

Applied Deep Learning

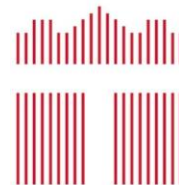


Retrieval-Augmented Generation (RAG)



November 20th, 2024

<http://adl.miulab.tw>



National
Taiwan
University
國立臺灣大學

Knowledge Access in LLMs



當然知道,蔡英文是中華民國第14、15任總統,也是台灣歷史上首位女性總統。

蔡英文於2016年以817萬票當選中華民國總統,2020年成功連任。作為民主進步黨主席,她在兩岸關係、能源轉型、社會福利等議題上有諸多政策主張。

在兩岸關係方面,蔡英文堅持維護台灣主權獨立與民主自由的核心價值。在她的領導下,台灣持續強化自我防衛能力並尋求國際支持。同時她也呼籲北京當局應以和平理性的方式處理分歧。

在內政方面,蔡英文政府致力推動能源轉型政策如非核家園計畫;積極落實長照2.0及幼托公共化等社會福利措施;推動司法改革及轉型正義工作;並因應數位時代來臨而提出「數位國家・創新經濟發展方案」等產業政策藍圖。

身為一名女性政治人物與國家領導人,蔡英文不僅代表了性別平權的里程碑,也象徵著台灣民主政治的成熟發展。她的一言一行備受各界矚目。

你知道蔡英文嗎?



Issue in LLMs: Hallucinations



你知道台大的陳繡儂嗎？



我知道陳繡儂教授,他是台灣大學~~電機工程學系~~的知名教授。以下是一些關於他的資訊:

1. 學術背景:陳繡儂教授畢業於台灣大學~~電機工程學系~~,並在美國~~史丹佛大學~~取得博士學位。
2. 研究領域:他的研究專長包括~~半導體元件、奈米電子學、積體電路設計~~等。
3. 重要成就:陳教授在~~半導體~~領域有許多創新研究和發明,曾獲得多項國際獎項肯定,如IEEE Fellow ~~國科會傑出研究獎~~等。
4. 教育貢獻:除了研究之外,陳教授也非常重視教育。他開設多門課程深受學生歡迎,並指導許多碩博士生從事前瞻性研究。
5. 產業合作:陳繡儂教授與台灣~~半導體產業~~有密切合作關係。他曾擔任~~台積電顧問委員會委員~~之一,協助推動產業技術發展。
6. 社會影響力:身為一位傑出的科研工作者和教育家,陳繡儂教授對台灣科技人才培育及產業發展有重要貢獻。

LLMs cannot memorize all details about long-tail information.

RAG: Retrieval-Augmented Generation



Web search ▶

你知道台大的陳縉儂嗎？



根據搜尋結果,陳縉儂是國立臺灣大學資訊工程學系的教授。她出生於台灣台北,在卡內基美隆大學獲得博士學位,同時也擁有該校的碩士學位和國立臺灣大學資訊工程學系的學士及碩士學位。她的研究領域主要為語言理解、機器智慧、口語對話系統、深度學習及自然語言處理等相關領域。

陳縉儂曾榮獲多項獎項,如IEEE ASRU 2013及IEEE SLT 2010的最佳學生論文獎、ISCA INTERSPEECH 2012之最佳學生論文提名以及ACL CLP之傑出碩士論文獎。在美國攻讀博士時,她決定有機會就要回台教書。畢業後,她先在美國微軟做博士後研究工作,年薪約450萬新台幣;成為微軟正職研究員後年薪更達近千萬新台幣。

然而陳縉儂選擇回到母校任教助理教授一職。她花費大量時間準備課程並指導第一屆帶領的研究生進入麻省理工學院攻讀博士。除了教書和研究外,

陳縉儂也熱愛舞蹈如嘻哈舞蹈等。

總而言之, 陳縉儂是一位優秀且充滿熱忱的人工智慧專家與教育者。

Sources:

 blog.104.com.tw

 [facebook.com](https://www.facebook.com)

 csie.ntu.edu.tw

Knowledge grounding is useful for reducing hallucinations.

Issue in LLMs: Outdated Knowledge



台灣現任總統是蔡英文女士。

台灣現任總統是誰？



LLMs' knowledge is easily outdated and hard to update

- Other solution: knowledge editing (Hsueh+, 2024)

6 WebGPT (Nakano et al., 2021)

Supervised fine-tuning via collected demonstration

a question in ELI5

Which has more words, the Harry Potter series or The Lord of the Rings?

a human-written answer with

The Harry Potter series contains more words than the Lord of the Rings trilogy. Including The Hobbit, the entire Lord of the Rings series contains 576,459 words^[2]. In comparison, the Harry Potter series contains 1,084,170 words^[1].

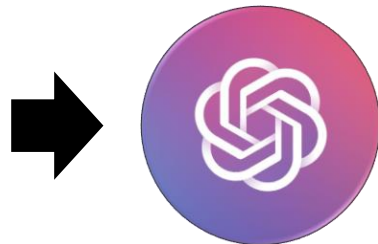
1. How Many Words are in Harry Potter? – Word Counter (wordcounter.io)
2. How many words does the novel The Lord of the Rings by J.R.R. Tolkien contain? | The Lord of the Rings: The Fellowship of the Ring Questions | Q & A | GradeSaver (www.gradesaver.com)

a question in ELI5

a human-written answer with references

demonstration

⋮



fine-tuning GPT-3

7

Token Continuation for Search Engine

1) Query generation

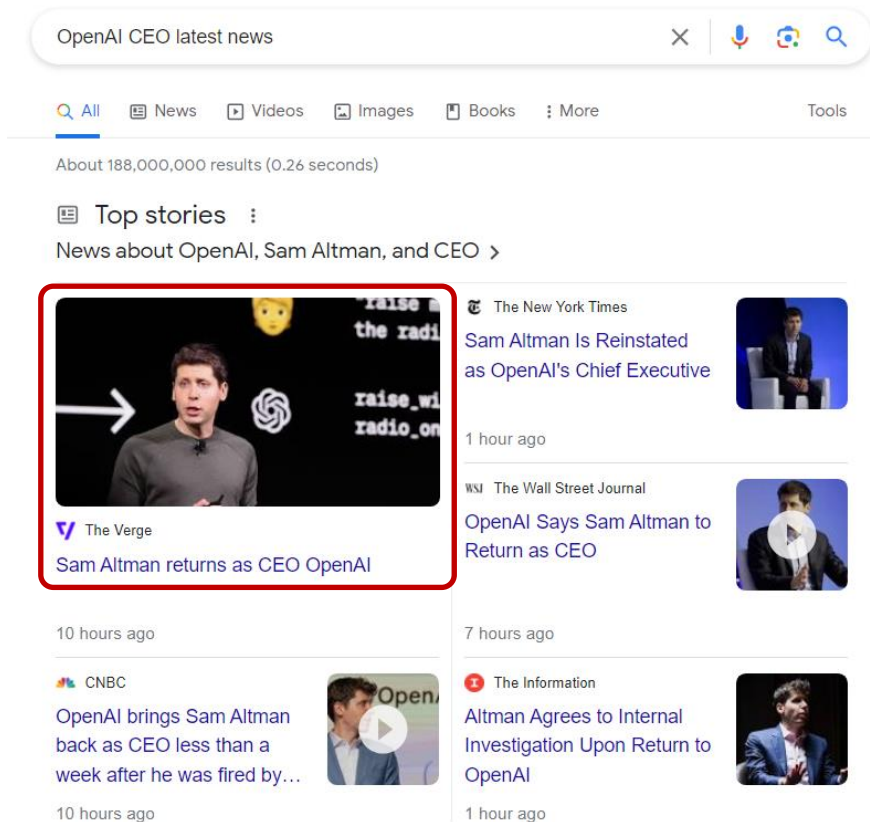
OpenAI CEO的最新動向如何？

[SEARCH] OpenAI CEO latest news [END]

2) Click the reference document

[CLICK] 1 [END]

Supervised learning of
human actions



8 Learning to Generate Query

目前OpenAI CEO的最新狀況為何?

✓ Searching for: **OpenAI CEO latest news**

✓ Generating answers for you...

OpenAI的CEO Sam Altman在上周五被解雇，但是根据最新消息¹，他已经被重新任命为CEO。此外，OpenAI的董事会也进行了改组，Larry Summers被任命为董事会成员¹。这些变化引起了业界的关注和震动^{1 2 3}。

¹: MSN News ²: MSN Money ³: GizmoChina

From Microsoft Start Partners



Economist Larry Summers joins th...

Sam Altman is back as OpenAI CE...

Learn more:

[1. msn.com](#)

[2. msn.com](#)

[3. gizmochina.com](#)

[4. arstechnica.com](#)

[5. cnn.com](#)

1 of 30 ●

Prior Probabilities Learned in LLMs

台灣現任總統是



LLM

蔡英文

Pre-training data is outdated

我好朋友是護士，



LLM

她對待病人很有耐心

他對待病人很有耐心

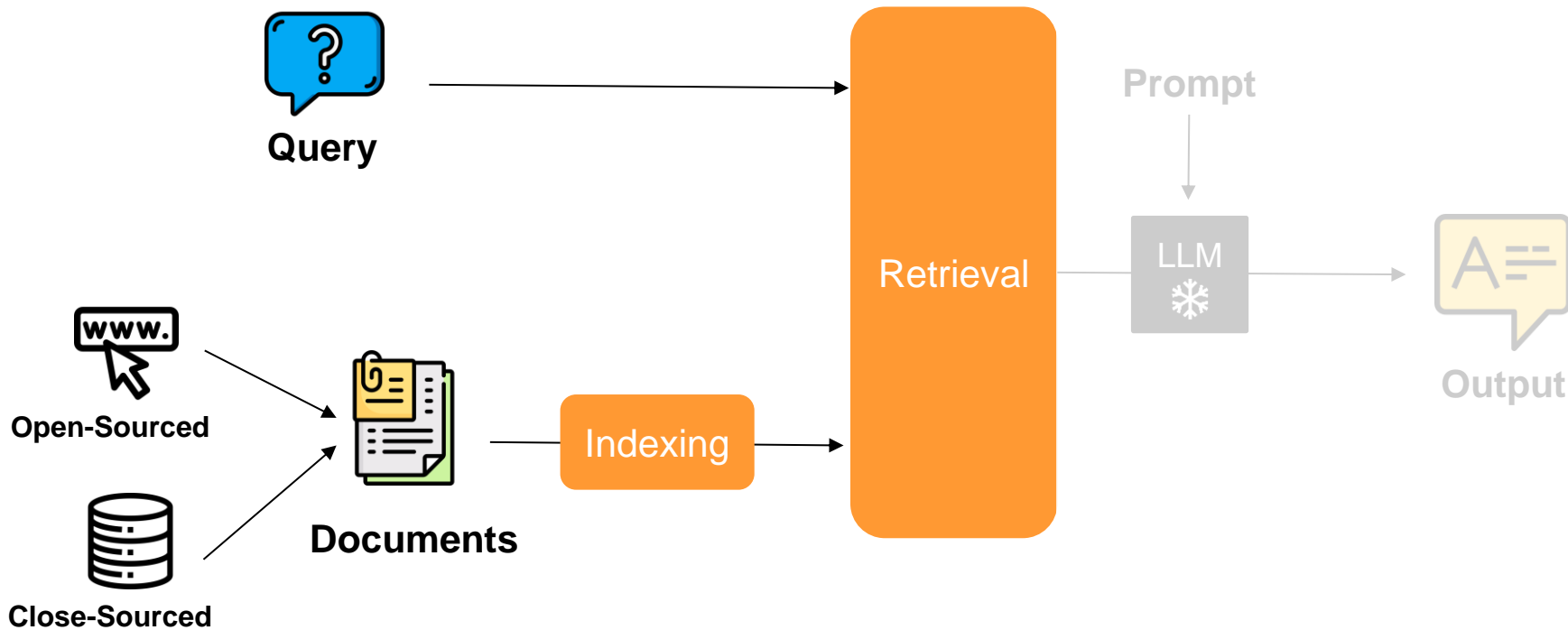
Generation follows the distribution in the pre-training data

10

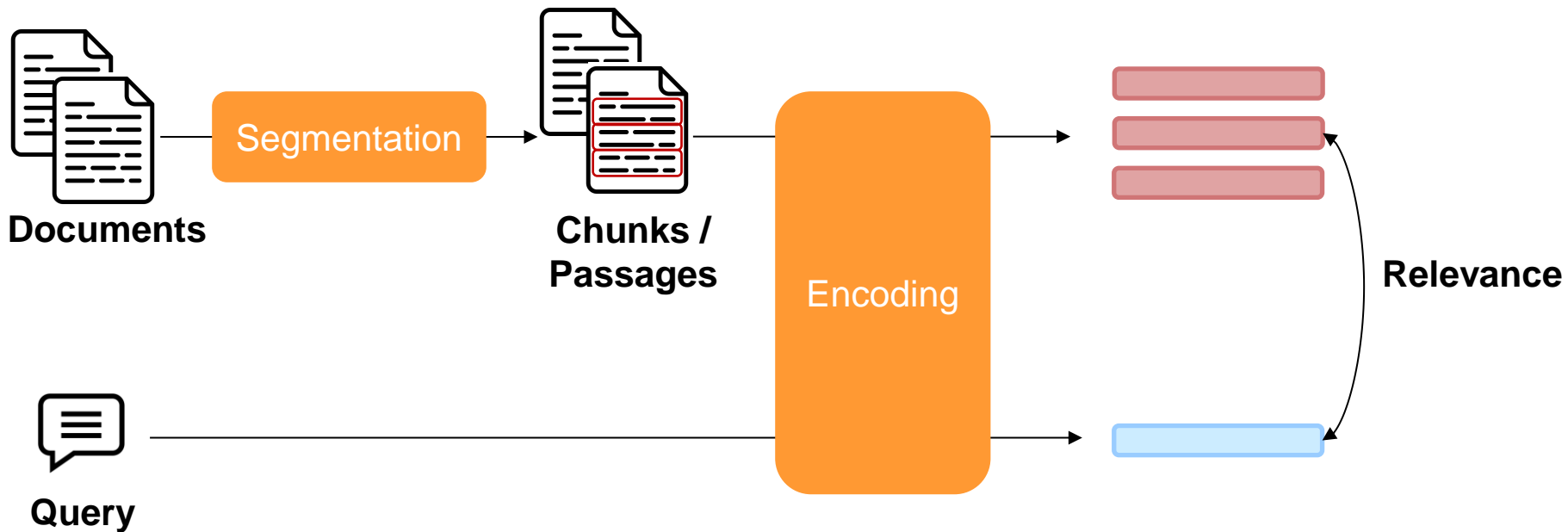
RAG Framework

Long-tail Knowledge, Dynamically Changed Knowledge, Knowledge not in Pre-training

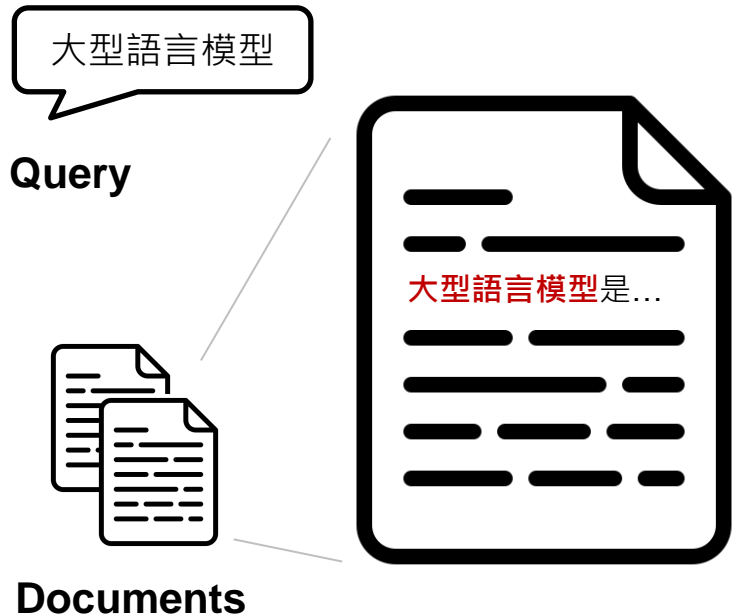
RAG Framework



Indexing & Retrieval



Sparse Retrieval



N-gram (TF-IDF)

q=what is nlp

| | | | |
|----------|--|---|---|
| | $d_1 = \text{what is life ?}$ candy is life ! | $d_2 = \text{nlp is an acronym for}$ natural language processing | $d_3 = \text{I like to do}$ good research on nlp |
| what | $\begin{pmatrix} 0.33 \\ 0 \end{pmatrix}$ | $\begin{pmatrix} 0.25 \\ 0.125 \end{pmatrix}$ | $\begin{pmatrix} 0 \\ 0 \end{pmatrix}$ |
| candy | $\begin{pmatrix} 0.33 \\ 0.33 \end{pmatrix}$ | $\begin{pmatrix} 0 \\ 0.25 \end{pmatrix}$ | $\begin{pmatrix} 0.125 \\ 0 \end{pmatrix}$ |
| nlp | $\begin{pmatrix} 0.33 \\ 0 \end{pmatrix}$ | $\begin{pmatrix} 0.25 \\ 0 \end{pmatrix}$ | $\begin{pmatrix} 0.125 \\ 0 \end{pmatrix}$ |
| is | $\begin{pmatrix} 0.33 \\ 0 \end{pmatrix}$ | $\begin{pmatrix} 0.25 \\ 0 \end{pmatrix}$ | $\begin{pmatrix} 0.125 \\ 0 \end{pmatrix}$ |
| language | $\begin{pmatrix} 0.33 \\ 0 \end{pmatrix}$ | $\begin{pmatrix} 0.25 \\ 0 \end{pmatrix}$ | $\begin{pmatrix} 0.125 \\ 0 \end{pmatrix}$ |
| ... | $\begin{pmatrix} \dots \\ \dots \end{pmatrix}$ | $\begin{pmatrix} \dots \\ \dots \end{pmatrix}$ | $\begin{pmatrix} \dots \\ \dots \end{pmatrix}$ |

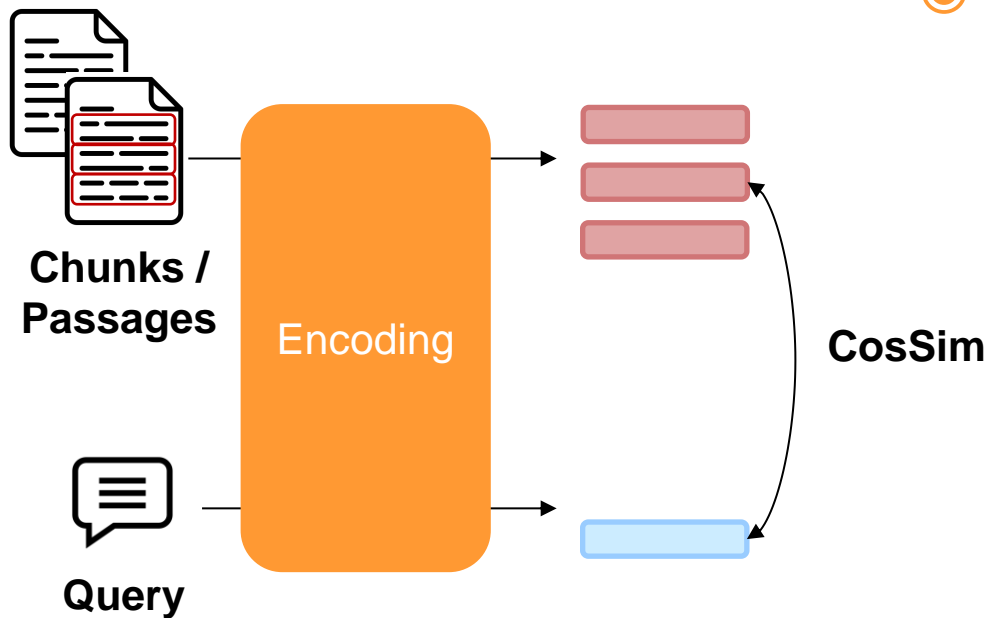
$q*d_1 = 0.165$ $q*d_2 = 0.0825$ $q*d_3 = 0.0413$

BM25

$$\text{score}(D, Q)$$

$$= \sum_{i=1}^n \text{IDF}(q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{\text{avgdl}}\right)}$$

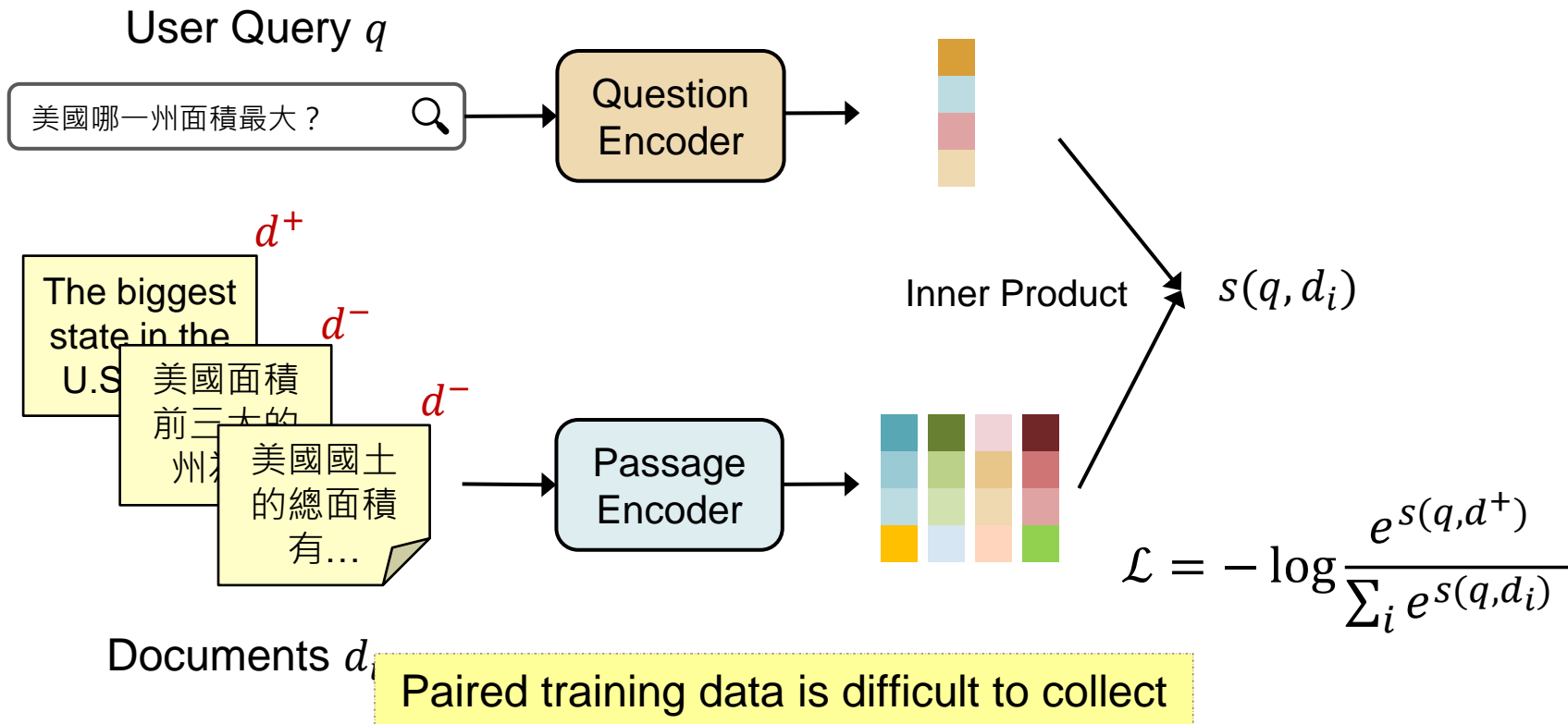
Dense Retrieval



Neural encoder

- Off-the-shelf embeddings
 - Pre-trained BERT
 - Pre-trained GPT
- Learned embeddings w/ contrastive learning
 - DPR (Karpukhin+, 2020)
 - Contriever (Izacard+, 2022)

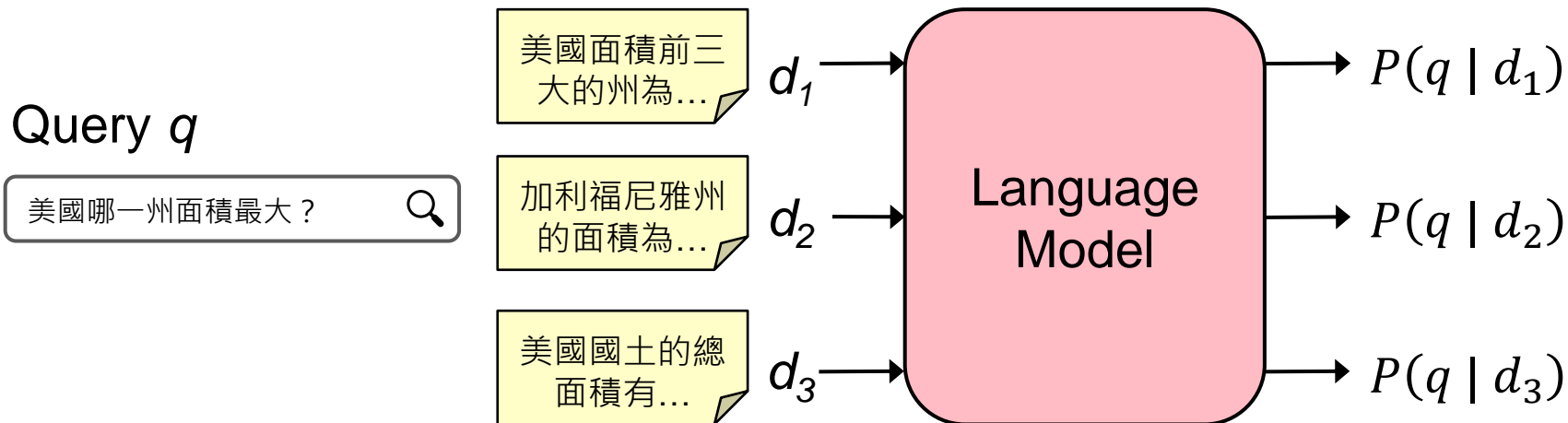
Training Dense Retrievers w/ Paired Data



Query Likelihood

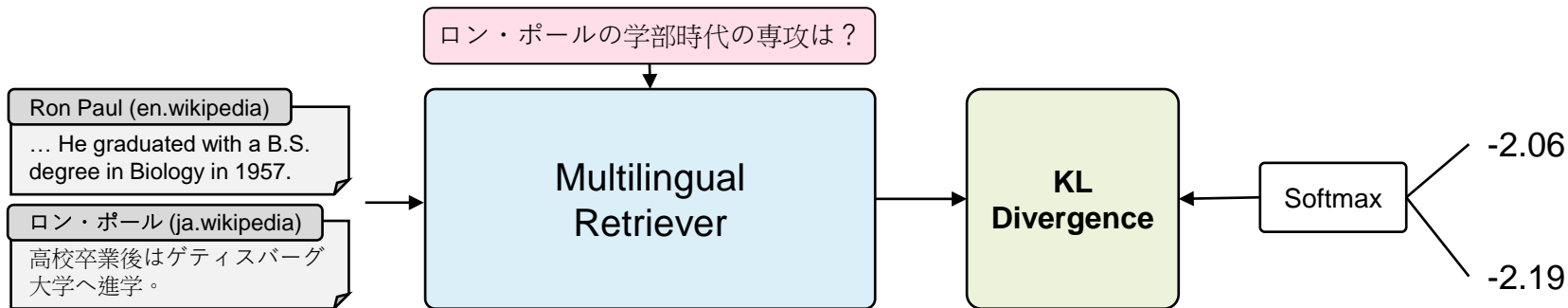
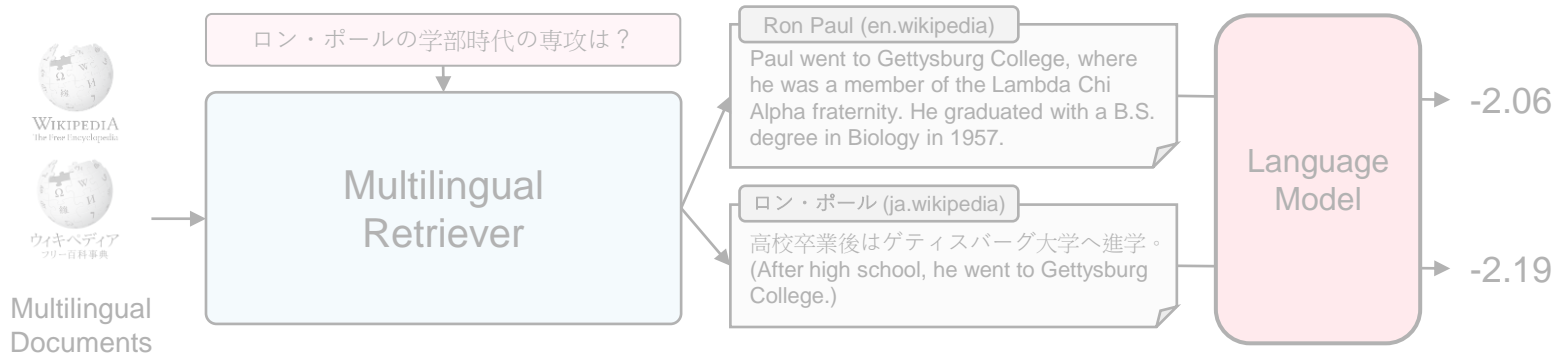
- LMs estimate conditional probability of text generation:

$$P(s) = \prod_{i=1}^{|s|} P(w_i | w_{<i})$$



Unsupervised Multilingual Dense Retrieval

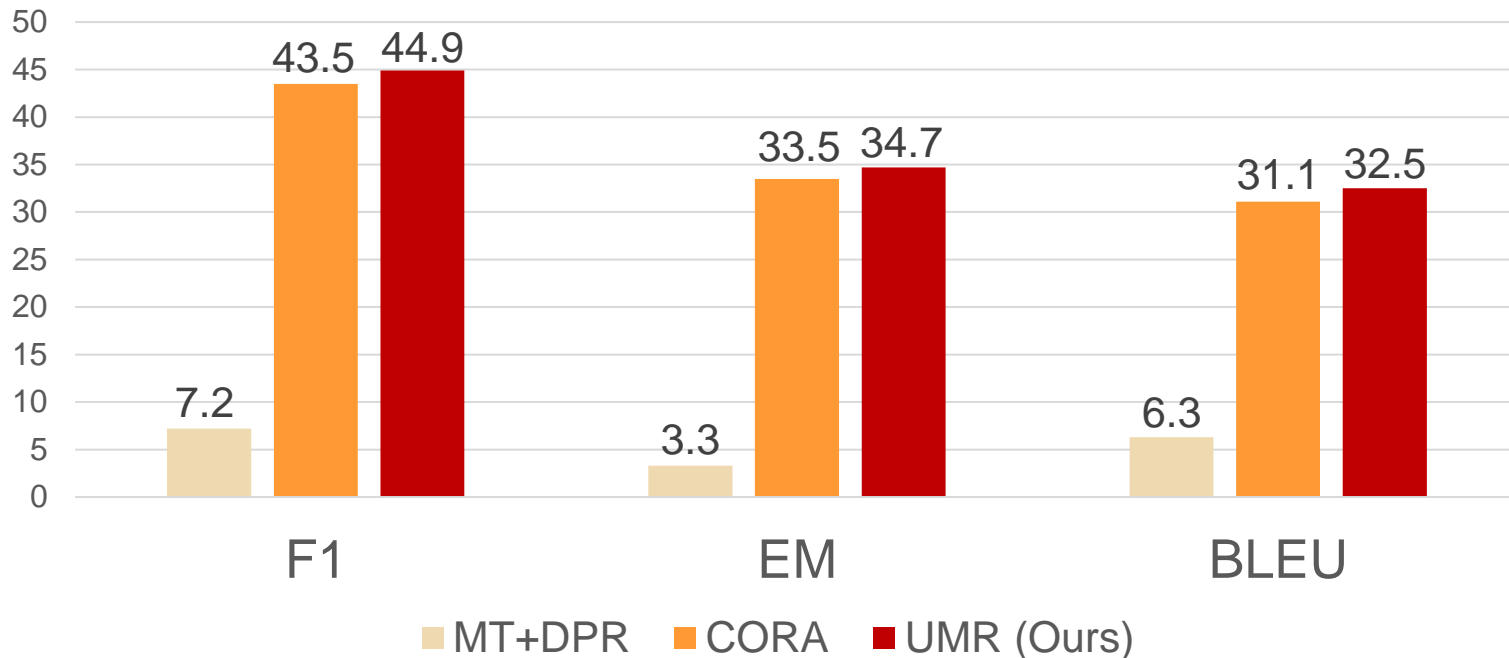
(Huang & Chen, 2024)



Unsupervised Multilingual Dense Retrieval

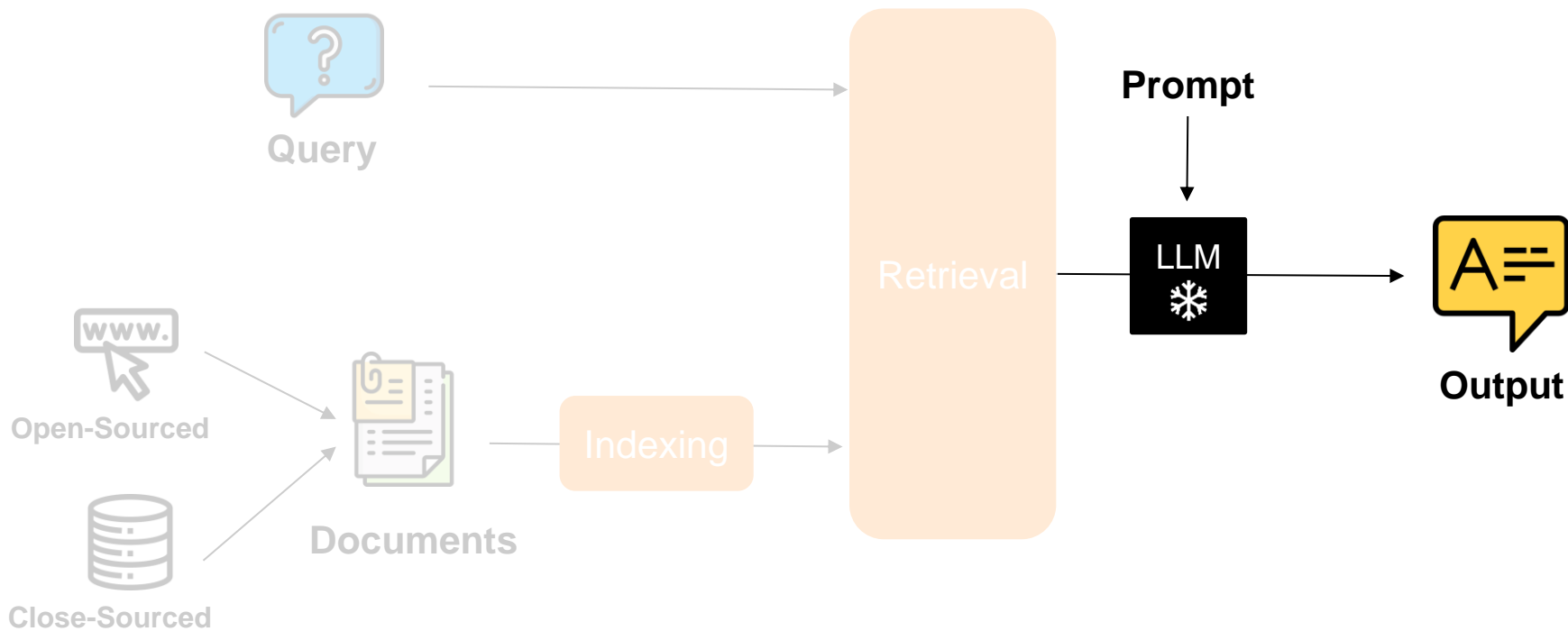
(Huang & Chen, 2024)

Performance on XOR-TYDI QA



Training with query likelihood outperforms training with paired data

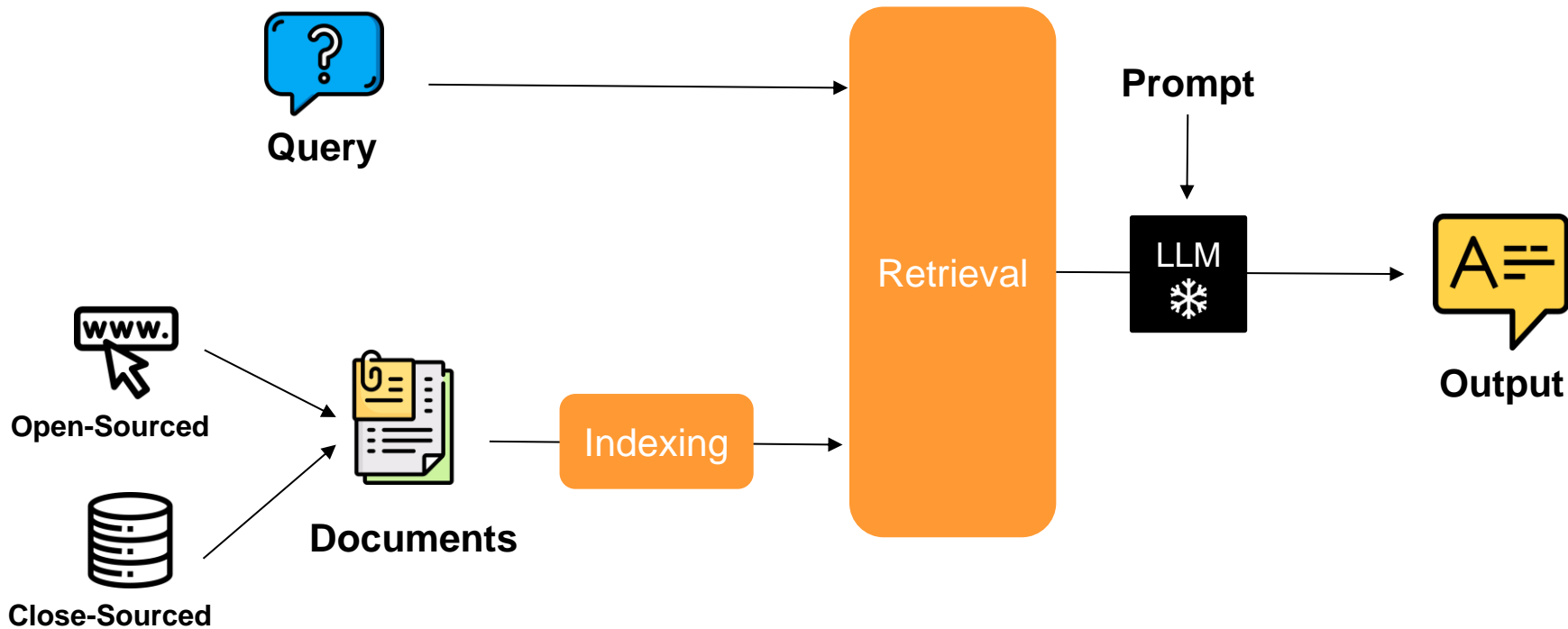
RAG Framework



Prompt Example for RAG

```
def question_answering(context, query):  
    prompt = f"""  
        Give the answer to the user query delimited by triple backticks ```{query}```\  
        using the information given in context delimited by triple backticks ```{context}```.  
  
        If there is no relevant information in the provided context, try to answer yourself,  
        but tell user that you did not have any relevant context to base your answer on.  
        Be concise and output the answer of size less than 80 tokens.  
        """  
  
    response = get_completion(instruction, prompt, model="gpt-3.5-turbo")  
    answer = response.choices[0].message["content"]  
    return answer
```

RAG Framework



Issues in RAG

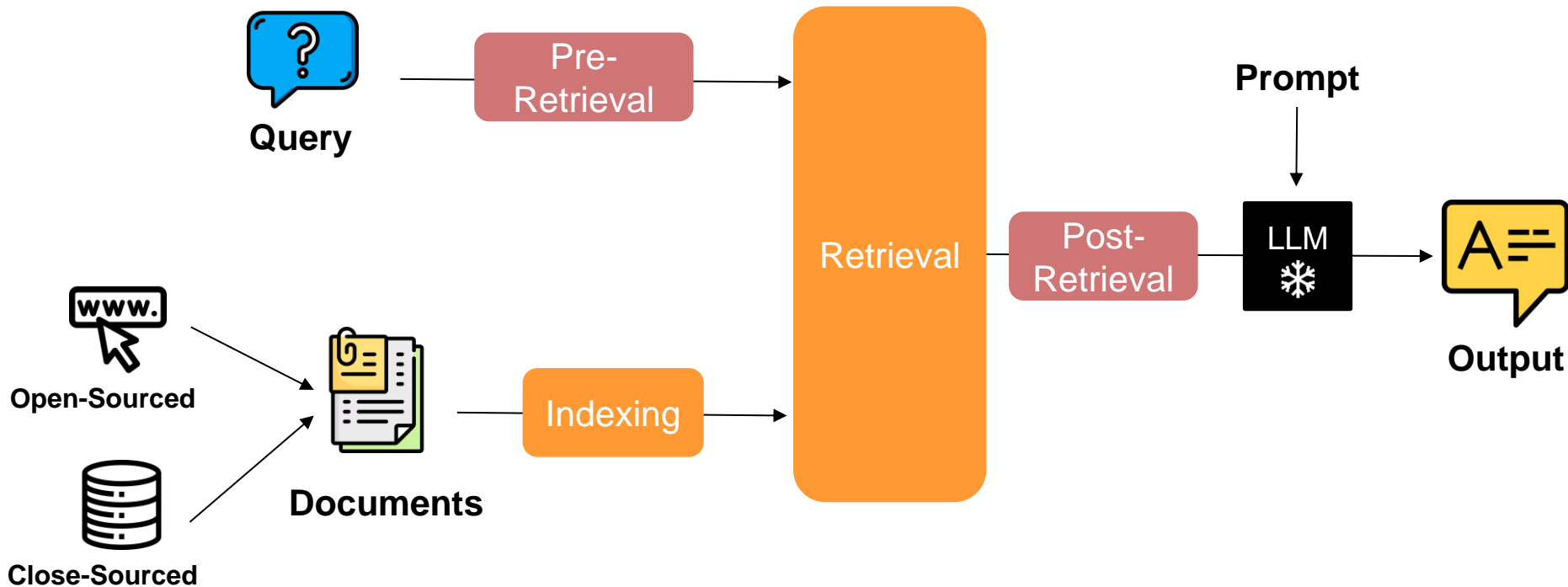
○ Retrieval

- Misaligned or irrelevant chunks
- Redundant information from multiple sources

○ Generation

- Hallucinations not supported by the retrieved context
- Suffering from irrelevance, toxicity, or bias in the output
- Overly relying on retrieved information and simply echoing it

Advanced RAG Framework



○ Pre-retrieval

- **Query rewriting**: make query suitable for the retrieval task
- **Context enrichment**: expand more chunks for LLM consideration
- **Fusion retrieval / hybrid search**: sparse and dense retrieval in parallel

○ Post-retrieval

- **Reranking**: relocate the most relevant content
- **Context compressing**: select the essential information and shorten the context

Pointwise vs. Pairwise Reranking

- Pairwise reranking is more accurate

Rate the relevance of the query and the context with a score from 1 to 5, where 1 means “completely irrelevant” and 5 means “completely relevant”.

Query: {query}

Context: {context}

Score:

3

Pointwise Reranking

Which context is more relevant to the query (A or B)?

Query: {query}

Context A: {contextA}

Context B: {contextB}

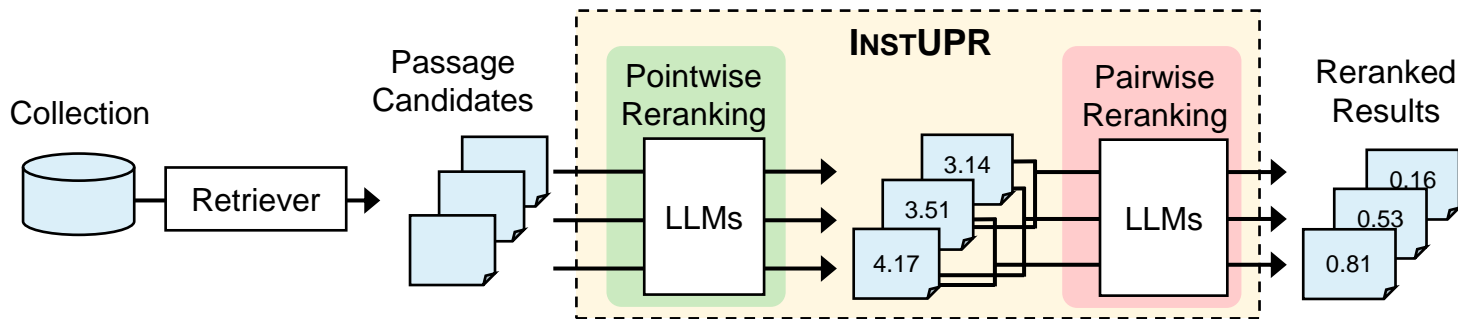
A

Pairwise Reranking

Zero-shot Instruction-based Reranking

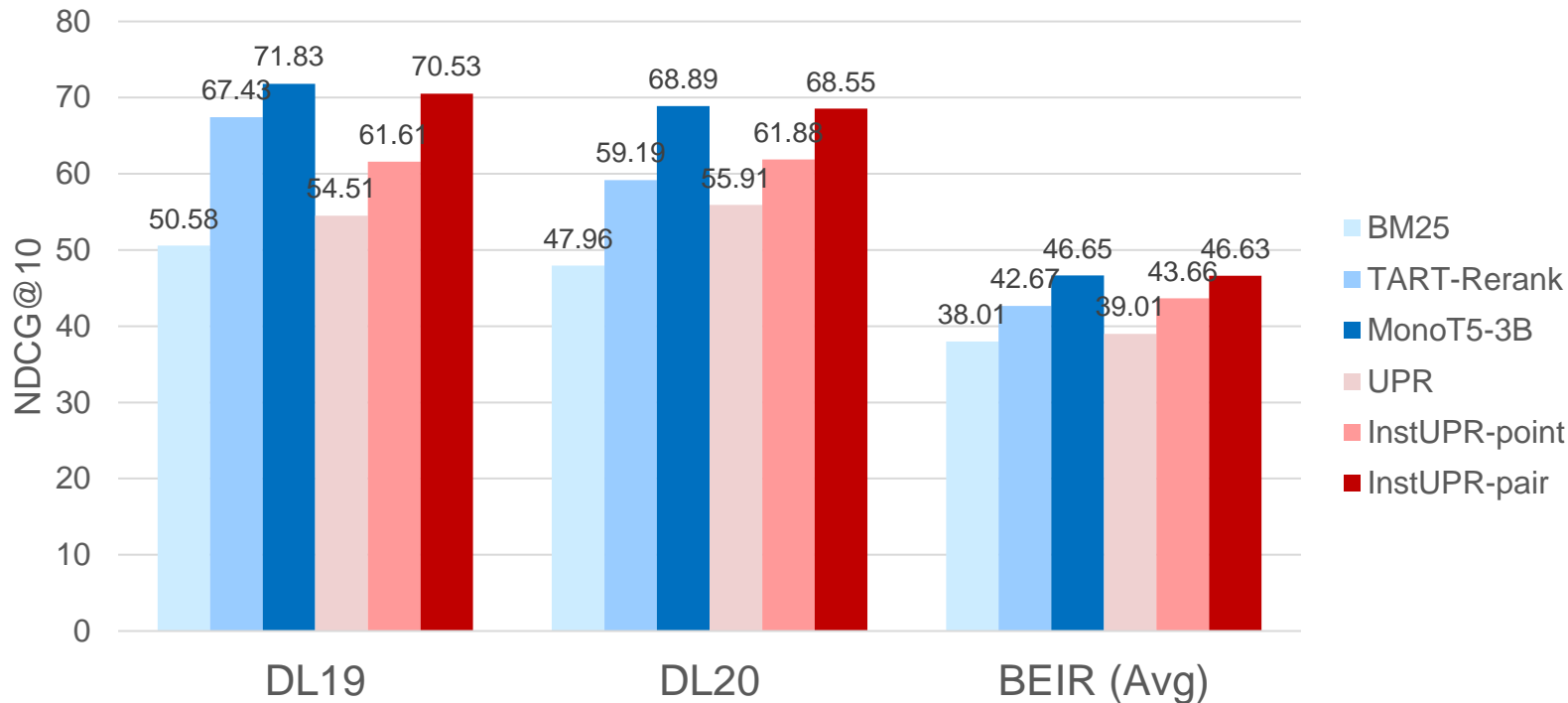
(Huang & Chen, 2024)

- LLMs can follow instructions to perform diverse tasks
- Idea: LLMs for passage reranking



Zero-shot Instruction-based Reranking

(Huang & Chen, 2024)



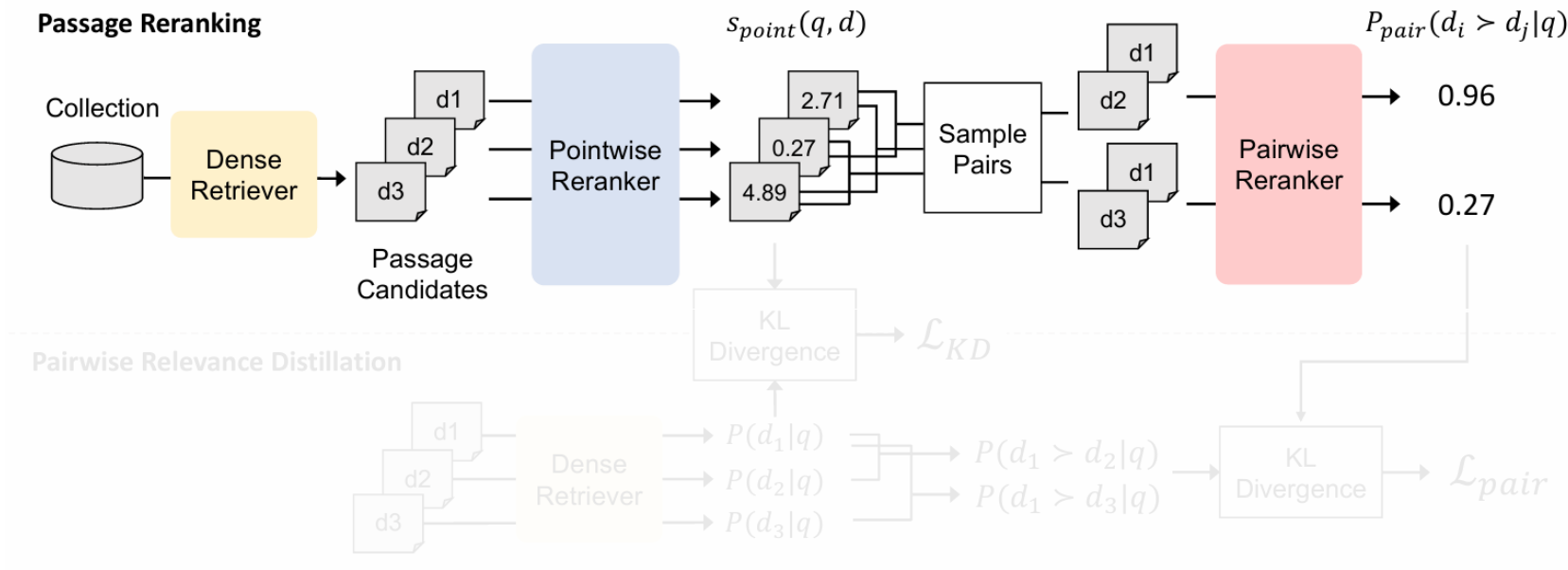
Zero-shot instruction-based reranking performs comparably with supervised rerankers

PAIRDISTILL: Reranking-Improved Retriever

(Huang & Chen, 2024)

- Idea: iteratively *train the retriever* w/ loss from the previous iteration

$$\mathcal{L} = \mathcal{L}_{CL} + \lambda_{KD} \cdot \mathcal{L}_{KD} + \lambda_{pair} \cdot \mathcal{L}_{pair}$$



ITER-RETGEN: Iterative Retrieval & Generation

(Shao+. 2023)

- Idea: concatenate *previous output* with the query for the next retrieval

Question: q

What is the height of the player who won the 2015 AFL Rising Star award?

Retrieval: $q \rightarrow \mathcal{D}_q$

Title: 2015 AFL Rising Star Context: The NAB AFL Rising Star award is given annually to a stand out young player in the Australian Football League. **The award was won by Jesse Hogan of Melbourne**

Retrieval-Augmented Generation: $\mathcal{D}_q \parallel q \rightarrow y_1$

The 2015 AFL Rising Star award was won by Jesse Hogan of Melbourne. Jesse Hogan is a professional Australian rules footballer. He is **198 cm tall**. So the answer is 198 cm

Iteration 1

Question: q

What is the height of the player who won the 2015 AFL Rising Star award?

Generation-Augmented Retrieval: $y_1 \parallel q \rightarrow \mathcal{D}_{y_1 \parallel q}$

Title: Jesse Hogan Context: Jesse Hogan ... playing for the Melbourne Football Club. A key forward, **Hogan is 1.95 m tall** ... made his AFL debut in the 2015 season and won the Ron Evans Medal as the AFL Rising Star

Retrieval-Augmented Generation: $\mathcal{D}_{y_1 \parallel q} \parallel q \rightarrow y_2$

The 2015 AFL Rising Star award was won by Jesse Hogan of Melbourne. Jesse Hogan is **1.95 m tall**. So the answer is 1.95 m

Iteration 2

30

RAG Roadmap

What, How, When to Retrieve?

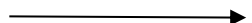
Roadmap of RAG Evolution

What to
retrieve?

How to use
retrieval?

When to
retrieve?

Text Chunks



Input layer



At beginning of generation (Once)

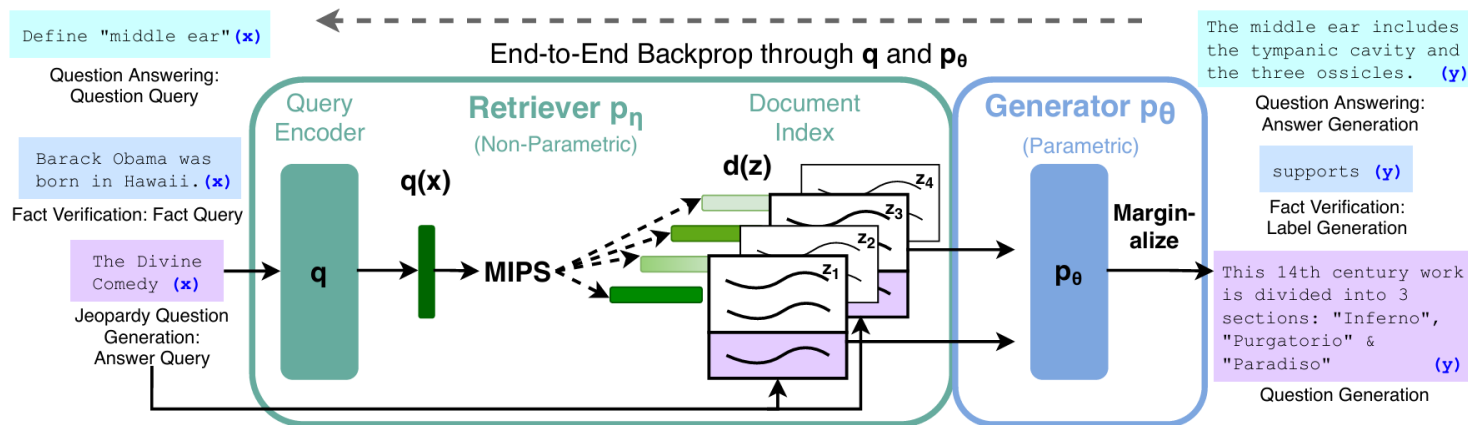
RAG (Lewis+, 2020)

End-to-End RAG (Lewis+, 2020)

Idea: train a retriever and a generator to improve performance

- Retriever:** maximize overall likelihood by optimizing *mixture weights* over documents
- Generator:** maximize generation likelihood given single retrieved document

$$P_{\text{RAG}}(y|x) \approx \prod_i \sum_{z \in \text{top-k}(p(\cdot|x))} \underbrace{p_{\eta}(z|x)}_{\text{Retriever}} \underbrace{p_{\theta}(y_i|x, z, y_{1:i-1})}_{\text{Generator}}$$



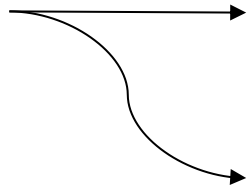
Roadmap of RAG Evolution

What to
retrieve?

How to use
retrieval?

When to
retrieve?

Text Chunks



Input layer

Intermediate layers

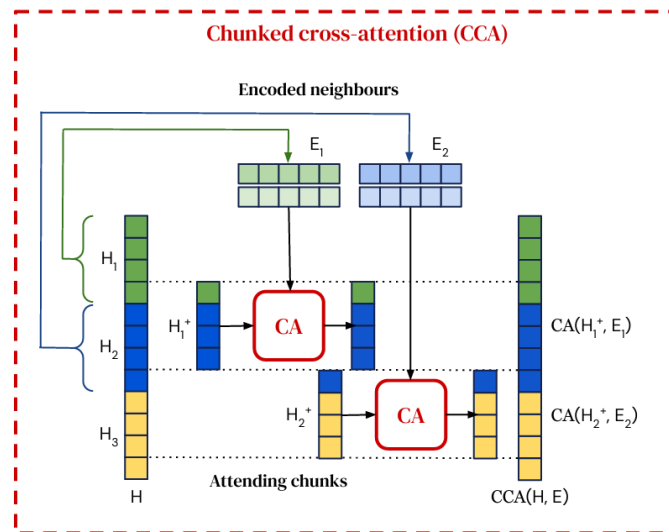
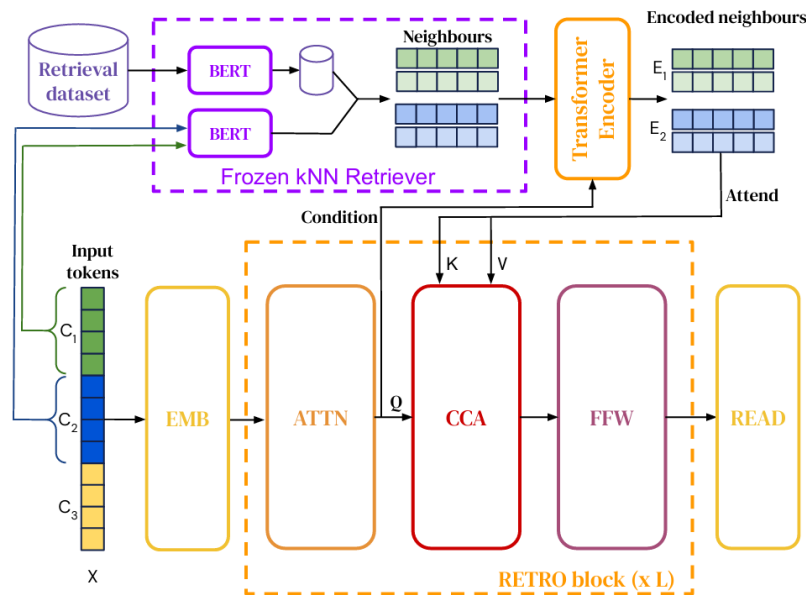
RETRO (Borgeaud+, 2022)

At beginning of generation (Once)

RAG (Lewis+, 2020)

RETRO (Borgeaud+, 2022)

- Idea: inject retrieved information in the *intermediate* layers



More efficient than injecting in the input layer → training required

Roadmap of RAG Evolution

What to
retrieve?

How to use
retrieval?

When to
retrieve?

Text Chunks



Input layer



At beginning of generation (Once)

RAG (Lewis+, 2020)



Intermediate layers

RETRO (Borgeaud+, 2022)

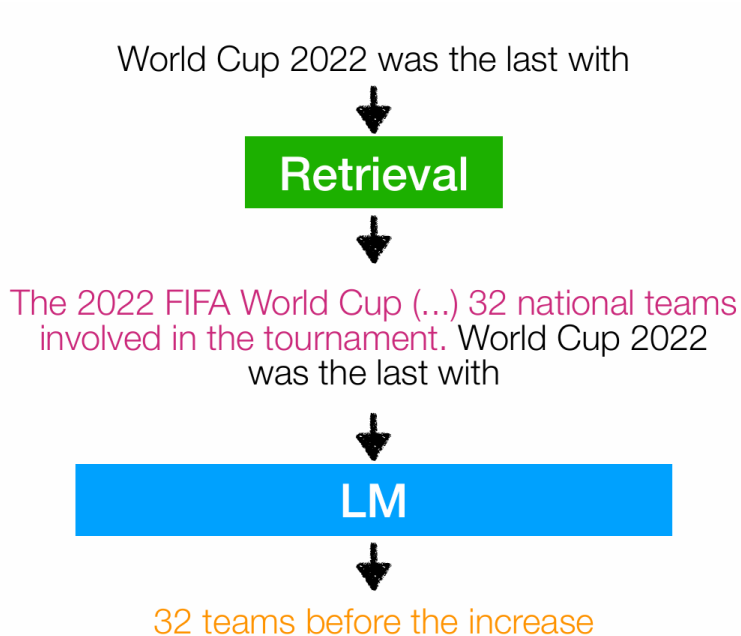


Every n tokens

Retrieve-in-context
(Ram+, 2023; Shi+, 2023)

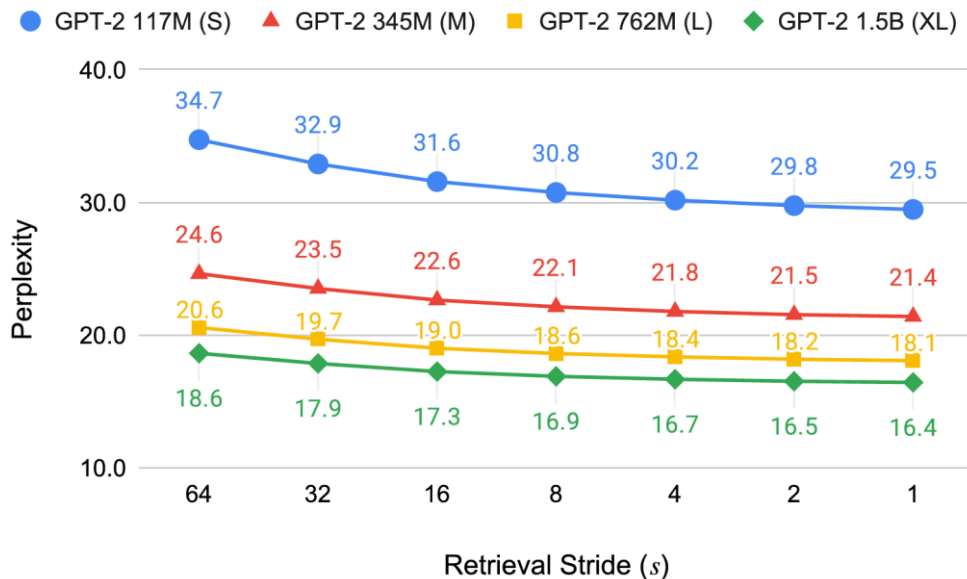
Retrieval-in-context LM (Ram+, 2023; Shi+, 2023)

- Idea: perform retrieval *every n tokens*



Retrieval-in-context LM (Ram+, 2023; Shi+, 2023)

- Idea: perform retrieval *every n tokens*



Retrieving more frequently helps → higher cost in inference (slower)

Roadmap of RAG Evolution

What to
retrieve?

How to use
retrieval?

When to
retrieve?

Text Chunks



Input layer



At beginning of generation (Once)

RAG (Lewis+, 2020)



Intermediate layers

RETRO (Borgeaud+, 2022)



Every n tokens

Retrieve-in-context
(Ram+, 2023; Shi+, 2023)

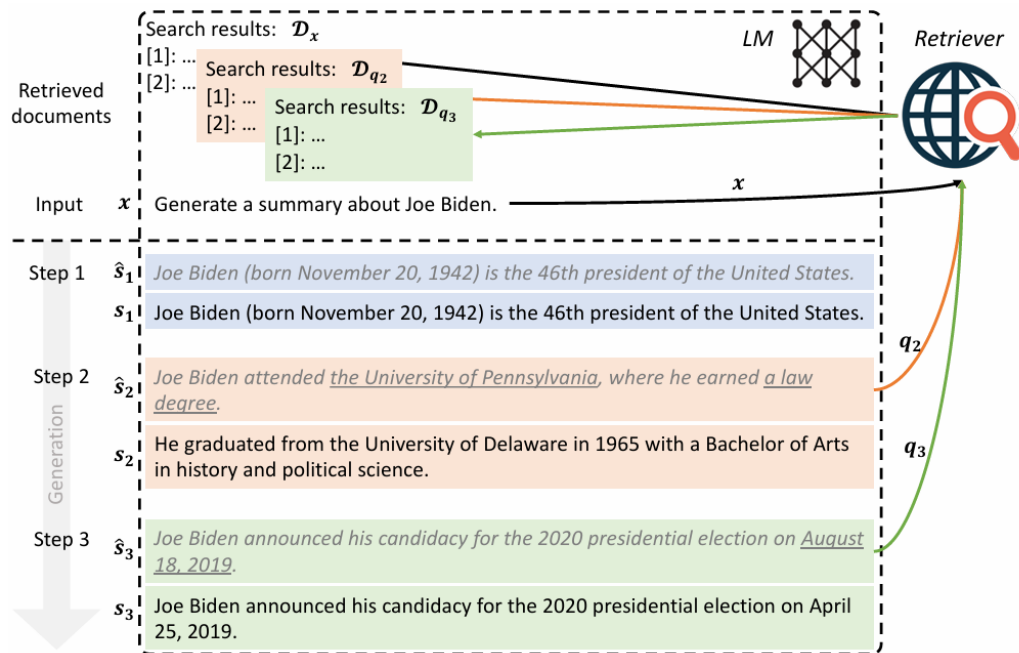


Adaptively

FLARE
(Jiang+, 2023)

Triggering Retrieval w/ Uncertainty

- FLARE (Jiang+, 2023) generates content first then retrieves if LM *certainty is low*

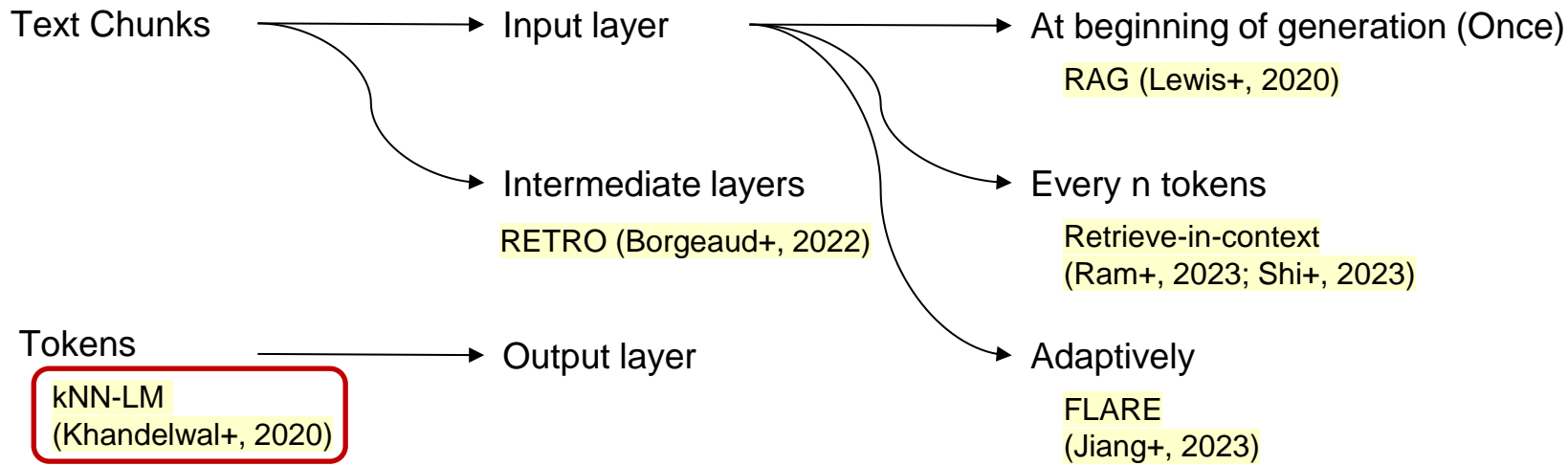


Roadmap of RAG Evolution

What to
retrieve?

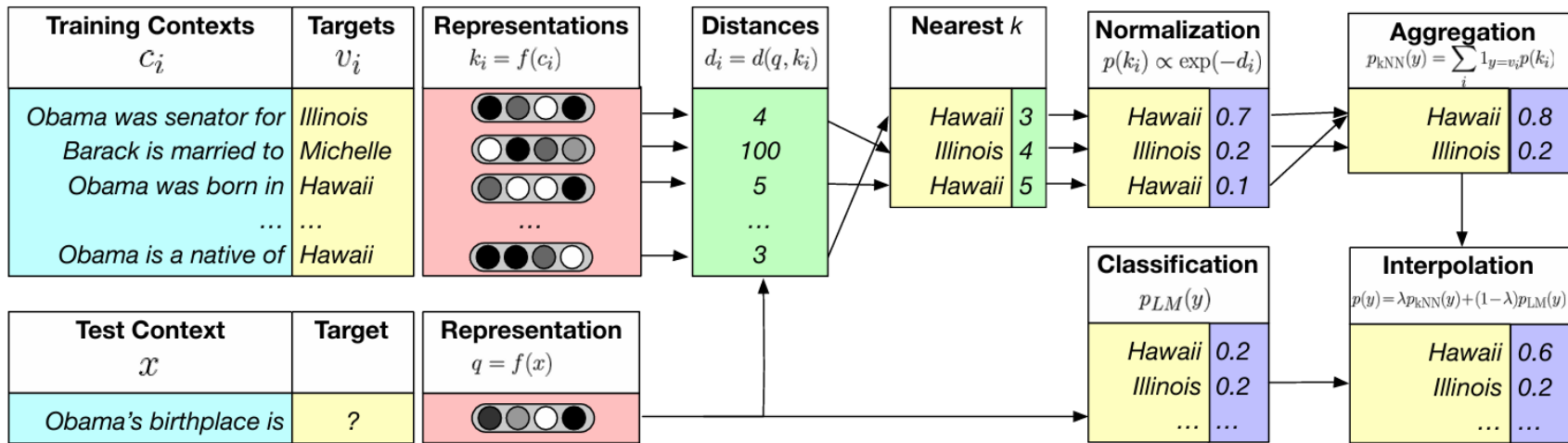
How to use
retrieval?

When to
retrieve?



Token-Level Softmax Modification

- kNN-LM (Khandelwal+, 2020) retrieves *similar examples* and uses the following token from them



More fine-grained, compute-efficient → space-expensive

Roadmap of RAG Evolution

What to
retrieve?

How to use
retrieval?

When to
retrieve?

Text Chunks



Input layer



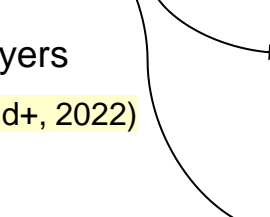
At beginning of generation (Once)

RAG (Lewis+, 2020)



Intermediate layers

RETRO (Borgeaud+, 2022)



Every n tokens

Retrieve-in-context
(Ram+, 2023; Shi+, 2023)

Tokens



Output layer



Adaptively

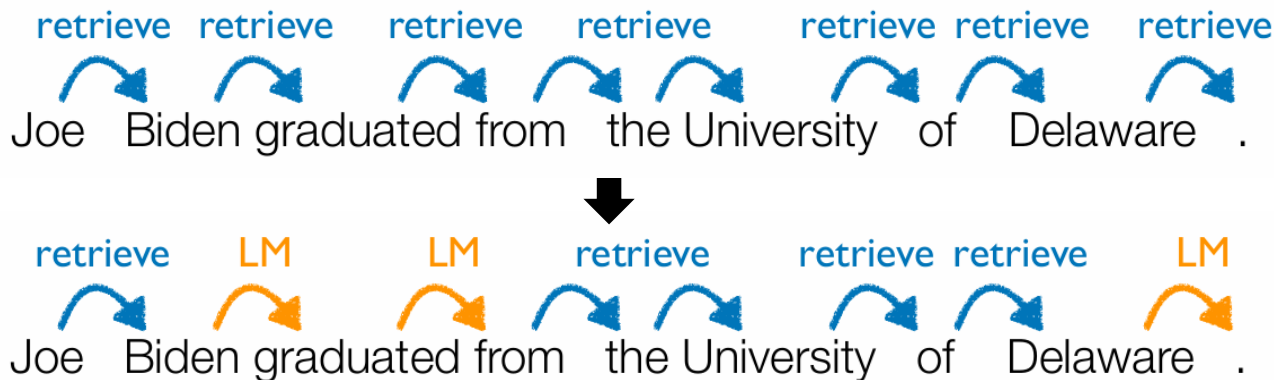
kNN-LM
(Khandelwal+, 2020)

Efficient NN-LM
(He+, 2021)

FLARE
(Jiang+, 2023)

Token-Level Adaptive Retrieval

- kNN-LM requires retrieval for each token → inefficient
- He+, 2021 improved efficiency by *adaptive retrieval*



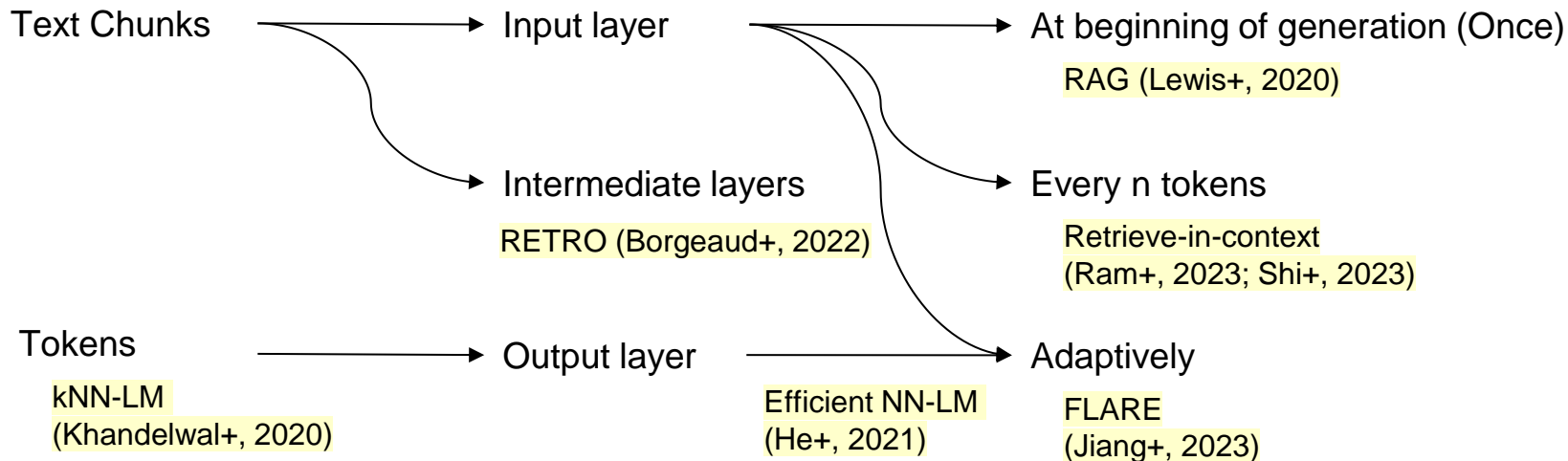
$$P_{k\text{NN-LM}}(y | x) = (1 - \lambda(x))P_{\text{LM}}(y | x) + \lambda(x)P_{k\text{NN}}(y | x)$$

Roadmap of RAG Evolution

What to
retrieve?

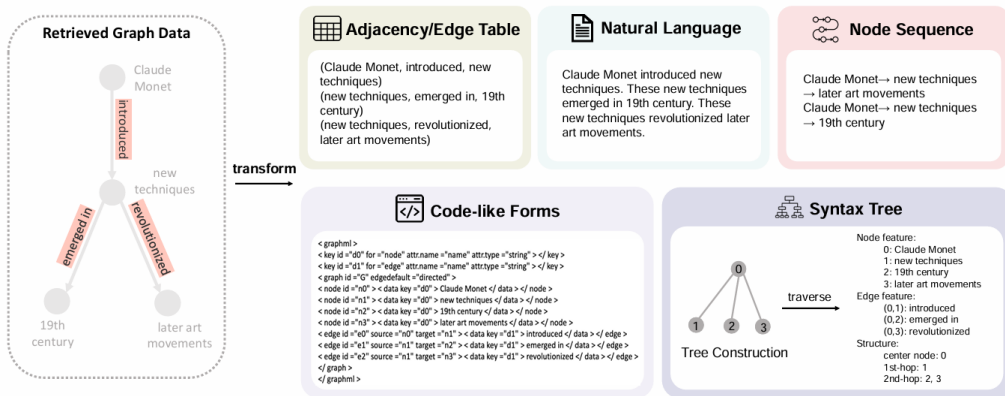
How to use
retrieval?

When to
retrieve?

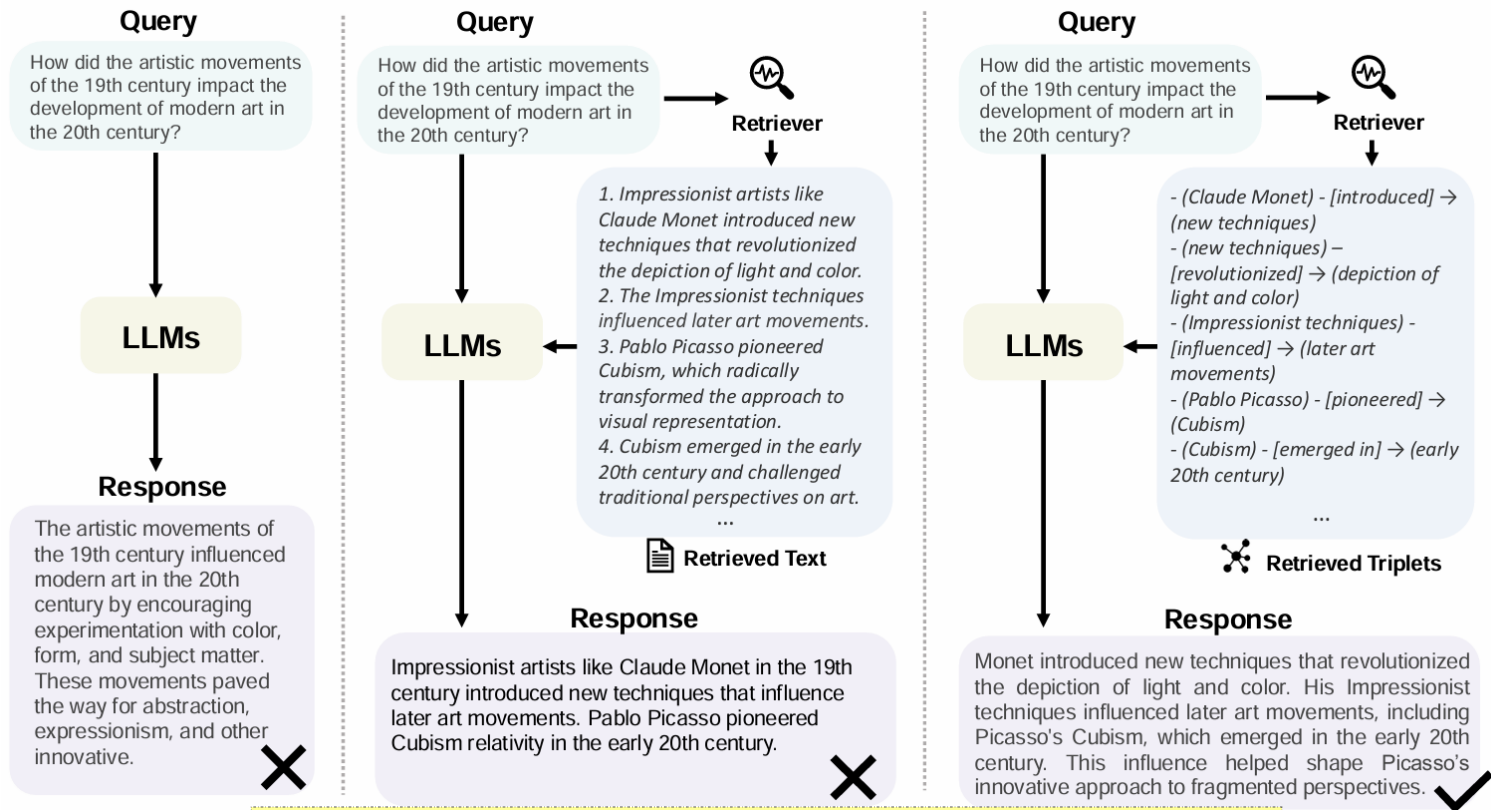


Retrieval Sources

- **Unstructured** data: text (web or in-house)
- **Semi-structured** data: text + table (e.g. PDF)
 - Issues: data corruption, retrieve information from tables
 - Methods: transform table into text and use text-only RAG → suboptimal
- **Structured** data: knowledge graphs (KGs)
 - Search related entities and relations to form the context

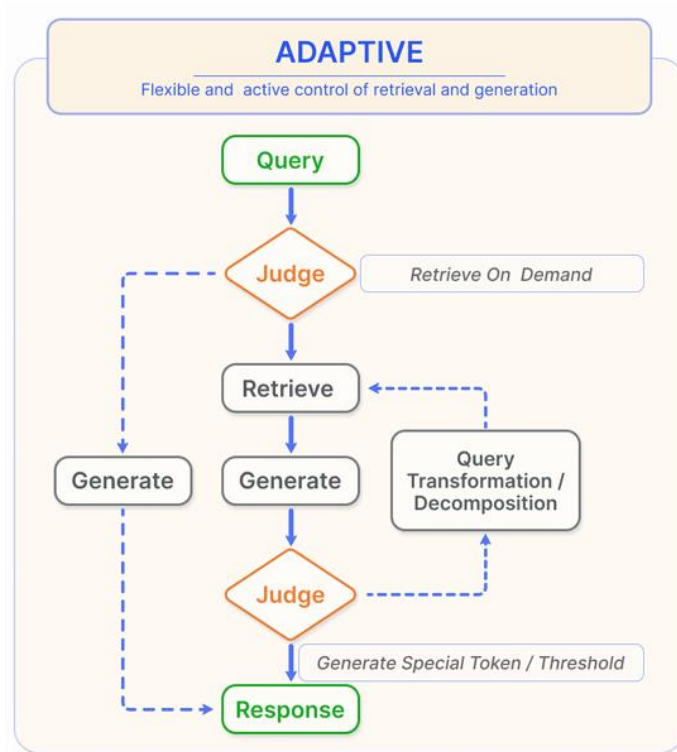
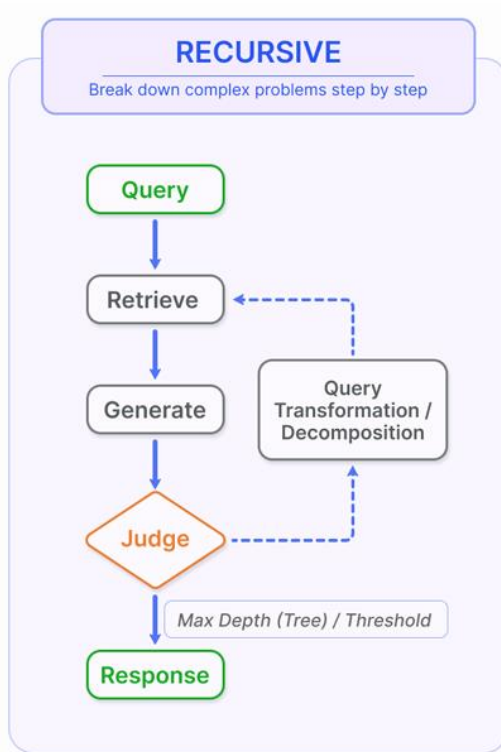
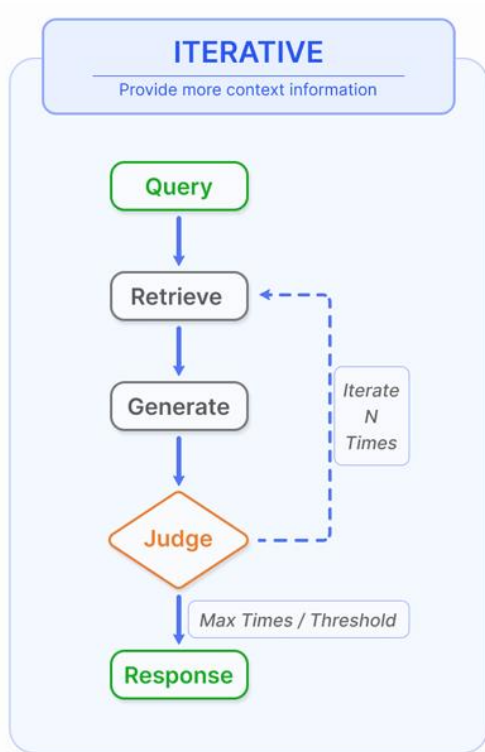


GraphRAG (Peng+, 2024)

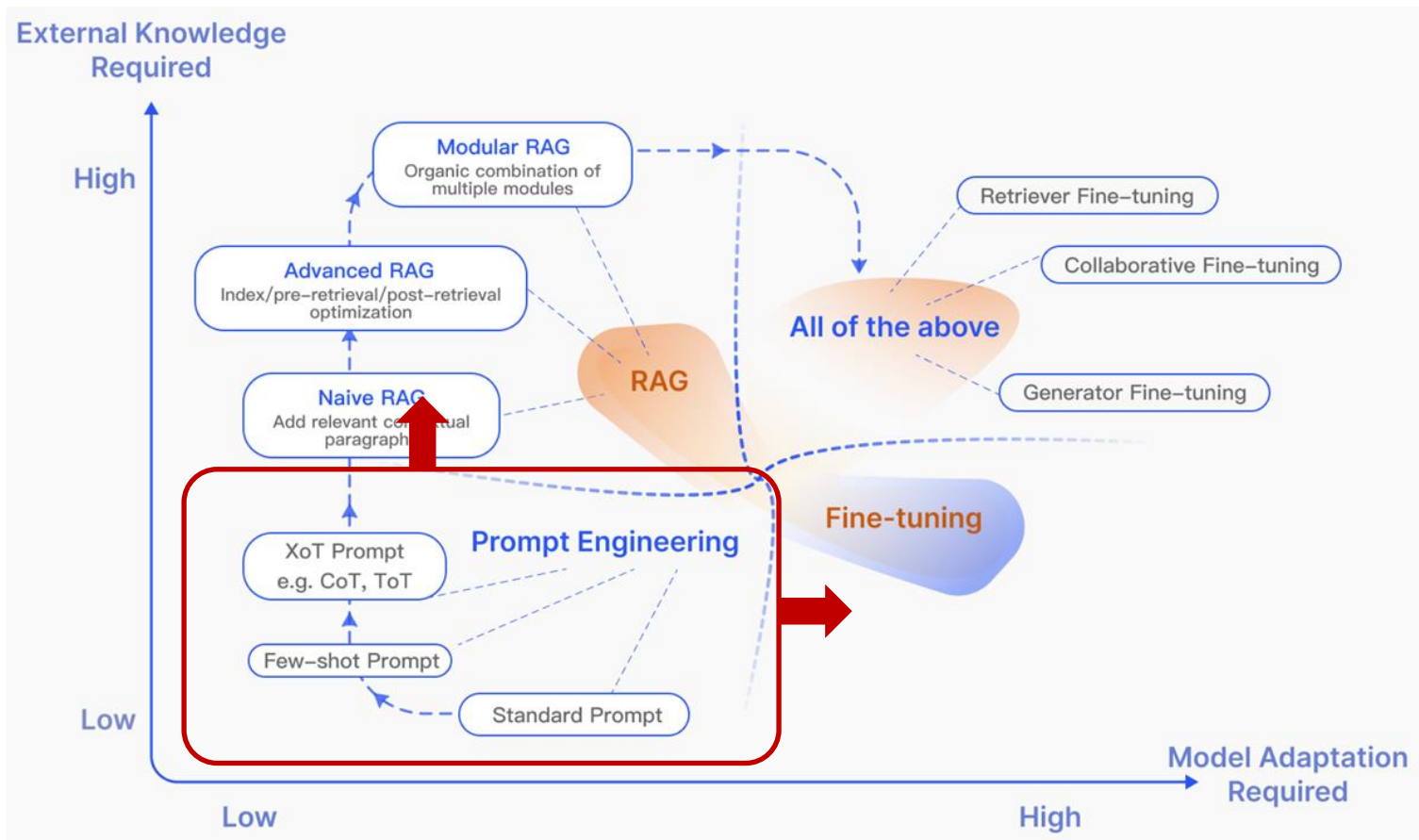


KG can better help reasoning-intensive tasks

Three Types of RAG (Gao+, 2023)



Prompting vs. RAG vs. Fine-Tuning (Gao+, 2023)



Prompting vs. RAG vs. Fine-Tuning (Gao+, 2023)

檢察官甚麼情況可以申請羈押？



CS Student

- **Prompting:** directly answering it based on the internal knowledge
- **RAG:** checking legal *textbooks* and answering it based on them
- **Fine-tuning:** taking a related *course* and then answering it

Pros and Cons

○ Prompting

- Pros: simple, efficient
- Cons: fully relying on the internal knowledge

○ RAG

- Pros: suitable for dynamic environments, high interpretability
- Cons: high latency, retrieval quality

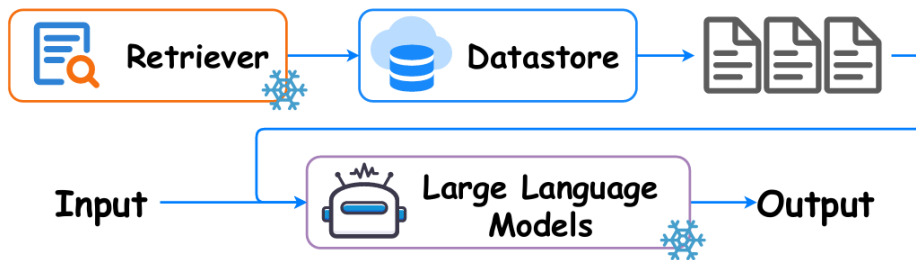
○ Fine-tuning

- Pros: deep customization of the model's behavior and style
- Cons: high cost due to retraining

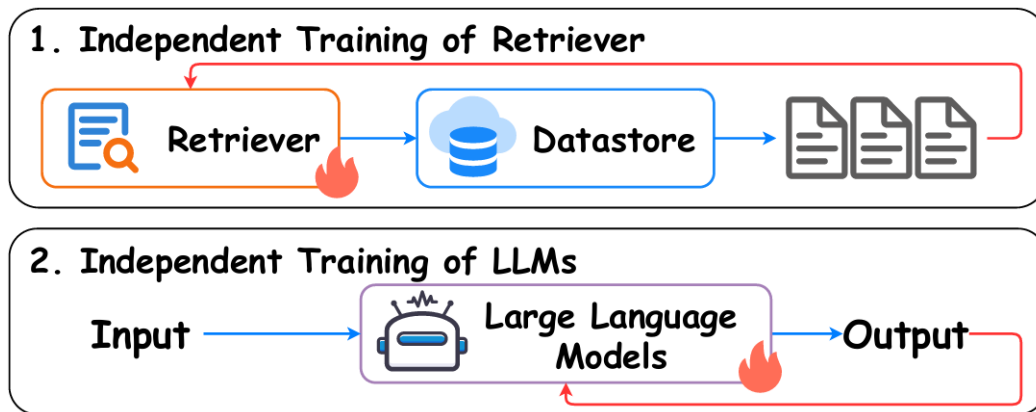
Choosing between RAG and FT depends on **data dynamics**, **customization**, and **computational capabilities** in the application context.

RA-LLM Training

● Training-free

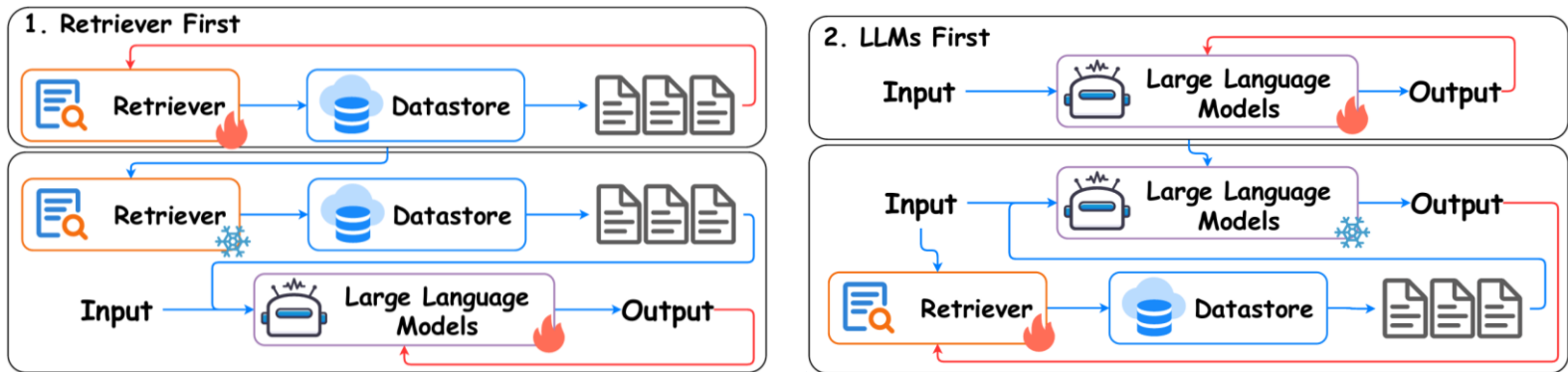


Independent training

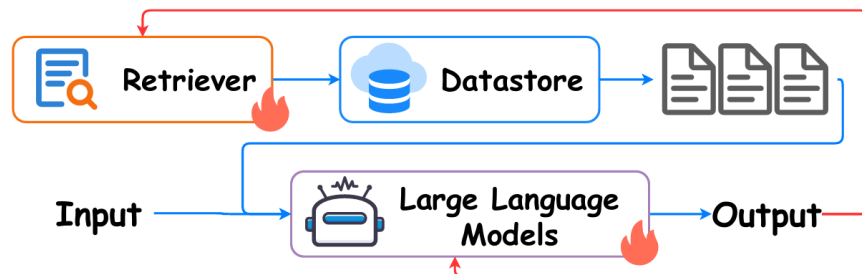


RA-LLM Training

Sequential training



Joint training



RAG Applications

- RAG are better for knowledge-intensive tasks

Question Answering

Fact Verification

Dialogue

- General NLP tasks (generation)

Summarization

Machine Translation

Code & Proof Generation

- General NLP tasks (classification)

NLI

Sentiment Analysis

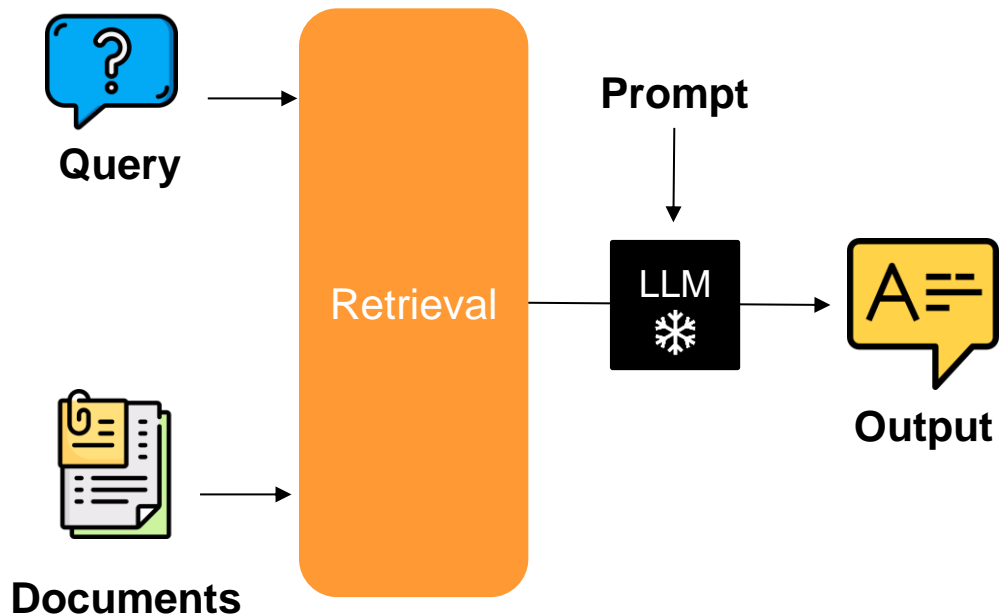
Commonsense Reasoning

RAG Advantages

- **Accuracy:** RAG reduces hallucinations
- **Up-to-date:** RAG can access the latest information
- **Interpretability / Reliability:** RAG cites references for fact checking
- **Customization:** RAG can utilize knowledge from specific domains
- **Safety / Privacy:** RAG uses external data
- **Cost-efficiency:** No model updating in RAG

Concluding Remarks

- When to use RAG?
 - Long-tail knowledge
 - Knowledge update
 - Interpretability
 - Safety / Privacy



References

- [ACL 2023 Tutorial: Retrieval-based Language Models and Applications](#)
- [CMU Advanced NLP Course](#)