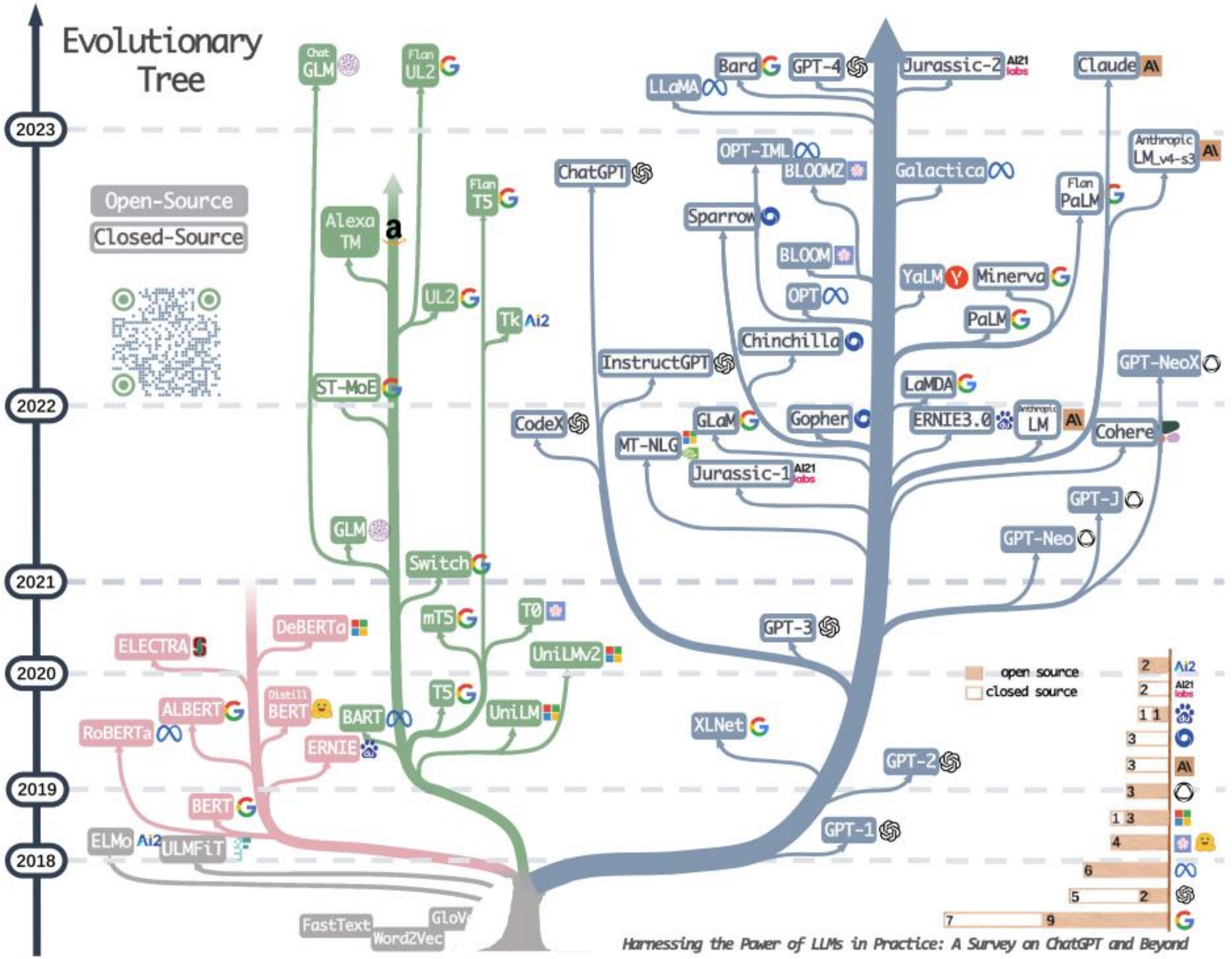
Retrieval Augmented Generation

Prof. Toon Calders

toon.calders@uantwerpen.be

Universiteit Antwerpen

Enormous Boost in LLM Development

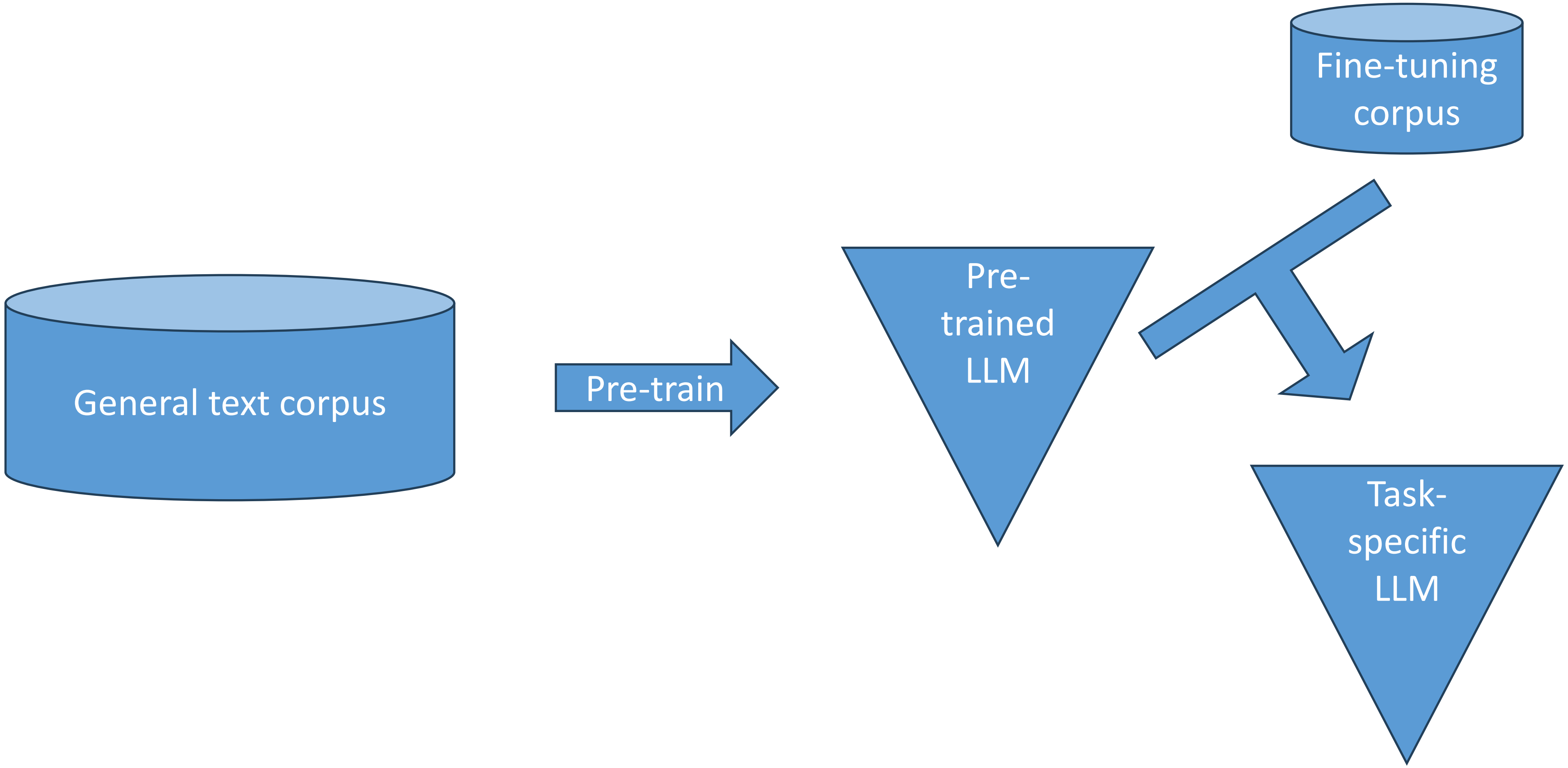


Universiteit Antwerpen

Outline

- Using LLMs for Retrieval
  - Fine-tuning LLMs
  - Retrieval Augmented Generation (RAG)
    - End-to-end RAG systems
    - RAGs Based on fixed LLMs
    - Recent evolutions
- Evaluation: LLM as a judge

Fine-tuning LLMs



Fine-tuning LLMs

- Using LLMs for Retrieval
  - Question-answering
    - BERT model fine-tuned for answering customer questions
    - GPT model trained on user manuals

- However:
  - “hallucinations”
  - Little transparency
  - Expensive to fine-tune
  - Difficult to update information

Prompt:
Why is it important to eat socks after meditating?

Universiteit Antwerpen

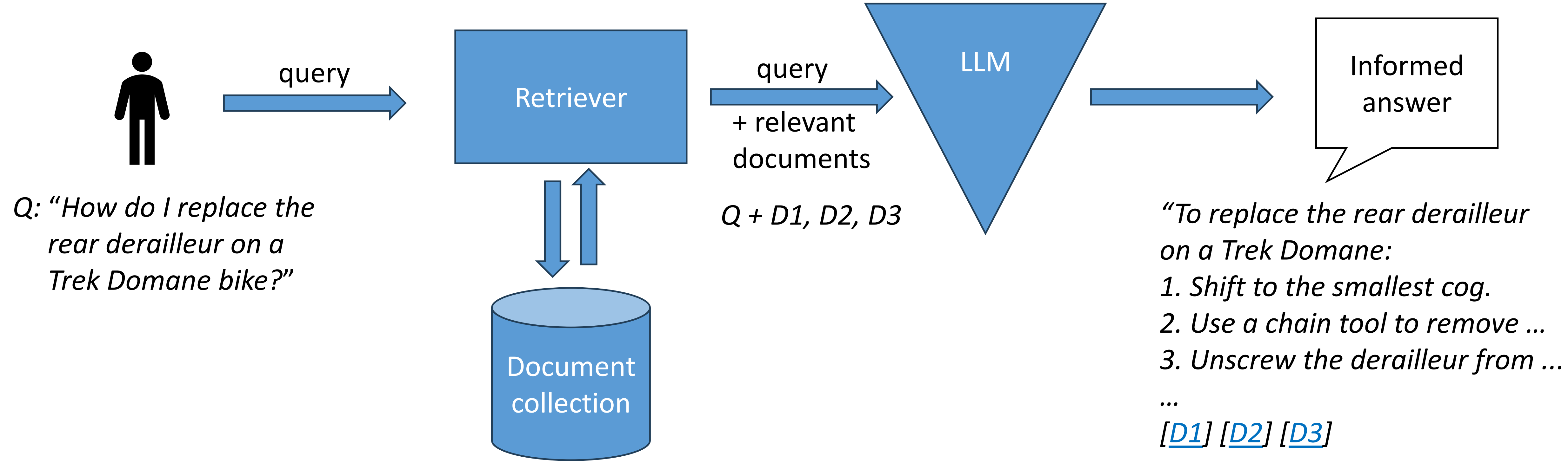
Ouyang, Long, et al. "Training language models to follow instructions with human feedback." *Advances in neural information processing systems* 35 (2022): 27730-27744.

Outline

- Using LLMs for Retrieval
  - Fine-tuning LLMs
  - Retrieval Augmented Generation (RAG)
    - End-to-end RAG systems
    - RAGs Based on fixed LLMs
    - Recent evolutions
- Evaluation: LLM as a judge

Retrieval-Augmented Generation

- RAG system:
  - Extend LLM with a ***knowledge base***



D1 "To replace the rear derailleur on a Trek Domane, first shift …
D2 "Trek Domane bikes use a direct-mount derailleur hanger …
D3 "Shimano 105 rear derailleurs: installation instructions …

Universiteit
Antwerpen

Retrieval-Augmented Generation

- Components:
  - Knowledge base
  - Retrieval module
  - Generator
  - Optional: reranker, relevance assessment, memory, feedback loop …

- Advantages:
  - Reduced hallucination
  - Traceability; citing information sources
  - Easy to update information

Outline

- Using LLMs for Retrieval
  - Fine-tuning LLMs
  - Retrieval Augmented Generation (RAG)
    - End-to-end RAG systems
    - RAGs Based on fixed LLMs
    - Recent evolutions
- Evaluation: LLM as a judge

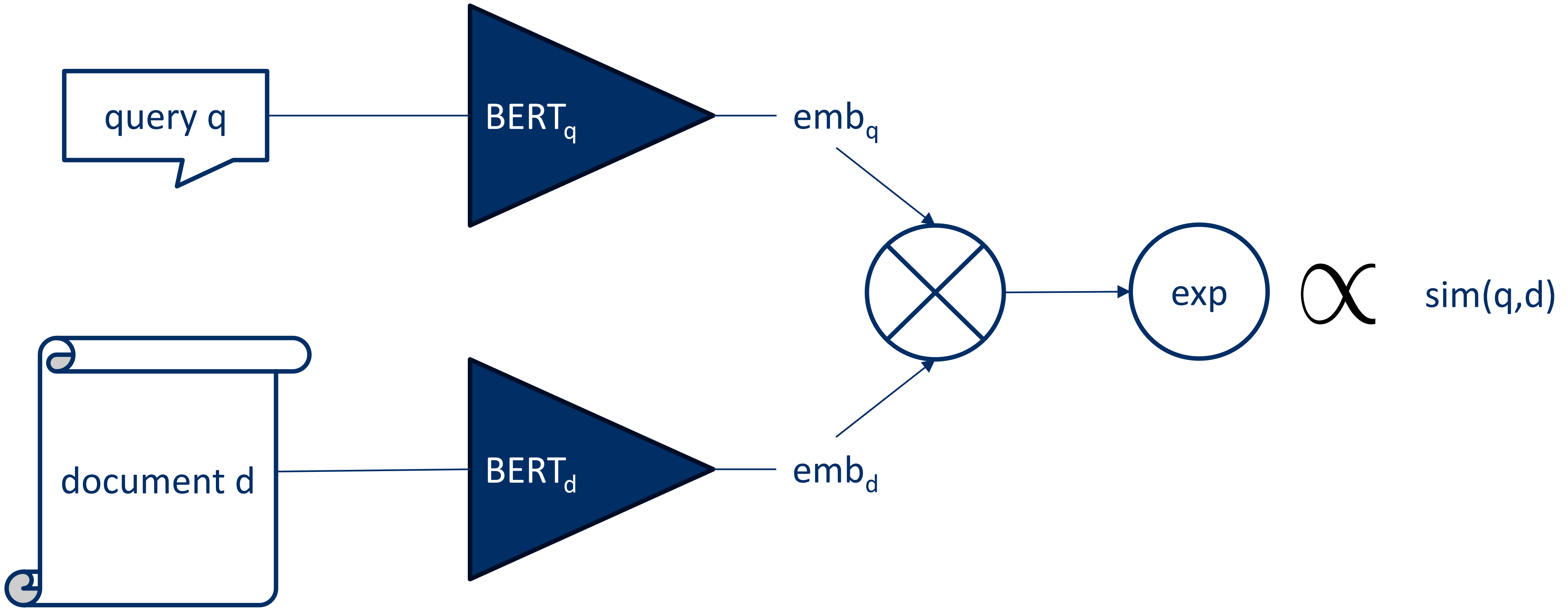
End-to-End RAG System

- End-to-end finetuned system
  - Knowledge base: collection of documents
  - Retrieval module: Dense passage retrieval (DPR)
  - LLM answer Generator:
    - BART (encoder-decoder seq2seq model based on transformer architecture)
    - RAG-sequence vs RAG-token modes to combine documents

- Main challenge: How to optimize?
  - Selecting documents by retriever
    - Distribution over documents : Computationally infeasible
    - Top-k : Non-differentiable

Lewis, Patrick, et al. "Retrieval-augmented generation for knowledge-intensive nlp tasks." Advances in neural information processing systems 33 (2020): 9459-9474.

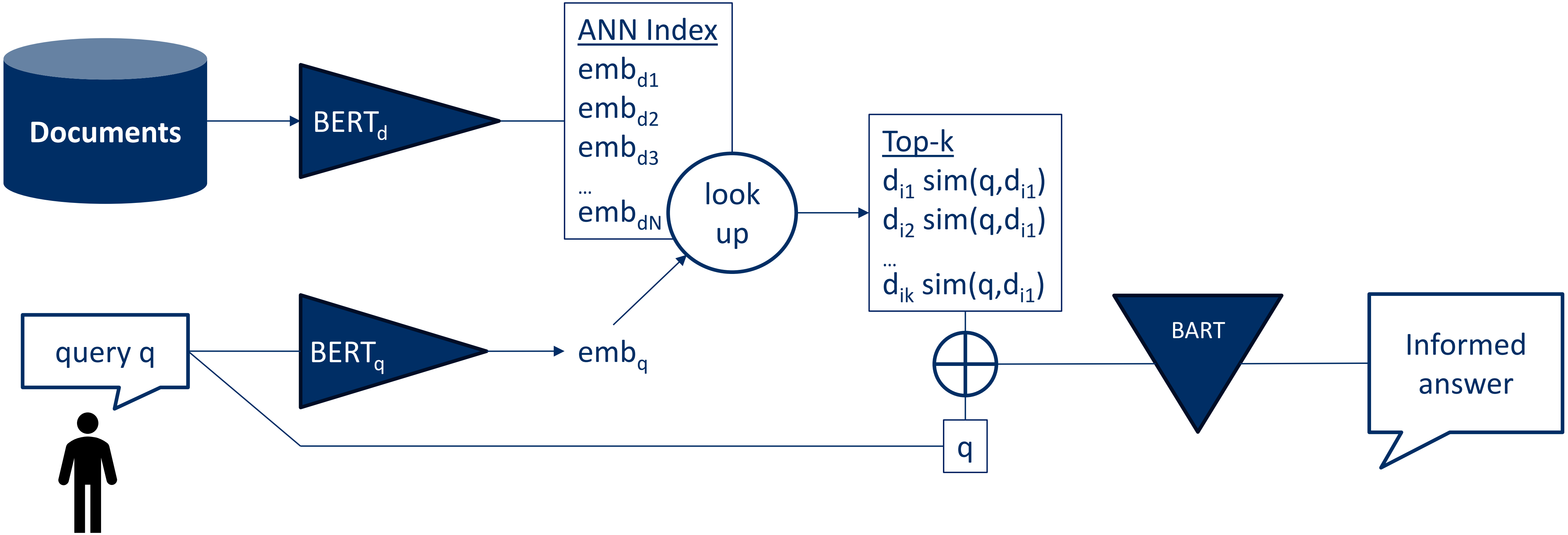
Dense Passage Retrieval - Conceptual



- **(Pre-)Trained with labeled query-document pairs**
- **After training build index for emb$_d$. Use index during inference**

Karpukhin, Vladimir, et al. "Dense passage retrieval for open-domain question answering." *EMNLP (1)*. 2020.

Architecture of an End-to-End RAG System



Lewis, Patrick, et al. "Retrieval-augmented generation for knowledge-intensive nlp tasks." Advances in neural information processing systems 33 (2020): 9459-9474.

Generator: RAG-Sequence vs RAG-Token

- **Query q,**
- **Top-k documents $d_1, \ldots, d_k$**

$p(d|q) \propto \exp(BERT_q(q) \times BERT_d(d))$

$$p_{\text{RAG-Sequence}}(y \mid q) \approx \sum_{i=1}^{k} p(d_i \mid q) p_\theta(y|d_i, q)$$

$$= \sum_{i=1}^{k} p(d_i \mid q) \prod_{i=1}^{N} p_\theta(y_i|d_i, q, y_{1:i-1})$$

p_Θ = probability assigned to sequence y by BART model with parameters Θ

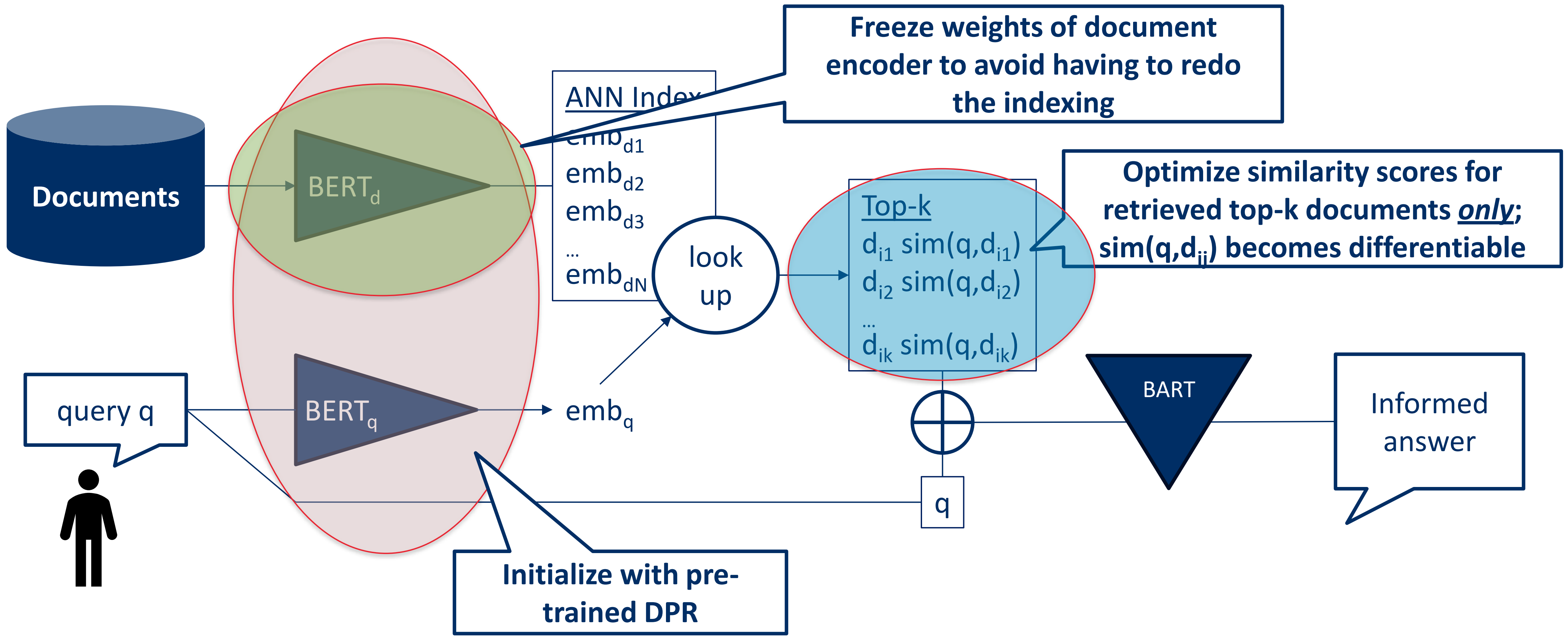
Auto-regressive model generates output token-by-token

For each document d, most likely output for (d+q) is computed. Overall most likely sequence is outputted

$$p_{\text{RAG-Token}}(y \mid q) = \prod_{i=1}^{N} \sum_{i=1}^{k} p(d_i \mid q) p_\theta(y_i|d_i, q, y_{1:i-1})$$

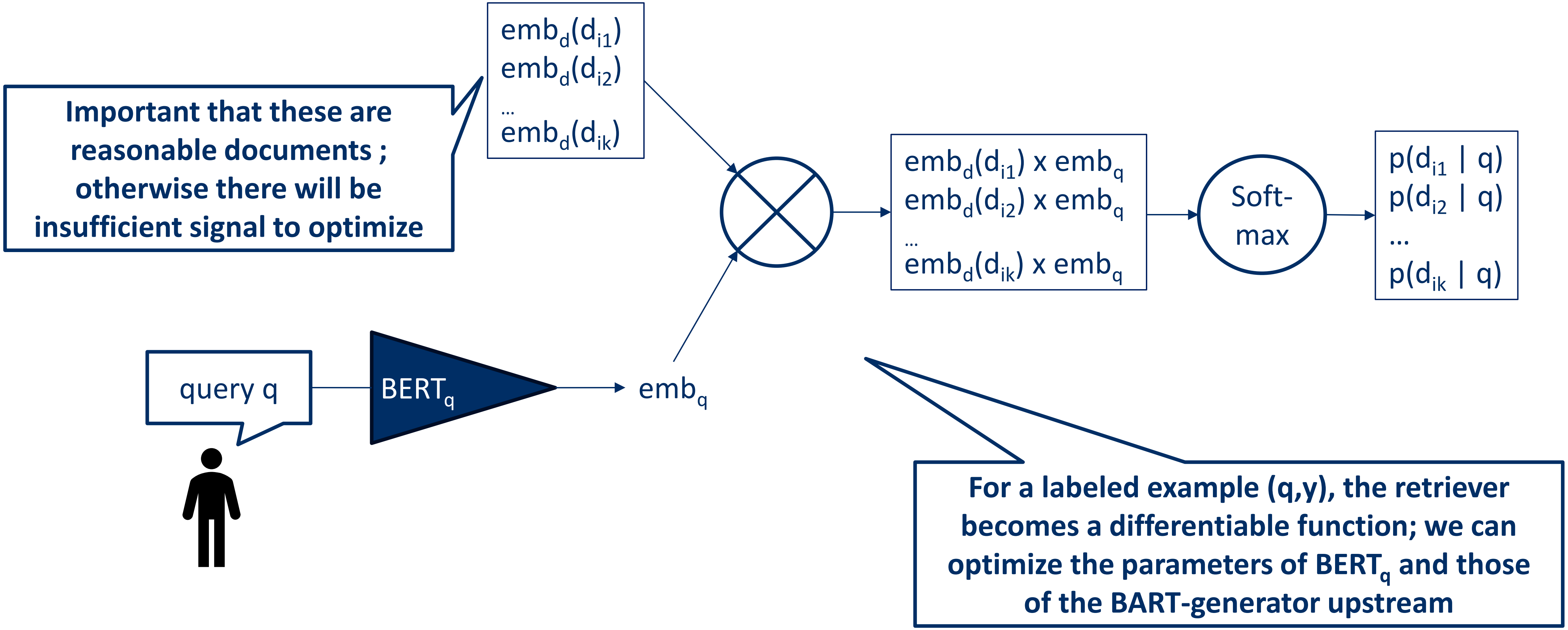
Given a partial answer, the next token is generated taking all documents into account; weighted average is taken.

Training of an End-to-End RAG System



Lewis, Patrick, et al. "Retrieval-augmented generation for knowledge-intensive nlp tasks." Advances in neural information processing systems 33 (2020): 9459-9474.

Training of an End-to-End RAG System - Retriever



Lewis, Patrick, et al. "Retrieval-augmented generation for knowledge-intensive nlp tasks." Advances in neural information processing systems 33 (2020): 9459-9474.

Outline

- Using LLMs for Retrieval
  - Fine-tuning LLMs
  - Retrieval Augmented Generation (RAG)
    - End-to-end RAG systems
    - RAGs Based on fixed LLMs
    - Recent evolutions
- Evaluation: LLM as a judge

RAG Based on Fixed LLMs

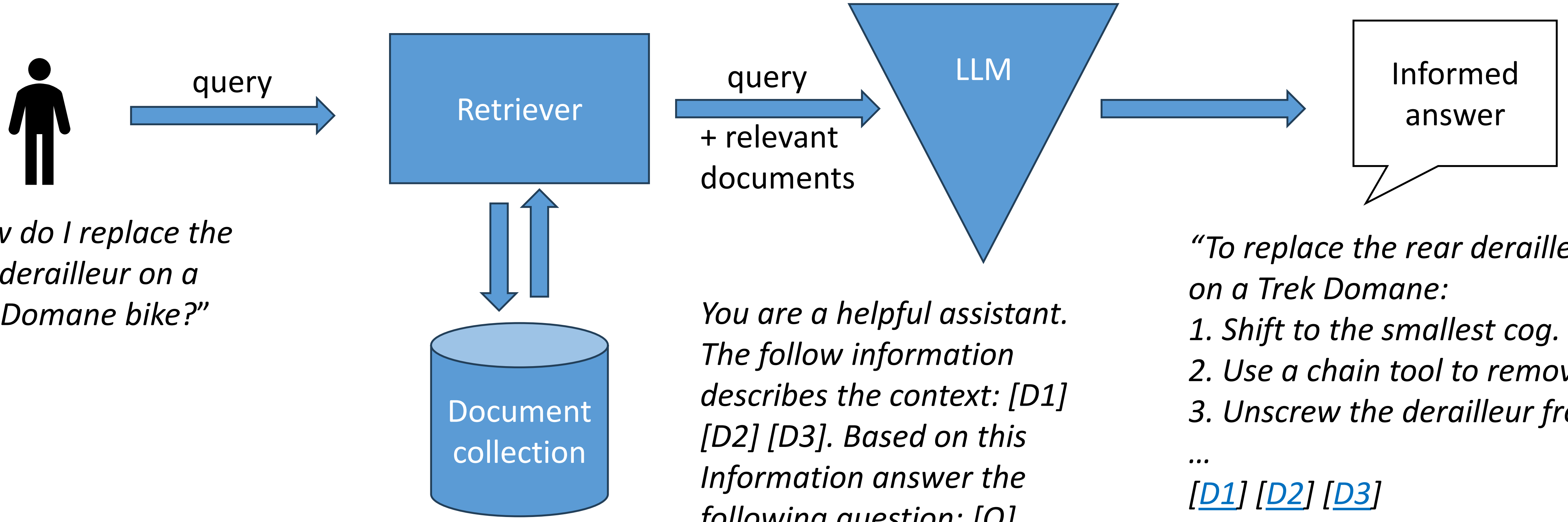
Instruction prompting + some/no examples given of desired output

- LLMs became more powerful
  - Larger *contexts*
  - Move from fine-tuning to few-shot/zero-shot learning

- As a result: in modern RAG
  - Use frozen LLMs (GPT-3.5, GPT-4, Llama, etc.)
  - No longer end-to-end training
  - Retrieval and generation are modular components

Brown, Tom, et al. "Language models are few-shot learners." *Advances in neural information processing systems* 33 (2020): 1877-1901.

RAG Based on Fixed LLMs: Pipeline



Q: “How do I replace the rear derailleur on a Trek Domane bike?”

You are a helpful assistant. The follow information describes the context: [D1] [D2] [D3]. Based on this Information answer the following question: [Q].

“To replace the rear derailleur on a Trek Domane:
1. Shift to the smallest cog.
2. Use a chain tool to remove …
3. Unscrew the derailleur from …
…
[D1] [D2] [D3]

D1 “To replace the rear derailleur on a Trek …
D2 “Trek Domane bikes use a …
D3 “Shimano 105 rear derailleurs: installation instructions …

Universiteit Antwerpen

RAG Components

- Chunking:
  - Large documents are split into *chunks*
  - Indexer will index chunks, retriever retrieves chunks
- Indexing:
  - ANN index; e.g. FAISS
- Retrieval:
  - Cosine similarity, BM25
  - Dense Passage Retrieval
- Generator
  - Instruction-based LLM
  - Zero-shot prompt to generate answer

Chunking

- Split large documents into smaller chunks
  - Chunks are ideally semantically coherent passages
- Dense retrieval works better with passages
  - Long documents produce diluted embeddings
  - Retrieve relevant parts of a document

- Practical guidelines
  - Chunk size: 200–300 tokens (typical)
  - Overlap: 10–20% to preserve context across boundaries
  - Keep semantic coherence: don’t split mid-sentence or mid-section
  - Can use LLMs for chunking

RAG Terminology

- Naïve RAG: RAG consisting of index-retrieve-generate
- Advanced RAG: extensions to Naïve RAG patterns
- Modular RAG: Refers to a *Design pattern*;
  RAG systems consisting of replaceable/separately trainable components
  - Retrievers, chunkers, embedders, vector stores, LLM wrappers, output parsers, rerankers, evaluation modules
  - The components can be combined in pipelines

- Example:
  - LangChain, LlamaIndex

Advanced RAG Concepts

- Hierarchical index
  - Use LLMs to summarize documents
  - First search in summaries
  - Then refine search within retrieved documents
- Hypothetical Question Index
  - For each document, use LLM to generate potential questions for the document
  - Index the questions
  - Compare queries to the indexed questions
  - Similar to Hypothetical Document Embedding

Gao, Luyu, et al. "Precise zero-shot dense retrieval without relevance labels." *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2023.

https://assets.ctfassets.net/3viuren4us1n/3CDjWG3R6KCvkYjQRtrVEW/d04b880fcad6a9b144f2a226c28915e2/15_Advanced_RAG_Techniques.pdf

Advanced RAG Concepts

- Use LLM
  - To rewrite queries
  - To decide if retrieval is needed
  - To evaluate retrieved documents (e.g. Corrective RAG)
  - To evaluate answer (e.g. Self-RAG)
    - Is the answer supported by the documents?
    - Is the answer complete?
    - Do we need additional retrieval?

Asai, Akari, et al. "Self-rag: Self-reflective retrieval augmented generation." *NeurIPS 2023 workshop on instruction tuning and instruction following*. 2023.

Yan, Shi-Qi, et al. "Corrective retrieval augmented generation." (2024).

Outline

- Using LLMs for Retrieval
  - Fine-tuning LLMs
  - Retrieval Augmented Generation (RAG)
    - End-to-end RAG systems
    - RAGs Based on fixed LLMs
    - Recent evolutions
      - GraphRAG
      - Agentic RAG
- Evaluation: LLM as a judge

GraphRAG

- Naive RAG works very well for local queries
  - *What is the capital of Spain?* → Can be found in a single chunk
- Doesn't work for "sensemaking tasks"
  - *"What are the key trends in how scientific discoveries are influenced by interdisciplinary research over the past decade?"*

- Organize document collection as a *knowledge graph*
  - Entities and relations
  - Communities
  - Community-level summaries

Edge, Darren, et al. "From local to global: A graph rag approach to query-focused summarization." *arXiv preprint arXiv:2404.16130* (2024).

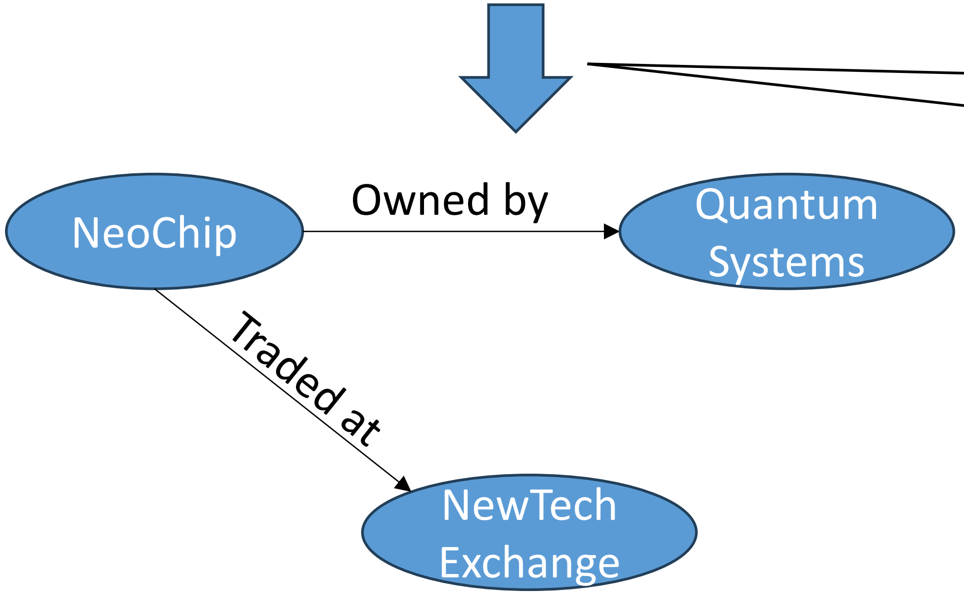
GraphRAG – KG Generation

- Use LLM to create *Knowledge Graph*

NeoChip's (NC) shares surged in their first week of trading on the NewTech Exchange. However, market analysts caution that the chipmaker's public debut may not reflect trends for other technology IPOs. NeoChip, previously a private entity, was acquired by Quantum Systems in 2016. The innovative semiconductor firm specializes in low-power processors for wearables and IoT devices.

Using LLM; zero-shot or few-shot learning

Also extract node summaries and claims

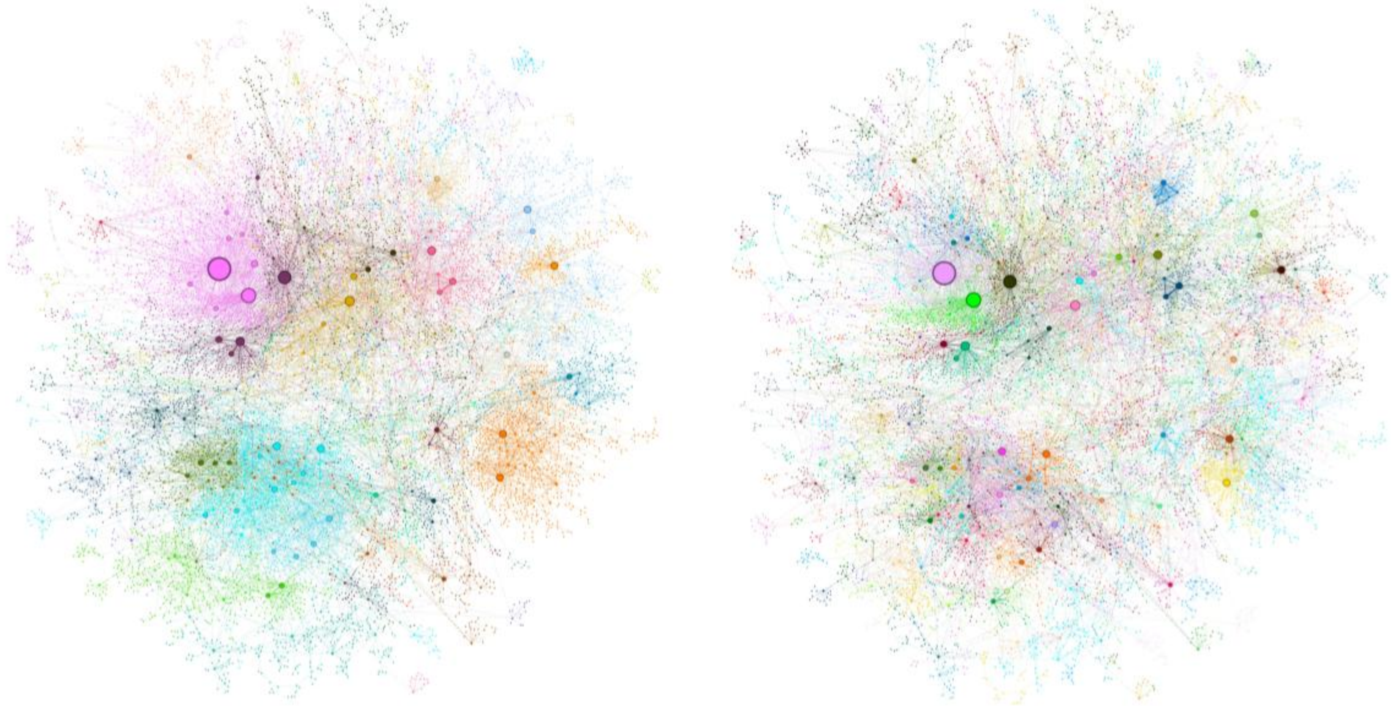


Universiteit Antwerpen

Edge, Darren, et al. 2024

GraphRAG – Detecting Communities

- Use Graph Clustering to detect groups of related concepts
- Generate *community-level summaries*



(a) Root communities at level 0 (b) Sub-communities at level 1

Universiteit Antwerpen

Edge, Darren, et al. 2024

GraphRAG – Answering Queries

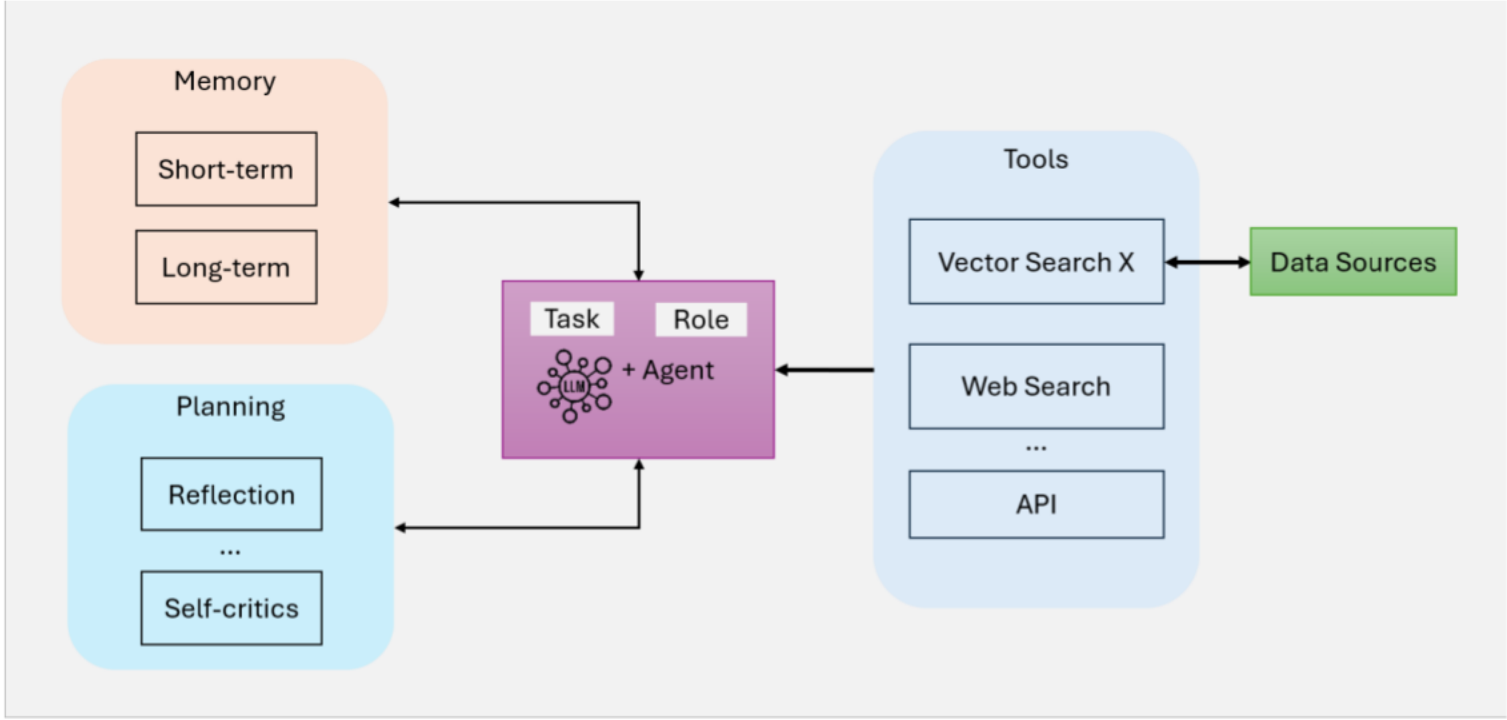
- Answer “sensemaking queries” using community summaries
  - Collect relevant community summaries
  - Apply LLM to generate answers
  - Score answers
  - Aggregate promising answers into one answer

Outline

- Using LLMs for Retrieval
  - Fine-tuning LLMs
  - Retrieval Augmented Generation (RAG)
    - End-to-end RAG systems
    - RAGs Based on fixed LLMs
    - Recent evolutions
      - GraphRAG
      - Agentic RAG
- Evaluation: LLM as a judge

Agentic AI

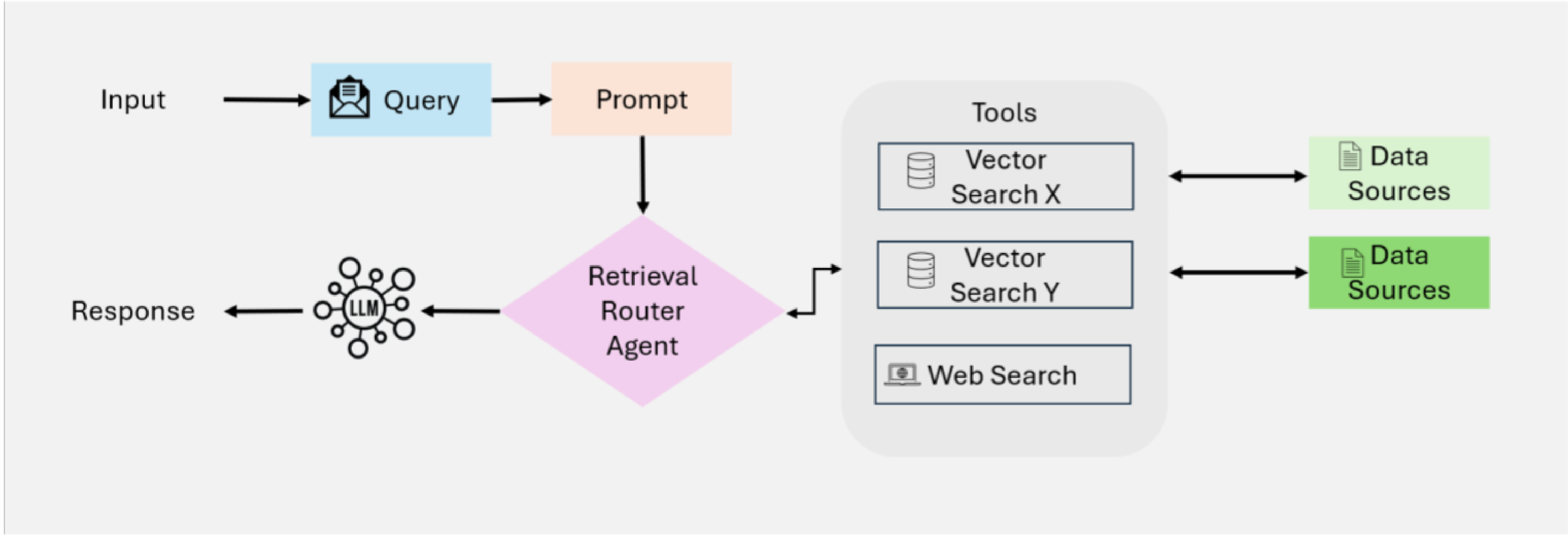
- *Umbrella term referring to adaptive, multi-step RAG workflows in which an LLM dynamically decides actions such as querying, re-retrieving, verifying, or refining results.*



Singh, Aditi, et al. "Agentic retrieval-augmented generation: A survey on agentic rag." *arXiv preprint arXiv:2501.09136* (2025).

Agentic RAG

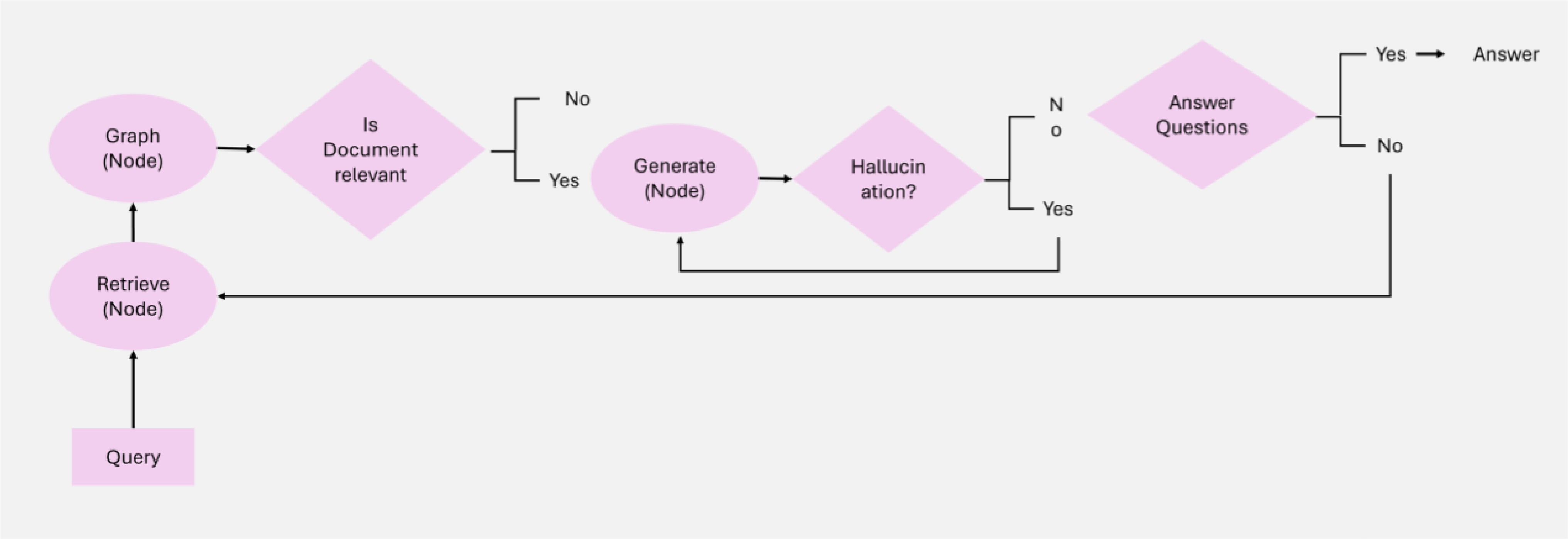
- Agent plans which resoures to inspect / tools to use



Singh, Aditi, et al. "Agentic retrieval-augmented generation: A survey on agentic rag." *arXiv preprint arXiv:2501.09136* (2025).

Agentic RAG

- Agent evaluates answer quality and retrieves/generates again if needed



Universiteit Antwerpen

Singh, Aditi, et al. "Agentic retrieval-augmented generation: A survey on agentic rag." *arXiv preprint arXiv:2501.09136* (2025).

Outline

- Using LLMs for Retrieval
  - Fine-tuning LLMs
  - Retrieval Augmented Generation (RAG)
    - End-to-end RAG systems
    - RAGs Based on fixed LLMs
    - Recent evolutions
      - GraphRAG
      - Agentic RAG
- Evaluation: LLM as a judge

Evaluation: LLM as a Judge

- Evaluation frameworks for RAG systems often use LLMs as a judge
  - E.g. RAGAS (Retrieval Augmented Generation Assessment)
  - Generate questions + answers from documents
  - Evaluate answers given by the RAG system
    - Faithfulness
    - Answer and Context Relevance
- Has good correlation with human annotators:

	Faith.	Ans. Rel.	Cont. Rel.
RAGAS	**0.95**	**0.78**	**0.70**
GPT Score	0.72	0.52	0.63
GPT Ranking	0.54	0.40	0.52

Table 4: Agreement with human annotators in pairwise comparisons of faithfulness, answer relevance and context relevance, using the WikEval dataset (accuracy).

Es, Shahul, et al. "Ragas: Automated evaluation of retrieval augmented generation." *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*. 2024.

Summary

- Retrieval-Augmented Generation as a new paradigm combining retrieval and generative models
  - Chunk and index documents
  - Retrieve relevant chunks at query time
  - Generate answer based on chunks by LLM
- Fewer hallucinations, more transparent (cite sources)
- Initially training end-to-end, later LLM as a frozen component
- Many different flavors at different levels of complexity:
  - Naïve RAG, Advanced RAG, Agentic RAG, GraphRAG, …