UNIVERSITY OF OXFORD

DEPARTMENT OF COMPUTER SCIENCE

Retrieval-Augmented Generation for Large Language Models

GenAI BootCamp

YUAN HE | AUG 2024

What is Language Model?

Language Model

- A statistical model that determines the **probability distribution** of **linguistic units** (words, sentences, etc.) in a language.

- **Sequential** language modelling: $P(w_i|w_{<i})$
  - E.g., London is the capital of the UK

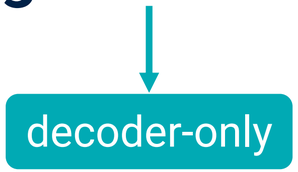
- **Masked** language modelling: $P(w_i|w_{<i}, w_{>i})$
  - E.g., London is the [MASK] of the UK.

Large Language Model

- “Large” is a relative term

- Often consider **GPT-3 (2020)** as one of the first LLMs
  - GPT-3 scales up 100x parameters compared to its predecessors

- Often assumed as **generative** – sequential language modelling



Prompt for LLMs

- Recall sequential language modelling: $P(w_i|w_{<i})$
- Prompt is the conditioning text c: $P(w_i|c, w_{<i})$
- Prompts can be queries, instructions, and auxiliary contexts, e.g.,
  - "What is a prompt in the context of generative AI?"
  - "Translate the following into Chinese: [TEXT] "
  - "Given [EHR] , what would be the best course of treatment for managing their newly diagnosed chronic kidney disease?"

LLMs are powerful, but

Challenges with LLMs

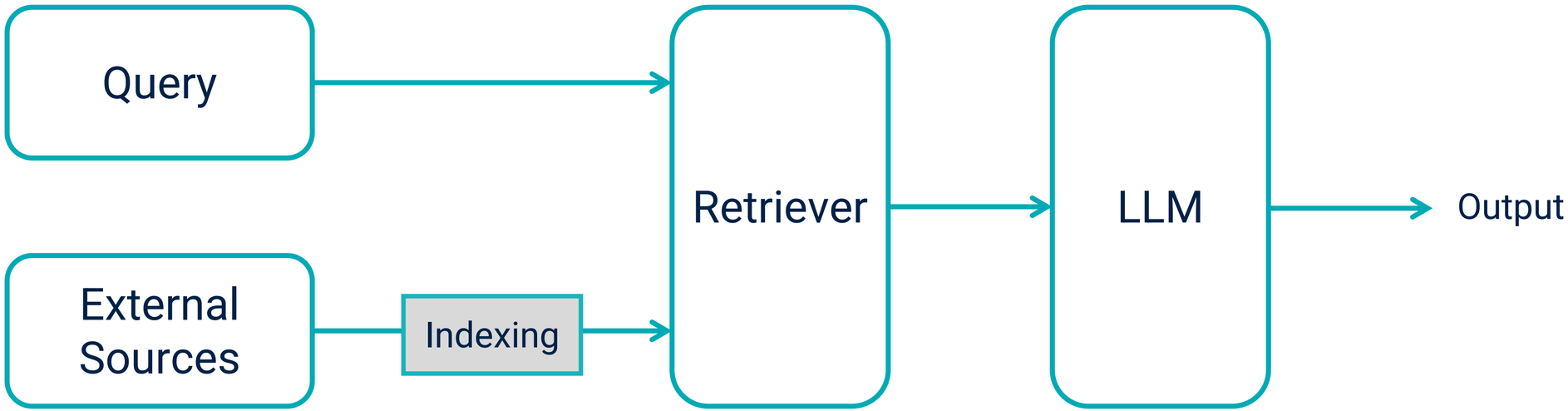
- **Hallucination**
  - LLMs tend to produce fabricated information when they don’t know

- **Data Privacy Concerns**
  - Training on proprietary data (e.g., EHR, company data) raises privacy and security issues; we prefer to avoid it

- **Adaptability**
  - Difficulties in incorporating new knowledge on the fly make LLMs less responsive to real-time changes

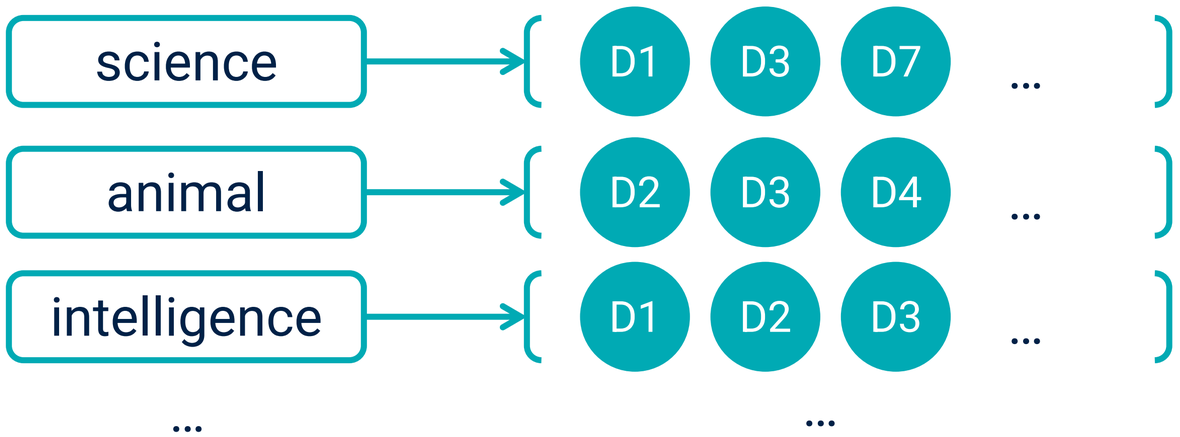
RAG is a possible solution

Retrieval-Augmented Generation (RAG)



Indexing

- **Inverted Index (Sparse Retrieval)**
  - Building an Inverted Index to store token-documents pairs
  - Retrieve through statistical algorithms like TF-IDF, BM25

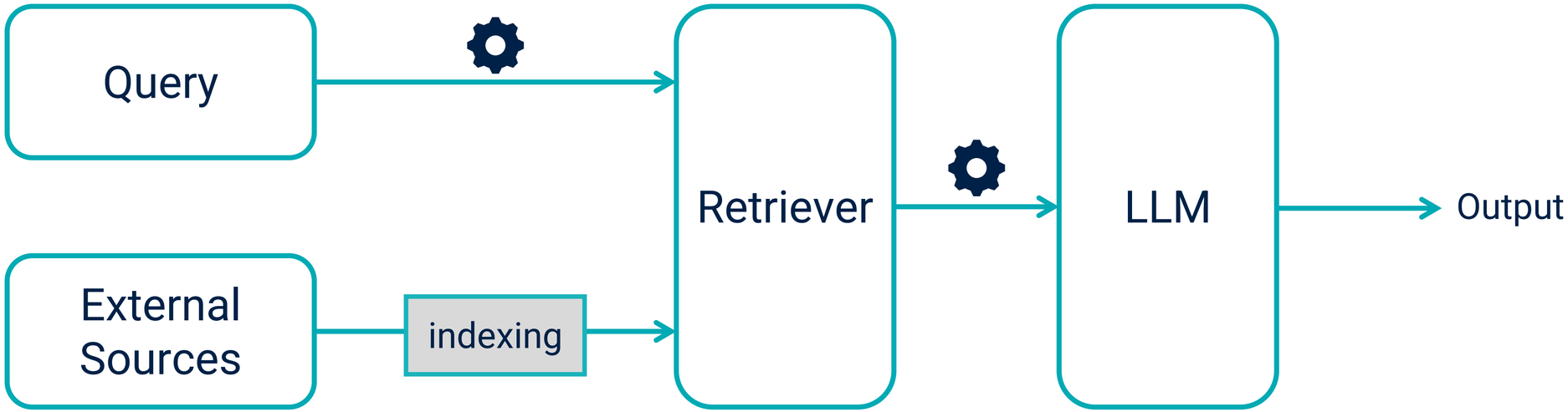


Indexing

- **Vector Database (Dense Retrieval)**
  - Building a Vector Database to store document embeddings
  - Retrieve through machine learning algorithms like KNN, ANN

Can we do any better?

RAG + Pre/Post-processing



Pre-Retrieval Processing

- **Routing**
  - Directing a query to the most appropriate source
  - Or even not using a retriever

- **Rewriting**
  - Refine the query based on a feedback mechanism
  - Convert the query into a formal query (for GraphRAG)

- **Expansion**
  - Incorporating meta-data, instructions, etc.

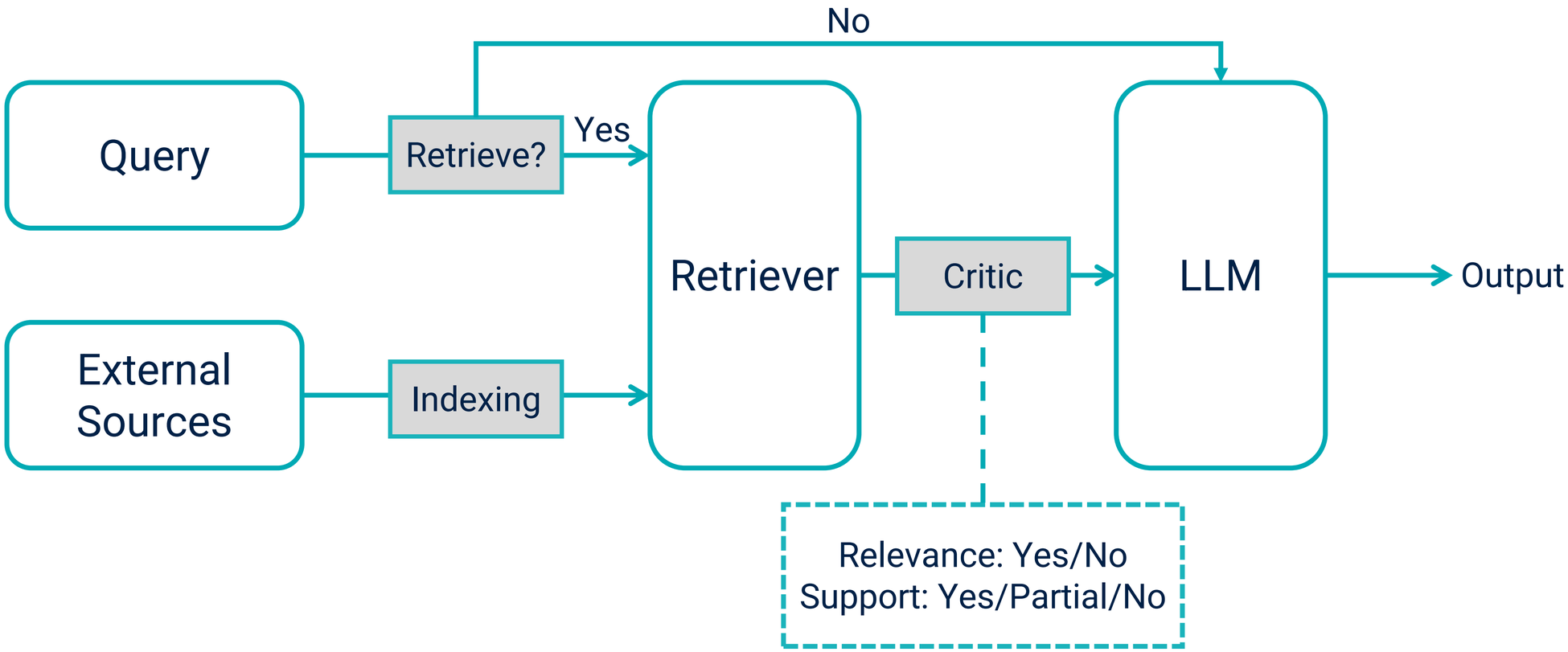
Post-Retrieval Processing

- **Reranking**
  - Adjusting the order of retrieved documents or results based on more advanced criteria (e.g., beyond similarity)

- **Summary**
  - Input prompt is too long -- generating concise summaries of the retrieved documents

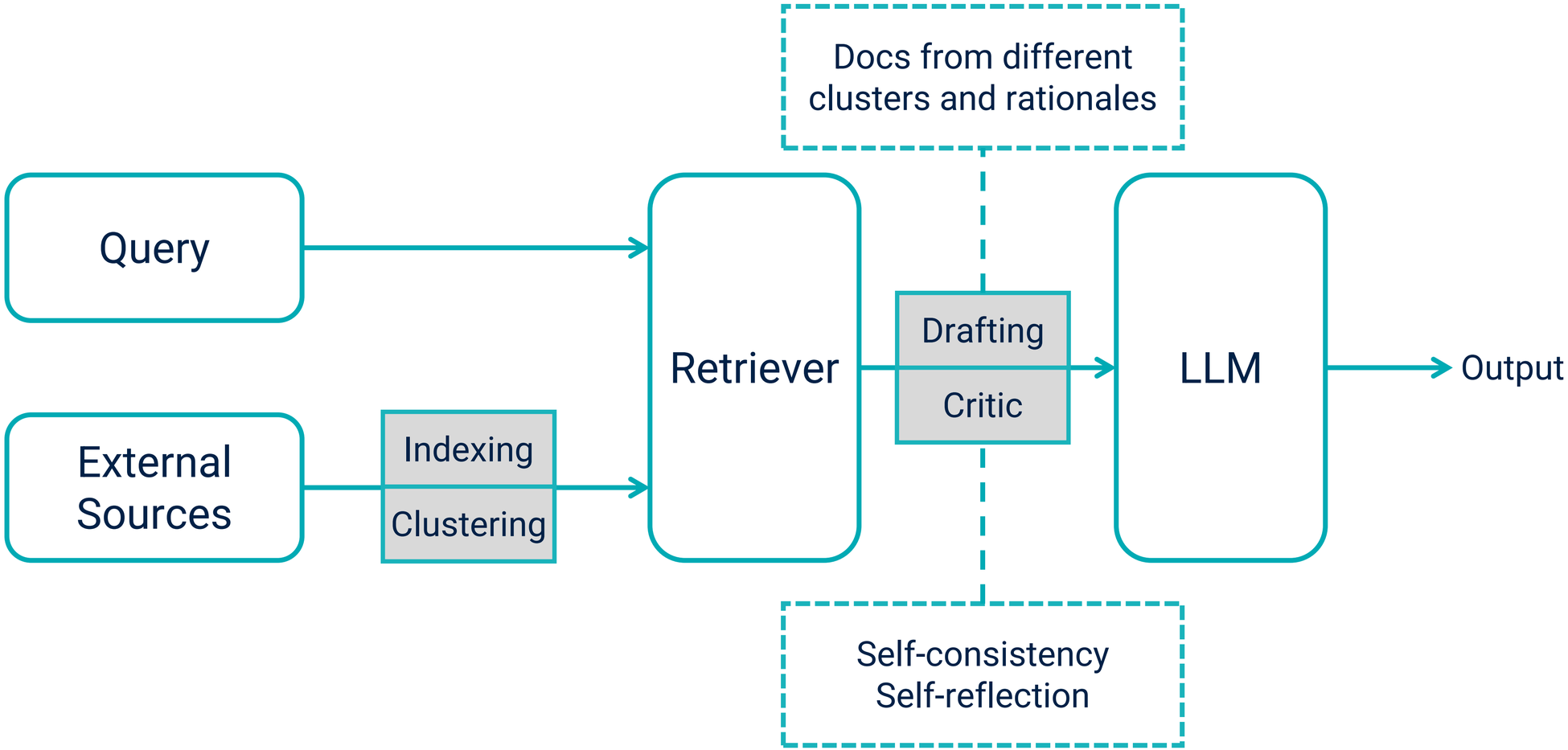
- **Fusion**
  - Merging retrieved contents from various sources or multiple queries

Self-Reflective RAG



Asai et al. “Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection”. ICLR 2024.

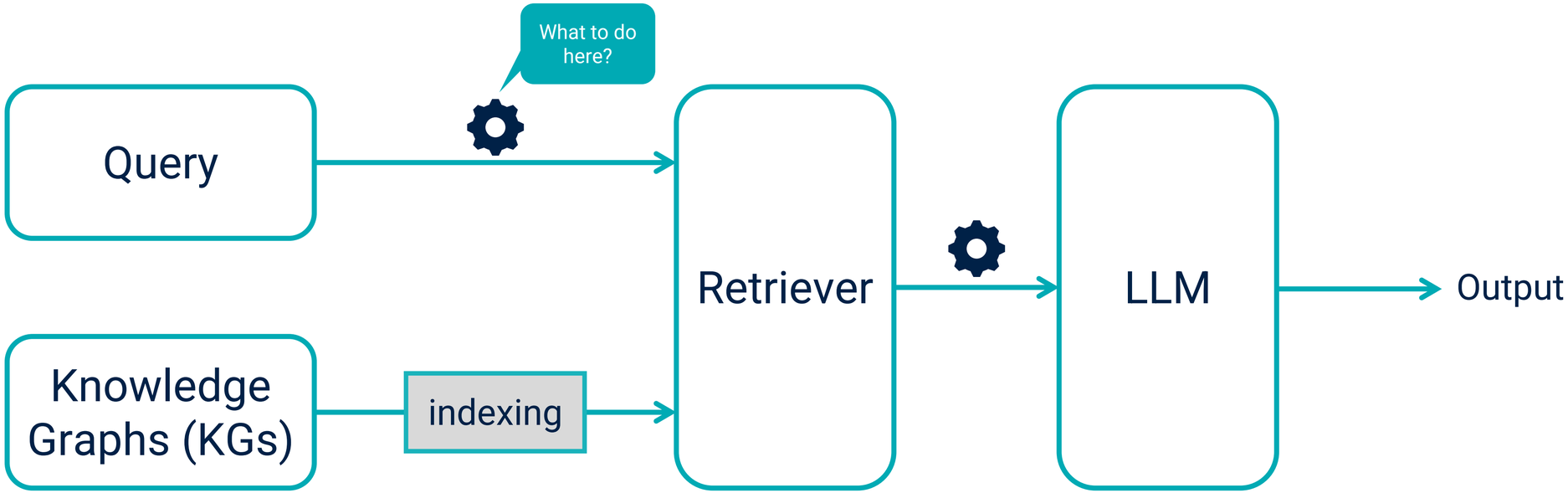
Speculative RAG



Wang et al. “Speculative RAG: Enhancing Retrieval Augmented Generation through Drafting”. Arxiv 2024.

Beyond text documents?

GraphRAG

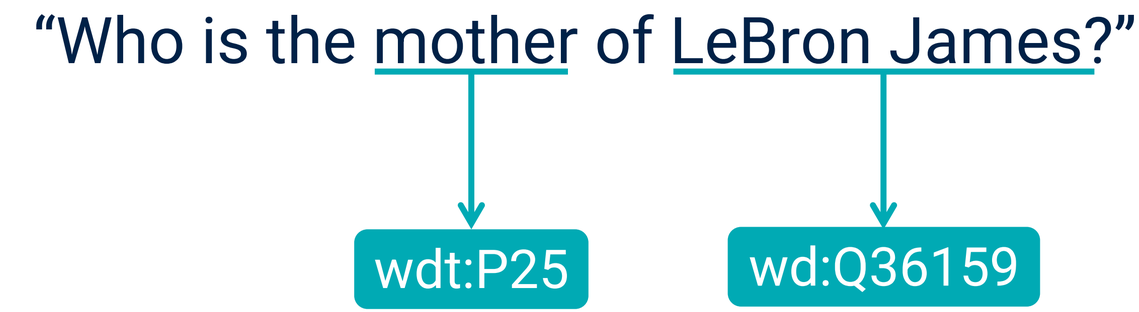

What to do here?
Query
Knowledge Graphs (KGs)
indexing
Retriever
LLM
Output

Pre-Retrieval Processing (GraphRAG)

Entity Recognition

Entity Linking

Query Rewriting




```sparql
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>

SELECT ?motherLabel
WHERE {
  wd:Q36159 wdt:P25 ?mother .
  SERVICE wikibase:label { bd:serviceParam wikibase:language "[AUTO_LANGUAGE],en".
}
```

Pre-Retrieval Processing (GraphRAG)

- **Alternative 1**
  - It is possible to train an LLM to do the whole processing altogether

- **Alternative 2**
  - Without converting the query into a formal query, we can conduct the retrieval over KG embeddings

GraphRAG vs RAG

Pros

- Better reasoning
- More accurate context
- Handle complex query

Cons

- Rely on the completeness of KGs
- Increased complexity
- Slower retrieval

Conclusion

- RAG is essentially information retrieval + generation

- RAG can help LLMs:
  - Reduce hallucination
  - Avoid expensive re-training
  - Access proprietary data without memorising them

- RAG can be improved by multi-stage processing and critic modules
  - GraphRAG is an extension of RAG to handle structured knowledge bases

THANKS!