Retrieval and Retrieval Augment Generation (RAG)

JIAWEI HAN
SIEBEL SCHOOL OF COMPUTING AND DATA SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN
JUNE 29, 2026

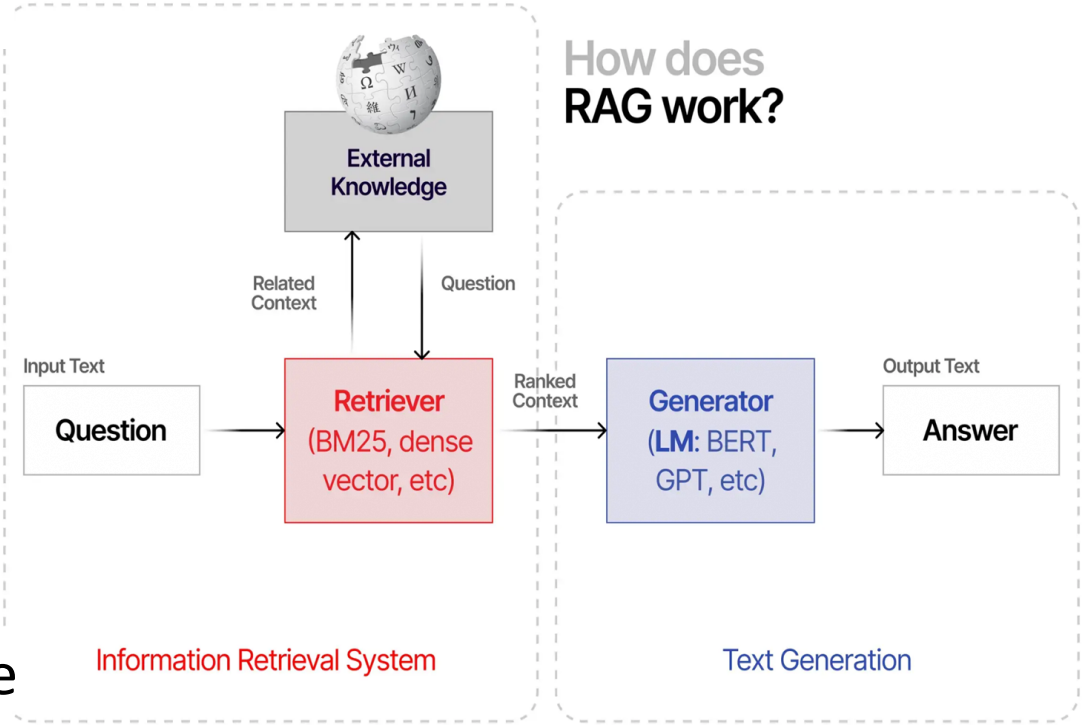
Outline

- Retrieval Augment Generation (RAG): A Brief introduction
- Retrieval and Generation Architecture for RAG
- Enhancing Information Retrieval with Corpus Knowledge and Semantic Structures
- Retrieval Improvement: Query/Pseudo-document generation and reranking
- Advanced RAG Methods: Active-RAG, Self-RAG, GraphRAG, HippoRAG, G-Retriever
- RAG by Exploring Semantic Structures: Hypercube, Structure RAG
- Exploring RAG Applications: MolE-RAG
- Open Challenges and Future Directions

What Is Retrieval-Augmented Generation (RAG)?

- Retrieval-Augmented Generation (RAG): A paradigm that combines information retrieval systems with large language models (LLMs).
  - Instead of relying solely on knowledge stored in model parameters, RAG **dynamically retrieves relevant external information and conditions** generation on the retrieved evidence
    - Enhance the accuracy and credibility of the generation
    - Allow for continuous knowledge updates and integration of domain-specific information
    - Synergistically merge LLMs' intrinsic knowledge with the vast, dynamic external data
- Essential components of RAG:
  - Retrieval → Augmentation → Generation

Image: https://markovate.com/blog/connect-llm-using-rag/



P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. tau Yih, T. Rocktäschel, S. Riedel, and D. Kiela. **Retrieval-augmented generation for knowledge-intensive NLP tasks**. In NeurIPS, 2020.

Retrieval Augmented Generation: A Brief History

- The original RAG framework: Introduced by researchers at Meta AI in 2020
  - P. Lewis, et al., "*Retrieval-augmented generation for knowledge-intensive NLP tasks*"
- **Phase I: Classical Retrieval + Neural Generation (2020–2022)**
  - Architecture: **Query → Retriever → Top-k Documents → Generator → Answer**
- **Phase II: Foundation Models + RAG (2023)**
  - **Query → Embedding Model → Vector Database → Retrieved Chunks → LLM**
  - Chunking strategies, embedding learning, vector databases, context engineering
    - Examples: LangChain RAG, LlamaIndex, Enterprise QA systems
- **Phase III: Agentic and Advanced RAG (2024–Present)**
  - ***Multi-step Retrieval:*** The model retrieves repeatedly while reasoning
    - Examples: Self-RAG, FLARE, IRCoT
  - ***Agentic RAG:*** LLMs act as retrieval planners
    - Search, query rewriting, tool selection, evidence verification

Outline

- Retrieval Augment Generation (RAG): A Brief introduction
- Retrieval and Generation Architecture for RAG
- Enhancing Information Retrieval with Corpus Knowledge and Semantic Structures
- Retrieval Improvement: Query/Pseudo-document generation and reranking
- Advanced RAG Methods: Active-RAG, Self-RAG, GraphRAG, HippoRAG, G-Retriever
- RAG by Exploring Semantic Structures: Hypercube, Structure RAG
- Exploring RAG Applications: MolE-RAG
- Open Challenges and Future Directions

Retrieval in the RAG Framework

- **Retrieval:** Given a query, retrieval is to find relevant documents from external knowledge sources
  - Prepare information from raw data
  - Chunk documents and create embeddings
  - Store embeddings into a vector database
  - Embed the query and retrieve relevant documents

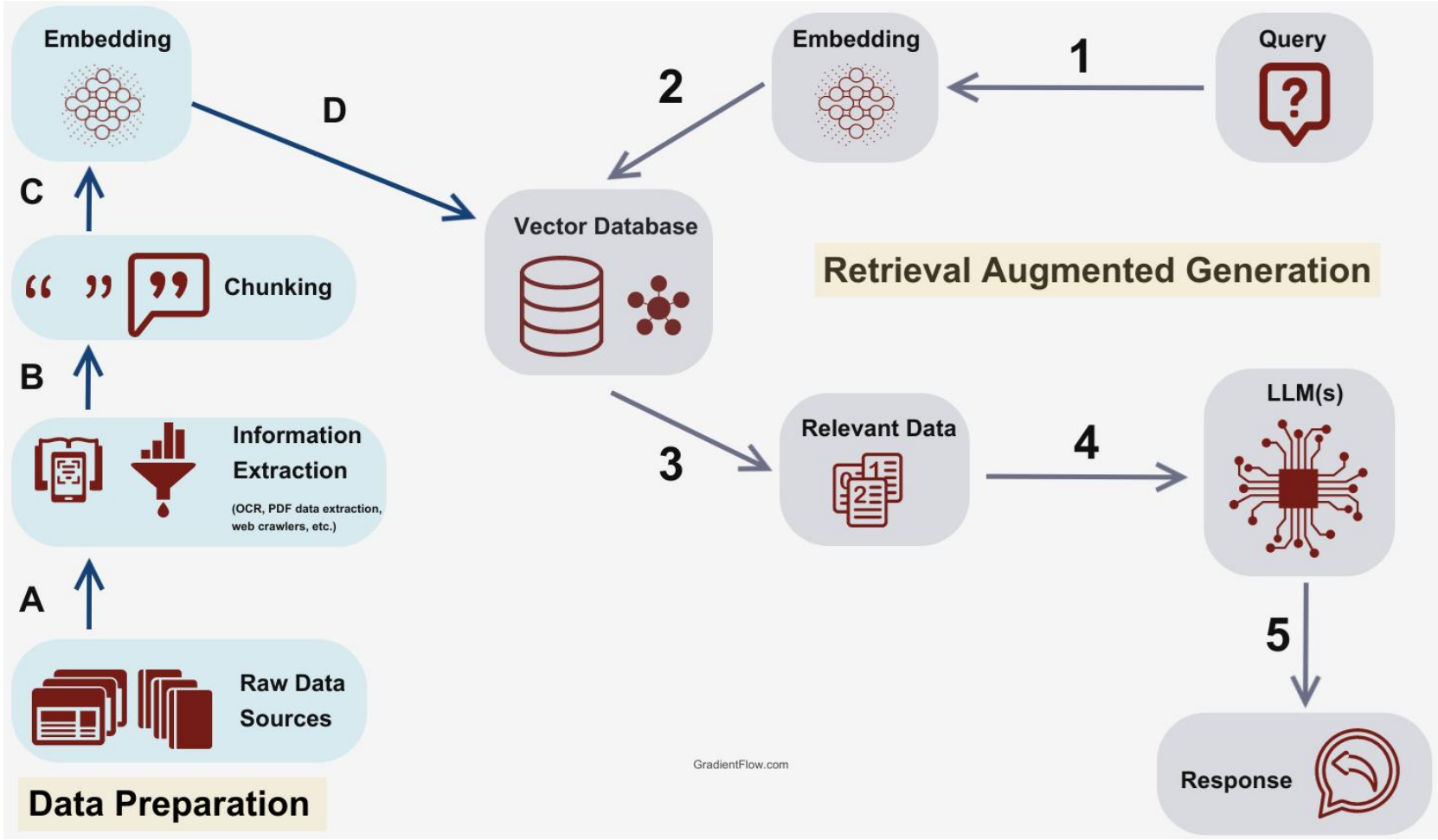
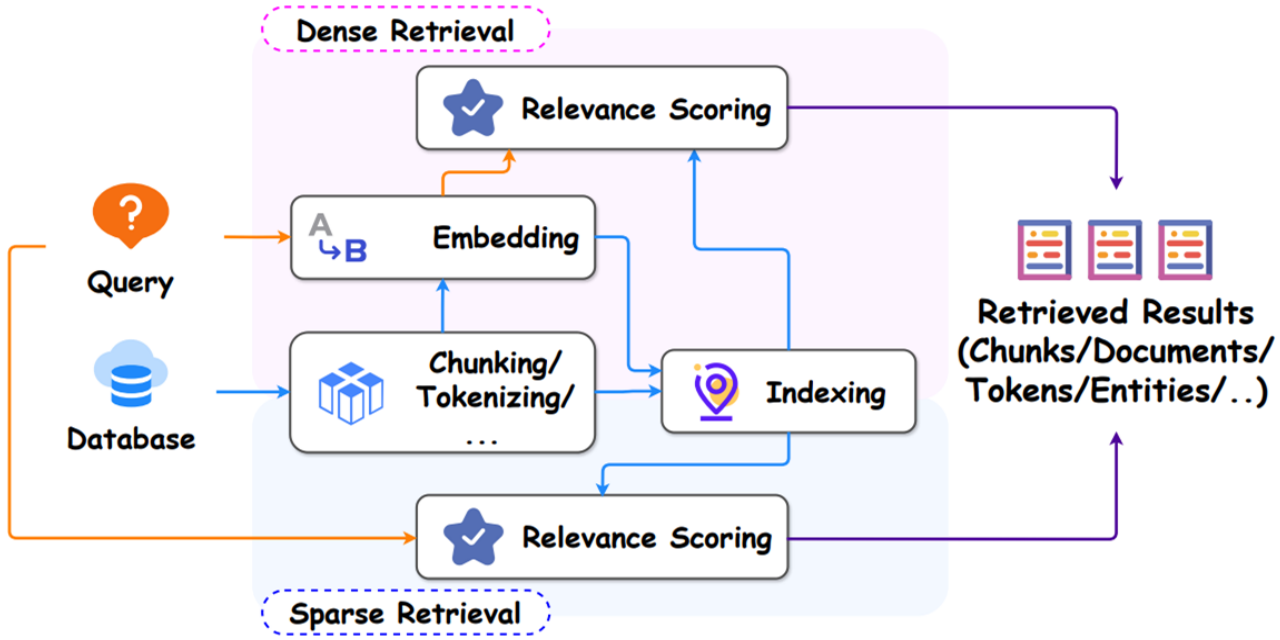


Image: https://gradientflow.com/techniques-challenges-and-future-of-augmented-language-models/

Retriever: Sparse vs. Dense Retrieval

- **Sparse Retrieval**: Word-based, text retrieval
  - Ex.: TF-IDF (BM25) with inverted index matching of raw data input
  - Disadv.: not able to retrieve context where words don't match as it is
- **Dense retrieval**: Embeds queries and docs into vector spaces
  - Adv.: Trainable, holding more flexibility and potential in adaptation
  - Ex.: Contriever, ColBERT, Specter, E5, Qwen3-Embedding
  - Encoders are typically pre-trained with general corpus (e.g., wikipedia)



- **Hybrid Approach:** Combines dense vector representations with sparse one-hot vectors

Hybrid vector approaches: Take advantage of the computational efficiency of sparse dot products over dense vector operations. (Luan, Yi; et al., "Sparse, Dense, and Attentional Representations for Text Retrieval", TACL (2021), arxiv:2005.00181)

Dense Retrieval Methods

- **Methods:** One-encoder retriever vs. bi-encoder
  - **One-encoder retriever**: *Pre-trained on large-scale unaligned documents by contrastive learning*
    - *Versatility* (i.e., generalizing better to new domains/tasks) (e.g., Contriever, Spider)
  - **Bi-encoder**: *Construct two-stream encoders (one for the query and the other for the documents)*
    - Learn embeddings with a small number of question and gold passage pairs via a simple dual-encoder framework
    - Ex.: Dense Passage Retriever (DPR): Uses a BERT-based backbone and is pre-trained specifically for the OpenQA task with question-answer pair data
    - DPR model finetuned on target datasets has better performance than TFIDF or One-encoder
- A **dense passage retriever (DPR):** Fetch relevant passages with regards to the query based on the similarity between the high-quality low-dimensional continuous representation of passages and queries

$$\text{sim}(q,p) = E_Q(q)^\intercal E_P(p)$$

q, *p*, E_Q, and E_P correspond to query text, passage text, BERT model that outputs query representation, and BERT model that outputs passage representation.

For training, dataset is represented as $D = {<$q_1$, p_1, p_2, p_3.. p_n>, <q_2, p_2, p_1, p_3.. p_n> ...}$, here q_i is the *i*-th query, and each q is paired with its positive example and some negative examples

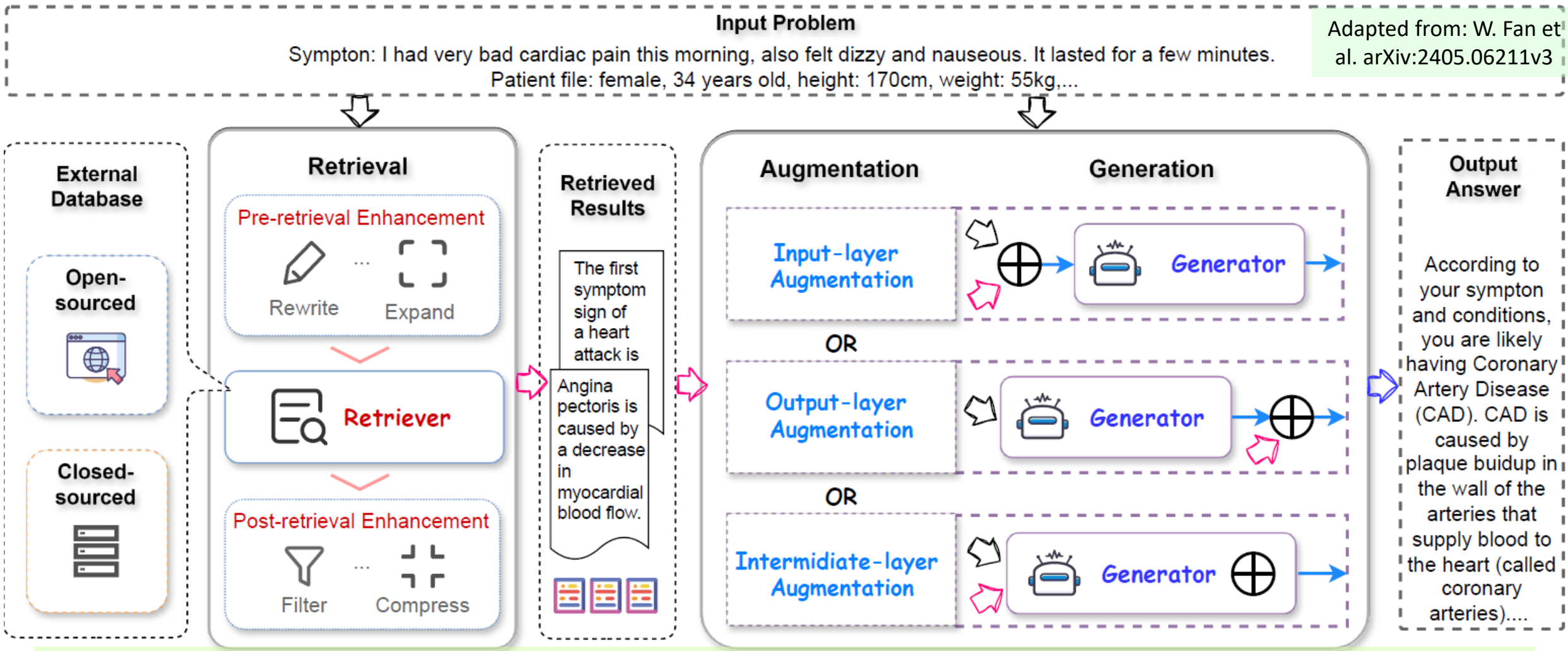
Retrieval Granularity: Passage vs. Entity

- **Retrieval granularity**: The retrieval unit in which the corpus is indexed, e.g., *document, passage, token, or other levels like entity, phrase, sentence, proposition*
  - Coarse-grained units can provide more relevant info, but may lead to redundancy
  - A text *chunk/passage* may contain compact and complete info with less redundancy and irrelevancy, becoming the mainstream retrieval text granularity in RAG
  - *Token* retrieval is more suitable in cases requiring rare patterns or out-of-domain data
- **Entity retrieval**: Designed from the perspective of knowledge rather than language
  - In KG, retrieval granularity includes Entity, Triplet, and sub-Graph
  - Entities as Experts (EAE) model divides the parameter space of LMs according to entity
  - Learn entity representations from the text along with other model parameters with the Wikipedia database and represent knowledge with entity memory
  - More effective for entity-centric tasks & more efficient in space compared to token-wise retrieval

Retrieval Sources, Datasets, and Indexing

- **Retrieval Sources**: Text, semi-structured data (PDF, table) and structured data (KG) (may need to incorporate TableGPT, KnowledgeGPT, G-retriever (GNN)), or LLM generated content)
- **Closed-sourced DB**: Stores key-value pairs for knowledge, which can be constructed in various ways
  - Wikipedia is one of the most commonly applied general retrieval sets in early RAG work
  - Domain-specific database is also used for downstream tasks
- **Indexing**: Docs will be processed, segmented & transformed into Embeddings to be stored
  - **Chunking Strategy**: Split a doc into chunks on a fixed number (e.g., 100, 256, 512) of tokens
    - Larger chunks can capture more context, but also generate more noise, higher costs
    - *Small2Big*: Use sentences (small) as unit, preceding/following sentences as (big) context to LLMs
  - **Metadata Attachments**: Chunks can be enriched with metadata
  - **Structural Index**: Use a hierarchical structure for docs
  - **KG index**: Utilize KG in constructing the hierarchical structure of documents and build an index between multiple documents using KG
- **Open-source**: Applying Internet searching engine avoids the maintenance of search index and can access up-to-date knowledge

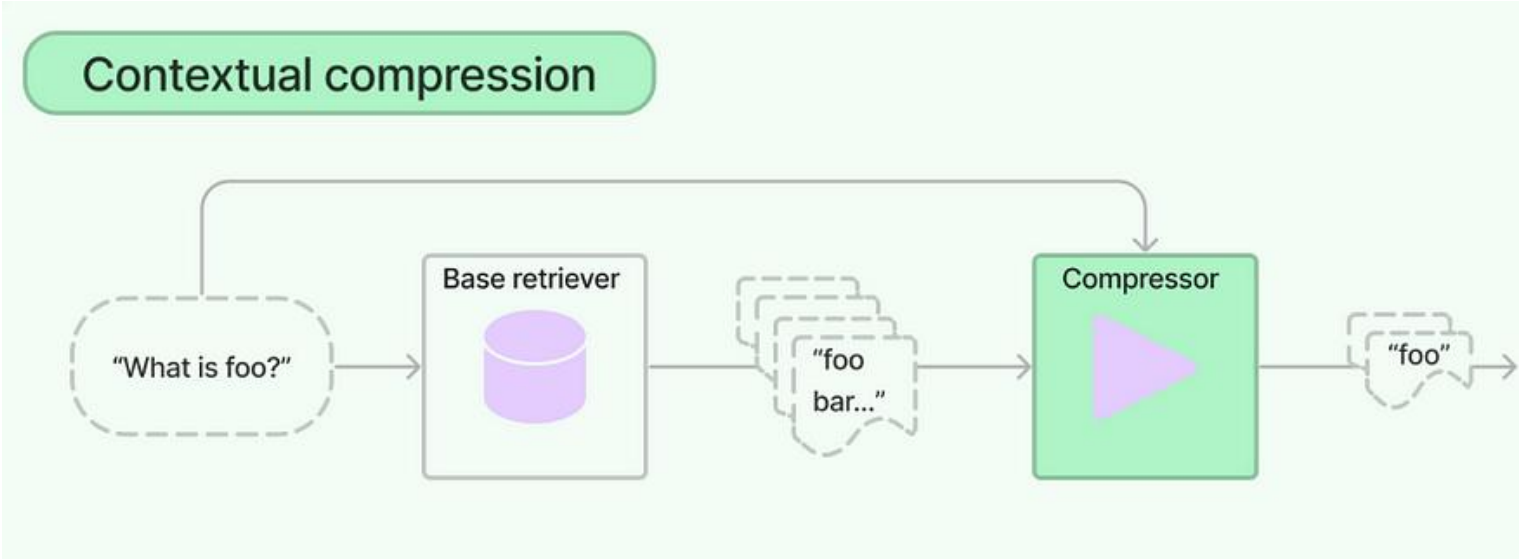
RAG Framework for QA Task



RA framework for a QA task consists of three main components: retrieval, augmentation, and generation. Retrieval may have different procedures with various designs, which optionally includes pre-retrieval and post-retrieval processes. The retrieved documents are further leveraged in generation with the augmentation module, which may be at different integration stages.

Generation in RAG

- **Generator:** Use retrieved data to construct a detailed and coherent response
- **Generator Types**
  - **White-box:** Allow parameter optimization, which can be trained to adapt to different retrieval and augmentation approaches
  - **Black-box**: Only allow prompting without access to the internal structure
- **Challenge: redundant or overly long contexts**
  - **Context compression**: Excessive context introduces noise, diminishes the perception of key info.
  - **Reranking**: Reorder doc chunks to highlight the most pertinent results first

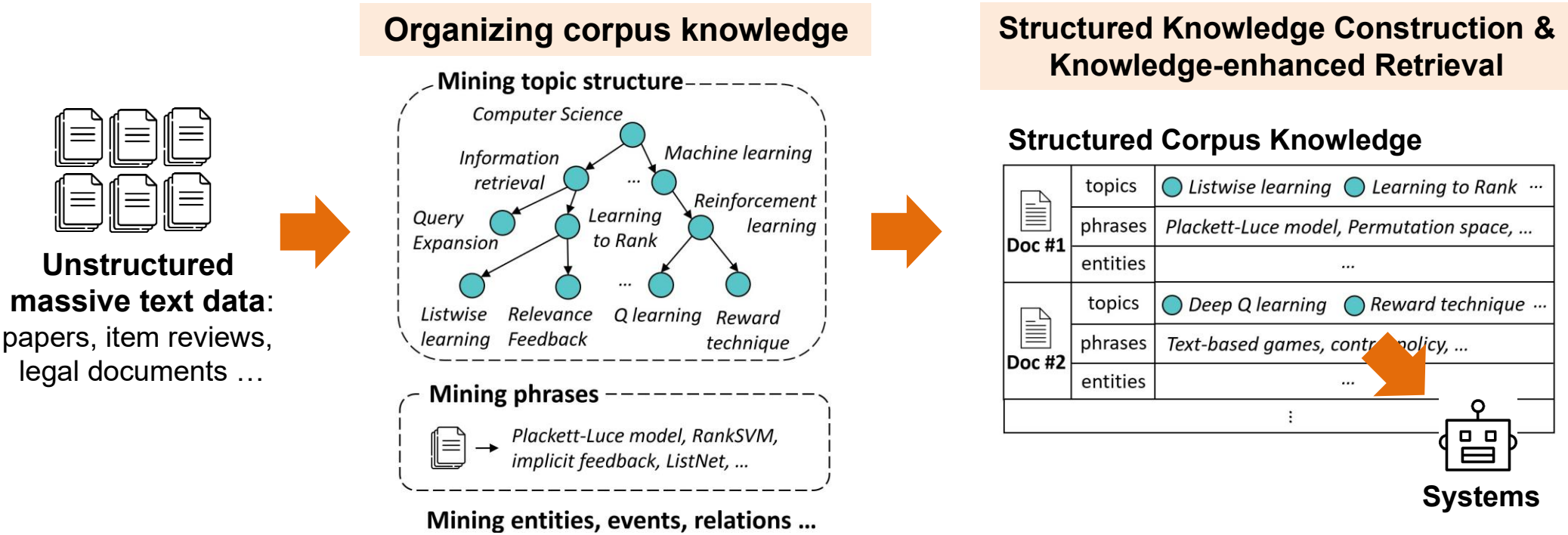


Outline

- Retrieval Augment Generation (RAG): A Brief introduction
- Retrieval and Generation Architecture for RAG
- Enhancing Information Retrieval with Corpus Knowledge and Semantic Structures
- Retrieval Improvement: Query/Pseudo-document generation and reranking
- Advanced RAG Methods: Active-RAG, Self-RAG, GraphRAG, HippoRAG, G-Retriever
- RAG by Exploring Semantic Structures: Hypercube, Structure RAG
- Exploring RAG Applications: MolE-RAG
- Open Challenges and Future Directions

Corpus Knowledge Guided Information Retrieval

- **Goal:** Adapt retrieval systems to specialized domains with minimal human efforts



- Corpus knowledge is important for theme-specific applications, where no sufficient annotation is available for fully training a retriever
- Example of corpus knowledge:
  - Topic information
  - Document graph
  - Theme-specific taxonomy
  - Knowledge graph
  - Pseudo-relevance feedback

Improving Retrieval Using a Corpus Topical Taxonomy

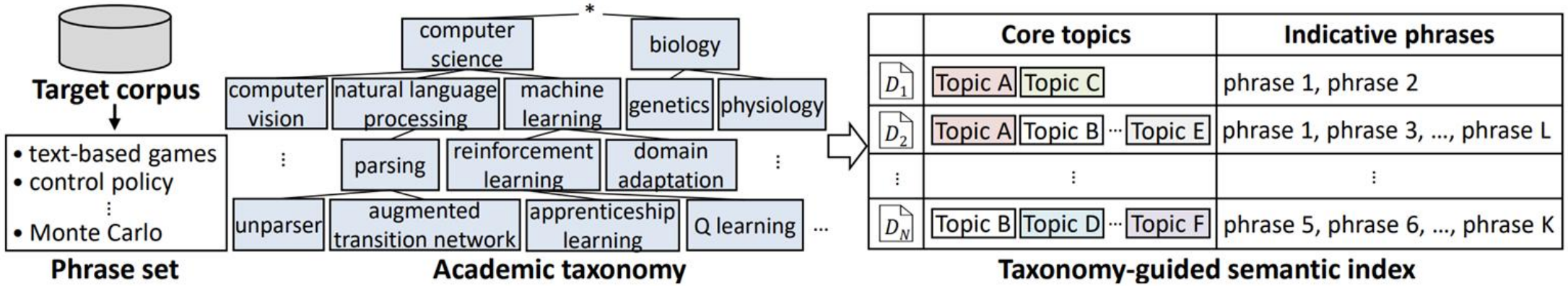
- Retrieval in *theme-specific applications* poses three challenges
  - Spanning specialized terminology and niche content
  - Limited contexts of user query
  - Specialized user interests and search intents
- A **topical taxonomy** may outline the latent topic structure and reflect user-interested aspects
- **ToTER (Topical Taxonomy Enhanced Retrieval)** identifies the central topics of queries and docs with the guidance of the taxonomy, and exploits their topical relatedness to supplement missing contexts
- As a plug-and-play framework, ToTER can be flexibly employed to enhance various PLM-based retrievers

SeongKu Kanget al., "Improving Retrieval in Theme-specific Applications using a Corpus Topical Taxonomy", WWW'24

(a) Academic domain		(b) Product domain	
Query	Provable data possession at untrusted stores	Query	#1 black natural hair dye without ammonia or peroxide
Document A (label: relevant rank: top-173)	Pors: **proofs of retrievability** for large files. In this paper, we define and explore proofs of retrievability (PORs). ... A POR may be viewed as a kind of **cryptographic proof** of knowledge (POK). ... We view PORs as an important tool for **semi-trusted** online archives. Existing cryptographic techniques help users **ensure the privacy and integrity** of files they retrieve. ...	Document A (label: relevant rank: top-70)	ONC NATURALCOLORS (1N **Black**) 4 fl. oz. (120 mL). Healthier permanent **hair dye** with certified organic ingredients, **ammonia free**, vegan friendly, 100 gray coverage. ... Cruelty-free and vegan. It is time to make the clean choice.
		Document B (label: irrelevant rank: top-11)	Roux Fanci-full Rinse 16 Hidden Honey. Tones and enhances gray and blonde hair. Rinses in and shampoos out. **No ammonia or peroxide**. ... 15 applications per bottle, temporary **hair color**, 15 ounce bottle.
ToTER rank: top-10	Topic classes: cryptography, trusted computing, digital content, computer network, computer security, computer science	ToTER rank: top-5 (Doc.A) top-32 (Doc.B)	Topic classes: hair color, hair coloring products, hair care, beauty & personal care
	Core phrases: encryption, access control, security, key, server		Core phrases: dye, permanent, lasting, permanent hair color, ammonia free

Taxonomy-Guided Semantic Index

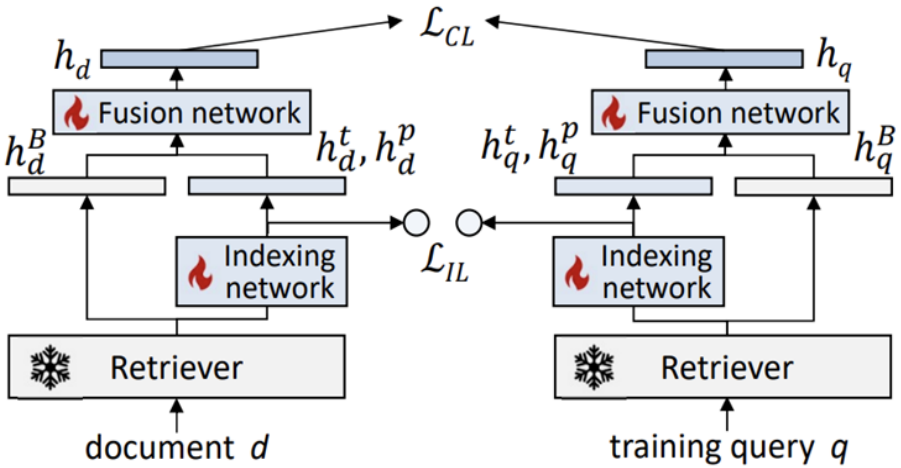
- For each document, construct two-levels of semantic index
  - **Topic-level**: Use tree-based similarity search to identify a set of candidate topics and then use LLM to identify a set of core topics for each document
  - **Phrase-level**: Use discriminative phrase mining to identify a set of phrases represents the fine-grained concepts in the documents



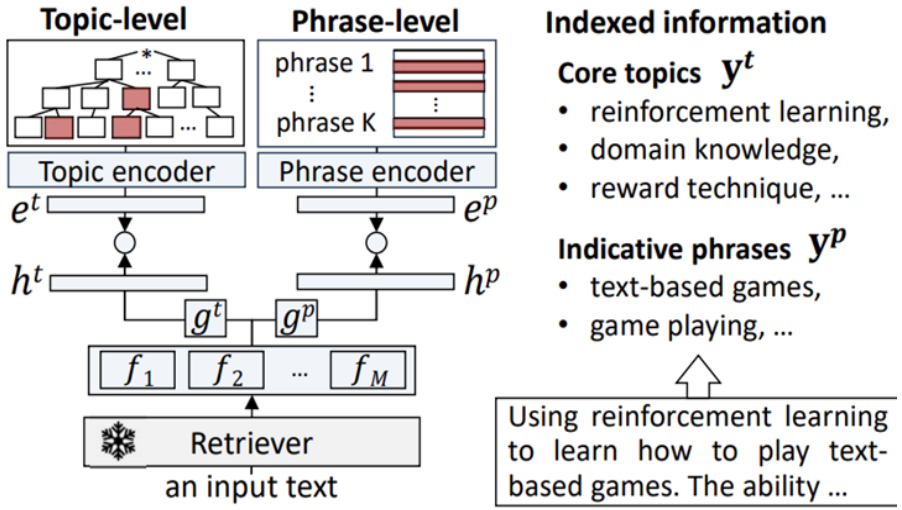
S. Kang, et al, "Taxonomy-guided Semantic Indexing for Academic Paper Search", EMNLP'24

Index-Grounded Fine-Tuning and Retrieval

- Index Network: trains the indexing network to identify core topics and indicative phrases from the text
  - Efficient inference for query
- The semantic indexing is combined with a base retriever as an add-on and fine-tuned with standard contrastive loss
- Topic-aware negative mining: utilize both topical and lexical overlaps to select negative documents

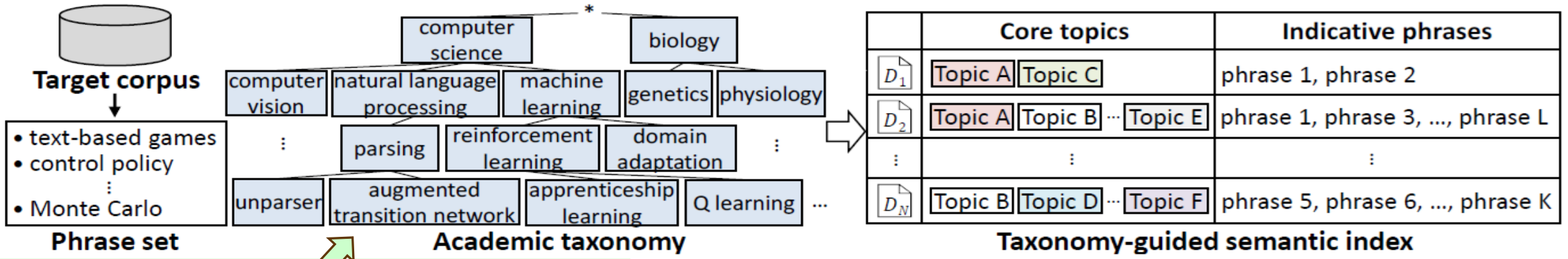


(a) Index-grounded fine-tuning



(b) Index learning with the indexing network

Taxonomy-guided Semantic Indexing for Science Info Retrieval



Taxonomy enhancement & semantic index construction

Semantic indexing outperforms dense retrieval and BM25

[Query] " Learning to win by reading manuals in a Monte-Carlo framework "

infer

*manuals, **domain knowledge**, decision making, **Monte Carlo**, Markov Chain Monte Carlo, sample selection, **text-based games**, control, **reinforcement learning**, **natural language interface** ....*

Papers in the corpus

[Paper A (irrelevant)] "A challenge dataset for the open-domain machine comprehension of text. We present MCTest, a freely available set of stories and associated questions intended for research on the machine comprehension of text. Previous work on ... solving a more restricted goal ..."

[Paper B (relevant)] "Using reinforcement learning to learn how to play text-based games. The ability to learn optimal control policies in systems where action space is defined by sentences in natural language ... optimisation of dialogue systems. Text-based games with multiple endings ..."

TSI Indexed information

topic level	*benchmark dataset crowdsourcing comprehension approach question answering textual question natural language inference **natural language processing** **artificial intelligence** ...*
phrase level	*machine comprehension reading comprehension benchmark multiple choice open domain question question answering crowdsourcing screening workers relation extraction ...*
topic level	***reinforcement learning** representation learning reward technique **natural language interface** **domain knowledge** Q learning **Monte Carlo** **natural language processing** **artificial intelligence***
phrase level	***reinforcement learning** **text-based games** learning environment text representation **game playing** **feedback** **decision** bootstrap **Monte Carlo** **control policy** dialogue systems ...*

overall textual similarity

Dense retriever (⇨): Paper A (Top-1), Paper B (Top-89)

+ common academic concepts

with TSI (⇨): Paper A (Top-69), Paper B (Top-2)

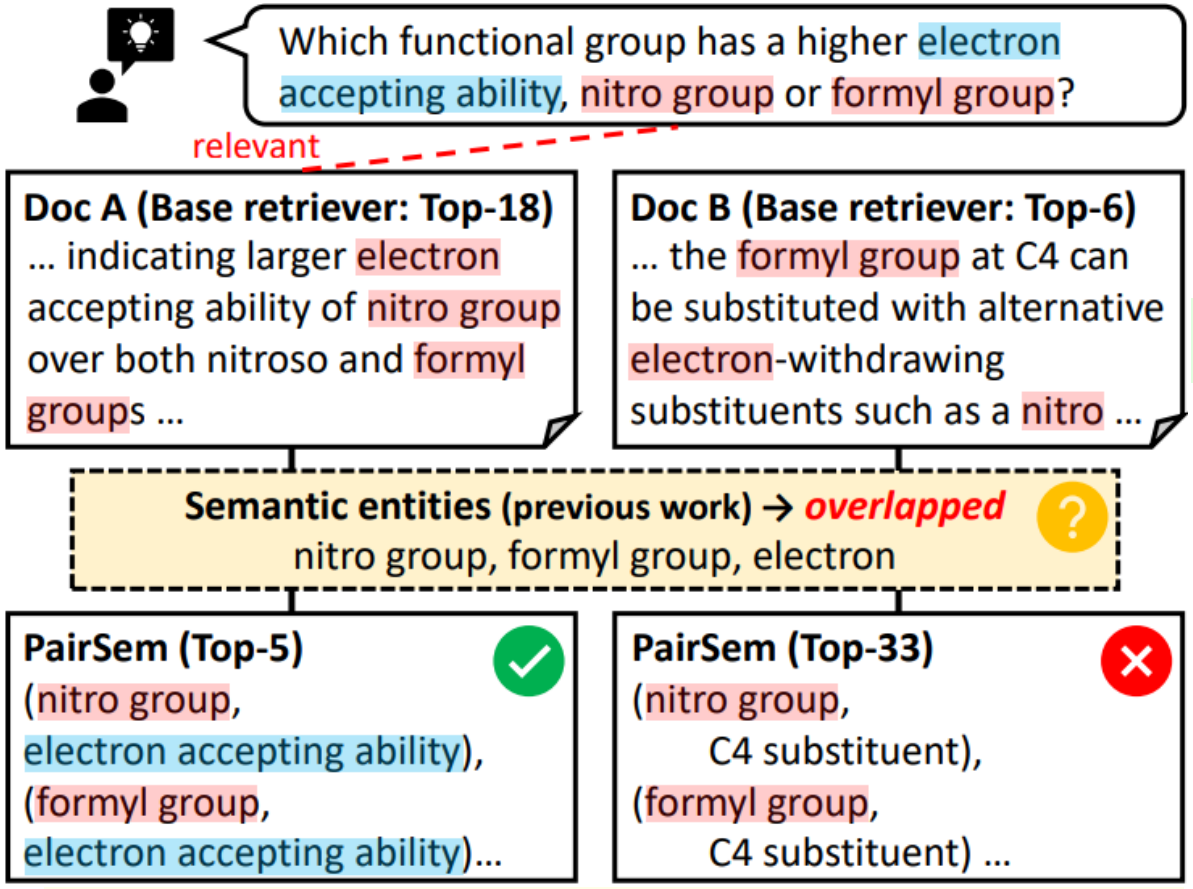
Retrieval Performance with Taxonomy-guided Semantic Indexing

		CSFCube						DORIS-MAE					
		N@5	N@10	M@5	M@10	R@50	R@100	N@5	N@10	M@5	M@10	R@50	R@100
	BM25	0.307	0.310	0.088	0.134	0.504	0.635	0.354	0.330	0.079	0.107	0.490	0.669
SPECTER-v2	no Fine-Tuning	0.352	0.337	0.108	0.151	0.524	0.680	0.385	0.360	0.079	0.113	0.551	0.709
	FFT	0.372	0.368	0.123	0.169	0.576	0.692	0.408	0.387	0.084	0.122	0.562	0.736
	aFT	0.378	0.344	0.119	0.160	0.578	0.696	0.400	0.372	0.080	0.115	0.558	0.714
	FFT w/ GRF	0.331	0.317	0.112	0.152	0.561	0.705	0.400	0.379	0.087	0.123	**0.586**	**0.756**
	FFT w/ ToTER	0.406	0.375	0.135	0.179	0.591	0.710	0.423	0.394	0.091	0.128	0.563	0.736
	JTR	0.379	0.352	0.118	0.157	0.598	0.699	0.395	0.380	0.080	0.118	0.548	0.713
	TaxoIndex	0.458†*	0.417†*	0.144†*	0.198†*	**0.633**†*	0.741†*	0.447†*	0.421†*	0.104†*	0.144†*	0.578†	**0.756**†
	TaxoIndex ++	**0.469**†*	**0.426**†*	**0.158**†*	**0.209**†*	0.621†*	**0.746**†*	**0.449**†*	**0.424**†*	**0.105**†*	**0.145**†*	0.581†	0.751†
Contriever-MS	no Fine-Tuning	0.340	0.311	0.095	0.130	0.551	0.682	0.427	0.398	0.084	0.124	0.507	0.635
	FFT	0.364	0.328	0.110	0.149	0.589	0.705	0.453	0.407	0.093	0.132	0.510	0.652
	aFT	0.346	0.328	0.104	0.151	0.578	0.711	0.439	0.402	0.089	0.128	0.507	0.648
	FFT w/ GRF	0.353	0.313	0.108	0.136	0.559	0.669	0.447	0.381	0.090	0.120	0.511	0.643
	FFT w/ ToTER	0.375	0.353	0.121	0.169	**0.597**	0.724	0.458	**0.423**	0.097	0.139	0.539	0.703
	JTR	0.351	0.331	0.105	0.152	0.578	0.714	0.435	0.408	0.089	0.132	0.516	0.667
	TaxoIndex	0.400†*	**0.386**†*	0.135†*	0.184†*	0.596†	**0.726**†	0.461	**0.423**†	0.100	0.141†	0.557†*	0.729†*
	TaxoIndex ++	**0.421**†*	**0.386**†*	**0.144**†*	**0.185**†*	0.595	**0.726**†	**0.463**	0.422†	**0.101**	**0.142**†	**0.560**†*	**0.733**†*

S. Kang, et al, "Taxonomy-guided Semantic Indexing for Academic Paper Search", EMNLP'24

PairSem: LLM-Guided Pairwise Semantic Matching for Scientific Document Retrieval

- PairSem (Pairwise Semantic Matching): Representing relevant knowledge as entity–aspect pairs, capturing complex, multi-faceted relationships



Example: Retrieval in Chemistry domain (ChemLit-QA): PairSem captures the semantic information of entity-aspect pairs

- **Offline preprocessing**: Generate corpus-grounded pairwise semantics for documents in corpus, and
- **Online inference**: generate pairs for ad-hoc queries and leverages semantic matching to enhance retrieval

Zero-shot pair generation with prompting

P_{pair} = "Given a scientific document, extract all scientific entities. Then, for each entity, find all associated aspects and generate (entity, aspect) pairs"

P_{format} = "Output only (entity, aspect) pairs using the following XML structure for each pair:<pair> <entity>entity name</entity><aspect>aspect phrase </aspect></pair>"

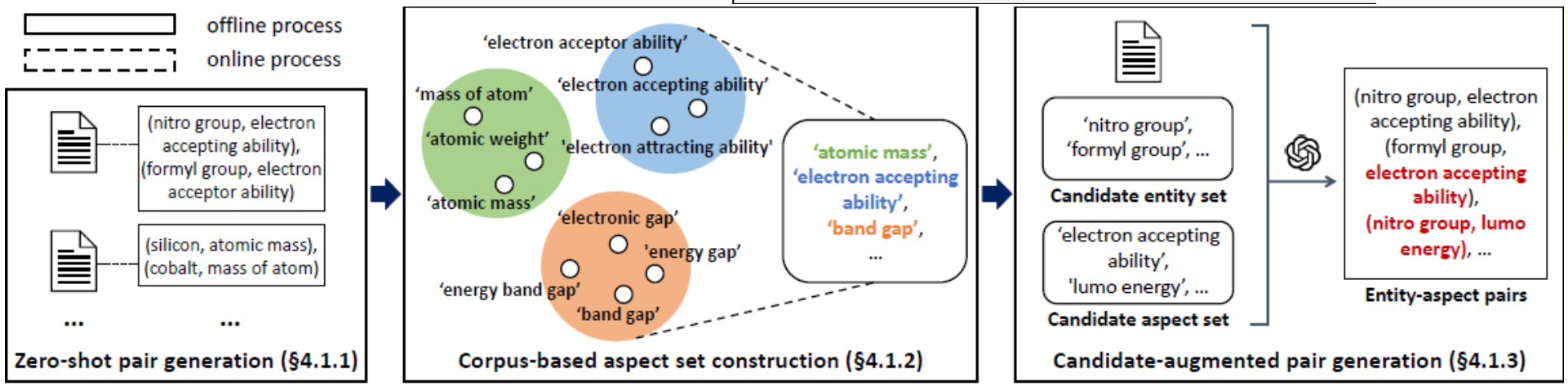
Then we perform *synonym resolution* and *corpus-based entity/aspect set construction*

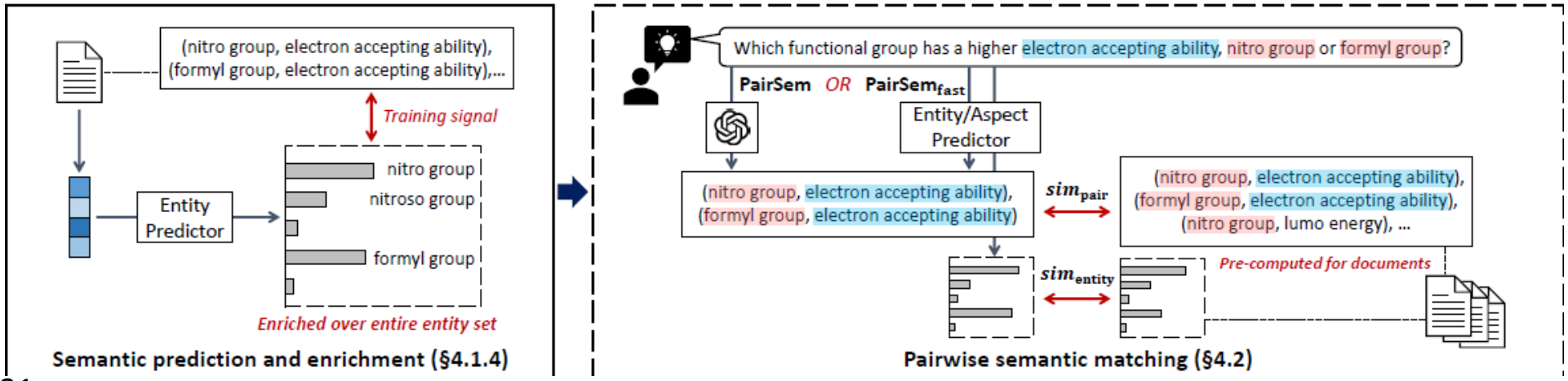
PairSem: Pairwise Semantic Matching Framework

P_{cluster} = "Given a list of scientific entities, find clusters of synonyms that describe the same academic concept. Then, for each cluster, generate a representative entity."

P'_{format} = "Output clusters and representatives using the following XML structure for each cluster: <cluster><entities>entity1, entity2,...</entities> <representative>representative entity </representative></cluster>"

P'_{pair} ="Given a scientific document, find all relevant entities from $\{\mathcal{E}_d^{\text{cand}}\}$. Then, for each entity, find all associated aspects from $\{\mathcal{A}_d^{\text{cand}}\}$, and generate relevant (entity, aspect) pairs"





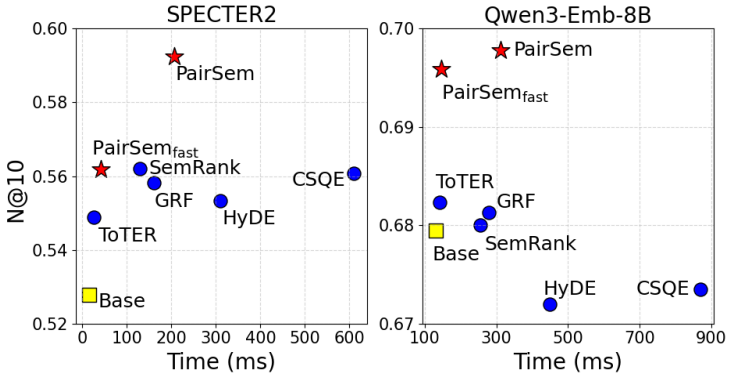
Offline: ***candidate-augmented pair generation*** and ***entity prediction and enrichment***

Online: Upon query, we conduct pairwise semantic matching for retrieval:
(1) query semantic generation with learned relevance (fast);
(2) enhancing retrieval with pairwise semantic matching

Wonbin Kweon, et al., "PairSem: LLM-Guided Pairwise Semantic Matching for Scientific Document Retrieval", WWW'2016

PairSem: Performance Study

	Method	ChemLit-QA				SciFact				LitSearch			
		N@10	N@20	R@20	R@50	N@10	N@20	R@20	R@50	N@10	N@20	R@20	R@50
SPECTER2	Base	0.5279	0.5968	0.6923	0.8132	0.6616	0.6701	0.8382	0.9090	0.3335	0.3520	0.5567	0.6547
	BERT-QE [76]	0.5263	0.6089	0.6942	0.8149	0.6594	0.6724	0.8510	0.9119	0.3276	0.3487	0.5454	0.6486
	LADR [26]	0.5311	0.6103	0.7064	0.8212	0.6656	0.6834	0.8580	0.9203	0.3408	0.3671	0.5723	0.6641
	ToTER [21]	0.5488	0.6114	0.7190	0.8310	0.6642	0.6815	0.8581	0.9147	0.3401	0.3662	0.5719	0.6655
	PairSem$_{fast}$	**0.5618***	**0.6343***	**0.7398***	**0.8538***	**0.6846***	**0.7010***	**0.8844***	**0.9330***	**0.3579***	**0.3808***	**0.5861***	**0.6869***
	HyDE [13]	0.5534	0.6081	0.7289	0.8319	0.6776	0.6842	0.8556	0.9107	0.3548	0.3767	0.6025	0.6954
	GRF [32]	0.5581	0.6162	0.7330	0.8439	0.6894	0.6961	0.8677	0.9221	0.3512	0.3729	0.5921	0.6913
	CSQE [29]	0.5608	0.6121	0.7303	0.8322	0.6713	0.6881	0.8607	0.9164	0.3496	0.3718	0.5868	0.6901
	SemRank [72]	0.5620	0.6318	0.7479	0.8459	0.6910	0.7001	0.8682	0.9250	0.3657	0.3804	0.6324	0.7017
	PairSem	**0.5925***	**0.6656***	**0.7667***	**0.8684***	**0.7100***	**0.7282***	**0.9056***	**0.9403***	**0.3845***	**0.4037***	**0.6474***	**0.7229***
Contriever-MS	Base	0.6279	0.6891	0.7431	0.8344	0.6768	0.6898	0.8619	0.9120	0.3694	0.3784	0.5753	0.6627
	BERT-QE [76]	0.6131	0.6877	0.7415	0.8349	0.6702	0.6839	0.8576	0.9114	0.3681	0.3799	0.5768	0.6631
	LADR [26]	0.6334	0.6921	0.7490	0.8411	0.6783	0.6921	0.8661	0.9205	0.3755	0.3923	0.5906	0.6649
	ToTER [21]	0.6319	0.6916	0.7475	0.8408	0.6770	0.6904	0.8638	0.9137	0.3805	0.3980	0.5964	0.6678
	PairSem$_{fast}$	**0.6438***	**0.7086***	**0.7641***	**0.8552***	**0.6922***	**0.7060***	**0.8824***	**0.9377***	**0.3990***	**0.4189***	**0.6224***	**0.6889***
	HyDE [13]	0.6304	0.6987	0.7481	0.8432	0.6818	0.6942	0.8705	0.9154	0.3714	0.3826	0.5801	0.6718
	GRF [32]	0.6342	0.6976	0.7558	0.8457	0.6974	0.7011	0.8731	0.9201	0.3846	0.3941	0.5986	0.6811
	CSQE [29]	0.6217	0.6837	0.7441	0.8365	0.6834	0.6981	0.8636	0.9141	0.3853	0.3935	0.5864	0.6723
	SemRank [72]	0.6335	0.7014	0.7646	0.8424	0.6931	0.7096	0.8655	0.9187	0.3854	0.4067	0.6057	0.6807
	PairSem	**0.6501***	**0.7121***	**0.7756***	**0.8612***	**0.7112***	**0.7239***	**0.8859***	**0.9427***	**0.4139***	**0.4314***	**0.6241***	**0.6973***
Qwen3-Embedding-8B	Base	0.6795	0.7491	0.8384	0.9149	0.7695	0.7759	0.9300	0.9567	0.5802	0.5982	0.7916	0.8400
	BERT-QE [76]	0.6613	0.7338	0.8227	0.9103	0.7538	0.7681	0.9257	0.9495	0.5811	0.5995	0.7939	0.8445
	LADR [26]	0.6801	0.7536	0.8411	0.9170	0.7717	0.7812	0.9373	0.9603	0.5906	0.6042	0.7989	0.8532
	ToTER [21]	0.6823	0.7560	0.8476	0.9205	0.7809	0.7875	0.9417	0.9583	0.5948	0.6078	0.8022	0.8577
	PairSem$_{fast}$	**0.6959***	**0.7663***	**0.8599***	**0.9297**	**0.7900***	**0.7959**	**0.9467**	**0.9667***	**0.6125***	**0.6263***	**0.8103**	**0.8634**
	HyDE [13]	0.6720	0.7471	0.8367	0.9201	0.7584	0.7647	0.9261	0.9445	0.5728	0.5961	0.7814	0.8338
	GRF [32]	0.6813	0.7502	0.8438	0.9161	0.7728	0.7816	0.9354	0.9526	0.5854	0.6009	0.7946	0.8501
	CSQE [29]	0.6735	0.7428	0.8341	0.9113	0.7615	0.7728	0.9312	0.9570	0.5791	0.5977	0.7897	0.8395
	SemRank [72]	0.6800	0.7494	0.8446	0.9171	0.7824	0.7884	0.9407	0.9601	0.5866	0.6018	0.7935	0.8518
	PairSem	**0.6978***	**0.7661***	**0.8560***	**0.9280***	**0.7906***	**0.7970***	**0.9433**	**0.9700***	**0.6113***	**0.6273***	**0.8184***	**0.8751***



Time-accuracy trade-off on

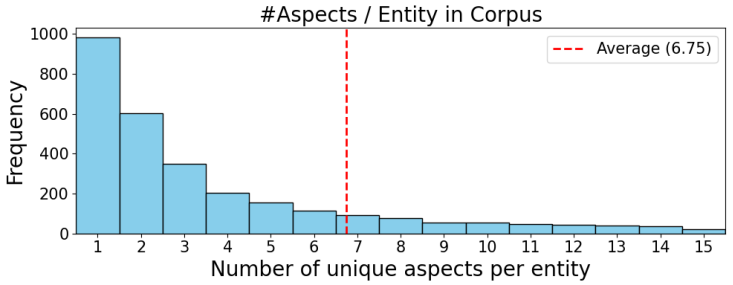


Table 4: Results with biomedical retrievers on SciFact

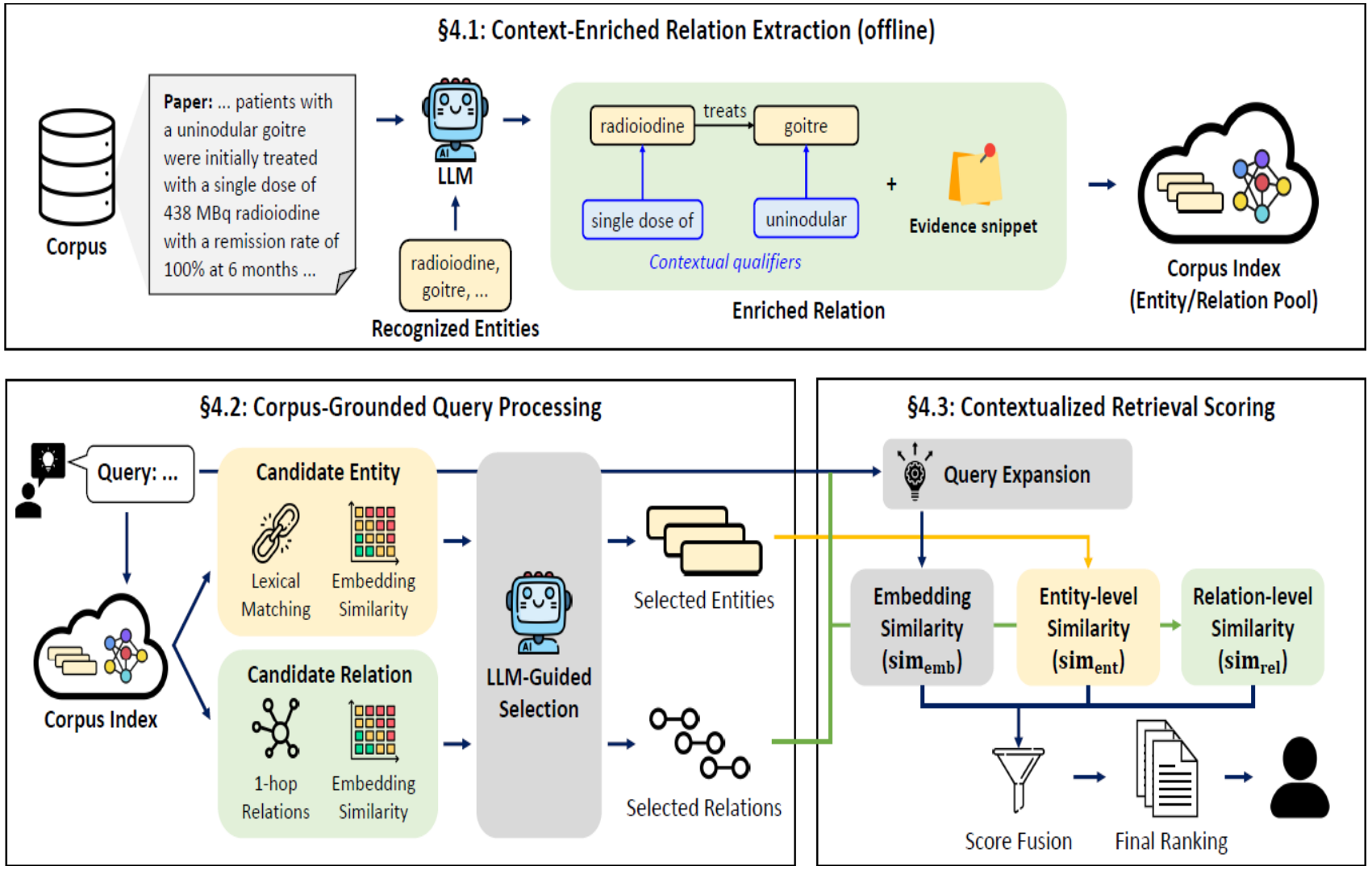
Model	N@10	N@20	N@50	N@100
MedCPT	0.7246	0.7417	0.7507	0.7531
+ **PairSem**	**0.7473**	**0.7619**	**0.7665**	**0.7690**
BMRetriever-1B	0.7573	0.7671	0.7718	0.7742
+ **PairSem**	**0.7673**	**0.7794**	**0.7890**	**0.7909**

PairSem enhances the retrieval performance of MedCPT & BMRetriever-1B

ContextRank: Enhancing Scientific Document Retrieval with Context-Enriched Relations

- ContextRank: A plug-and-play framework that extracts context enriched relations by attaching contextual qualifiers to each relational slot, encoding fine grained scientific knowledge
- It extracts context-enriched relations and integrates 3 complementary relevance signals: embedding-level, entity-level, and relation level similarities

Three key steps: (1) Context-enriched relation extraction (offline); (2) corpus-grounded query processing; and (3) Contextualized retrieval scoring



Wonbin Kweon, Chih-Hsuan Wei, Xueqiang Xu, Runchu Tian, Pengcheng Jiang, Zhizheng Wang, Seongku Kang, Zhiyong Lu, Jiawei Han, "ContextRank: Enhancing Scientific Document Retrieval with Context-Enriched Relations", arXiv:

ContextRank: Performance Study

ContextRank outperforms other methods for info retrieval on three science datasets

Method	SciFact				ChemLit-QA				LitSearch			
	N@10	N@20	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20	R@10	R@20
Base retriever: SPECTER2												
Base	0.6616	0.6701	0.8029	0.8382	0.5279	0.5968	0.5435	0.6923	0.3335	0.3520	0.4856	0.5567
BERT-QE	0.6594	0.6724	0.8234	0.8510	0.5263	0.6089	0.5513	0.6942	0.3276	0.3487	0.4851	0.5454
HyDE	0.6891	0.6919	0.8330	0.8607	0.5569	0.6175	0.5634	0.7331	0.3620	0.3679	0.5018	0.6057
CSQE	0.6944	0.6936	0.8394	0.8721	0.5729	0.6247	0.5810	0.7471	0.3782	0.3974	0.5204	0.6133
GRF	0.6860	0.6920	0.8288	0.8669	0.5609	0.6194	0.5747	0.7380	0.3671	0.3836	0.5115	0.6063
ToTER	0.6911	0.6962	0.8371	0.8756	0.5614	0.6213	0.5756	0.7424	0.3660	0.3785	0.5152	0.5982
MixGR	0.7054	0.7126	0.8404	0.8814	0.5718	0.6261	0.5895	0.7483	0.3728	0.3895	0.5312	0.6373
SemRank	0.7083	0.7165	0.8386	0.8894	0.5759	0.6321	0.6081	0.7548	0.3744	0.3940	0.5350	0.6418
HippoRAG2	0.7122	0.7331	0.8576	0.9090	0.5840	0.6638	0.6197	0.7689	0.3870	0.4061	0.5455	0.6482
PairSem	0.7113	0.7302	0.8571	0.9069	0.5942	0.6694	0.6242	0.7751	0.3913	0.4127	0.5506	0.6509
ContextRank	**0.7447***	**0.7516***	**0.8852***	**0.9209***	**0.6231***	**0.6922***	**0.6681***	**0.8057***	**0.4240***	**0.4444***	**0.5701***	**0.6681***

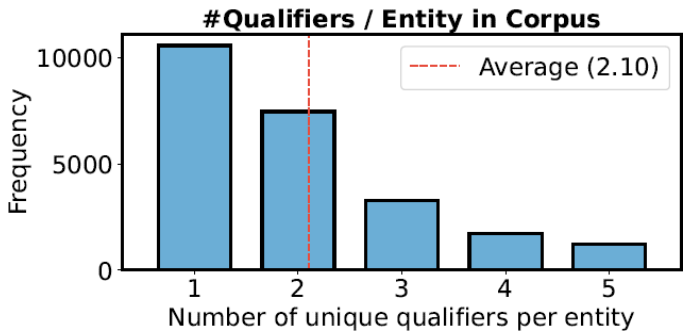
Data Statistics for Experiments

Dataset	#documents	#queries	doc/query
SciFact	5,183	300	1.13
ChemLit-QA	1,528	1,025	6.00
LitSearch	64,183	597	1.07

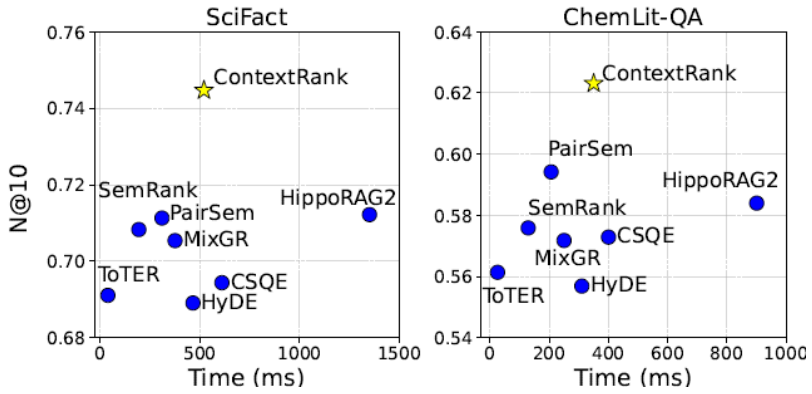
ContextRank plugs-in and boosts retrieval performance for every LM

Base Retriever	SciFact				ChemLit-QA				LitSearch			
	N@5	N@10	R@5	R@10	N@5	N@10	R@5	R@10	N@5	N@10	R@5	R@10
E5-Mistral-7B	0.7499	0.7757	0.8208	0.8939	0.5610	0.6192	0.4263	0.6283	0.5138	0.5433	0.6037	0.6945
+ CONTEXTRANK	**0.7632**	**0.7834**	**0.8458**	**0.9022**	**0.6563**	**0.7154**	**0.5034**	**0.7230**	**0.5577**	**0.5812**	**0.6671**	**0.7289**
GritLM-7B	0.7548	0.7733	0.8297	0.9049	0.5892	0.6473	0.4477	0.6546	0.5625	0.5870	0.6640	0.7371
+ CONTEXTRANK	**0.7774**	**0.8016**	**0.8551**	**0.9256**	**0.6556**	**0.7206**	**0.5045**	**0.7345**	**0.5862**	**0.6036**	**0.6857**	**0.7580**
Qwen3-Embedding-8B	0.7338	0.7695	0.8199	0.9055	0.6263	0.6795	0.4822	0.6883	0.5606	0.5802	0.6425	0.7176
+ CONTEXTRANK	**0.7662**	**0.7854**	**0.8519**	**0.9172**	**0.6755**	**0.7420**	**0.5231**	**0.7587**	**0.5700**	**0.5929**	**0.6567**	**0.7251**

ContextRank: More Performance + Case Study



unique contextual qualifiers per entity in corpus of SciFact dataset



Time-accuracy trade-off with SPECTER2

Base Retriever	SciFact			
	N@5	N@10	R@5	R@10
MedCPT	0.6919	0.7246	0.7866	0.8611
+ CONTEXTRANK	**0.7391**	**0.7569**	**0.8294**	**0.8772**
BMRetriever-1B	0.7272	0.7573	0.8065	0.8692
+ CONTEXTRANK	**0.7467**	**0.7678**	**0.8221**	**0.8820**

Boosting biomedical retrievers on SciFact

Detailed case study on SciFact dataset with SPECTER2

For HippoRAG2, we highlight the overlapped relation in Grey

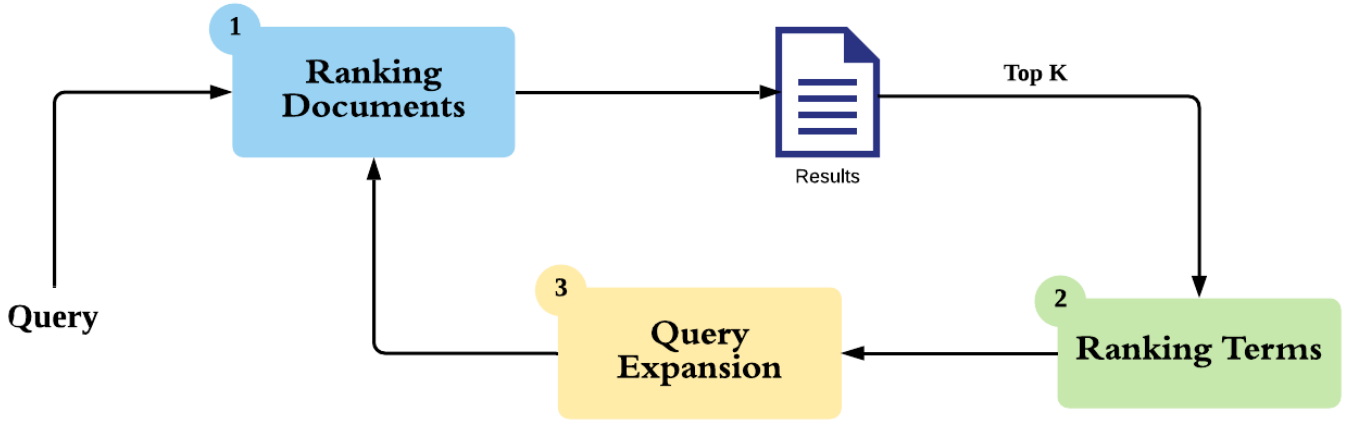
For ContextRank, blue texts are the extracted qualifiers, and relevant qualifiers are highlighted in blue

Query	Radioiodine treatment of non-toxic multinodular goitre reduces thyroid volume.		
	Selected relations (by ContextRank): (radioiodine:two doses of, treats:by reducing thyroid volume, goitre:non-toxic multinodular)		
Doc A (*relevant*)	... Patients treated with two doses as well as those developing hyperthyroidism had a significant reduction in thyroid volume ... makes the use of **radioiodine** an attractive alternative to surgery in selected cases of non-toxic multinodular **goitre** ...	Doc B (*irrelevant*)	... We studied 201 consecutive patients who received a relatively fixed dose of **radioiodine** ... Patients with a uninodular **goitre** were initially treated with a single dose of 438 MBq **radioiodine** with a remission rate of 100% at 6 months ...
HippoRAG2 rank: 8	**Extracted relation:** (radioiodine, treats, goitre), ...	**HippoRAG2** rank: 5	**Extracted relation:** (radioiodine, treats, goitre), ...
ContextRank rank: 1	**Extracted relation:** (radioiodine:**two doses of**, treats:**by reducing thyroid volume**, goitre:**non-toxic multinodular**) ...	**ContextRank** rank: 4	**Extracted relation:** (radioiodine:**single dose of**, treats:**with a remission rate of 100% at 6 months**, goitre:**uninodular**) ...

Outline

- Retrieval Augment Generation (RAG): A Brief introduction
- Retrieval and Generation Architecture for RAG
- Enhancing Information Retrieval with Corpus Knowledge and Semantic Structures
- Retrieval Improvement: Query/Pseudo-document generation and reranking 
- Advanced RAG Methods: Active-RAG, Self-RAG, GraphRAG, HippoRAG, G-Retriever
- RAG by Exploring Semantic Structures: Hypercube, Structure RAG
- Exploring RAG Applications: MolE-RAG
- Open Challenges and Future Directions

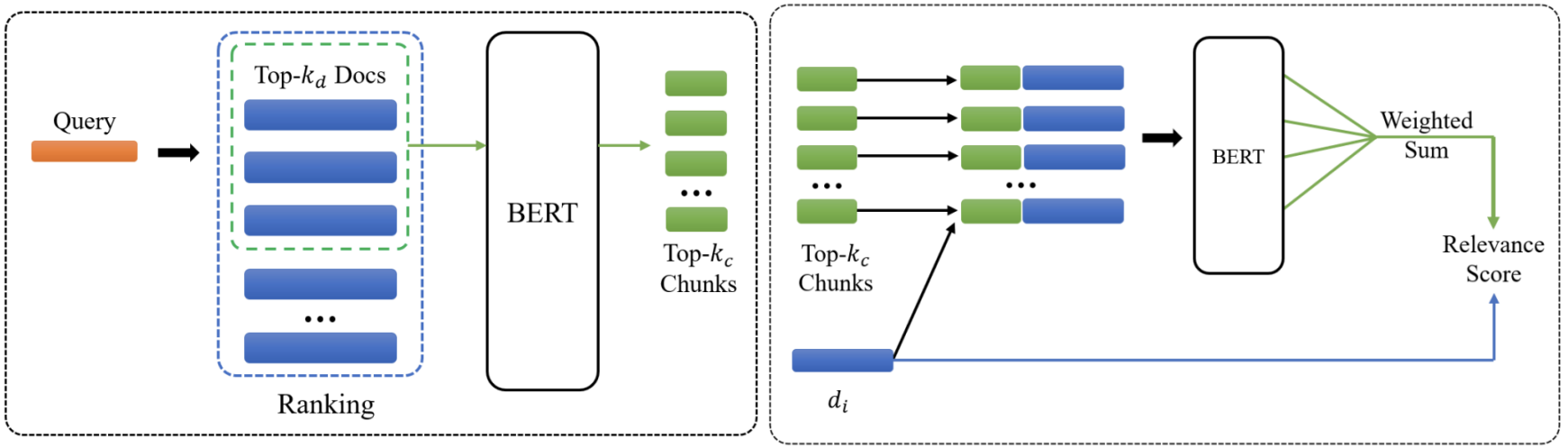
Pseudo Relevance Feedback (PRF)



- Earlier works on Pseudo Relevance Feedback (PRF) aim to enrich the initial query with frequent terms extracted from an initial ranking results (e.g., Bo1, KL, RM3)
- Explore how deep learning methods, especially language models, can help the Pseudo Relevance Feedback model in retrieval
  - Various kinds of textual signals are explored for PRF
    - E.g., text chunks, documents, …

BERT-QE: Contextualized Query Expansion

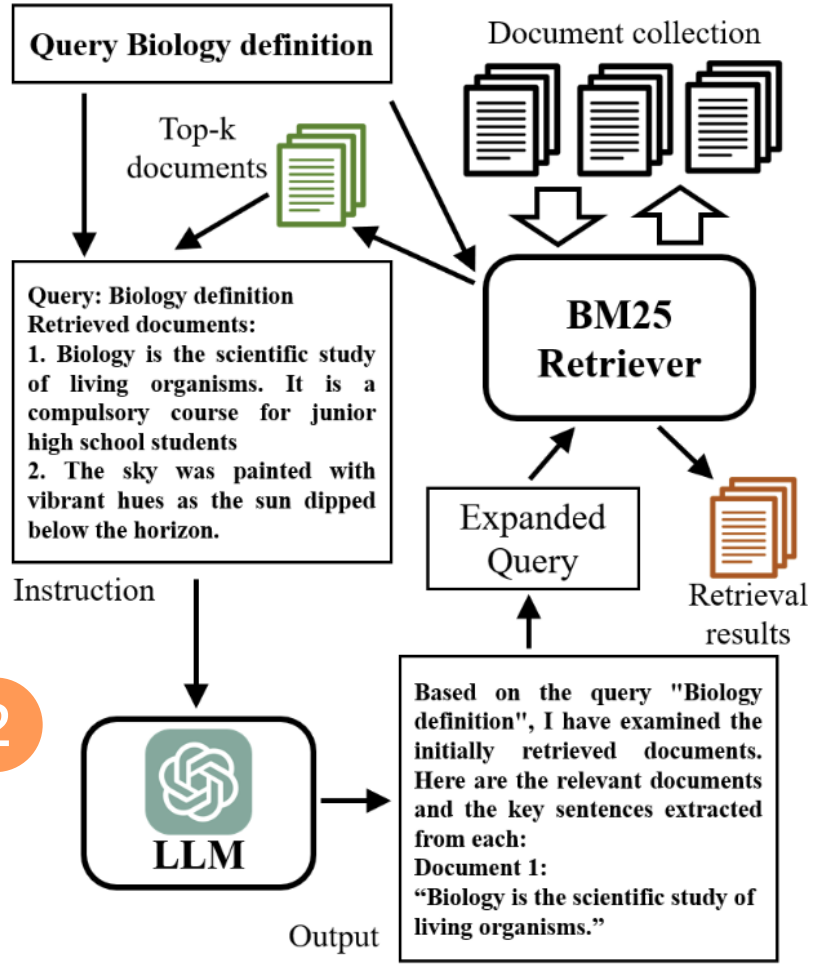
- Use a BERT-based model to select relevant text chunks from the initial retrieval results
- **Text Chunks:** Top-ranked documents are split into overlapping text chunks
- **Final ranking:** A document is matched with each selected feedback chunk, and the scores are combined using the similarity between each chunk and query as weights



Zhi Zheng, Kai Hui, Ben He, Xianpei Han, Le Sun, Andrew Yates, "BERT-QE: Contextualized Query Expansion for Document Re-ranking", EMNLP'20

CSQE: Corpus-Steered Query Expansion with LLMs

- **Main idea:** Feeding LLMs pseudo-relevant documents to generate an expanded query



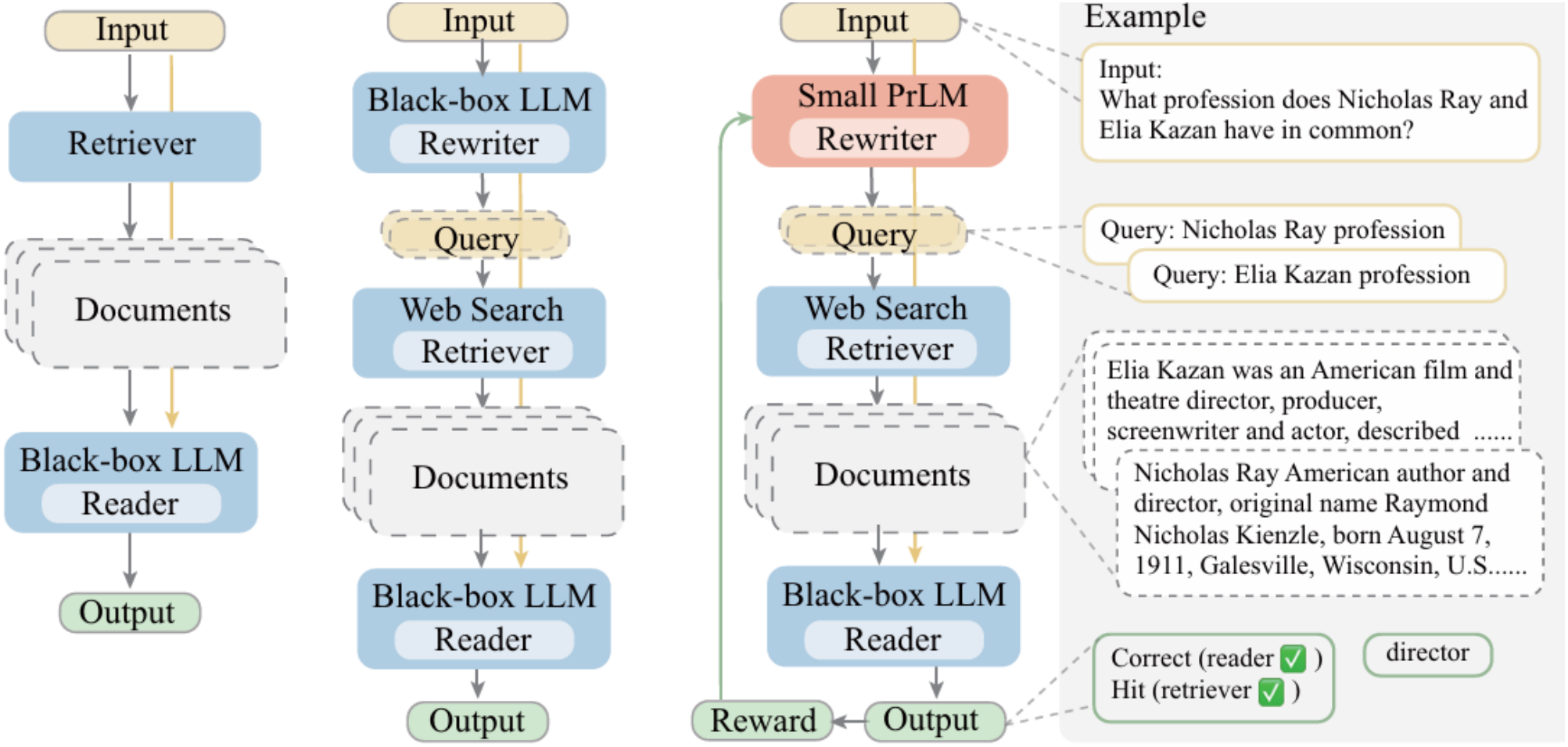
1 Using BM25 retriever, retrieve initial top-K documents (e.g., pseudo-relevance)

2 Prompt LLM to **extract key sentences** from top-k documents

3 Expand query with the extracted key sentences and retrieve the final results

Yibin Lei, Yu Cao, Tianyi Zhou, Tao Shen, Andrew Yates, "Corpus-Steered Query Expansion with Large Language Models", EACL'24

Query Rewriting for Retrieval-Augmented LLMs



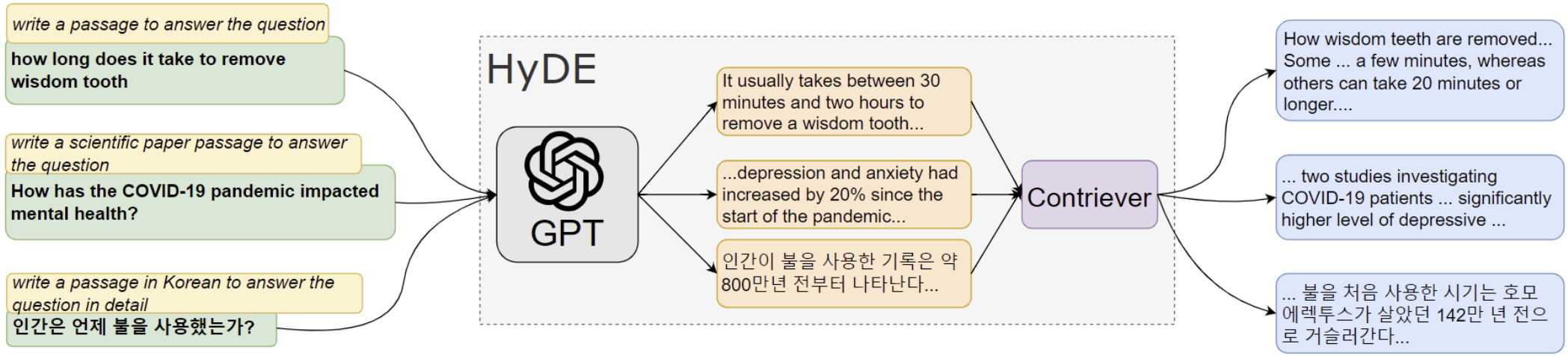
(a) Retrieve-then-read (b)Rewrite-retrieve-read (c) Trainable rewrite-retrieve-read

A trainable scheme for the Rewrite-Retrieve Read framework. A small language model is adopted as a trainable rewriter to cater to the black-box LLM reader. The rewriter is trained using the feedback of the LLM reader by reinforcement learning.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, Nan Duan, "Query Rewriting for Retrieval-Augmented Large Language Models", EMNLP'23

HyDE: Hypothetical Document Embedding

- Generating hypothetical document for a given query, and use it as expanded query

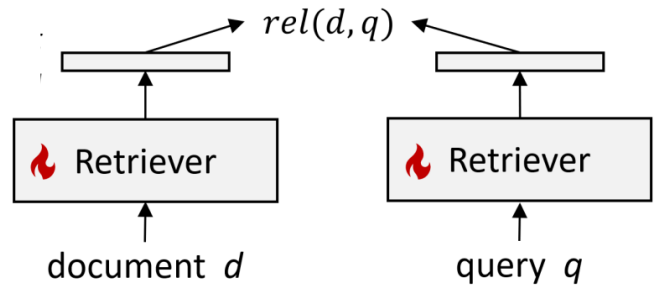


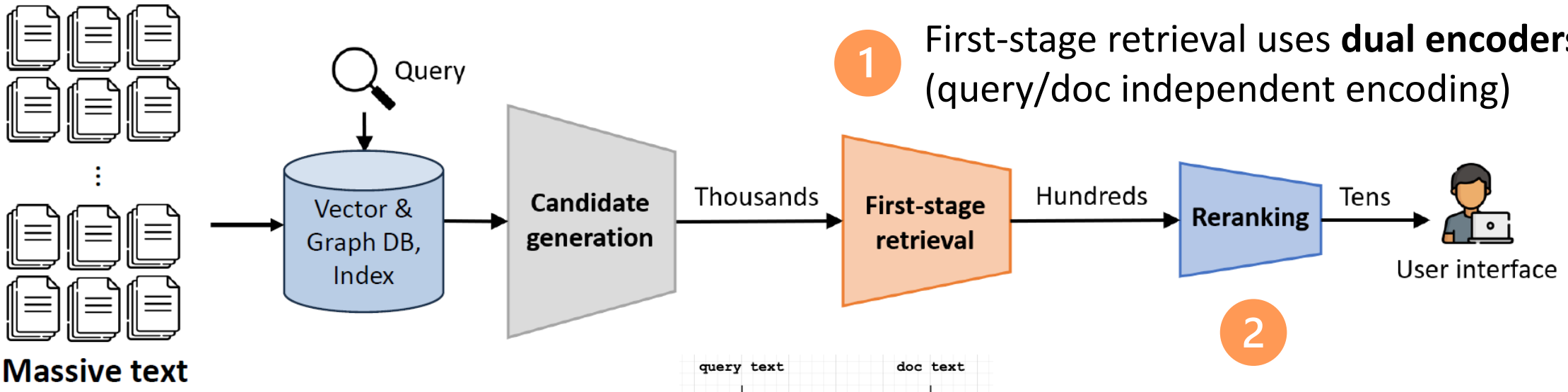
	DL19			DL20		
	mAP	nDCG@10	Recall@1k	mAP	nDCG@10	Recall@1k
Unsupervised						
BM25	30.1	50.6	75.0	28.6	48.0	78.6
Contriever	24.0	44.5	74.6	24.0	42.1	75.4
HyDE	**41.8**	**61.3**	**88.0**	**38.2**	**57.9**	**84.4**

Luyu Gao, Xueguang Ma, Jimmy Lin, Jamie Callan, “Precise Zero-Shot Dense Retrieval without Relevance Labels”, ACL’23

Reranking

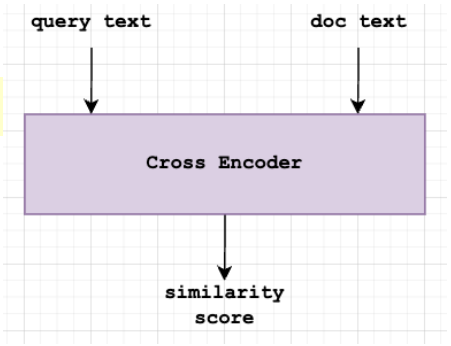
- **2-stage retrieval framework:** retrieve-then-rerank





First-stage retrieval uses **dual encoders** (query/doc independent encoding)

Figure from: https://blog.lukesalamone.com/posts/dual-encoders-ranking/

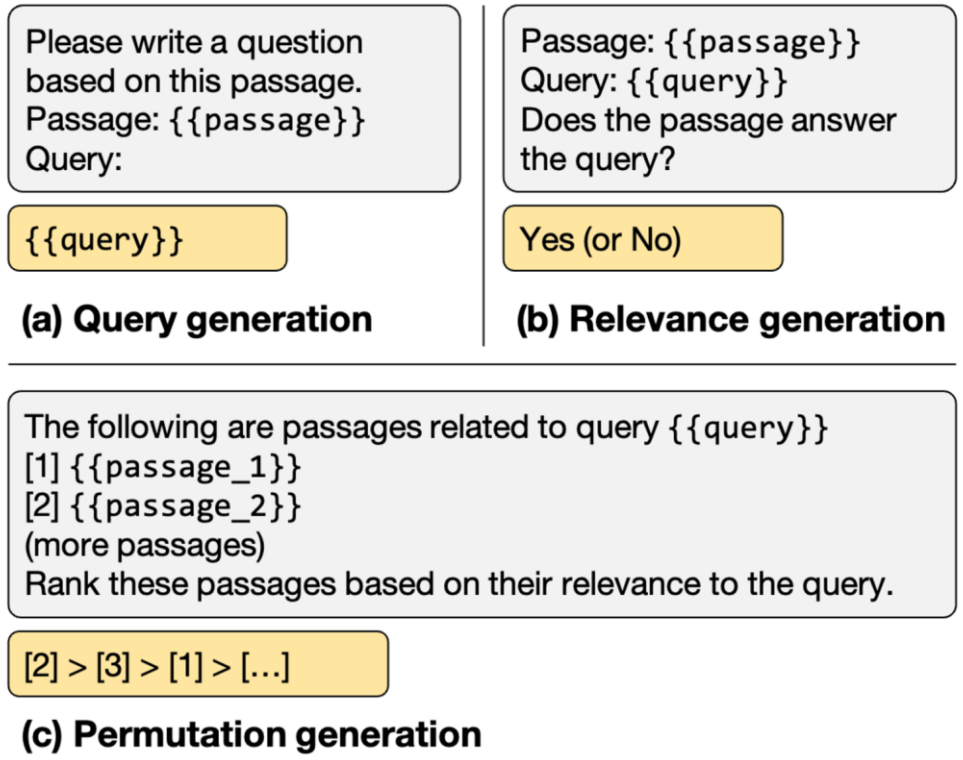


Second-stage reranking typically uses **cross-encoder** (query + initial retrived results)

LLM as Reranker

- Different strategies to re-rank passages with LLMs
  - **Query Generation:** Use the log probability of LLMs to generate the query based on the passage as the score
  - **Relevance Generation:** Instruct LLMs to output relevance judgments
  - **Permutation Generation:** Generate a ranked list of a group of passages using LLMs

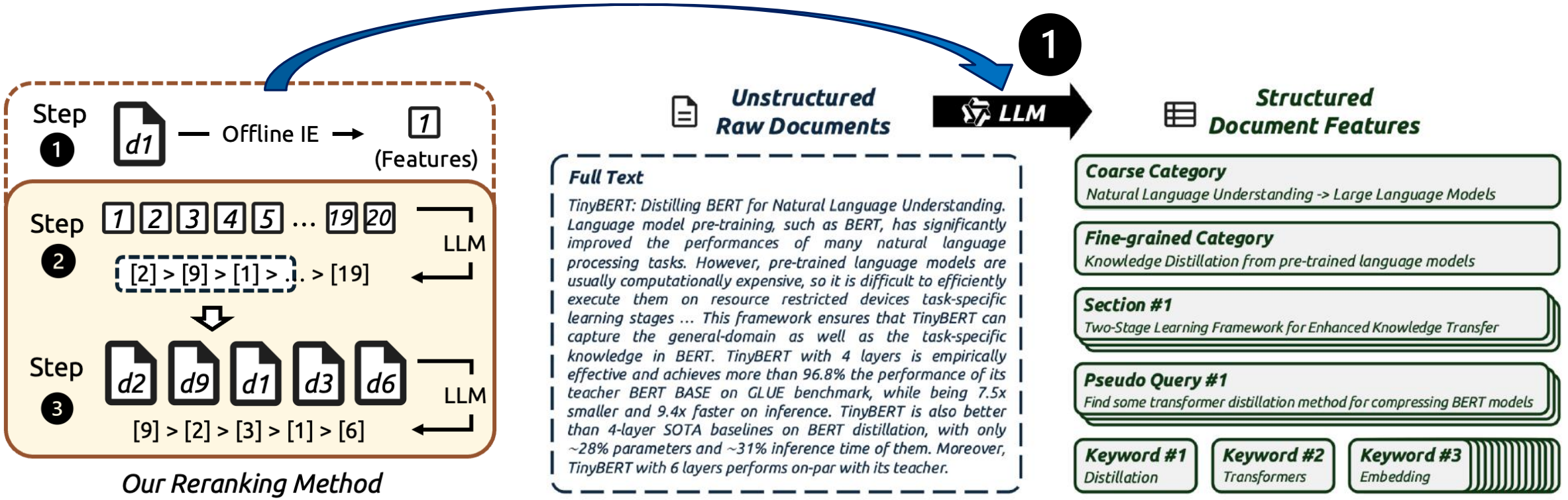
- Limitation: Context window of LLM!
  - LLMs can only compare documents that fit within their context window.
    - E.g., 2k tokens for 1 doc, 32k context length → 16 documents can be reranked



Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, Zhaochun Ren, "Is ChatGPT Good at Search? Investigating Large Language Models as Re-Ranking Agents", EMNLP'23

CoRank: LLM-Based Compact Reranking

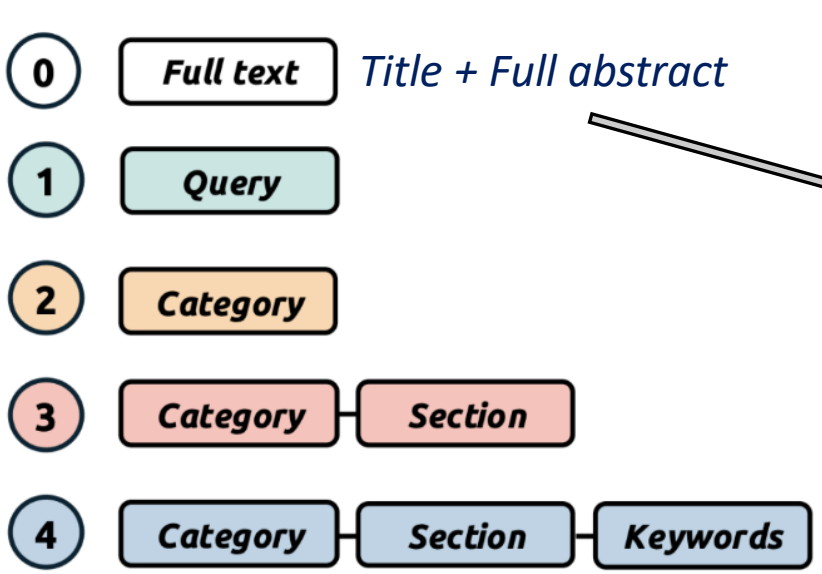
- **Problem:** Pairwise (sliding window) methods are highly inefficient
- **Main idea:** Generate compact & structured document features and use them instead of the full text

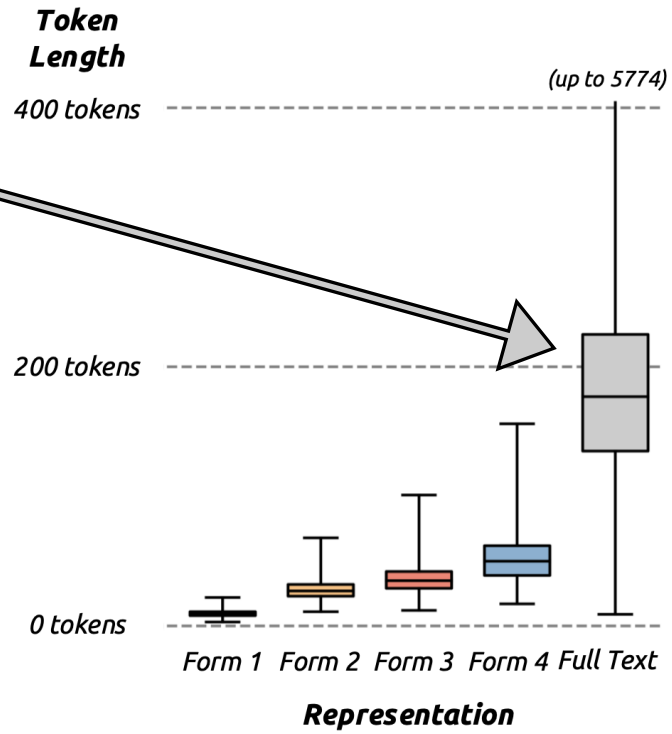


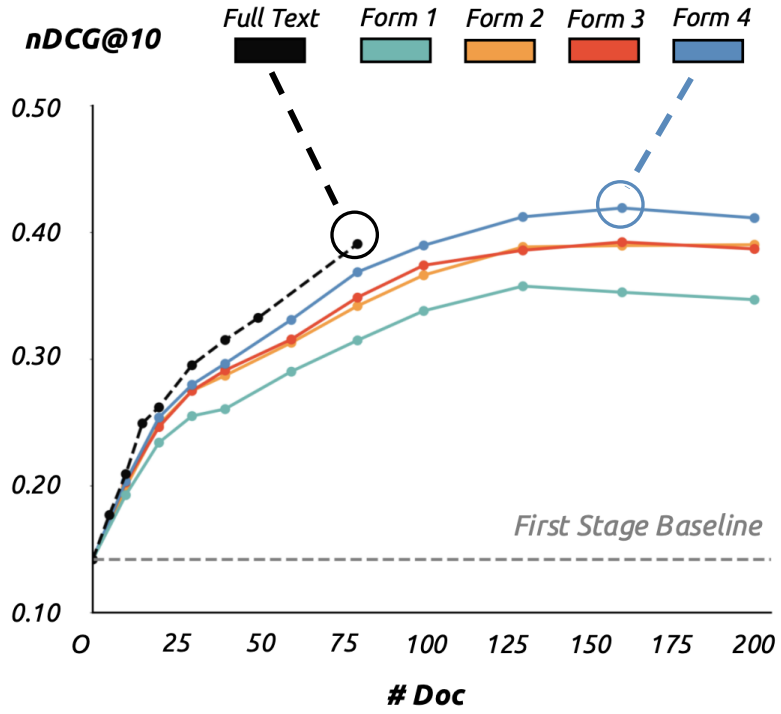
Runchu Tian, Xueqiang Xu, Bowen Jin, SeongKu Kang, Jiawei Han,
"CoRank: LLM-Based Compact Reranking with Document Features for Scientific Retrieval", KDD'26

CoRank: Performance

- **Main idea:** Generate compact document features and use them instead of the full text



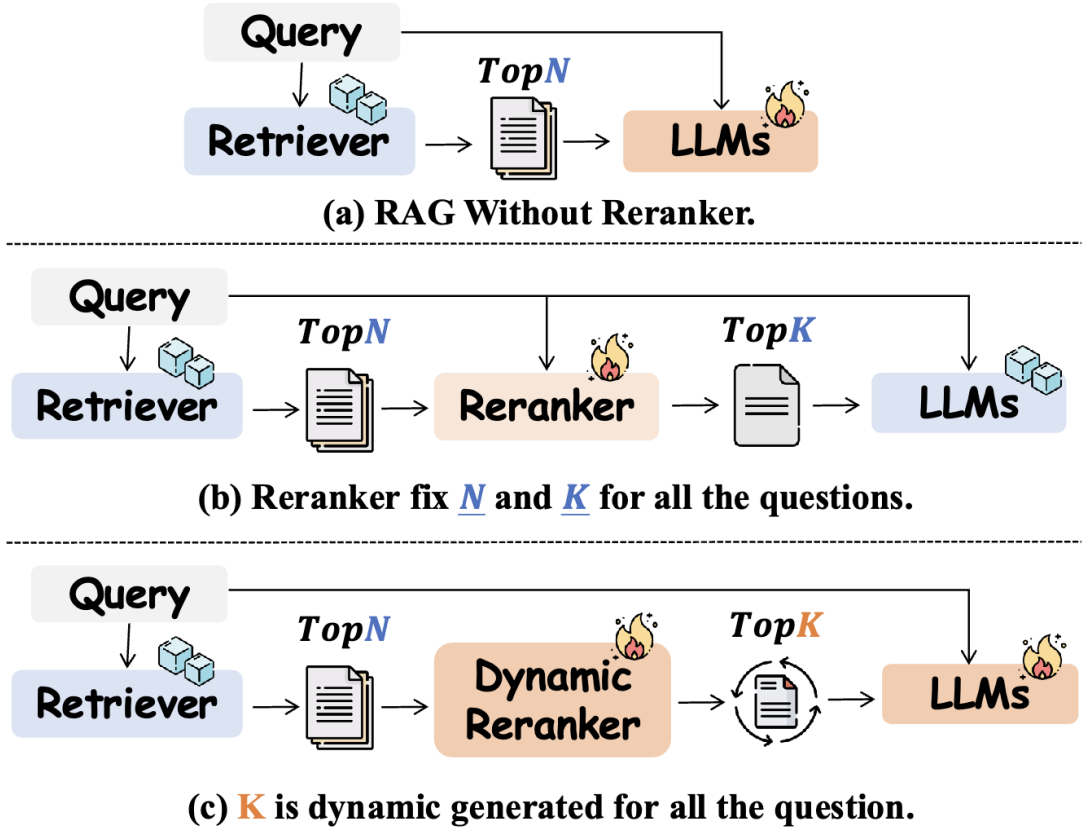




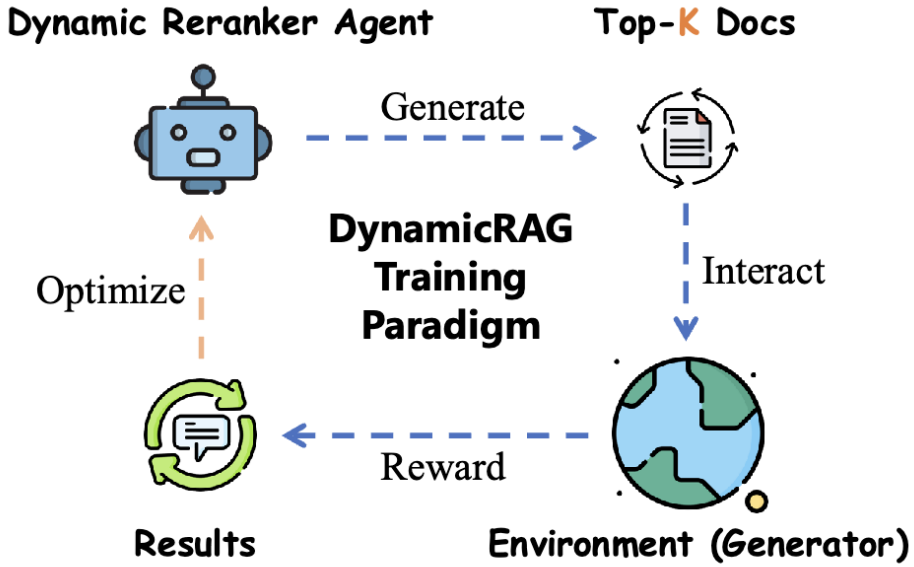
We can include more docs in the context, and get higher nDCG@10

Runchu Tian, Xueqiang Xu, Bowen Jin, SeongKu Kang, Jiawei Han,
"CoRank: LLM-Based Compact Reranking with Document Features for Scientific Retrieval", KDD'26

DynamicRAG: Leveraging Outputs of LLMs as Feedback



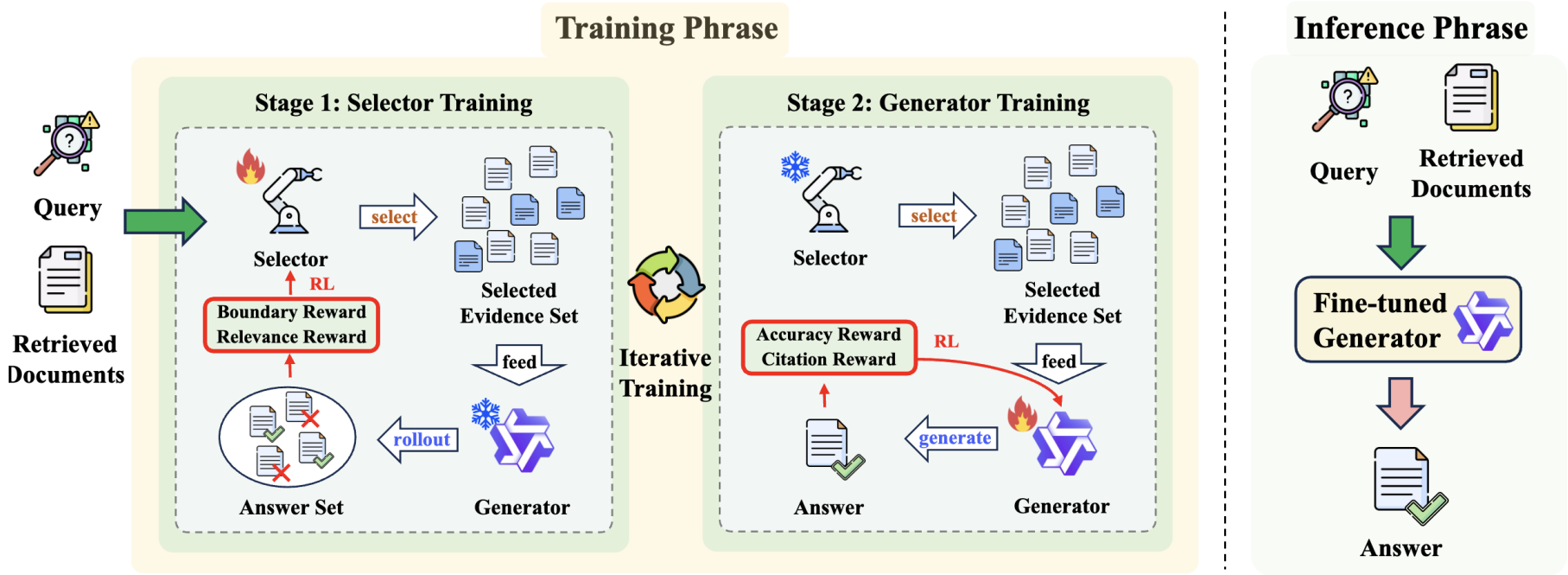
The reranker dynamically adjusts both the order and number of retrieved documents (top K) based on the query



Model the reranker as an agent optimized through reinforcement learning (RL), using rewards derived from LLM output quality

Sun, et al. "DynamicRAG: Leveraging Outputs of Large Language Model as Feedback for Dynamic Reranking in Retrieval-Augmented Generation" *NeurIPS 2025*

BAR-RAG: Boundary-Aware Evidence Selection for Robust RAG



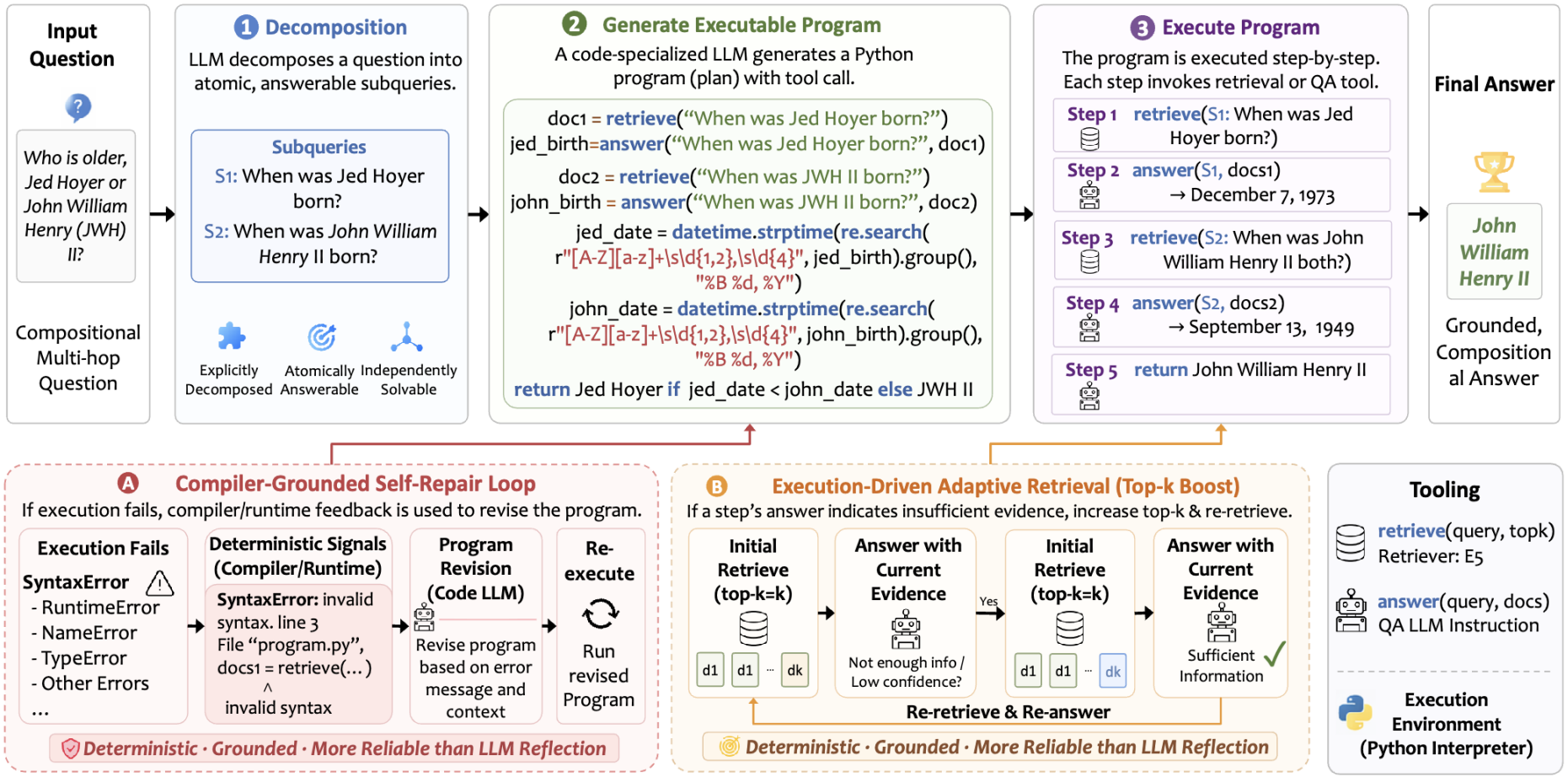
Most retrievers/rerankers optimize solely for relevance, not considering if the evidence is good for the generator

BAR-RAG: boundary-aware evidence selector that targets the generator's Goldilocks Zone: Evidence that is challenging yet sufficient for inference and thus provides the strongest learning signal.

BAR-RAG trains the selector with reinforcement learning using generator feedback and adopts a two-stage pipeline that fine-tunes the genera tor under the induced evidence distribution to mitigate the distribution mismatch between training and inference.

Sun, et al. "Rethinking the Reranker: Boundary-Aware Evidence Selection for Robust Retrieval-Augmented Generation", ICML 2026

PyRAG: Executable Multi-Hop Reasoning for RAG



- PyRAG: Reformulate multi-hop RAG as program synthesis and execution
- Represent the reasoning process as program execution over retrieval and QA tools
- Expose intermediate states as variables, producing deterministic feedback through execution
- Yield an inspectable trace of the entire reasoning process
- Enable compiler-grounded self-repair and execution-driven adaptive retrieval without any additional training

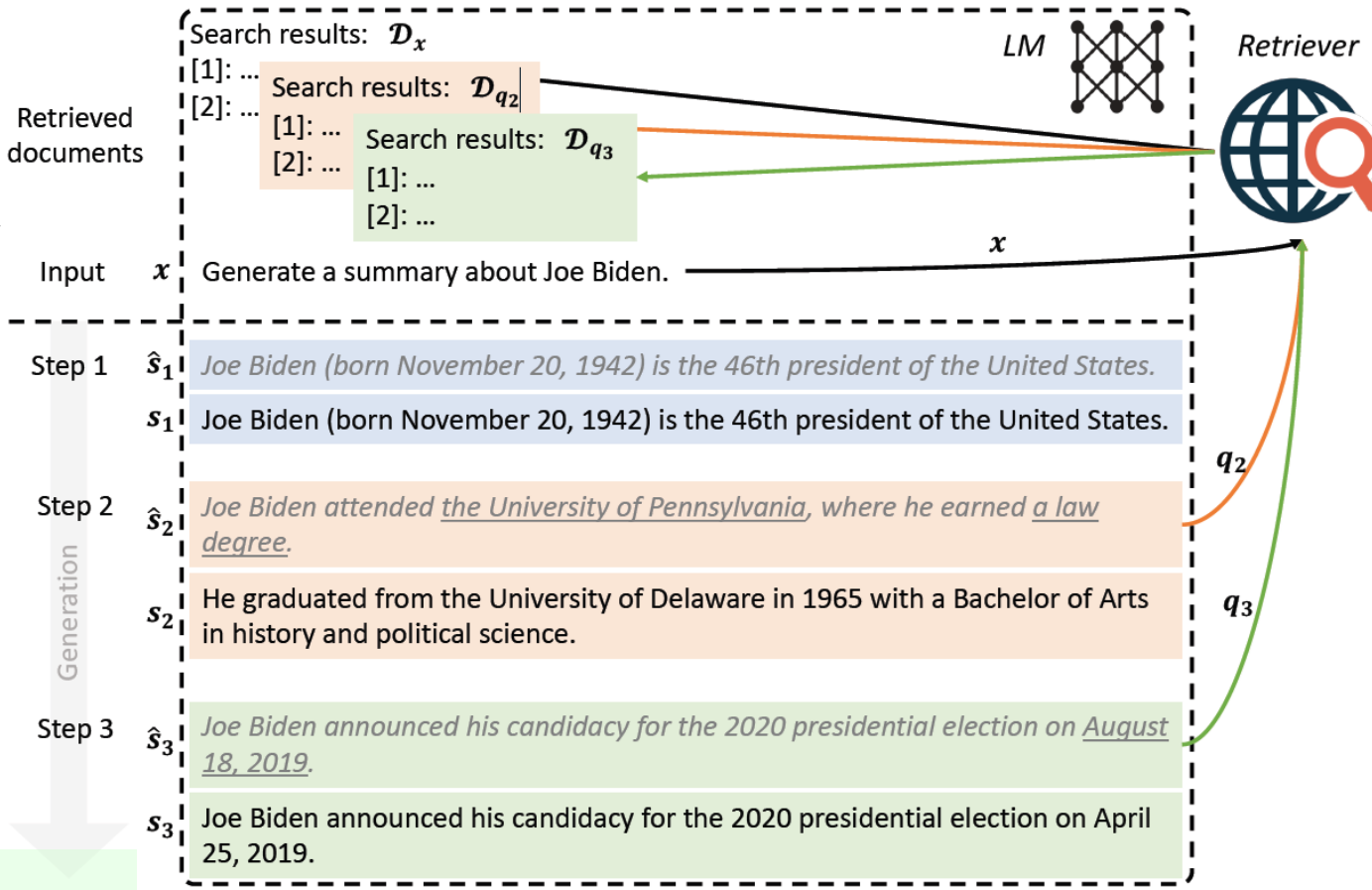
Sun, et al. "Retrieval is Cheap, Show Me the Code: Executable Multi-Hop Reasoning for Retrieval-Augmented Generation" *arXiv: 2605.12975*

Outline

- Retrieval Augment Generation (RAG): A Brief introduction
- Retrieval and Generation Architecture for RAG
- Enhancing Information Retrieval with Corpus Knowledge and Semantic Structures
- Retrieval Improvement: Query/Pseudo-document generation and reranking
- Advanced RAG Methods: Active-RAG, Self-RAG, GraphRAG, HippoRAG, G-Retriever
- RAG by Exploring Semantic Structures: Hypercube, Structure RAG
- Exploring RAG Applications: MolE-RAG
- Open Challenges and Future Directions

FLARE: Forward-Looking Active Retrieval Augmented Generation

- **FLARE (Forward-looking Active Retrieval Augmented Generation)**: Iteratively combining prediction and info retrieval
- **Most existing** retrieval augmented LMs employ a retrieve-and-generate setup that **only retrieves info once based on the input**
- **Active RAG**: **Continually gathering info throughout generation, and actively decide when and what to retrieve across the course of the generation**
- LLM actively predicts upcoming content & uses these predictions as queries to retrieve relevant info.



In FLARE, each sentence or segment generated by the LLM is evaluated for confidence. ***If the confidence level is below a certain threshold, the model uses the generated content as a query to retrieve relevant info, which is then used to regenerate or refine the sentence.*** This iterative process ensures that each part of the response is informed by the most relevant and current info available.

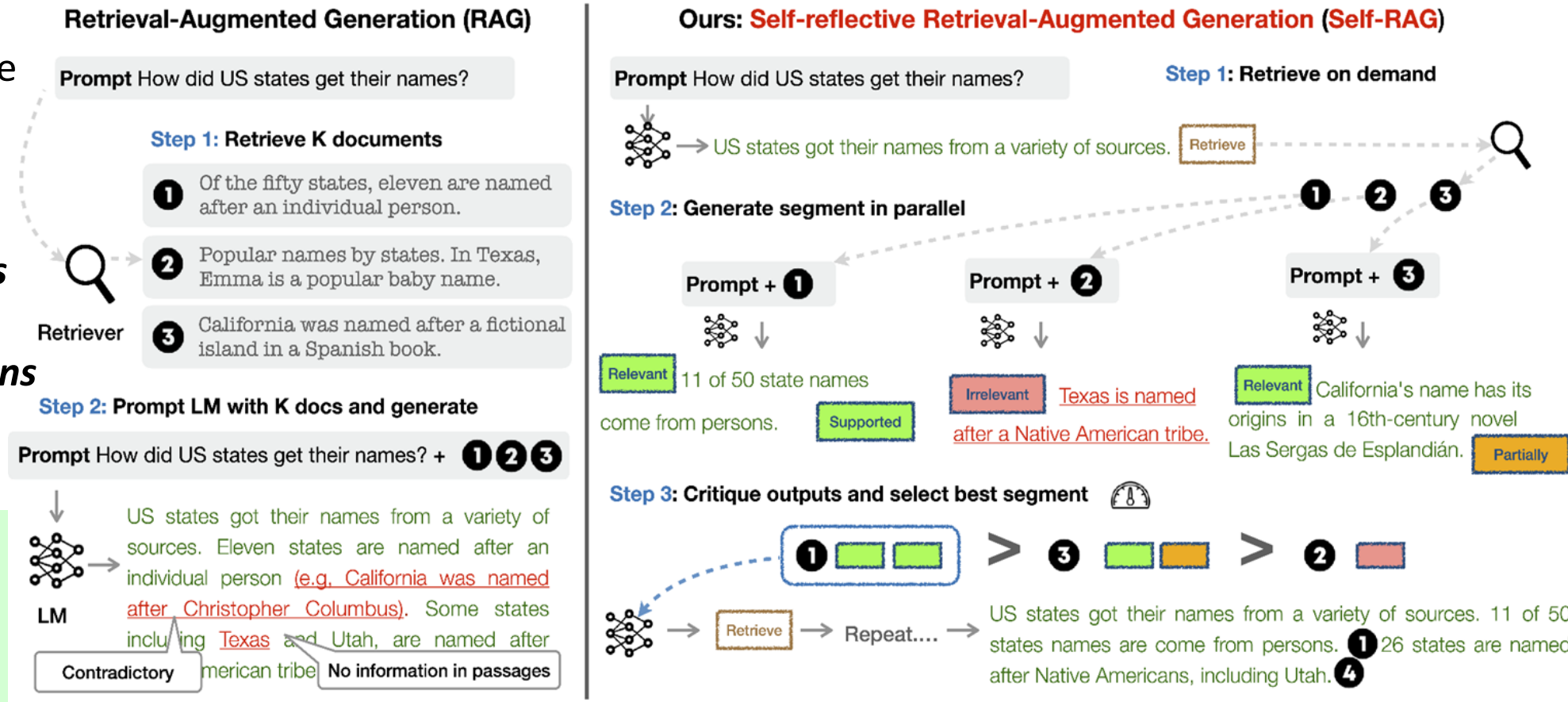
FLARE iteratively generates a temporary next sentence (shown in gray italic) & check whether it contains low-probability tokens (indicated with underline)

Z. Jiang, F. F. Xu, L. Gao, Z. Sun, Q. Liu, J. Dwivedi-Yu, Y. Yang, J. Callan, and G. Neubig, "Active retrieval augmented generation," 2023.

Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection

- **Limitation of Naive RAG:** (1) Retrieve a fixed number of passages w/o considering if retrieval is necessary or passages are relevant; (2) no guarantee that generations are entailed by cited evidence

- **SelfRAG:** Trains a single arbitrary LM that *adaptively retrieves passages on-demand*
- ***Generates*** and ***reflects on retrieved passages and its own generations using special tokens***



Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, Hannaneh Hajishirzi: "Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection", arXiv:2310.11511

SelfRAG: Reflection Tokens

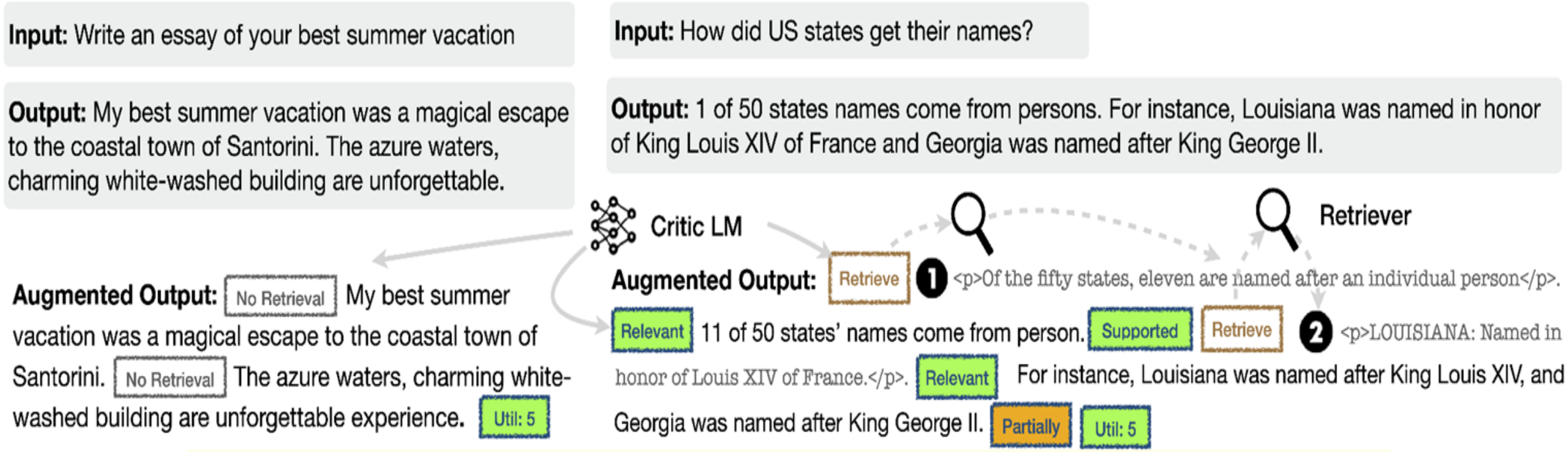
- At every segment, we can:
  - **Retrieve:** Decode a retrieval token to decide if retrieval is needed and fetch top documents if so
  - **Generate:** Without retrieval, predicts the next segment; with retrieval, critiques relevance before generating output
  - **Critique:** Evaluates if retrieved passages support generation and finalizes the response's utility

Type	Input	Output	Definitions
Retrieve	x / x, y	{yes, no, continue}	Decides when to retrieve with $\mathcal{R}$
IsRel	x, d	{**relevant**, irrelevant}	d provides useful information to solve x.
IsSup	x, d, y	{**fully supported**, partially supported, no support}	All of the verification-worthy statement in y is supported by d.
IsUse	x, y	{**5**, 4, 3, 2, 1}	y is a useful response to x.

Table 1: Four types of reflection tokens used in SELF-RAG. Each type uses several tokens to represent its output values. The bottom three rows are three types of Critique tokens, and **the bold text** indicates the most desirable critique tokens. x, y, d indicate input, output, and a relevant passage, respectively.

SelfRAG: Training

- Training
  - **Critic Training:** Augment diverse instruction-output data with retrieved passages and reflection tokens from the Retriever
  - **Generator Training:** Train the LM using next-token prediction to generate natural continuations and special tokens for retrieval or critique



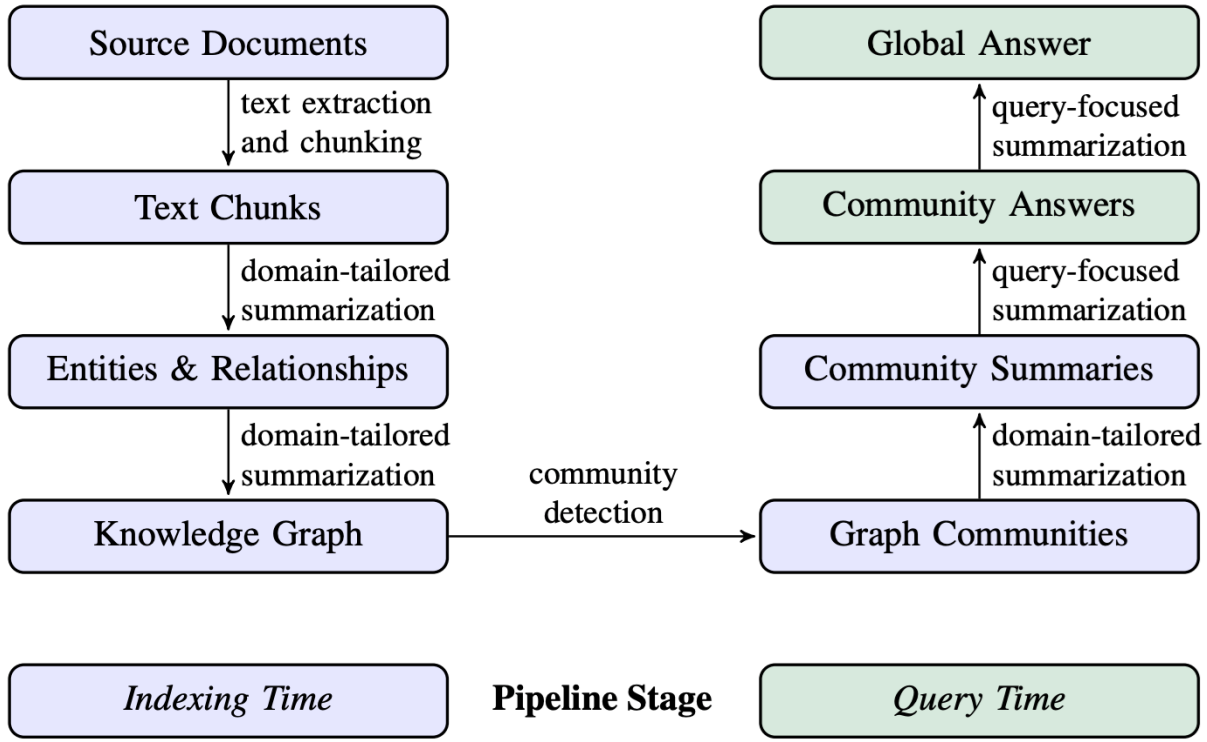
The left example does not require retrieval while the right one requires retrieval; thus, passages are inserted

GraphRAG

- **GraphRAG**
  - Use an LLM to build a graph index in two stages:
    - Derive an entity knowledge graph from the source documents
    - Pre-generate community summaries for all groups of closely related entities
- LLM-generated KG to understand entity relationships for improved retrieval
- Split graph into communities and create summaries to assist search
- Summarize themes and aggregate information for whole dataset reasoning

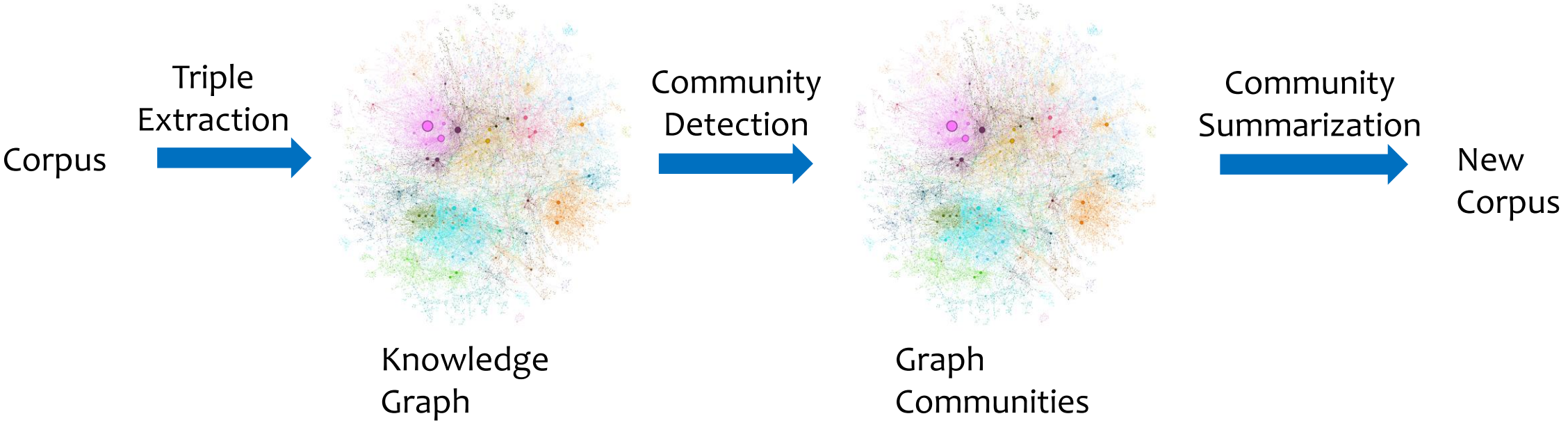
Edge, Darren, et al. "From Local to Global: A GraphRAG Approach to Query-Focused Summarization", arXiv:2404.16130

- **Limitation of RAG**
  - Struggle with complex queries that involve multiple pieces of information
  - Fail to provide all relevant context that may span the dataset



GraphRAG: Graph Community Detection and Textualization

Overview of GraphRAG



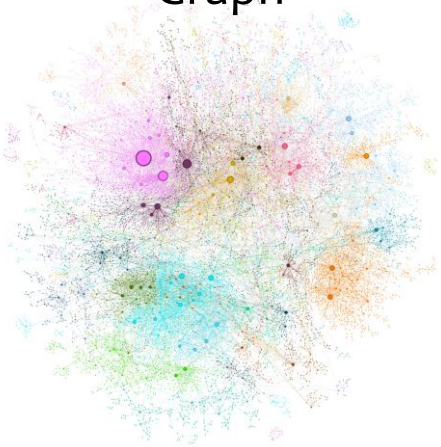
The core idea is to "reindex" the corpus into structurally meaningful text chunks

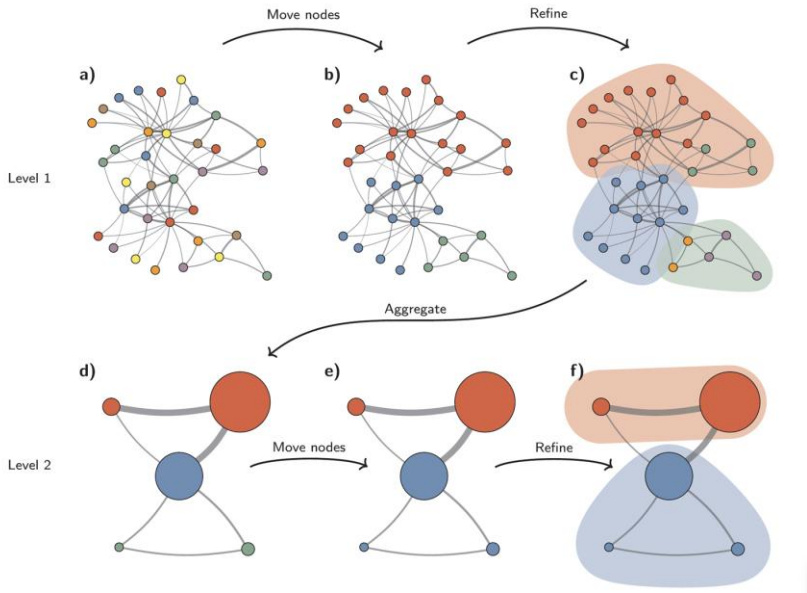
Edge, et al. "From local to global: A graph rag approach to query-focused summarization." *arXiv preprint arXiv:2404.16130*

GraphRAG: Graph Community Detection and Textualization

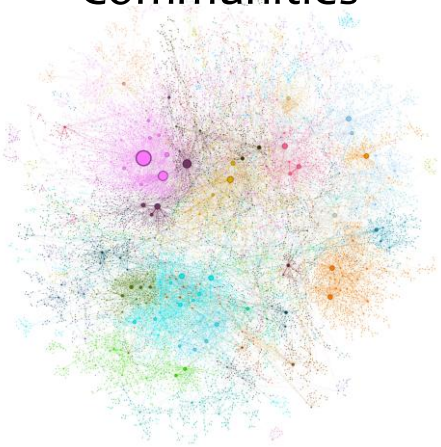
Community Detection

Knowledge Graph





Graph Communities



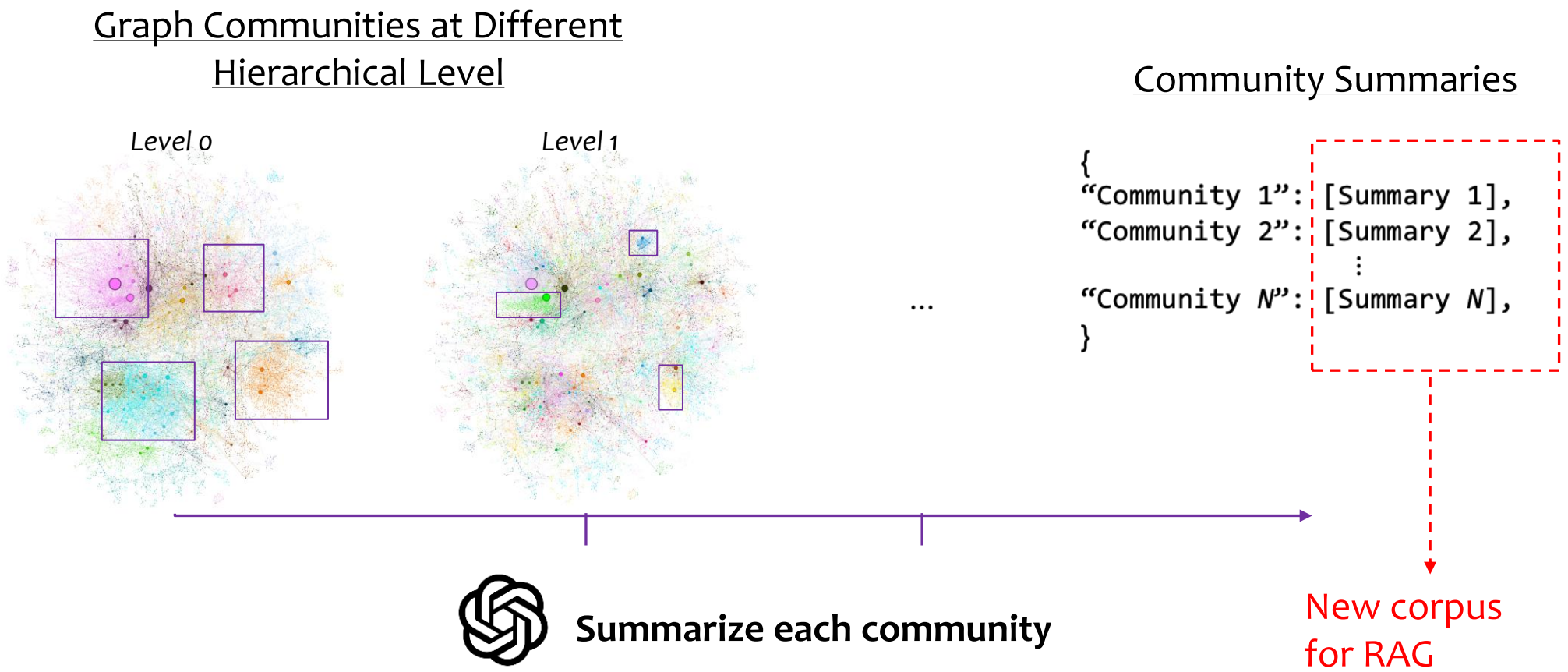
Leiden algorithm

(Leiden is a community detection algorithm that identifies well-connected groups of nodes in networks while guaranteeing connected communities and optimal node assignments through iterative refinement.)

Edge, et al. "From local to global: A graph rag approach to query-focused summarization." *arXiv preprint arXiv:2404.16130*

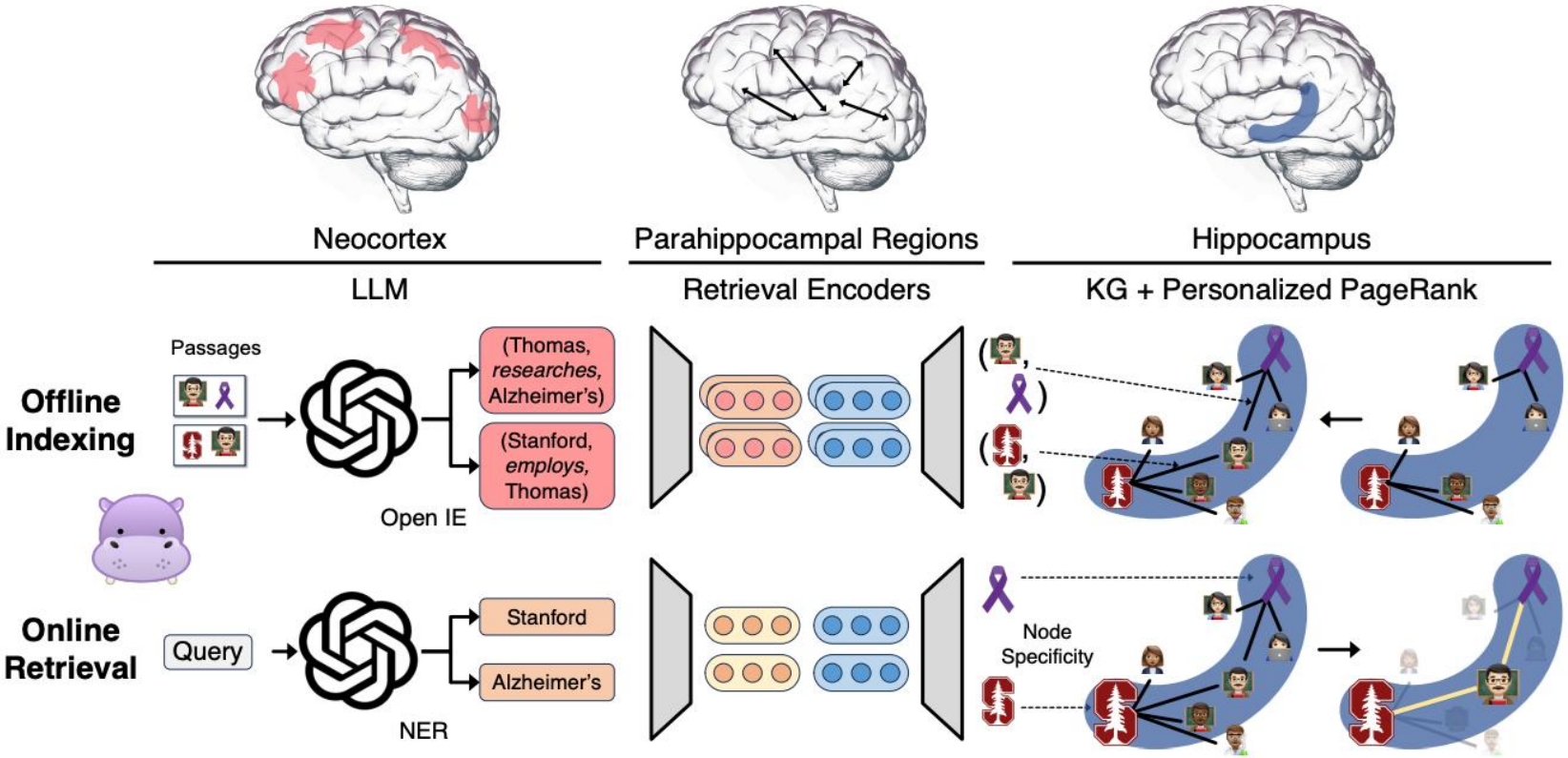
GraphRAG: Graph Community Detection and Textualization

Community Textualization



Edge, et al. "From local to global: A graph rag approach to query-focused summarization." *arXiv preprint arXiv:2404.16130*

HippoRAG: Design and Motivation



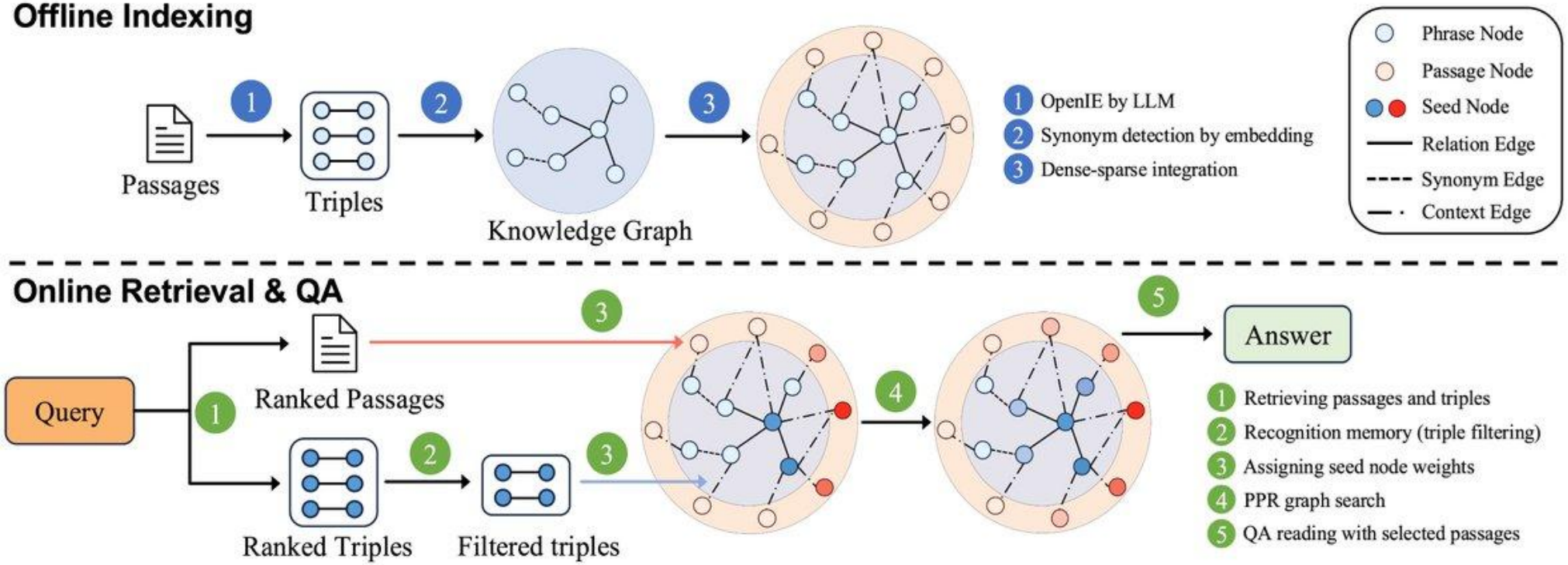
- Motivation
  - RAG's isolated encoding prevents it from connecting disparate knowledge elements
  - Human long-term memory continuously integrates info for complex sense-making and associative reasoning

- Construct KG for all the passages in the corpus using LLM
- Retrieve relevant info by finding paths in the KG between the entities in the query, using PageRank

Gutiérrez, Bernal Jiménez, et al. "HippoRAG: Neurobiologically Inspired Long-Term Memory for Large Language Models" NeurIPS 2024.
Gutiérrez, Bernal Jiménez, et al. "From RAG to Memory: Non-Parametric Continual Learning for Large Language Models", arXiv:2502.14802

HippoRAG: Methodology

- **Offline indexing:** Encode and associate info within corpus by constructing a corpus-based knowledge graph
- **Online retrieval:**
  - Leverage contextual signals over different data granularities
  - An LLM to mimic human recognition memory to determine which nodes should be omitted
  - Personalized PageRank algorithm

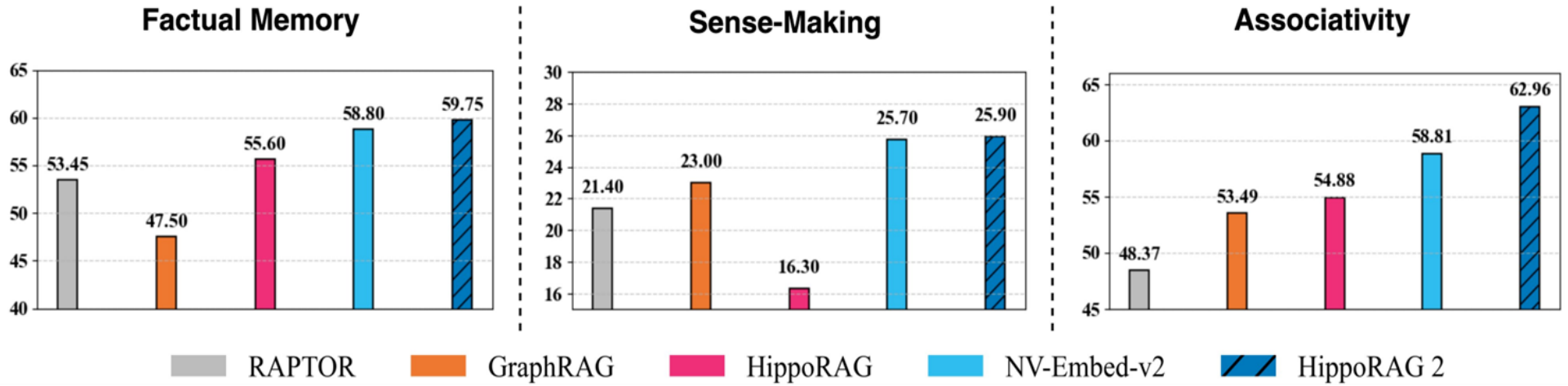


HippoRAG: Experiment results

- Strong performance in both simple factual memory tasks as well as complex associativity and sense-making tasks
- Other structure-augmented RAG systems such as LightRAG and GraphRAG underperformed compared to both HippoRAG 2 and NV-Embed-V2 on all three task categories by substantial margins, even though they use 600 to 1000% more compute for offline indexing

	MuSiQue		2Wiki		HotpotQA	
	R@2	R@5	R@2	R@5	R@2	R@5
IRCoT + BM25 (Default)	34.2	44.7	61.2	75.6	65.6	79.0
IRCoT + Contriever	39.1	52.2	51.6	63.8	65.9	81.6
IRCoT + ColBERTv2	41.7	53.7	64.1	74.4	**67.9**	82.0
IRCoT + HippoRAG (Contriever)	43.9	56.6	75.3	93.4	65.8	82.3
IRCoT + HippoRAG (ColBERTv2)	**45.3**	**57.6**	**75.8**	**93.9**	67.0	**83.0**

	HippoRAG 2	RAPTOR	LightRAG	GraphRAG
Input Tokens	9.2M (100.0%)	1.7M (18.5%)	68.5M (744.6%)	115.5M (1255.4%)
Output Tokens	3.0M (100.0%)	0.2M (6.7%)	18.3M (610.0%)	36.1M (1203.3%)

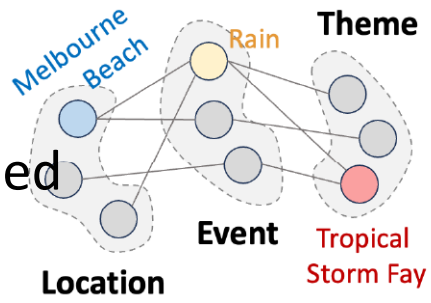


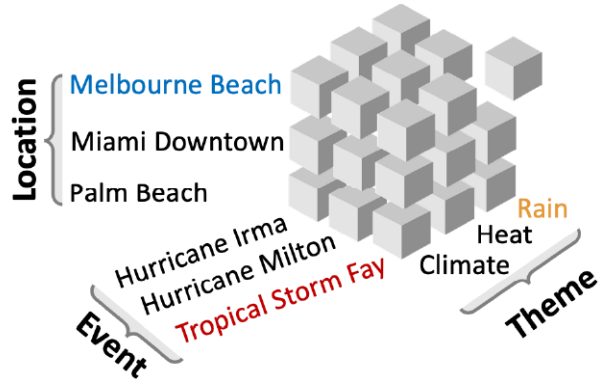
Outline

- Retrieval Augment Generation (RAG): A Brief introduction
- Retrieval and Generation Architecture for RAG
- Enhancing Information Retrieval with Corpus Knowledge and Semantic Structures
- Retrieval Improvement: Query/Pseudo-document generation and reranking
- Advanced RAG Methods: Active-RAG, Self-RAG, GraphRAG, HippoRAG, G-Retriever
- RAG by Exploring Semantic Structures: Hypercube, Structure RAG 
- Exploring RAG Applications: MolE-RAG
- Open Challenges and Future Directions

Hypercube-Based Retrieval-Augmented Generation

- Most RAG methods overlook the essential, multi-dimensional, and structured semantic info. in docs
- Hypercube: Index and allocate documents in a pre-defined multi-dimensional space
- Hypercube-RAG decomposes a query along with pre-defined hypercube dimensions and retrieves relevant docs from cubes by aligning the decomposed components with corresponding dimensions
- Hypercube-RAG: Retrieve relevant documents from the corresponding cube cells
  - Accuracy: via multidimensional, cube label-based search
  - Efficiency: a constant time complexity O(c)
  - Explainability: via compact, fine-grained labels associated in cube cells





Case Study

Query: ***How much rainfall did Melbourne Beach, Florida receive from Tropical Storm Fay?***

Hypercube-RAG

Answer: *25.28 inches* ✓
Retrieved Docs: [565, 246, 534]
Doc 565: … *Fay took her time going northward and dumped tremendous amounts of rain along the way Melbourne Beach, Fl., received as much as 25.28 inches of rain …*

Semantic-RAG (e5)

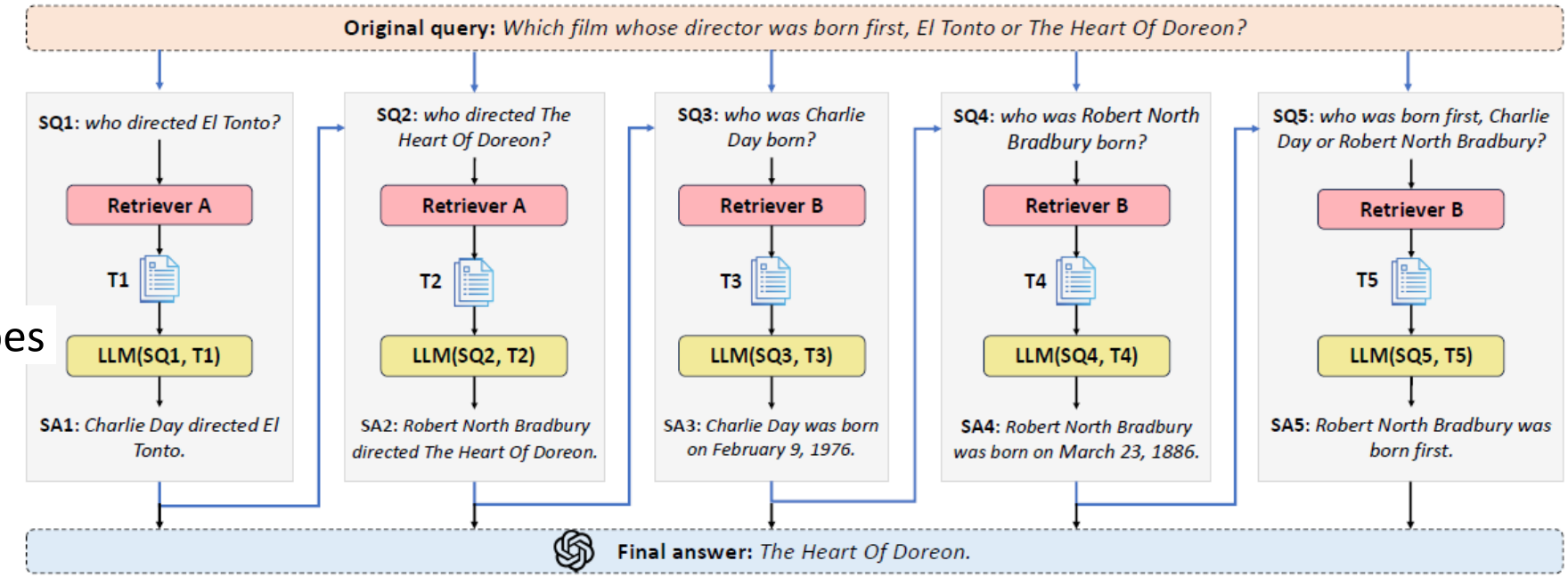
Answer: *The documents do not provide information about the rainfall that Melbourne Beach, Florida received from Tropical Storm Fay.*
Retrieved Docs: [451, 186, 364]
Doc 451: *…In Florida, 51 percent of the state was in severe to extreme drought by the end of 2010. …Gainesville received only 12.95 inches of precipitation, compared to the previous record low of 15.25 inches …*

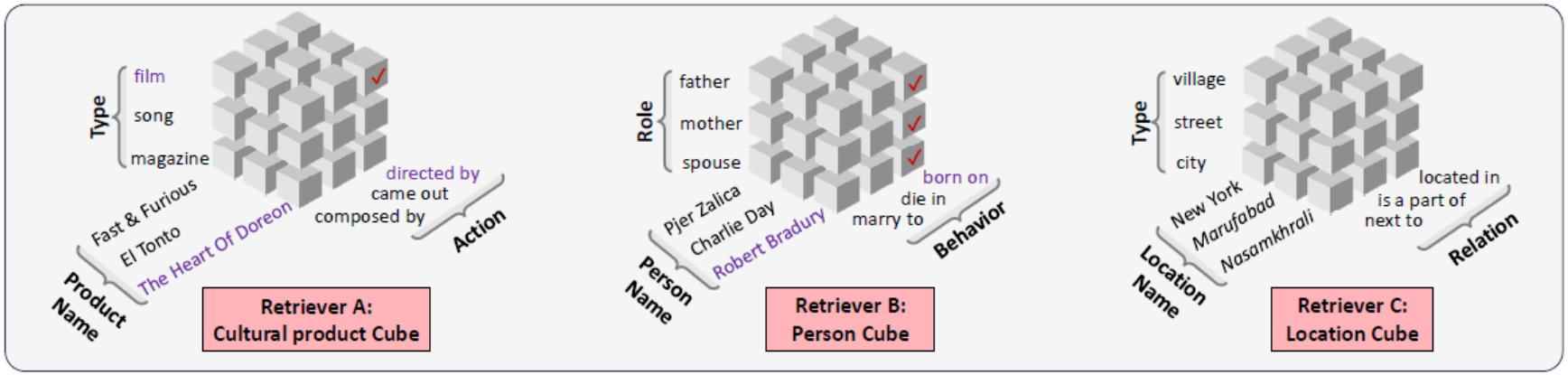
Jimeng Shi, Sizhe Zhou, Bowen Jin, Wei Hu, Runchu Tian, Shaowen Wang, Giri Narasimhan, Jiawei Han, "Hypercube-Based Retrieval-Augmented Generation for Scientific Question-Answering", arXiv: 2505.19288

Multi-cube RAG for Multi-hop Question-Answering

- Query: *"Which film whose director was born first, El Tonto or The Heart of Doreon?"* can only be answered with multiple cubes

- MultiCube-RAG for multi-hop question answering (QA)
- Features
  - Training-free
  - Select the most suitable cubes at each step to acquire the relevant knowledge & boost the subsequent reasoning process
  - High efficiency and inherent explainability





Jimeng Shi, et al, "Multi-cube RAG for Multi-hop Question-Answering", arXiv:2510.

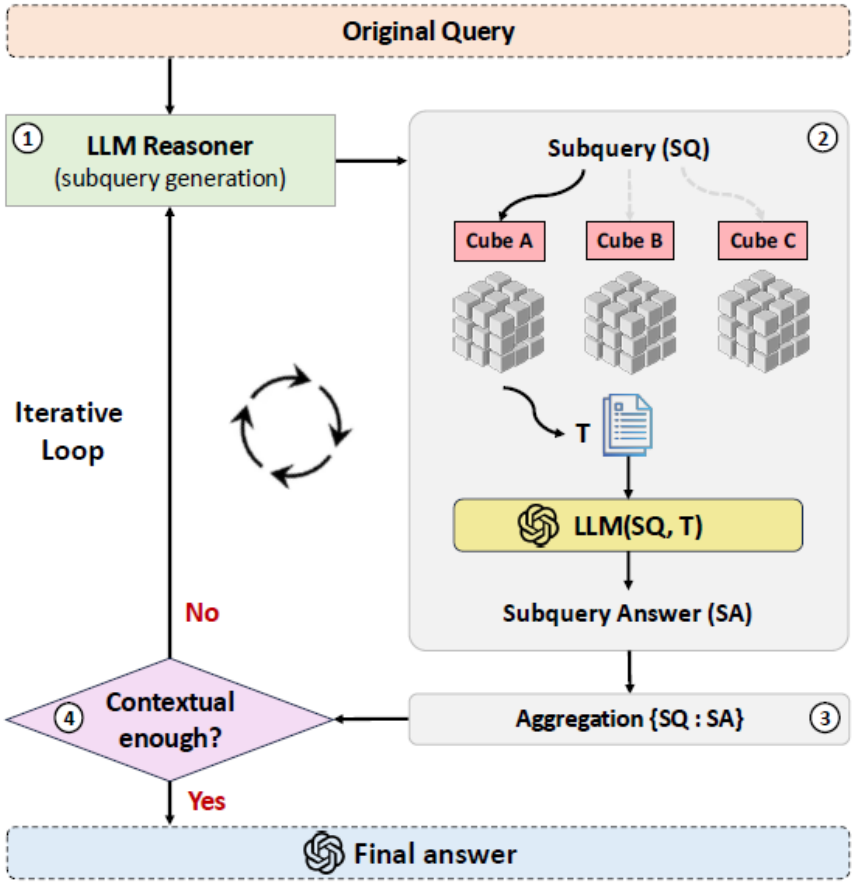
MultiCube-RAG for Multi-turn Reasoning and Retrieval

- Core idea of MultiCube-RAG: “divide and conquer”
  - An original query is first decomposed into a simple one-hop subquery
  - Then it selects the most suitable cube retriever to fetch relevant documents to output the subquery answer
  - It in turn helps the subsequent reasoning and retrieval
- LLM reasoner for subquery generation
  - Initially, the LLM is prompted only based on the original query to generate the logical subquery

$$SQ_1 = \mathcal{LLM}(\text{prompt}, q).$$

  - For the following iteration, i, the LLM is prompted based on the original query, the intermediate subqueries (SQ), and their corresponding sub-answers (SA) from MultiCube-RAG to continue reasoning by generating subsequent subqueries

$$SQ_i = \mathcal{LLM}(\text{prompt}, q, \text{SQ}_{1:i-1}, \text{SA}_{1:i-1}).$$



Multicube-RAG Performance

Dataset Statistics

Datasets	# of docs	# of questions	Question Length	Answer Length
2WikiMultiHopQA	6,119	1,000	4~27	1~14
HotpotQA*	9,811	609	5~42	1~14
MuSiQue	11,656	1,000	6~40	1~18
LV-Eval	22, 849	124	6~31	1~8

Retrieval Efficiency

Methods	2WikiQA ($k = 6,119$)	HotpotQA ($k = 9,811$)	MuSiQue ($k = 11,656$)	LV-Eval ($k = 22,849$)
BM25	10.91	38.64	62.18	73.42
NV-Embed-v2	18.93	37.17	43.05	79.33
GraphRAG	446.11	1679.06	4468.48	21802.49
HippoRAG 2	5518.34	15129.08	9872.33	26438.23
MultiCube-RAG	2.83	20.06	56.39	134.82

The natural retrieval explainability of Multi-cube RAG ensures the retrieval process can be supervisable and verifiable, offering a significant and necessary feature for those high-stakes domain applications.

QA performance (%) comparisons between **MultiCube-RAG** and baselines

Methods	2WikiMultihopQA		HotpotQA		MuSiQue		LV-Eval		Average	
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
Generator: GPT-4o-mini										
BM25 [34]	46.9	51.8	49.1	59.8	17.9	27.3	6.5	9.4	30.2	37.5
NV-Embed-v2 (7B) [23]	54.4	60.8	**57.3**	71.0	32.8	46.0	7.3	10.0	37.9	46.9
RAPTOR [35]	39.7	39.7	50.6	64.7	27.7	39.2	5.6	9.2	30.7	38.2
GraphRAG [8]	45.7	61.0	51.4	67.6	27.0	42.0	4.9	11.0	32.2	45.4
HippoRAG [11]	59.4	67.3	46.3	60.0	24.0	35.9	4.8	7.6	33.6	42.7
HippoRAG 2 [12]	60.5	69.7	56.3	**71.1**	35.0	49.3	10.5	14.0	40.6	51.0
IRCoT [42]	54.8	62.6	54.8	64.7	18.4	27.8	8.9	12.3	33.4	41.6
RankCoT [50]	52.5	62.7	54.4	67.4	23.8	35.8	5.6	8.2	33.4	43.6
MultiCube-RAG (Ours)	**63.2**	**71.5**	**57.3**	69.4	**39.5**	**50.9**	**11.3**	**14.2**	**42.8**	**51.5**
Generator: Llama3.3-70B-Instruct										
BM25 [34]	38.1	41.9	52.0	63.4	20.3	28.8	4.0	5.9	28.6	35.0
NV-Embed-v2 (7B) [23]	57.5	61.5	**62.8**	75.3	34.7	45.7	7.3	9.8	40.6	48.1
RAPTOR [35]	47.3	52.1	56.8	69.5	20.7	28.9	2.4	5.0	31.8	38.9
GraphRAG [8]	51.4	58.6	55.2	68.6	27.3	38.5	4.8	11.2	34.7	44.2
HippoRAG [11]	65.0	71.8	52.6	63.5	26.2	35.1	6.5	8.4	37.6	44.7
HippoRAG 2 [12]	65.0	71.0	62.7	**75.5**	37.2	48.6	9.7	12.9	43.6	52.0
MultiCube-RAG (Ours)	**66.8**	**74.3**	58.3	70.9	**38.8**	**49.7**	**11.2**	**14.2**	**43.8**	**52.3**

StructRAG: Motivation and Methodology

- Better leverage LLMs to transform scattered information into various structure formats
  - Hybrid info structuring mechanism: Different tasks require different knowledge structure representations for more precise reasoning

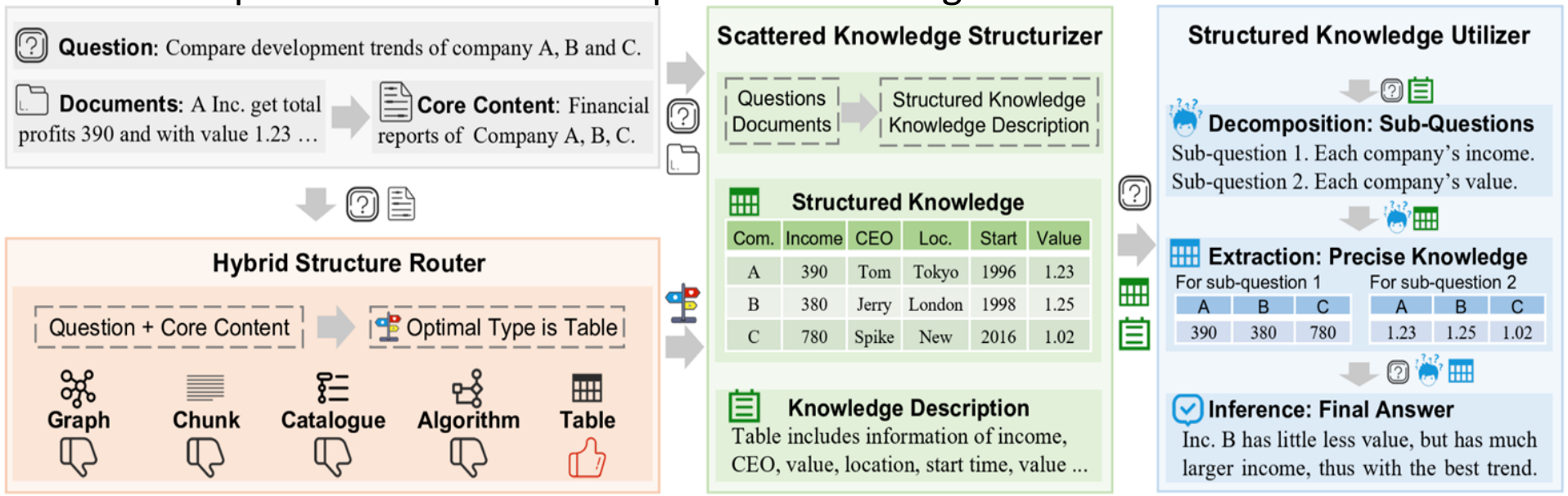


Figure 1: The overview of StructRAG framework, including an hybrid structure router to select the optimal structure type based on task requirements, a scattered knowledge structurizer to convert raw documents into structured knowledge, and a structured knowledge utilizer to decompose complex question and then effectively using the structured knowledge to infer the final answer.

- **Hybrid Structure Router:** select the most optimal structure type from five candidate structure types
- **Scattered Knowledge Structurizer:** extracts the textual knowledge scattered across raw documents for reconstruction
- **Structured Knowledge Utilizer:** LLM-based knowledge utilizer to facilitate question decomposition, precise knowledge extraction, and final answer inference

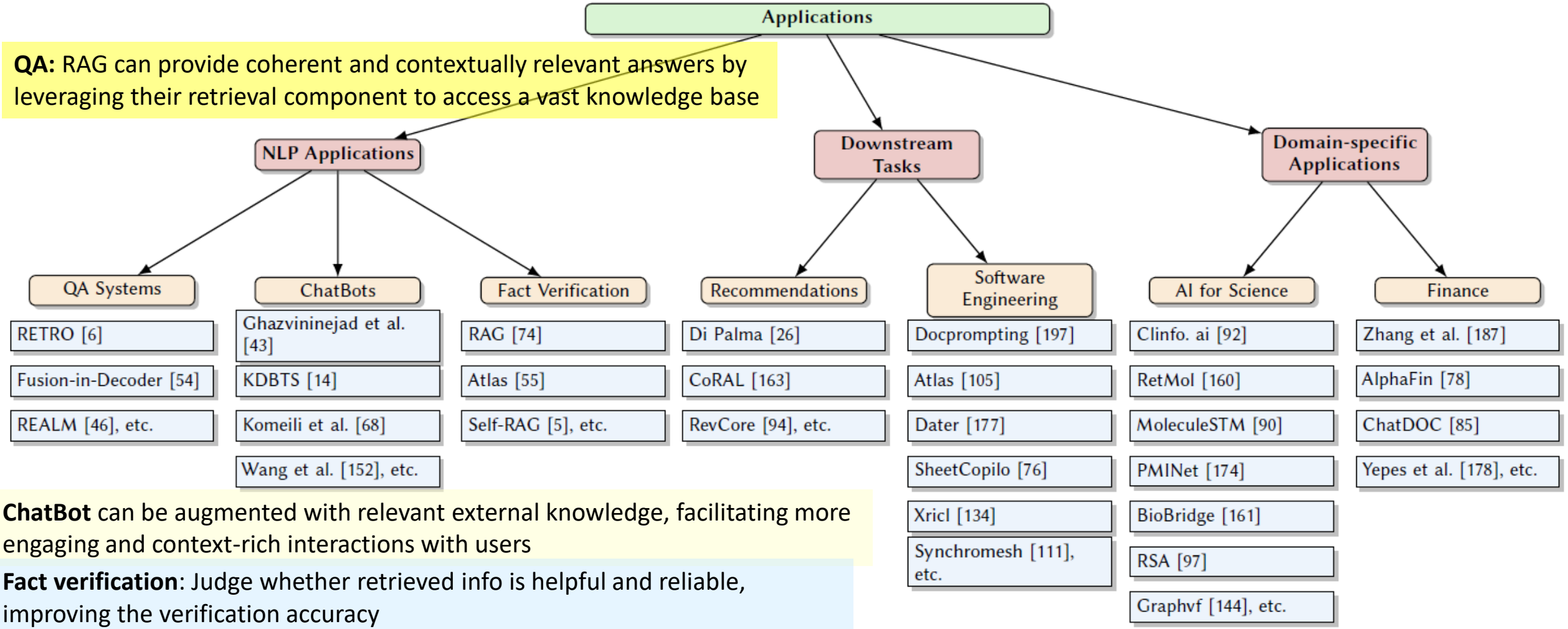
Zhuoqun Li, Xuanang Chen, Haiyang Yu, Hongyu Lin, Yaojie Lu, Qiaoyu Tang, Fei Huang, Xianpei Han, Le Sun, Yongbin Li, "StructRAG: Boosting Knowledge Intensive Reasoning of LLMs via Inference-time Hybrid Information", ICLR 2025

Outline

- Retrieval Augment Generation (RAG): A Brief introduction
- Retrieval and Generation Architecture for RAG
- Enhancing Information Retrieval with Corpus Knowledge and Semantic Structures
- Retrieval Improvement: Query/Pseudo-document generation and reranking
- Advanced RAG Methods: Active-RAG, Self-RAG, GraphRAG, HippoRAG, G-Retriever
- RAG by Exploring Semantic Structures: Hypercube, Structure RAG
- Exploring RAG Applications: MolE-RAG
- Open Challenges and Future Directions

RAG: Applications

QA: RAG can provide coherent and contextually relevant answers by leveraging their retrieval component to access a vast knowledge base



ChatBot can be augmented with relevant external knowledge, facilitating more engaging and context-rich interactions with users

Fact verification: Judge whether retrieved info is helpful and reliable, improving the verification accuracy

Providing personalized and contextually relevant **recommendations** by integrating retrieval and generation processes

Software Eng.: Retrieve top-ranked codes/summaries from the codebase and aggregate them with input to enhance code generation summarization

A Performance Study on MedRAG

5 Datasets	7,663 Questions	2-4 Choices
MMLU-Med	Which of the following best describes ... ?	A / B / C / D
MedQA-US	A 72-year-old man comes to the physicians ... ?	A / B / C / D
MedMCQA	Axonal transport is:	A / B / C / D
PubMedQA*	Is anorectal endosonography valuable ... ?	Yes / No / Maybe
BioASQ-Y/N	Is medical hydrology the same as Spa ...?	Yes / No

Figure 1: Composition of the MIRAGE benchmark.

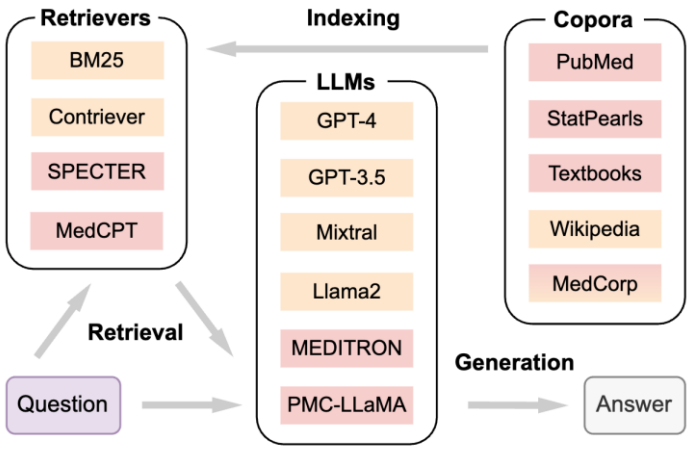
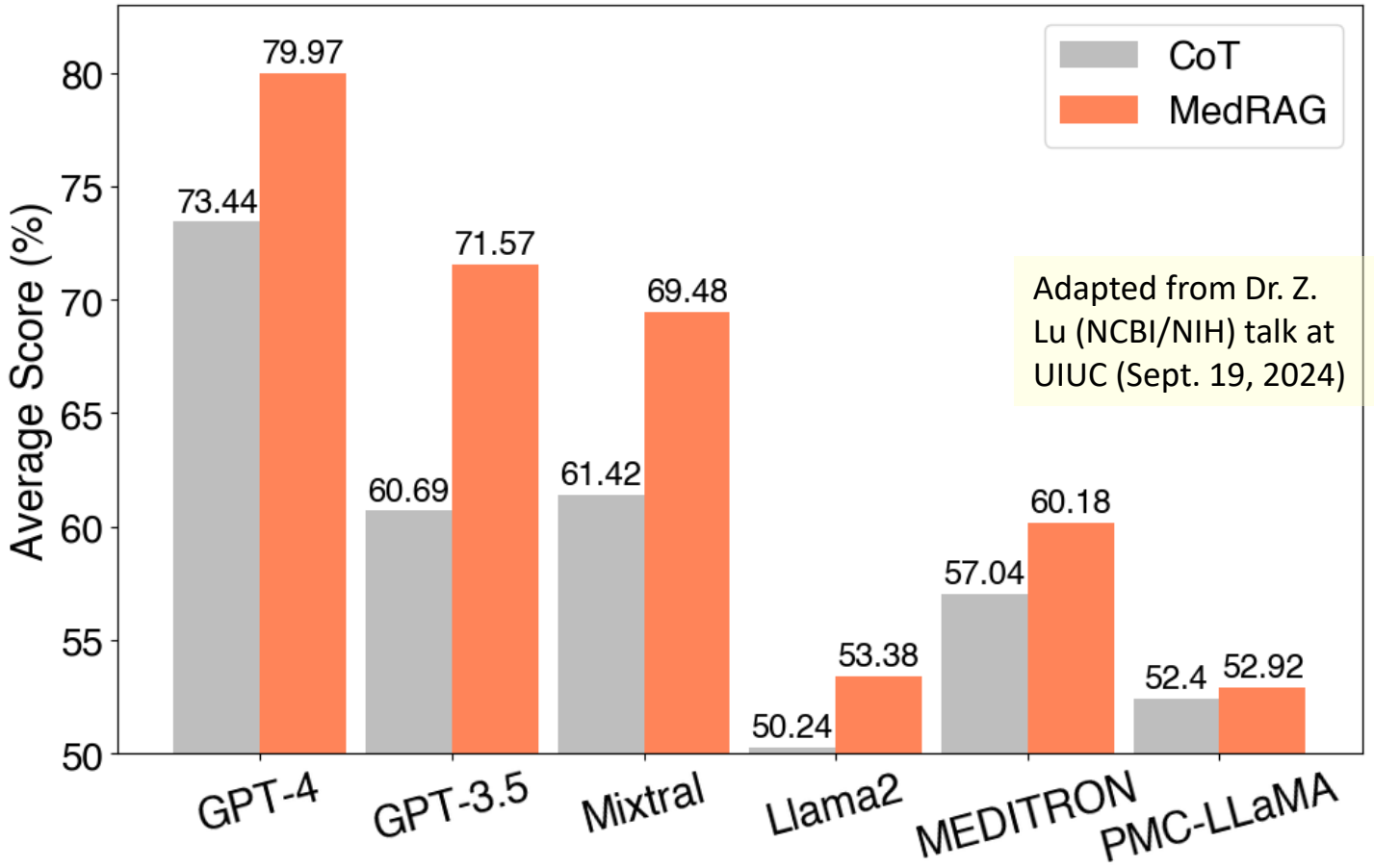


Figure 2: Component overview of the MEDRAG toolkit.

Medical **I**nformation **R**etrieval-**A**ugmented **G**eneration **E**valuation



Adapted from Dr. Z. Lu (NCBI/NIH) talk at UIUC (Sept. 19, 2024)

MedRAG improves the accuracy of different LLMs over chain-of-thought (CoT) prompting

Guangzhi Xiong, Qiao Jin, Zhiyong Lu, Aidong Zhang, "Benchmarking Retrieval-Augmented Generation for Medicine", arXiv:2402.13178

MolE-RAG, A Training-free Framework for LLM-based Molecular Property Prediction

Does CN1C(=O)N2C=NC(=C2N=N1)C(=O)N have blood-brain barrier permeability?

MolE-RAG

Context Injection

Synonyms & IUPAC: Temozolomide, Temodar

Descriptors: TPSA 24.4, LogP 1.45, HBD 1

Functional groups: tetrazine, carboxamide, amine group

Structural Retrieval

similar molecules (AtomPair):

CN1C(=O)N2C=NC(=C2N=N1)C(=O)NC similarity: 0.834

CN1C(=O)N2C=CC(=C2N=N1)C(=O)N similarity: 0.781

Text Retrieval

TMZ crosses the CNS for efficient drug delivery and drug……

Role of Temozolomide in the Treatment of Brain Cancers……

Temozolomide is an anticancer medication that treats brain…..

Yes, it is BBB permeable.

Scientific research needs deeper knowledge than those captured in LLMs

1 Textual RAG

1 Query Construction & Filtering

Queries combine a task description, domain keywords, and molecule names from PubChem.

CNS, Permeable… + Efflux, Soluble….. + Carmustine, BCNU…..

2 Literature Retrieval

Retrieve external literature evidence from chemical and biomedical sources.

PubMed, Wikipedia, Semantic Scholar, uspto

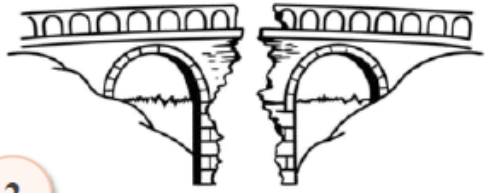
Carmustine for Tumors — Carmustine is highly lipid-soluble — PMID: 330416

3 Evidence Selection

Take the top 5 text snippets ranked by BM25.

Top 5 Text Snippets: 1 — 0.87; 2 — 0.78; 3 — 0.75; … 5 — 0.72

Bridging the Gap



2 Molecular Context

Synonyms & IUPAC

Maps the molecule to known names.

[c1ccccc1] → PubChem → [Benzine]

RDKit Descriptors

Adds computable physicochemical properties.

LogP: 0.46, HBD: 1, RotB: 3, HBA: 2, MW: 191.12 g/mol, QED: 0.35

Function Groups

Identifies key chemical substructures and group annotation

Amide, Phenol, Aromatic Ring

3 Structural RAG

1 Fingerprint Encoding

Each molecule is encoded using molecular fingerprints to retrieve the most similar training molecules.

2 Similarity Search

Retrieve the top 5 most similar molecules from the training set.

0.87, 0.82, 0.71, 0.68

3 Task-Adaptive Selection

Select the best fingerprint for each task based on validation set.

Different fingerprint per task: BBBP ≠ BACE ≠ ··· SIDER

Task	Best Fingerprint
BBBP	ECFP2
BACE	Topological Torsion
ESOL	Atom Pair
…	…

9 Molecular Property Prediction Tasks

Binary Classification: BBBP, ClinTox, BACE, Tox21, Sider

Regression: Esol, Lipo, Freesolv

LLM-augmented with (i) retrieved text passages, (ii) structurally similar labeled molecules, and (iii) molecule-specific descriptors for Chemistry prediction

Joey Chan, Wonbin Kweon, Ashley Shin, Niharika Bhattacharjee, Pengcheng Jiang, Yue Guo, Jiawei Han, "MolE-RAG: Molecular Structure-Enhanced Retrieval-Augmented Generation for Chemistry", arXiv:2606.05693

MolE-RAG: Experiments and Performance Study

Research question: For each chemistry task, can we identify data and methods that enable general-purpose LLMs or ChemLLMs to match or surpass specialized models?

- **MOLE-RAG** augments the prompt with retrieved text, structurally similar labeled molecules, and molecule-specific descriptors
- **6 Classification tasks**: BBBP, BACE, ClinTox, HIV, Tox21, and SIDER
- **3 Regression tasks:** ESOL, FreeSolv, and Lipophilicity
- Compare against a set of **supervised and self-supervised graph baselines**: MGCN, GROVER, and MolCLR across three GNN backbones (GCN, GIN, CMPNN)
- Test effect of **adding MolE-RAG on top of 7 current LLM models**: Llama3.2-3B, Mistral 7B-Inst, Qwen3-4B-Inst, ChemDFM 14B, GPT-4o-mini, GPT-5.4-nano

	Classification						Regression		
Dataset	BBBP	Tox21	SIDER	ClinTox	BACE	HIV	ESOL	FreeSolv	Lipophilicity
# Molecules	2039	7831	1427	1478	1513	41127	1128	642	4200
# Tasks	1	12	27	2	1	1	1	1	1
MGCN	85.0	70.7	55.2	63.4	73.4	—	1.266	3.349	1.113
GROVER	84.1	80.0	58.2	70.7	83.3	67.5	1.475	3.235	0.974
MolCLR$_{GCN}$	73.2	73.5	61.0	85.9	82.6	76.8	1.102	2.241	0.819
MolCLR$_{GIN}$	73.5	76.7	60.7	90.4	83.5	77.6	1.091	2.017	0.824
MolCLR$_{CMPNN}$	72.4	78.4	59.7	88.0	85.0	77.8	0.911	2.021	0.875
SchNet	84.8	76.6	54.5	71.7	76.6	—	1.045	3.215	0.909
Llama$_{3.2\text{-}3B}$									
Baseline	48.2	50.0	49.9	48.3	50.4	51.8	3.175	5.995	4.095
+MolE-RAG	48.3	52.5	50.3	48.8	51.8	58.7	2.181	4.343	1.706
Mistral$_{7B\text{-}Inst}$									
Baseline	46.1	53.1	50.4	50.0	48.6	45.9	3.474	12.585	3.347
+MolE-RAG	74.6	54.8	50.8	62.1	76.8	62.4	2.890	4.128	1.210
Qwen3$_{4B\text{-}Inst}$									
Baseline	53.0	49.6	51.1	50.3	48.9	53.3	2.537	5.524	2.456
+MolE-RAG	**80.1**	66.2	52.8	56.6	71.1	**73.5**	1.852	2.700	1.105
ChemDFM$_{14B}$									
Baseline	79.0	55.0	49.2	49.7	68.2	55.2	4.086	6.523	2.347
+MolE-RAG	78.1	**68.8**	53.9	59.4	**80.4**	69.2	1.467	3.476	1.021
GPT-4o-mini									
Baseline	54.9	52.3	51.6	52.0	51.3	60.0	2.835	5.475	1.667
+MolE-RAG	74.7	59.3	53.5	**63.3**	57.4	61.3	1.171	**2.555**	0.964
GPT-5.4-nano									
Baseline	53.4	52.7	51.6	52.8	54.3	50.5	1.723	5.851	1.448
+MolE-RAG	77.0	67.0	**56.6**	62.7	78.3	65.1	**1.138**	2.880	**0.894**

Outline

- Retrieval Augment Generation (RAG): A Brief introduction
- Retrieval and Generation Architecture for RAG
- Enhancing Information Retrieval with Corpus Knowledge and Semantic Structures
- Retrieval Improvement: Query/Pseudo-document generation and reranking
- Advanced RAG Methods: Active-RAG, Self-RAG, GraphRAG, HippoRAG, G-Retriever
- RAG by Exploring Semantic Structures: Hypercube, Structure RAG
- Exploring RAG Applications: MolE-RAG
- Open Challenges and Future Directions 

RAG: Issues and Future Work

- **RAG vs. Long Context**: Is RAG still necessary when LLMs are not constrained by context?
- **RAG Robustness:** "Misinformation can be worse than no information"
  - Analyze which type of docs should be retrieved, evaluate the relevance of the docs
- **Hybrid Approaches**: Combine RAG w. fine-tuning; introduce SLMs w. specific functionalities into RAG
- **Scaling laws vs. possible Inverse Scaling Law of RAG** (i.e., smaller models outperform larger ones)
- **Production-Ready RAG:** efficiency, recall, ensuring data security (protect doc sources or metadata)
- **Multi-modal RAG:**
  - Image: Retrieving and generating text and images; efficient visual language pre-training, zero-shot image-to-text conversions; image generation to steer the LM's text generation
  - Audio and Video: Convert machine-translated data into speech-translated data; voice-to-text conversion; prediction of event boundaries and textual descriptions
  - Code: Retrieving code examples that align with developers' objectives through encoding and frequency analysis; knowledge graph question-answering

References: Survey Papers

- Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li (2024), “A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models”, arXiv:2405.06211v3
- Y. Gao et al, (2023), “Retrieval-Augmented Generation for Large Language Models: A Survey”, arXiv:2312.10997
- Wayne Xin Zhao, Jing Liu, Ruiyang Ren and Ji-Rong Wen, “Dense Text Retrieval based on Pretrained Language Models: A Survey” arXiv: 2211.14876 (arxiv.org)
- Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, Jianfeng Gao, "Large Language Models: A Survey", https://arxiv.org/abs/2402.06196
- Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig (2021), “Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing”, arXiv:2107.13586v1
- L. Wang, C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang, Z. Chen, J. Tang, X. Chen, Y. Lin et al., “A survey on large language model based autonomous agents,” arXiv:2308.11432.
- Z. Xi, W. Chen, X. Guo, W. He, Y. Ding, B. Hong, M. Zhang, J. Wang, S. Jin, E. Zhou et al., “The rise and potential of large language model-based agents: A survey,” arXiv:2309.07864, 2023.
- Z. Durante, Q. Huang, N. Wake, R. Gong, J. S. Park, B. Sarkar, R. Taori, Y. Noda, D. Terzopoulos, Y. Choi, K. Ikeuchi, H. Vo, L. Fei-Fei, and J. Gao, “Agent AI: Surveying the horizons of multimodal interaction,” arXiv:2401.03568
- Yu Zhang, Xiusi Chen, Bowen Jin, Sheng Wang, Shuiwang Ji, Wei Wang, Jiawei Han, “A Comprehensive Survey of Scientific Large Language Models and Their Applications in Scientific Discovery”, in Proc. 2024 Conf. on Empirical Methods in Natural Language Processing (EMNLP’24), Nov. 2024

References (I)

- Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, Hannaneh Hajishirzi: “Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection”, arXiv:2310.11511
- Joey Chan, Wonbin Kweon, Ashley Shin, Niharika Bhattacharjee, Pengcheng Jiang, Yue Guo, Jiawei Han, "MolE-RAG: Molecular Structure-Enhanced Retrieval-Augmented Generation for Chemistry", arXiv:2606.05693
- Linyi Ding, Jinfeng Xiao, Sizhe Zhou, Chaoqi Yang, Jiawei Han, “Topic-Oriented Open Relation Extraction with A Priori Seed Generation”, in Proc. 2024 Conf. on Empirical Methods in Natural Language Processing (EMNLP’24)
- Edge, Darren, et al. "From Local to Global: A GraphRAG Approach to Query-Focused Summarization“, arXiv:2404.16130
- Guo, et al. "Deepseek-r1: Incentivizing reasoning capability in LLMs via reinforcement learning." Nature 2025
- Gutiérrez, Bernal Jiménez, et al. "HippoRAG: Neurobiologically Inspired Long-Term Memory for Large Language Models" NeurIPS 2024
- Gutiérrez, Bernal Jiménez, et al. "From RAG to Memory: Non-Parametric Continual Learning for Large Language Models“, arXiv:2502.14802
- Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V. Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, Bryan Hooi, "G-retriever: Retrieval-augmented generation for textual graph understanding and question answering." NeurIPS 2024
- Pengcheng Jiang, Jiacheng Lin, Lang Cao, Runchu Tian, SeongKu Kang, Zifeng Wang, Jimeng Sun, Jiawei Han, “DeepRetrieval: Hacking Real Search Engines and Retrievers with LLMs via Reinforcement Learning”, COLM’2025
- Pengcheng Jiang, Xueqiang Xu, Jiacheng Lin, Jinfeng Xiao, Zifeng Wang, Jimeng Sun, Jiawei Han“s3: You Don't Need That Much Data to Train a Search Agent via RL,” EMNLP 2025
- Z. Jiang, F. F. Xu, L. Gao, Z. Sun, Q. Liu, J. Dwivedi-Yu, Y. Yang, J. Callan, and G. Neubig, “Active retrieval augmented generation,” 2023
- Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan O Arik, Dong Wang, Hamed Zamani, and Jiawei Han, “Search-R1: Training LLMs to Reason and Leverage Search Engines with Reinforcement Learning,” COLM’25

References (II)

- Harshit Joshi, Priyank Shethia, Jadelynn Dao, and Monica S. Lam, "Contexts are Never Long Enough: Structured Reasoning for Scalable Question Answering over Long Document Sets", ArXiv:2604-22294
- SeongKu Kang, Shivam Agarwal, Bowen Jin, Dongha Lee, Hwanjo Yu, and Jiawei Han, “Improving Retrieval in Theme-specific Applications using a Corpus Topical Taxonomy”, in Proc. of The Web Conference 2024 (WWW’2024)
- SeongKu Kang, Yunyi Zhang, Pengcheng Jiang, Dongha Lee, Jiawei Han, Hwanjo Yu, “Taxonomy-guided Semantic Indexing for Academic Paper Search”, in Proc. 2024 Conf. on Empirical Methods in Natural Language Processing (EMNLP’24), Nov. 2024
- SeongKu Kang, Bowen Jin, Wonbin Kweon, Yu Zhang, Dongha Lee, Jiawei Han, Hwanjo Yu, “Improving Scientific Document Retrieval with Concept Coverage-based Query Set Generation”, WSDM’25
- Wonbin Kweon, Runchu Tian, Seongku Kang, Pengcheng Jiang, Zhiyong Lu, Jiawei Han and Hwanjo Yu, “PairSem: LLM-Guided Pairwise Semantic Matching for Scientific Document Retrieval”, WWW'26
- Dongha Lee, Jiaming Shen, Seongku Kang, Susik Yoon, Jiawei Han and Hwanjo Yu, “TaxoCom: Topic Taxonomy Completion with Hierarchical Discovery of Novel Topic Clusters”, in Proc. The ACM Web Conf. 2022 (WWW’22), April 2022
- P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. tau Yih, T. Rocktäschel, S. Riedel, and D. Kiela. “Retrieval-augmented generation for knowledge-intensive NLP tasks”, In NeurIPS, 2020.
- Zhuoqun Li, Xuanang Chen, Haiyang Yu, Hongyu Lin, Yaojie Lu, Qiaoyu Tang, Fei Huang, Xianpei Han, Le Sun, Yongbin Li, "StructRAG: Boosting Knowledge Intensive Reasoning of LLMs via Inference-time Hybrid Information”, ICLR 2025
- Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, Nan Duan, “Query Rewriting for Retrieval-Augmented Large Language Models”, EMNLP’23
- Chaitanya Malaviya, Subin Lee , Sihao Chen, Elizabeth Sieber, Mark Yatskar, Dan Roth, “ExpertQA : Expert-Curated Questions and Attributed Answers”, NAACL’2024

References (III)

- Rodrigo Nogueira, Wei Yang, Jimmy Lin, Kyunghyun Cho, “Document Expansion by Query Prediction”, arXiv’19
- O. Ovadia, et al (2023), “Fine-tuning or retrieval? comparing knowledge injection in LLMs,” arXiv:2312.05934
- B. Paranjape, S. Lundberg, S. Singh, H. Hajishirzi, L. Zettlemoyer, and M. T. Ribeiro, “Art: Automatic multi-step reasoning and tool-use for large language models,” 2023
- Jimeng Shi, Sizhe Zhou, Bowen Jin, Wei Hu, Runchu Tian, Shaowen Wang, Giri Narasimhan, Jiawei Han, “Hypercube-Based Retrieval-Augmented Generation for Scientific Question-Answering”, arXiv: 2505.19288
- Jimeng Shi, Wei Hu, Runchu Tian, Bowen Jin, Wonbin Kweon, SeongKu Kang, Yunfan Kang, Dingqi Ye, Sizhe Zhou, Shaowen Wang, Jiawei Han, "MultiCube-RAG for Multi-hop Question Answering", arXiv:2602.15898\
- Jiashuo Sun, et al. “DynamicRAG: Leveraging Outputs of Large Language Model as Feedback for Dynamic Reranking in Retrieval-Augmented Generation” *NeurIPS 2025*
- Jiashuo Jiang, et al. “Reasoning-Enhanced Healthcare Predictions with Knowledge Graph Community Retrieval” ICLR 2025
- Jiashuo Sun, et al. “Rethinking the Reranker: Boundary-Aware Evidence Selection for Robust Retrieval-Augmented Generation” ICML 2026
- Jiashuo Sun, et al. “Retrieval is Cheap, Show Me the Code: Executable Multi-Hop Reasoning for Retrieval-Augmented Generation” *arXiv preprint arXiv: 2605.12975*
- Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, Zhaochun Ren, “Is ChatGPT Good at Search? Investigating Large Language Models as Re-Ranking Agents”, EMNLP’23
- JingjinWang and Jiawei Han, “PropRAG: Guiding Retrieval with Beam Search over Proposition Paths”, EMNLP’2025

References (IV)

- Minzheng Wang, Longze Chen, Cheng Fu, Shengyi Liao, Xinghua Zhang, Bingli Wu, Haiyang Yu, Nan Xu, Lei Zhang, Run Luo2, Yunshui Li, Min Yang, Fei Huang, Yongbin Li, "Leave No Document Behind: Benchmarking Long-Context LLMs with Extended Multi-Doc QA", ArXiv: 2405-17419v2
- Xiao Wang, Craig Macdonald, Nicola Tonellotto, Iadh Ounis, “Pseudo-Relevance Feedback for Multiple Representation Dense Retrieval”, ICTIR’21
- Junlin Wu, Xianrui Zhong, Jiashuo Sun, Bolian Li, Bowen Jin, Jiawei Han, Qingkai Zeng, "Structure-R1: Dynamically Leveraging Structural Knowledge in LLM Reasoning through Reinforcement Learning", arXiv:2510.15191
- Zhi Zheng, Kai Hui, Ben He, Xianpei Han, Le Sun, Andrew Yates, “BERT-QE: Contextualized Query Expansion for Document Re-ranking”, EMNLP’20
- Sizhe Zhou, Yu Meng, Bowen Jin, Jiawei Han, “Grasping the Essentials: Tailoring Large Language Models for Zero-Shot Relation Extraction”, in Proc. 2024 Conf. on Empirical Methods in Natural Language Processing (EMNLP’24), Nov. 2024