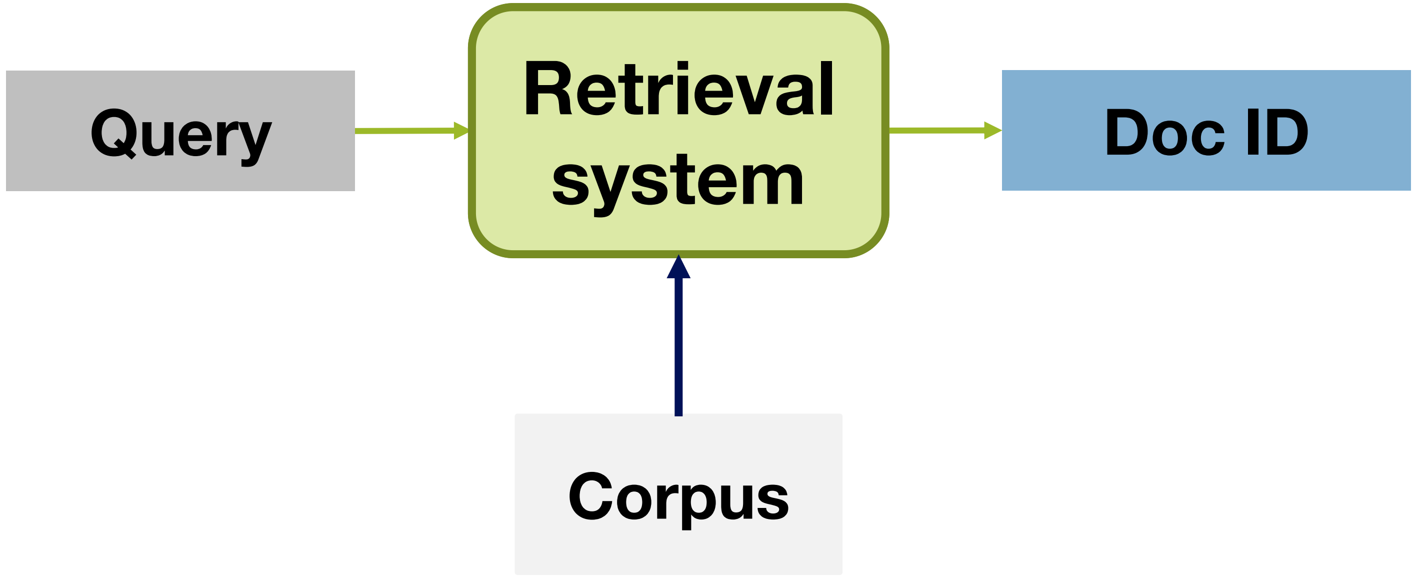
Generative Retrieval
from Search to Recommendation

Zhaochun Ren| LIACS

Sep 15, 2025

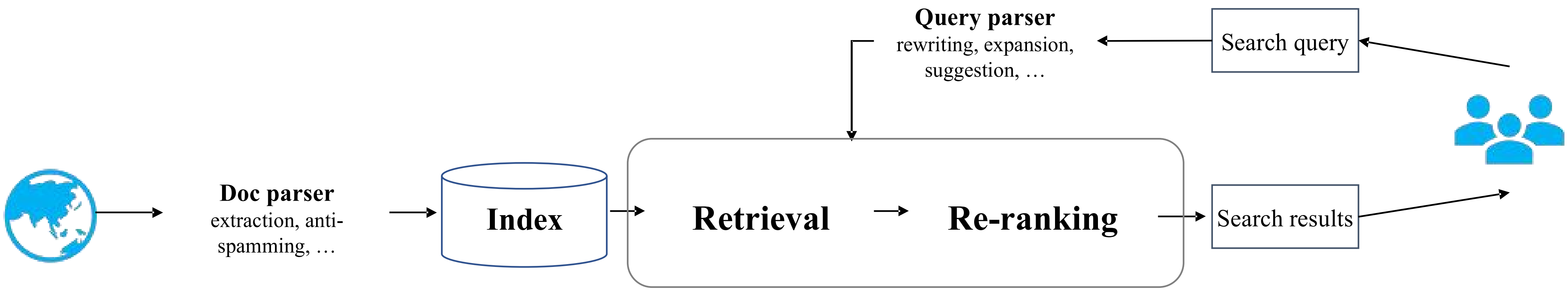
Universiteit Leiden
The Netherlands

Retrieval overview



Pipeline of index-retrieval-ranking

- **Index**: Build an index for each document in the entire corpus
- **Retriever**: Find an initial set of candidate documents for a query
- **Re-ranker**: Determine the relevance degree of each candidate



Differentiable Search Index

Passage rank list

constrained generation

Query → Document identifiers (doc id)

Query: Who is the singer of *Does He Love You*?

→ **Autoregressive Model** →

Generated: recorded as a duet by American country music artists

Rank 1st passage:
Title: Does He Love You
Body: … recorded as a duet by American country music artists …
ID: 32 – 16 – 18

Rank 2nd passage:
Title: Does He Love You
Body: …
ID: 32 – 16 – 16

Other different designs of document identifiers:

Title: Does He Love You **ID:** 32 – 16 – 18

[1] Transformer Memory as a Differentiable Search Index, 2022

Differentiable Search Index

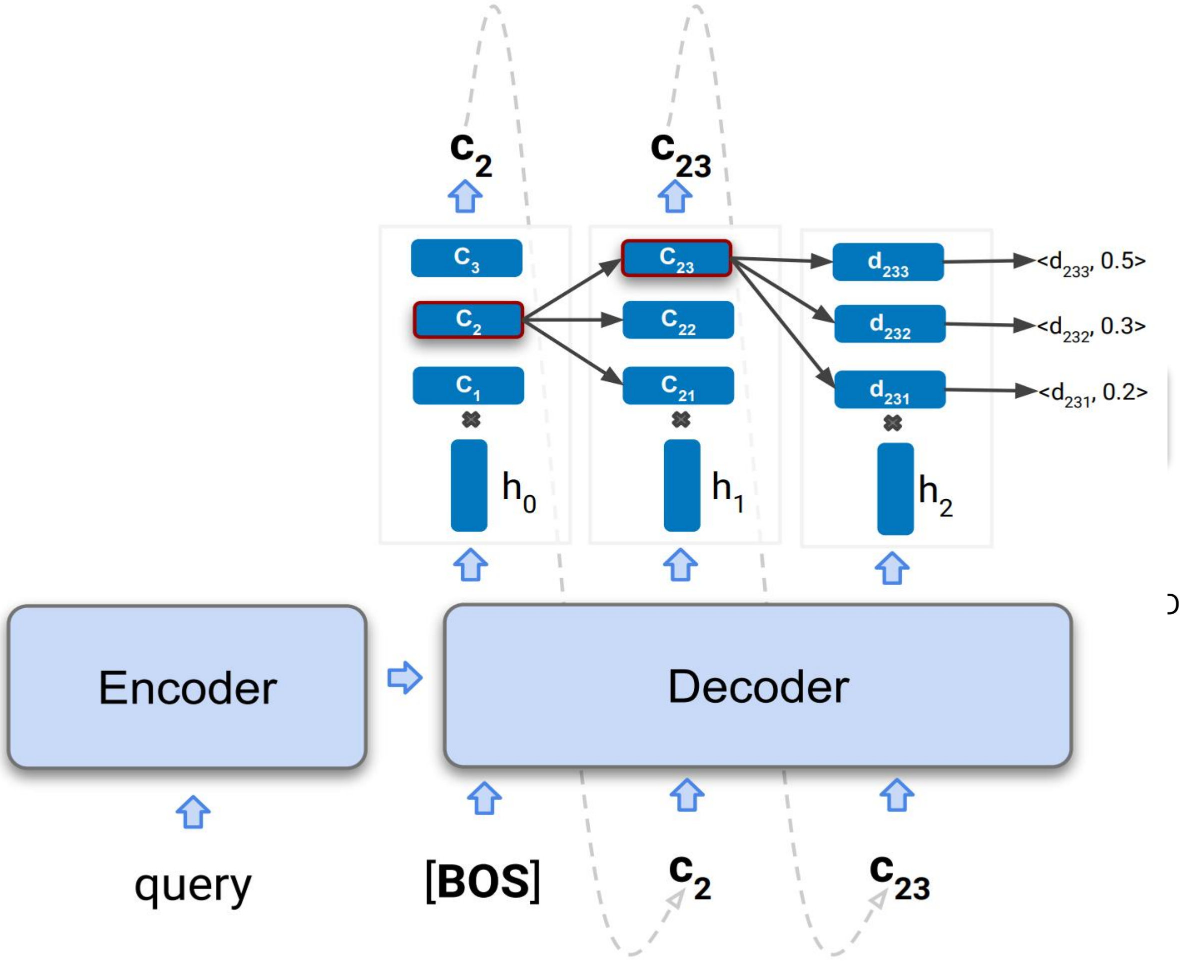
- GR exploits a Seq2Seq encoder-decoder architecture to generate a ranked list of docids for an input query, in an autoregressive fashion.

- ➢ Directly generate relevant documents.
- ➢ Encoder-decoder transformer architecture, e.g., T5, BART.
- ➢ Fully end-to-end paradigm.
- ➢ May not achieve comparable performance.

[1] Transformer Memory as a Differentiable Search Index, 2022

Relation between GR and dense retrieval

- It is difficult to distinguish between bi-encoder dense retrieval models at first glance, especially when using atomic IDs.
- A common interpretation for hierarchical semantic ID is tree index (for single vector model).
- fails to explain non-hierarchical IDs (e.g., title, n-gram) and limited by single vector



[1] Generative Retrieval as Dense Retrieval, GenIR workshop@SIGIR 2023

Relation between GR and dense retrieval

- **Multi-vector dense retrieval (MVDR)** e.g., ColBERT, COIL, PLAID, etc.
  - One of the prevalent re-ranking models
- **Generative retrieval (GR)** e.g., GENRE, SEAL, DSI, NCI, GenRet, etc.
  - A new paradigm that directly generates relevant documents

We discover that GR and MVDR ***share the same framework*** *for* measuring query-document relevance

SIGIR 2024 Washington, D.C.

[1] Generative Retrieval as Multi-Vector Dense Retrieval, In SIGIR 2024

Relation between GR and dense retrieval

- We can do simple generalization
  - Single-vector ↔ atomic ID
  - Multi-vector ↔ semantic ID

- Easily get some (preliminary) answers by looking deeply into the model architecture.

- a. how decoder performs matching
- b. explain non-hierarchical IDs, e.g., title, ngram, etc.
- c. limited by single vector → relation with multi-vector frameworks

[1] Generative Retrieval as Multi-Vector Dense Retrieval, In SIGIR 2024

Brief summary

- MVDR can be generalized into

$$\sum_{ij} (\boldsymbol{D}^\top \boldsymbol{Q} \odot \boldsymbol{A})_{ij}$$

- GR computes the relevance as

$$\sum_{ij} (\boldsymbol{E}_d^\top \boldsymbol{Q} \odot \boldsymbol{A})_{ij}$$

Both methods share the same framework to compute the relevance

$$\sum_{ij} (\boldsymbol{D}^\top \boldsymbol{Q} \odot \boldsymbol{A})_{ij}$$

[1] Generative Retrieval as Multi-Vector Dense Retrieval, 2024

Comparison of MVDR and GR

component in $\sum_{ij}(\boldsymbol{D}^\top\boldsymbol{Q} \odot \boldsymbol{A})_{ij}$	**MVDR**	**GR**
$\boldsymbol{D}$ doc encoding	$\boldsymbol{D}$ (token vector)	$\boldsymbol{E}$ (embedding vector)
$\boldsymbol{Q}$ query encoding	$\boldsymbol{Q}$ (token vector)	$\boldsymbol{Q}$ (token vector)
$\boldsymbol{A}$ alignment matrix	Sparse	Dense and learned

[1] Generative Retrieval as Multi-Vector Dense Retrieval, 2024

Findings

- We discover a new relationship between MVDR and GR
- For a more detailed discussion and experiment, please refer to our paper :)
  - about low-rank alignment, decomposition, alignment direction, case study, etc.
- **Missing**: effect of multi-layer interaction, query generation, etc.
- **Analyze other scenarios**: binary identifiers? indexing stage, etc.
- **Trends**: MVDR and GR are more and more alike

[1] Generative Retrieval as Multi-Vector Dense Retrieval, 2024

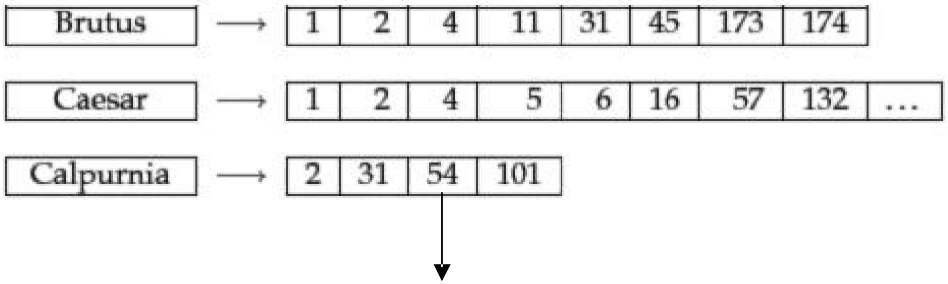
Workflows of generative retrieval

DocID design	Model training
Inference	Applications

[1] Recent advances in generative information retrieval, SIGIR 2024 tutorial

DocID design in generative retrieval

Traditional information retrieval

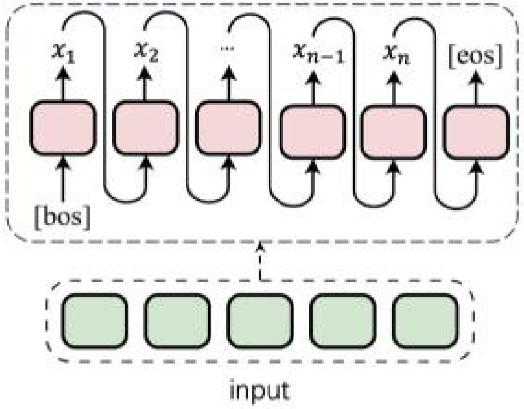


Document features

As an entry

Generative retrieval

Docid: xx xxx x

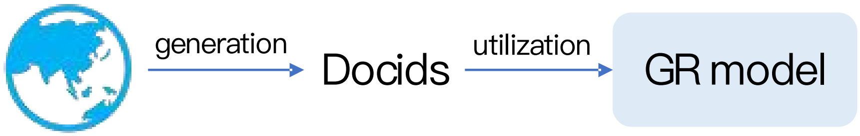


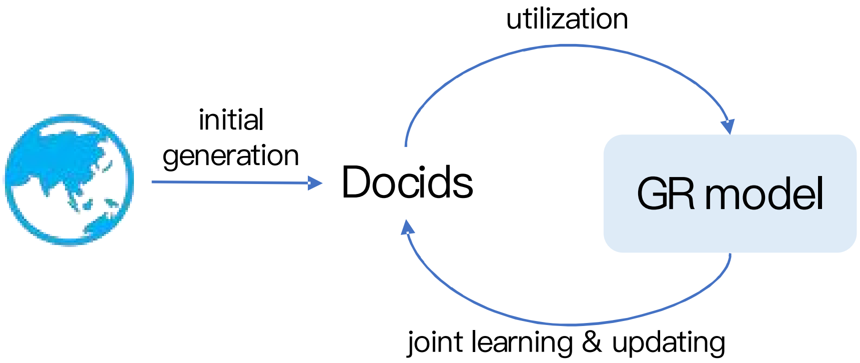
For generation

How to design docID for documents in GR?

[1] Generative Information Retrieval, SIGIR 2024 tutorial

DocID design in generative retrieval



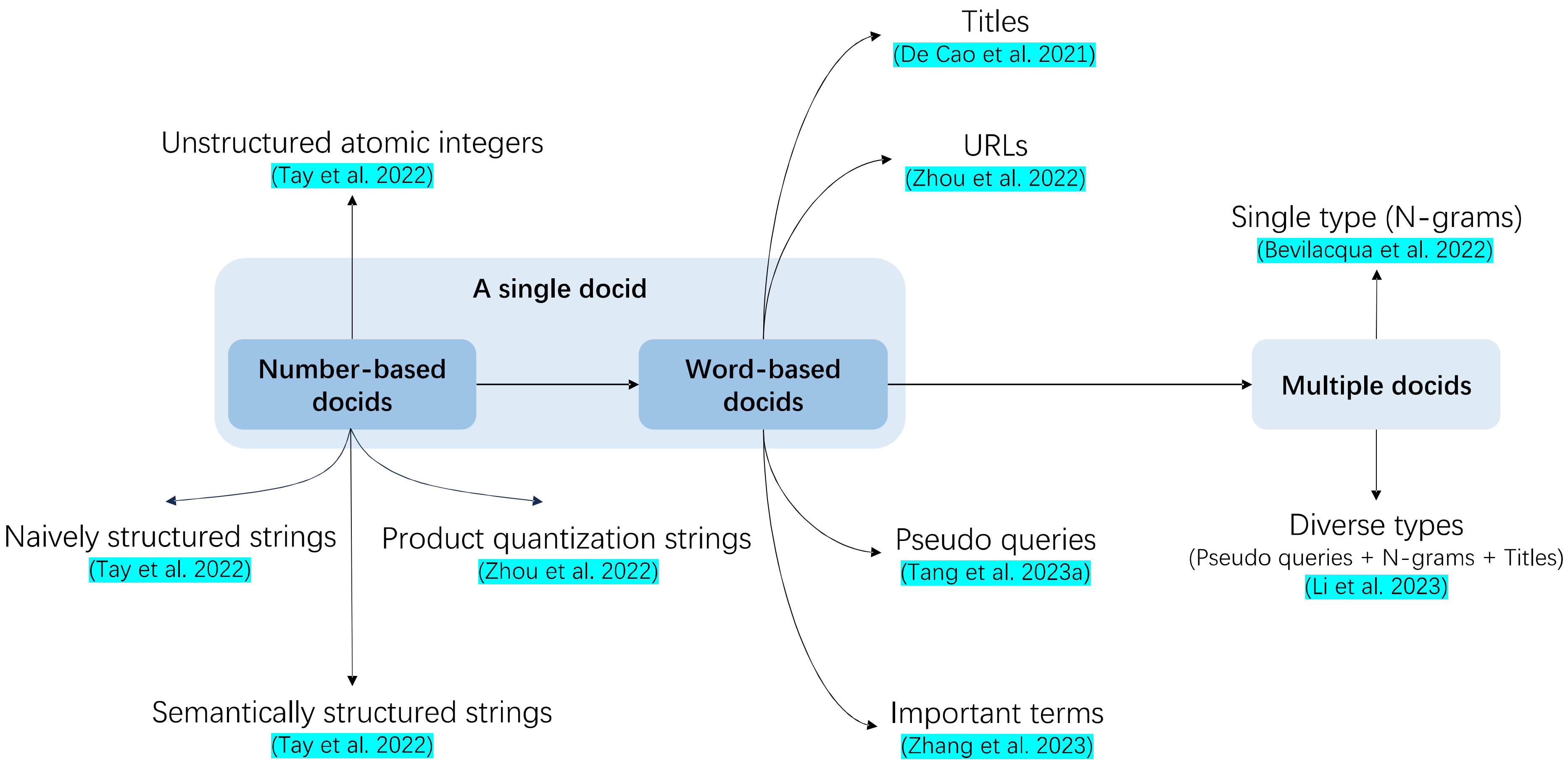


- Pre-defined static docids

- Learnable docids

[1] Generative Information Retrieval, SIGIR 2024 tutorial

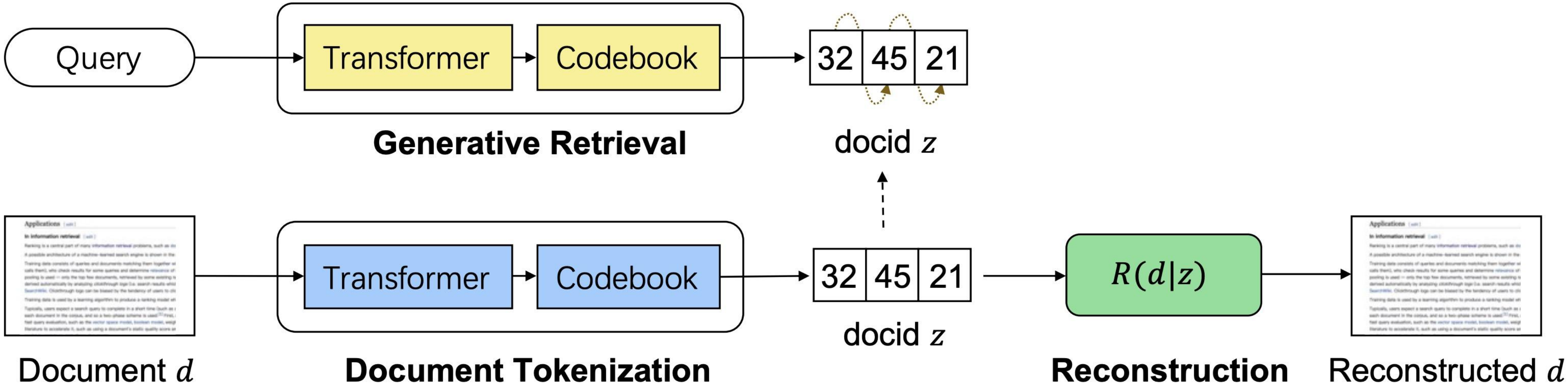
Roadmap of pre-defined static docIDs



[1] Generative Information Retrieval, SIGIR 2024 tutorial

Learnable DocID: GenRet

Propose Document Tokenization Learning Algorithm



Based on discrete auto-encoding

Intuition

A reconstruction model can **reconstruct** the document from docid.
The docid captures the **semantic** information of the document.

[1] Learning to Tokenize for Generative Retrieval, 2023

Generalization challenges in GR

Studies over unseen data/unseen tasks

Universiteit Leiden
The Netherlands

Generative retrieval over dynamic corpora

- Generative Retrieval (GR) shows promise, but its effectiveness in dynamic corpora is largely unexplored.
- Systematically evaluated various current GR models and traditional IR models in dynamic settings.
  - Traditional IR: BM25, DPR, etc.
  - Numeric-based: DSI, GenRET, NCI, etc.
  - Text-based: SEAL, MINDER, etc.

We exploit the reasons behind GR performance differences.

SIGIR 2025
Padova
ITALY

[1] Replication and Exploration of Generative Retrieval over Dynamic Corpora, 2025

Experiments

Performance as documents are incrementally added:

Retrieval initial documents.

Method	DocID Type	NQ (Hit@10)						
		$\mathcal{D}_0$	$\mathcal{D}_1$	$\mathcal{D}_2$	$\mathcal{D}_3$	$\mathcal{D}_4$	$\mathcal{D}_5$	$F_n \downarrow$
Sparse retrieval								
BM25	Term Weight	0.647	0.625	0.611	0.598	0.573	0.573	0.051
Dense retrieval								
DPR	Dense Vector	0.725	0.704	0.696	0.686	0.670	0.660	0.042
DPR-HN	Dense Vector	0.826	0.801	0.797	0.776	0.773	0.768	0.043
Generative retrieval								
DSI-SE	Category Nums	0.718	0.710	0.706	0.702	0.699	0.696	**0.015**
Ultron-PQ	Category Nums	0.795	0.785	0.780	0.780	0.762	0.755	0.023
NCI	Category Nums	**0.871**	0.856	0.844	**0.839**	0.811	0.802	0.041
GenRET	Category Nums	0.858	0.853	0.836	0.829	0.812	0.796	0.033
Ultron-URL	URL Path	0.816	0.810	0.794	0.781	0.780	0.768	0.029
SEAL	N-gram	0.809	0.806	0.788	0.774	0.774	0.763	0.028
MINDER	Multi-text	0.838	0.828	0.813	0.811	0.801	0.773	0.033
LTRGR	Multi-text	0.862	**0.857**	**0.846**	0.827	**0.813**	**0.807**	0.032

Retrieval newly added documents.

Method	DocID Type	NQ (Hit@10)						
		$\mathcal{D}_0$	$\mathcal{D}_1$	$\mathcal{D}_2$	$\mathcal{D}_3$	$\mathcal{D}_4$	$\mathcal{D}_5$	$GA_n \uparrow$
Sparse retrieval								
BM25	Term Weight	0.647	0.620	0.588	0.598	0.552	0.571	0.586
Dense retrieval								
DPR	Dense Vector	0.725	0.580	0.587	0.570	0.531	0.544	0.562
DPR-HN	Dense Vector	0.826	0.645	0.644	0.626	0.621	0.624	0.632
Generative retrieval								
DSI-SE	Category Nums	0.718	0.231	0.203	0.221	0.185	0.205	0.209
Ultron-PQ	Category Nums	0.795	0.548	0.549	0.542	0.539	0.532	0.542
NCI	Category Nums	**0.871**	0.464	0.437	0.433	0.358	0.323	0.403
GenRET	Category Nums	0.858	0.361	0.419	0.401	0.357	0.354	0.378
Ultron-URL	URL Path	0.816	0.553	0.545	0.543	0.541	0.532	0.543
SEAL	N-gram	0.809	0.744	0.736	0.727	0.727	0.725	0.732
MINDER	Multi-text	0.838	0.803	0.751	0.746	0.742	0.736	0.756
LTRGR	Multi-text	0.862	**0.831**	**0.803**	**0.811**	**0.779**	**0.773**	**0.799**

- Retrieving Initial Documents:
  - All Methods (BM25, DPR, GR) show stable performance and low forgetting.
  - GR often exhibits good resistance to forgetting, especially numeric-based ones.

- Retrieving Newly Added Documents:
  - BM25 & DPR demonstrate stable generalization ability.
  - GR performance varies Greatly:
    - Numeric-based DocIDs: Poor generalization on new documents (sharp performance drop).
    - Text-based DocIDs (Except Ultron-URL) : Strong generalization on new documents.

What does constraint GR?

- **Key Challenge – Generalization:**
  - GR models on unseen *out-of-distribution* corpora is not well.
  - Existing works mainly focus on training strategies (e.g. scaling to diverse data) to improve generalization.
  - The fundamental limitations imposed by GR's **constrained auto-regressive decoding** on generalization remain largely unexplored.
- **Motivation:**
  - Investigate the theoretical limitations introduced by the constrained decoding process in GR.
  - Aim to understand how forcing generation to valid docIDs (and using decoding algorithms like beam search) might inherently constrain GR's ability to generalize to new corpora.

[1] Constrained Auto-Regressive Decoding Constrains Generative Retrieval, 2025

SIGIR 2025 Padova ITALY

Research question

Can well-trained GR models generalize directly to different domains?

Awareness of semantics

Can *constrained decoding* handle the domain variation?

We assume the model has perfect knowledge

This is what our work tries to explore

Findings

- Constraints
  - Step-wise valid constraints will change model's relevance prediction
  - We have a lower-bound on the distribution change

$$\mathrm{KL}\,[P \parallel Q] \gtrsim \boxed{\frac{0.05\mathbb{E}^2[A_i]}{p}}$$

After constraints

Desired distribution

Related to size and concentration of the downstream corpus

- Beam search
  - For data model of sparse and thick tail relevance distribution
  - The precision is perfect but top-k recall is bounded by 0.5+o(1)
  - The main reason is the use of marginal distribution in each step
  - k is the number of relevant documents

Trade-off & Design Implications

[1] Constrained Auto-Regressive Decoding Constrains Generative Retrieval, 2025

What does constraint GR?

- **Key Challenge – Generalization:**
  - GR models on unseen *out-of-distribution* corpora is not well.
  - The fundamental limitations imposed by GR's **constrained auto-regressive decoding** on generalization remain largely unexplored.
  - Two kinds of generalization challenges:

    Generalization challenge in newly added documents

    Generalization challenge in out-of-distribution IR tasks (unseen tasks)

SIGIR 2025 Padova ITALY

[1] Constrained Auto-Regressive Decoding Constrains Generative Retrieval, 2025

Initial and newly added documens

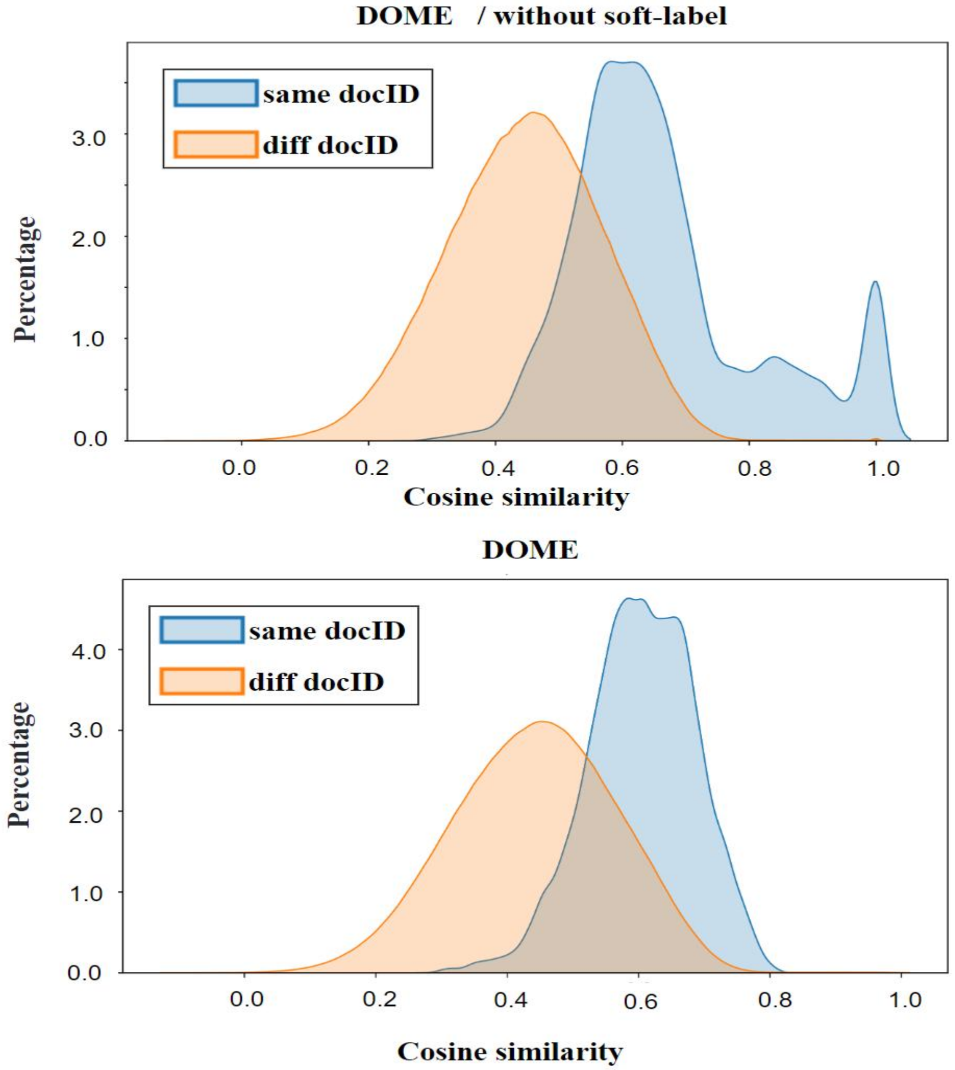
- Behavioral analysis on the initial and newly added document sets using a hierarchical k-means–based docID GR model.

Generalization in newly added documents

- **New problem framing:** Adapt GR to *new documents* by **editing the semantics→docID mapping**, not full model retraining **(DOME)**.
- **Targeted edit scope:** Use **patching analysis** to locate and edit only parameters responsible for mapping representations to docIDs.
- **GR-aware editing procedure:**
  - **Pseudo-query generation** per new doc to cover diverse intents (many-to-many q–d).
  - **Soft→hard label annealing** to preserve graded relevance patterns while learning unseen docIDs.
- **Forgetting-aware design:** Updates the new mappings **without degrading** existing ones.

Soft→hard label annealing

- One-hot edits for a new docID **overwrite graded relevance** across many queries → **forgetting** and degraded retrieval.
- Start with a **soft target** that blends the model's original distribution with the ground-truth token, then **anneal** to one-hot.
- **Integrates new docID mappings smoothly**, preserves existing relevance structure
- **reduces catastrophic forgetting** while improving new-doc retrieval.



Efficient, stable adaptation without full retraining

- **Efficiency:** Dramatically **reduces adaptation time** vs. incremental retraining—no full corpus re-indexing or large-scale fine-tuning.
- **Accuracy on new docs:** Significant **Recall@10 / Acc gains** on **NQ** and **MS-MARCO** for newly added items.
- **Robustness: Strong resistance to catastrophic forgetting** on original corpus.
- **Practicality:** Scalable updates for dynamic corpora; drop-in for GR systems using hierarchical/k-means docIDs or similar designs.

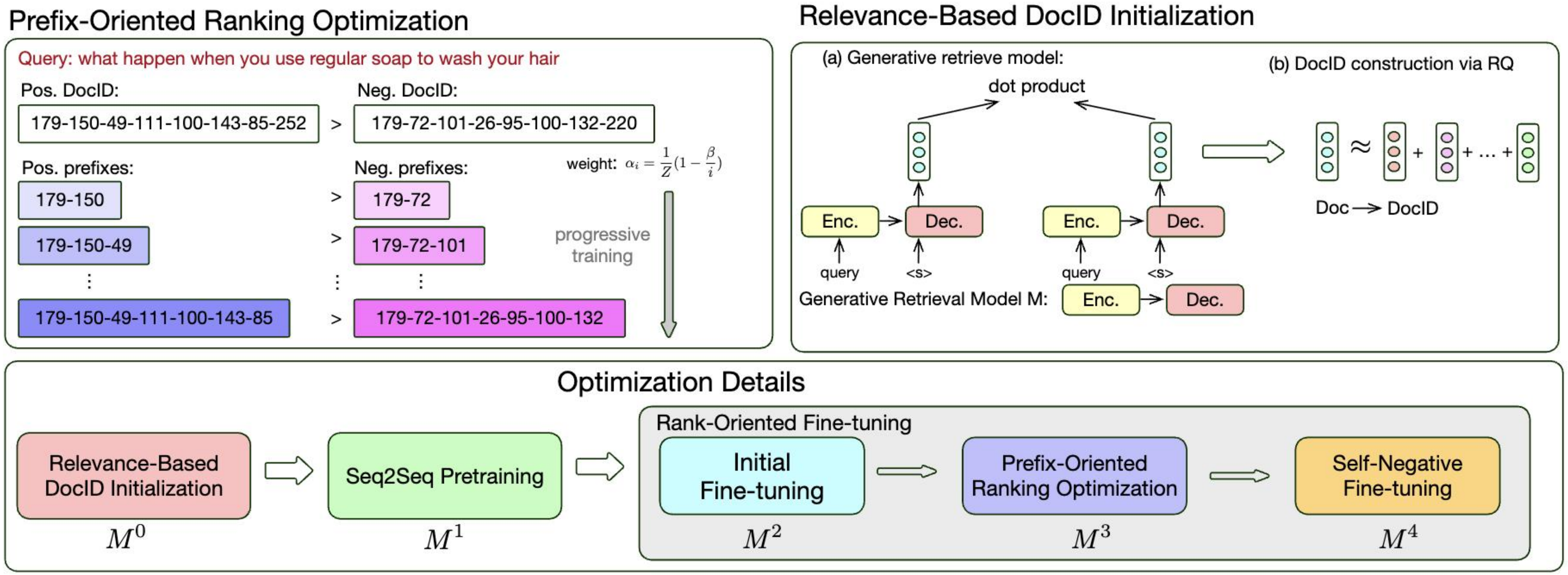
Scaling GR

- GR has strong theoretical appeal: no external index, end-to-end training, easy integration with large language models for tasks like open-domain QA and conversational search.
- GR models only work well on **small or synthetic collections**. On large-scale real-world benchmarks, even simple sparse methods like BM25 significantly outperform GR.
- The research community has raised skepticism about the real-world utility of GR due to these poor large-scale results.

Why do generative retrieval models fail at scale, and how can we design them to be effective on large real-world retrieval benchmarks?

RIPOR

- Prefix-Oriented Ranking Optimization
- Relevance-Based DocID Construction
- Three-Stage Optimization Pipeline



[1] Scalable and Effective Generative Information Retrieval, The WebConf 2024

What does constraint GR?

- **Key Challenge – Generalization:**
  - GR models on unseen *out-of-distribution* corpora is not well.
  - The fundamental limitations imposed by GR's **constrained auto-regressive decoding** on generalization remain largely unexplored.
  - Two kinds of generalization challenges:

    Generalization challenge in newly added documents

    Generalization challenge in out-of-distribution IR tasks (unseen tasks)

[1] Constrained Auto-Regressive Decoding Constrains Generative Retrieval

SIGIR 2025 Padova ITALY

Initial and newly added documens

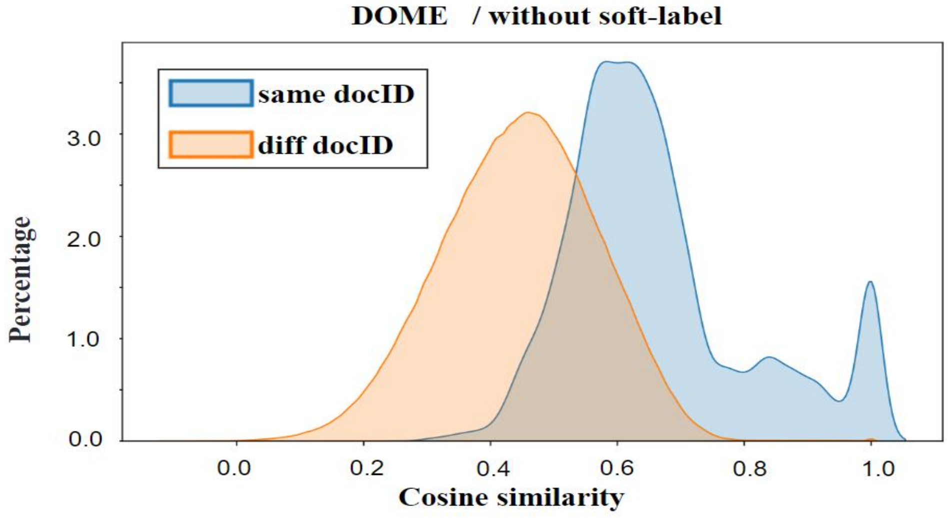
- Behavioral analysis on the initial and newly added document sets using a hierarchical k-means–based docID GR model.

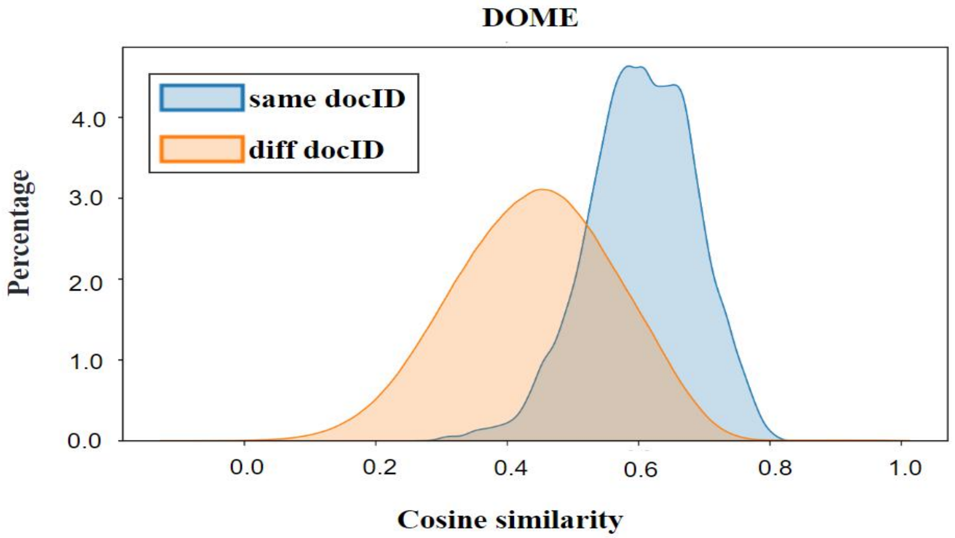
Generalization in newly added documents

- **New problem framing:** Adapt GR to *new documents* by **editing the semantics→docID mapping**, not full model retraining **(DOME)**.
- **Targeted edit scope:** Use **patching analysis** to locate and edit only parameters responsible for mapping representations to docIDs.
- **GR-aware editing procedure:**
  - **Pseudo-query generation** per new doc to cover diverse intents (many-to-many q–d).
  - **Soft→hard label annealing** to preserve graded relevance patterns while learning unseen docIDs.
- **Forgetting-aware design:** Updates the new mappings **without degrading** existing ones.

Soft→hard label annealing

- One-hot edits for a new docID **overwrite graded relevance** across many queries → **forgetting** and degraded retrieval.
- Start with a **soft target** that blends the model's original distribution with the ground-truth token, then **anneal** to one-hot.
- **Integrates new docID mappings smoothly**, preserves existing relevance structure
- **reduces catastrophic forgetting** while improving new-doc retrieval.





Efficient, stable adaptation without full retraining

- **Efficiency:** Dramatically **reduces adaptation time** vs. incremental retraining—no full corpus re-indexing or large-scale fine-tuning.
- **Accuracy on new docs:** Significant **Recall@10 / Acc gains** on **NQ** and **MS-MARCO** for newly added items.
- **Robustness: Strong resistance to catastrophic forgetting** on original corpus.
- **Practicality:** Scalable updates for dynamic corpora; drop-in for GR systems using hierarchical/k-means docIDs or similar designs.

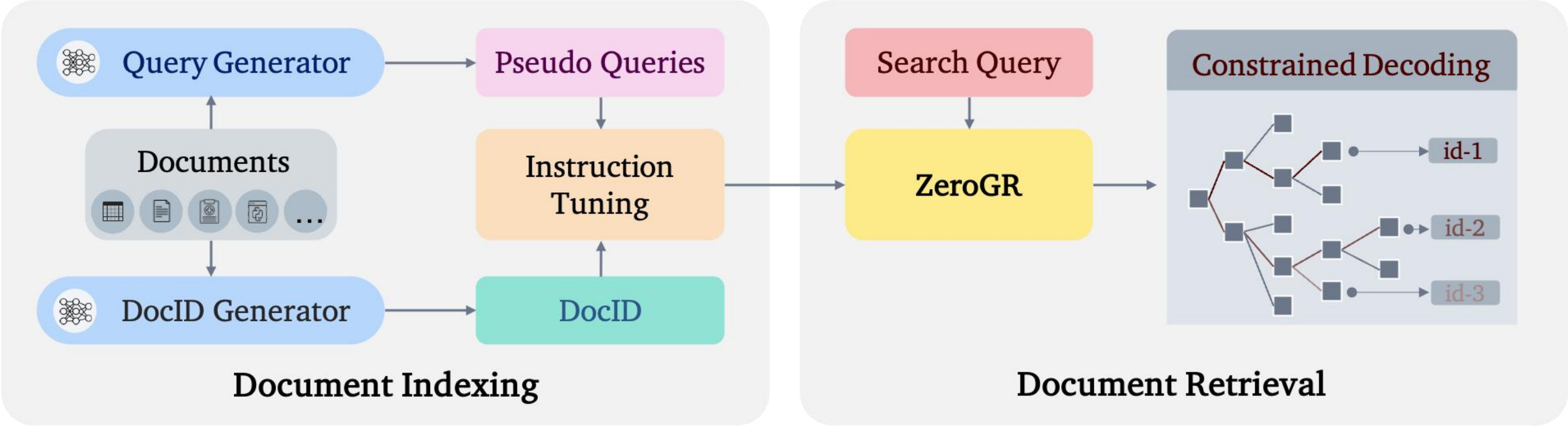
Generalization to unseen tasks

- Dense Retrieval (DR) is strong but bounded by embedding dimensionality; misses LM generative power.
- Generative Retrieval (GR) encodes corpus in parameters → generate docids at query time.
- **Gap:** GR trained in-domain struggles to generalize to *unseen* tasks (zero-shot, heterogeneous corpora, task-specific relevance).

- **Question: How to make GR *generalize* across tasks with no supervision?**

ZeroGR at a glance

- Leverages **natural-language task instructions** to adapt GR without labels.
- Three components:
  - **Unified DocID generator** $GD_{\psi_{\mathcal{D}}} \rightarrow$ short, keyword-rich docids for any modality (text/tables/code).
  - **Instructed query generator** $GD_{\theta_{\mathcal{D}}} \rightarrow$ diverse pseudo-queries from task instruction.
  - **Reverse-annealed decoding** → balanced precision/recall when generating ranked docids.



Experimental setups

- Datasets: curated from MAIR training splits + additional instruction-tuning data.
  - **Coverage:** 69 tasks across 6 domains; ~41M query–doc pairs with instructions.
  - **Domain stats (illustrative):** Medical (5), Finance (8), Academic (16), Code (13), Legal (7), Web (17).

MAIR dataset

[1] MAIR: A Massive Benchmark for Evaluating Instructed Retrieval, EMNLP 2024

Experimental setups

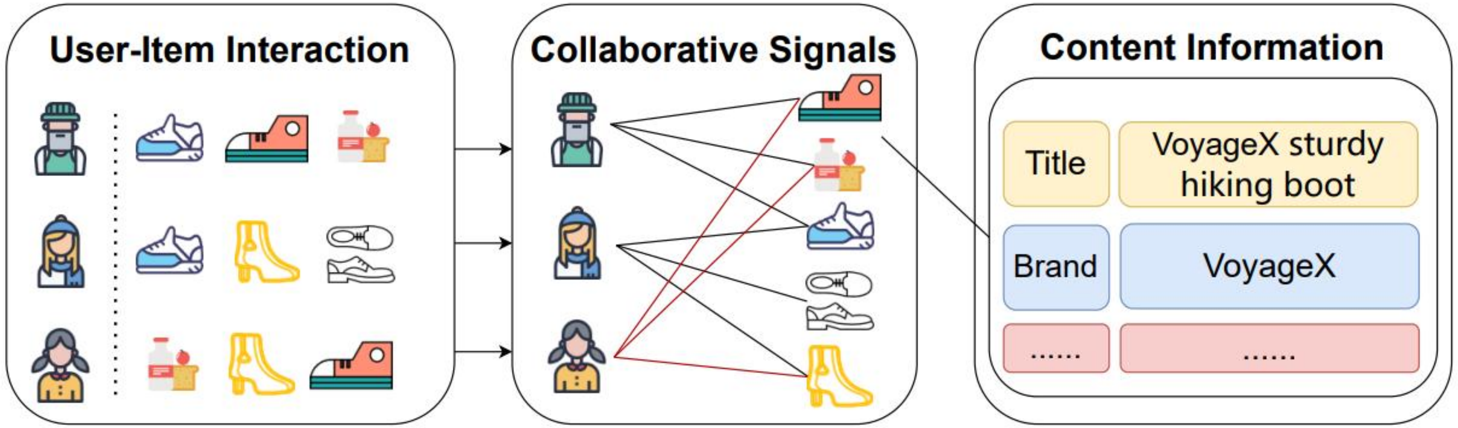
- **Evaluation benchmarks:** BEIR (12 tasks) and MAIR (seen vs. **unseen** subsets).
- **Metrics:** Acc@1, nDCG@10, Recall@100.
- **Implementation:**
  - Llama-based models for docid gen, query gen, and GR index;
  - fixed LR 5e-5; 5 epochs for 1B components.
- **Baselines:**
  - **Sparse:** BM25 (BM25S).
  - **Single-task dense:** Contriever-MARCO, GTR-base/large.
  - **Multi-task dense:** E5-Base/Large, BGE-Base/Large, OpenAI-Embed-v3-Small.
  - **Instruction-tuned dense:** E5-Mistral-7B-instr, GritLM-7B.
  - **GR competitors (for BEIR table):** GENRE, GENRET, GLEN, TIGER.

Main results on MAIR and BEIR

- ZeroGR Average **Acc@1 = 41.1** on MAIR
  - above BM25, Contriever/GTR/E5/BGE, and OpenAI-Embed-v3-Small.
- **Unseen subsets:** State-of-the-art on Apple, MB, PM.A, DD, NCL (examples) → robust transfer.
- **Efficiency:** 3B-param GR rivaling/ surpassing 7B instruction-tuned dense retrievers.

- **Average:** ZeroGR **44.9** vs GENRET **41.1**; outperforms on SciFact, FiQA, Covid, etc.
- Per-dataset highlights: best on ArguAna, SciFact, FiQA, Covid; competitive on NFCorpus.

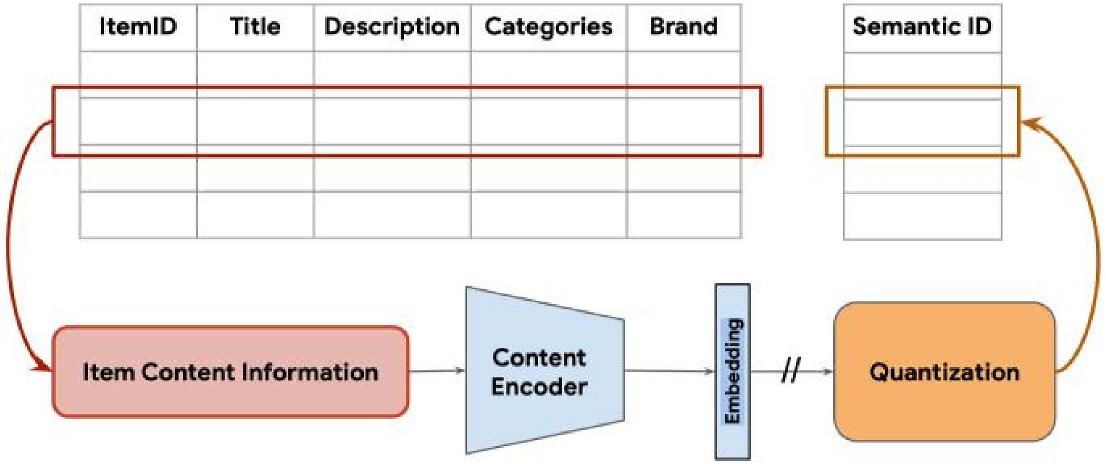
Generative recommendation



Generative models have emerged as a promising utility to enhance recommender systems.

- **Collaborative signals** refer to the knowledgecontained in user-item interactions.
- **Content information** refers to the textual description of items.

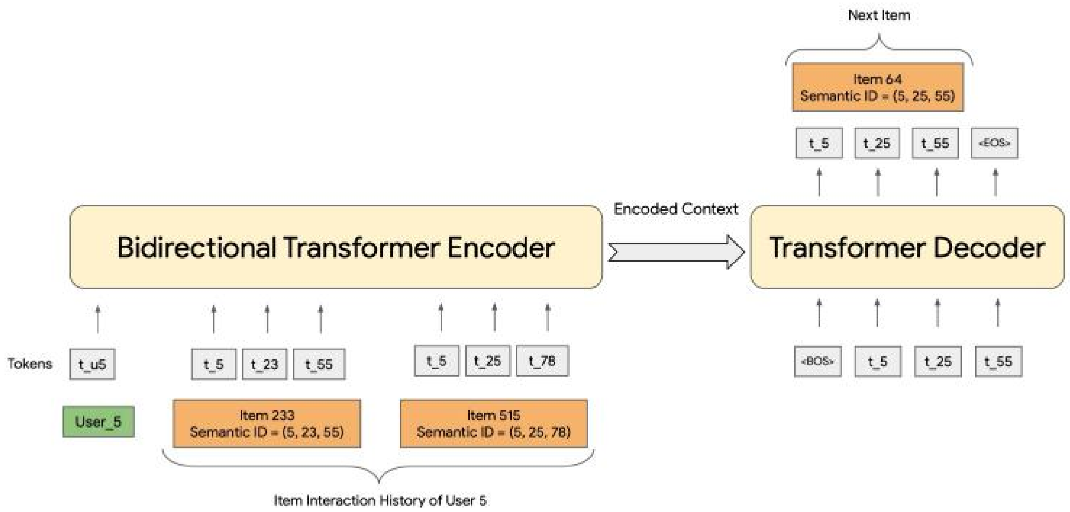
Generative recommendation: TIGER



- **Docid**: Product quantization strings
- **Docid training**: Train a residual-quantized variational autoencoder model with a docid reconstruction loss and a multi-stage quantization loss

[1] Recommender Systems with Generative Retrieval

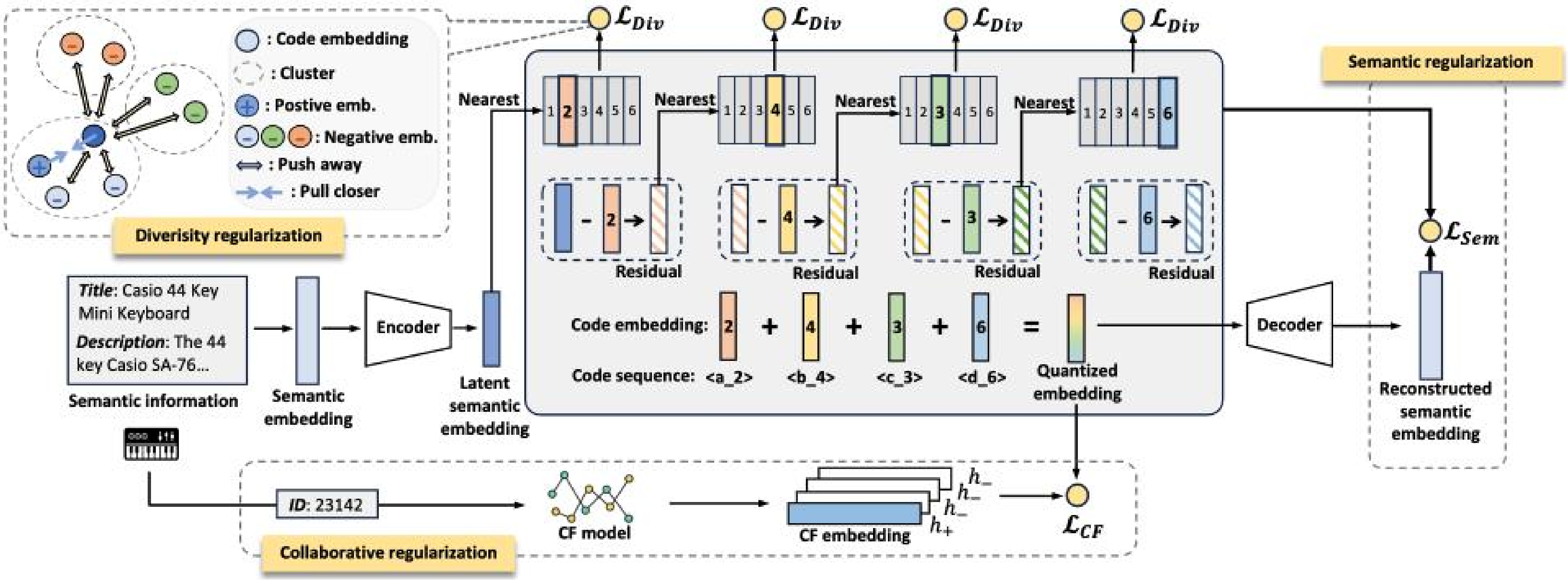
Generative recommendation: TIGER



- **Recommendation training**
  - Construct item sequences for every user by sorting chronologically the items they have interacted with
  - Given item sequences, the model is to predict the next item with MLE
- **Inference**: Beam search

[1] Recommender Systems with Generative Retrieval

LETTER – Generative Recommendation

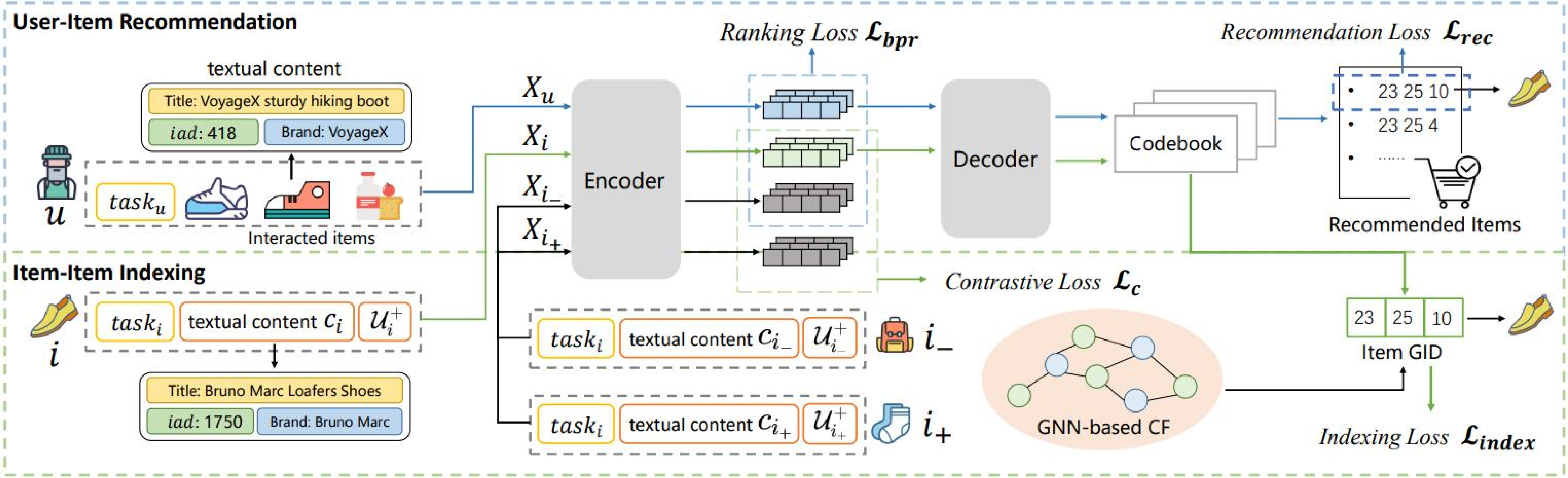


Integrate hierarchical **semantics**, **collaborative signals**, and code assignment **diversity** to satisfy the essential requirements of identifiers.

[1] Learnable Item Tokenization for Generative Recommendation.

Generative recommendation: ColaRec

- User-Item Recommendation aims to map the user's interacted items with textual content into the GID of the recommended item.
- Item-Item Indexing targets on the mapping from item side information into the item's GID.



Unifying search and recommendation

Search -> Recommendation

Query:
Hamilton Musical

rec.

Top 10 Things Hamilton Got Factually Right and Wrong
WatchMojo.com
1.4M views • 4 years ago

Search query can reflect user short-term interest, which can assist recommendation

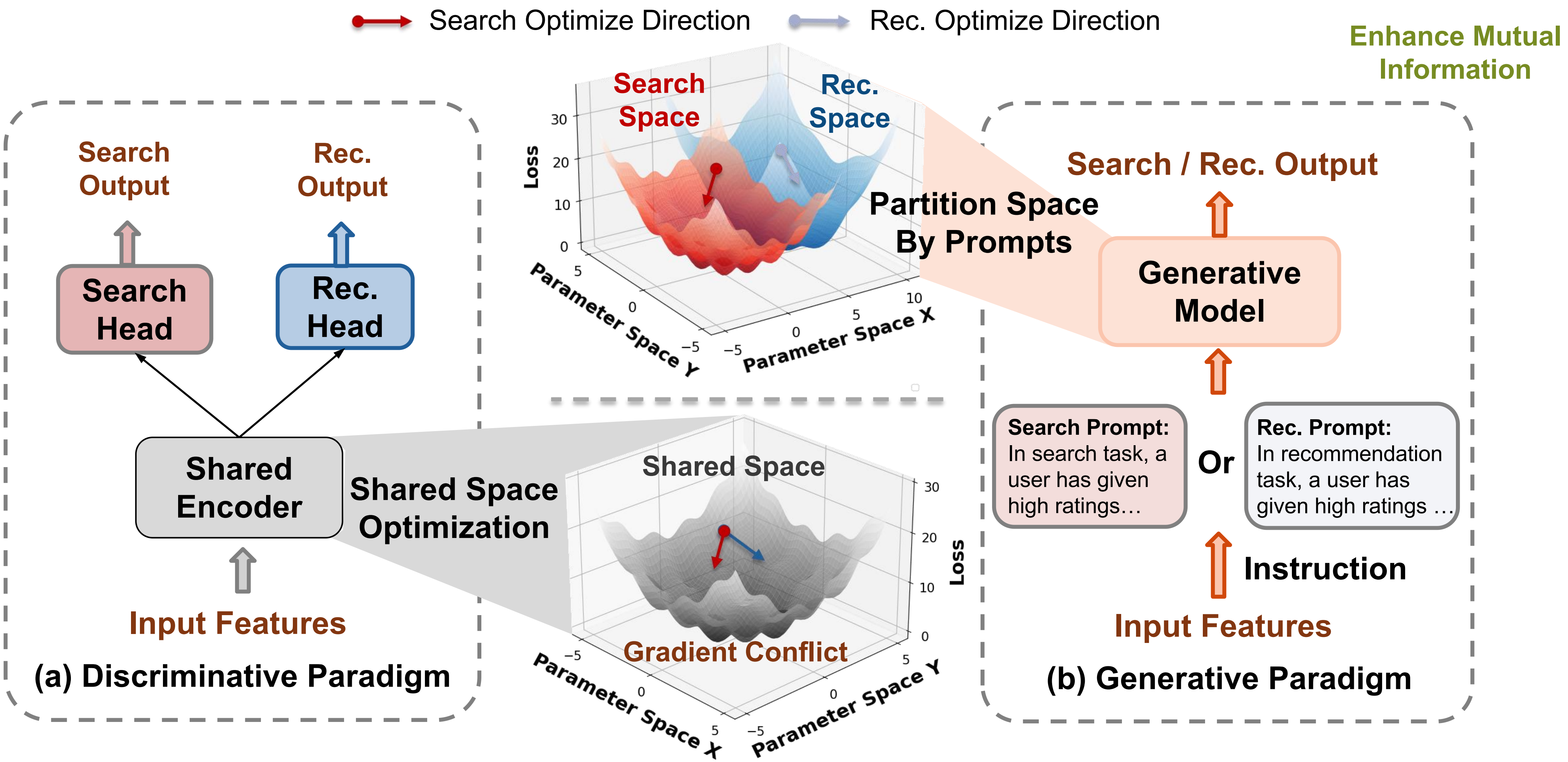
Recommendation -> Search

rec.

Query:
Action Movie

The preference shown in recommendation can offer more personalized search result.

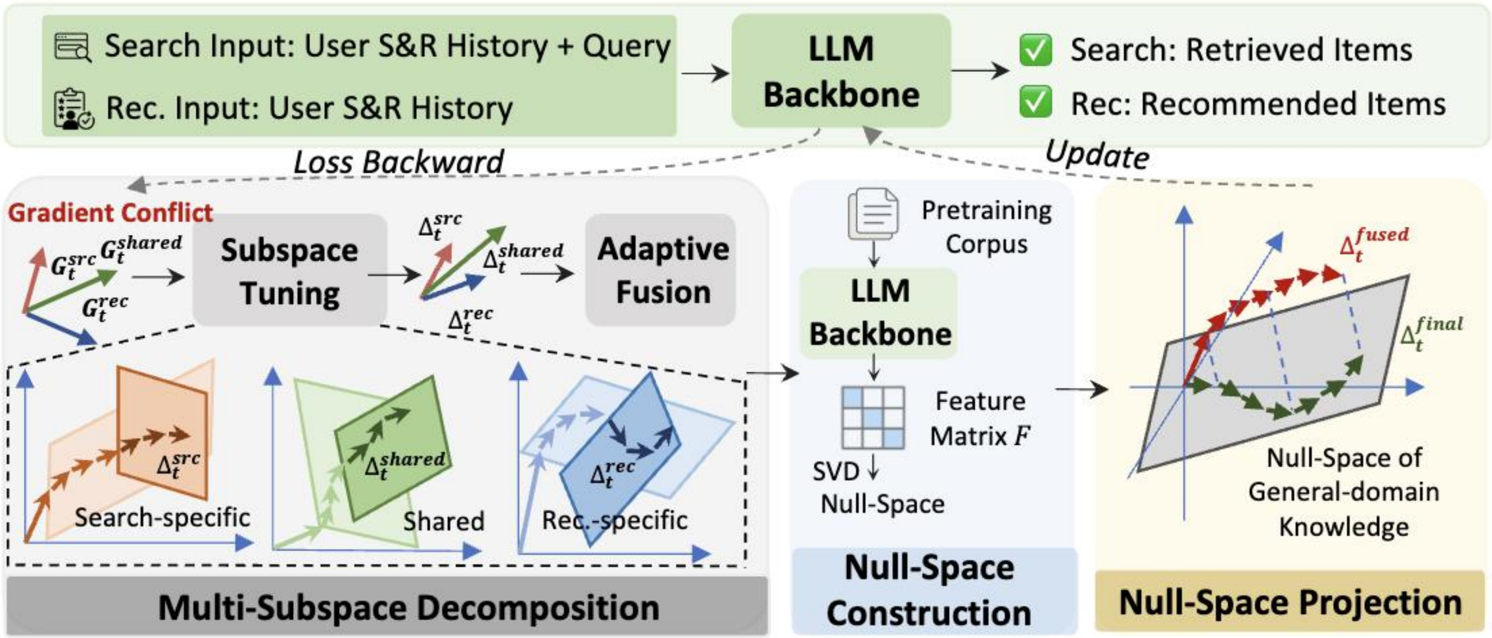
Generative paradigm



Based on this, we propose a novel Generative paradigm for unifying Search and Recommendation (S&R), abbreviated as GenSR.

Unifying search and recommendation

- **Ongoing work:**
  - We find that using PEFT to unify S\&R will cause (1) Gradient conflicts across tasks; (2) Shifts in user intent understanding.
  - We adapt **LLMs with billions of parameters** for **unified S\&R** without full fine-tuning, enabling **efficient parameter updates through subspace tuning** while mitigating gradient conflict and preserving general-domain knowledge.



Takeaways

- **Generative retrieval (GR)**
  - Learnable doclD design is the key for enhancing GR performance
  - MVDR and GR share the same relevance score framework
  - Numberic docID struggle to generalize to new documents in dynamic corpora without retraining
  - Text docID show better generalization capabilities in such dynamic settings
- **Challenges and problems in GR**
  - Constrained decoding introduces a fundamental error for GR
  - Beam search holds perfect precision but top-k recall is bounded by 0.5+o(1)
- **Generalization challenge in GR**
  - Generalize to dynamic corpora (Text-based DocIDs has strong generalization capability on new documents)
  - Generalize to unseen task (Leverages natural-language task instructions to adapt GR without labels)

Takeaways

- **Generative recommender systems**
  - DSI-like generation recommendation
  - GID design is the key for generative recommendation
  - Integrating collaborative signals and content signals
- **Unifying search and recommendation**
  - Unify search and recommendation under generative paradigm
  - Unify search and recommendation under LLMs with billions of parameters

Reference

[1] Wu, S., Ren, Z., Xin, X., Yang, J., Zhang, M., Chen, Z., ... & Ren, P. (2025, July). Constrained Auto-Regressive Decoding Constrains Generative Retrieval. In Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 2429-2440).
[2] Sun, W., Yan, L., Chen, Z., Wang, S., Zhu, H., Ren, P., ... & Ren, Z. (2023). Learning to tokenize for generative retrieval. Advances in Neural Information Processing Systems, 36, 46345-46361.
[3] Zhang, Z., Ma, X., Sun, W., Ren, P., Chen, Z., Wang, S., ... & Ren, Z. (2025, July). Replication and Exploration of Generative Retrieval over Dynamic Corpora. In Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 3325-3334).
[4] Yang, J., Li, Y., Zhao, J., Wang, H., Ma, M., Ma, J., ... & Ren, P. (2024). Uncovering Selective State Space Model's Capabilities in Lifelong Sequential Recommendation. arXiv preprint arXiv:2403.16371.
[5] Gao, S., Fang, J., Tu, Q., Yao, Z., Chen, Z., Ren, P., & Ren, Z. (2024, May). Generative news recommendation. In Proceedings of the ACM Web Conference 2024 (pp. 3444-3453).
[6] Wu, Shiguang, et al. "Generative retrieval as multi-vector dense retrieval." Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2024.
[7] Zhang, X., Xu, B., Ren, Z., Wang, X., Lin, H., & Ma, F. (2024, July). Disentangling id and modality effects for session-based recommendation. In Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval (pp. 1883-1892).
[8] Tang, Y., Zhang, R., Guo, J., & de Rijke, M. (2023, November). Recent advances in generative information retrieval. In Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region (pp. 294-297).

Reference

[9] Sun, W., Kong, K., Ma, X., Wang, S., Yin, D., de Rijke, M., Ren, Z., & Yang, Y. (2025). ZeroGR: A Generalizable and Scalable Framework for Zero-Shot Generative Retrieval. Under review.

[10] Lyu, Y., Zhang, X., Ren, Z., & de Rijke, M. (2024). Cognitive Biases in Large Language Models for News Recommendation. arXiv preprint arXiv:2410.02897.

[11] Wang, Y., Ren, Z., Sun, W., Yang, J., Liang, Z., Chen, X., ... & Xin, X. (2024, October). Content-Based Collaborative Generation for Recommender Systems. In Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (pp. 2420-2430).

[12] Yang, Z., Ren, Z., Wang, Y., Sun, H., Zhu, X., & Liao, X. (2024). Situation-aware empathetic response generation. Information Processing & Management, 61(6), 103824.

[13] Sun, W., Shi, Z., Wu, J., Yan, L., Ma, X., Liu, Y., ... & Ren, Z. (2024). MAIR: A Massive Benchmark for Evaluating Instructed Retrieval. arXiv preprint arXiv:2410.10127.

[14] Wu, S., Xin, X., Ren, P., Chen, Z., Ma, J., de Rijke, M., & Ren, Z. (2024). Learning Robust Sequential Recommenders through Confident Soft Labels. ACM Transactions on Information Systems, 43(1), 1-27.

[15] Zhao, J., Wang, W., Xu, C., Wang, X., Ren, Z., & Verberne, S. (2025). Unifying Search and Recommendation: A Generative Paradigm Inspired by Information Theory. arXiv preprint arXiv:2504.06714.

[16] Zhao, J., Wang, Z., Pan, S., Verberne, S., & Ren, Z. (2025). Unifying Search and Recommendation in LLMs via Gradient Multi-Subspace Tuning. Under review.

[17] Zhang, Z., Wang, Z., Ma, X., Wang, S., Yin, D., Xin, X., ... & Ren, Z. (2025). Model Editing for New Document Integration in Generative Information Retrieval. Under review.

Reference

Ongoing and future work will be released later. You're welcome to follow me on Google Scholar for updates.

Thanks for your attention!

Zhaochun Ren
z.ren@liacs.leidenuniv.nl

Universiteit Leiden
The Netherlands