Generative Information Retrieval

SIGIR 2024 tutorial – Sections 6 & 7

Yubao Tang[a], Ruqing Zhang[a], **Zhaochun Ren**[b], Jiafeng Guo[a] and **Maarten de Rijke**[c]

https://generative-ir.github.io/

July 14, 2024

[a] Institute of Computing Technology, Chinese Academy of Sciences & UCAS
[b] Leiden University
[c] University of Amsterdam

Section 6: Applications

A range of target tasks

Fact Verification
De Cao et al. 2021, Chen et al. 2022b, Chen et al. 2022a, Thorne et al. 2022, Lee et al. 2023

Open Domain QA
De Cao et al. 2021, Chen et al. 2022b, Zhou et al. 2022, Lee et al. 2023

Entity Linking
De Cao et al. 2021, Chen et al. 2022b, Lee et al. 2023

Knowledge-intensive language tasks

SCAN ME

A range of target tasks

Fact Verification
De Cao et al. 2021, Chen et al. 2022b, Chen et al. 2022a, Thorne et al. 2022, Lee et al. 2023

Open Domain QA
De Cao et al. 2021, Chen et al. 2022b, Zhou et al. 2022, Lee et al. 2023

Entity Linking
De Cao et al. 2021, Chen et al. 2022b, Lee et al. 2023

Multi-hop retrieval
Lee et al. 2022

Recommendation
Si et al. 2023, Rajput et al. 2023

Code retrieval
Naddem et al. 2022

More retrieval tasks

SCAN ME

A range of target tasks

Fact Verification
De Cao et al. 2021, Chen et al. 2022b, Chen et al. 2022a, Thorne et al. 2022, Lee et al. 2023

Open Domain QA
De Cao et al. 2021, Chen et al. 2022b, Zhou et al. 2022, Lee et al. 2023

Entity Linking
De Cao et al. 2021, Chen et al. 2022b, Lee et al. 2023

Multi-hop retrieval
Lee et al. 2022

Recommendation
Si et al. 2023, Rajput et al. 2023

Code retrieval
Naddem et al. 2022

Official site retrieval
Tang et al. 2023a

Industry retrieval tasks

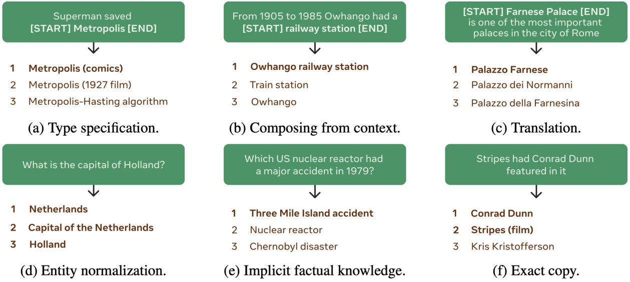
How to adapt a GR model for a task?

- Docid design
- Training approach
- Inference strategy

Knowledge-intensive language tasks

Image source: KILT

KILT example: GENRE [De Cao et al., 2021]



(a) Type specification. (b) Composing from context. (c) Translation.

(d) Entity normalization. (e) Implicit factual knowledge. (f) Exact copy.

- Entity retrieval: Entity disambiguation, document retrieval, and etc
- Corpus: Wikipedia
- Input: Query
- Output: Destination/ relevant pages' title

"Autoregressive Entity Retrieval". De Cao et al. [2021]

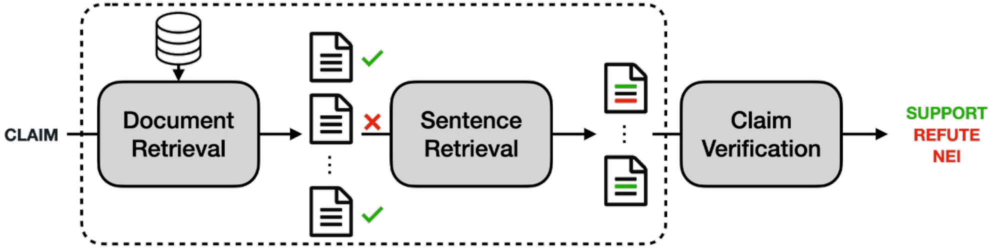
KILT example: GENRE [De Cao et al., 2021]

- **Docid**: Titles
- **Training**: MLE objective with document-title and query-title pairs
- **Inference**: Constrained beam search with a prefix tree

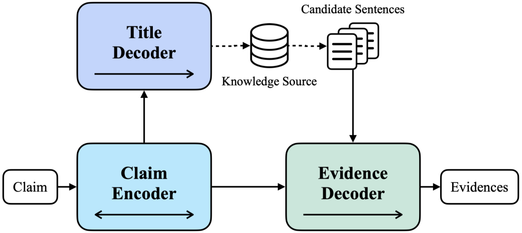
"Autoregressive Entity Retrieval". De Cao et al. [2021]

KILT example: GERE [Chen et al., 2022]

- Fact verification: Verify a claim using multiple evidential sentences from trustworthy corpora
  - Input: Claim
  - Output: Support/Refute/Not enough information



KILT example: GERE [Chen et al., 2022]



- **Docid**: Titles
- **Training**: MLE objective with claim-title and claim-evidence pairs
- **Inference**: Constrained beam search with a prefix tree

"GERE: Generative evidence retrieval for fact verification". Chen et al. [2022]

Multi-hop retrieval [Lee et al., 2022]

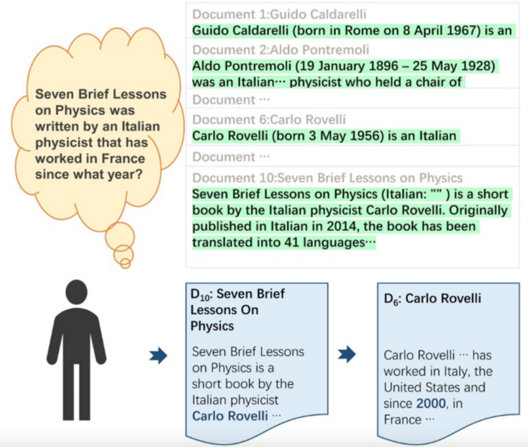
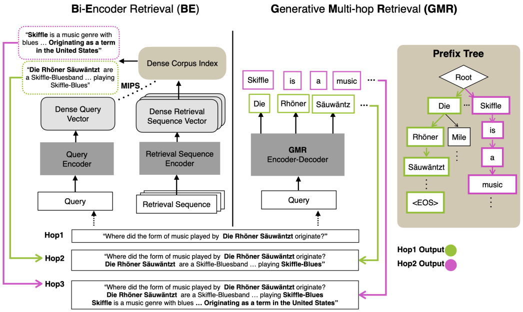


Image source: Memory enhances ChatGPT performance in multi-hop QA

- Multi-hop retrieval
  - One needs to retrieve multiple documents that together provide sufficient evidence to answer the query
  - Previously retrieved items are appended to the query while iterating through multiple hops

"Generative multi-hop retrieval". Lee et al. [2022]

Multi-hop retrieval [Lee et al., 2022]



- **Docid**: Word-based answer
- **Jointly training**:
  - Indexing: Randomly select the first *m* words of the document as input and predict the remaining words with MLE
  - Retrieval: Learn pseudo query-answer pairs with MLE
- **Inference**: Constrained beam search with a prefix tree

"Generative multi-hop retrieval". Lee et al. [2022]

Item recommendation [Rajput et al., 2023]

- Sequential recommendation: Help users discover content of interest; ubiquitous in various recommendation domains
  - Input: User history
  - Output: Next item docid

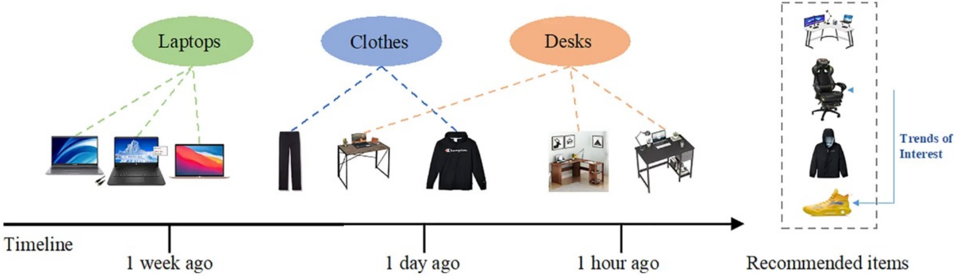
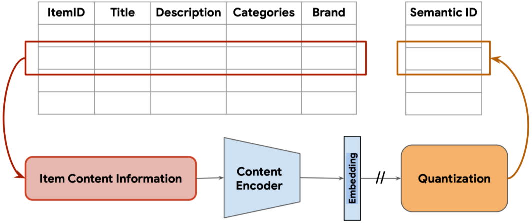


Image source: Ma et al. [2023]

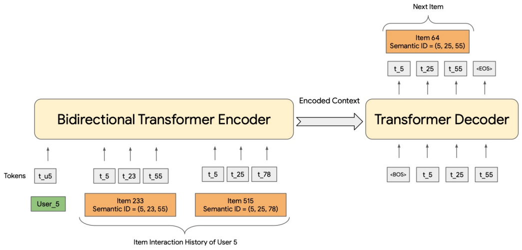
Item recommendation [Rajput et al., 2023]



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

"Recommender Systems with Generative Retrieval". Rajput et al. [2023]

Item recommendation [Rajput et al., 2023]



- **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

"Recommender Systems with Generative Retrieval". Rajput et al. [2023]

Code retrieval [Nadeem et al., 2022]

- Code retrieval: A model takes natural language queries as input and, in turn, relevant code samples from a database are returned
  - Input: Query
  - Output: Relevant code samples

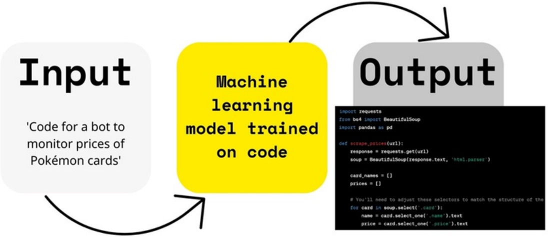
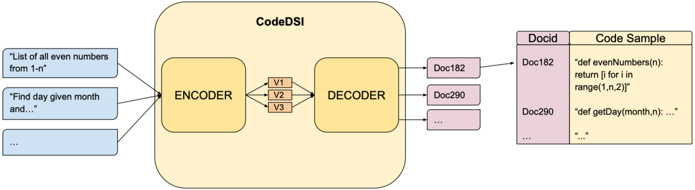


Image source: AI Code Generation Models: The Big List

Code retrieval [Nadeem et al., 2022]



- **Docid**: Naively structured strings/ semantically structured strings
- **Training**: Standard indexing loss with code-docid pairs and retrieval loss with query-docid pairs
- **Inference**: Beam search

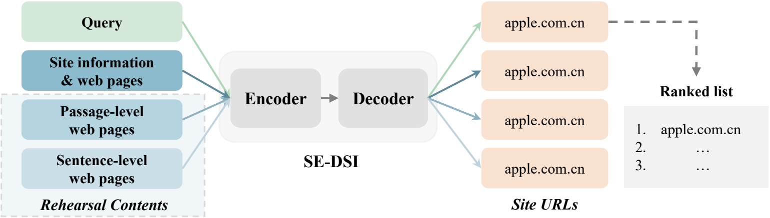
"CodeDSI: Differentiable Code Search". Nadeem et al. [2022]

Official site retrieval [Tang et al., 2023]

- Official sites: Web pages that have been operated by universities, departments, or other administrative units

"Semantic-Enhanced Differentiable Search Index Inspired by Learning Strategies". Tang et al. [2023]

Official site retrieval [Tang et al., 2023]



- **Docid**: Unique site URLs
- **Jointly training**:
  - Indexing: Learn site information (site name/ site domain/ ICP record) - docid pairs, web pages-docid pairs, and important web pages-docid pairs with MLE
  - Retrieval: Learn query - docid pairs with MLE
- **Inference**: Constrained beam search with a prefix tree

"Semantic-Enhanced Differentiable Search Index Inspired by Learning Strategies". Tang et al. [2023]

Overall performance

Tasks (Datasets)	GR method & DR baseline	Retrieval performance	Memory cost	Inference time
KILT (Wikipedia)	GENRE	83.6 RP ✓	2.1 GB ✓	-
	DPR+BERT	72.9 RP	70.9GB	-
Fact Verification- Document retrieval (FEVER)	GERE	84.3 P ✓	-	5.35ms ✓
	RAG	62.17 P	-	13.89ms
Multi-hop retrieval (EntailTree & HotpotQA)	GMR	52.5 F1 ✓	2.95 GB ✓	-
	ST5	16.9 F1	15.81GB	-
Sequential recommendation (Sports and Outdoors)	TIGER	1.81 nDCG@5 ✓	-	-
	S^3-Rec	1.61 nDCG@5	-	-
Code retrieval (CodeSearchNet)	CodeDSI	90.4 Acc ✓	-	-
	CodeBERT	89.8 Acc	-	-
Official site retrieval (Industry online data)	SE-DSI	+42.4 R@20 ✓	-31 times ✓	-2.5 times ✓
	DualEnc	-	-	-

Data source: papers mentioned above

Overall performance

Tasks (Datasets)	GR method & DR baseline	Retrieval performance	Memory cost	Inference time
KILT (Wikipedia)	GENRE	83.6 RP ✓	2.1 GB ✓	-
	DPR+BERT	72.9 RP	70.9GB	-
Fact Verification- Document retrieval (FEVER)	GERE	84.3 P ✓	-	5.35ms ✓
	RAG	62.17 P	-	13.89ms
Multi-hop retrieval (EntailTree & HotpotQA)	GMR	52.5 F1 ✓	2.95 GB ✓	-
	ST5	16.9 F1	15.81GB	-
Sequential recommendation (Sports and Outdoors)	TIGER	1.81 nDCG@5 ✓	-	-
	S^3-Rec	1.61 nDCG@5	-	-
Code retrieval (CodeSearchNet)	CodeDSI	90.4 Acc ✓	-	-
	CodeBERT	89.8 Acc	-	-
Official site retrieval (Industry online data)	SE-DSI	+42.4 R@20 ✓	-31 times ✓	-2.5 times ✓
	DualEnc	-	-	-

Data source: papers mentioned above

The performance of current GR methods can only compete with part of dense retri
baselines, but still falls short compared to full-ranking methods

Applications: limitations

- The current performance of GR can only be compared to the index-retrieval stage of certain dense retrieval methods
- Generalizing to ultra-large-scale corpora remains a challenge
- How to adapt to the significant dynamic changes in large-scale corpora for online applications

Section 7:
Challenges & Opportunities

Tutorial summary

- **Definition & preliminaries**
- **Generative retrieval: docid design**
  - Single docids: number-based and word-based identifiers
  - Multiple docids: single type and diverse types
- **Generative retrieval: training approaches**
  - Stationary scenarios: supervised learning and pre-training
  - Dynamic scenarios
- **Generative retrieval: inference strategies**
  - Single docids: constrained greedy search, constrained beam search and FM-index
  - Multiple docids: aggregation functions
- **Generative retrieval: applications**

SCAN ME

Pros of generative retrieval

Information retrieval in the era of language models

Pros of generative retrieval

Information retrieval in the era of language models

- Encode the global information in corpus; optimize in an end-to-end way
- The semantic-level association extending beyond mere signal-level matching

Pros of generative retrieval

Information retrieval in the era of language models

- Encode the global information in corpus; optimize in an end-to-end way
- The semantic-level association extending beyond mere signal-level matching
- Constraint decoding over thousand-level vocabulary
- Internal index which eliminates large-scale external index

Cons of generative retrieval: Scalability

- Large-scale real-word corpus
  - Current research can generalize from corpora of hundreds of thousands to millions
  - How to accurately memorize vast amounts of complex data?

Cons of generative retrieval: Scalability

- Large-scale real-word corpus
  - Current research can generalize from corpora of hundreds of thousands to millions
  - How to accurately memorize vast amounts of complex data?
- Highly dynamic corpora
  - Document addition, removal and updates
  - How to keep such GR models up-to-date?
  - How to learn on new data without forgetting old ones?

Cons of generative retrieval: Scalability

- Large-scale real-word corpus
  - Current research can generalize from corpora of hundreds of thousands to millions
  - How to accurately memorize vast amounts of complex data?
- Highly dynamic corpora
  - Document addition, removal and updates
  - How to keep such GR models up-to-date?
  - How to learn on new data without forgetting old ones?
- Multi-modal/granularity/language search tasks
  - Different search tasks leverage very different indexes
  - How to unify different search tasks into a single generative form?
  - How to capture task specifications while obtaining the shared knowledge?

SCAN ME

Cons of generative retrieval: Scalability

- Large-scale real-word corpus
  - Current research can generalize from corpora of hundreds of thousands to millions
  - How to accurately memorize vast amounts of complex data?
- Highly dynamic corpora
  - Document addition, removal and updates
  - How to keep such GR models up-to-date?
  - How to learn on new data without forgetting old ones?
- Multi-modal/granularity/language search tasks
  - Different search tasks leverage very different indexes
  - How to unify different search tasks into a single generative form?
  - How to capture task specifications while obtaining the shared knowledge?
- Combining GR with retrieval-augmented generation (RAG)
  - How to integrate GR with RAG to enhance the effectiveness of both?

SCAN ME

Cons of generative retrieval: Controllability

For a failure issue, it is often unclear what modeling knobs one should turn to fix the model's behavior

Cons of generative retrieval: Controllability

For a failure issue, it is often unclear what modeling knobs one should turn to fix the model's behavior

- Interpretability
  - Black-box neural models
  - How to provide credible explanation for the retrieval process and results?

Cons of generative retrieval: Controllability

For a failure issue, it is often unclear what modeling knobs one should turn to fix the model's behavior

- Interpretability
  - Black-box neural models
  - How to provide credible explanation for the retrieval process and results?
- Debuggable
  - Attribution analysis: how to conduct causal traceability analysis on the causes, key links and other factors of specific search results?
  - Model editing: how to accurately and conveniently modify training data or tune hyperparameters in the loss function?

Cons of generative retrieval: Controllability

For a failure issue, it is often unclear what modeling knobs one should turn to fix the model's behavior

- Interpretability
  - Black-box neural models
  - How to provide credible explanation for the retrieval process and results?
- Debuggable
  - Attribution analysis: how to conduct causal traceability analysis on the causes, key links and other factors of specific search results?
  - Model editing: how to accurately and conveniently modify training data or tune hyperparameters in the loss function?
- Robustness
  - When a new technique enters into the real-world application, it is critical to know not only how it works in average, but also how would it behave in abnormal situatio

Cons of generative retrieval: User-centered

Searching is a socially and contextually situated activity with diverse set of goals and needs for support that must not be boiled down to a combination of text matching and text generating algorithms [Shah and Bender, 2022]

Source: [Shah and Bender, 2022]

Cons of generative retrieval: User-centered

Searching is a socially and contextually situated activity with diverse set of goals and needs for support that must not be boiled down to a combination of text matching and text generating algorithms [Shah and Bender, 2022]

- Human information seeking behavior
- Transparency
- Provenance
- Accountability

Source: [Shah and Bender, 2022]

Cons of generative retrieval: Performance

The current performance of GR can only be compared to the index-retrieval stage of traditional methods, and it has not yet achieved the additional improvement provided by re-ranking

So much to do ...

- Closed-book: The language model is the only source of knowledge leveraged during generation, e.g.,
  - Capturing document ids in the language models
  - Language models as retrieval agents via prompting
- Open-book: The language model can draw on external memory prior to, during and after generation, e.g.,
  - Retrieve-augmented generation of answers
  - Tool-augmented generation of answers

So much to do . . .

Cater for long-term effects

- How to combine the short-term relevance goal with long-term goals such as diversity

Source: [Deffayet et al., To appear, Zhou et al., 2023]

So much to do . . .

Cater for long-term effects

- How to combine the short-term relevance goal with long-term goals such as diversity

Address needs of interactive environments

- Interactive systems must operate under high degrees of uncertainty
  - User feedback, non-stationarity, exogenous factor, user preferences, . . .

Source: [Deffayet et al., To appear, Zhou et al., 2023]

So much to do ...

Cater for long-term effects

- How to combine the short-term relevance goal with long-term goals such as diversity

Address needs of interactive environments

- Interactive systems must operate under high degrees of uncertainty
  - User feedback, non-stationarity, exogenous factor, user preferences, ...

Searching/recommending slates of items

- Interface of many search/recommendation platforms requires showing combinations of results to users on the same page
- Different combinations may lead to different short vs. long-term outcomes
- Problem thus becomes combinatorial in nature, intractable for most applicatio

Source: [Deffayet et al., To appear, Zhou et al., 2023]

Resources and sharing

Sharing more than code

- Models
- ...

Reducing compute resources

So much to do . . .

Re-invent information retrieval in the age of large language models!

Q & A

Thank you for joining us today!

All materials are available at

https://TheWebConf2024-generative-IR.github.io

References

References i

J. Chen, R. Zhang, J. Guo, Y. Fan, and X. Cheng. Gere: Generative evidence retrieval for fact verification. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2184–2189, 2022.

N. De Cao, G. Izacard, S. Riedel, and F. Petroni. Autoregressive entity retrieval. In *International Conference on Learning Representations*, 2021.

R. Deffayet, T. Thonet, D. Hwang, V. Lehoux, J.-M. Renders, and M. de Rijke. Sardine: A simulator for automated recommendation in dynamic and interactive environments. *ACM Transactions on Recommender Systems*, To appear.

H. Lee, S. Yang, H. Oh, and M. Seo. Generative multi-hop retrieval. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1417–1436, 2022.

J. Ma, T. Sun, and X. Zhang. Time highlighted multi-interest network for sequential recommendation. *Computers, Materials & Continua*, 76(3), 2023.

U. Nadeem, N. Ziems, and S. Wu. Codedsi: Differentiable code search. *arXiv preprint arXiv:2210.00328*, 2022.

SCAN ME

S. Rajput, N. Mehta, A. Singh, R. H. Keshavan, T. Vu, L. Heldt, L. Hong, Y. Tay, V. Q. Tran, J. Samost, et al. Recommender systems with generative retrieval. *arXiv preprint arXiv:2305.05065*, 2023.

C. Shah and E. M. Bender. Situating search. In *Proceedings of the 2022 Conference on Human Information Interaction and Retrieval*, pages 221–232, 2022.

Y. Tang, R. Zhang, J. Guo, J. Chen, Z. Zhu, S. Wang, D. Yin, and X. Cheng. Semantic-enhanced differentiable search index inspired by learning strategies. In *29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023.

Y. Zhou, Z. Dou, and J.-R. Wen. Enhancing generative retrieval with reinforcement learning from relevance feedback. In *EMNLP 2023: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.