Generative Information Retrieval

SIGIR 2024 tutorial – Section 1

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

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

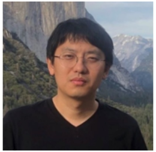
July 14, 2024

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

About presenters

Yubao Tang
PhD student
@ICT, CAS

Ruqing Zhang
Faculty
@ICT, CAS



Zhaochun Ren
Faculty
@LEI

Jiafeng Guo
Faculty
@ICT, CAS

Maarten de Rijke
Faculty
@UvA

Information retrieval

Information retrieval (IR) is the activity of obtaining information system resources that are relevant to an information need from a collection of those resources.

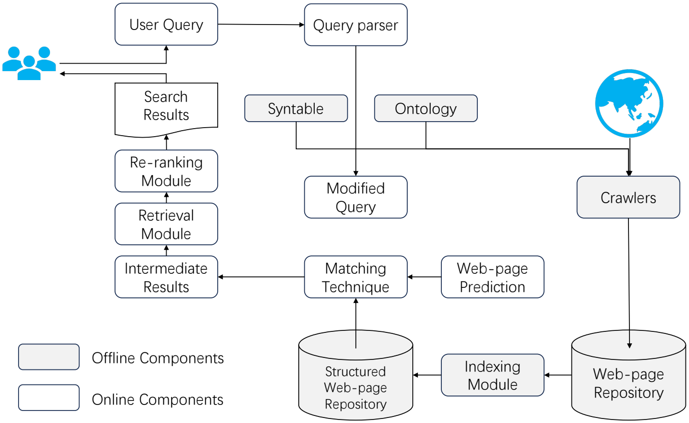
Information need
Relevance Relationship
Information repository

Given: User query (keywords, question, image, . . .)

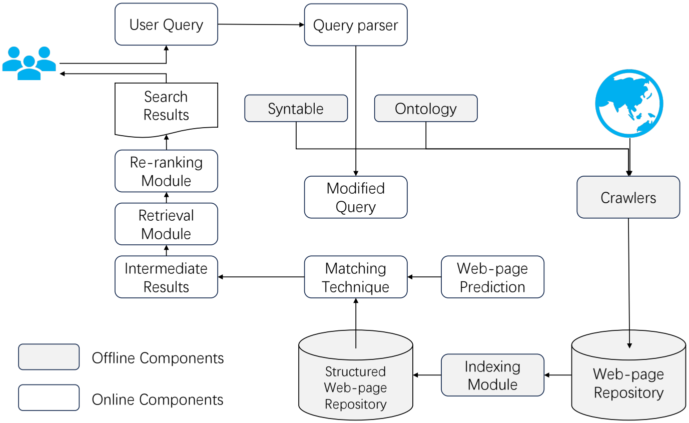
Rank: Information objects (passages, documents, images, products, . . .)

Ordered by: Relevance scores

Complex architecture design behind search engines



Complex architecture design behind search engines



User Query

Query parser

Search Results

Syntable

Ontology

Re-ranking Module

Modified Query

Crawlers

Retrieval Module

Intermediate Results

Matching Technique

Web-page Prediction

▨ Offline Components

□ Online Components

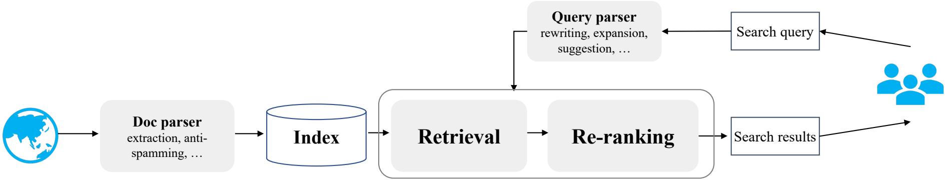
Structured Web-page Repository

Indexing Module

Web-page Repository

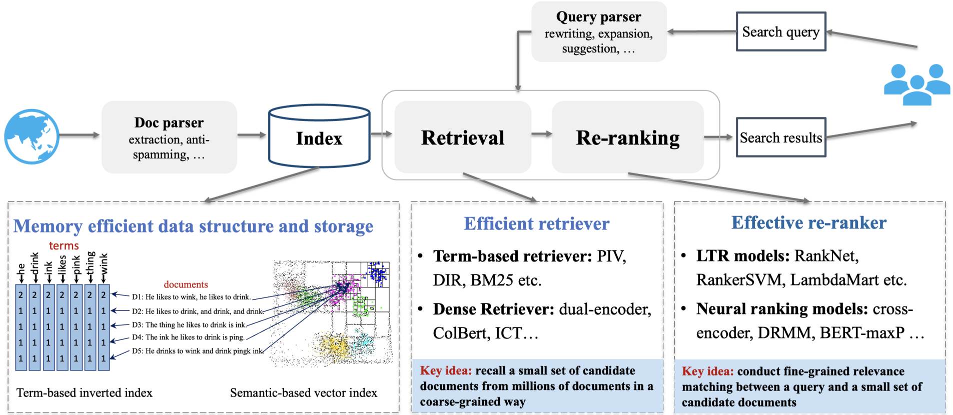
- **Advantages**:
  - Pipelined paradigm has withstood the test of time
  - Advanced machine learning and deep learning approaches applied to many components of modern systems

Core pipelined paradigm: 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

Index-Retrieval-Ranking: Disadvantages



Query parser
rewriting, expansion, suggestion, …

Search query

Doc parser
extraction, anti-spamming, …

Index

Retrieval

Re-ranking

Search results

Memory efficient data structure and storage

D1: He likes to wink, he likes to drink.
D2: He likes to drink, and drink, and drink.
D3: The thing he likes to drink is ink.
D4: The ink he likes to drink is ping.
D5: He drinks to wink and drink pingk ink.

Term-based inverted index

Semantic-based vector index

Efficient retriever

- **Term-based retriever:** PIV, DIR, BM25 etc.
- **Dense Retriever:** dual-encoder, ColBert, ICT…

Key idea: recall a small set of candidate documents from millions of documents in a coarse-grained way

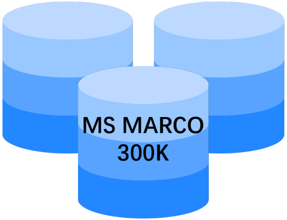
Effective re-ranker

- **LTR models:** RankNet, RankerSVM, LambdaMart etc.
- **Neural ranking models:** cross-encoder, DRMM, BERT-maxP …

Key idea: conduct fine-grained relevance matching between a query and a small set of candidate documents

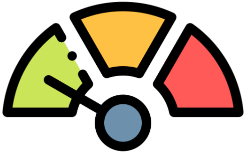
- **Effectiveness**: Heterogeneous ranking components are usually difficult to be optimized in an end-to-end way towards the global objective

Index-Retrieval-Ranking: Disadvantages



Big storage

GTR (Dense retrieval)
Memory size 1430MB



Slow inference speed

GTR (Dense retrieval)
Online latency 1.97s

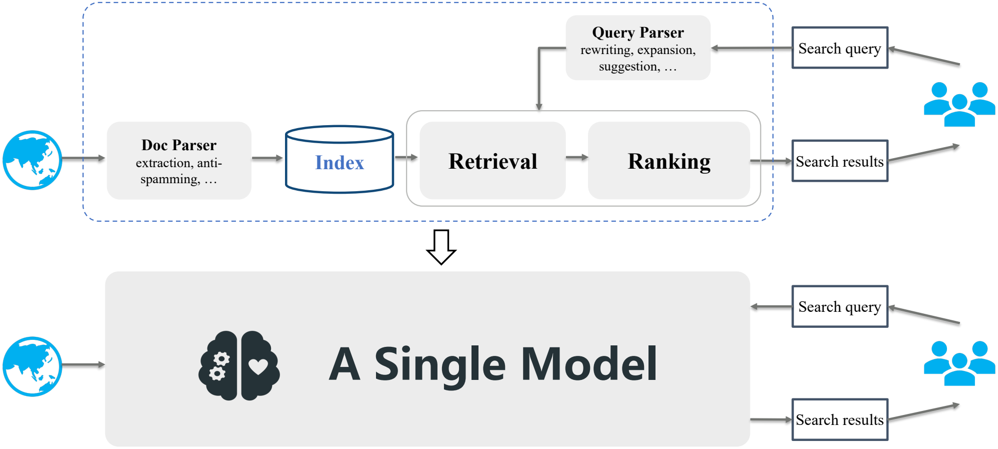
- **Efficiency**: A large document index is needed to search over the corpus, leading to significant memory consumption and computational overhead

Source: [Sun et al., 2023]

What if we replaced the pipelined architecture with a single consolidated model that efficiently and effectively encodes all of the information contained in the corpus?

SCAN ME

Opinion paper: A single model for IR



Source: [Metzler et al., 2021]

Generative language models

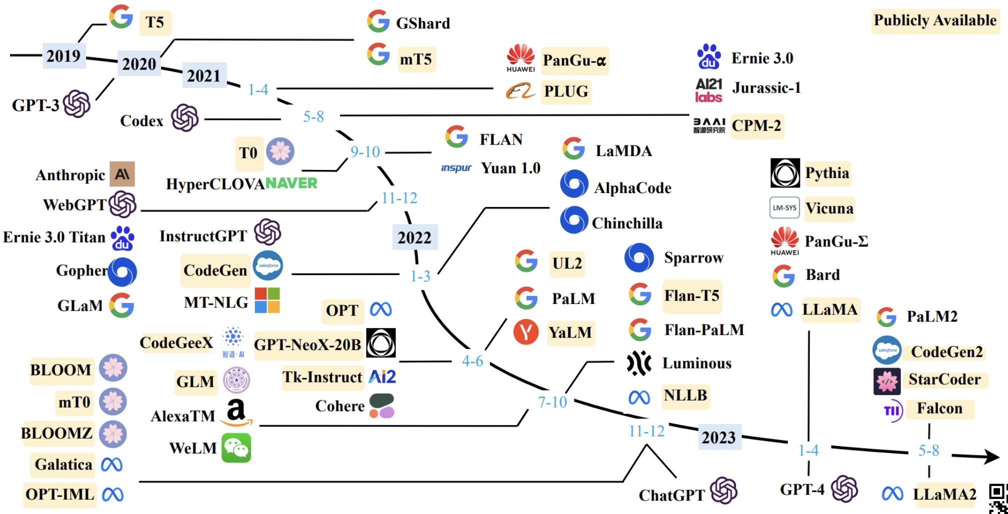


Image source: [Zhao et al., 2023]

Two families of generative retrieval

- 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.,
  - Retrieval augmented generation of answers
  - Tool-augmented generation of answers

Source: [Najork, 2023]

Two families of generative retrieval

- 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.,
  - Retrieval augmented generation of answers
  - Tool-augmented generation of answers

Source: [Najork, 2023]

Closed-book generative retrieval

The IR task can be formulated as a sequence-to-sequence (Seq2Seq) generation problem

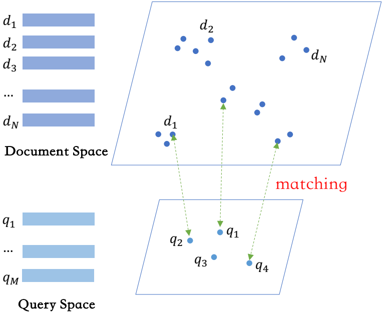
Closed-book generative retrieval

The IR task can be formulated as a sequence-to-sequence (Seq2Seq) generation problem

- **Input**: A sequence of query words
- **Output**: A sequence of document identifiers

Neural IR models: Discriminative vs. Generative

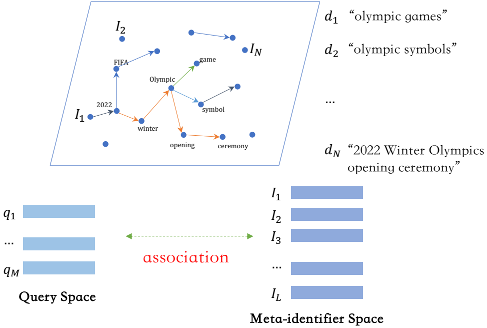
Discriminative



$$p(R=1|q,d) \approx \;\ldots\; \approx argmax\; s(\vec{q},\vec{d})$$

(probabilistic ranking principle)

Generative

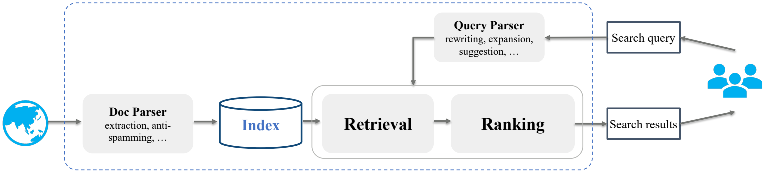


$$p(q|d) \approx p(docID|q) = argmax\; \mathrm{p}((I_1,\ldots,I_k)|q)$$

(query likelihood)

Why generative retrieval?

Heterogeneous objectives



Query Parser
rewriting, expansion, suggestion, …

Search query

Doc Parser
extraction, anti-spamming, …

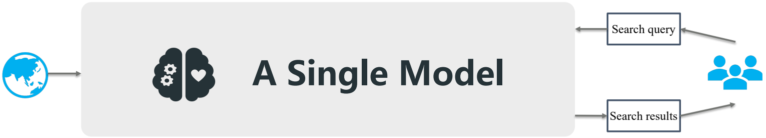
Index

Retrieval

Ranking

Search results

A global objective



A Single Model

Search query

Search results

- **Effectiveness**: Knowledge of all documents in corpus is encoded into model parameters, which can be optimized directly in an end-to-end manner

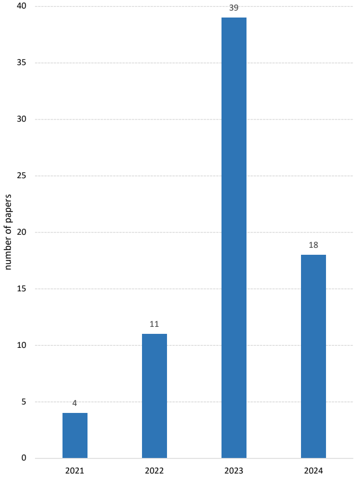
Why generative retrieval?

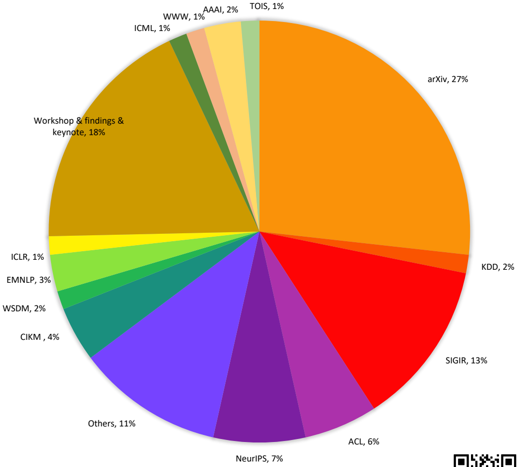
	Dense retrieval	Generative retrieval
Memory size (MS MARCO 300K)	**GTR** **1430MB**	**GenRet** **860MB**
Online latency	**GTR** **1.97s**	**GenRet** **0.16s**

- **Efficiency**: Main memory computation of GR is the storage of document identifiers and model parameters
- Heavy retrieval process is replaced with a light generative process over the vocabulary of identifiers

Data source: [Sun et al., 2023]

Statistics of related publications





The data statistics cover up to July 10, 2024.

SCAN ME

Goals of the tutorial

- We will cover key developments on generative information retrieval (mostly 2021–2024)
    - **Problem definitions**
    - **Docid design**
    - **Training approaches**
    - **Inference strategies**
    - **Applications**

Goals of the tutorial

- We will cover key developments on generative information retrieval (mostly 2021–2024)
  - **Problem definitions**
  - **Docid design**
  - **Training approaches**
  - **Inference strategies**
  - **Applications**
- We are still far from understanding how to best develop generative IR architecture compared to traditional pipelined IR architecture:
  - Taxonomies of existing research and key insights
  - Our perspectives on the **current challenges** & **future directions**

Schedule

Time	Section	Presenter
13:30-13:50	Section 1: Introduction	Maarten de Rijke
13:50-14:20	Section 2: Definitions & Preliminaries	Zhaochun Ren
14:20-15:00	Section 3: Docid design	Yubao Tang
	15min coffee break	
15:15-15:55	Section 4: Training approaches	Weiwei Sun
15:55-16:15	Section 5: Inference strategies	Weiwei Sun
16:15-16:35	Section 6: Applications	Yubao Tang
16:35-16:50	Section 7: Challenges & Opportunities	Maarten de Rijke
16:50-17:00	Q & A	All

References

References i

D. Metzler, Y. Tay, D. Bahri, and M. Najork. Rethinking search: Making domain experts out of dilettantes. *SIGIR Forum*, 55(1):1–27, 2021.

M. Najork. Generative information retrieval (slides), 2023. URL https://docs.google.com/presentation/d/191AeVzPkh2OLy855tKDkz1uv-1pHV_9GxfntiTJPUug/.

W. Sun, L. Yan, Z. Chen, S. Wang, H. Zhu, P. Ren, Z. Chen, D. Yin, M. de Rijke, and Z. Ren. Learning to tokenize for generative retrieval. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.