KDD2024
BARCELONA, SPAIN

THE HONG KONG POLYTECHNIC UNIVERSITY
香港理工大學

RAG Meets LLM: Towards Retrieval-Augmented Large Language Models

Website: https://advanced-recommender-systems.github.io/RAG-Meets-LLMs/

Survey: https://arxiv.org/pdf/2405.06211

Wenqi Fan[1], Yujuan Ding[1], Shijie Wang[1], Liangbo Ning[1], Hengyun Li[1], Dawei Yin[2], Tat-Seng Chua[3], and Qing Li[1]

[1]The Hong Kong Polytechnic University, [2]Baidu Inc, [3]National University of Singapore

August 25th (Day 1), 10:00-13:00

KDD 2024, Barcelona, Spain

Survey

Website

Tutorial Outline

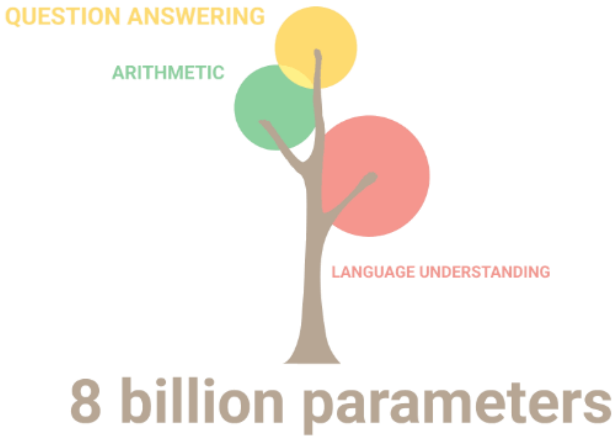
- **Part 1: Introduction of Retrieval Augmented Large Language Models (RA-LLMs) (Dr. Wenqi Fan)**
- **Part 2: Architecture** of RA-LLMs and **Main Modules** (Dr. Yujuan Ding)
- **Part 3: Learning** Approach of RA-LLMs (Liangbo Ning)
- **Part 4: Applications** of RA-LLMs (Shijie Wang)
- **Part 5: Challenges and Future Directions** of RA-LLMs (Dr. Wenqi Fan)
- **Part 6: Q&A**

Website of this tutorial
Check out the slides and more information!

Website

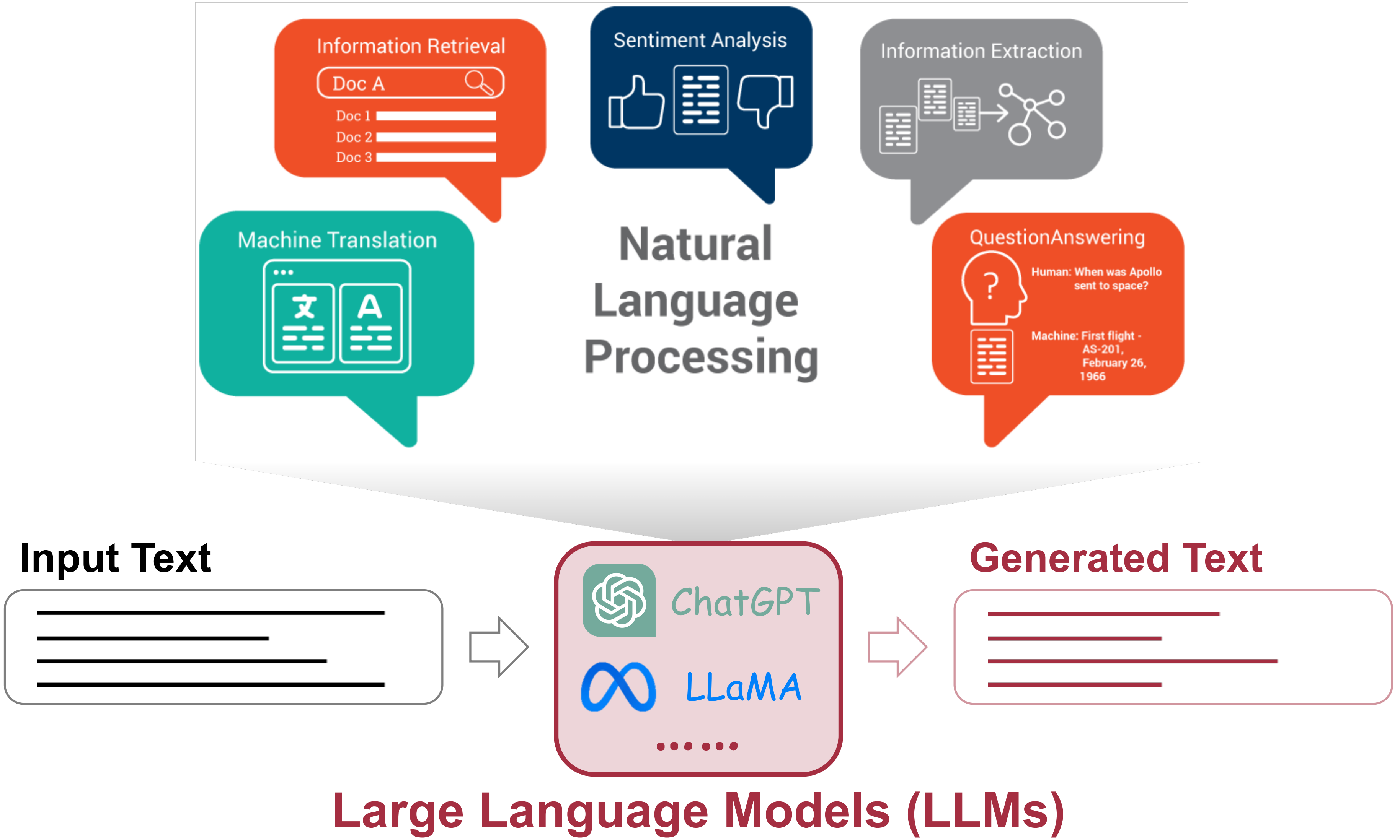
Large Language Models (LLMs)

OpenAI | Meta | Google AI T5 | ……

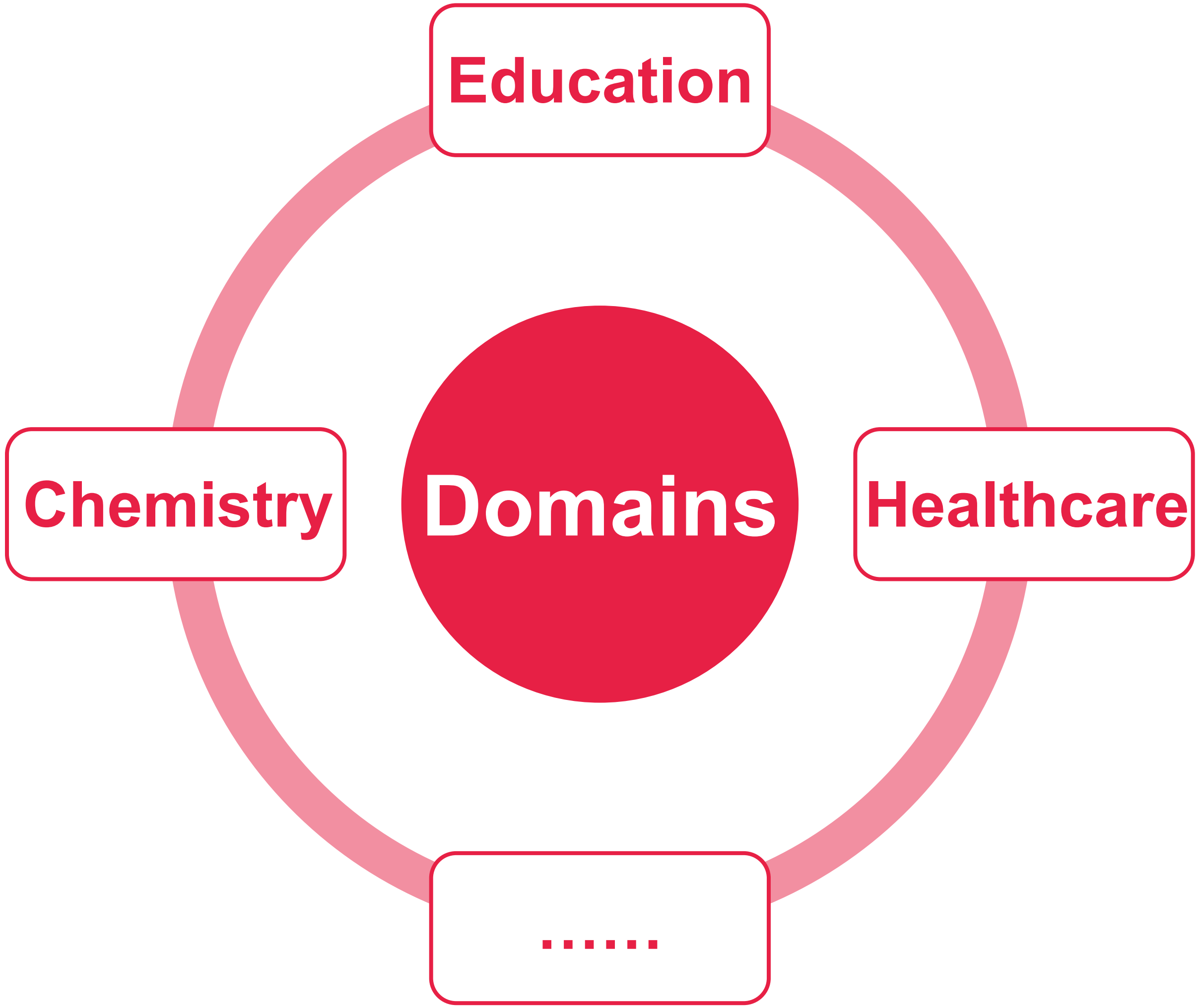


https://github.com/Hannibal046/Awesome-LLM/tree/main

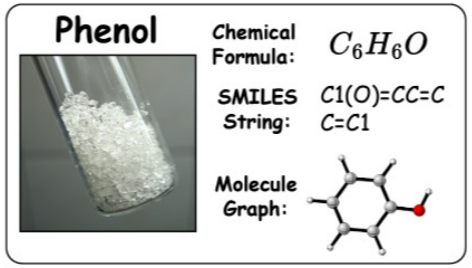
Large Language Models (LLMs)



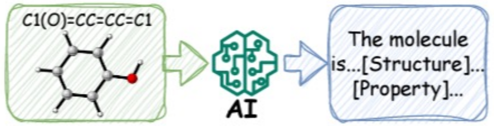
LLMs in Downstream Domains



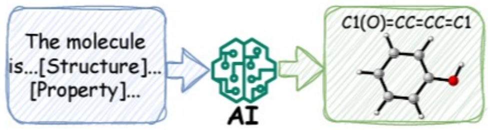
- **Molecule discovery**, etc.

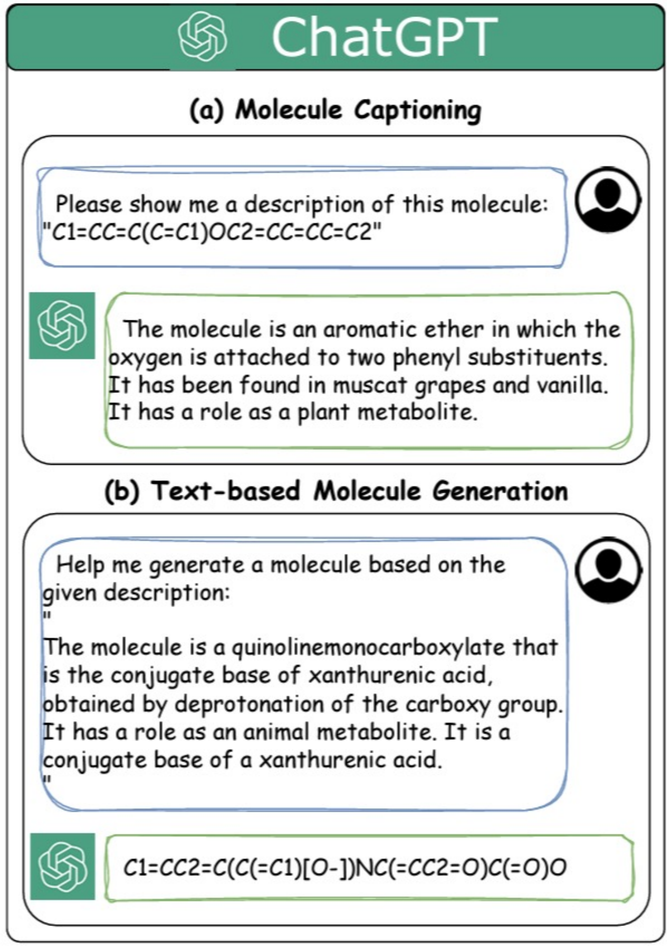


(a) Molecule Representations.



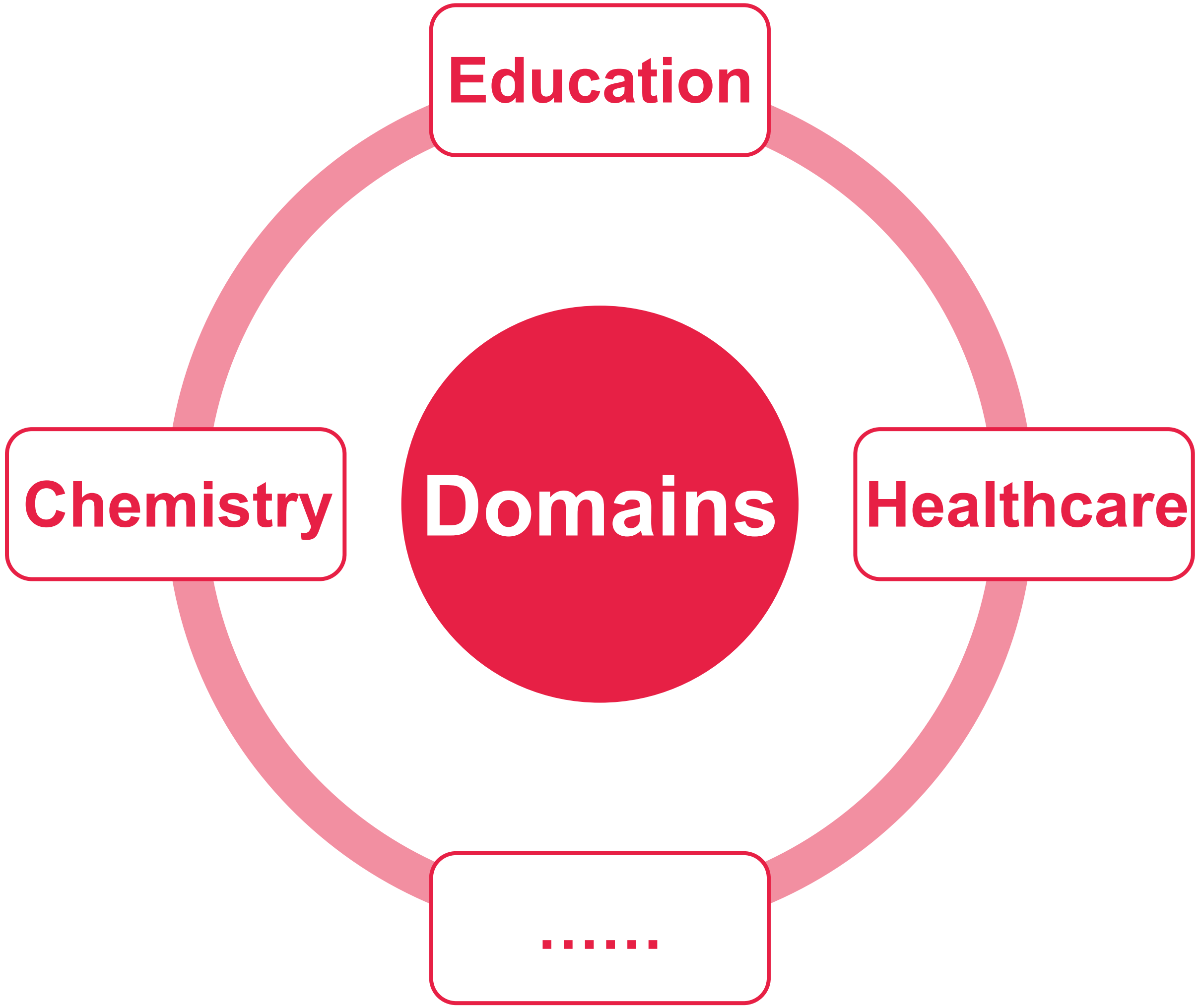
(b) Molecule Captioning.



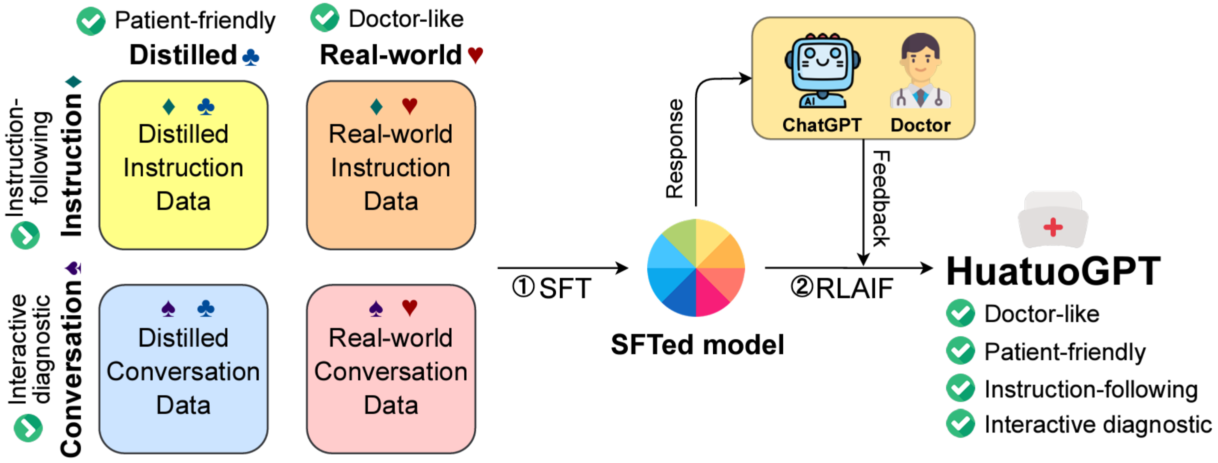


Li et al, 2024, Empowering Molecule Discovery for Molecule-Caption Translation with Large Language Models: A ChatGPT Perspective,
Liu et al., 2024, MolecularGPT: Open Large Language Model (LLM) for Few-Shot Molecular Property Prediction,

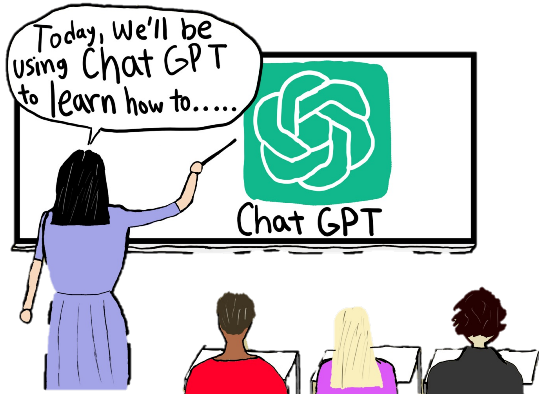
LLMs in Downstream Domains



❑ **Medical consultation**, etc.

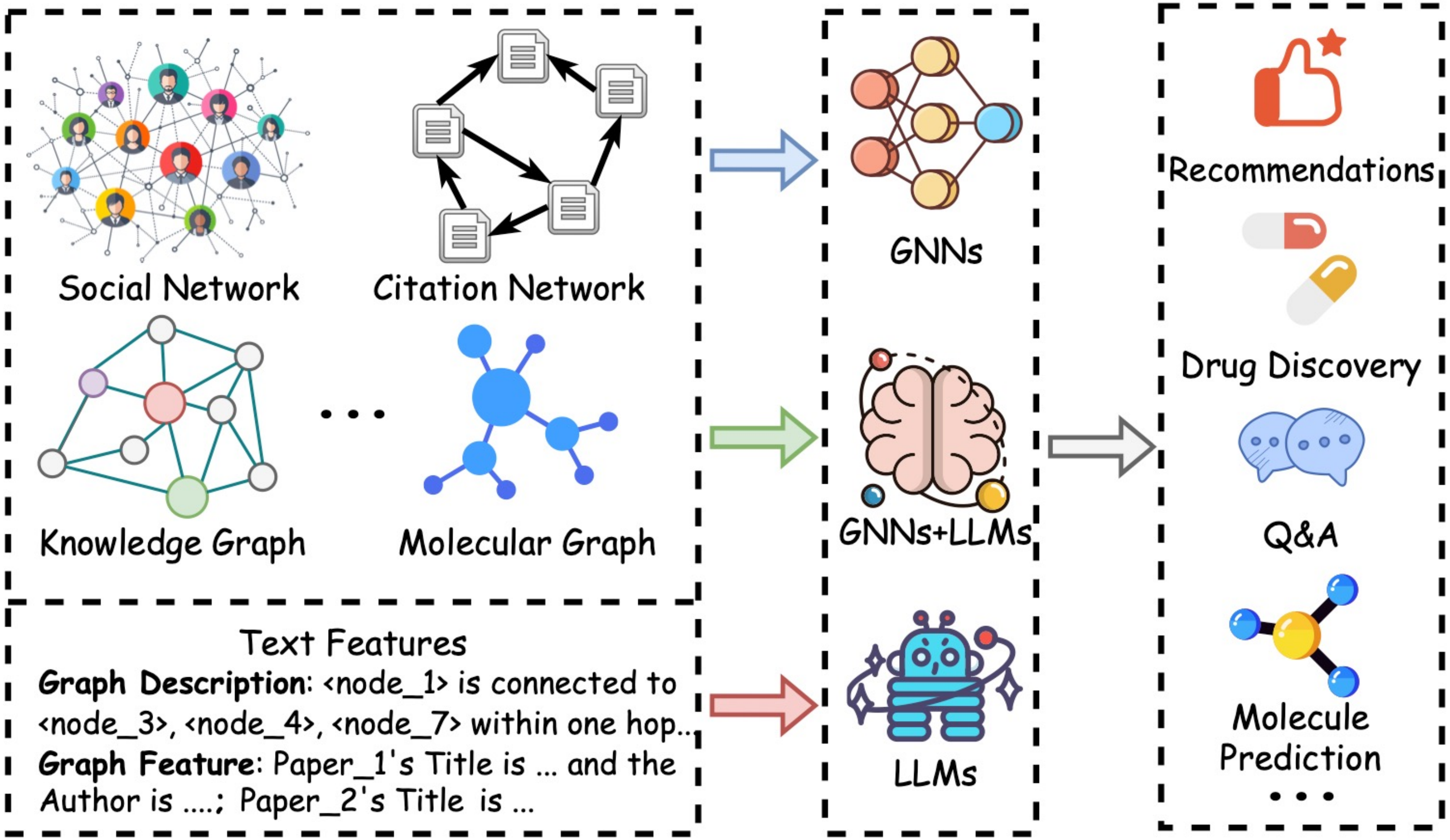


❑ **Curriculum & Teaching**, etc.



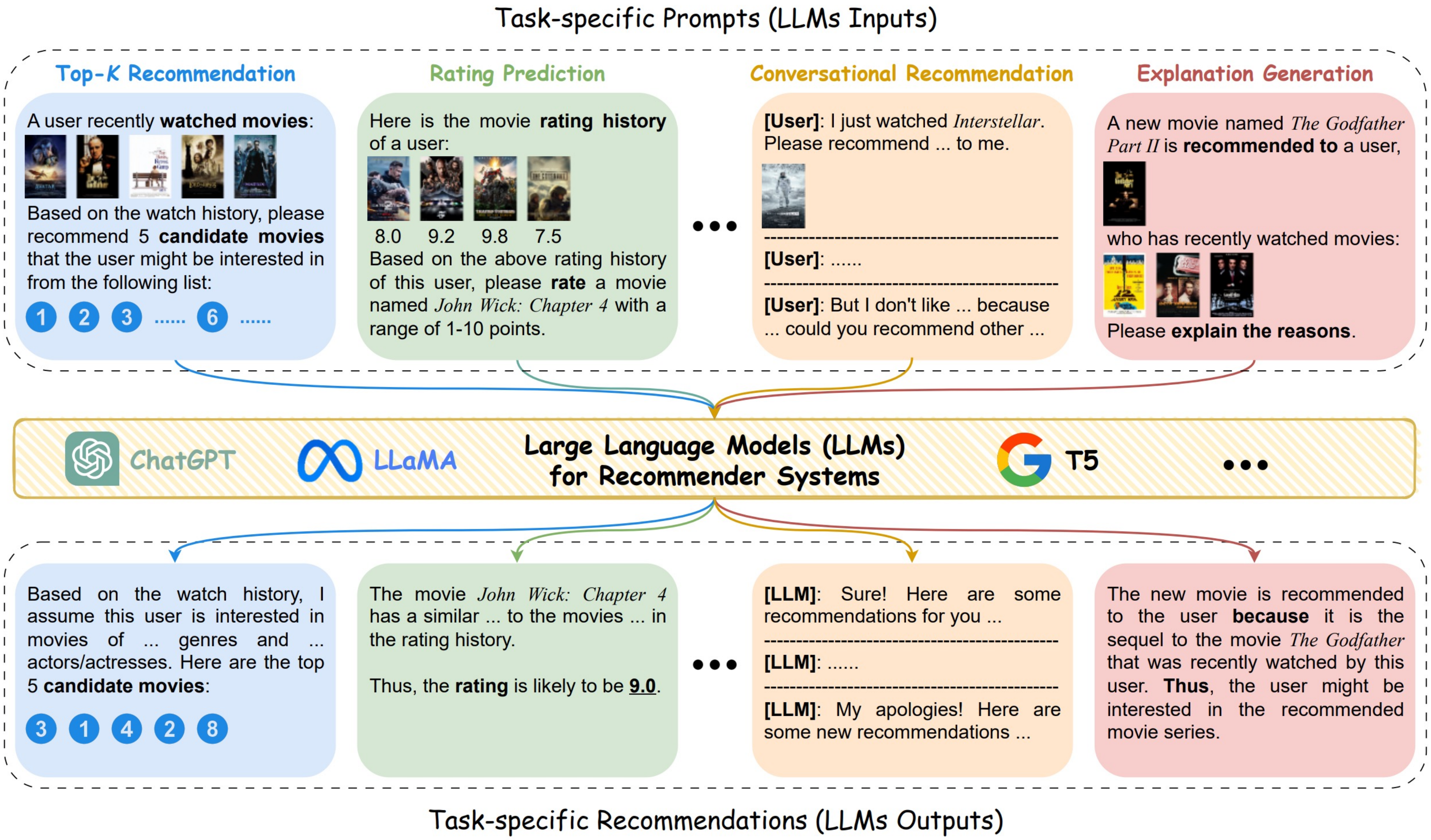
Zhang et al., 2023, HuatuoGPT, towards Taming Language Model to Be a Doctor

LLMs on Graph-structured Data



Fan, et al., 2024, Graph Machine Learning in the Era of Large Language Models (LLMs).

LLMs in Recommender Systems



Zhao, et al., 2024, Recommender systems in the era of large language models (llms).

Challenges and Risks of LLMs

Hallucination

The generation of inaccurate, nonsensical, or detached text, posing potential risks and challenges for organizations utilizing these models.

Domain-specific knowledge & expertise

LLMs might not perform well in many domain-specific fields like medicine, law, finance, and more, because of the lack of domain-specific knowledge and expertise.

Privacy

Various risks to data privacy and security exist at different stages of LLMs, which becomes particularly acute in light of incidents where sensitive internal data was exposed to LLMs.

Inconsistency

Sometimes they nail the answer to questions, other times they regurgitate random facts from their training data.

LLMs' Challenges in Vertical Domains

❑ Domain of Law

OXFORD

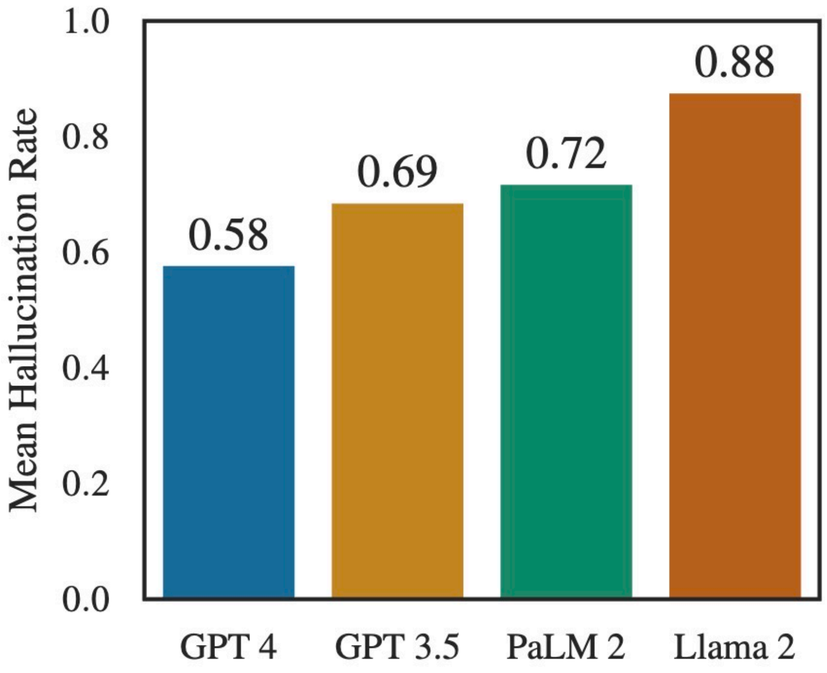
Journal of Legal Analysis, 2024, **16**, 64–93
https://doi.org/10.1093/jla/laae003
Advance access publication 26 June 2024
Article

Large Legal Fictions: Profiling Legal Hallucinations in Large Language Models

Matthew Dahl[*], Varun Magesh[†], Mirac Suzgun[‡], and Daniel E. Ho[§]

*In a new study by **Stanford RegLab** and **Institute for Human-Centered AI** researchers, it is demonstrated that legal hallucinations are pervasive and disturbing: **hallucination rates range from 69% to 88% in response to specific legal queries** for state-of-the-art language models.*

Hallucinations are common across all LLMs when they are asked a direct, verifiable question about a federal court case

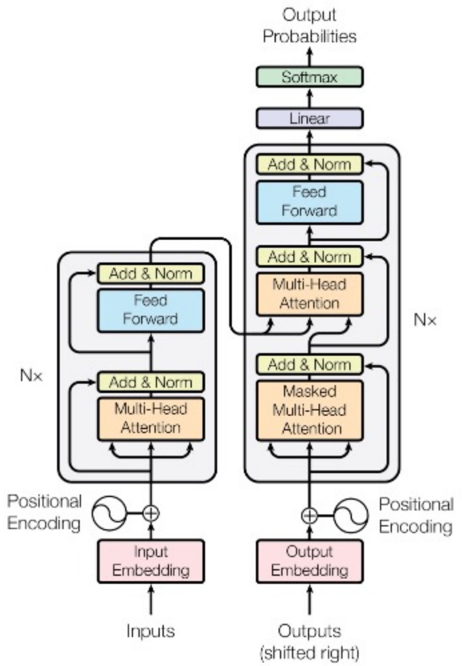


Dahl M, et al. 2024, Large legal fictions: Profiling legal hallucinations in large language models.

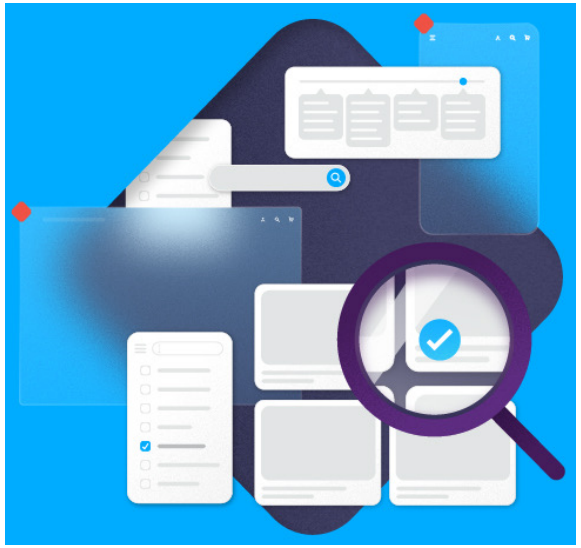
Why Large Language Models Work Well?

- Big Model + Big Training Data

Storing knowledge in the parametric model !



Storing knowledge in the non-parametric model?



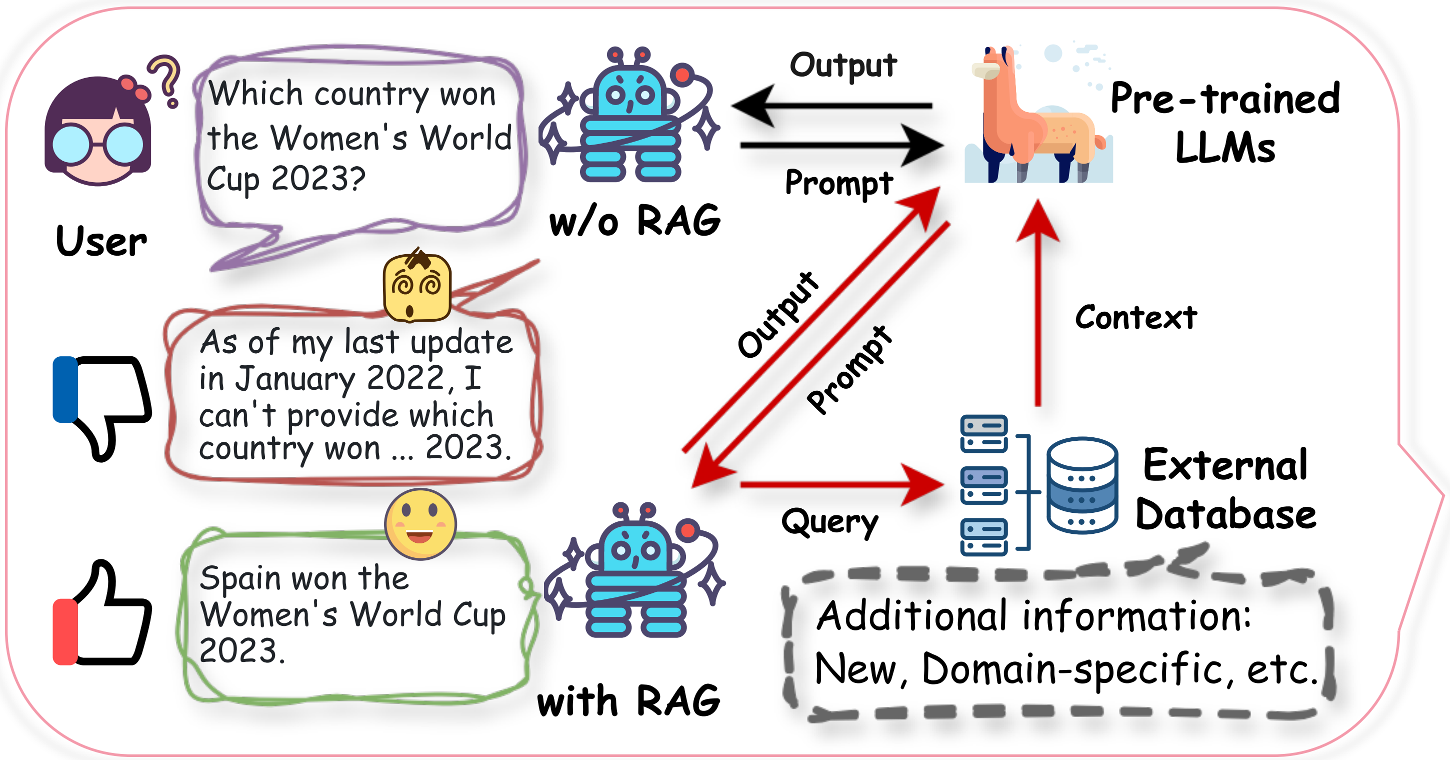
Information Retrieval (IR)

Retrieval-Augmented Large Language Models (RA-LLMs)

- LLMs **cannot memorize all** (particularly long-tail) knowledge in their parameters
- Lack of **domain-specific knowledge**, **updated information**, etc

Hallucination & Unable to answer → Re-training / Finetuning ?

Costly & Heavy Work

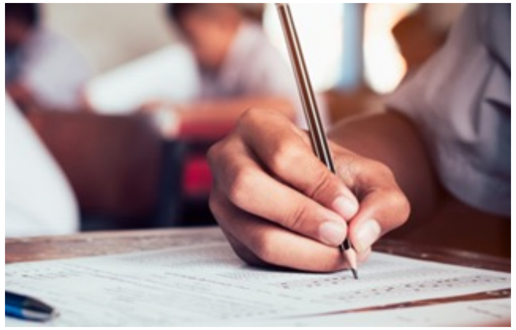


Retrieval-Augmented Generation (RAG) for LLM: **RA-LLMs**

Integrating Information Retrieval in Generation: **RA-LLM**

Data for Training LLMs
- Low quality
- General
- Fixed
- Hard to update

External Knowledge Base
- High-quality knowledge
- Specialized knowledge
- Scalable
- Easy-updated



Content generation

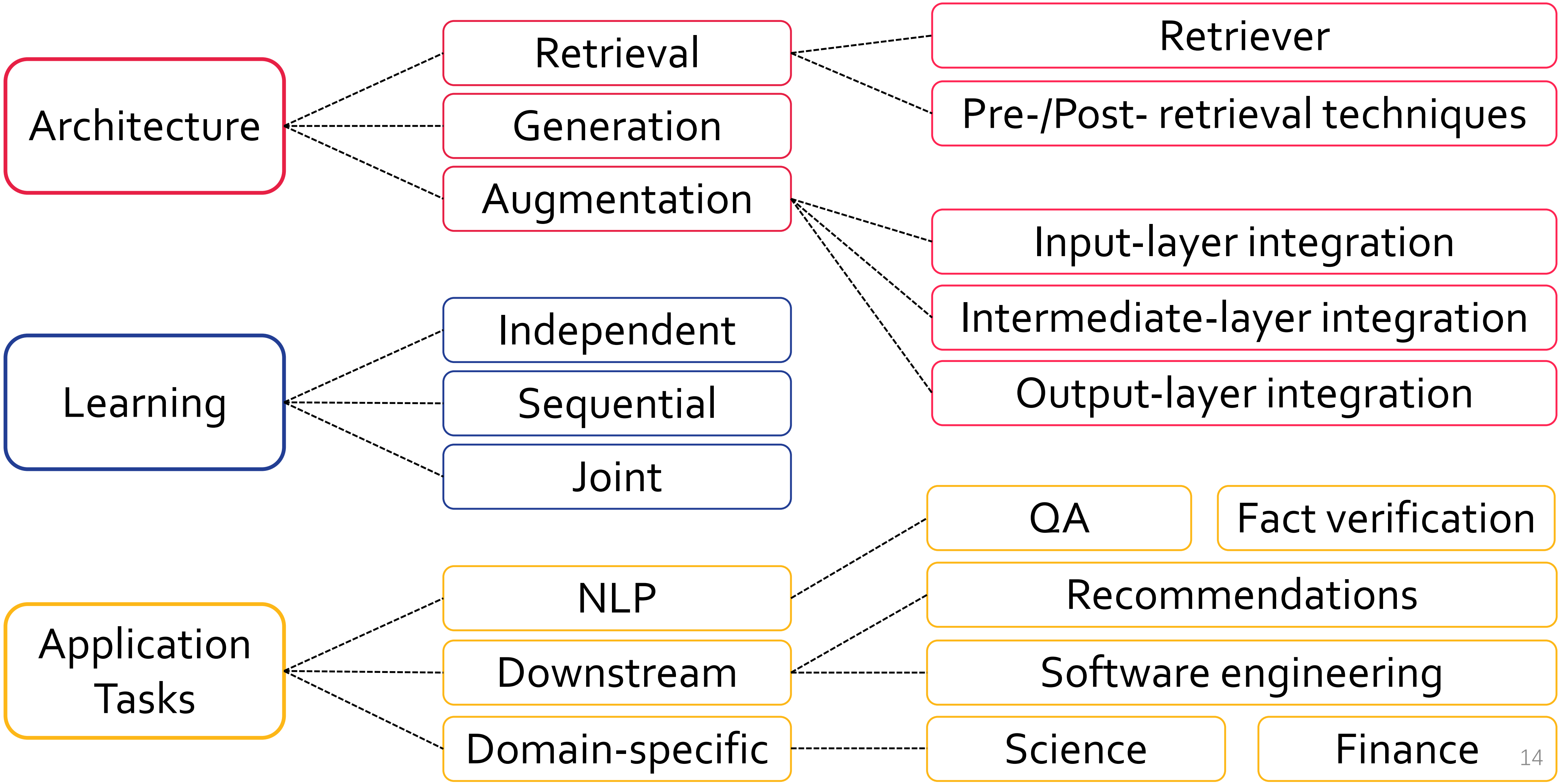
Close-book exam
(Hard mode, have to ***remember everything****)*

Information / Knowledge retrieval

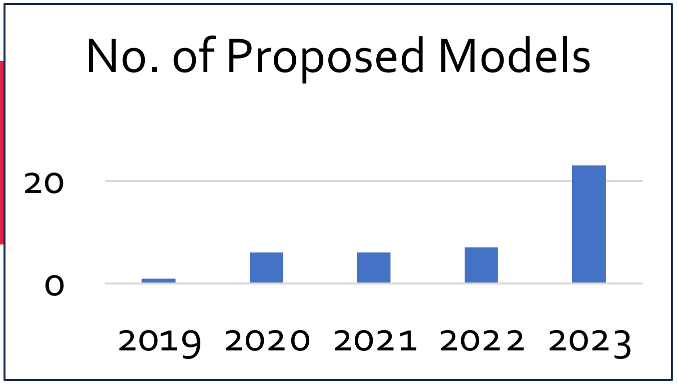
RA-LLMs

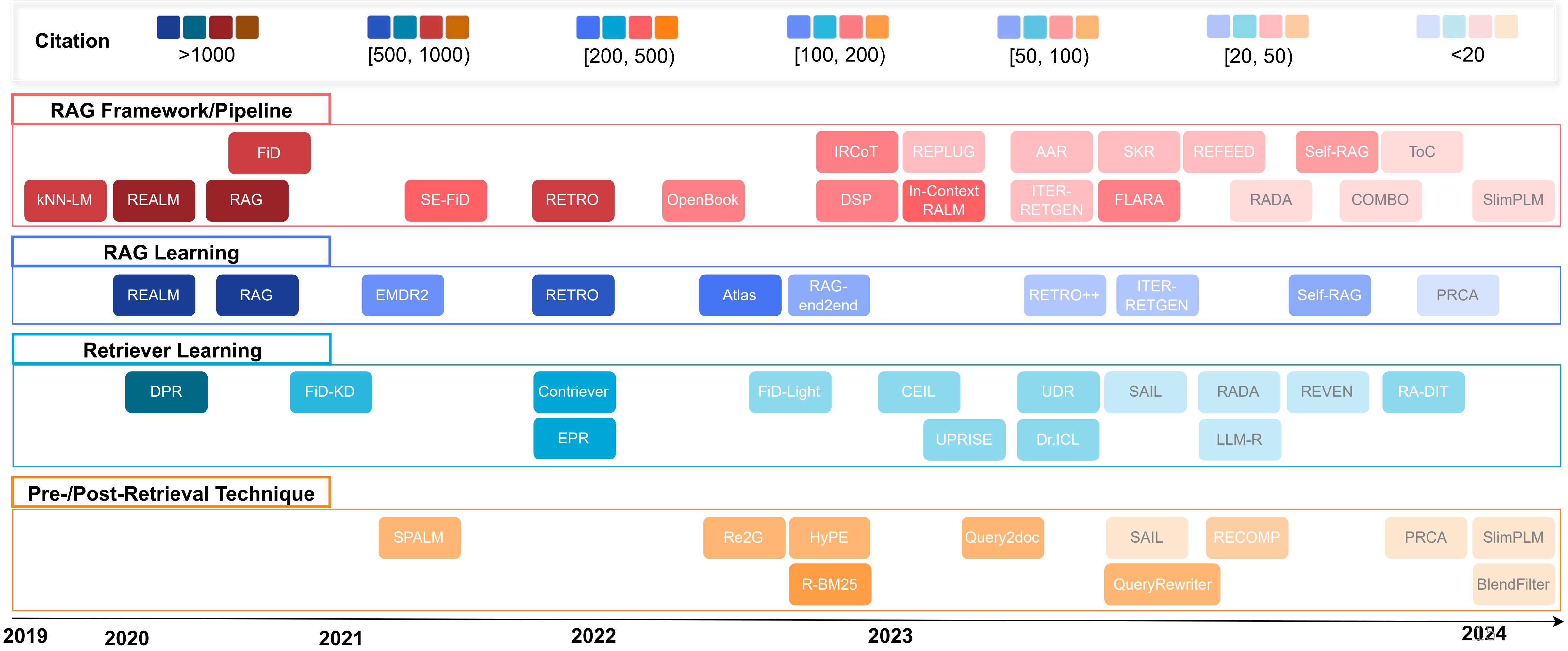
Open-book exam
(Easy mode, allow to search in reference)

RA-LLM Research Taxonomy



RAG & RA-LLM Model Development





A Comprehensive Survey Paper

A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models

Wenqi Fan
wenqifan03@gmail.com
The Hong Kong Polytechnic University, HK SAR

Yujuan Ding*
dingyujuan385@gmail.com
The Hong Kong Polytechnic University, HK SAR

Liangbo Ning
BigLemon1123@gmail.com
The Hong Kong Polytechnic University, HK SAR

Shijie Wang
shijie.wang@connect.polyu.hk
The Hong Kong Polytechnic University, HK SAR

Hengyun Li
neilhengyun.li@polyu.edu.hk
The Hong Kong Polytechnic University, HK SAR

Dawei Yin
yindawei@acm.org
Baidu Inc, China

Tat-Seng Chua
dcscts@nus.edu.sg
National University of Singapore, Singapore

Qing Li
csqli@comp.polyu.edu.hk
The Hong Kong Polytechnic University, HK SAR

Survey

Accepted by KDD'24

https://arxiv.org/pdf/2405.06211

Website of this tutorial
Check out the slides and more information!

Website

Recruitment

- Our research group (Prof. Qing LI & Dr. Wenqi FAN) is actively recruiting self-motivated **postdocs, Ph.D. students, research assistants**, etc. **Visiting scholars, interns, and self-funded students** are also welcome. Send us an email if you are interested.

  - Research areas: machine learning (ML), data mining (DM), artificial intelligence (AI), deep learning (DNNs), large language models (LLMs), graph neural networks (GNNs), computer vision (CV), natural language processing (NLP), etc.

  - Position details:
    https://wenqifan03.github.io/openings.html

Tutorial Outline

- **Part 1: Introduction** of Retrieval Augmented Large Language Models (RA-LLMs) (Dr. Wenqi Fan)
- **Part 2: Architecture of RA-LLMs and Main Modules (Dr. Yujuan Ding)**
- **Part 3: Learning** Approach of RA-LLMs (Liangbo Ning)
- **Part 4: Applications** of RA-LLMs (Shijie Wang)
- **Part 5: Challenges and Future Directions** of RA-LLMs (Dr. Wenqi Fan)

Website of this tutorial
Check out the slides and more information!

Website

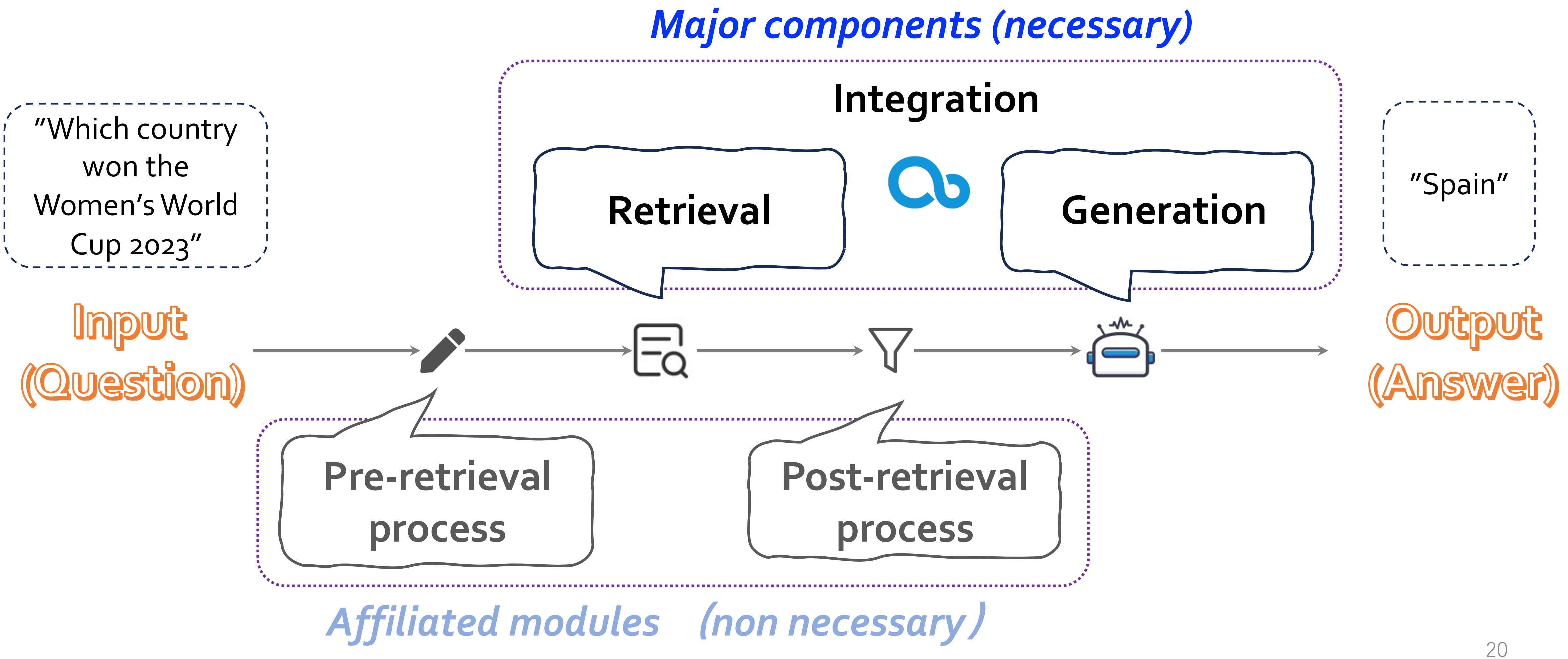
PART 2: Architecture of RA-LLMs and Main Modules

Presenter
Dr. Yujuan DING
HK PolyU

- **RA-LLM architecture overview**
- **Retriever in RA-LLMs**
- **Retrieval results integration**
- **Pre/Post-retrieval techniques**
- **Special RA-LLM paradigms**

RA-LLM Architecture: Standard Pipeline

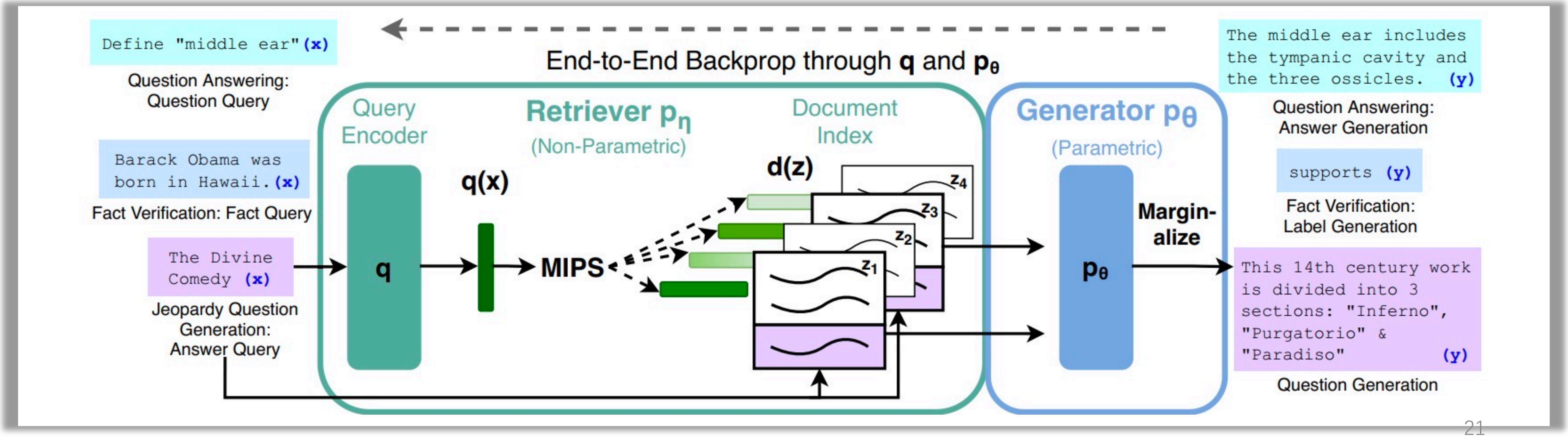
❑ **Technical component illustration in a RA-LLM for the Q&A task**



A Simple Retrieval-Augmented Generation Model

- **RAG**

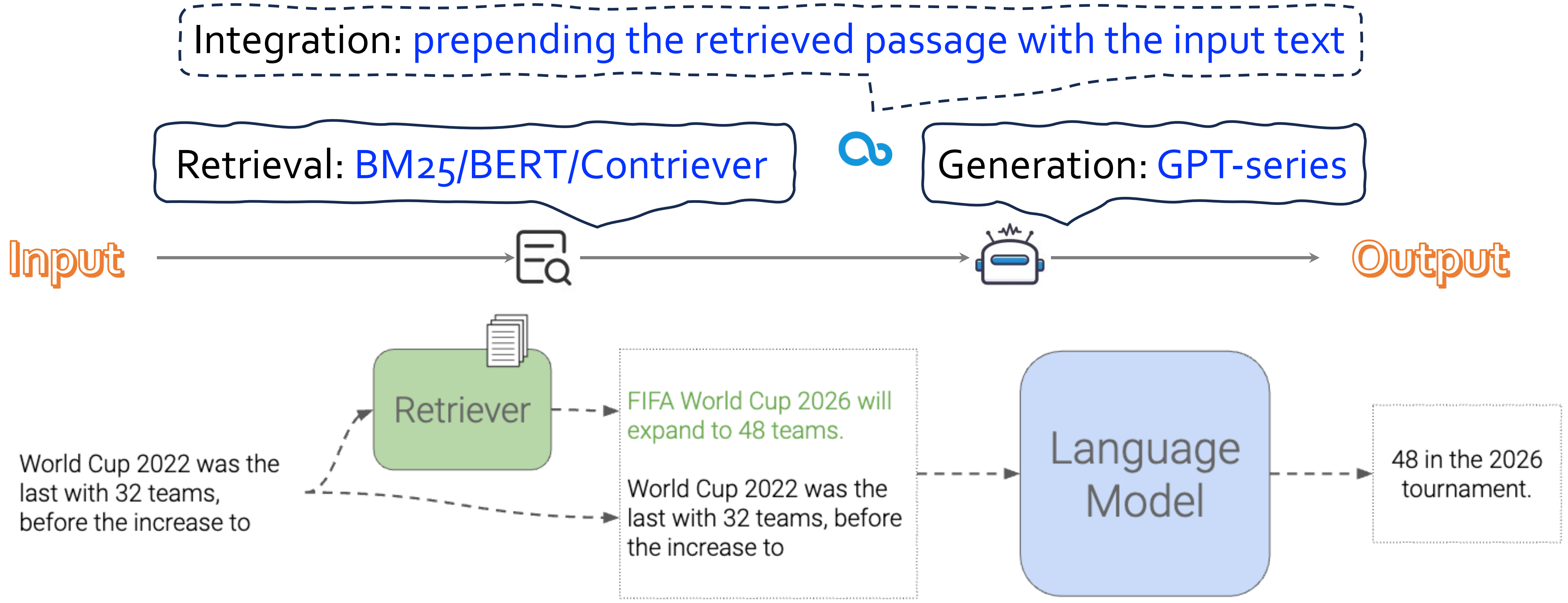
Integration: concatenating each retrieved passage with the question



Lewis et al. 2020. "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks"

A Simple Retrieval-Augmented Generation Model

- **In-Context RALM**



Ram et al. 2023, In-Context Retrieval-Augmented Language Models

PART 2: Architecture of RA-LLMs and Main Modules

Slides

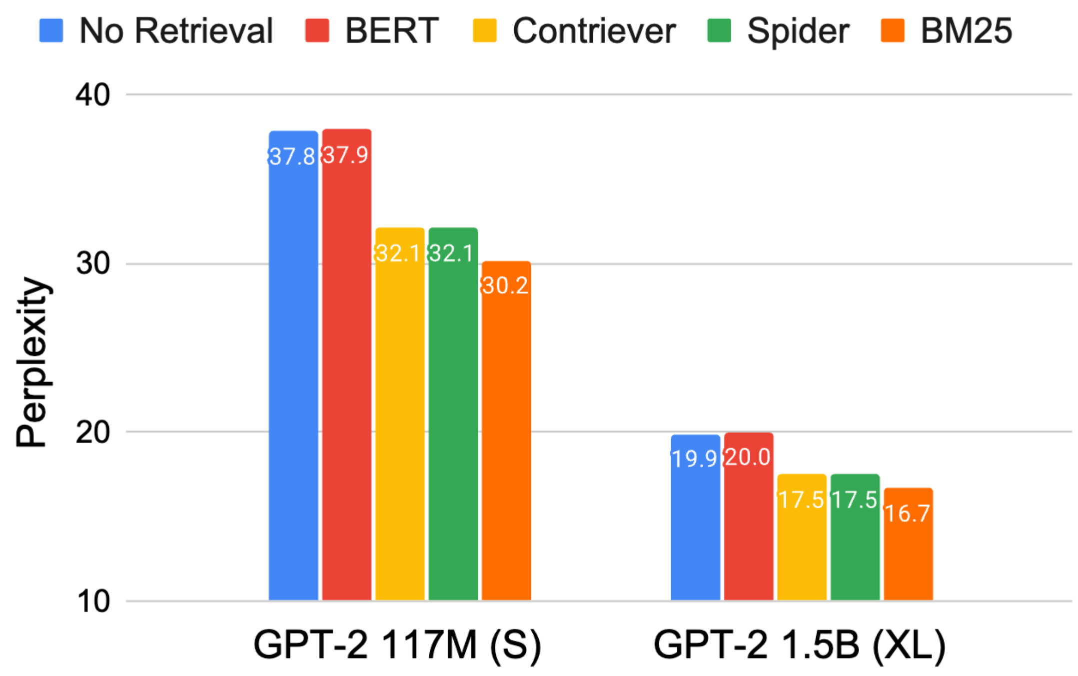
Website

Website of this tutorial

- RA-LLM architecture overview
- Retriever in RA-LLMs
- Retrieval results integration
- Pre/Post-retrieval techniques
- Special RA-LLM paradigms

RA-LLM Architecture: Retriever Types

- Different types of retriever deliver different generation performance



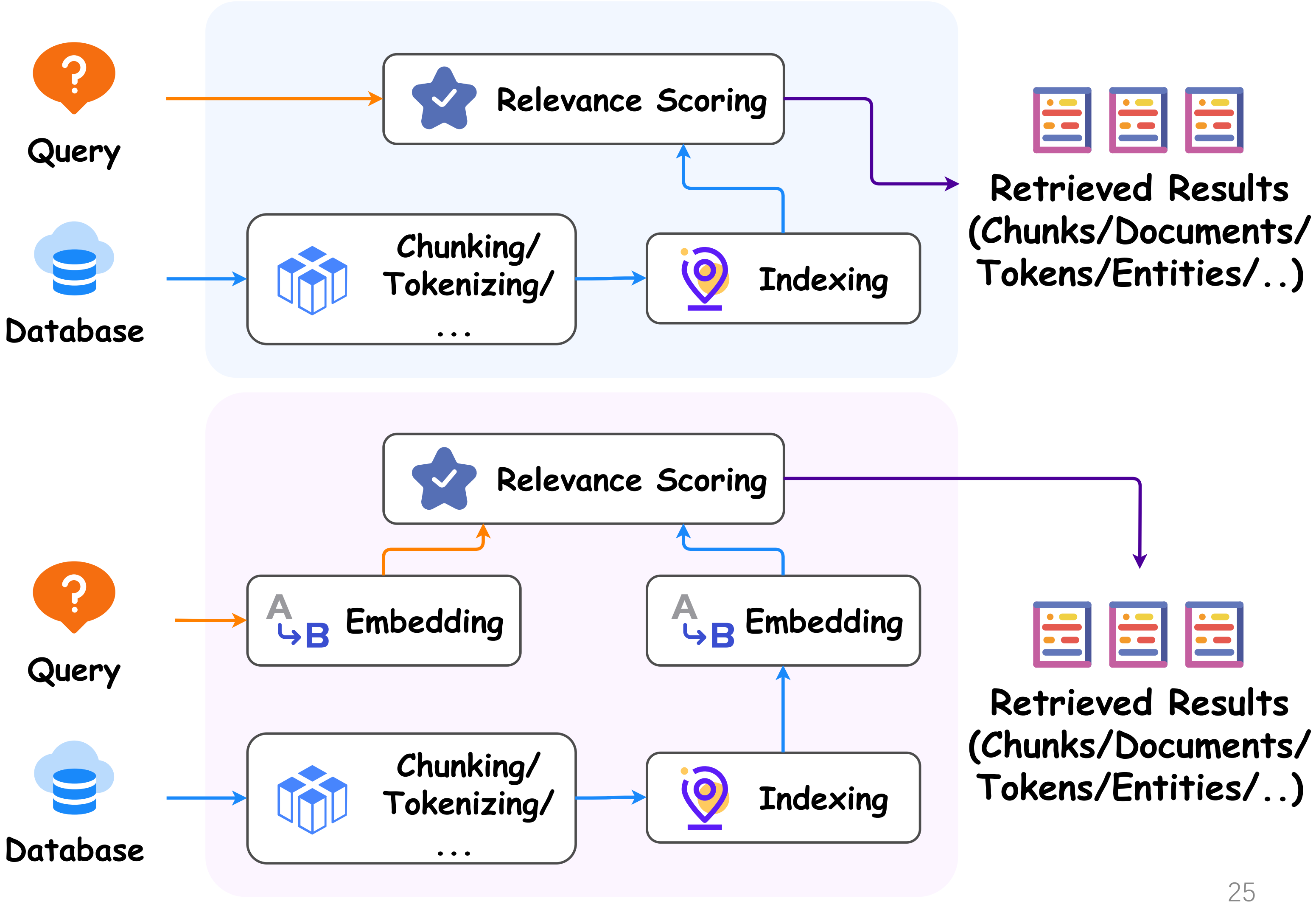
Relevance measurement	Retriever learning
Sparse	Task-specific pre-trained
Dense	General-purpose pre-trained

Ram et al. 2023, In-Context Retrieval-Augmented Language Models

Dense v.s. Sparse Retrievers

Sparse Retrievers (SR)

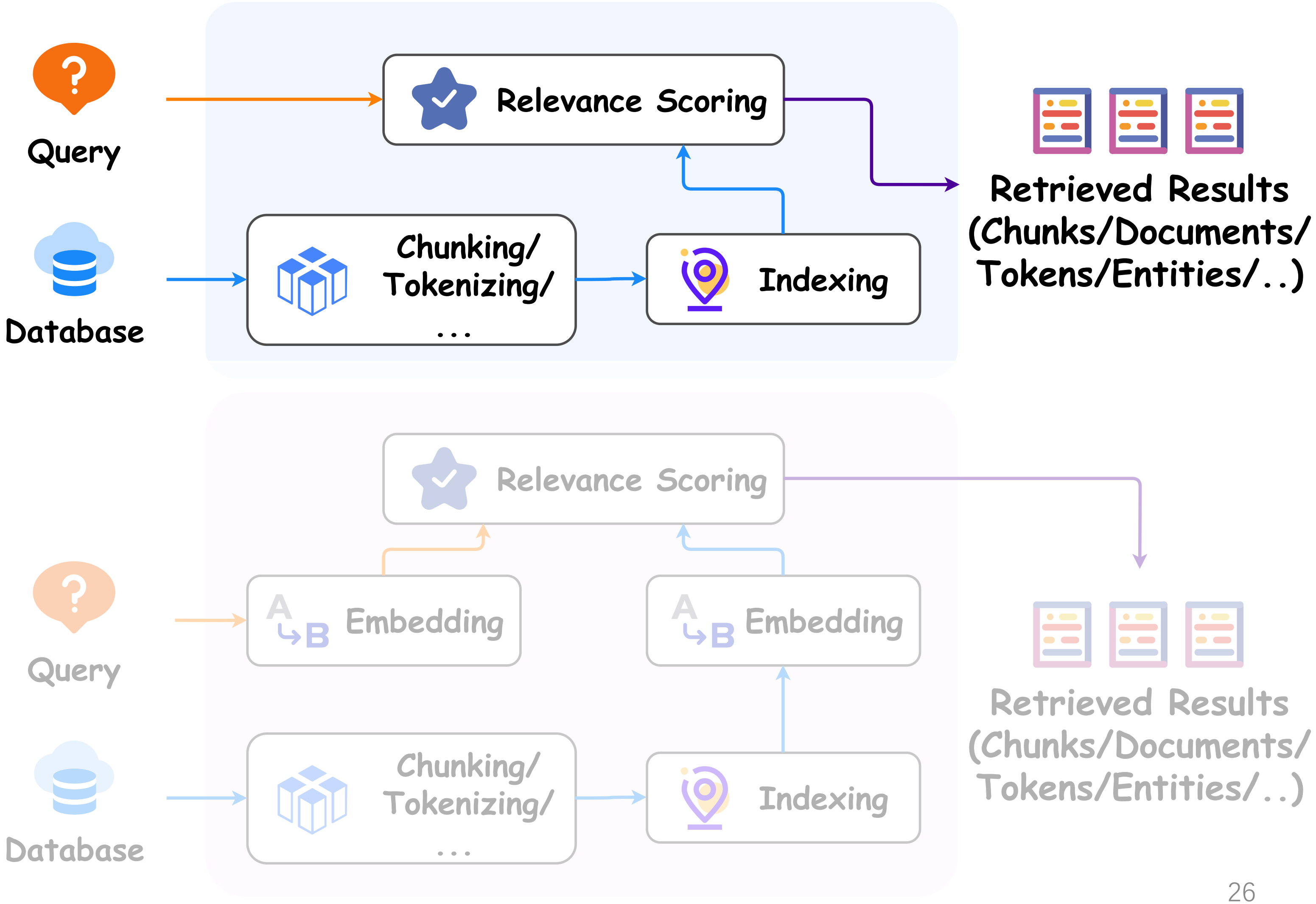
Dense Retrievers (DR)



Dense v.s. Sparse Retrievers

Sparse Retrievers (SR)

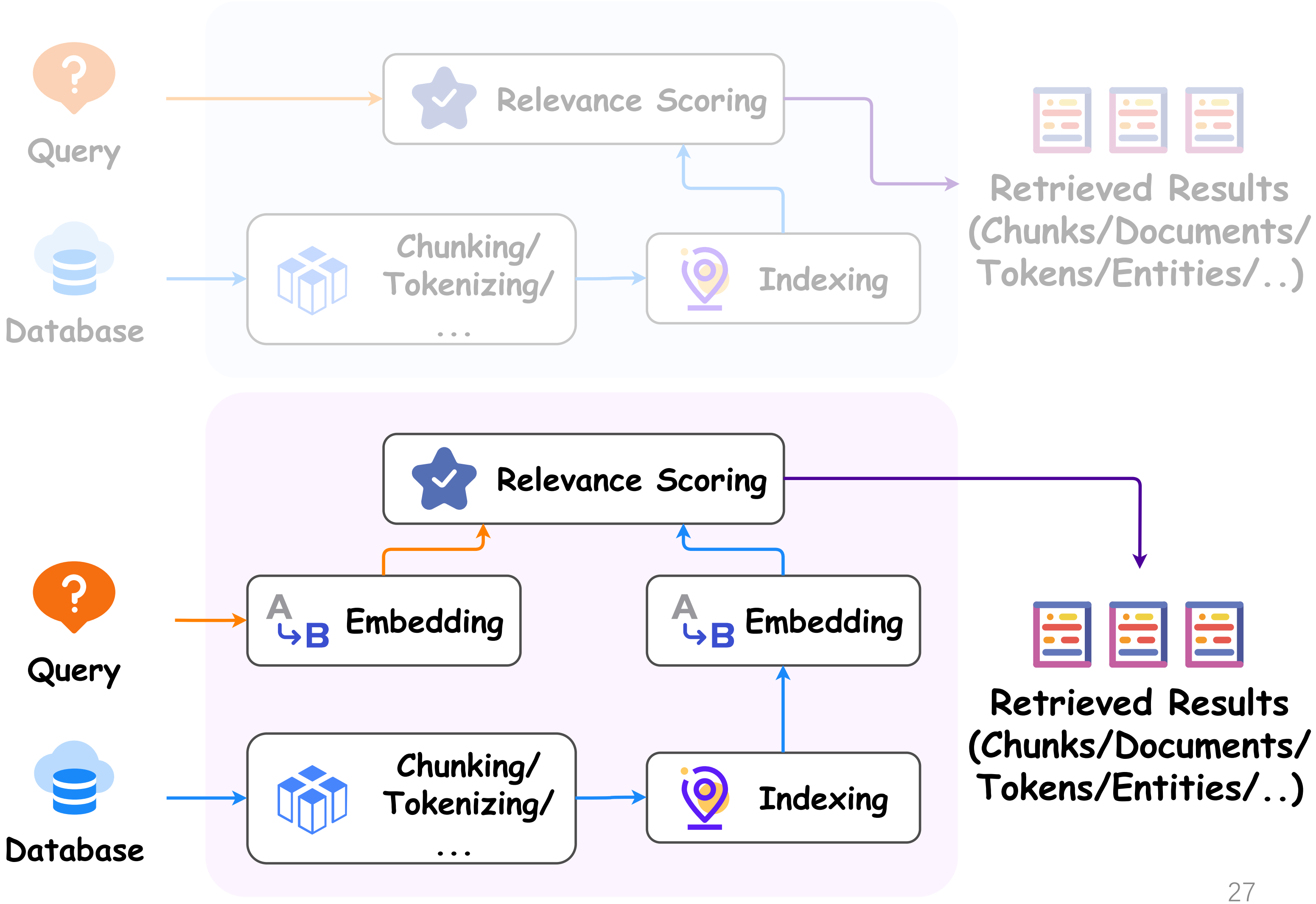
- Feasible to apply
- High efficiency
- Fine performance
- Example: TF-IDF, BM25



Dense v.s. Sparse Retrievers

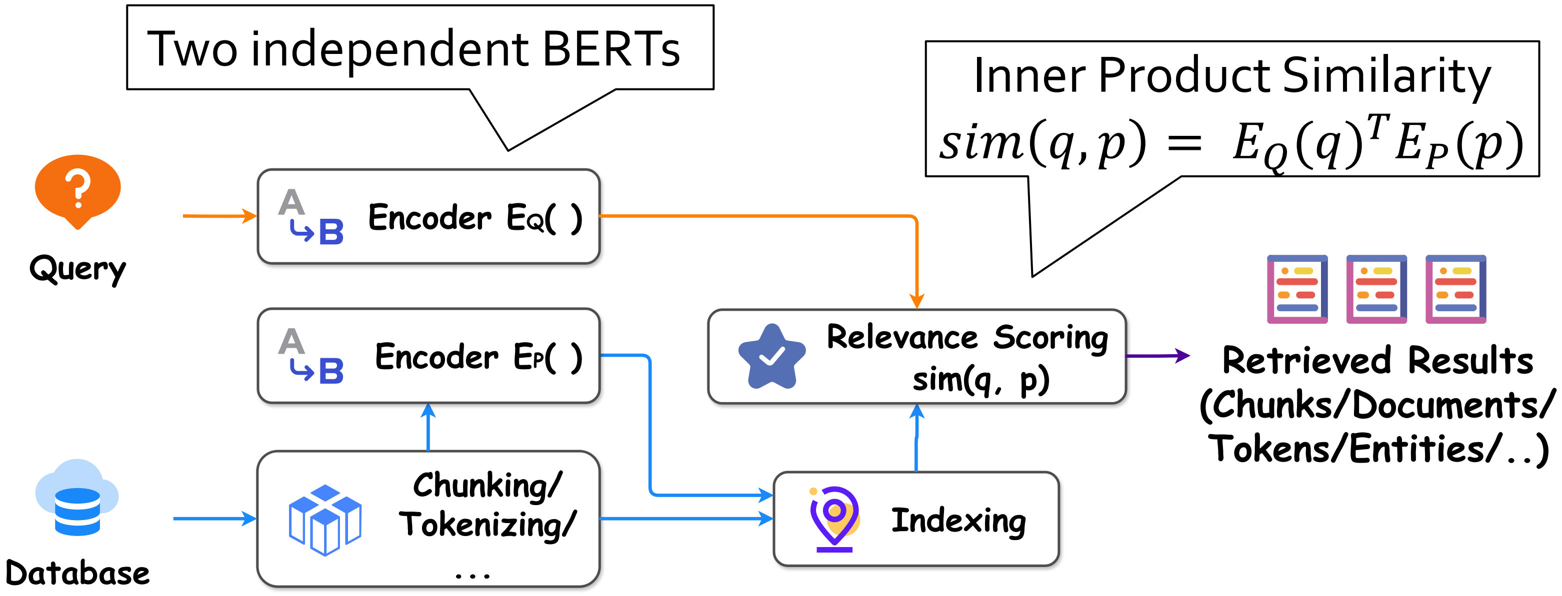
Dense Retrievers (DR)

- Allowing fine-tuning
- Better adaptation
- Customizable for more retrieval goals
- Example: DPR, Contriever



Task-Specific Pre-trained Retriever (Supervised)

- **Dense Passage Retriever (DPR):** Pretrained for Question Answering (QA)



Karpukhin et al. 2020. “Dense Passage Retrieval for Open-Domain Question Answering”

Task-Specific Pre-trained Retriever (Supervised)

- Dense Passage Retriever (DPR): **Pretrained for Question Answering (QA)**

- Learning Objective

$$L(q_i, p_i^+, p_{i,1}^-, \cdots, p_{i,n}^-)$$

$$= -\log \frac{e^{\mathrm{sim}(q_i, p_i^+)}}{e^{\mathrm{sim}(q_i, p_i^+)} + \sum_{j=1}^{n} e^{\mathrm{sim}(q_i, p_{i,j}^-)}}$$

- Training data: Question-Passage Sets

$$\mathcal{D} = \{\langle \boxed{q_i}, \boxed{p_i^+}, \boxed{p_{i,1}^-, \cdots, p_{i,n}^-} \rangle\}_{i=1}^{m}$$

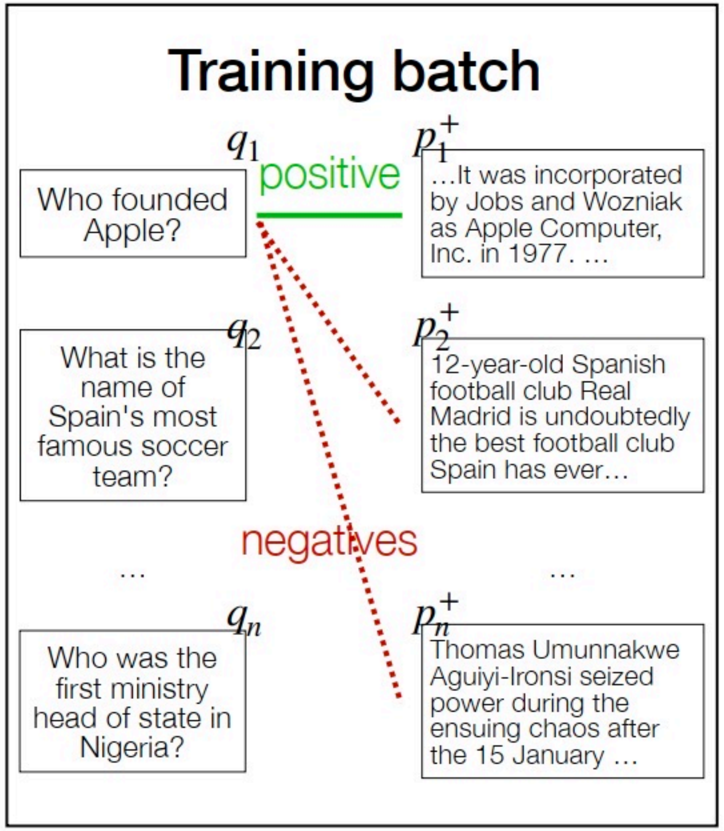
Question

Irrelevant passages

Relevant passage

Negative sample selection?

- Training with in-batch negatives



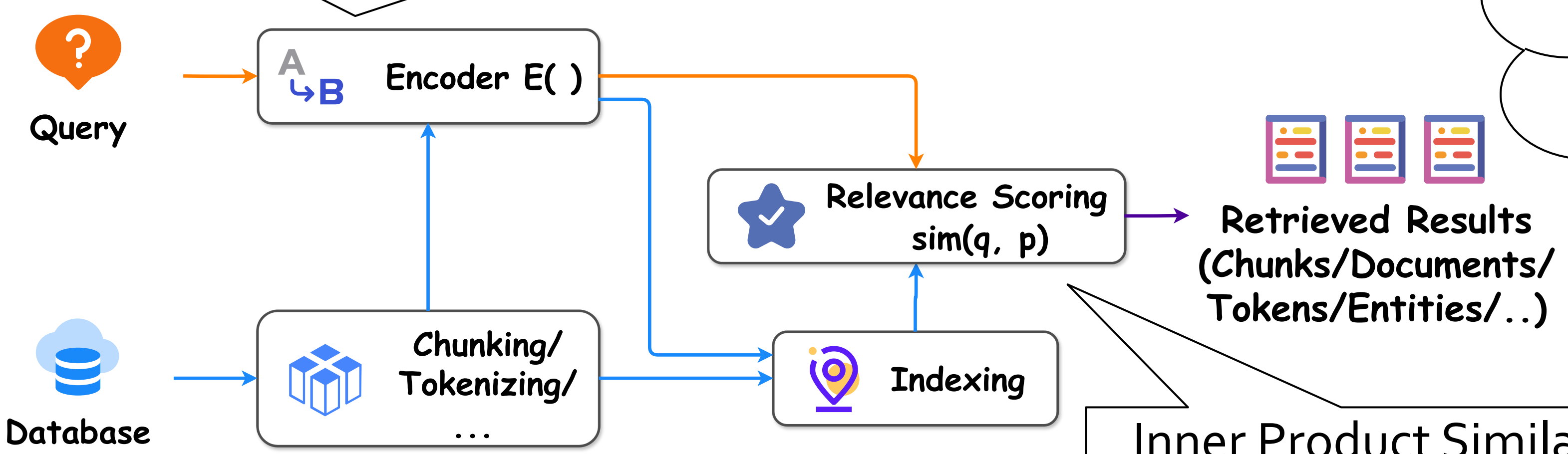
Karpukhin et al. 2020. "Dense Passage Retrieval for Open-Domain Question Answering"

General-Purpose Pre-trained Retriever (Unsupervised)

- **Contriever:** Pre-trained with unsupervised learning

Same BERT encoder for queries and documents

Might improve robustness in the context of zero-shot transfer or few-shot learning, while having no impact on other settings



Inner Product Similarity
$sim(q,p) = E(q)^T E(p)$

Contrastive learning with unaligned documents

$$\mathcal{L}(q, k_+) = -\frac{\exp(s(q,k_+)/\tau)}{\exp(s(q,k_+)/\tau) + \sum_{i=1}^{K} \exp(s(q,k_i)/\tau)}$$

Izacard et al. 2022. “Unsupervised Dense Information Retrieval with Contrastive Learning”

DPR & Contriever Performance on OpenQA Tasks

End-to-end QA (Exact Match) Accuracy

Training	Model	NQ	TriviaQA	WQ	TREC	SQuAD
Single	BM25+BERT (Lee et al., 2019)	26.5	47.1	17.7	21.3	33.2
Single	ORQA (Lee et al., 2019)	33.3	45.0	36.4	30.1	20.2
Single	HardEM (Min et al., 2019a)	28.1	50.9	-	-	-
Single	GraphRetriever (Min et al., 2019b)	34.5	56.0	36.4	-	-
Single	PathRetriever (Asai et al., 2020)	32.6	-	-	-	**56.5**
Single	$\text{REALM}_{\text{Wiki}}$ (Guu et al., 2020)	39.2	-	40.2	46.8	-
Single	$\text{REALM}_{\text{News}}$ (Guu et al., 2020)	40.4	-	40.7	42.9	-
	BM25	32.6	52.4	29.9	24.9	38.1
Single	DPR	**41.5**	56.8	34.6	25.9	29.8
	BM25+DPR	39.0	57.0	35.2	28.0	36.7
Multi	DPR	**41.5**	56.8	**42.4**	49.4	24.1
	BM25+DPR	38.8	**57.9**	41.1	**50.6**	35.8

Both widely applied in RAG and RA-LLMs

DPR in

RAG, FiD, RETRO, EPR, UDR, …

Contriever in

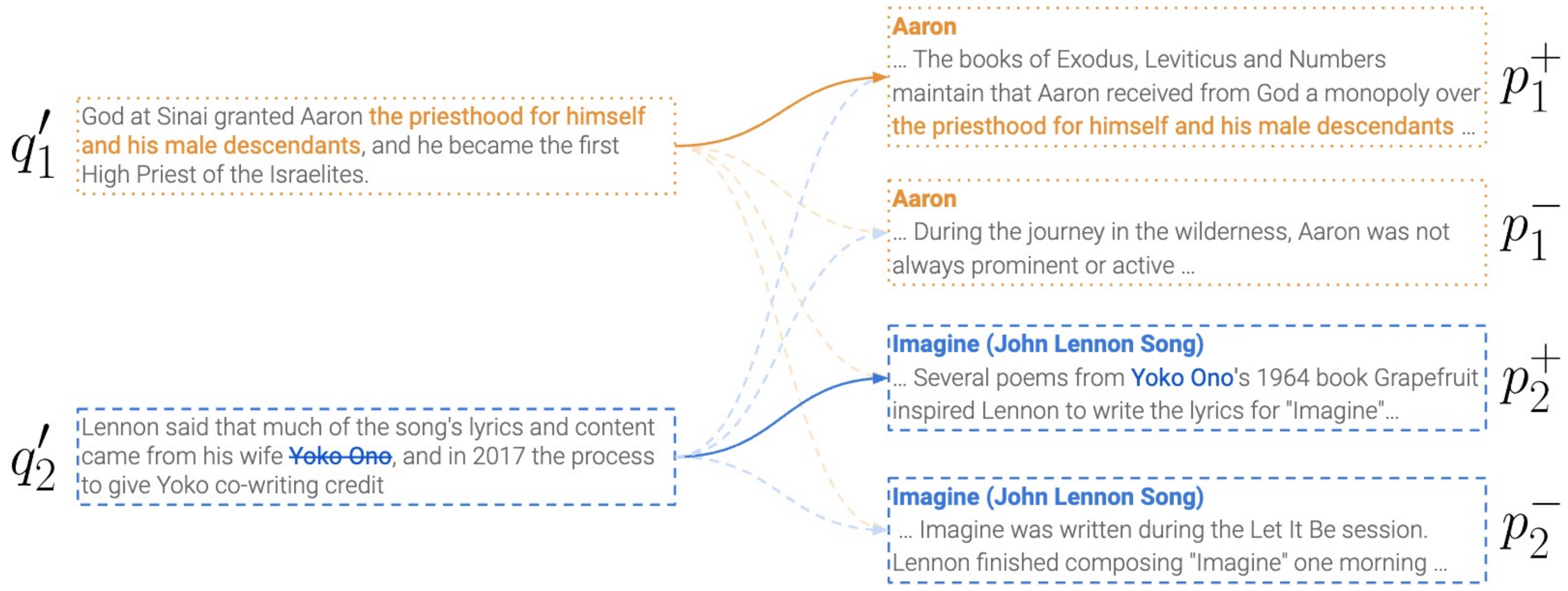
Self-RAG, Atlas, RAVEN, …

Both better than the sparse retriever!

	NaturalQuestions			TriviaQA		
	R@5	R@20	R@100	R@5	R@20	R@100
Inverse Cloze Task (Sachan et al., 2021)	32.3	50.9	66.8	40.2	57.5	73.6
Masked salient spans (Sachan et al., 2021)	41.7	59.8	74.9	53.3	68.2	79.4
BM25 (Ma et al., 2021)	-	62.9	78.3	-	**76.4**	**83.2**
Contriever	**47.8**	**67.8**	**82.1**	**59.4**	74.2	**83.2**
supervised model: DPR (Karpukhin et al., 2020)	-	78.4	85.4	-	79.4	85.0

Task-Specific Pre-trained Retriever (Unsupervised)

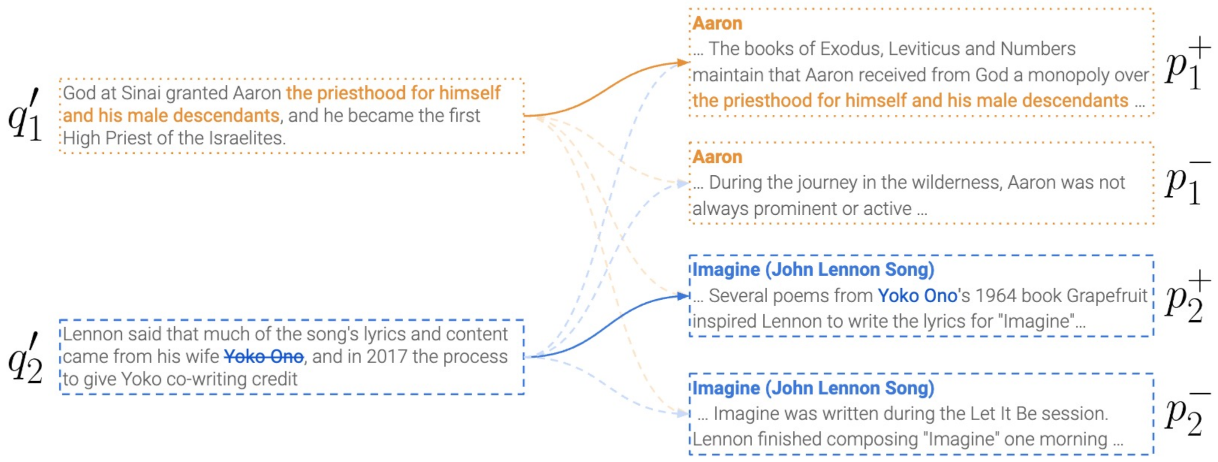
- **Spider** (Span-based unsupervised dense retriever)
  - **Recurring Span Retrieval**: It is based on the notion of recurring spans within a document: given two paragraphs with the same recurring span, we construct a query from one of the paragraphs, while the other is taken as the target for retrieval



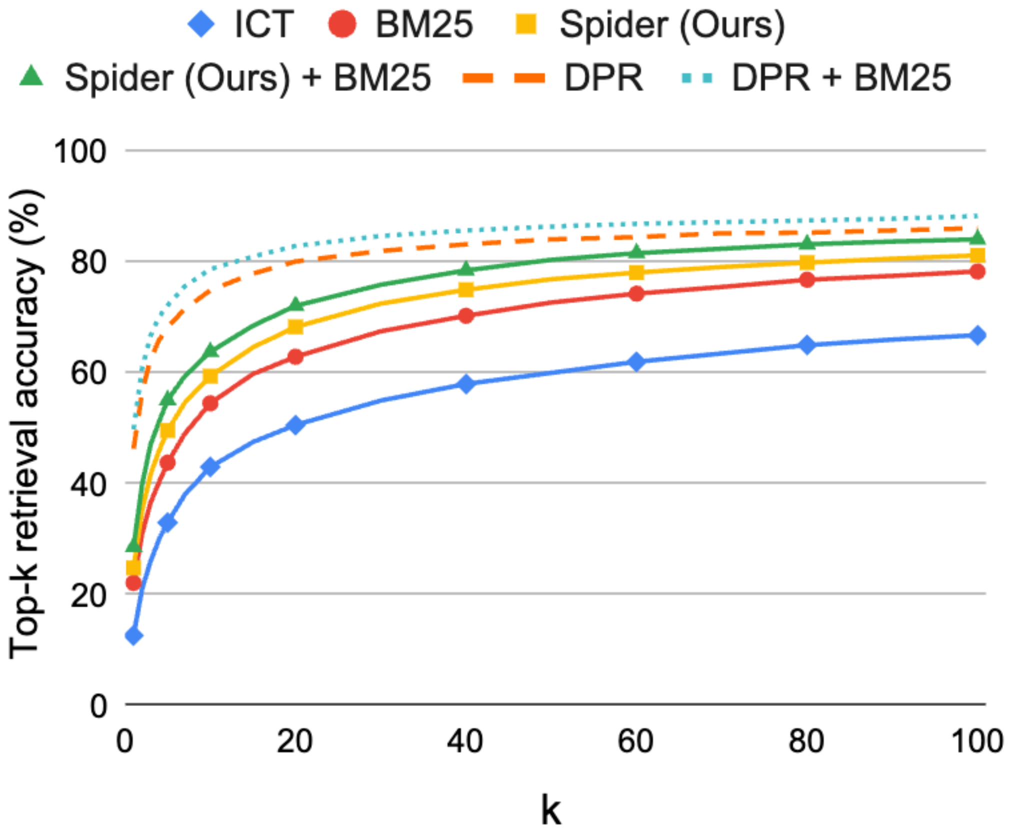
Ram et al., 2022, Learning to Retrieve Passages without Supervision

Task-Specific Pre-trained Retriever (Unsupervised)

- Learning and results of Spider

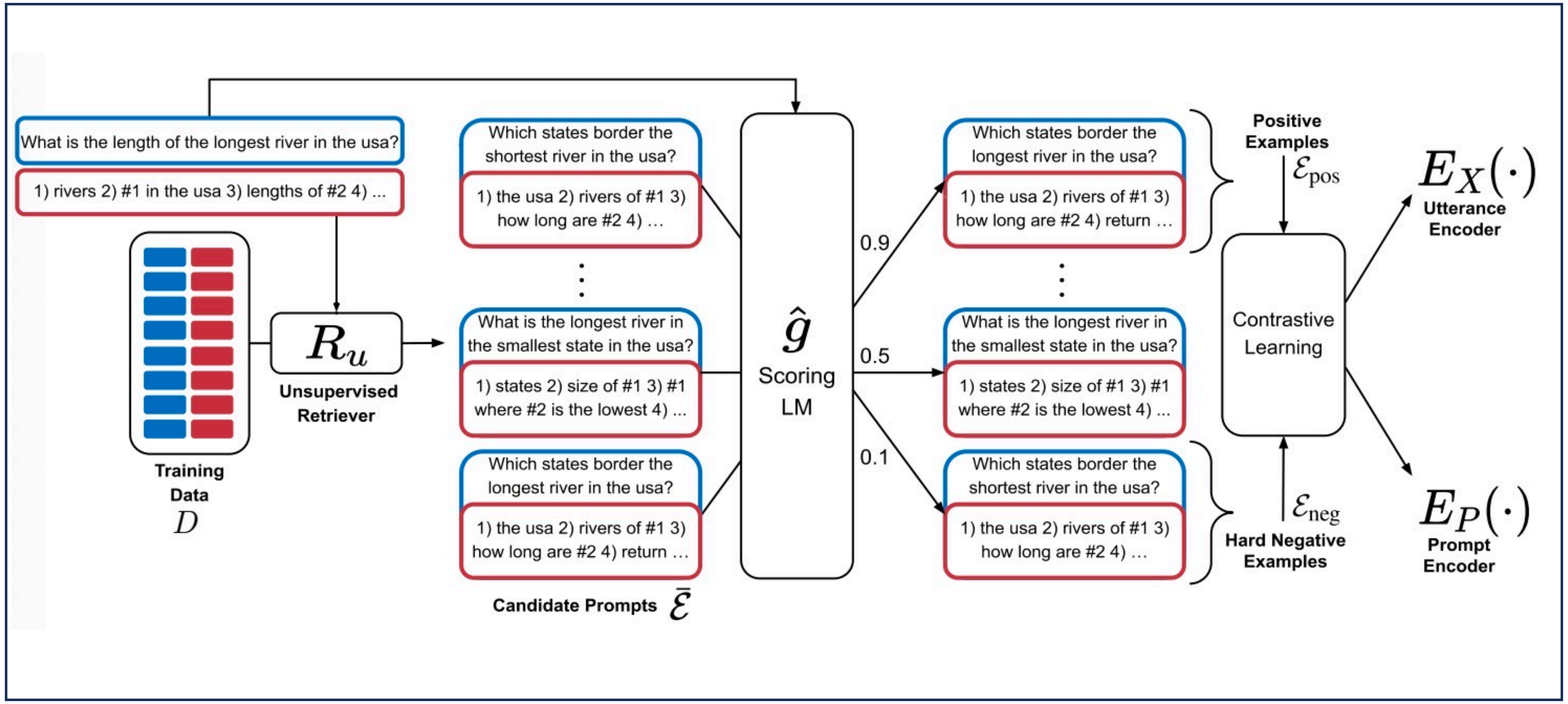


$$-\log \frac{\exp\left(s(q_i', p_i^+)\right)}{\sum_{j=1}^{m}\left(\exp\left(s(q_i', p_j^+)\right) + \exp\left(s(q_i', p_j^-)\right)\right)}$$

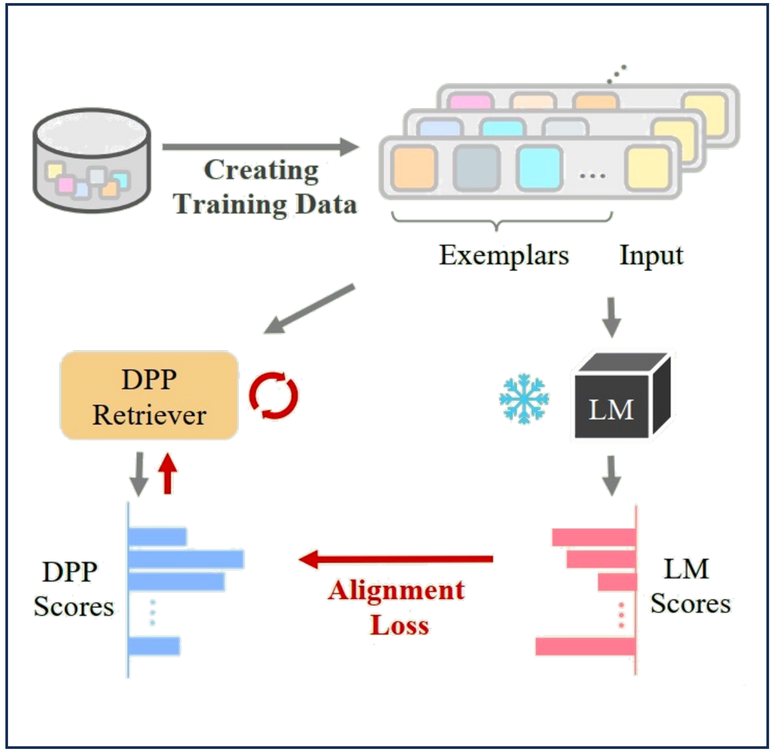


Retrievers for In-Context Learning of LLMs

❑ Prompt Retriever



❑ Examplar Retriever (CEIL)



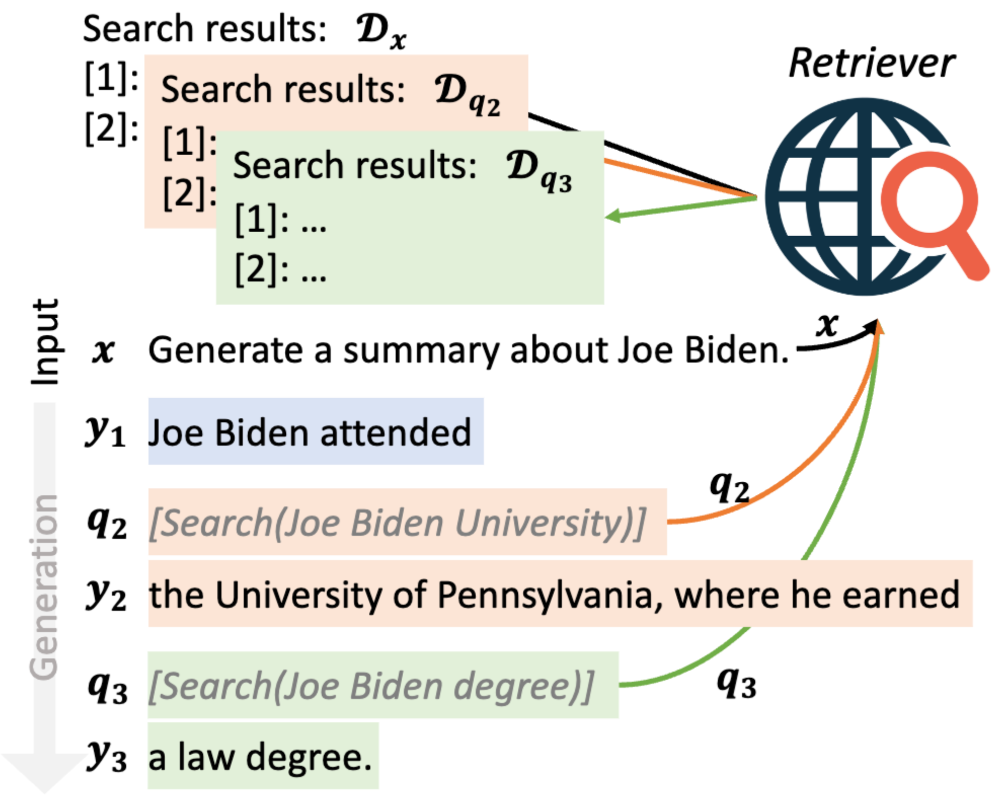
Rubin et al. 2023. "Learning To Retrieve Prompts for In-Context Learning"
Ye et al. 2023. "Compositional Exemplars for In-context Learning"

Search Engine as Retrievers

Google Bing

❑ Traditional retrieval methods

- ❖ May be difficult to update to real-time web documents
- ❖ May be a limit to the number of documents storable in the pre-defined database
- ❖ Will not take advantage of the high quality ranking that has been finely tuned in Internet Search engines over decades of use



Komeili et al., 2021, Internet-Augmented Dialogue Generation
Jiang et al. 2023, Active Retrieval Augmented Generation

PART 2: Architecture of RA-LLMs and Main Modules

Slides

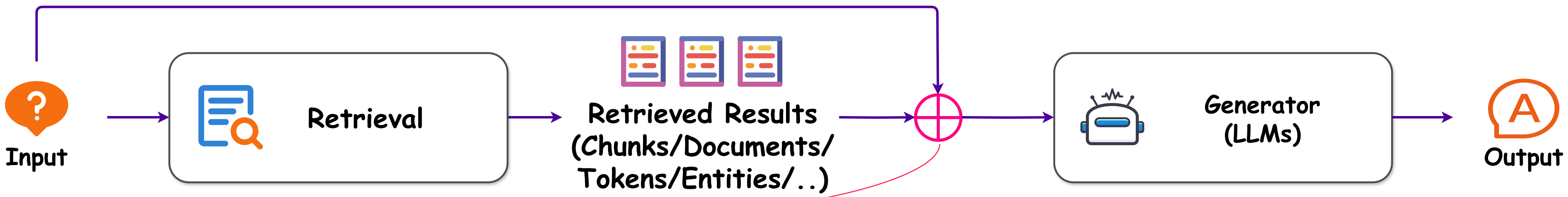
Website

Website of this tutorial

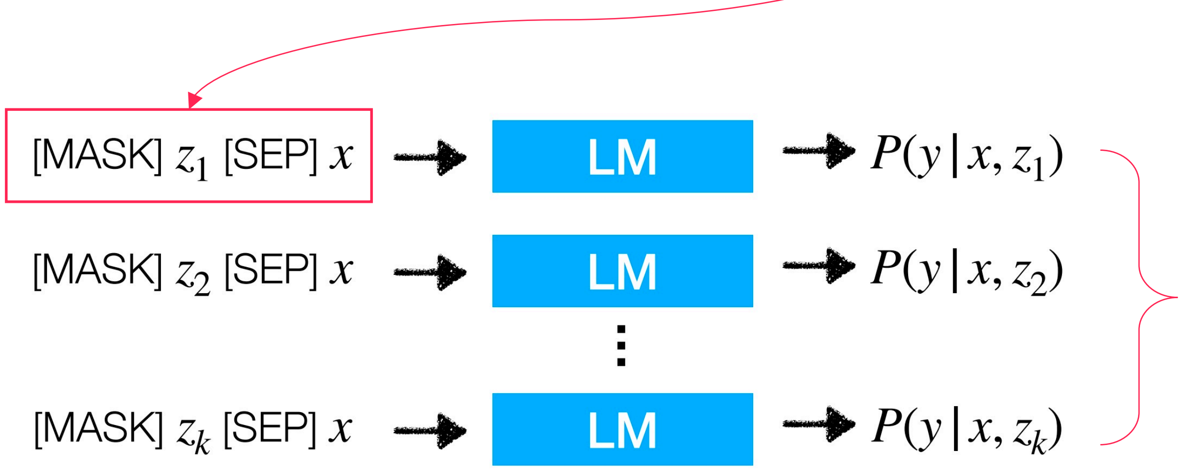
- **RA-LLM architecture overview**
- **Retriever in RA-LLMs**
- **Retrieval results integration**
- **Pre/Post-retrieval techniques**
- **Special RA-LLM paradigms**

Retrieved Results Integration: Input-layer Integration

❑ REALM



Integrating the retrieved passage z and x the original input



Weighted aggregating the prediction results based on all retrieved passages

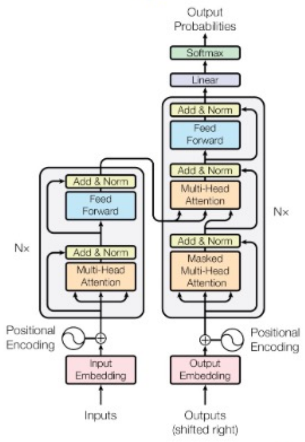
$$\sum_{z \in \mathcal{D}} P(z \mid x) P(y \mid x, z)$$

Guu et al. 2020. "REALM: Retrieval-Augmented Language Model Pre-Training"

Retrieval-Augmented Generator

Typical encoder: $p(y|x)$

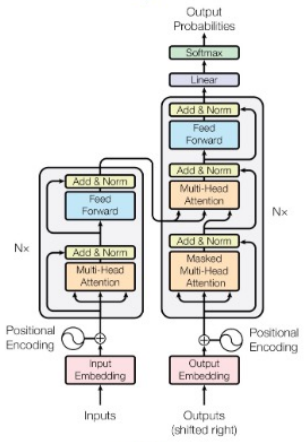
y= pounds



x: we paid 20 __ at the Buckingham Palace gift shop

Knowledge-augmented encoder:$p(y|x, z)$

y= pounds

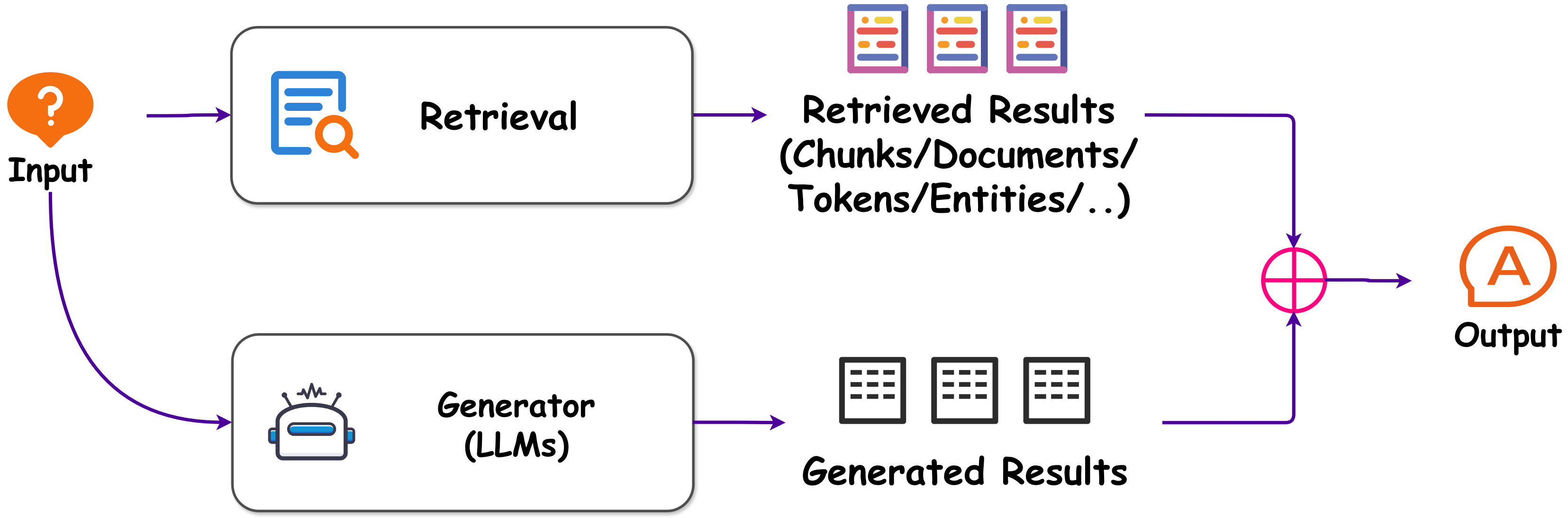


x: we paid 20 __ at the Buckingham Palace gift shop

z: Buckingham Palace is home to the British monarchy

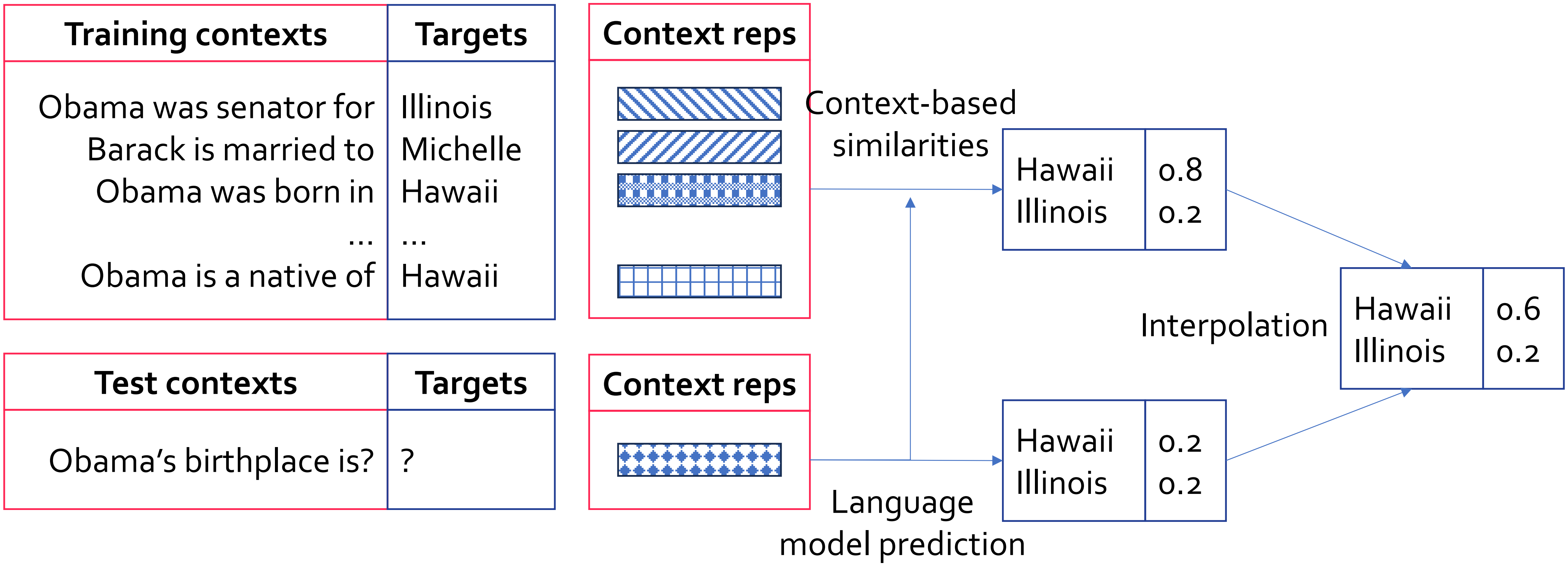
explicit knowledge

Retrieved Results Integration: Output-layer integration



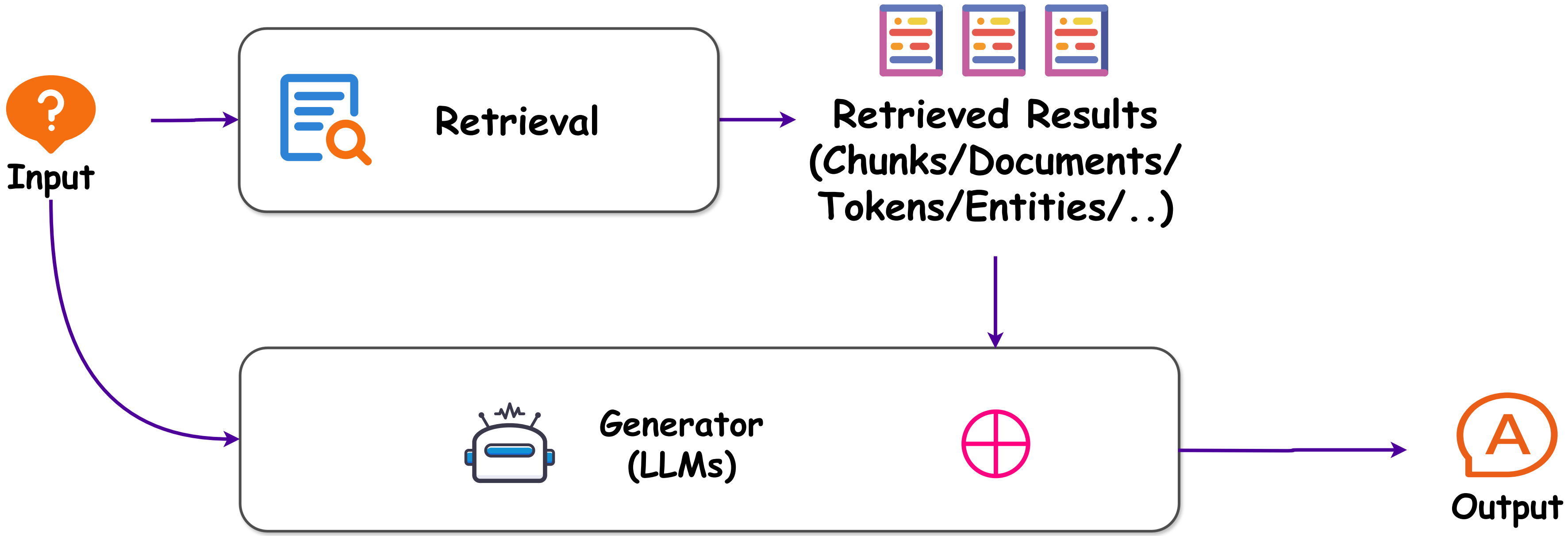
RA-LLM Architecture: Output-layer Integration

- **kNN-LM:** Combining retrieved probabilities and predicted ones in generation



Khandelwal el al. 2019. "Generalization through Memorization: Nearest Neighbor Language Models"

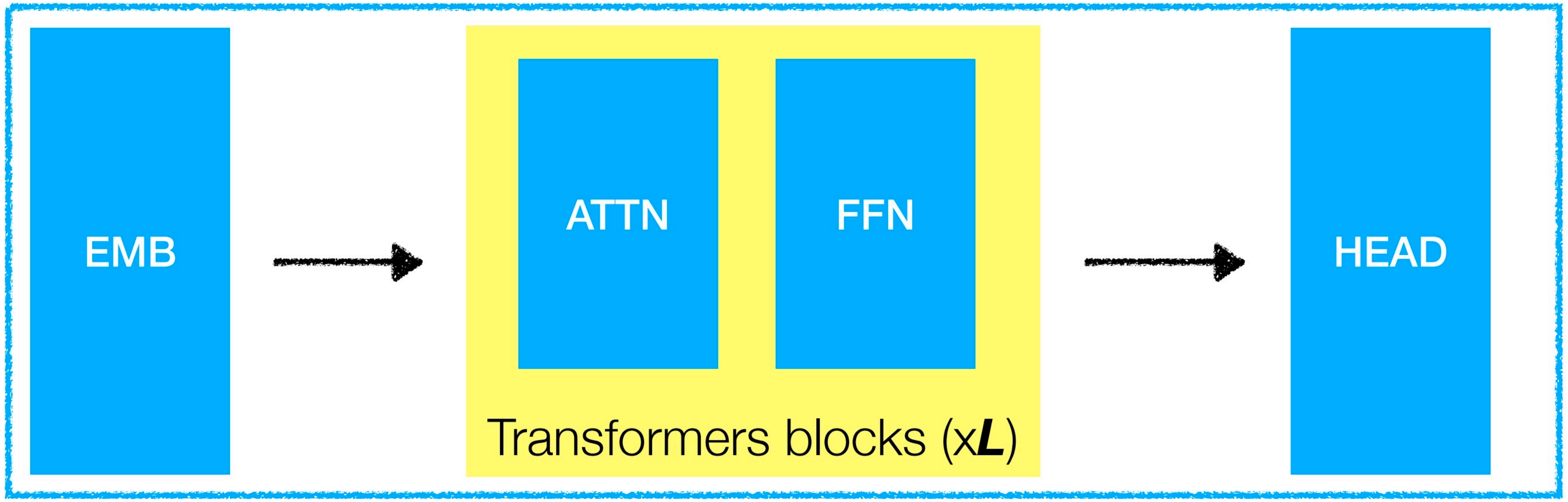
Retrieved Results Integration: Intermediate-layer Integration



Borgeaud et al. 2022. Improving Language Models by Retrieving from Trillions of Tokens

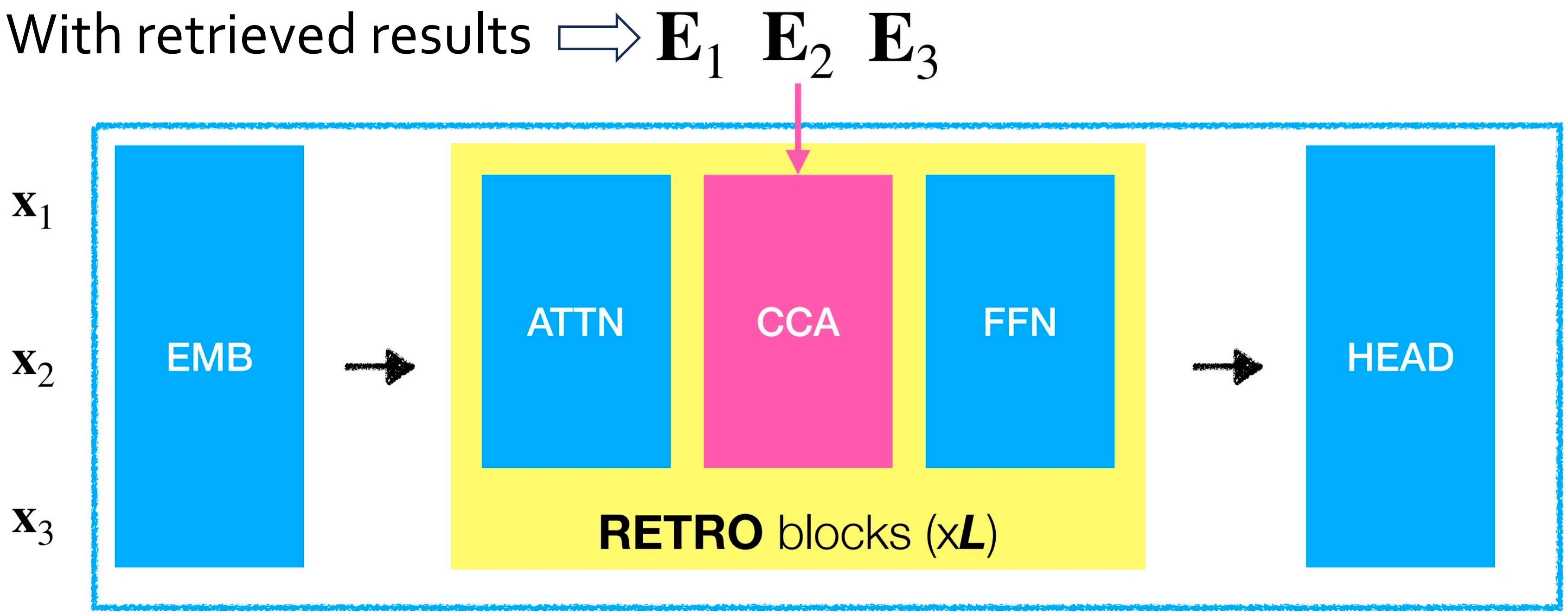
Retrieved Results Integration: Intermediate-layer Integration

Regular Decoder



Retrieved Results Integration: Intermediate-layer Integration

Decoder to incorporate retrieved results
(RETRO)

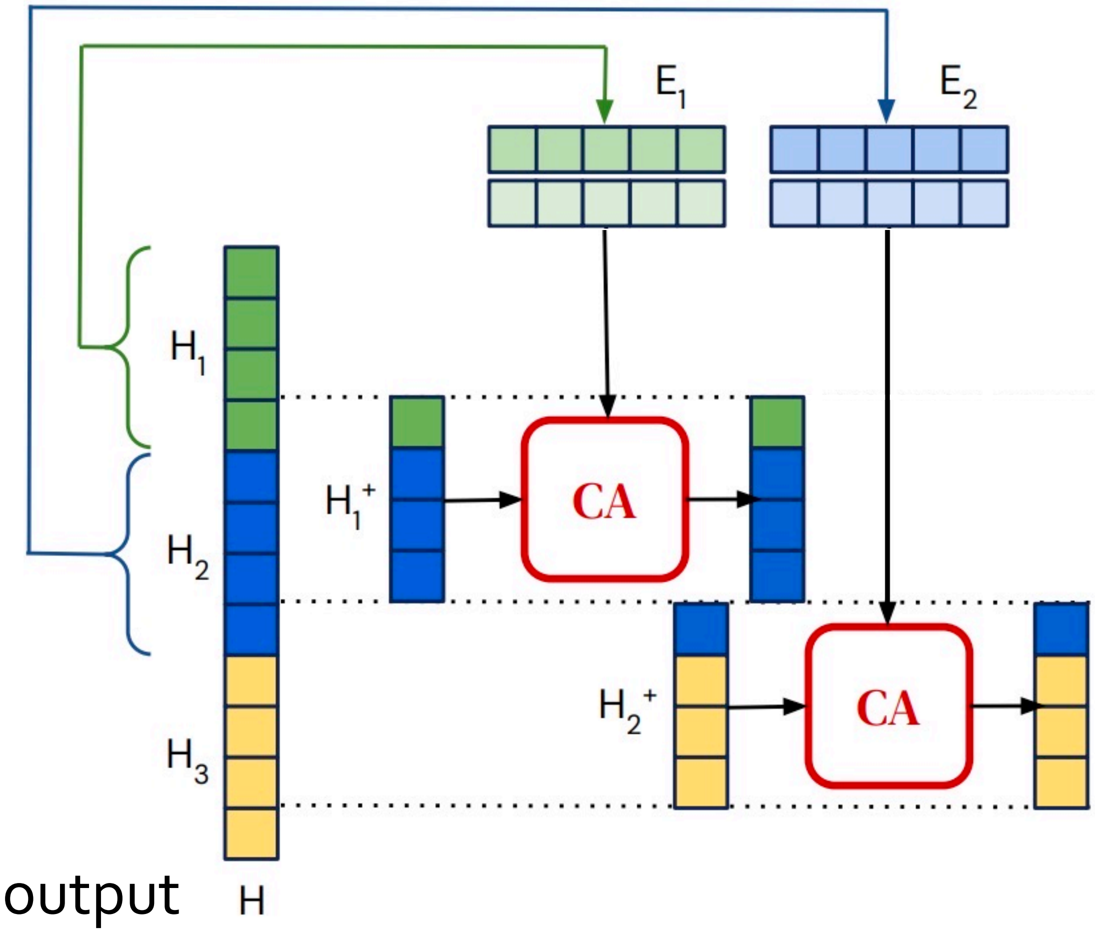


Chunked Cross Attention (CCA)

Borgeaud et al. 2022. Improving Language Models by Retrieving from Trillions of Tokens

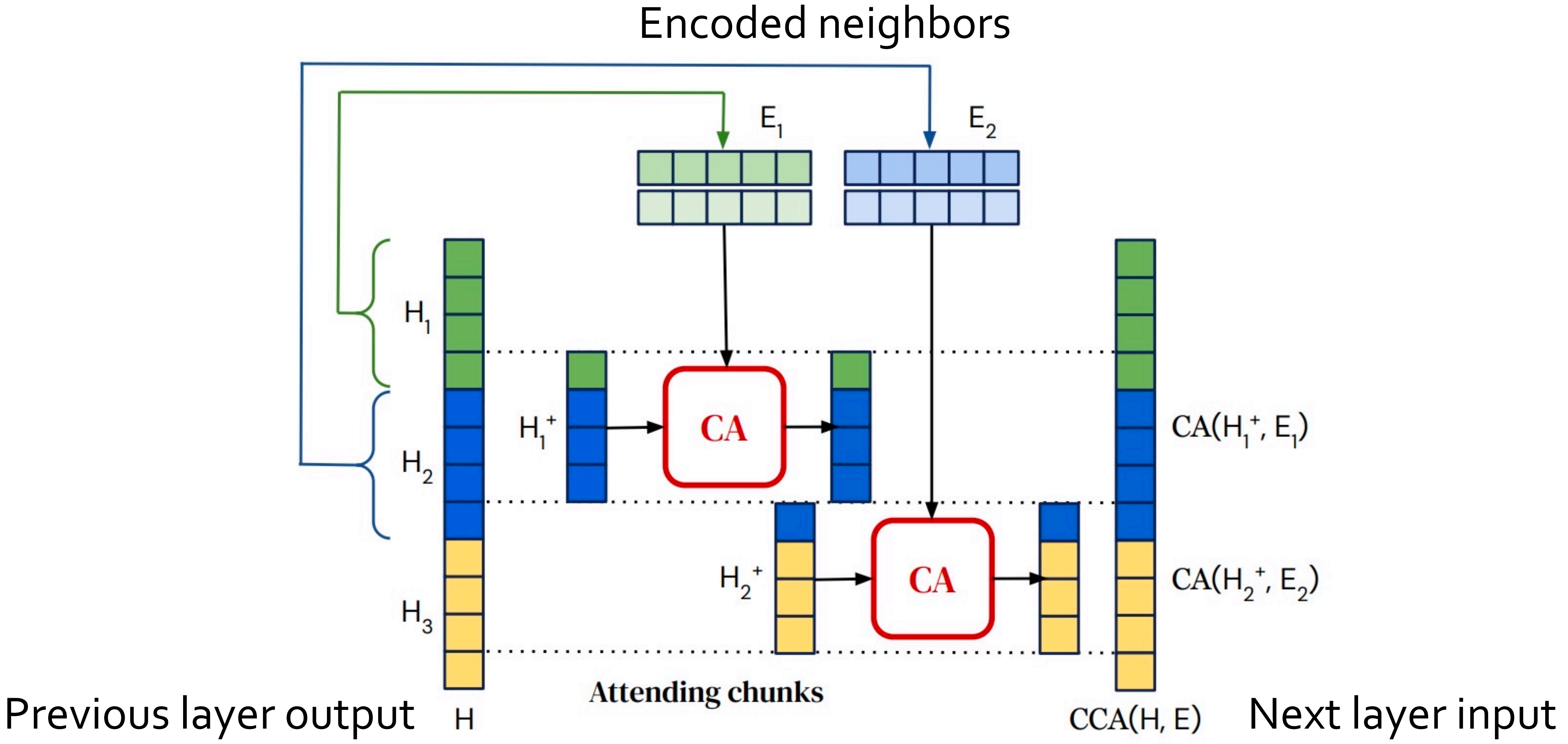
Retrieved Results Integration: Intermediate-layer Integration

Encoded neighbors



Previous layer output

Retrieved Results Integration: Intermediate-layer Integration



PART 2: Architecture of RA-LLMs and Main Modules

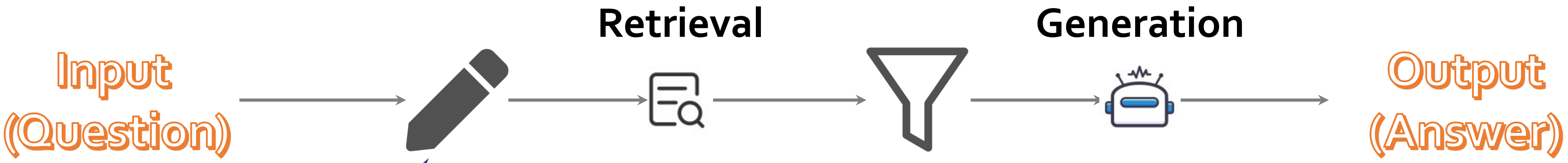
Slides

Website

Website of this tutorial

- RA-LLM architecture overview
- Retriever in RA-LLMs
- Retrieval results integration
- **Pre/Post-retrieval techniques**
- Special RA-LLM paradigms

Pre/Post-Retrieval Techniques



Pre-retrieval process: to improve the adaptation and effectiveness of the query

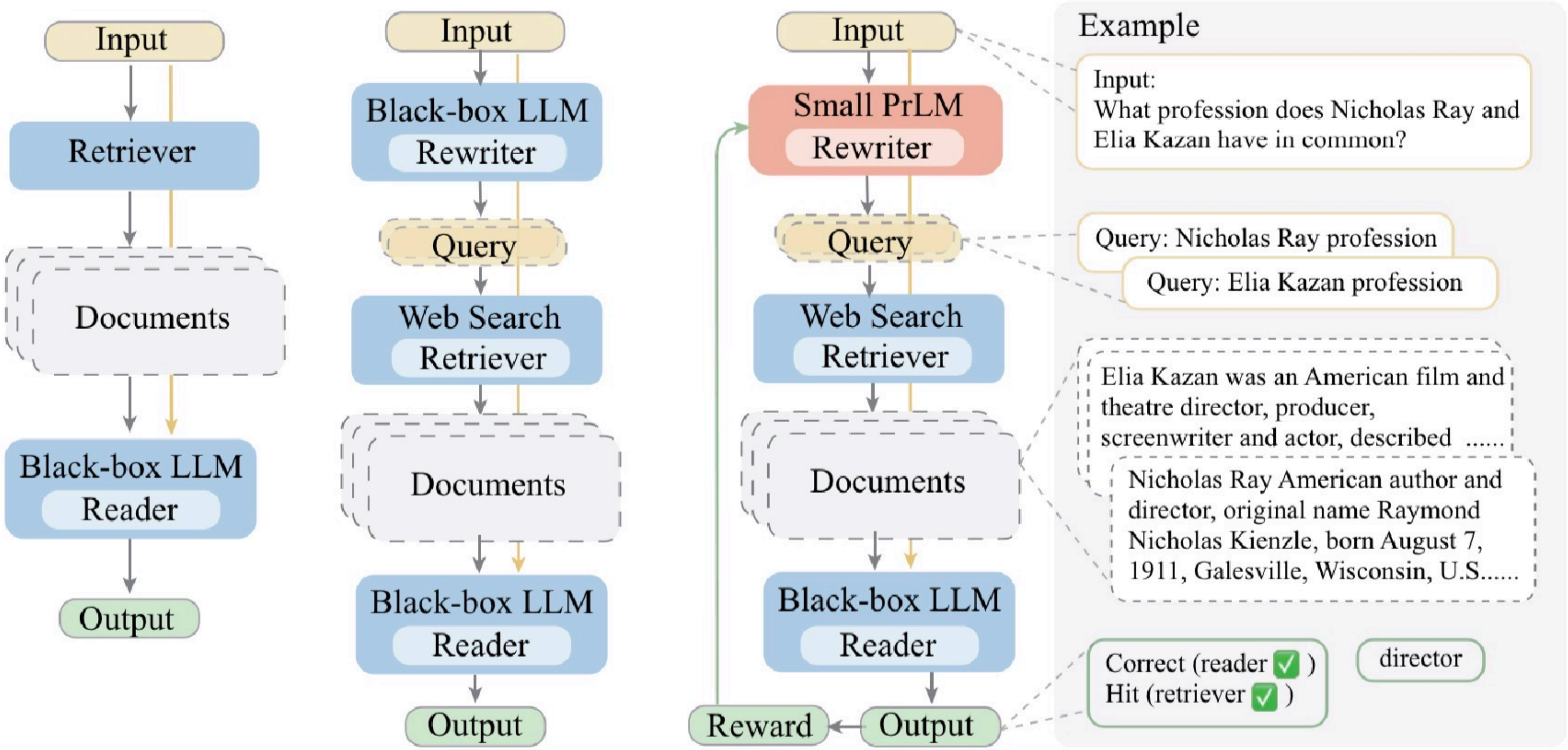
- Query rewriting
- Query decomposition
- Query expansion

Post-retrieval process: to select better results, merge multiple ones, etc

- Re-ranking
- Compression
- Correction

Pre-Retrieval Techniques

- **Query Rewriting**: to improve the adaptation of the query



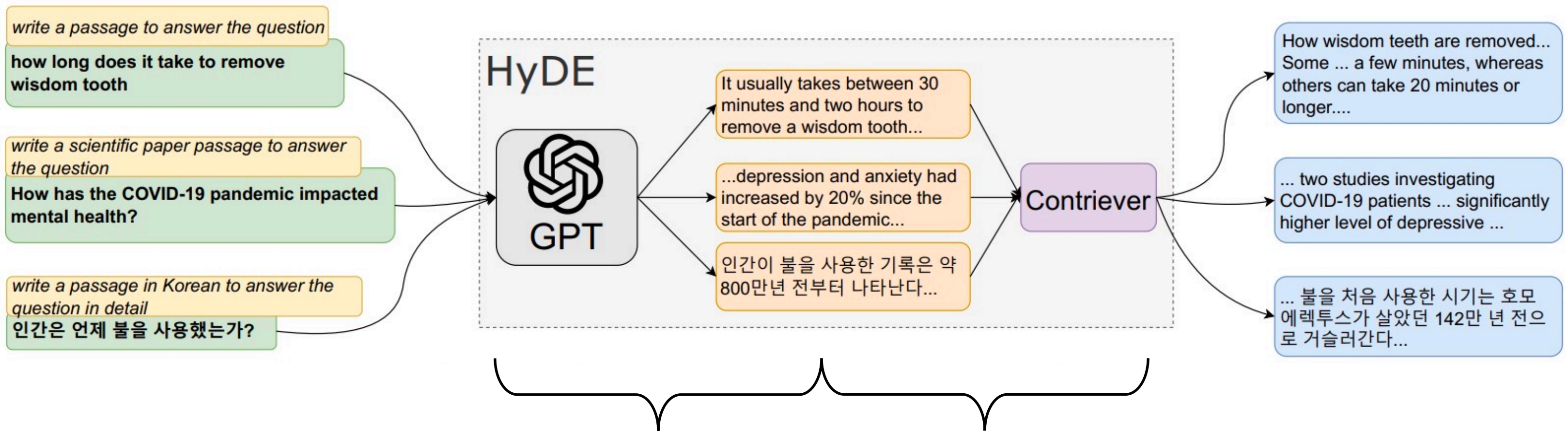
Model	EM	F_1
HotpotQA		
Direct	32.36	43.05
Retrieve-then-read	30.47	41.34
LLM rewriter	32.80	43.85
Trainable rewriter	34.38	45.97
AmbigNQ		
Direct	42.10	53.05
Retrieve-then-read	45.80	58.50
LLM rewriter	46.40	58.74
Trainable rewriter	47.80	60.71
PopQA		
Direct	41.94	44.61
Retrieve-then-read	43.20	47.53
LLM rewriter	46.00	49.74
Trainable rewriter	45.72	49.51

Works on different QA settings

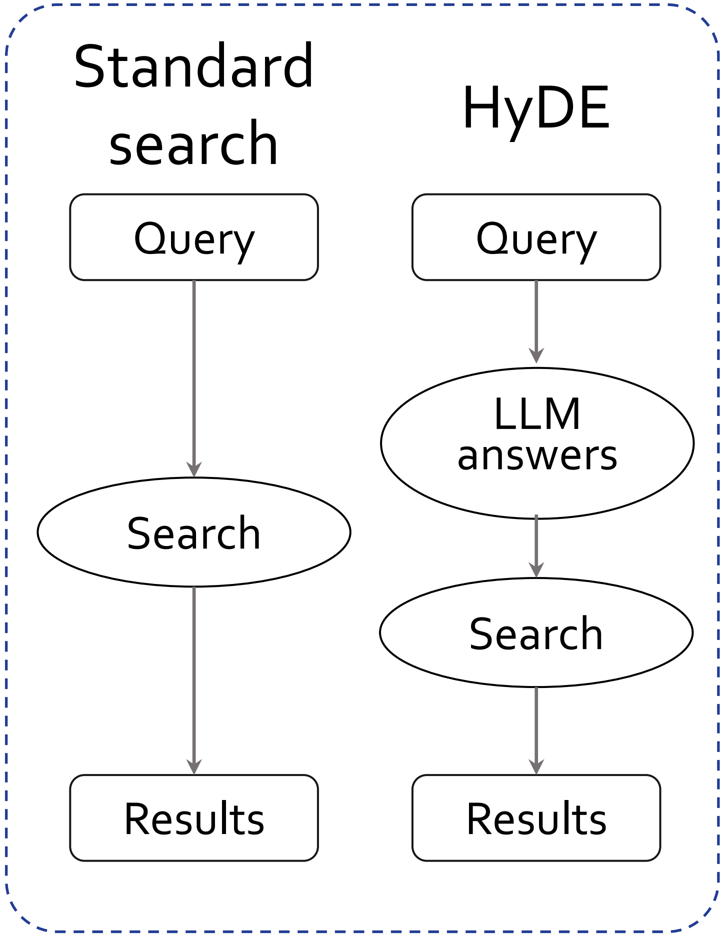
Wang et al. 2023. "Query rewriting for retrieval-augmented large language models"

Pre-Retrieval Techniques

- **HyDE**: Hypothetical Document Embeddings



1. Generative task   2. Document-document similarity task



Gao et al. 2022. "Precise zero-shot dense retrieval without relevance labels"

Pre-Retrieval Techniques

❑ Query Expansion

LLM Prompts

Write a passage that answers the given query:

Query: what state is this zip code 85282
Passage: Welcome to TEMPE, AZ 85282. 85282 is a rural zip code in Tempe, Arizona. The population is primarily white...

...

Query: when was pokemon green released
Passage:

Method	Fine-tuning	MS MARCO dev			TREC DL 19
		MRR@10	R@50	R@1k	nDCG@10
Sparse retrieval					
BM25	✗	18.4	58.5	85.7	51.2*
+ query2doc	✗	21.4$^{+3.0}$	65.3$^{+6.8}$	91.8$^{+6.1}$	**66.2**$^{+15.0}$
BM25 + RM3	✗	15.8	56.7	86.4	52.2
docT5query (Nogueira and Lin)	✓	**27.7**	**75.6**	**94.7**	64.2
Dense retrieval w/o distillation					
ANCE (Xiong et al., 2021)	✓	33.0	-	95.9	64.5
HyDE (Gao et al., 2022)	✗	-	-	-	61.3
DPR$_{\text{bert-base}}$ (our impl.)	✓	33.7	80.5	95.9	64.7
+ query2doc	✓	**35.1**$^{+1.4}$	**82.6**$^{+2.1}$	**97.2**$^{+1.3}$	**68.7**$^{+4.0}$

New query = original query + generated documents

$$q^+ = \text{concat}(q,\ [\text{SEP}],\ d')$$

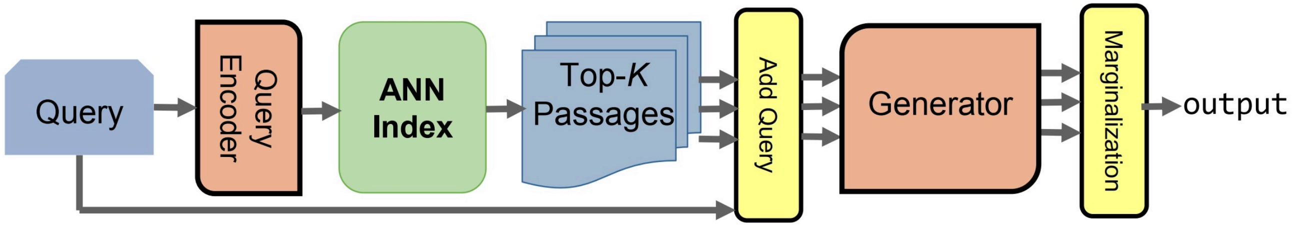
Works for both sparse and dense retrievers

Wang et al. 2023. "Query2doc: Query Expansion with Large Language Models"

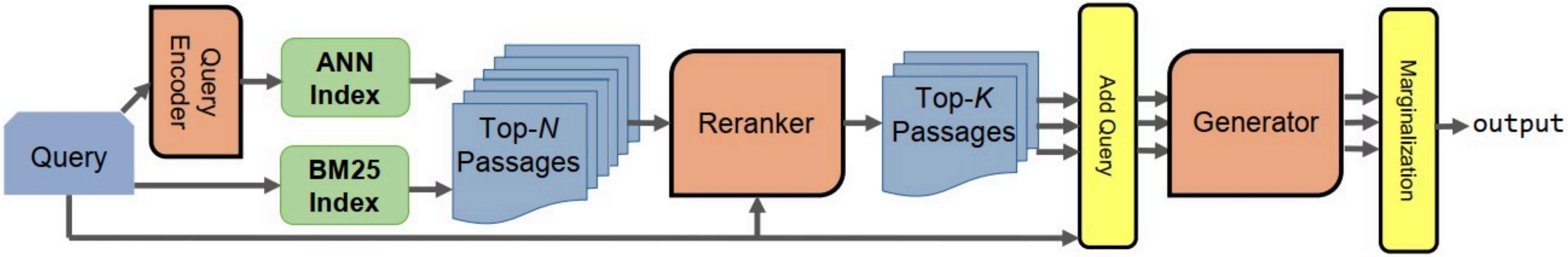
Post-Retrieval Techniques

❑ Retrieved Result Rerank (Re2G)

- Results from initial retrieval can be greatly improved through the use of a reranker
- Reranker allows merging retrieval results from sources with incomparable scores, e.g., BM25 and neura initial retrieval



Regular RAG pipeline

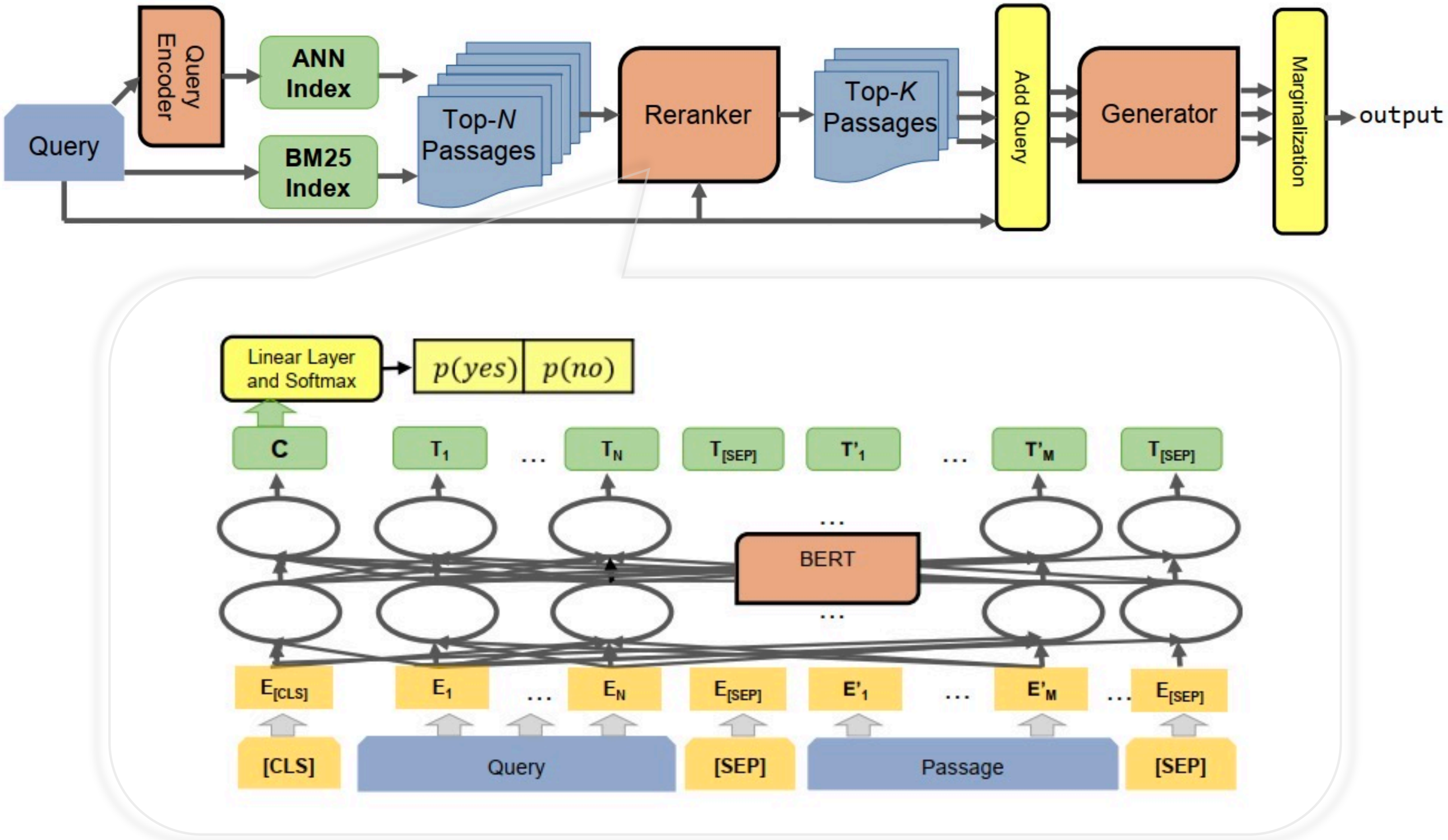


RAG with Reranker

Glass et al. 2022. "Re2G: Retrieve, Rerank, Generate"

Retrieved Result Rerank (Re2G) Model

- **Reranker: interaction model based on the sequence-pair classification**



Nogueira and Cho, 2019, Passage Re-ranking with BERT

Retrieved Result Rerank (Re2G) Performance

	T-REx		NQ		TriviaQA		FEVER		WoW	
	R-Prec	**R@5**	**R-Prec**	**R@5**	**R-Prec**	**R@5**	**R-Prec**	**R@5**	**R-Prec**	**R@5**
BM25	46.88	69.59	24.99	42.57	26.48	45.57	42.73	70.48	27.44	45.74
DPR Stage 1	49.02	63.34	56.64	64.38	60.12	64.04	75.49	84.66	34.74	60.22
KGI_0 DPR	65.02	75.52	64.65	69.60	60.55	63.65	80.34	86.53	**48.04**	**71.02**
Re^2G DPR	**67.16**	**76.42**	**65.88**	**70.90**	**62.33**	**65.72**	**84.13**	**87.90**	47.09	69.88
KGI_0 DPR+BM25	60.48	80.06	36.91	66.94	40.81	64.79	65.95	90.34	35.63	68.47
Reranker Stage 1	81.22	87.00	70.78	73.05	**71.80**	**71.98**	87.71	92.43	55.50	**74.98**
Re^2G Reranker	**81.24**	**88.58**	**70.92**	**74.79**	60.37	70.61	**90.06**	**92.91**	**57.89**	74.62

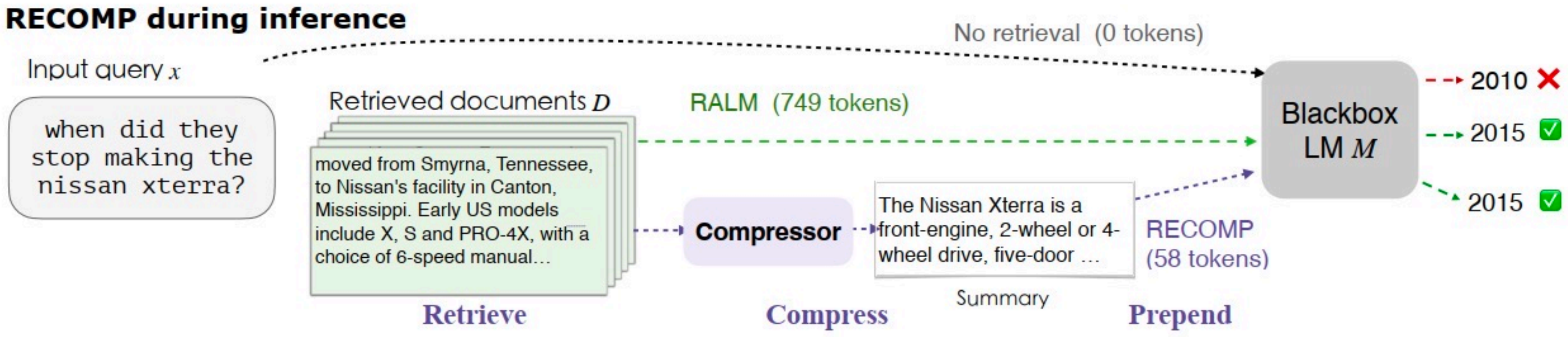
Significantly outperforms pipelines without the *Rerank* stage

Glass et al. 2022. “Re2G: Retrieve, Rerank, Generate”

Post-Retrieval Techniques

❑ Retrieved Result Compression

- ❖ To reduce the computational costs and also relieve the burden of LMs to identify relevant information in long retrieved documents.



❑ Compressor Learning Objectives

- ❖ Concise
- ❖ Effective
- ❖ Faithful

Xu et al. 2023. "RECOMP: Improving retrieval- augmented LMs with context compression and selective augmentation"

Retrieved Result Compression Performance

- QA tasks

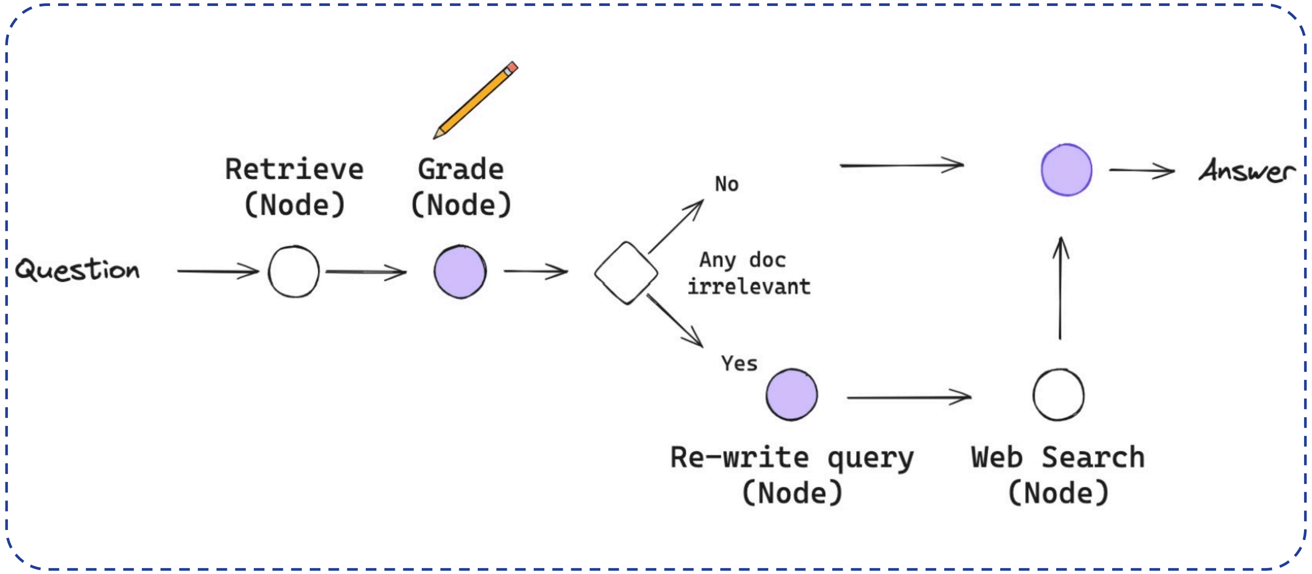
In-Context evidence	NQ # tok	NQ EM	NQ F1	TQA # tok	TQA EM	TQA F1	HotpotQA # tok	HotpotQA EM	HotpotQA F1
-	0	21.99	29.38	0	49.33	54.85	0	17.80	26.10
RALM without compression									
Top 1 documents	132	33.07	41.45	136	57.84	64.94	138	28.80	40.58
Top 5 documents	660	**39.39**	**48.28**	677	**62.37**	**70.09**	684	**32.80**	**43.90**
Phrase/token level compression									
Top 5 documents (NE)	338	23.60	31.02	128	54.96	61.19	157	22.20	31.89
Top 5 documents (BoW)	450	28.48	36.84	259	58.16	65.15	255	25.60	36.00
Extractive compression of top 5 documents									
Oracle	34	60.22	64.25	32	79.29	82.06	70	41.80	51.07
Random	32	23.27	31.09	31	50.18	56.24	61	21.00	29.86
BM25	36	25.82	33.63	37	54.67	61.19	74	26.80	38.02
DPR	39	34.32	43.38	41	56.58	62.96	78	27.40	38.15
Contriever	36	30.06	31.92	40	53.67	60.01	78	28.60	39.48
Ours	37	36.57	44.22	38	**58.99**	65.26	75	**30.40**	**40.14**

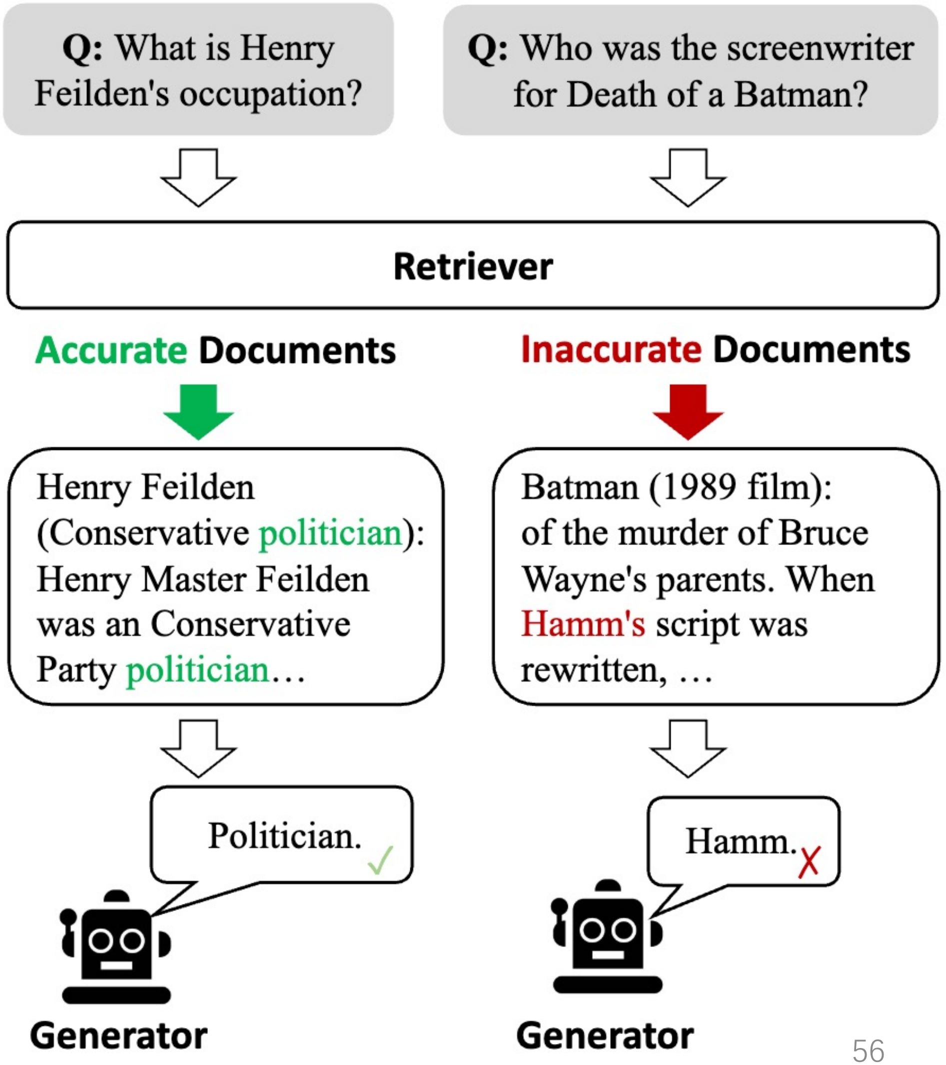
Outperforms representative sparse and dense retrievers

Xu et al. 2023. “RECOMP: Improving retrieval- augmented LMs with context compression and selective augmentation”

Post-Retrieval Techniques: Corrective RAG

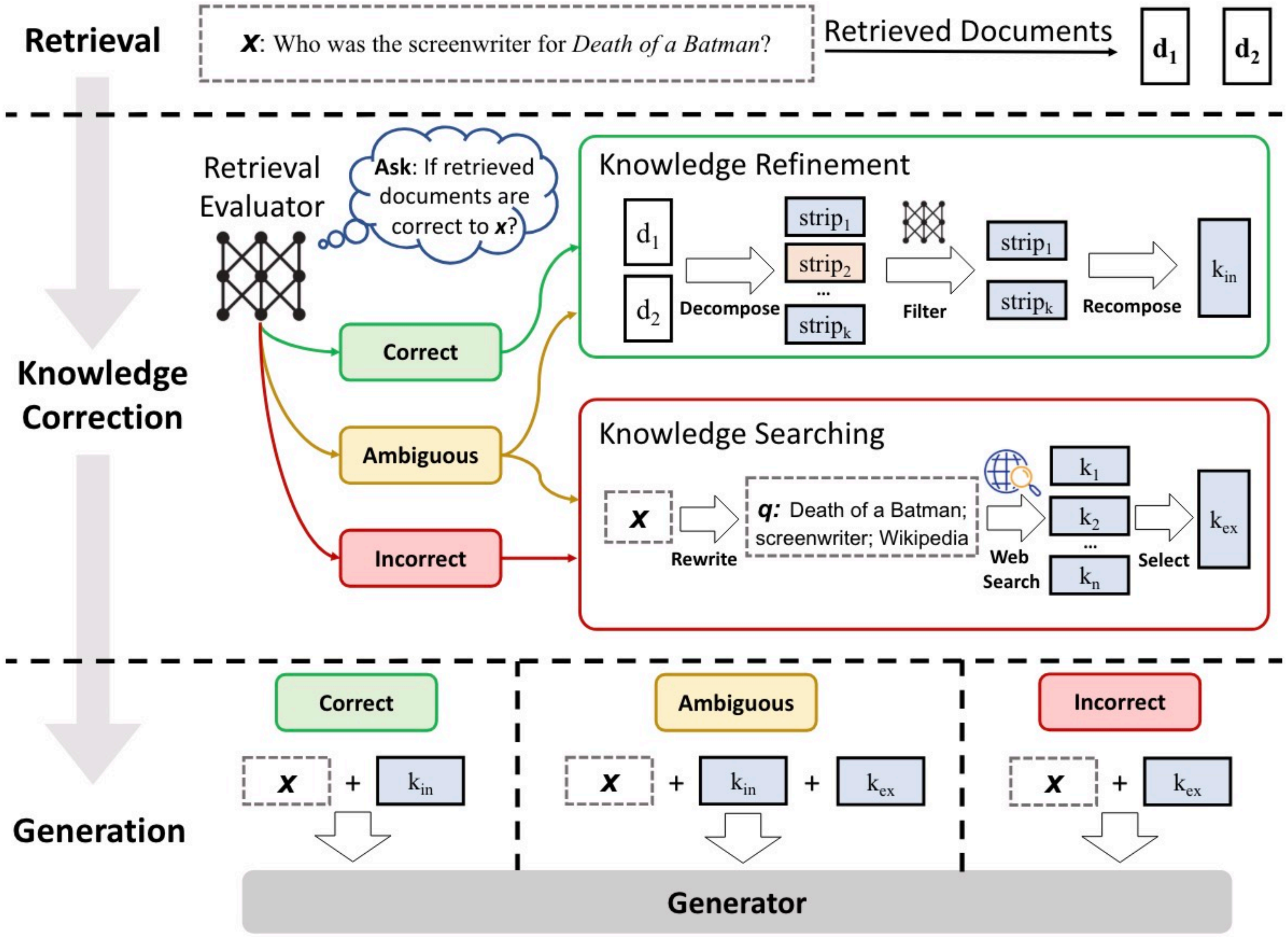
- Grading and correcting





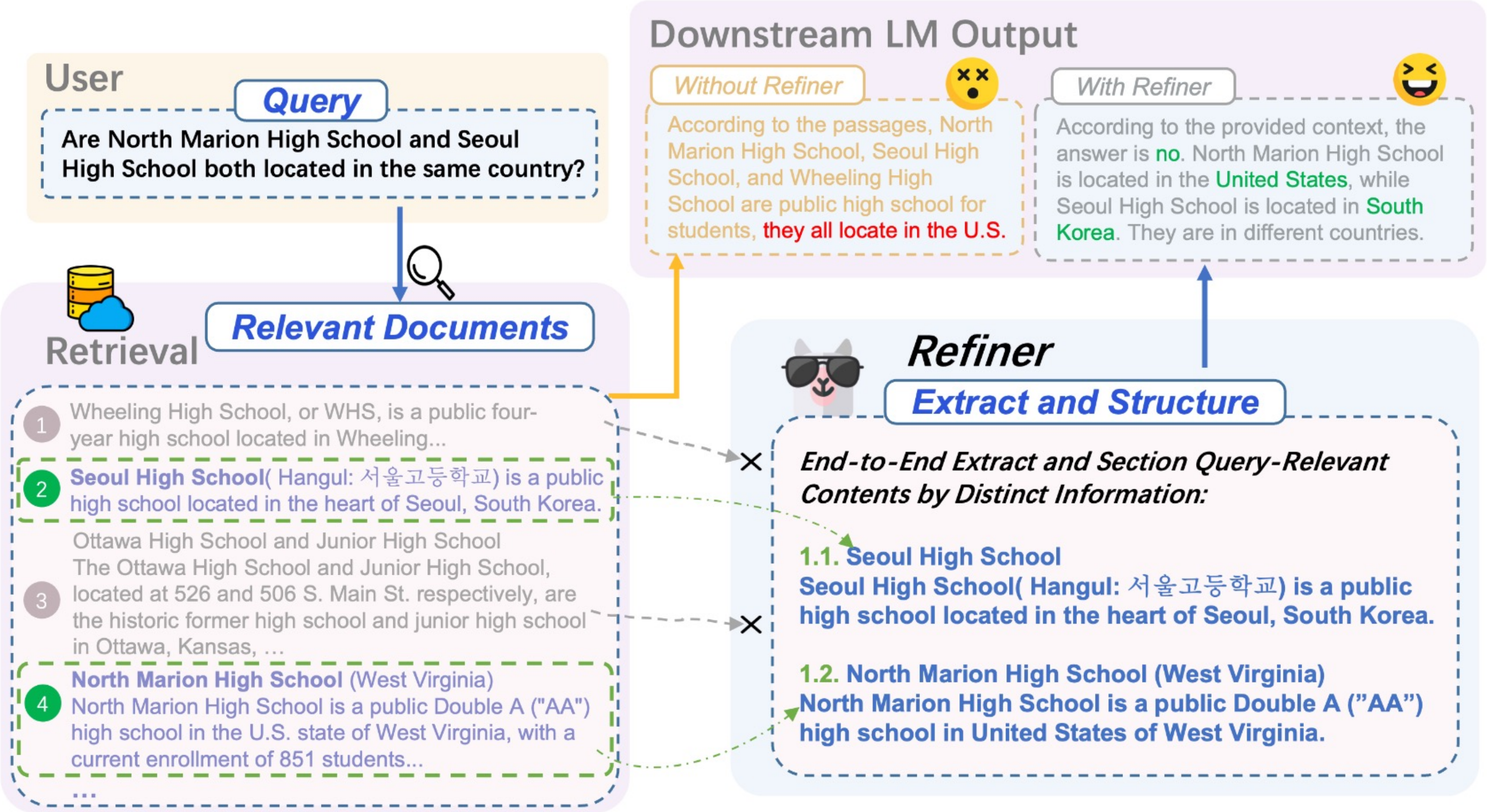
Yan et al., 2024, Corrective Retrieval Augmented Generation

Post-Retrieval Techniques: Corrective RAG



Post-Retrieval Techniques: Refiner

- **Refiner**: leveraging a single decoder-only LLM to adaptively extract query relevant contents verbatim along with the necessary context



Li et al., 2024, Refiner: Restructure Retrieval Content Efficiently to Advance Question-Answering Capabilities

PART 2: Architecture of RA-LLMs and Main Modules

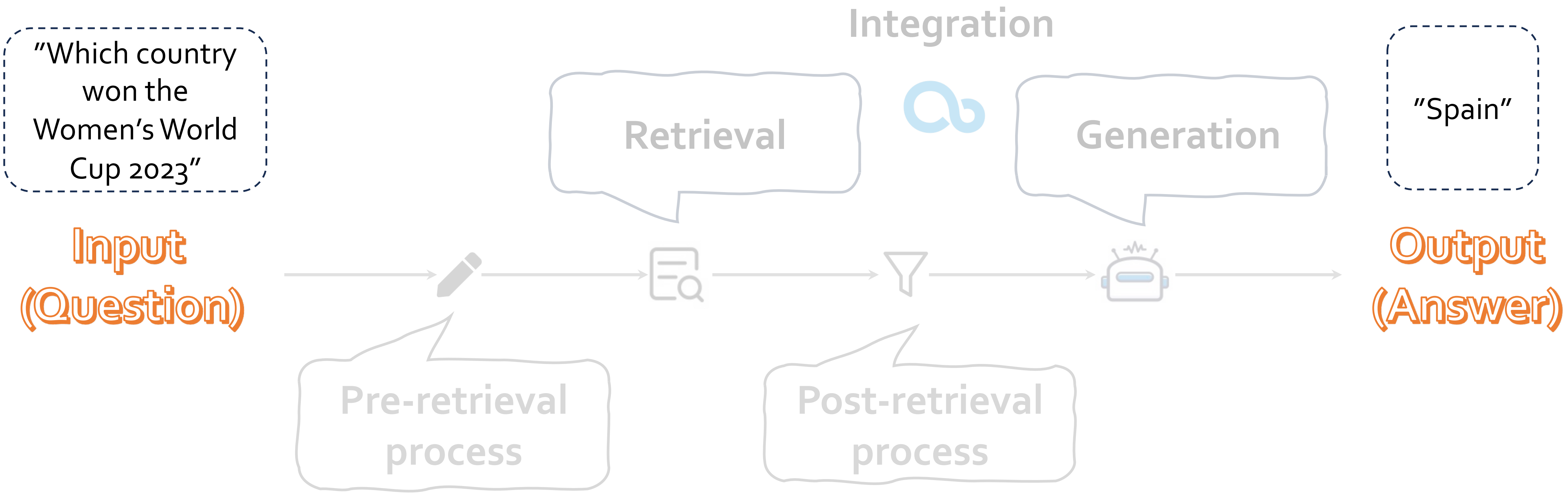
Slides

Website

Website of this tutorial

- **RA-LLM architecture overview**
- **Retriever in RA-LLMs**
- **Retrieval results integration**
- **Pre/Post-retrieval techniques**
- **Special RA-LLM paradigms**

Beyond Standard Pipelines and Components?



More?

Special RAG Pipeline: Self-Reflective RAG (SELF-RAG)

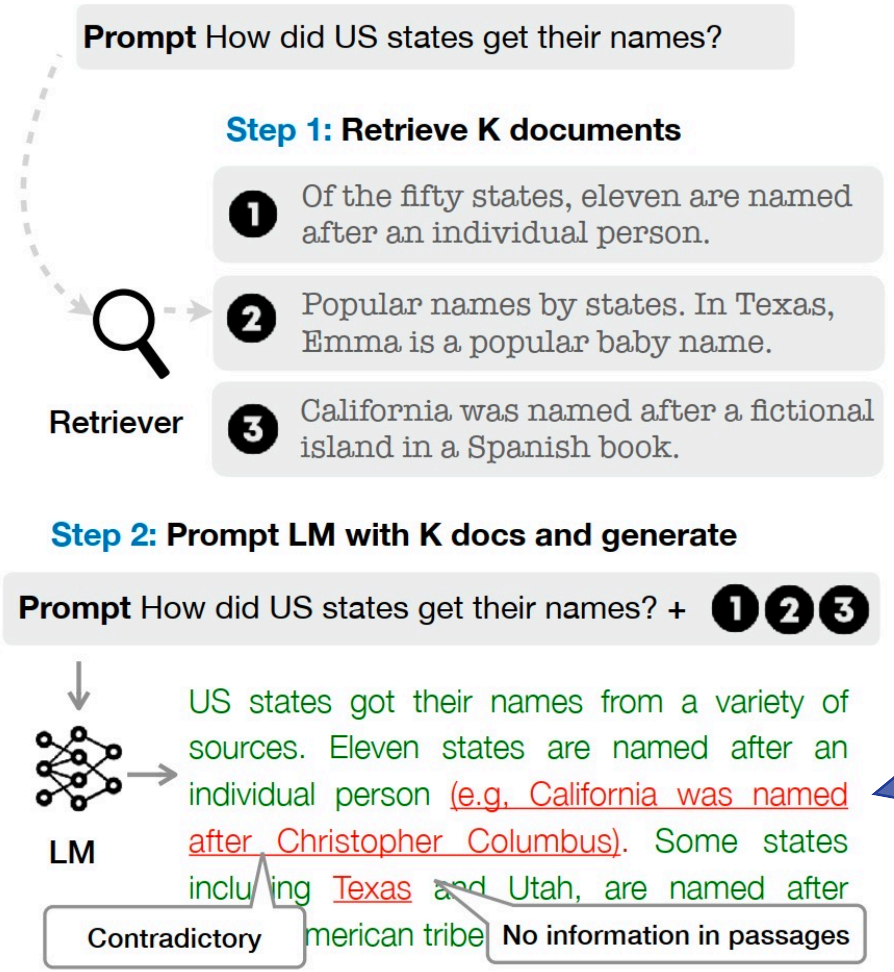
- General Retrieval-Augmented Generation (RAG)

Input

Retrieval

Retrieved result integration

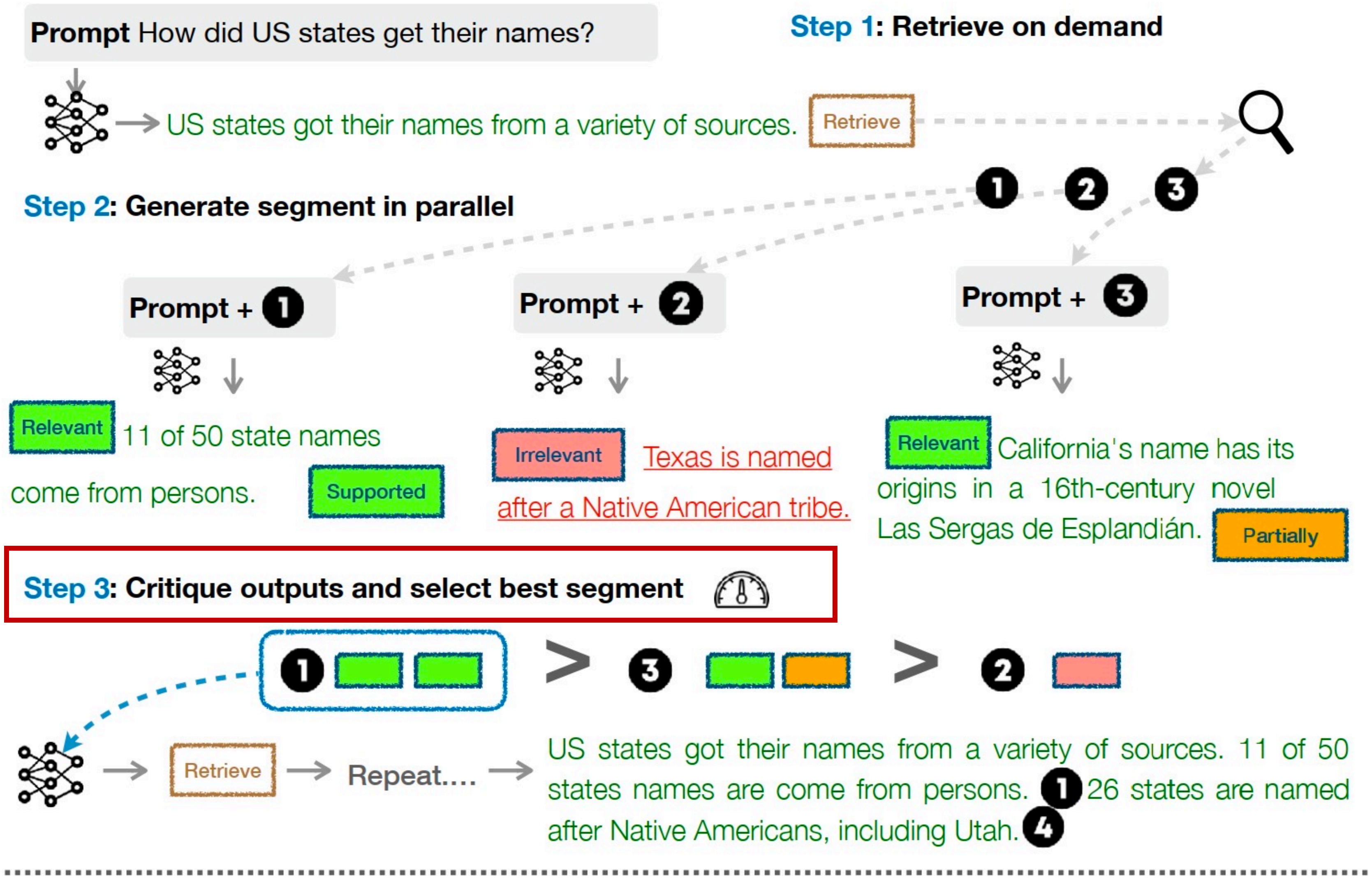
Generation



Retrieval-enhanced **generation results** are not necessarily useful or helpful !

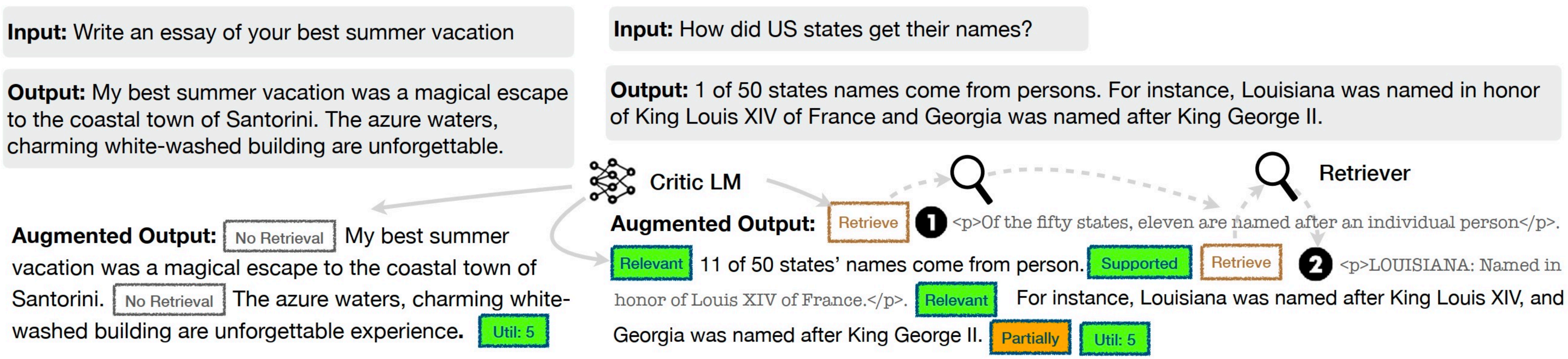
Asai A, et al.2023. “Self-rag: Learning to retrieve, generate, and critique through self-reflection”

SELF-RAG Overview



Key Technical Design in SELF-RAG

- Critic Model Training



- Four types of reflection tokens used in SELF-RAG

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.

SELF-RAG Algorithm

Algorithm 1 SELF-RAG Inference

Require: Generator LM $\mathcal{M}$, Retriever $\mathcal{R}$, Large-scale passage collections $\{d_1, \ldots, d_N\}$

Input: input prompt x and preceding generation $y_{<t}$, **Output:** next output segment y_t

$\mathcal{M}$ predicts $\boxed{\text{Retrieve}}$ given $(x, y_{<t})$

if $\boxed{\text{Retrieve}}$ == `Yes` **then**

  Retrieve relevant text passages $\mathbf{D}$ using $\mathcal{R}$ given (x, y_{t-1}) ▷ Retrieve

  $\mathcal{M}$ predicts $\boxed{\text{IsRel}}$ given x, d and y_t given $x, d, y_{<t}$ for each $d \in \mathbf{D}$ ▷ Generate

  $\mathcal{M}$ predicts $\boxed{\text{IsSup}}$ and $\boxed{\text{IsUse}}$ given x, y_t, d for each $d \in \mathbf{D}$ ▷ Critique

  Rank y_t based on $\boxed{\text{IsRel}}$, $\boxed{\text{IsSup}}$, $\boxed{\text{IsUse}}$ ▷ Detailed in Section 3.3

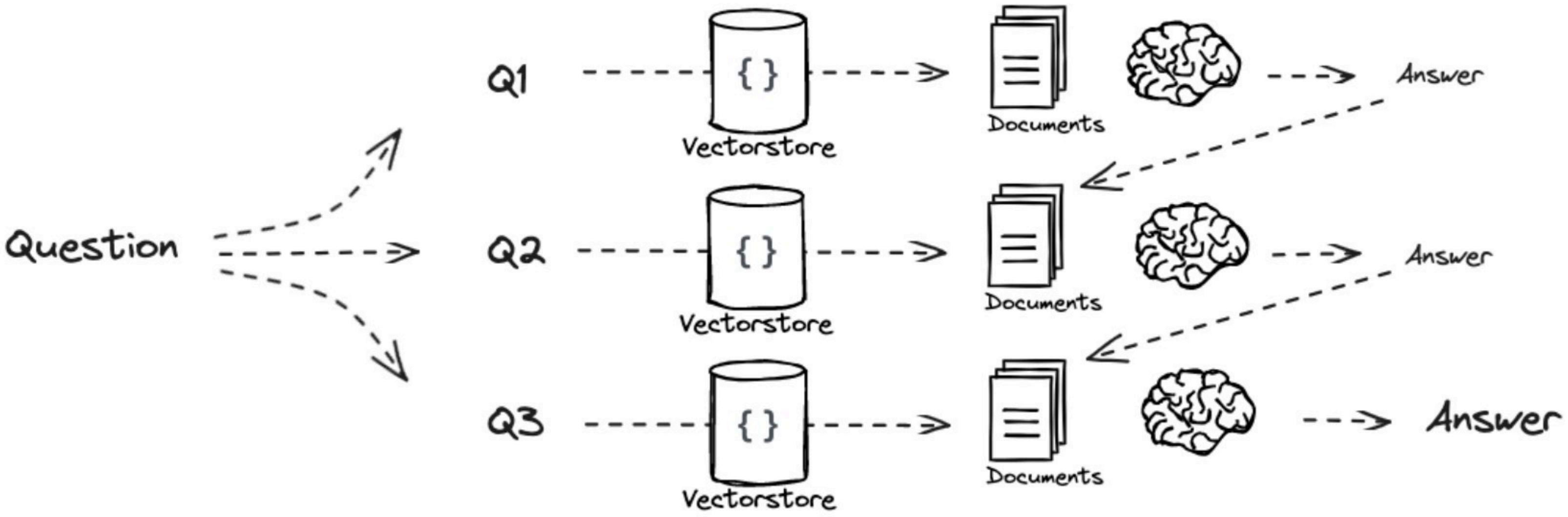
else if $\boxed{\text{Retrieve}}$ == `No` **then**

  $\mathcal{M}_{gen}$ predicts y_t given x ▷ Generate

  $\mathcal{M}_{gen}$ predicts $\boxed{\text{IsUse}}$ given x, y_t ▷ Critique

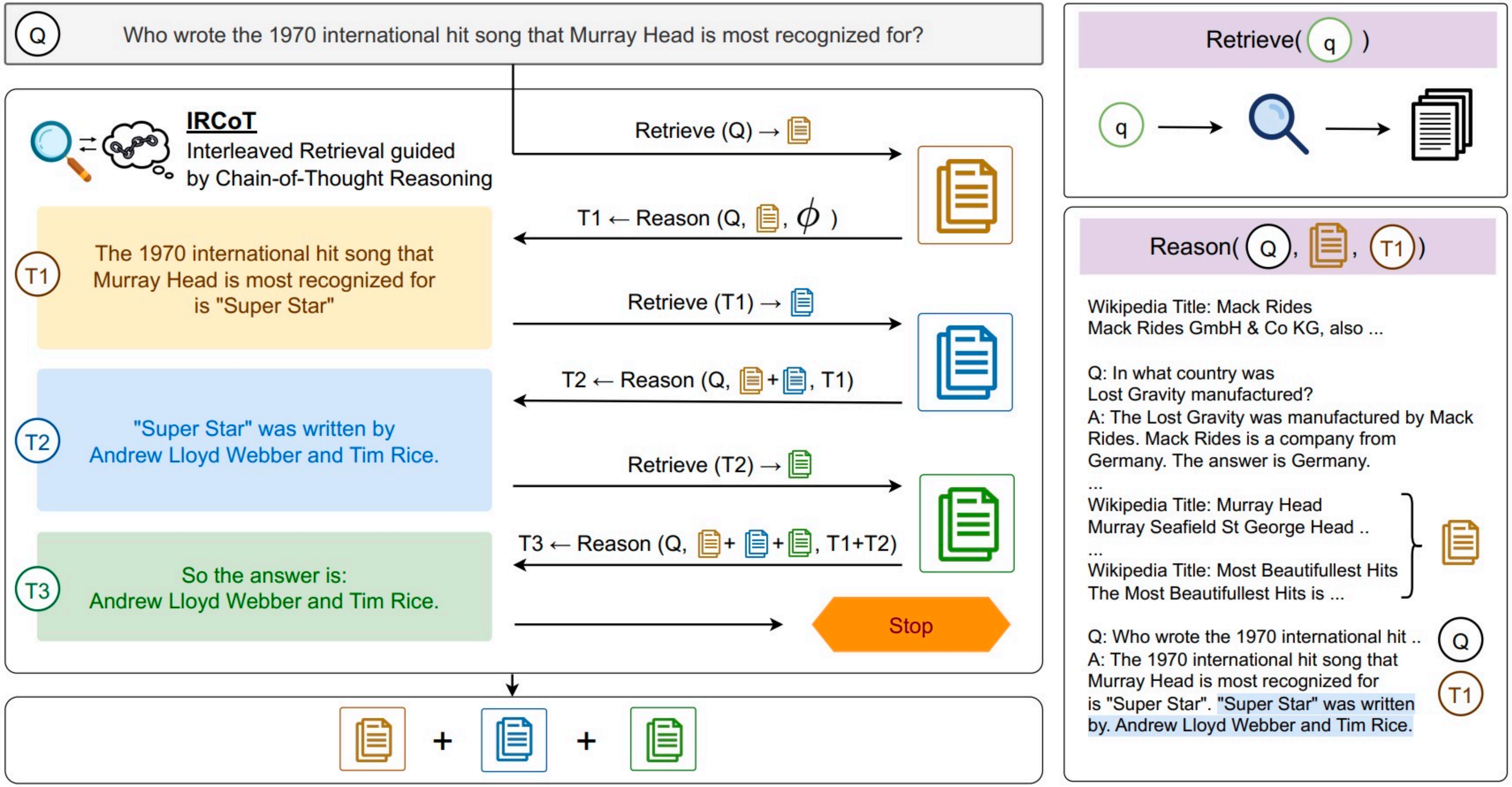
Special RAG Pipeline: Recursively Answer

- Chain-of-Thought + RAG
  - One-step retrieve-and-read approach is insufficient for multi-step QA
  - What to retrieve depends on what has already been derived, which in tern may depend on what was previously retrieved



Zhou, et al.2023. “Least-to-most prompting enables complex reasoning in large language models”
Trivedi, et al. 2023. “Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions”

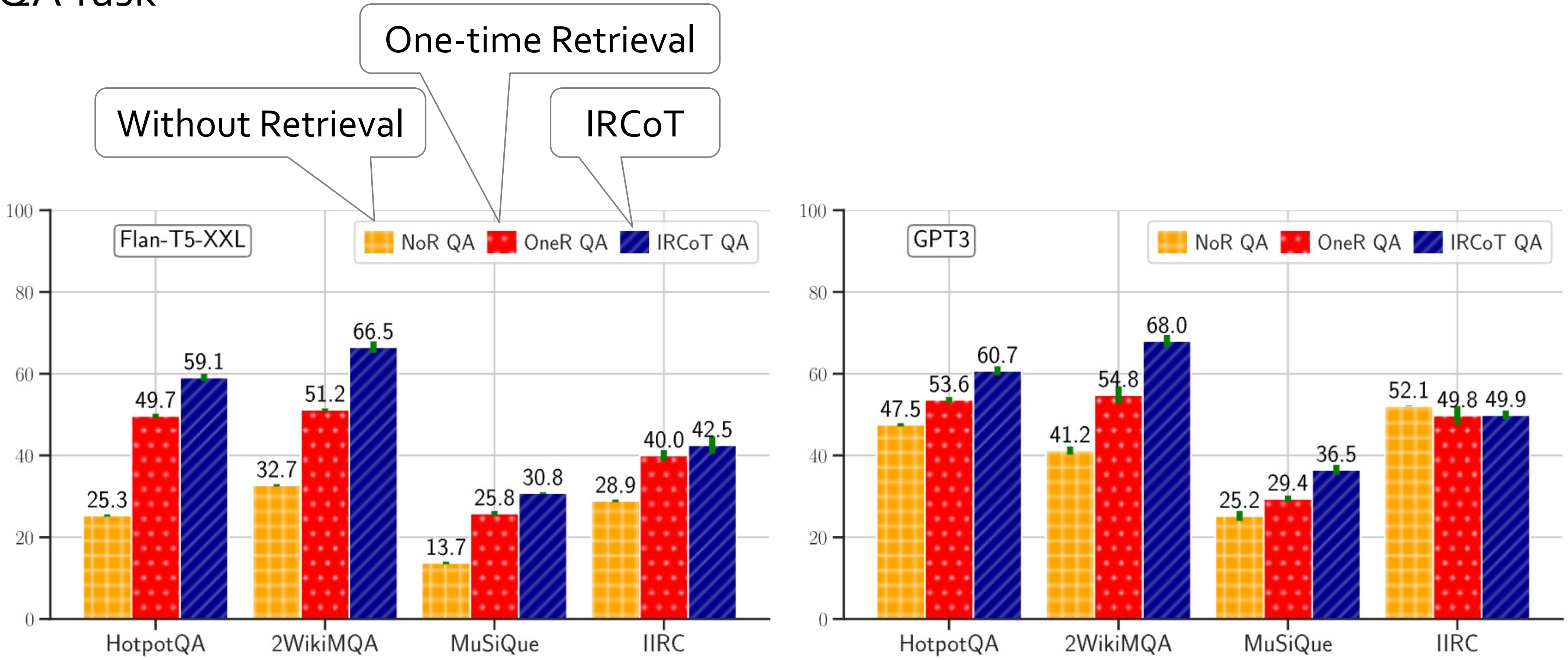
Interleaved Retrieval guided by Chain-of-Thought (IRCoT)



Trivedi, et al. 2023. “Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions”

IRCoT Performance

- QA Task



Tutorial Outline

- **Part 1: Introduction** of Retrieval Augmented Large Language Models (RA-LLMs) (Dr. Wenqi Fan)
- **Part 2: Architecture** of RA-LLMs and **Main Modules** (Dr. Yujuan Ding)
- **Part 3: Learning Approach of RA-LLMs (Liangbo Ning)**
- **Part 4: Applications** of RA-LLMs (Shijie Wang)
- **Part 5: Challenges and Future Directions** of RA-LLMs (Dr. Wenqi Fan)

Website of this tutorial
Check out the slides and more information!

Website