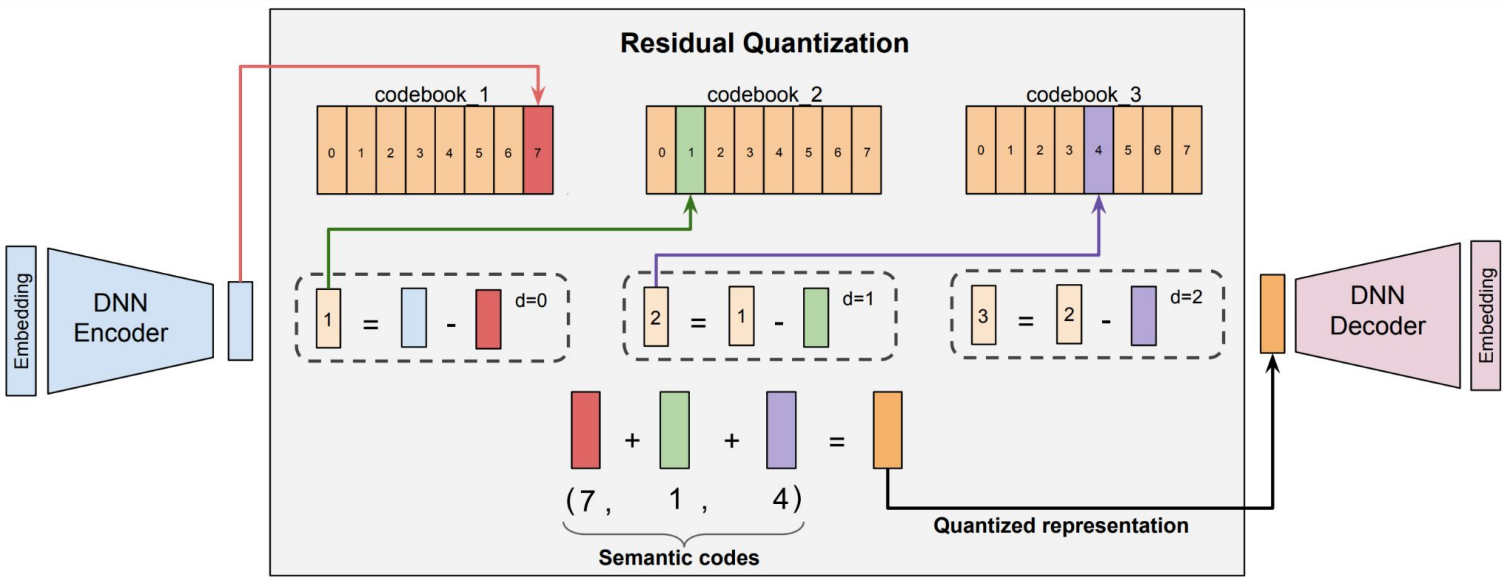
Pre-trained (fixed) sentence embedding model (**SentenceT5**)



Ni et al. Sentence-T5: Scalable Sentence Encoders from Pre-trained Text-to-Text Models. Findings of ACL 2022.
Rajput et al. Recommender Systems with Generative Retrieval. NeurIPS 2023.

SemID Construction – First Example: TIGER

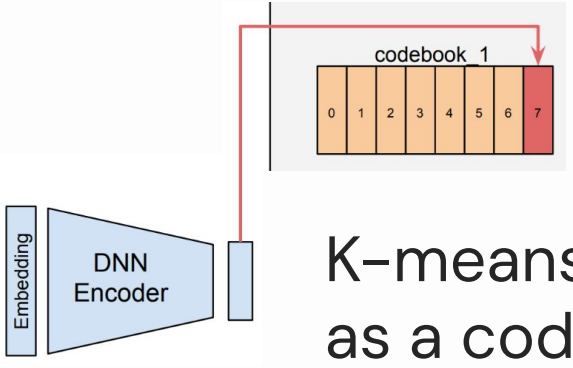
3. RQ-VAE Quantization (Vector ➤ IDs)



Zeghidour et al. SoundStream: An End-to-End Neural Audio Codec. TASLP 2022.
Rajput et al. Recommender Systems with Generative Retrieval. NeurIPS 2023.

SemID Construction – First Example: TIGER

3. RQ-VAE Quantization (Vector ➤ IDs)

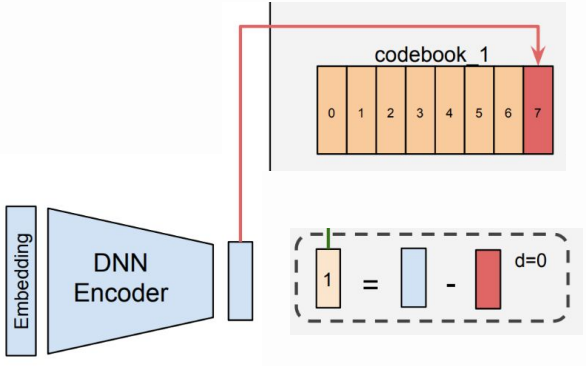


K-means: cluster center ID
as a code in the codebook

Zeghidour et al. SoundStream: An End-to-End Neural Audio Codec. TASLP 2022.
Rajput et al. Recommender Systems with Generative Retrieval. NeurIPS 2023.

SemID Construction – First Example: TIGER

3. RQ-VAE Quantization (Vector ➤ IDs)

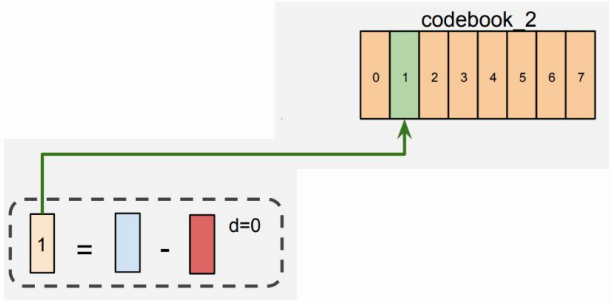


Residual of “input vector” and “clustering center vector”

Zeghidour et al. SoundStream: An End-to-End Neural Audio Codec. TASLP 2022.
Rajput et al. Recommender Systems with Generative Retrieval. NeurIPS 2023.

SemID Construction – First Example: TIGER

3. RQ-VAE Quantization (Vector ➤ IDs)

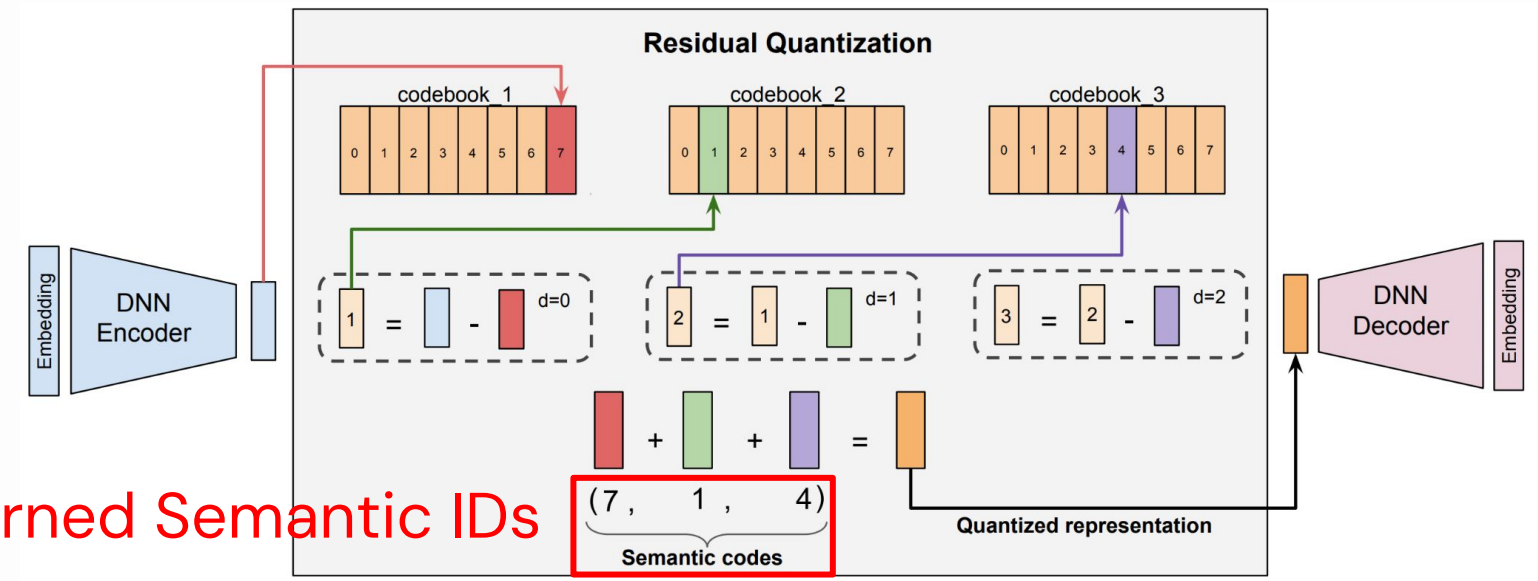


Residual as next level's input

Zeghidour et al. SoundStream: An End-to-End Neural Audio Codec. TASLP 2022.
Rajput et al. Recommender Systems with Generative Retrieval. NeurIPS 2023.

SemID Construction – First Example: TIGER

3. RQ-VAE Quantization (Vector ➤ IDs)



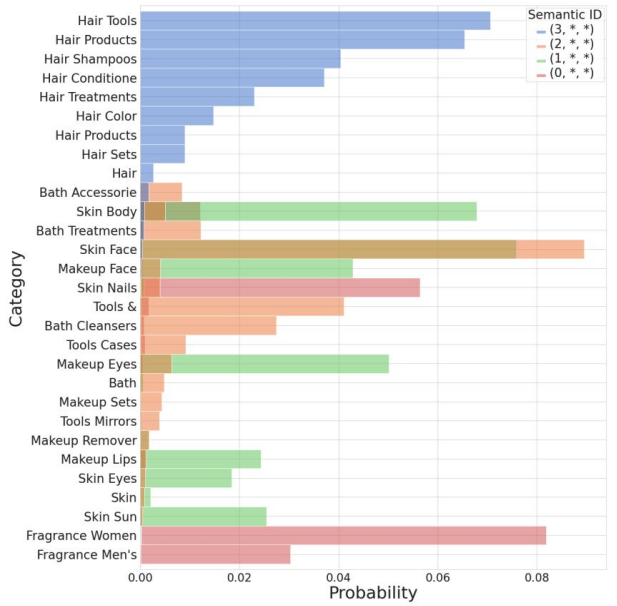
Learned Semantic IDs

Zeghidour et al. SoundStream: An End-to-End Neural Audio Codec. TASLP 2022.
Rajput et al. Recommender Systems with Generative Retrieval. NeurIPS 2023.

SemID Construction – First Example: TIGER

Properties of **RQ-VAE**-based SemIDs

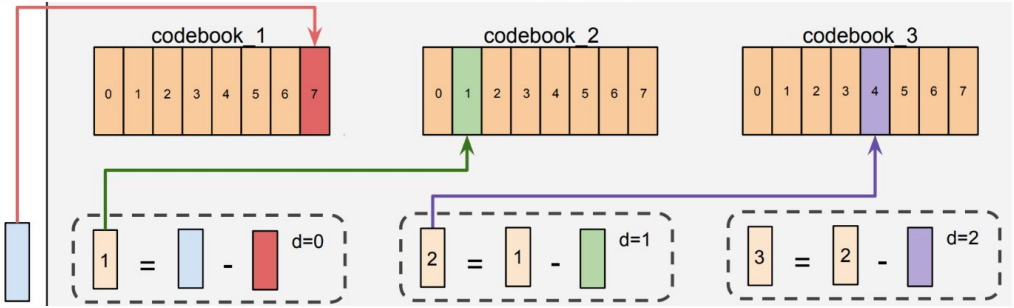
1. Semantic;



SemID Construction – First Example: TIGER

Properties of **RQ-VAE**-based SemIDs

1. Semantic;
2. Ordered / sequential dependent;



SemID Construction – First Example: TIGER

Collisions

(12, 24, 52)

(12, 24, 52)

SemID Construction – First Example: TIGER

Collisions

(12, 24, 52, 0)

(12, 24, 52, 1)

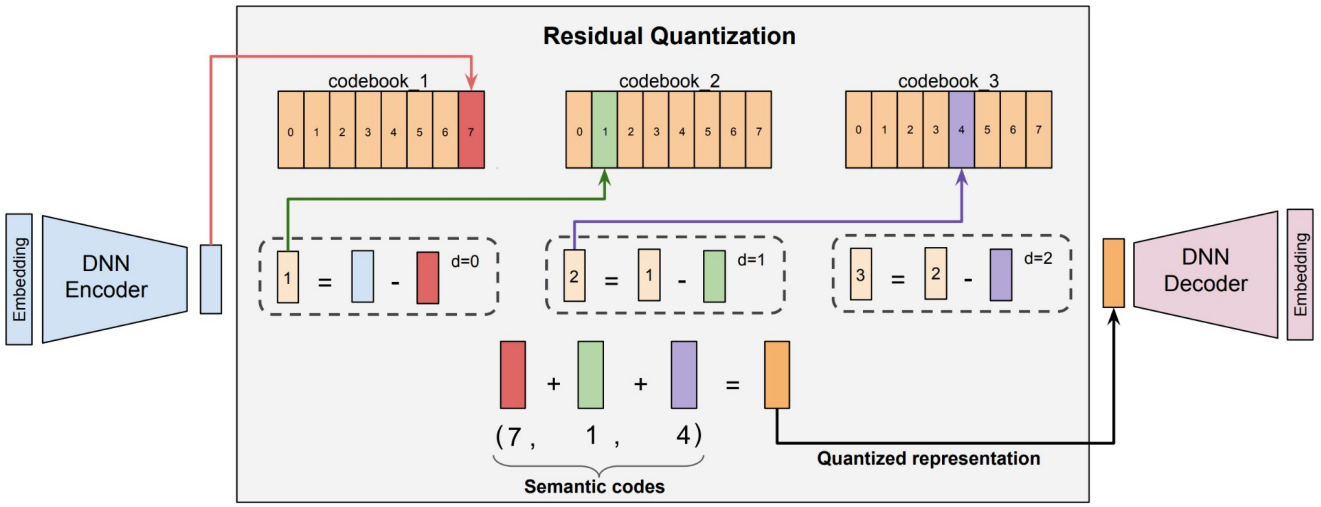
One extra token
to avoid conflicts

Part 1: Semantic ID Construction

(i) First example: TIGER and RQ-VAE-based SemIDs;
(ii) **Techniques** to construct SemIDs;

Techniques to Construct SemIDs

Residual Quantization: RQ-VAE



Techniques to Construct SemIDs

Residual Quantization: RQ-VAE

Issues:

- Enc-Dec Training Unstable
- Unbalanced IDs

Techniques to Construct SemIDs

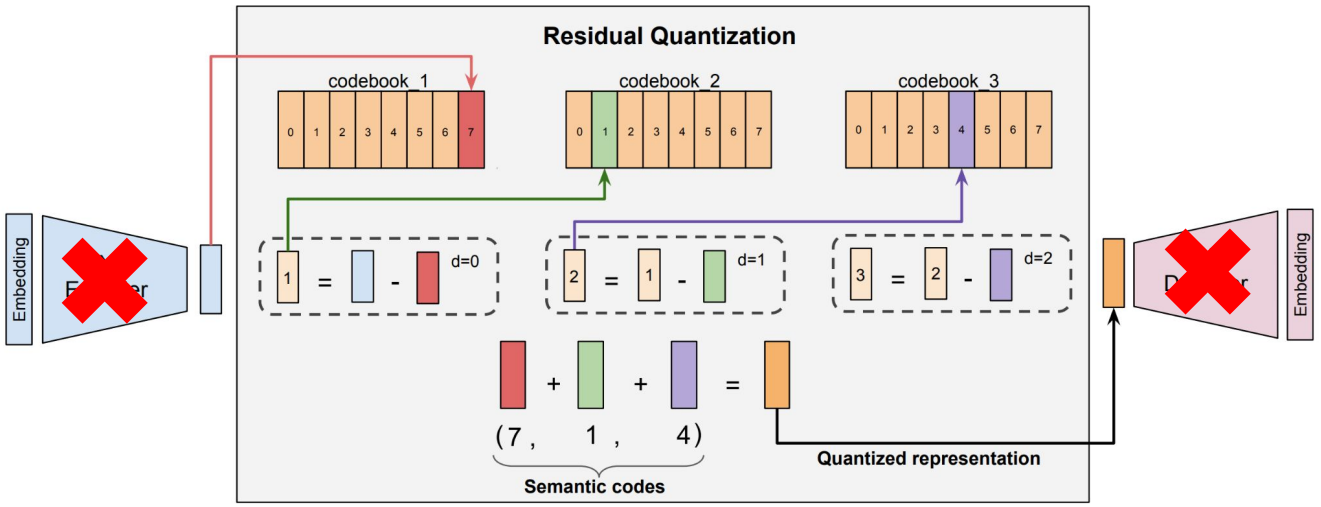
Residual Quantization: RQ-VAE

Issues:

- Enc-Dec Training Unstable
- Unbalanced IDs

Techniques to Construct SemIDs

Variants of Residual Quantization: RQ



Techniques to Construct SemIDs

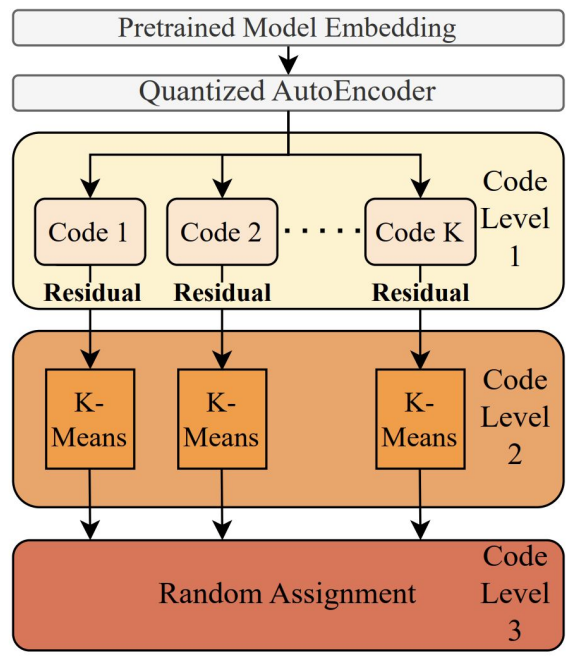
Residual Quantization: RQ-VAE

Issues:

- Enc-Dec Training Unstable
- Unbalanced IDs

Techniques to Construct SemIDs

Variants of Residual Quantization: VQ-VAE + K-Means



MBGen [CIKM'24]:

- Layer 1: VQ-VAE
- Layer 2: K-Means on residuals
- Layer 3: Random hash

Liu et al. Multi-Behavior Generative Recommendation. CIKM 2024.

Techniques to Construct SemIDs

Variants of Residual Quantization: RQ + K-Means

Algorithm 1: Balanced K-means Clustering

Input: Item set $\mathcal{V}$, number of clusters K

Compute $w \leftarrow |\mathcal{V}|/K$

Initialize centroids $C_l = \{\boldsymbol{c}_1^l, \ldots, \boldsymbol{c}_K^l\}$ with random selection;

repeat

  Initialize unassigned set $\mathcal{U} \leftarrow \mathcal{V}$

  for *each cluster* $k \in \{1, \ldots, K\}$ **do**

    Sort $\mathcal{U}$ by ascending distance from centroid $\boldsymbol{c}_k^l$;

    Assign $\mathcal{V}_k \leftarrow \mathcal{U}[0 : w-1]$;

    Update centroid $\boldsymbol{c}_k^l \leftarrow \frac{1}{w} \sum_{\boldsymbol{r}^l \in \mathcal{V}_k} \boldsymbol{r}^l$;

    Remove assigned items $\mathcal{U} \leftarrow \mathcal{U} \setminus \mathcal{V}_k$;

  end

until *Assignment convergence*;

Output: Optimized codebook $C_l = \{\boldsymbol{c}_1^l, \ldots, \boldsymbol{c}_K^l\}$

OneRec:

Limit the max #items per cluster

Deng et al. OneRec: Unifying Retrieve and Rank with Generative Recommender and Iterative Preference Alignment. arXiv:2502.18965.

Techniques to Construct SemIDs

Variants of Residual Quantization

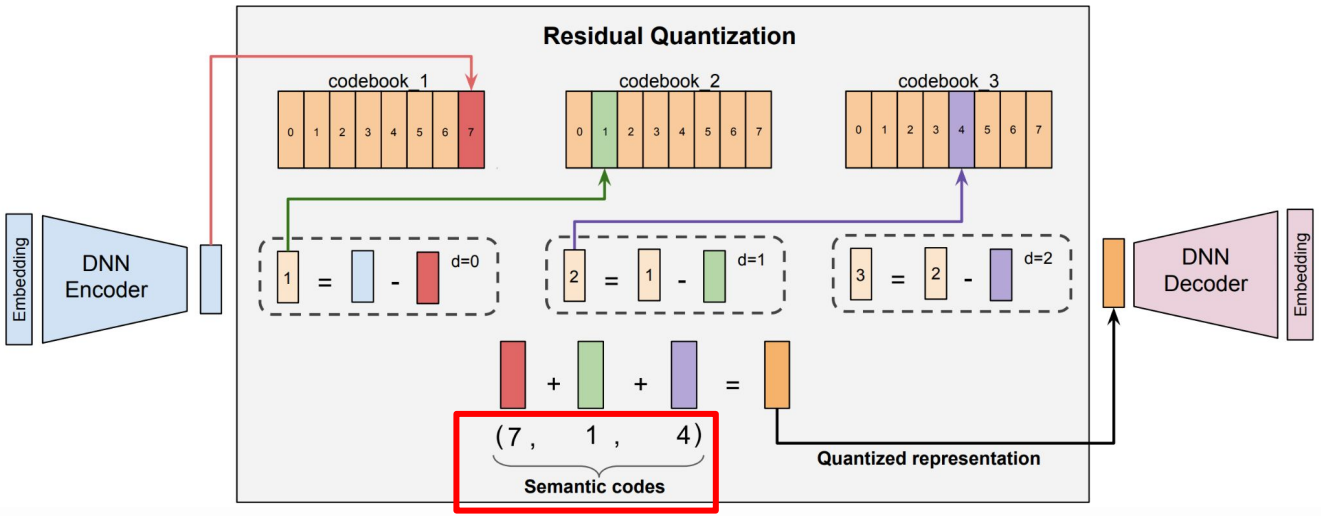
Table 1: Performance of GR models with by SIDs generated by different tokenization algorithms.

Methods	Recommendation Performance (Recall@5/Recall@10/NDCG@5/NDCG@10)											
	Beauty				Toys				Sports			
RK-Means	**0.0422**	**0.0639**	0.0277	0.0347	**0.0376**	**0.0577**	**0.0243**	**0.0308**	**0.0236**	**0.0353**	**0.0153**	**0.0191**
R-VQ	**0.0422**	0.0638	**0.0282**	**0.0351**	0.0327	0.0493	0.0209	0.0262	0.0234	0.0352	0.0151	0.0189
RQ-VAE	0.0404	0.0593	0.0268	0.0329	0.0342	0.0514	0.0224	0.0280	0.0205	0.0312	0.0132	0.0166

https://github.com/snap-research/GRID

Techniques to Construct SemIDs

Are the orders among tokens necessary?



Techniques to Construct SemIDs

Are the orders among tokens necessary?

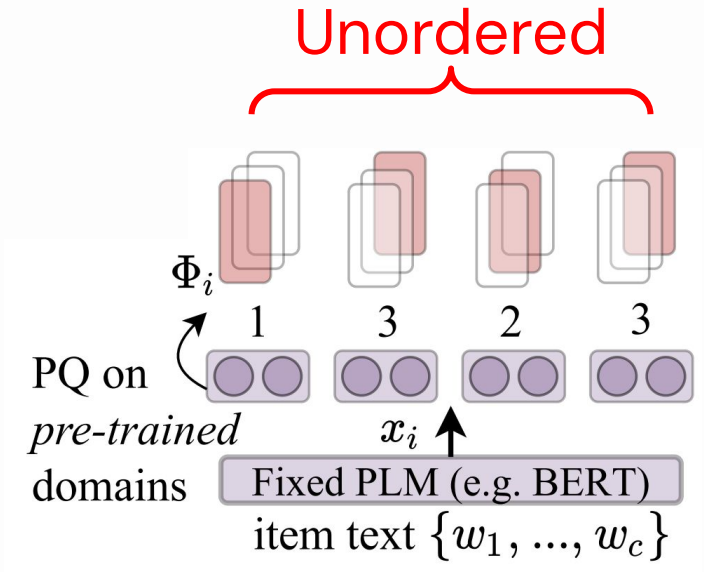
Orders

<-> autoregressive generation

<-> SemIDs have to be short

Techniques to Construct SemIDs

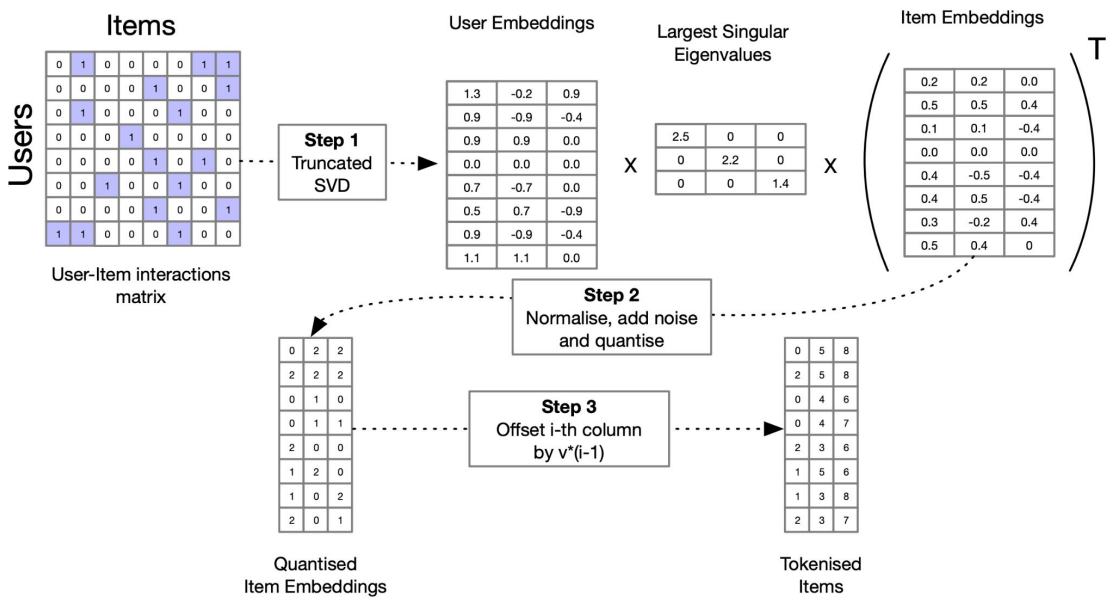
Product Quantization



Hou et al. Learning Vector-Quantized Item Representation for Transferable Sequential Recommenders. WWW 2023.

Techniques to Construct SemIDs

Product Quantization



Early exploration on PQ-based SIDs, but performance is not ideal

Autoregressive generation needs orders

Petrov et al. Generative Sequential Recommendation with GPTRec. Gen-IR@SIGIR 2023.

Techniques to Construct SemIDs

Product Quantization: RPG [KDD'25]

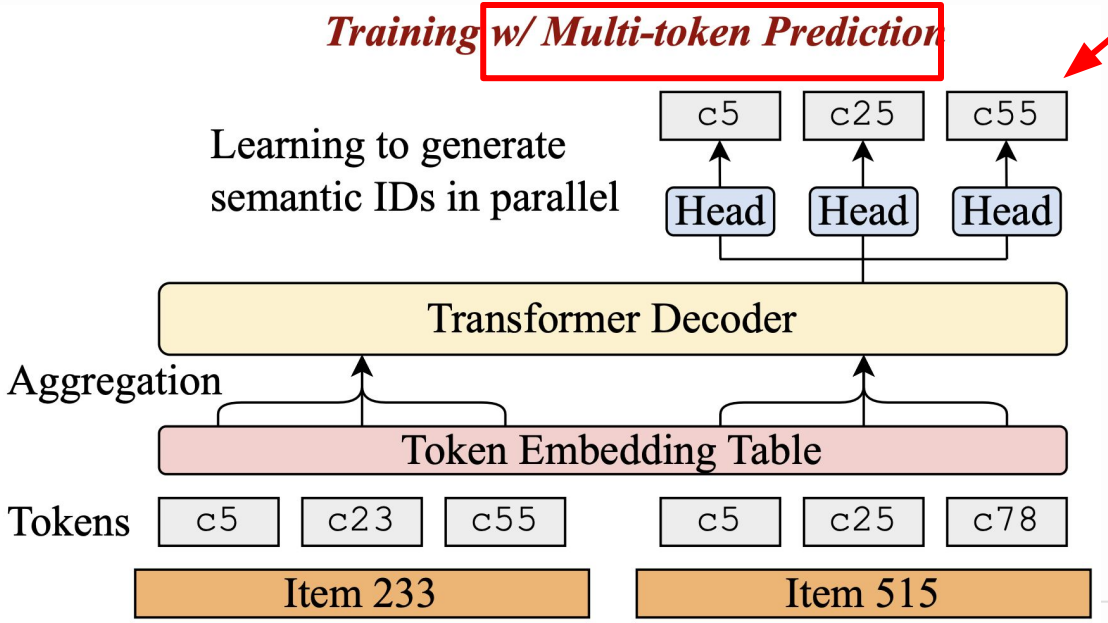
Long semantic IDs + parallel generation

Original (#digit=4): t34, t392, t600, t891

RPG (#digit=64): t34, t392, t600, t891, t1221, t1495, t1556, t1924, t2296, t2511, t2715, t3060, t3255, t3439, t3655, t3984, t4167, t4400, t4736, t4939, t5278, t5426, t5669, t6057, t6385, t6451, t6837, t7134, t7329, t7528, t7924, t8162, t8325, t8479, t8711, t9007, t9420, t9472, t9980, t10154, t10364, t10662, t10784, t11105, t11377, t11642, t11848, t12261, t12334, t12585, t12963, t13306, t13367, t13722, t13973, t14143, t14506, t14696, t14995, t15331, t15406, t15813, t16034, t16251

Techniques to Construct SemIDs

Product Quantization: RPG [KDD'25]

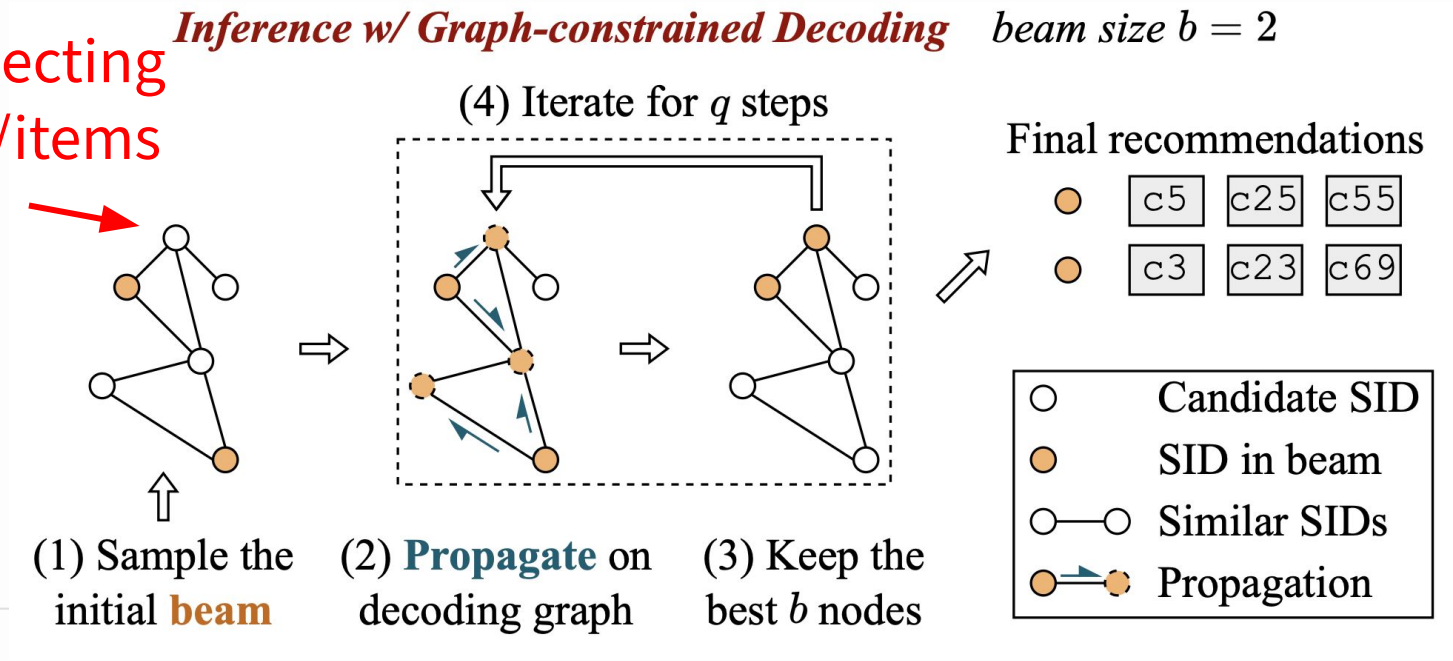


Using *OPQ* to produce semIDs w/o sequential dependencies

Techniques to Construct SemIDs

Product Quantization: RPG [KDD'25]

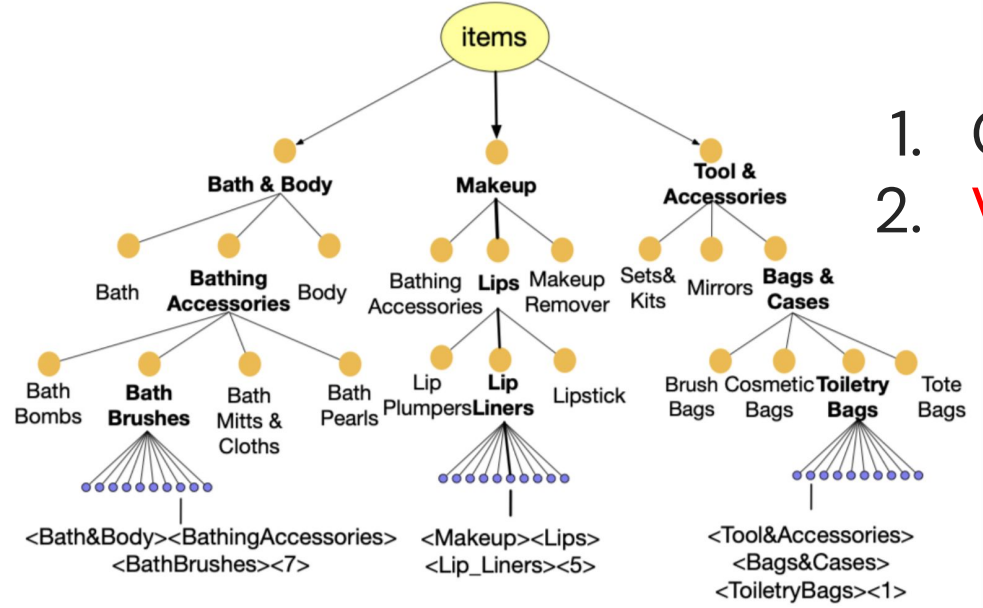
Graph: connecting similar SIDs/items (offline)



Hou et al. Generating Long Semantic IDs in Parallel for Recommendation. KDD 2025.

Techniques to Construct SemIDs

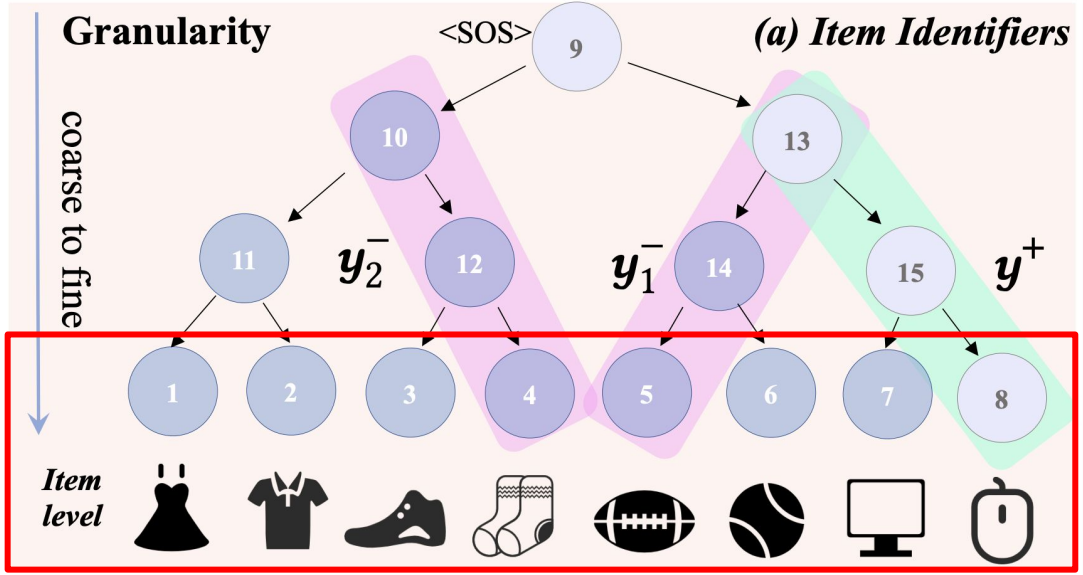
Hierarchical Clustering (Heuristics-based)



1. Ordered;
2. Variable-length SemIDs;

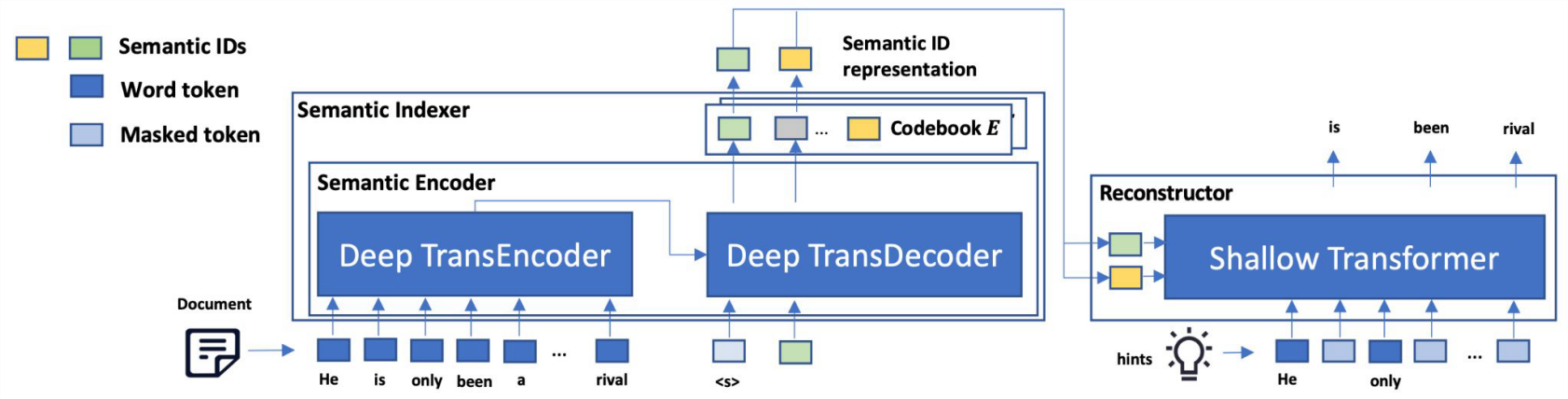
Techniques to Construct SemIDs

Hierarchical Clustering (Latent-based)



Techniques to Construct SemIDs

Language Model-based ID Generator

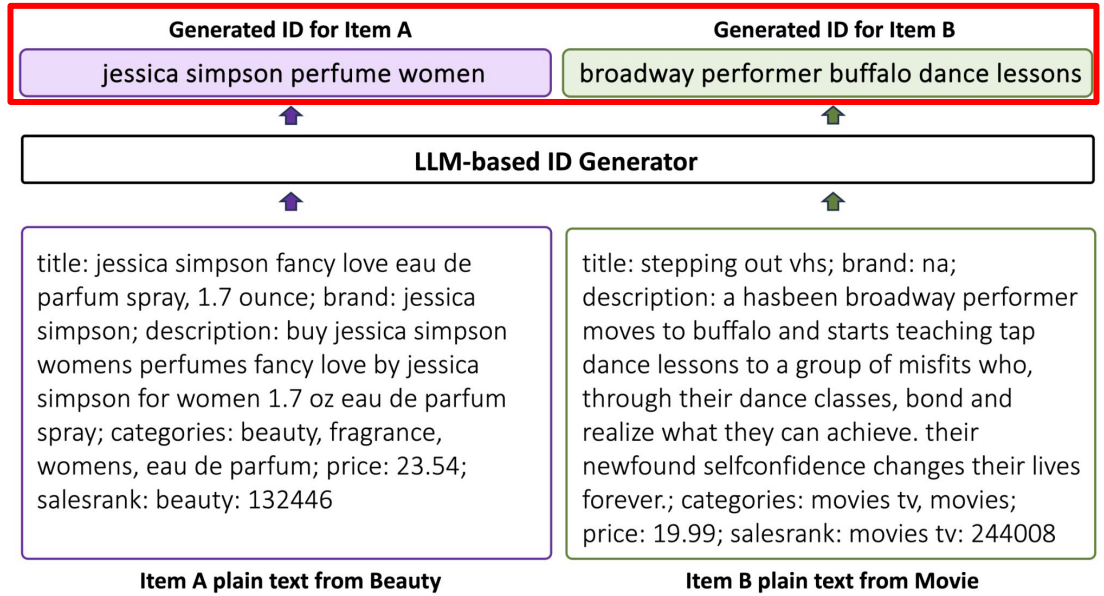


Natural language as inputs;
SemIDs as outputs

Jin et al. Language Models as Semantic Indexers. ICML 2024.

Techniques to Construct SemIDs

Language Model-based ID Generator



Words as SemIDs (like tagging)

Tan et al. IDGenRec: LLM-RecSys Alignment with Textual ID Learning. SIGIR 2024.

Summary of
Techniques to Construct SemIDs

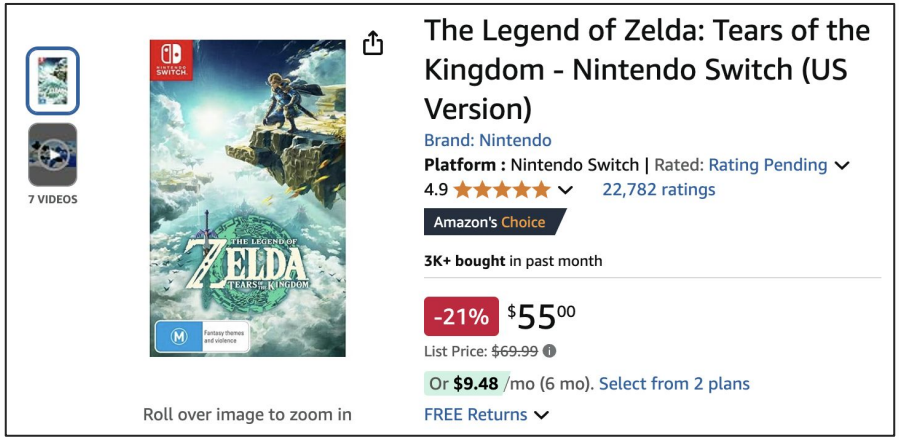
- Residual Quantization (ordered)
- Product Quantization (unordered)
- Hierarchical Clustering
- LM-based ID Generator

Part 1: Semantic ID Construction

(i) First example: TIGER and RQ-VAE-based SemIDs;
(ii) Techniques to construct SemIDs;
(iii) **Inputs** for SemID construction;

Inputs for SemID Construction

Input: all data associated with the item



Inputs for SemID Construction

Input: all data associated with the item

What exactly does “all data” mean? 🤷

Inputs for SemID Construction

Text or Multimodal Features

Text/Visual/Acoustic ➤ **Vector** ➤ **IDs**

Pretrained Encoder Quantization

ItemID **Title** **Description**

B097B2YWFX. The Legend of Zelda: Tears of the Kingdom - Nintendo Switch (US Version). An epic adventure across the land … threaten the kingdom? Video Games › Nintendo Switch › Games. Nintendo.

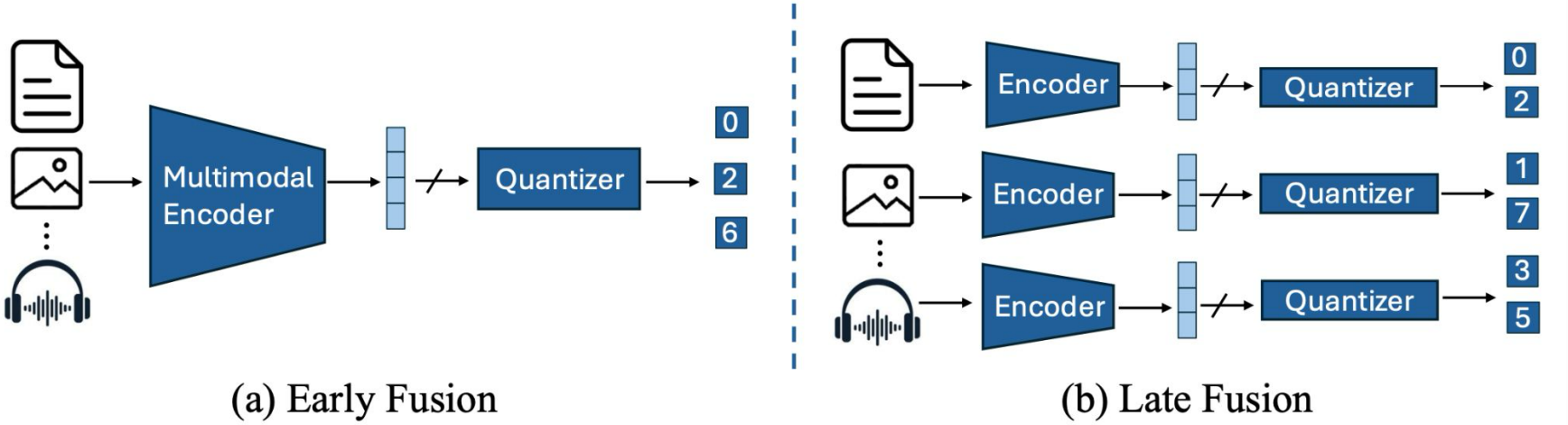
Categories **Brand**

Text

Multimodal

Inputs for SemID Construction

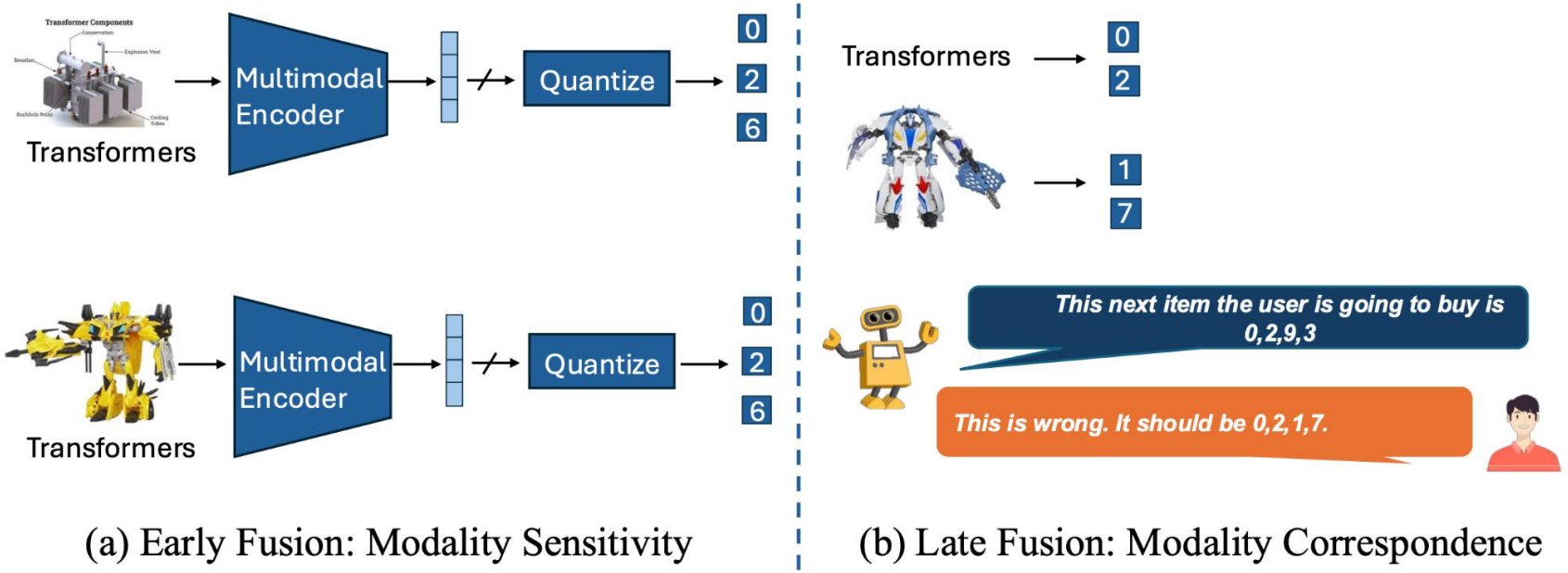
Text or Multimodal Features



Zhu et al. Beyond Unimodal Boundaries: Generative Recommendation with Multimodal Semantics. arXiv:2503.23333.

Inputs for SemID Construction

Text or Multimodal Features



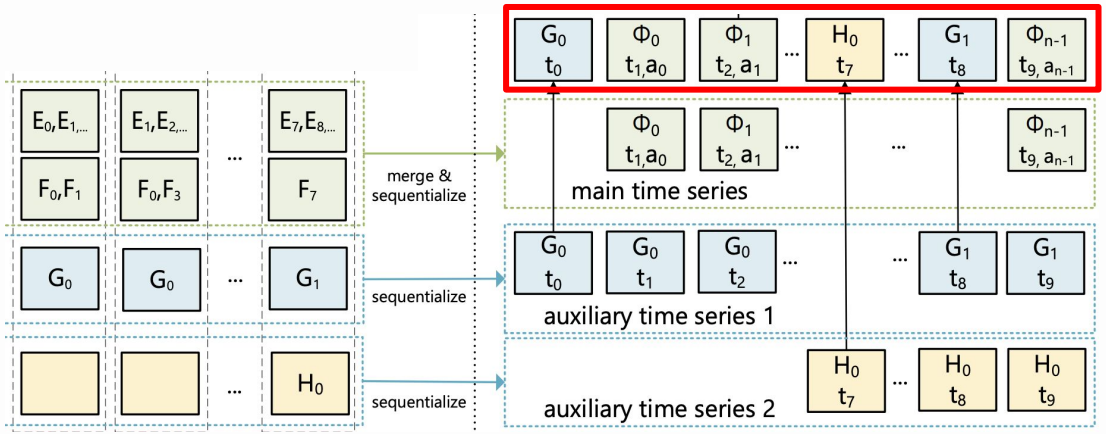
Zhu et al. Beyond Unimodal Boundaries: Generative Recommendation with Multimodal Semantics. arXiv:2503.23333.

Inputs for SemID Construction

Categorical Features

Categorical Features ➤ IDs

Merge & Sequentialize



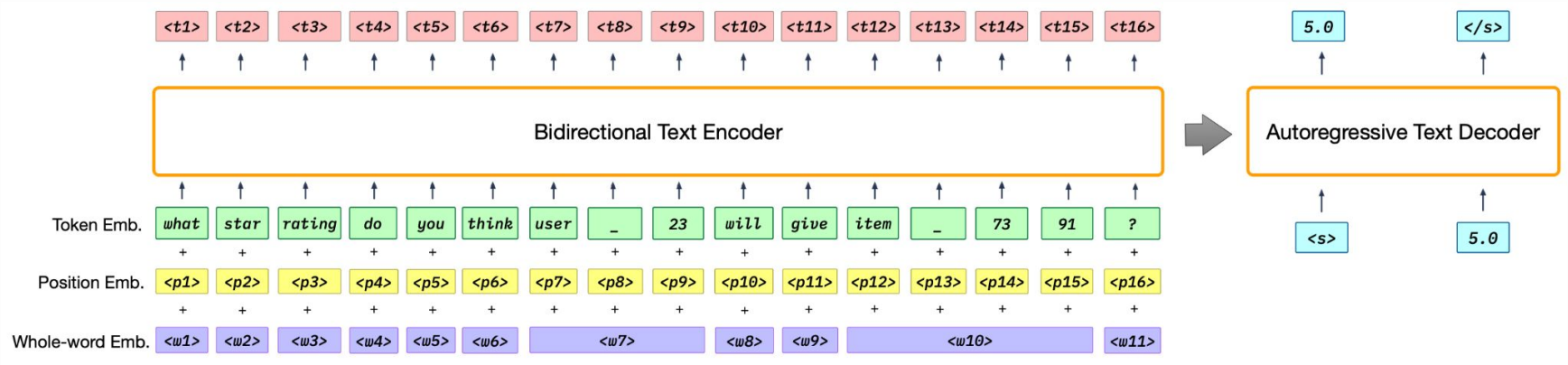
Zhai et al. Actions Speak Louder than Words: Trillion-Parameter Sequential Transducers for Generative Recommendations. ICML 2024.

Inputs for SemID Construction

No Features

Item ID ➤ **IDs**

Text Tokenizer



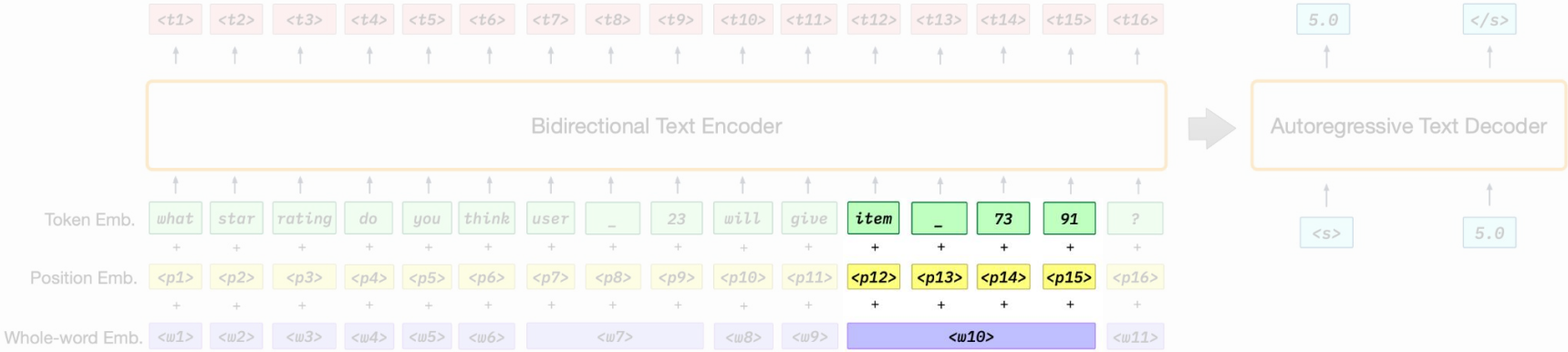
Geng et al. Recommendation as Language Processing (RLP): A Unified Pretrain, Personalized Prompt & Predict Paradigm (P5). RecSys 2022.

Inputs for SemID Construction

No Features

Item ID ➤ **IDs**

Text Tokenizer

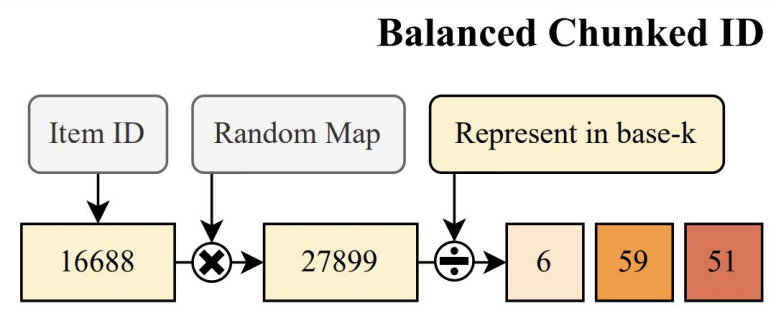


Geng et al. Recommendation as Language Processing (RLP): A Unified Pretrain, Personalized Prompt & Predict Paradigm (P5). RecSys 2022.

Inputs for SemID Construction

No Features

Random IDs



Inputs for SemID Construction

Input: all data associated with the item

(1) Item Metadata

Text / Multimodal / Categorical / No Features

Inputs for SemID Construction

Input: all data associated with the item

(1) Item Metadata

Text / Multimodal / Categorical / No Features

(2) Item Metadata + Behaviors

Inputs for SemID Construction

Input: all data associated with the item

(1) Item Metadata

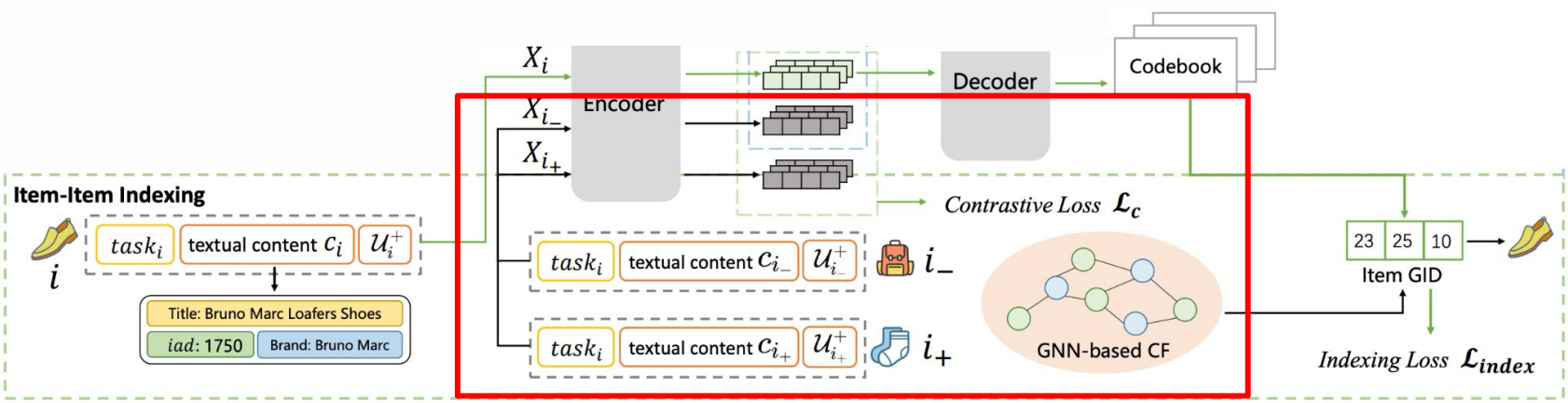
Text / Multimodal / Categorical / No Features

(2) Item Metadata + Behaviors

But how?

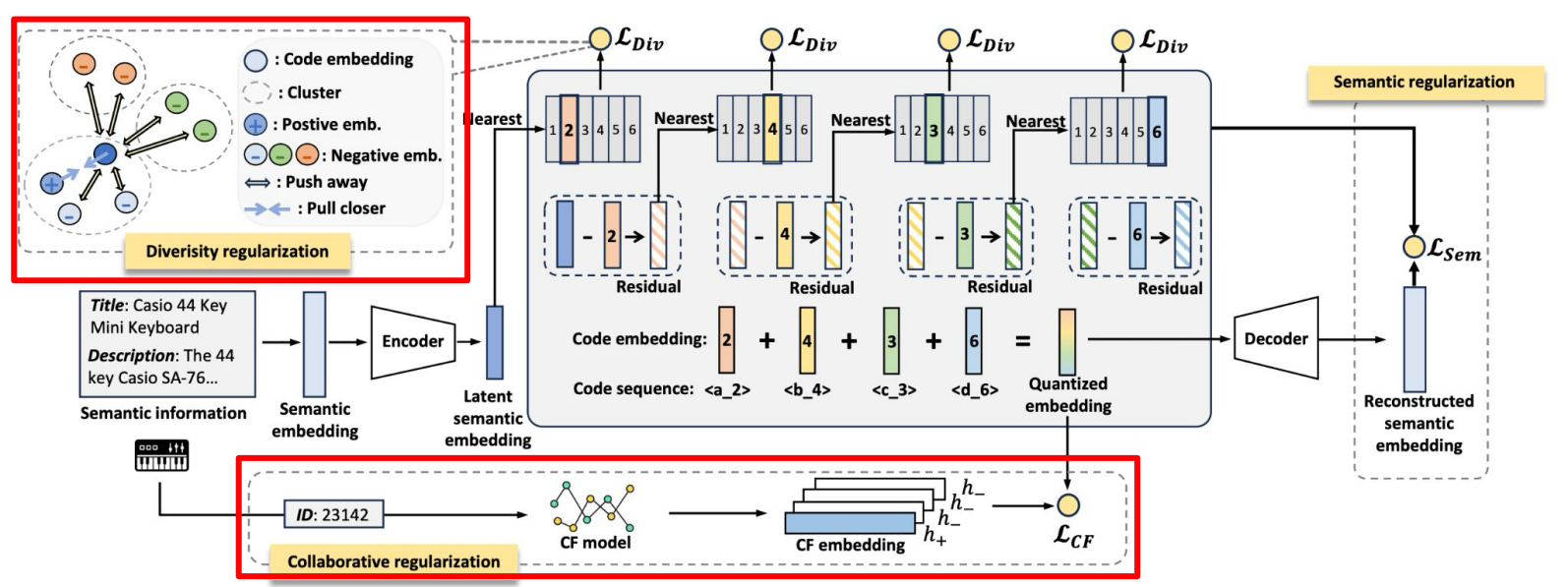
Inputs for SemID Construction

Residual Quantization + Item-level Regularization



Inputs for SemID Construction

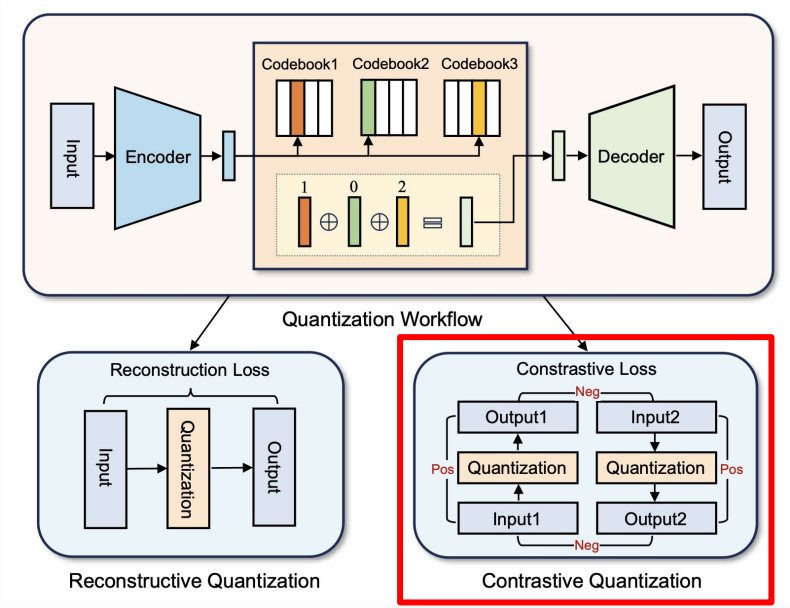
Residual Quantization + Item-level Regularization



Wang et al. Learnable Item Tokenization for Generative Recommendation. CIKM 2024.

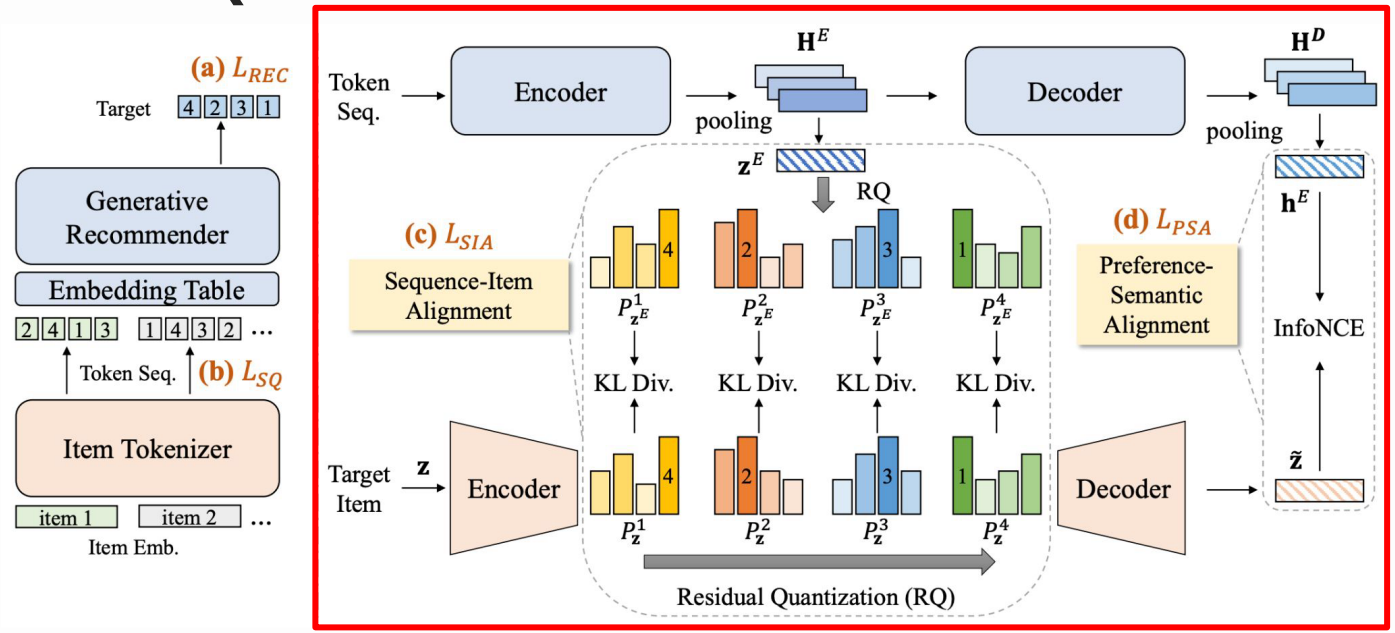
Inputs for SemID Construction

Residual Quantization + Item-level Regularization



Inputs for SemID Construction

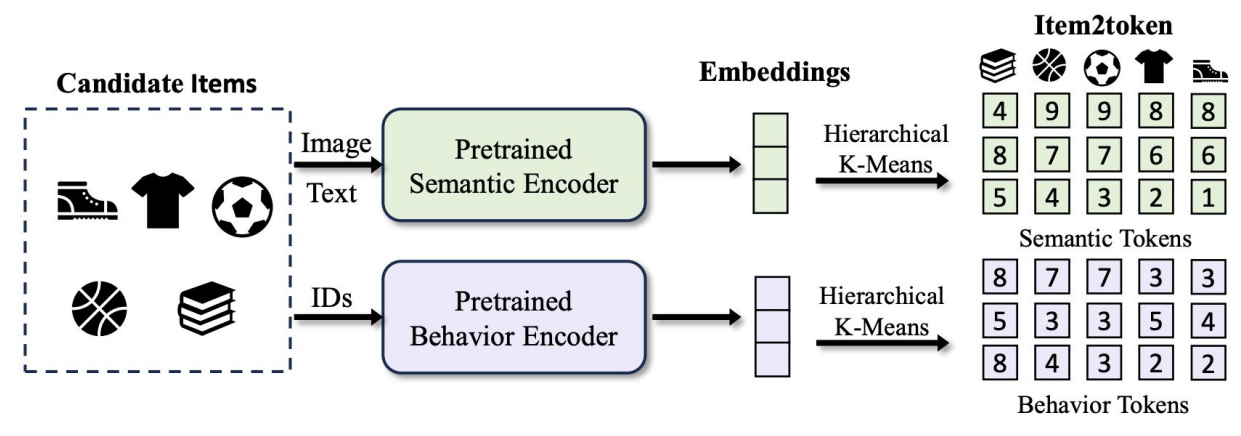
Residual Quantization + Recommendation Loss



Inputs for SemID Construction

Item Metadata + Behaviors

Fused Semantic IDs

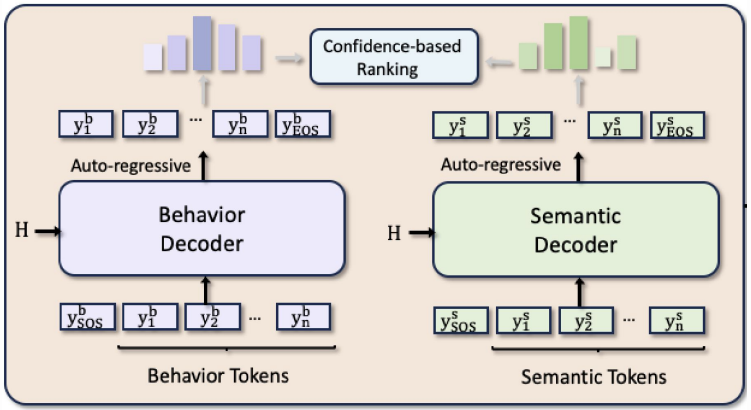


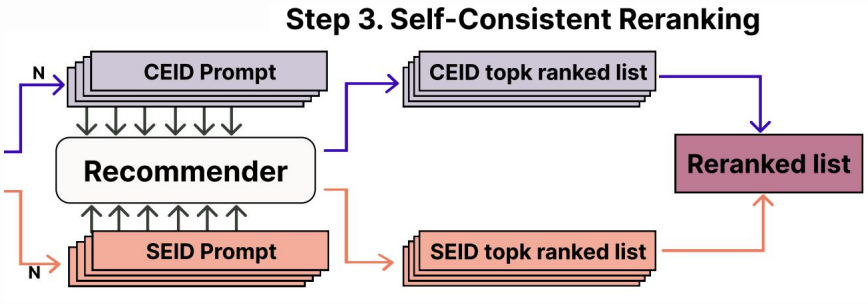
Wang et al. EAGER: Two-Stream Generative Recommender with Behavior-Semantic Collaboration. KDD 2024.

Inputs for SemID Construction

Item Metadata + Behaviors

Fused Semantic IDs + Two-stream Generation





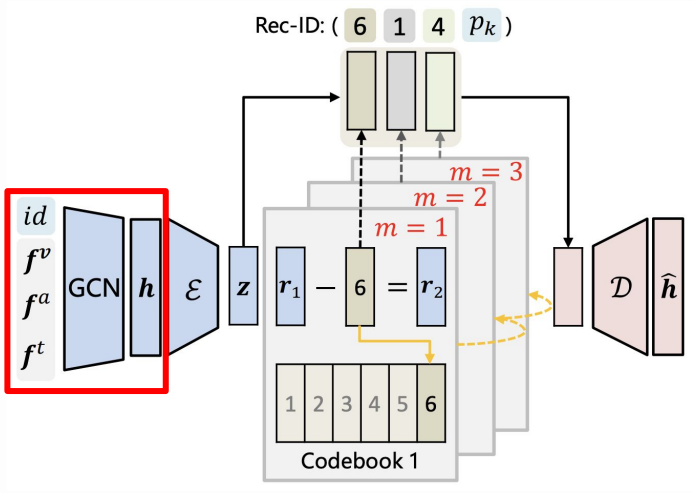
Wang et al. EAGER: Two-Stream Generative Recommender with Behavior-Semantic Collaboration. KDD 2024.
Kim et al. SC-Rec: Enhancing Generative Retrieval with Self-Consistent Reranking for Sequential Recommendation. arXiv:2408:08686.

Inputs for SemID Construction

Item Metadata + Behaviors

Fused Representations

User–Item Graph +
Semantic Features

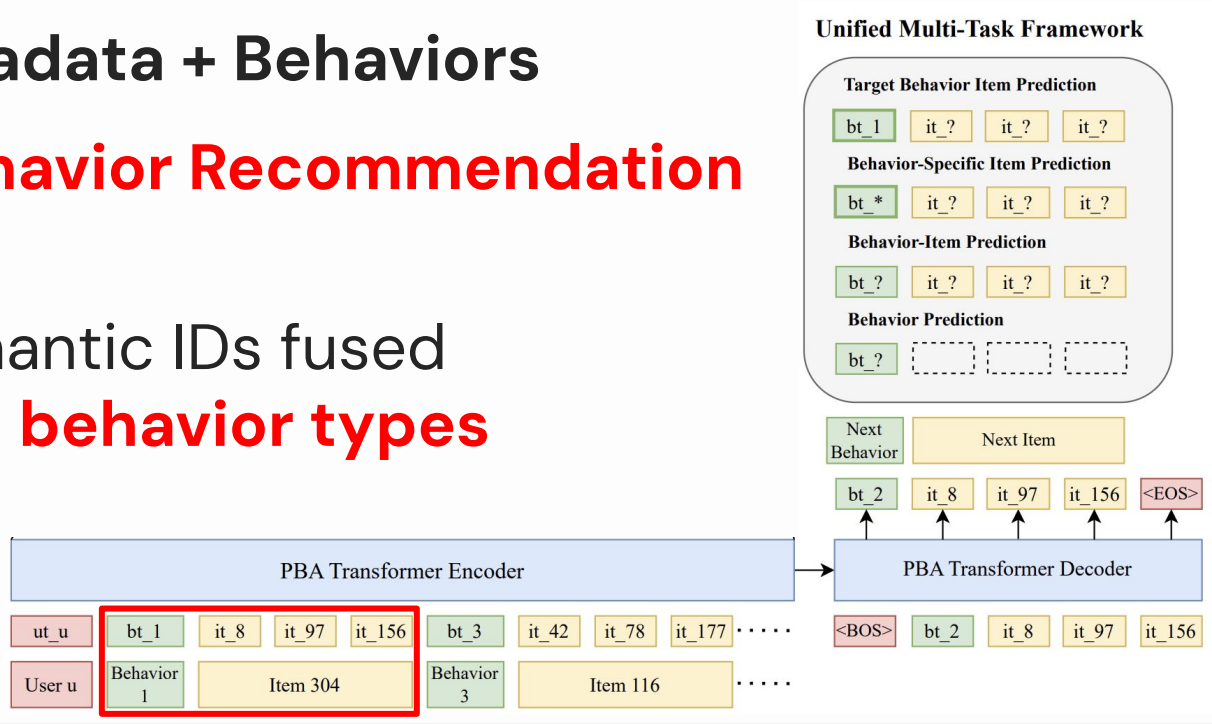


Inputs for SemID Construction

Item Metadata + Behaviors

Multi-Behavior Recommendation

Semantic IDs fused
with **behavior types**



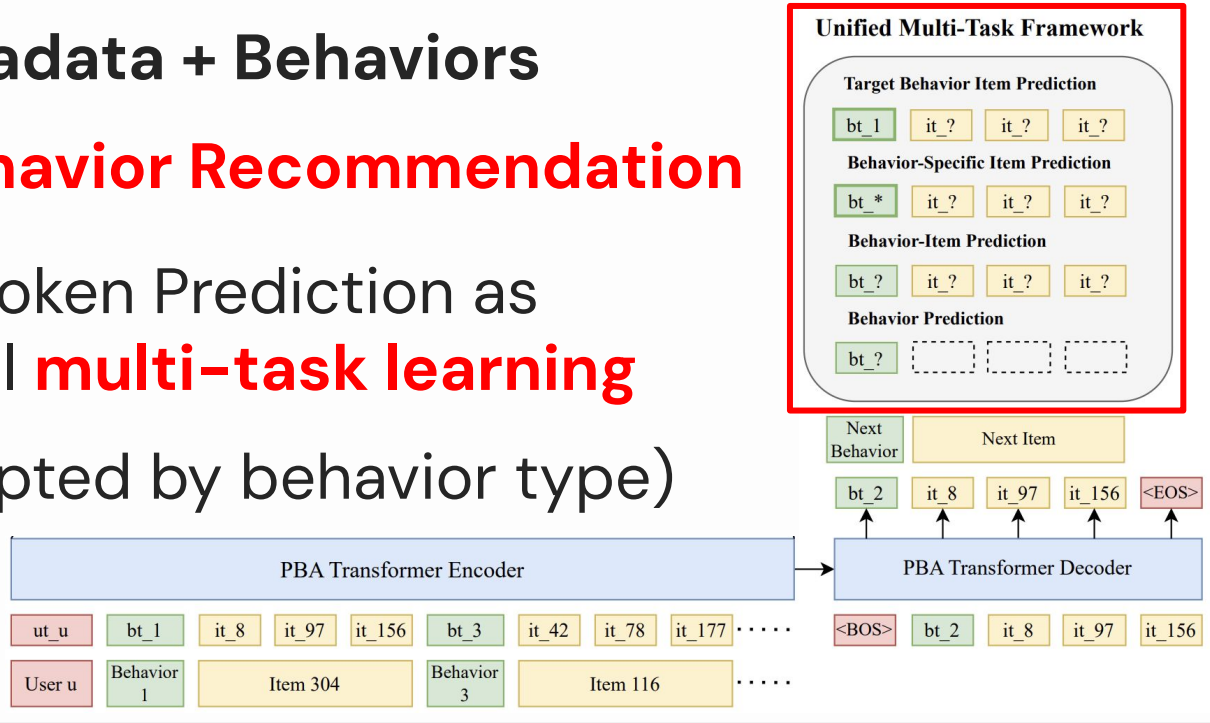
Inputs for SemID Construction

Item Metadata + Behaviors

Multi-Behavior Recommendation

Next Token Prediction as natural **multi-task learning**

(prompted by behavior type)



Inputs for SemID Construction

Context-independent

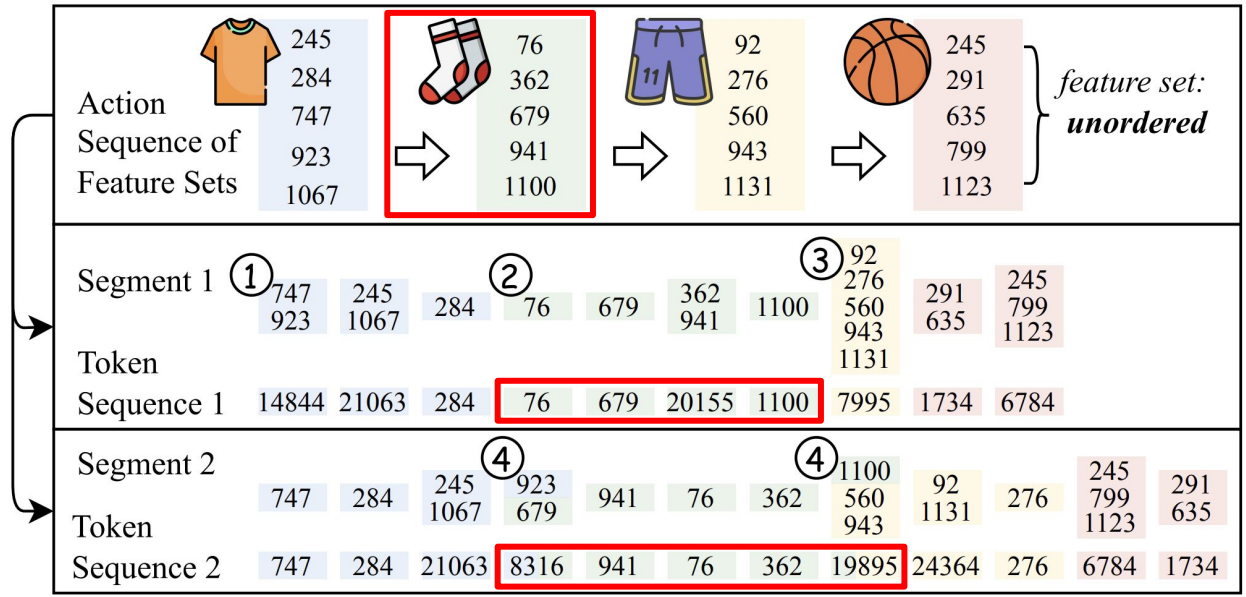
Action Tokenization	Example	Contextual	Unordered
Product Quantization	VQ-Rec (Hou et al., 2023)	✗	✓
Hierarchical Clustering	P5-CID (Hua et al., 2023)	✗	✗
Residual Quantization	TIGER (Rajput et al., 2023)	✗	✗
Text Tokenization	LMIndexer (Jin et al., 2024)	✗	✗
Raw Features	HSTU (Zhai et al., 2024)	✗	✗
SentencePiece	SPM-SID (Singh et al., 2024)	✗	✗

Same item ⇒ fixed semIDs in all sequences

Hou et al. ActionPiece: Contextually Tokenizing Action Sequences for Generative Recommendation. ICML 2025.

Inputs for SemID Construction

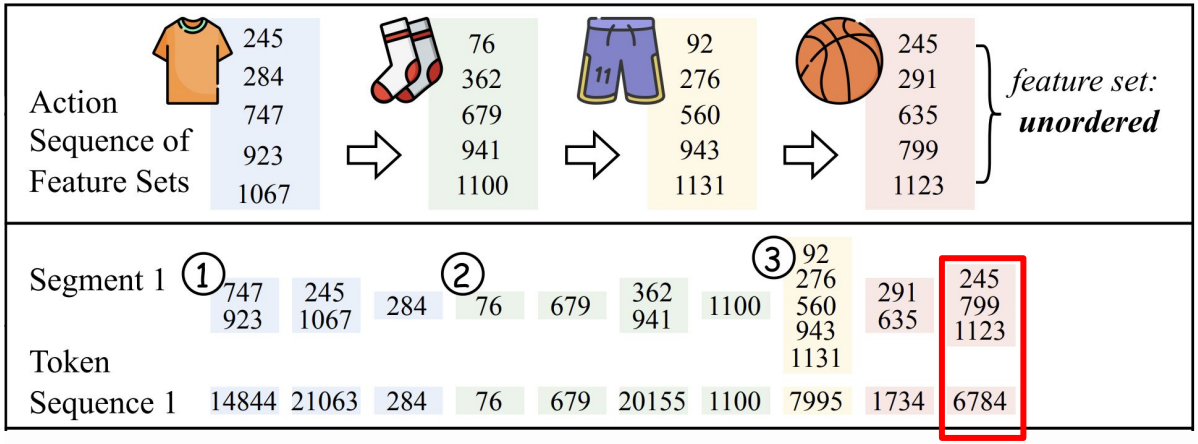
Context-independent ⇒ Context-aware



Same item ⇒ different semIDs based on context

Inputs for SemID Construction

Context-independent ⇒ Context-aware



Core Idea:

Merge frequently co-occurring features as new tokens

(**ActionPiece**: "WordPiece" tokenization for generative rec)

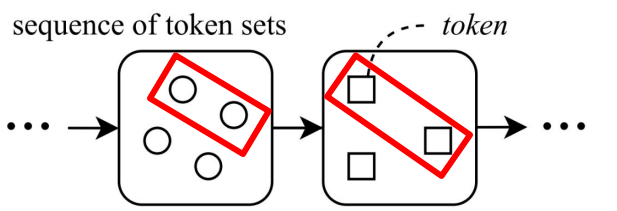
Inputs for SemID Construction

Context-independent ⇒ Context-aware

Algorithm 1 ActionPiece Vocabulary Construction

input Sequence corpus $\mathcal{S}'$, initial tokens $\mathcal{V}_0$, target size Q
output Merge rules $\mathcal{R}$, constructed vocabulary $\mathcal{V}$

Initialize vocabulary $\mathcal{V} \leftarrow \mathcal{V}_0$ # each initial token corresponds to one unique item feature
$\mathcal{R} \leftarrow \emptyset$
while $|\mathcal{V}| < Q$ **do**
  # ***Count:*** accumulate weighted token co-occurrences
  $\text{count}(\cdot,\cdot) \leftarrow \text{Count}(\mathcal{S}', \mathcal{V})$ # Algorithm 2
  # ***Update:*** Merge a frequent token pair into a new token
  Select $(c_u, c_v) \leftarrow \arg\max_{(c_i,c_j)} \text{count}(c_i, c_j)$
  $\mathcal{S}' \leftarrow \text{Update}(\mathcal{S}', \{(c_u, c_v) \rightarrow c_{\text{new}}\})$ # Algorithm 3
  $\mathcal{R} \leftarrow \mathcal{R} \cup \{(c_u, c_v) \rightarrow c_{\text{new}}\}$ # new merge rule
  $\mathcal{V} \leftarrow \mathcal{V} \cup \{c_{\text{new}}\}$ # add new token to the vocabulary
end while
return $\mathcal{R}, \mathcal{V}$



$$P(\bigcirc,\bigcirc) = \frac{|\bigcirc\!-\!\bigcirc|}{|<\bigcirc,\bigcirc>|} = \frac{4-1}{\binom{4}{2}}$$

$$P(\square,\square) = \frac{|\square\!-\!\square|}{|<\square,\square>|} = \frac{3-1}{\binom{3}{2}}$$

Features co-occurring within items

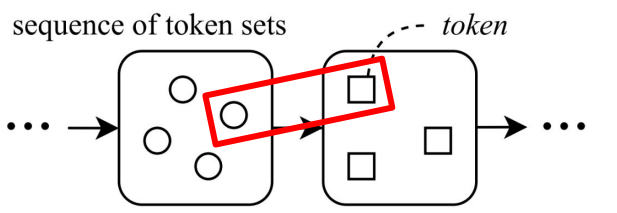
Inputs for SemID Construction

Context-independent ⇒ Context-aware

Algorithm 1 ActionPiece Vocabulary Construction

input Sequence corpus $\mathcal{S}'$, initial tokens $\mathcal{V}_0$, target size Q
output Merge rules $\mathcal{R}$, constructed vocabulary $\mathcal{V}$

Initialize vocabulary $\mathcal{V} \leftarrow \mathcal{V}_0$ # each initial token corresponds to one unique item feature
$\mathcal{R} \leftarrow \emptyset$
while $|\mathcal{V}| < Q$ **do**
  # ***Count:*** accumulate weighted token co-occurrences
  $\text{count}(\cdot,\cdot) \leftarrow \text{Count}(\mathcal{S}', \mathcal{V})$ # Algorithm 2
  # ***Update:*** Merge a frequent token pair into a new token
  Select $(c_u, c_v) \leftarrow \arg\max_{(c_i,c_j)} \text{count}(c_i, c_j)$
  $\mathcal{S}' \leftarrow \text{Update}(\mathcal{S}', \{(c_u, c_v) \rightarrow c_{\text{new}}\})$ # Algorithm 3
  $\mathcal{R} \leftarrow \mathcal{R} \cup \{(c_u, c_v) \rightarrow c_{\text{new}}\}$ # new merge rule
  $\mathcal{V} \leftarrow \mathcal{V} \cup \{c_{\text{new}}\}$ # add new token to the vocabulary
end while
return $\mathcal{R}, \mathcal{V}$



$$P(\bigcirc,\bigcirc) = \frac{|\bigcirc\!\!-\!\!\bigcirc|}{|<\bigcirc,\bigcirc>|} = \frac{4-1}{\binom{4}{2}}$$

$$P(\bigcirc,\square) = \frac{|\bigcirc\!\!-\!\!\square|}{|\bigcirc| \times |\square|} = \frac{1}{4 \times 3}$$

$$P(\square,\square) = \frac{|\square\!\!-\!\!\square|}{|<\square,\square>|} = \frac{3-1}{\binom{3}{2}}$$

Features co-occurring within or across items

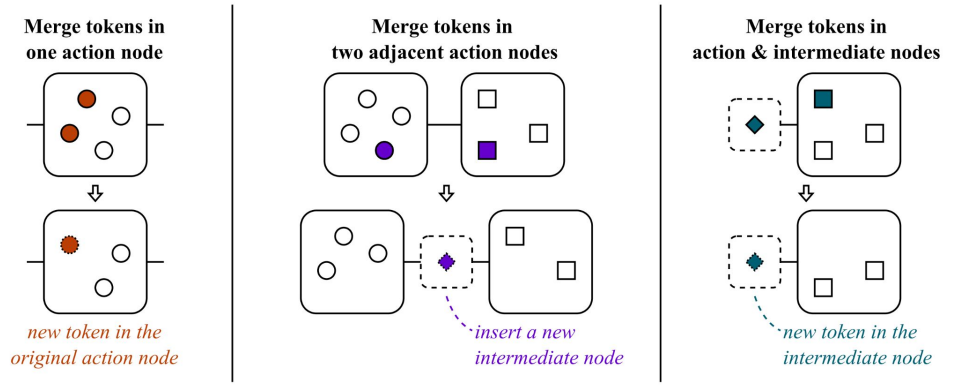
Inputs for SemID Construction

Context-independent ⇒ Context-aware

Algorithm 1 ActionPiece Vocabulary Construction

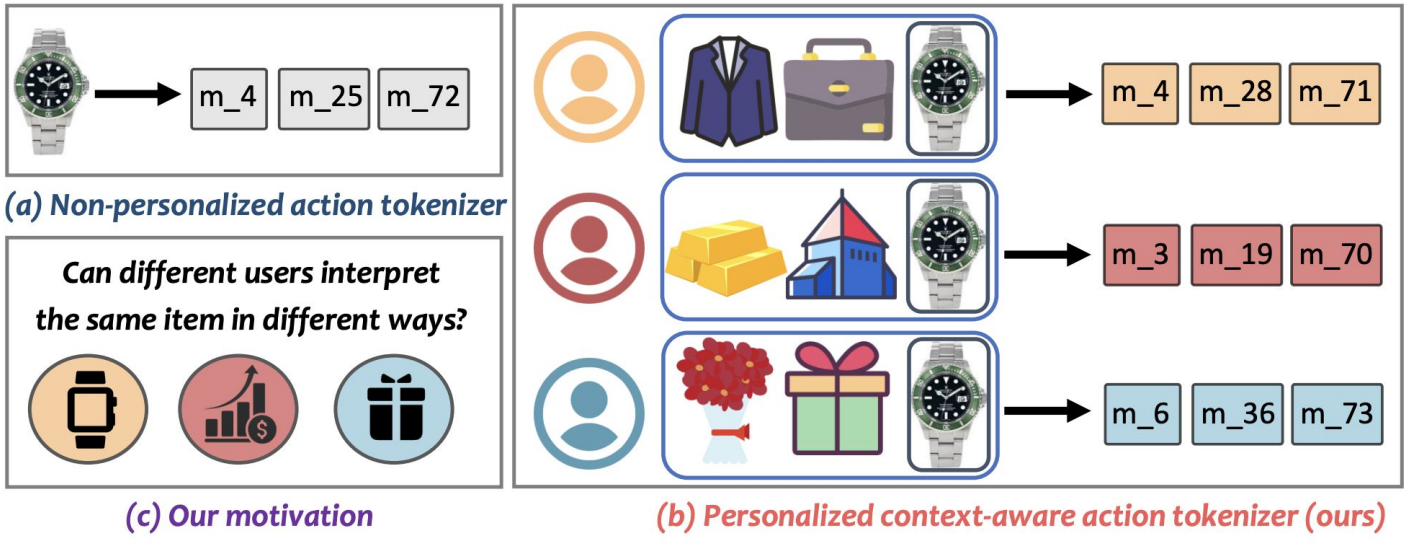
input Sequence corpus $\mathcal{S}'$, initial tokens $\mathcal{V}_0$, target size Q
output Merge rules $\mathcal{R}$, constructed vocabulary $\mathcal{V}$

Initialize vocabulary $\mathcal{V} \leftarrow \mathcal{V}_0$ # each initial token corresponds to one unique item feature
$\mathcal{R} \leftarrow \emptyset$
while $|\mathcal{V}| < Q$ **do**
  # ***Count:*** accumulate weighted token co-occurrences
  $\text{count}(\cdot,\cdot) \leftarrow \text{Count}(\mathcal{S}', \mathcal{V})$ # Algorithm 2
  # ***Update:*** Merge a frequent token pair into a new token
  Select $(c_u, c_v) \leftarrow \arg\max_{(c_i,c_j)} \text{count}(c_i, c_j)$
  $\mathcal{S}' \leftarrow \text{Update}(\mathcal{S}', \{(c_u, c_v) \rightarrow c_{\text{new}}\})$ # Algorithm 3
  $\mathcal{R} \leftarrow \mathcal{R} \cup \{(c_u, c_v) \rightarrow c_{\text{new}}\}$ # new merge rule
  $\mathcal{V} \leftarrow \mathcal{V} \cup \{c_{\text{new}}\}$ # add new token to the vocabulary
end while
return $\mathcal{R}, \mathcal{V}$



Inputs for SemID Construction

Context-independent ⇒ Context-aware ⇒ Personalized



Zhong et al. Pctx: Tokenizing Personalized Context for Generative Recommendation. arXiv:2510.21276.

Inputs for SemID Construction

Input: all data associated with the item

(1) Item Metadata

Text / Multimodal / Categorical / No Features

(2) Item Metadata + Behaviors

Regularization / Fusion

Context-independent ⇒ Context-aware

Part 1 Summary – SemID Construction

(1) First Example: TIGER

(2) Construction Techniques

(3) Inputs

Part 1 Summary – SemID Construction

(1) First Example: TIGER

(2) Construction Techniques

RQ, PQ, Clustering, LM-based generator

(3) Inputs

Part 1 Summary – SemID Construction

(1) First Example: TIGER

(2) Construction Techniques

RQ, PQ, Clustering, LM-based generator

(3) Inputs

Item Metadata (Text, Multimodal)

+ Behaviors (Regularization, Fusion, Context)

Part 2: SemID-based Generative Recommendation Model Architecture

SemID-based Recommender Architecture

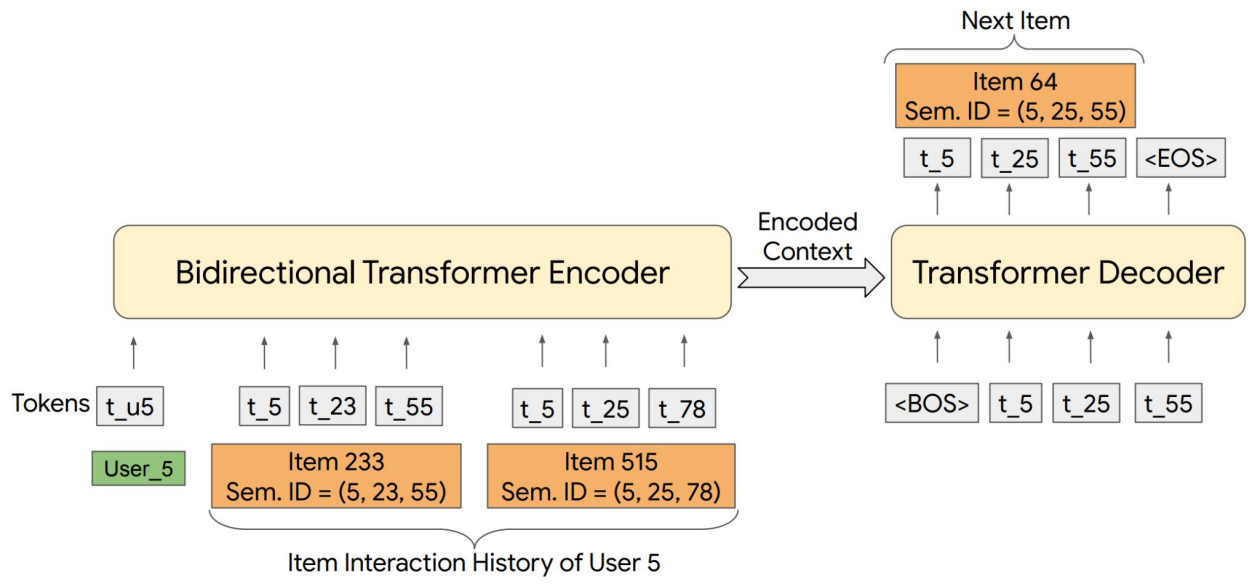
Recommendation as a **seq**-to-**seq** generation problem

Input: user interacted items $\{c_{11}, c_{12}, c_{13}, c_{14}, c_{21}, c_{22}, \ldots\}$

Output: next item $\{c_{t1}, c_{t2}, c_{t3}, c_{t4}\}$

SemID-based Recommender Architecture

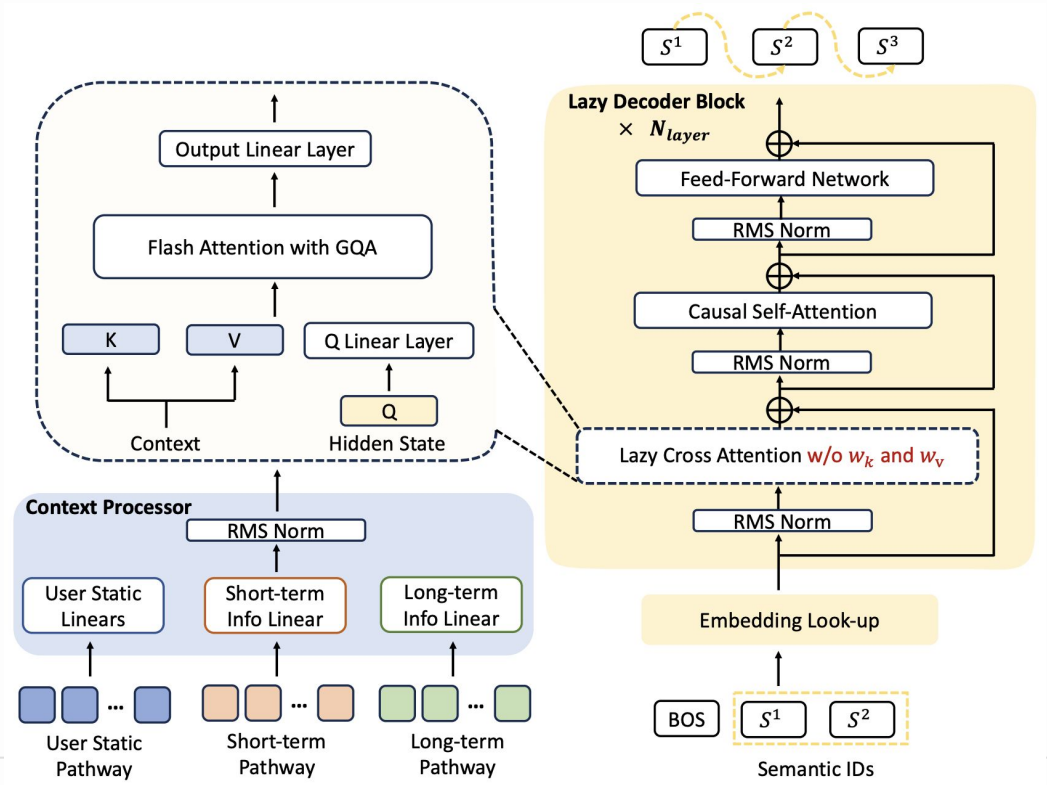
Architecture: Encoder-Decoder



SemID-based Recommender Architecture

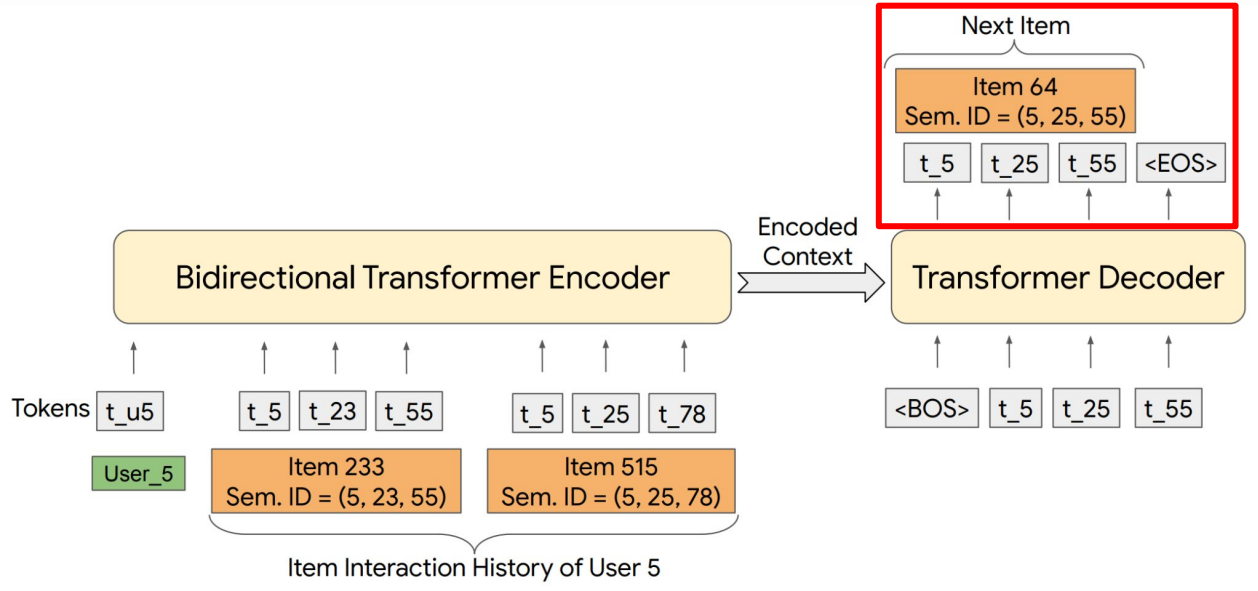
Architecture:

Decoder-Only?



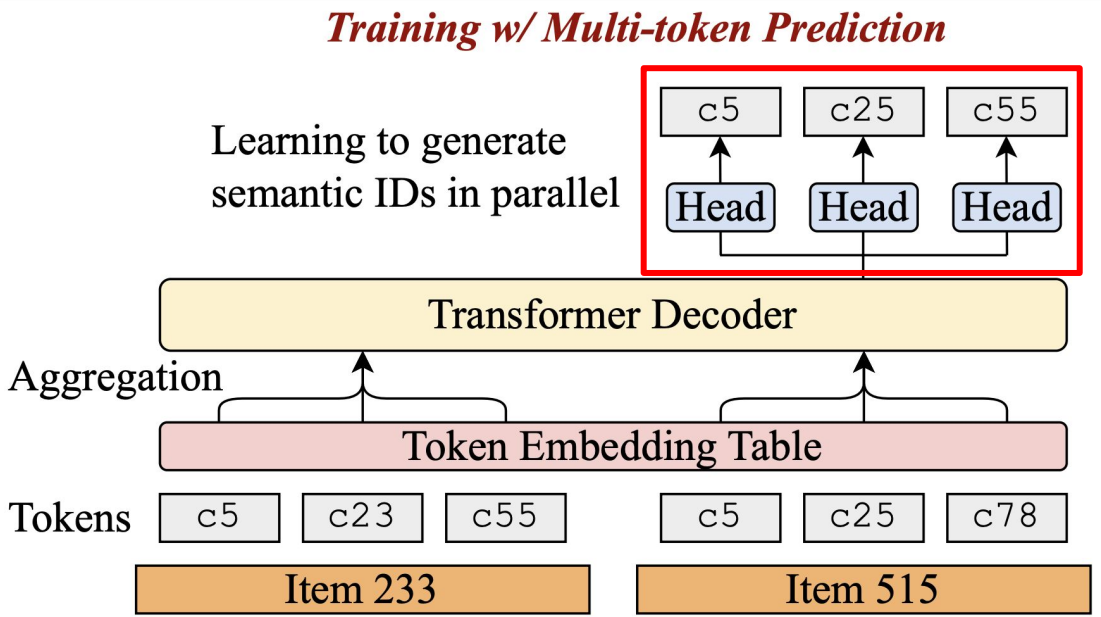
SemID-based Recommender Architecture

Objective: Next-Token Prediction (w/ RQ)



SemID-based Recommender Architecture

Objective: Multi-Token Prediction (w/ PQ)



Hou et al. Generating Long Semantic IDs in Parallel for Recommendation. KDD 2025.

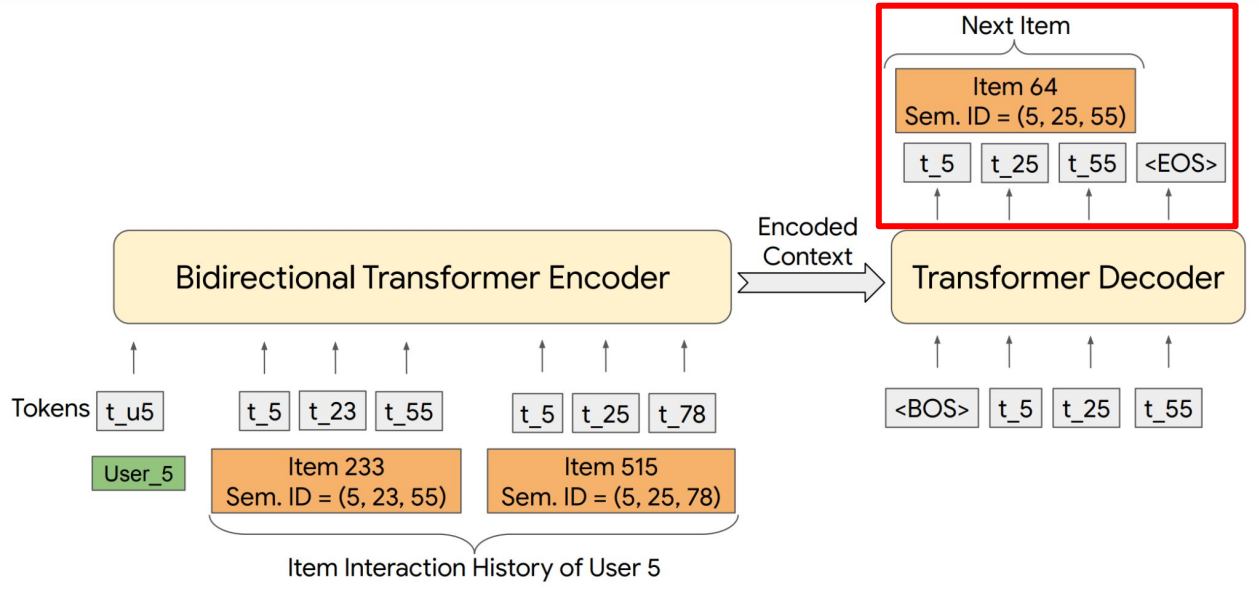
SemID-based Recommender Architecture

Objective: RL

Will introduce later at “**Align w/ LLMs**”

SemID-based Recommender Architecture

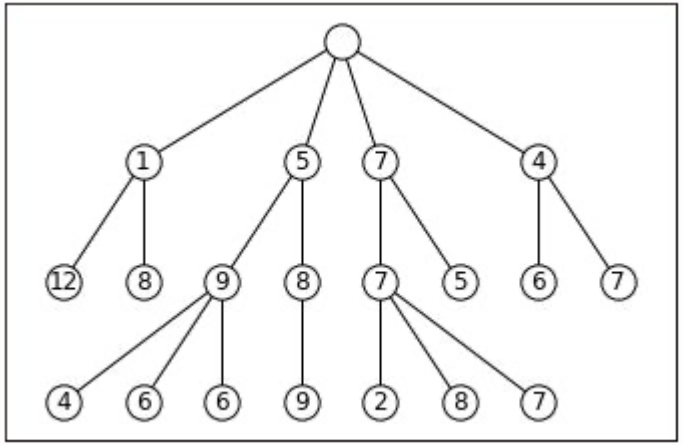
Inference: How to get a ranking list?



SemID-based Recommender Architecture

Inference: How to get a ranking list?

(Constrained) Beam Search



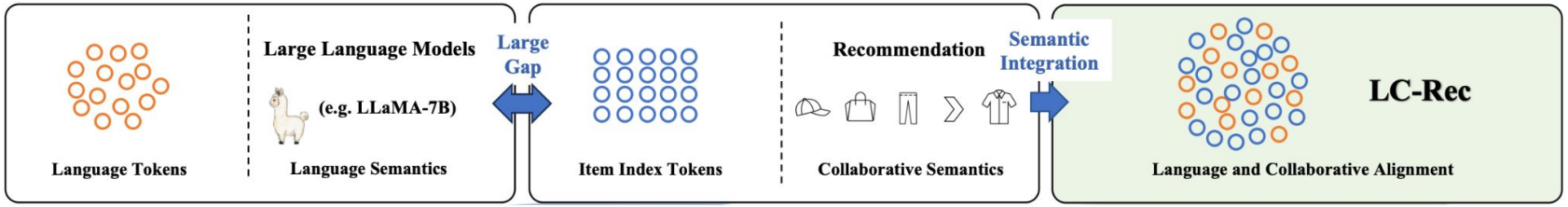
https://en.wikipedia.org/wiki/Beam_search

SemID-based Recommender Architecture

Inference: Can we do dense retrieval-like grounding?

SemID-based Recommender Architecture

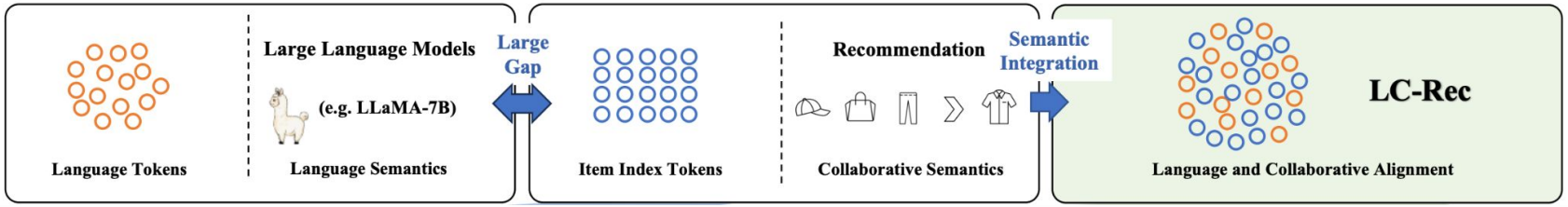
Align with LLMS – LC-Rec



Zheng et al. Adapting Large Language Models by Integrating Collaborative Semantics for Recommendation. ICDE 2024.

SemID-based Recommender Architecture

Align with LLMS – LC-Rec



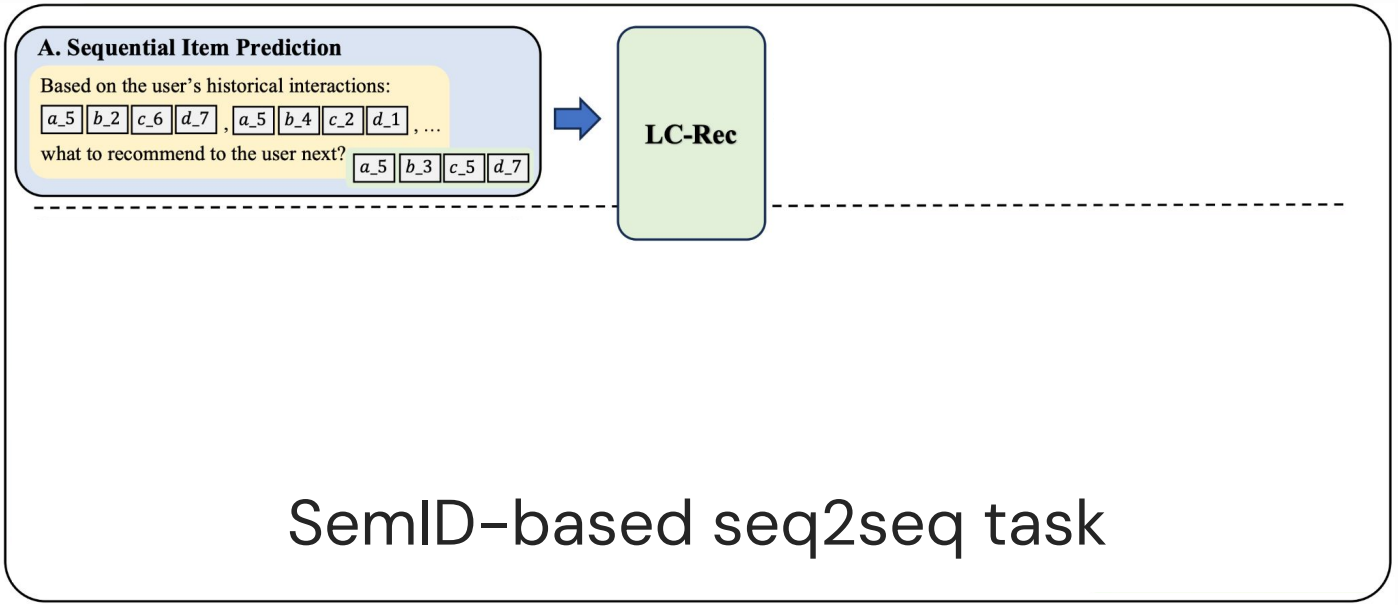
Core Idea:

Construct instructions containing both Semantic IDs and language tokens

Zheng et al. Adapting Large Language Models by Integrating Collaborative Semantics for Recommendation. ICDE 2024.

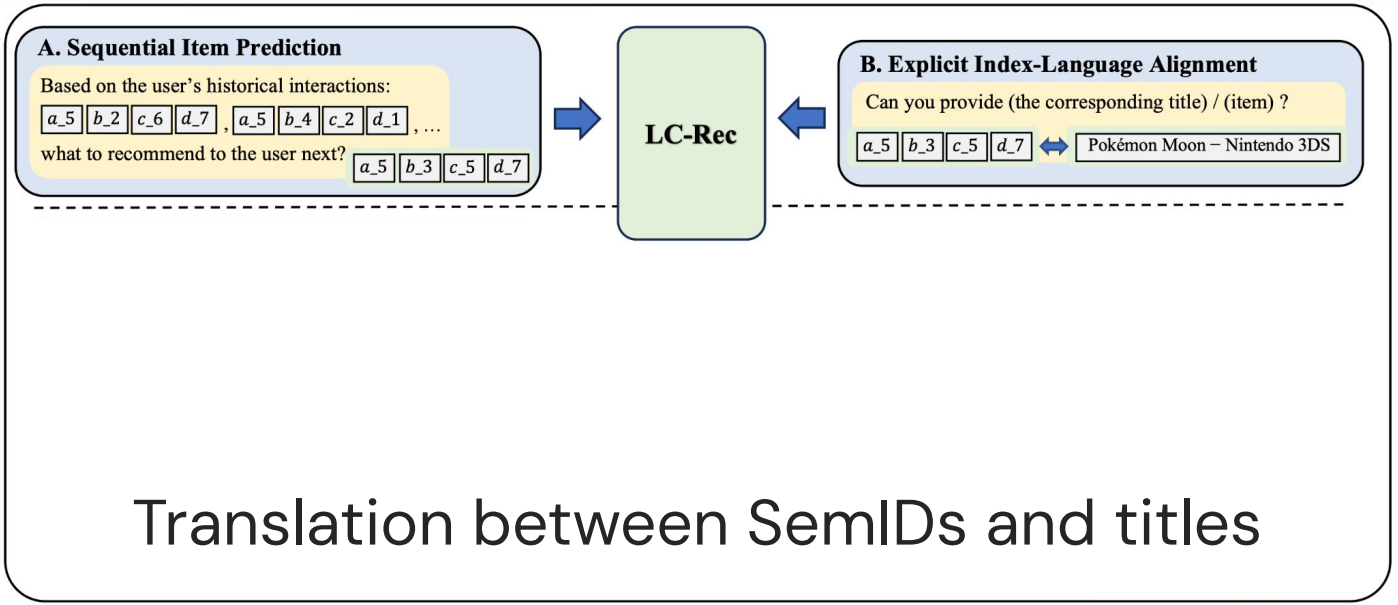
SemID-based Recommender Architecture

Align with LLMS – LC-Rec



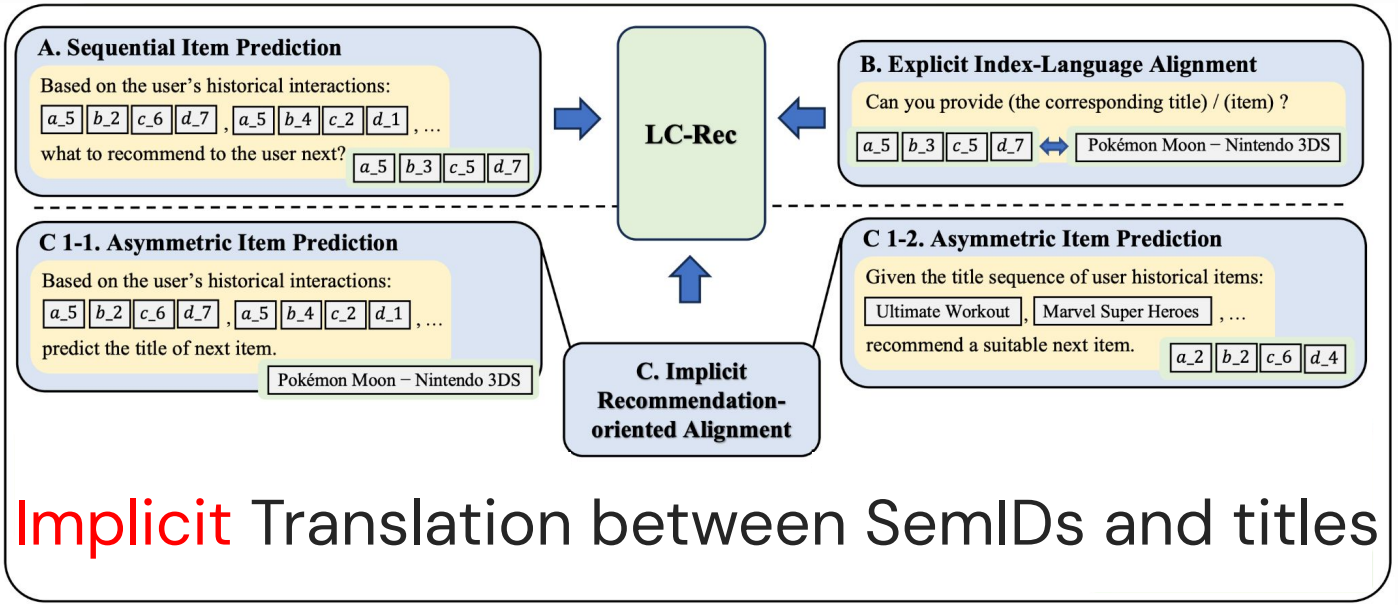
SemID-based Recommender Architecture

Align with LLMS – LC-Rec



SemID-based Recommender Architecture

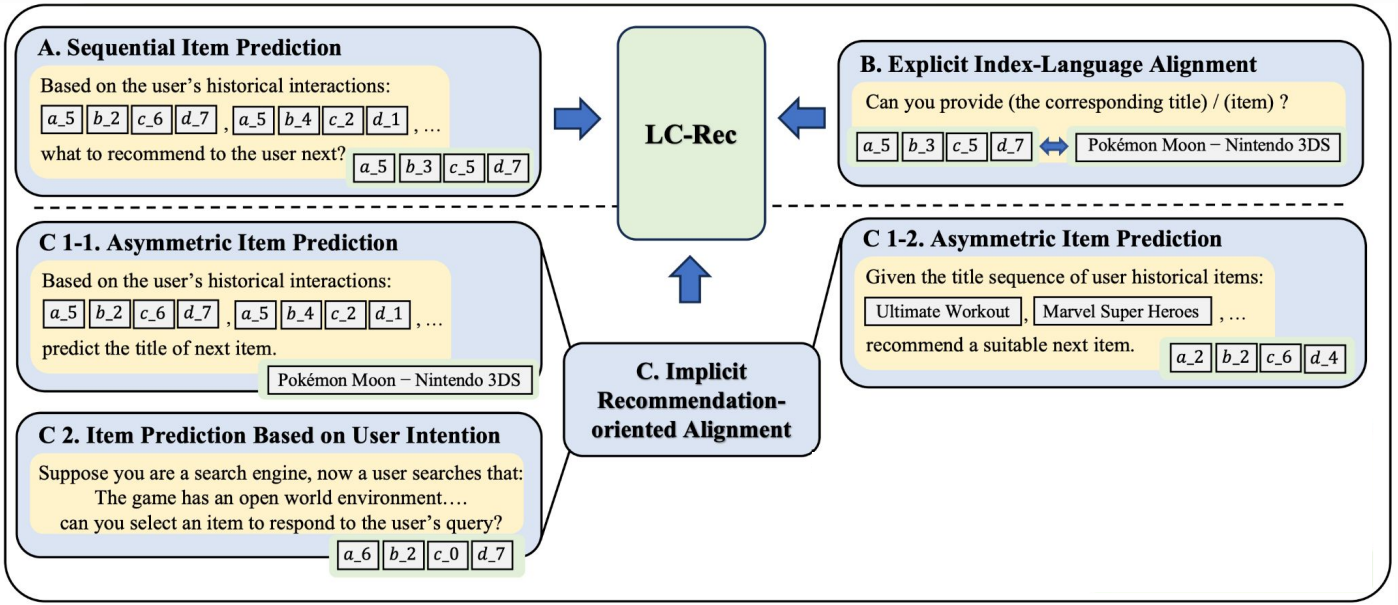
Align with LLMS – LC-Rec



Implicit Translation between SemIDs and titles

SemID-based Recommender Architecture

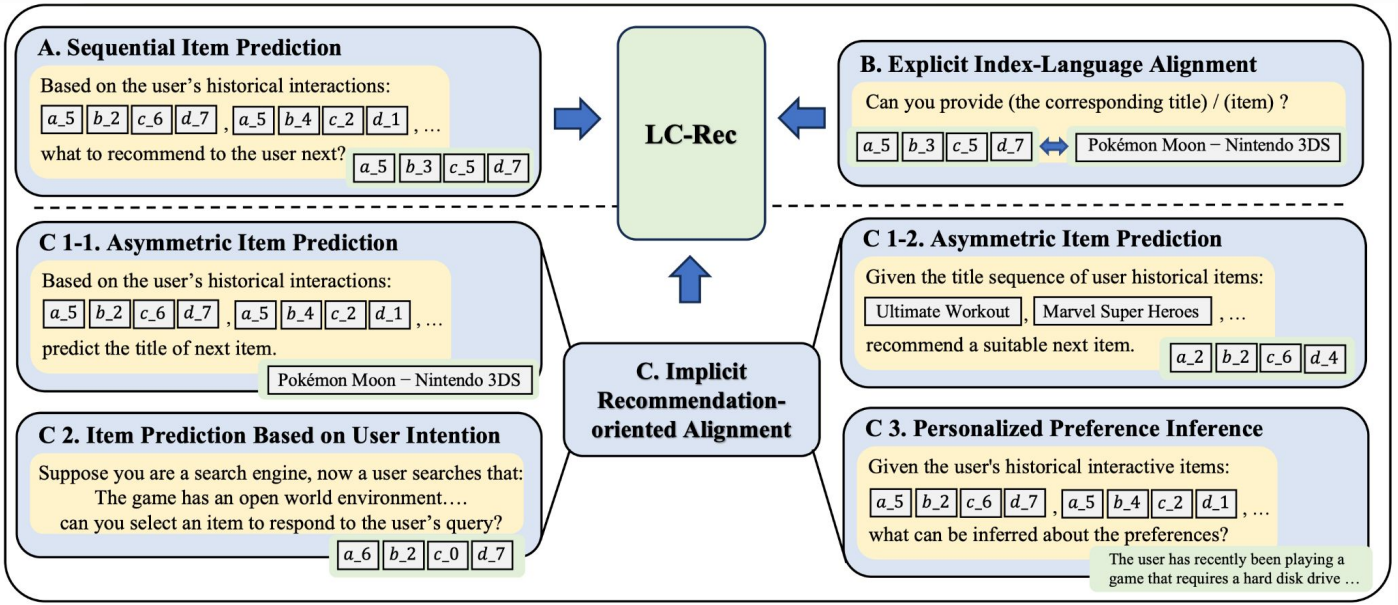
Align with LLMS – LC-Rec



Zheng et al. Adapting Large Language Models by Integrating Collaborative Semantics for Recommendation. ICDE 2024.

SemID-based Recommender Architecture

Align with LLMS – LC-Rec

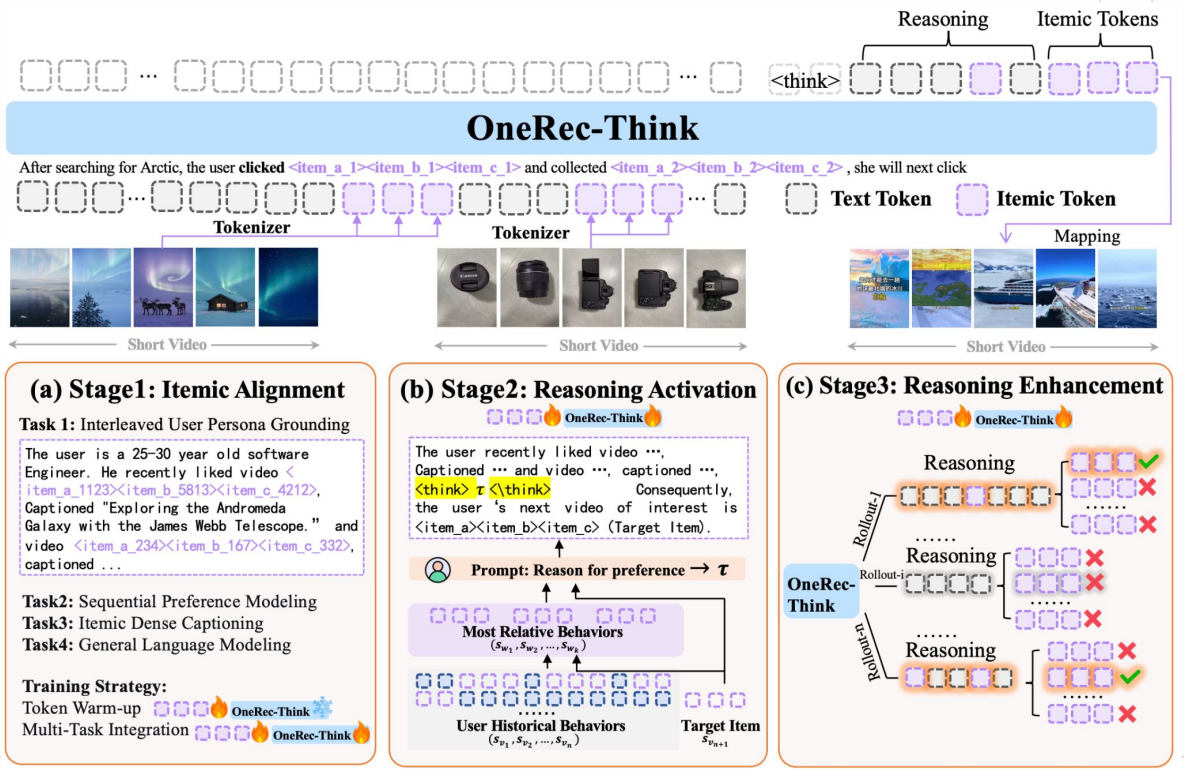


Zheng et al. Adapting Large Language Models by Integrating Collaborative Semantics for Recommendation. ICDE 2024.

SemID-based Recommender Architecture

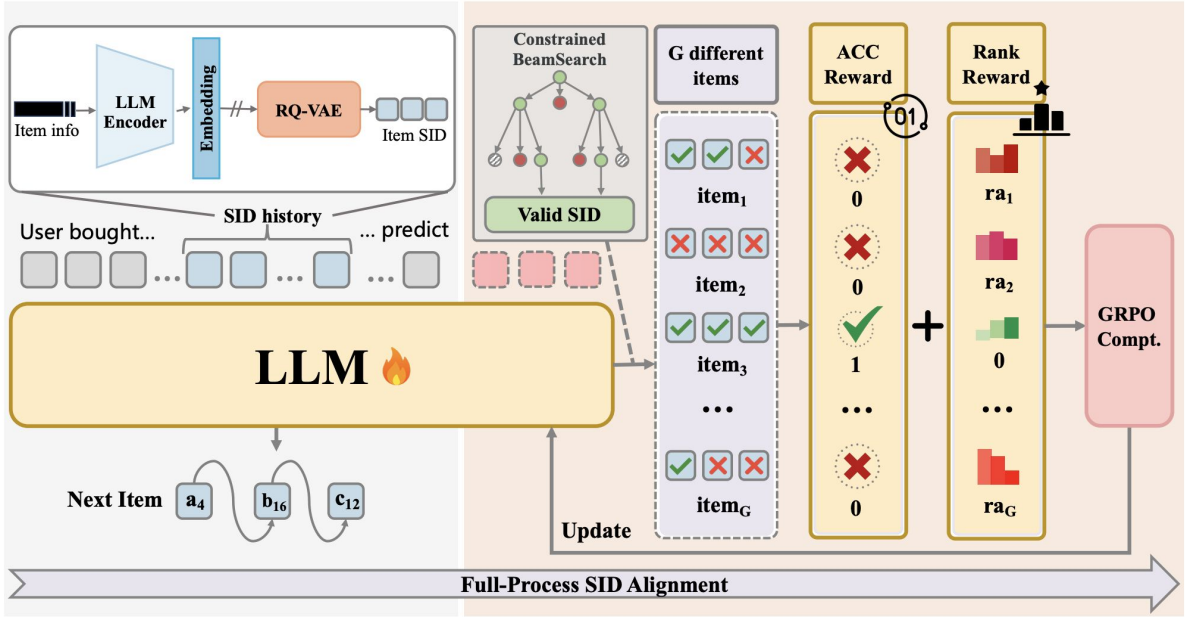
Align with LLMS

OneRec-Think



SemID-based Recommender Architecture

Align with LLMS: MiniOneRec



Part 2 Summary – Architecture

(1) Train from Scratch

(2) Align with LLMs

Part 2 Summary – Architecture

(1) Train from Scratch

Objective (NTP, MTP, RL)

Inference (Beam Search)

(2) Align with LLMs

Part 2 Summary – Architecture

(1) Train from Scratch

Objective (NTP, MTP, RL)

Inference (Beam Search)

(2) Align with LLMs

LC-Rec / OneRec-Think / MiniOneRec