RAG Engineering

👤 Keheliya Gallaba

🌐 keheliya.github.io

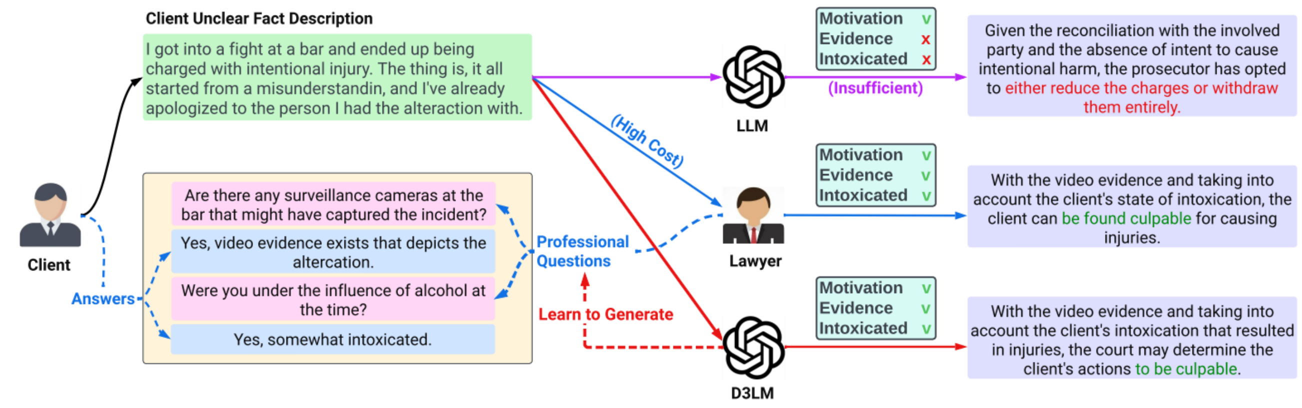
How to cite this session?

```
@misc{Gallaba2024RAGTutorial,
author = {Keheliya Gallaba and Dayi Lin and Ahmed E. Hassan},
title = {RAG Engineering},
howpublished = {Tutorial presented at the AIware Leadership Bootcamp 2024},
month = {November},
year = {2024},
address = {Toronto, Canada},
note = {Part of the AIware Leadership Bootcamp series.},
url = {https://aiwarebootcamp.io/slides/2024_aiwarebootcamp_gallaba_keheliya_ragengineering.pdf }}
```

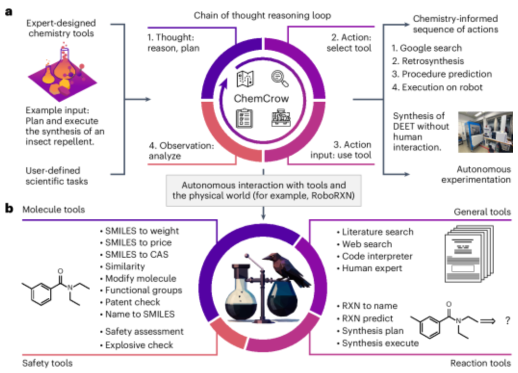
Overview of the session

- **Why RAG?:** What are the challenges in AIware development that RAG tries to address?
- **RAG Overview:** Overview of RAG and main concepts
  - Sparse Retrieval vs Dense Retrieval
  - Augmentation
  - Pre-processing and post-processing
- **Advanced Retrieval Concepts:** Combining multiple paradigms to achieve better retrieval
  - SPLADE
  - ColBERT
  - ColPali
- **Advanced RAG Patterns:** Integrating RAG with other advancements in foundation model domain
  - Contextual Retrieval
  - Self-RAG
  - Least-to-most prompting
  - IR-COT
- **Applications of RAG in SE:** Software Engineering as a case study how retrieval augmented generation has been used to improve SOTA
- **Limitations of RAG**
- **Productionizing challenges of RAG**

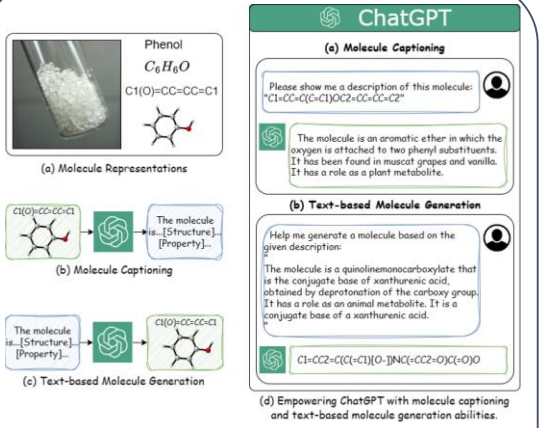
Foundation models have shown potential in many domains



Legal services

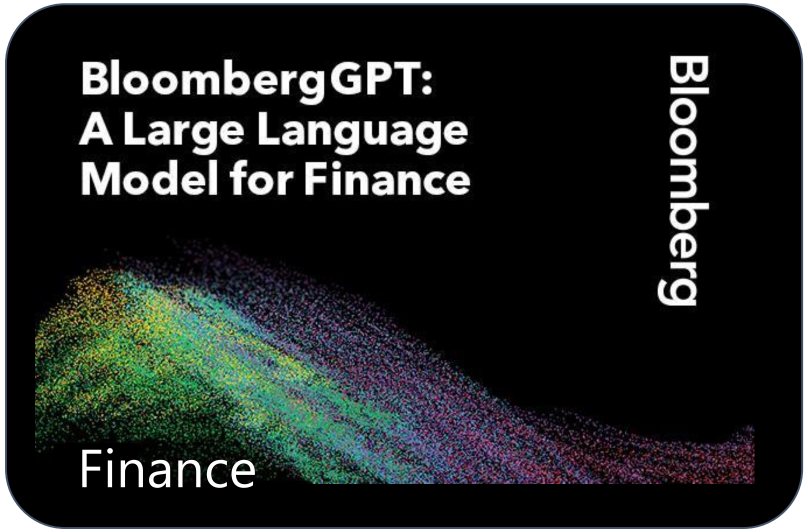


Chemical Synthesis

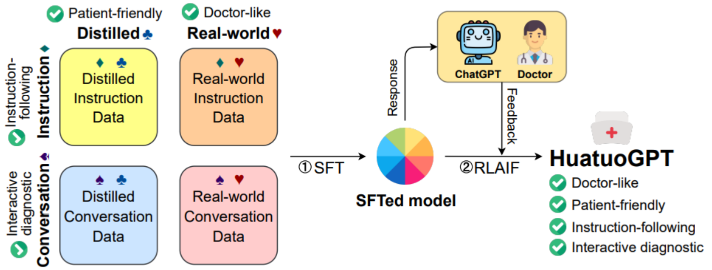


Molecules Discovery

HuatuoGPT, towards Taming Language Model to Be a Doctor https://aclanthology.org/2023.findings-emnlp.725/
BloombergGPT: A Large Language Model for Finance https://arxiv.org/pdf/2303.17564
ChemCrow: Augmenting large-language models with chemistry tools https://www.nature.com/articles/s42256-024-00832-8/
Empowering Molecule Discovery for Molecule-Caption Translation with Large Language Models: A ChatGPT Perspective https://arxiv.org/abs/2306.06615
https://github.com/Jeryi-Sun/LLM-and-Law



Finance



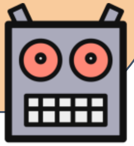
Medical Consultation

However there are some limitations hindering their usefulness

Outdated knowledge

Which country won the most gold medals in 2024 Olympics?

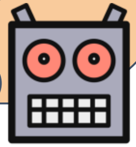
My knowledge cut-off was in December 2023, I can't answer which country won in 2024.



Lack of Domain-Specific Knowledge

What is the target blood pressure for a 50yr old man?

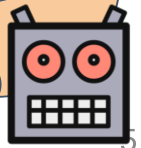
Hmmm..



Lack of Internal Data

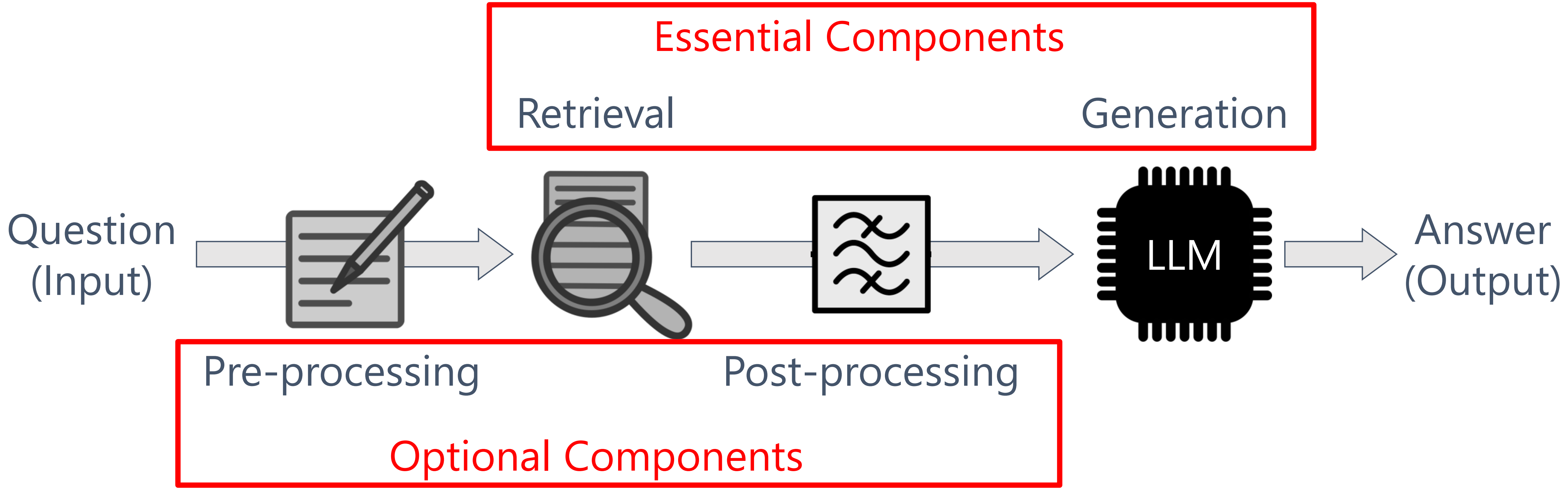
Who handles loan approvals in ABC department of XYZ corp?

I don't have private data from XYZ corp in my training set



Hallucinations! The tendency to provide "plausible-sounding" answers is too strong

Retrieval-Augmented Generation (RAG) to the rescue



Key idea: Augment FM's knowledge with appropriate facts from a knowledge base to enable grounded generation.

Grounding in Language Models

Key idea: Every claim in the response generated by an LLM can be attributed to a document in the user-specified knowledge base.

Retrieval Augmented Generation (RAG) is one technical solution. There are other approaches such as constrained decoding, guardrails, corroborate and revise (CAR), and corpus tuning.

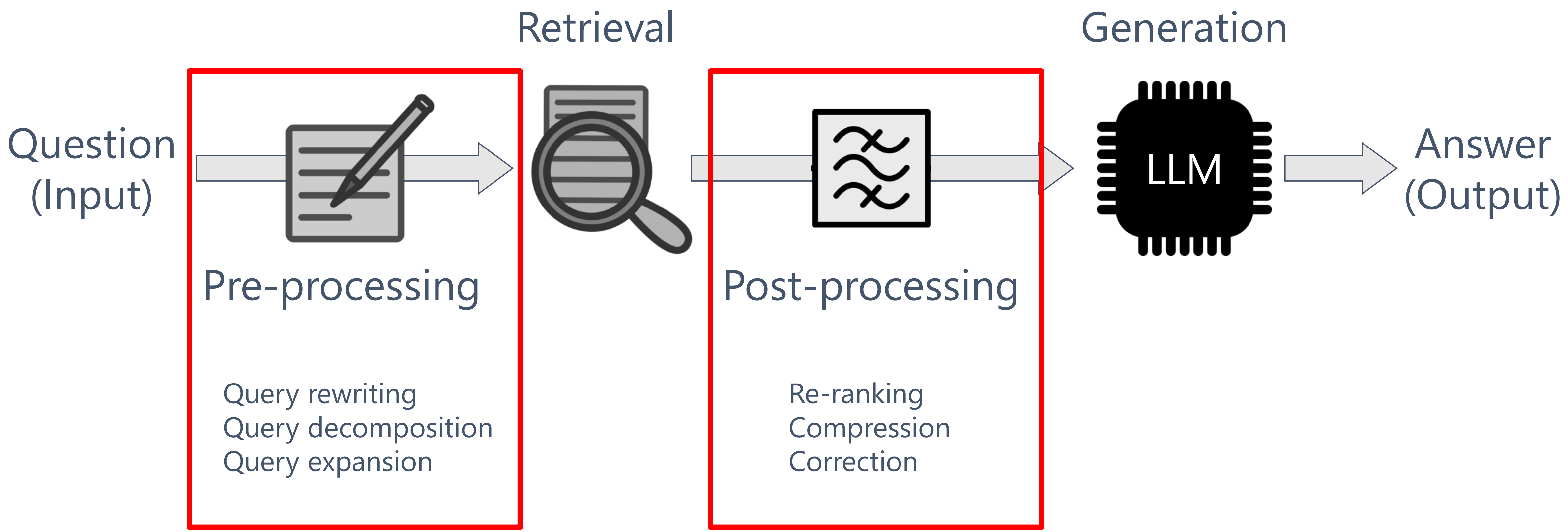
Grounding Vs Factuality

Grounding seeks attribution to a user-specific knowledge base.

Factuality seeks attribution to commonly agreed world knowledge.

Kenthapadi et al., "Grounding and Evaluation for Large Language Models: Practical Challenges and Lessons Learned (Survey)." KDD 2024.

Improving RAG performance with pre-processing and post-processing



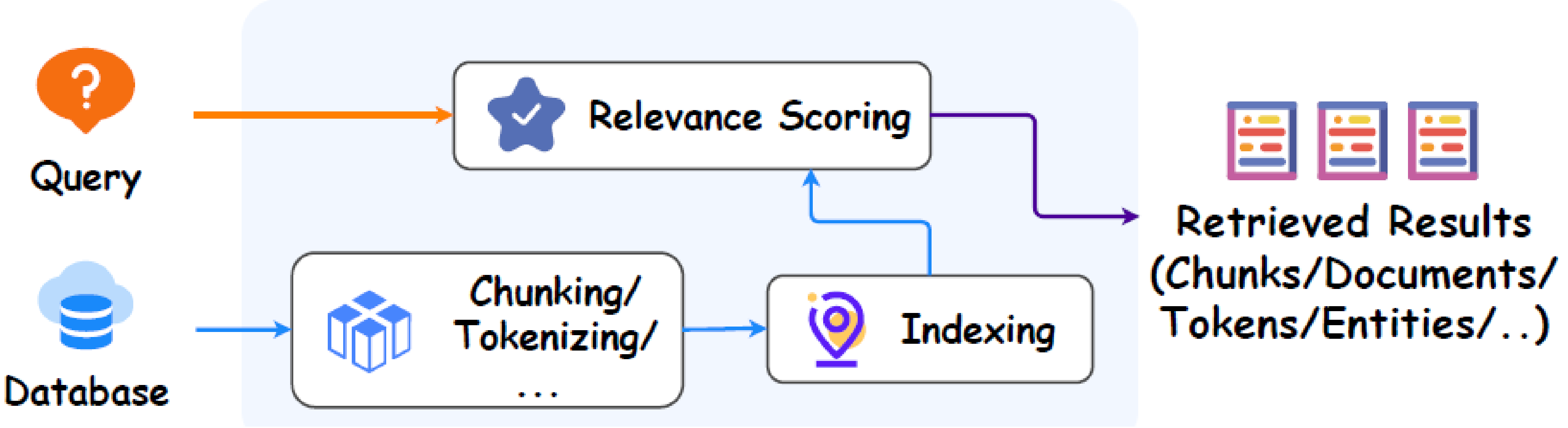
Overview of the session

- **Why RAG?:** What are the challenges in AIware development that RAG tries to address?
- **RAG Overview:** Overview of RAG and main concepts
  - Sparse Retrieval vs Dense Retrieval
  - Augmentation
  - Pre-processing and post-processing
- **Advanced Retrieval Concepts:** Combining multiple paradigms to achieve better retrieval
  - SPLADE
  - ColBERT
  - ColPali
- **Advanced RAG Patterns:** Integrating RAG with other advancements in foundation model domain
  - Contextual Retrieval
  - Self-RAG
  - Least-to-most prompting
  - IR-COT
- **Applications of RAG in SE:** Software Engineering as a case study how retrieval augmented generation has been used to improve SOTA
- **Limitations of RAG**
- **Productionizing challenges of RAG**

Dense vs Sparse Retrieval

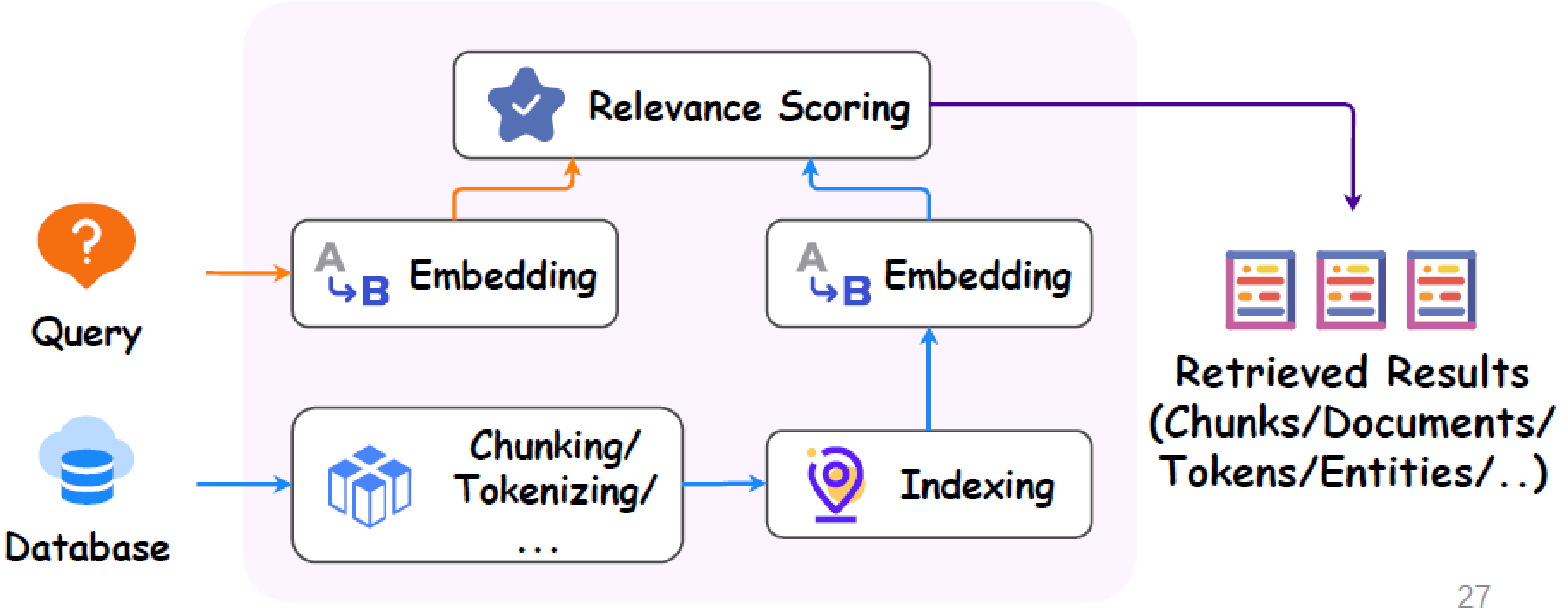
Sparse Retrievers

E.g., TF-IDF, BM25



Dense Retrievers

E.g., DPR, Contriever



Fan et al., A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models

Sparse Retrieval

Lexical approaches

TF-IDF (Term Frequency-Inverse Document Frequency) measures how important a word is to a document in a collection.

$$TF = \frac{\text{number of times the term appears in the document}}{\text{total number of terms in the document}}$$

$$IDF = log(\frac{\text{number of the documents in the corpus}}{\text{number of documents in the corpus contain the term}})$$

$$TF\text{-}IDF = TF * IDF$$

BM25 (Best Matching 25) aka Okapi BM25 builds upon TF-IDF by considering **document length** and applying a **saturation function** to term frequency. This is to prevent common words from dominating the results.

$$\sum_{i}^{n} IDF(q_i) \frac{f(q_i, D) \, * \, (k1 + 1)}{f(q_i, D) \, + \, k1 \, * \, (1 \, - \, b \, + \, b \, * \, \frac{fieldLen}{avgFieldLen})}$$

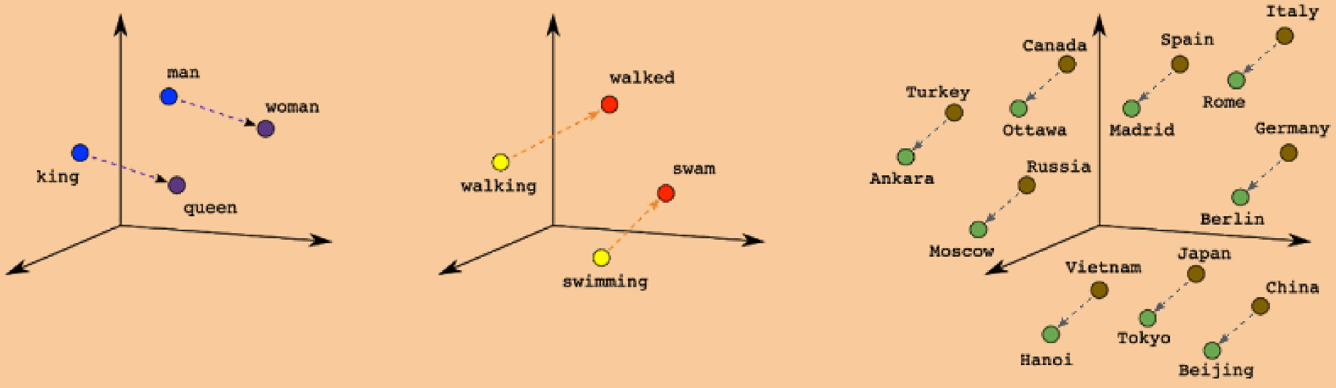
- Don’t need fine-tuning
- Effective for queries that include **unique identifiers** or **technical terms**.
  - E.g., "Error code TS-999"
- Relies on **exact keyword** matches, lacks semantic understanding.
- Struggles with complex queries and large datasets.

Side Quest

What are embeddings?

A way of representing data as points in space (n-dimensional coordinates a.k.a. **vectors**) where the locations of the points are **semantically meaningful**.

“Dog is man’s best friend.” ⇨ [0.03135875, 0.03640932, -0.00031054, 0.04873694, -0.03376802]



Male-Female Verb Tense Country-Capital

Helps to compute numerical **similarity scores** between data points.

Embeddings are **deterministic**. For a given set of input, embedding models will always generate the same output (unlike inference models which are non-deterministic).
Think of it as a hash or a form of compression.

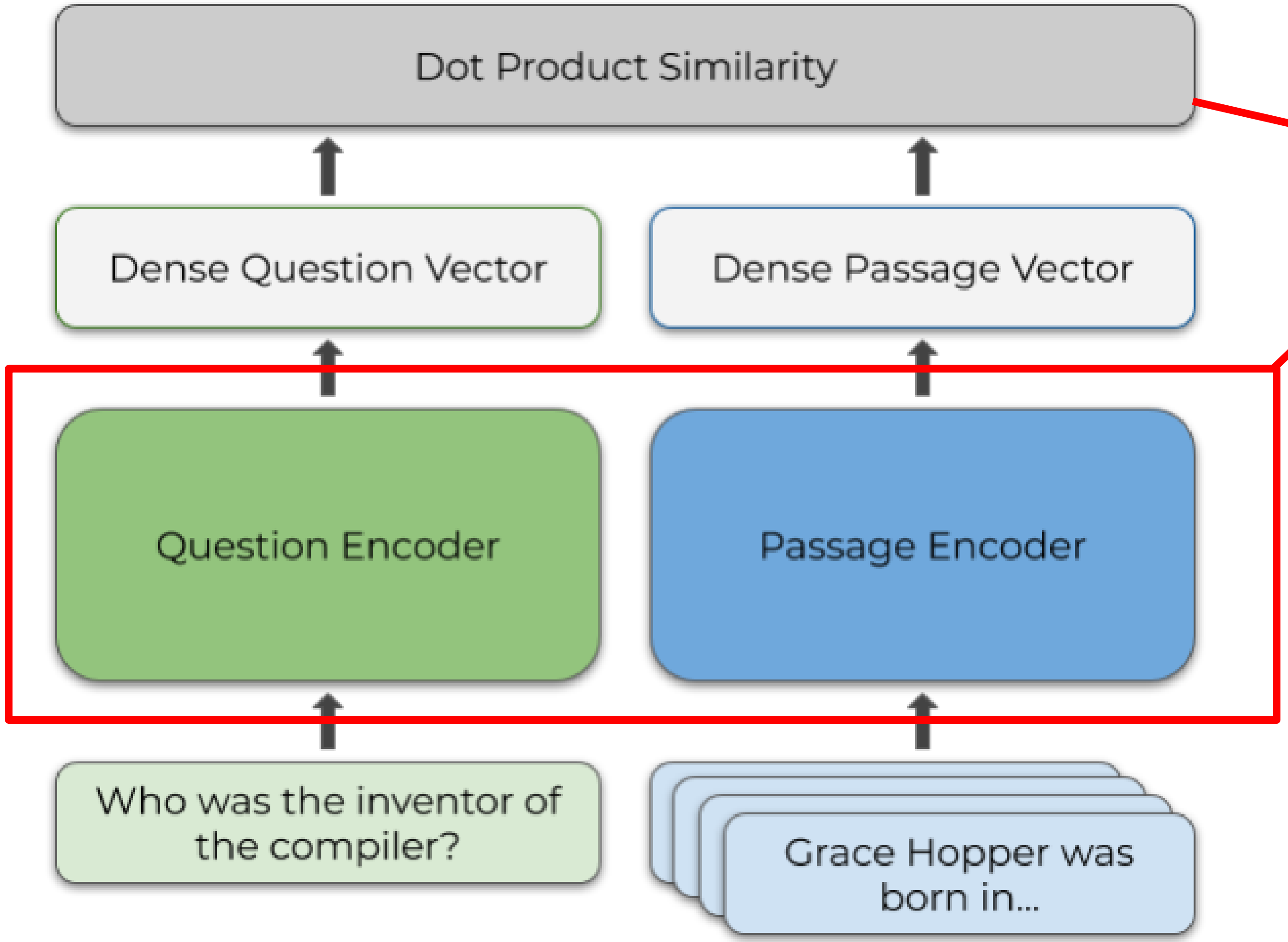
Models of note:

- Google’s **Word2vec** work from 2013 for word embedding
- OpenAI's **CLIP** model can map both **image** or **text** to the same embedding space.
- BERT, Instructor-xl, Ada-002, Sentence-transformers

https://cloud.google.com/blog/topics/developers-practitioners/meet-ais-multitool-vector-embeddings
Massive Text Embedding Leaderboard https://huggingface.co/spaces/mteb/leaderboard

Dense Retrieval

Dense Passage Retriever (DPR)



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

Inner Product Similarity

two distinct BERT encoders

- Pretrained for Open Domain Question Answering (ODQA) where learning objective is

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

- **Training data:** Question-Passage Sets with in-batch negatives

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

Question

Irrelevant passages

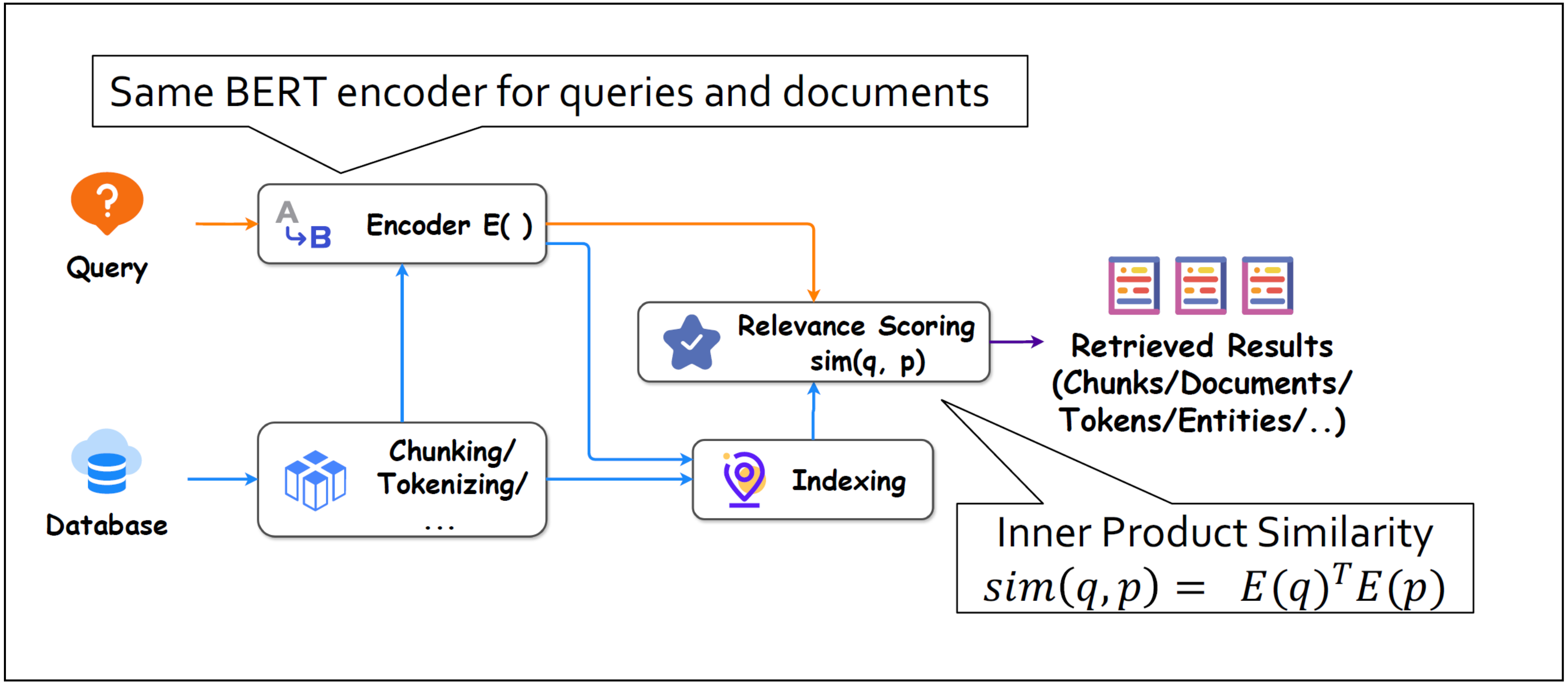
Relevant passage

Dense Retrieval

Contriever

	Additional data	SciFact	NFCorpus	FiQA
# queries		729	2,590	5,500
BM25	-	66.5	32.5	23.6
BERT	-	75.2	29.9	26.1
Contriever	-	84.0	33.6	36.4
BERT	MS MARCO	80.9	33.2	30.9
Contriever	MS MARCO	**84.8**	**35.8**	**38.1**

Key Idea: Pretraining without supervision using contrastive learning

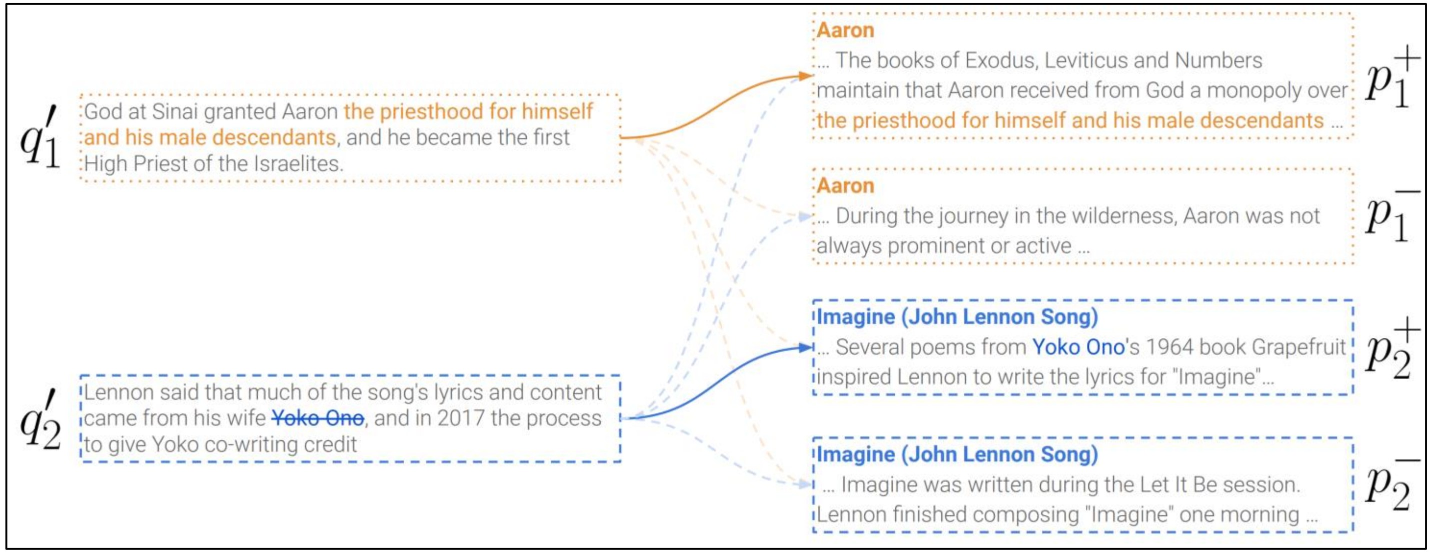


Izacard et al. 2022. "Unsupervised Dense Information Retrieval with Contrastive Learning"

SPIDER – Span-based Unsupervised Dense Retriever

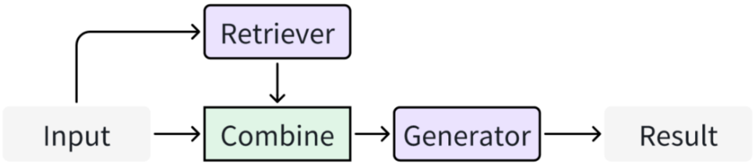
Key Ideas:

- **Recurring Span:** Given two passages with the same recurring span, a query is created from one of the paragraphs, while the other is taken as the target for retrieval.
- **Document passage** decomposition to leverage the inherent structure to get data tuples for contrastive learning
- **Task Specific:** open-domain question answering (ODQA)



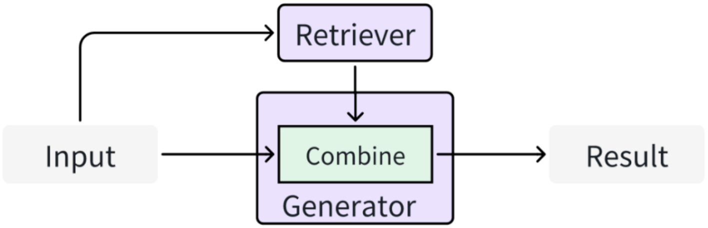
Ram et al., 2022, Learning to Retrieve Passages without Supervision

Retrieved Results Integration



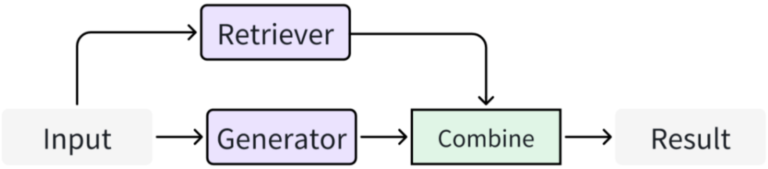
Input-layer integration
aka Query-based RAG

E.g., REALM, RetDream, DocPrompting



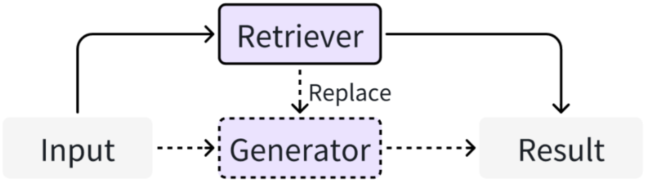
Intermediate-layer integration
aka Latent representation-based RAG

E.g., RETRO, EaE, ROME



Output-layer integration
aka Logit-based RAG

E.g., kNN-LM, EDITSUM, TRIME

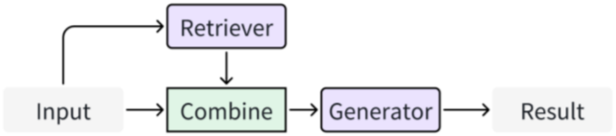


Speculative RAG

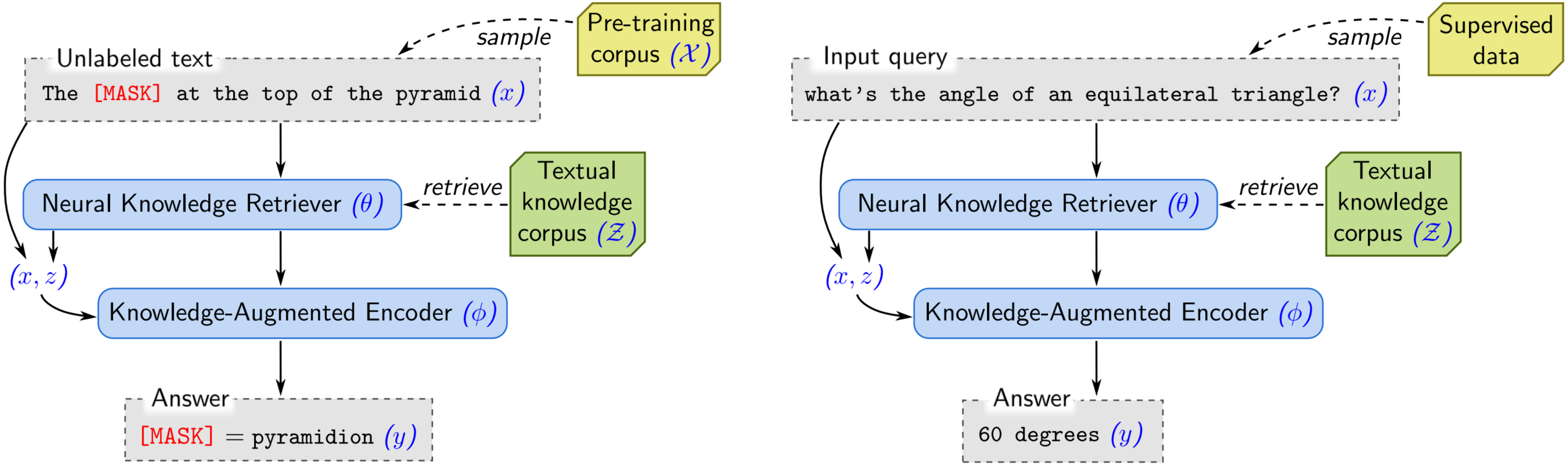
E.g., CoG, GPTCache, REST

Zhao et al., Retrieval-Augmented Generation for AI-Generated Content: A Survey

Input-layer Integration - REALM



Key Idea: augment language model pretraining with a neural knowledge retriever that retrieves knowledge from a textual knowledge corpus.



Unsupervised pre-training. The knowledge retriever and knowledge-augmented encoder are jointly pre-trained on the unsupervised language modeling task.

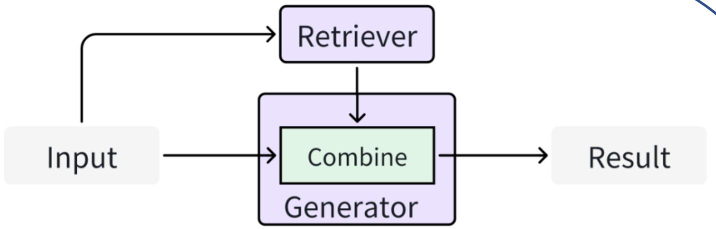
Supervised fine-tuning. After the parameters of the retriever (θ) and encoder (ϕ) have been pre-trained, fine-tuned on specific task, using supervised examples.

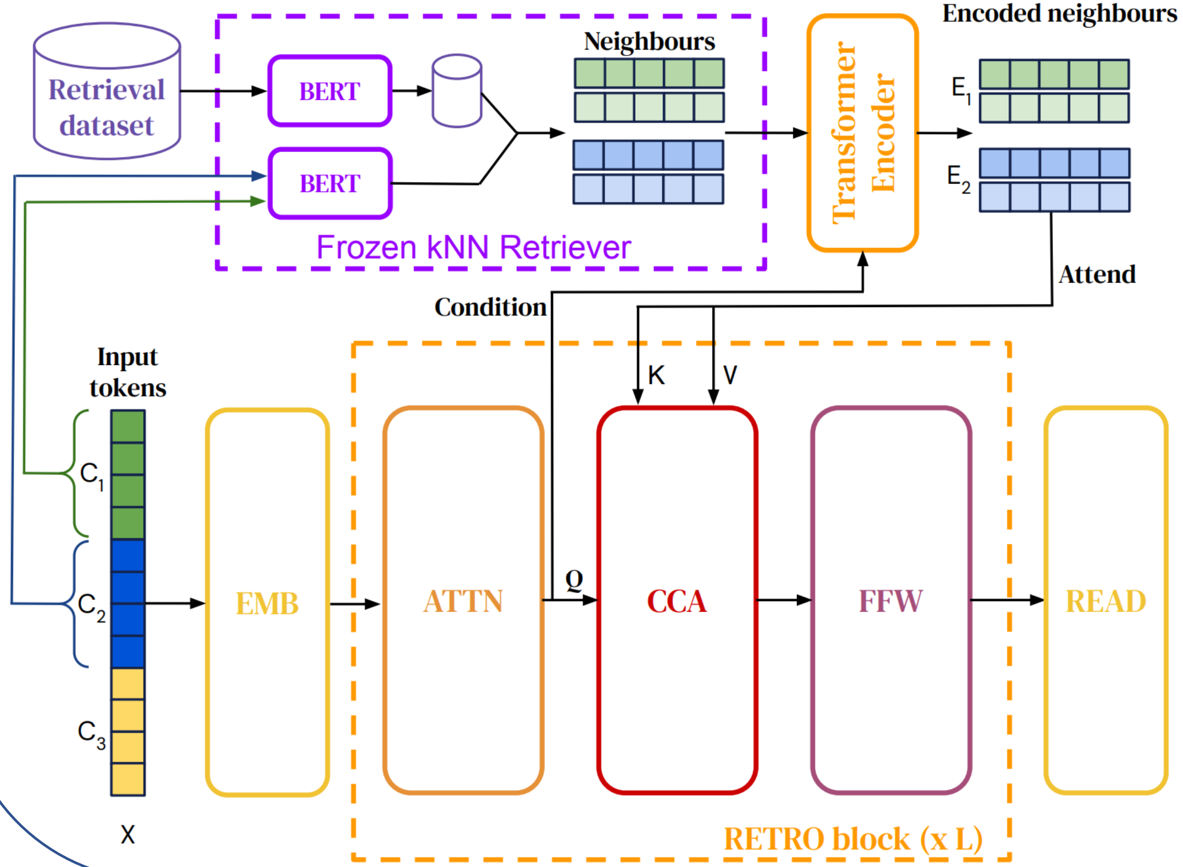
K. Guu, K. Lee, Z. Tung et al., "REALM: retrieval-augmented language model pre-training," ICML, 2020.

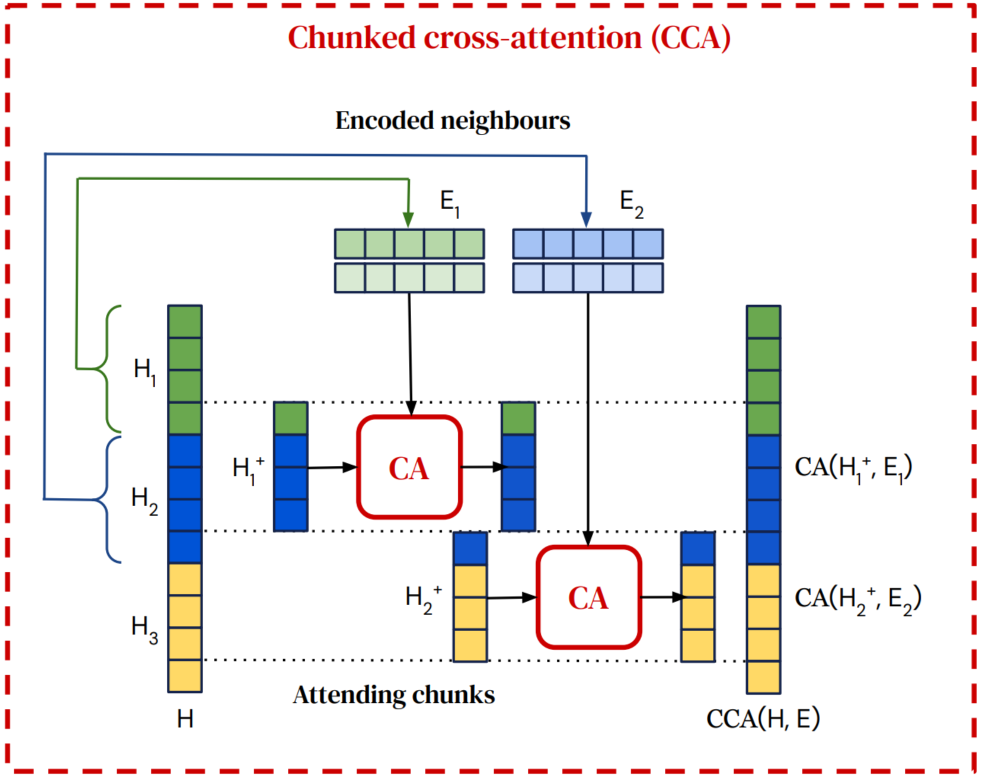
Retrieved Results Integration

Intermediate-layer Integration - RETRO

Key idea: a “retrieval-enhanced” autoregressive language model.
Use a chunked cross-attention module to incorporate the retrieved text.



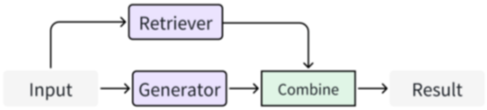


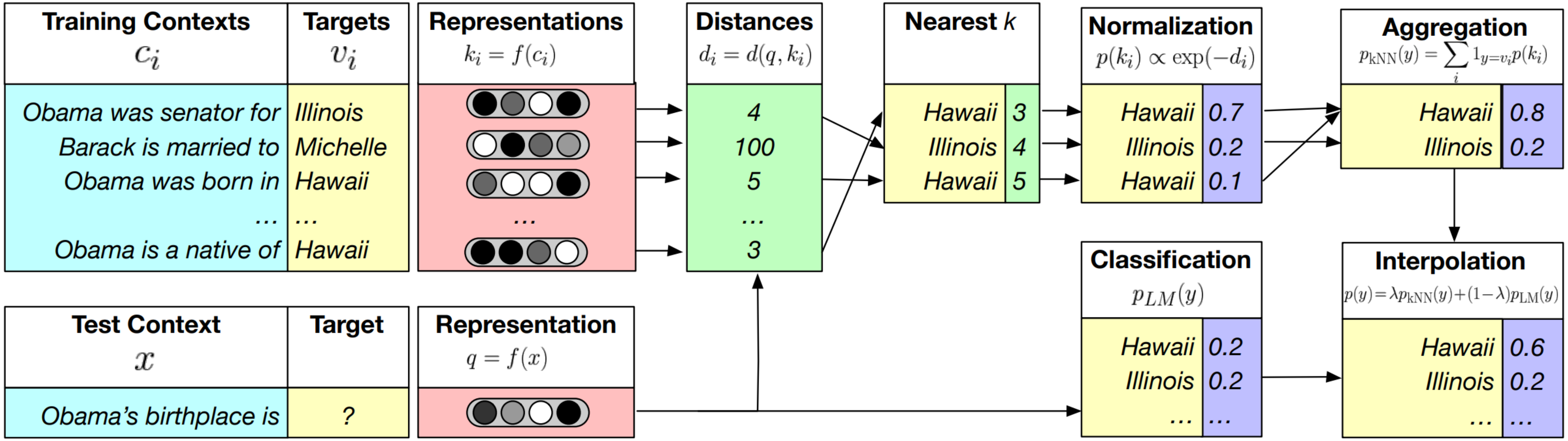


Borgeaud et al. 2022. Improving Language Models by Retrieving from Trillions of Tokens

Retrieved Results Integration

Output-layer Integration – kNN-LM





Key idea: Combining retrieved probabilities and predicted ones in generation

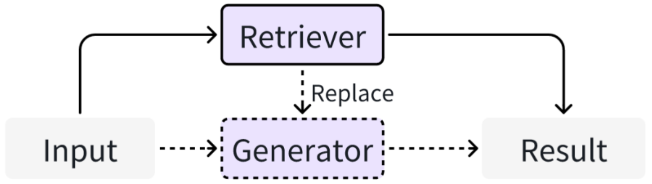
The approach can be applied to any neural language model.

Model	Perplexity (↓) Dev	Test	# Trainable Params
Baevski & Auli (2019)	17.96	18.65	247M
+Transformer-XL (Dai et al., 2019)	-	18.30	257M
+Phrase Induction (Luo et al., 2019)	-	17.40	257M
Base LM (Baevski & Auli, 2019)	17.96	18.65	247M
+kNN-LM	**16.06**	**16.12**	247M
+Continuous Cache (Grave et al., 2017c)	17.67	18.27	247M
+kNN-LM + Continuous Cache	**15.81**	**15.79**	247M

Khandelwal et al., "Generalization through memorization: Nearest neighbor language models," in ICLR, 2020.

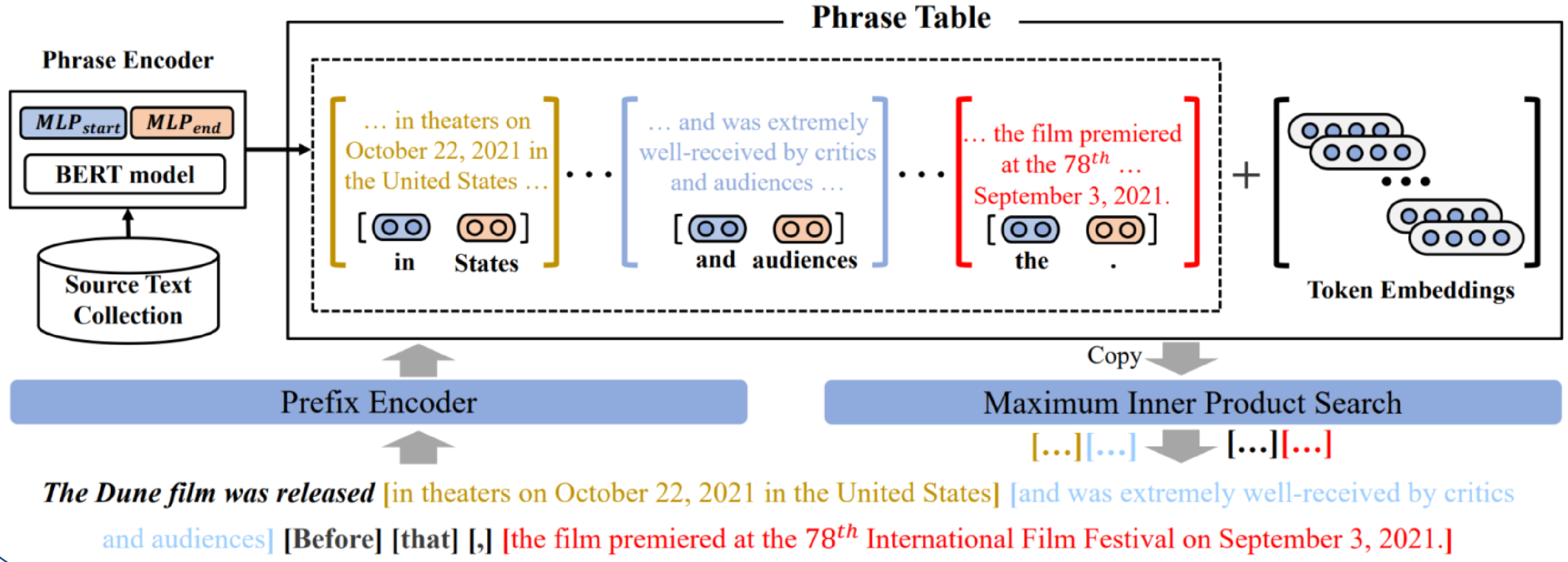
Retrieved Results Integration

Speculative RAG - CoG



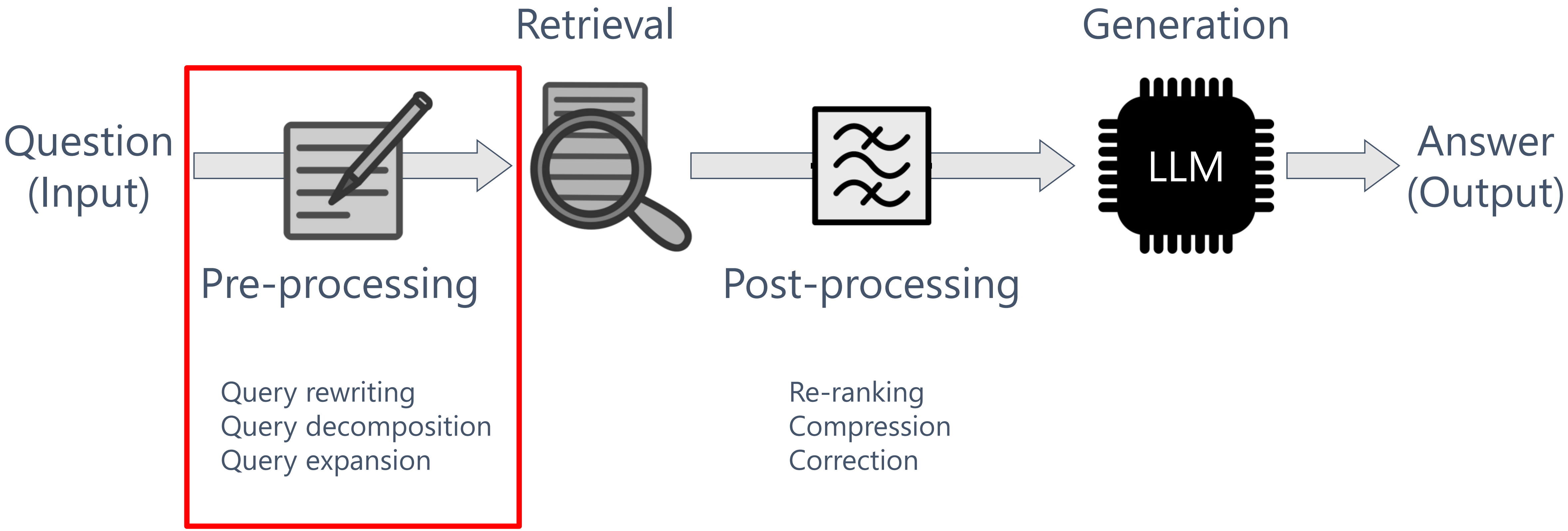
Key Idea: Formulate text generation as progressively copying text segments (e.g., words or phrases) from an existing text collection.

Generating text by retrieving **semantically coherent** and **fluent** phrases from other documents.



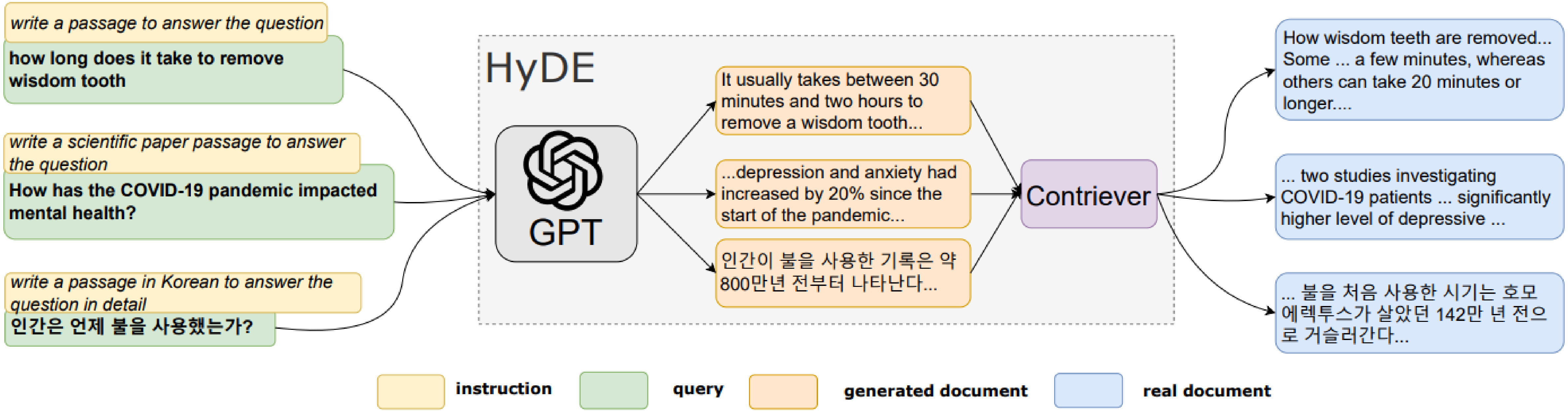
Tian Lan, et al. “Copy is All You Need.” In ICLR 2022

Improving RAG performance with pre-processing and post-processing



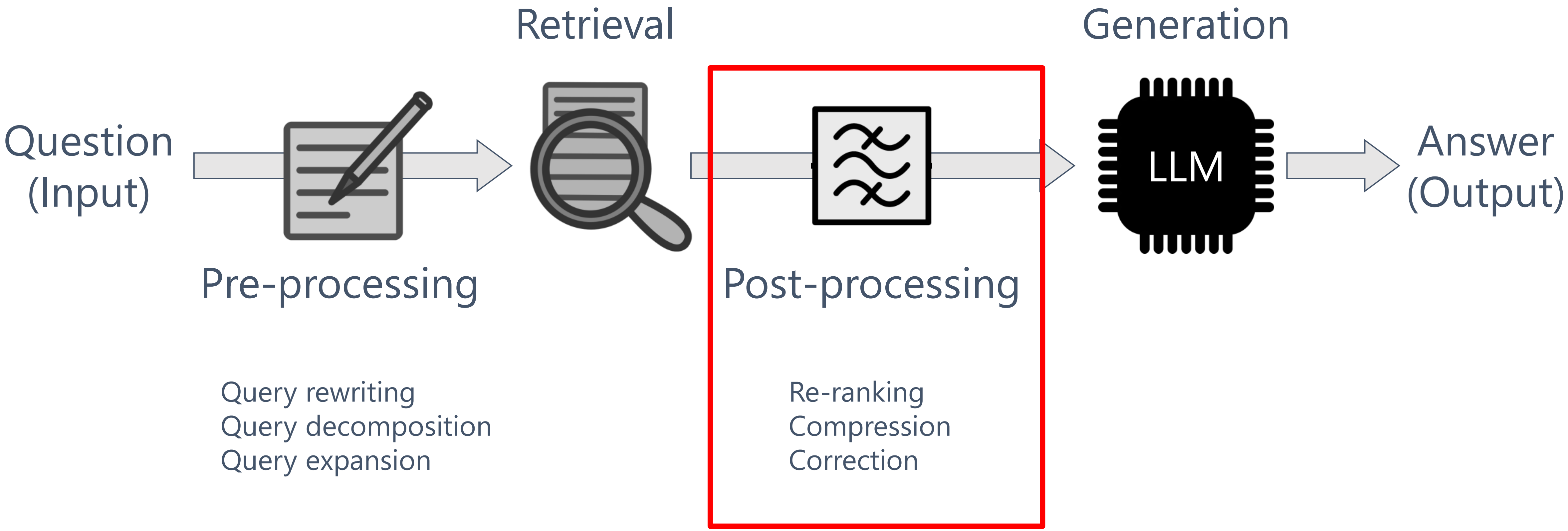
HyDE: Hypothetical Document Embeddings

Key Idea: Generate a "fake" hypothetical document that captures relevant textual patterns from the initial query. Then, encode each hypothetical document into an embedding vector and average them.



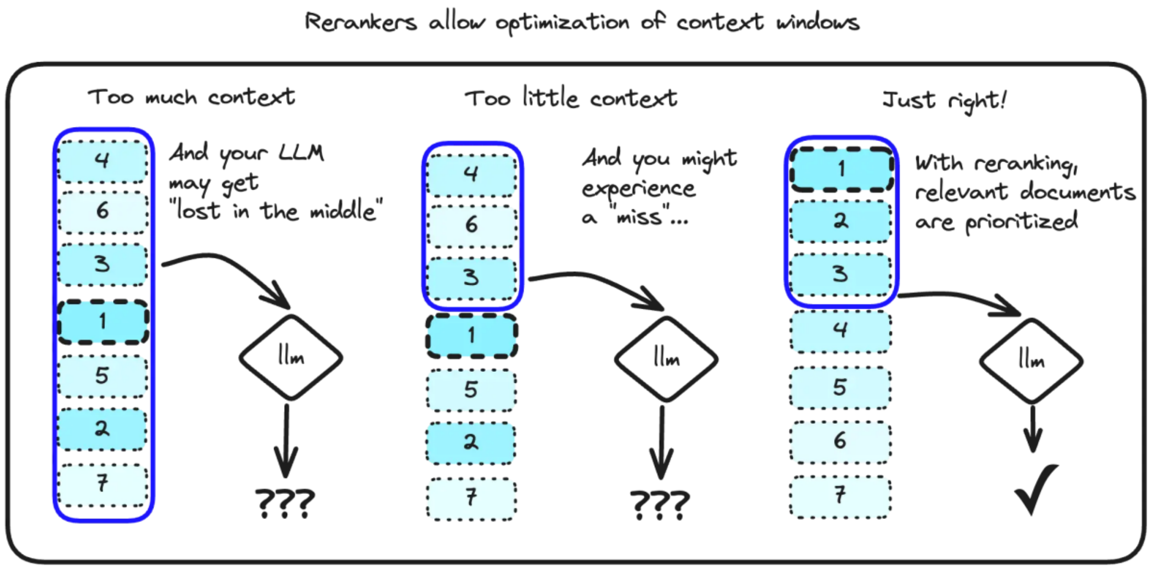
Gao et al. 2022. "Precise zero-shot dense retrieval without relevance labels"

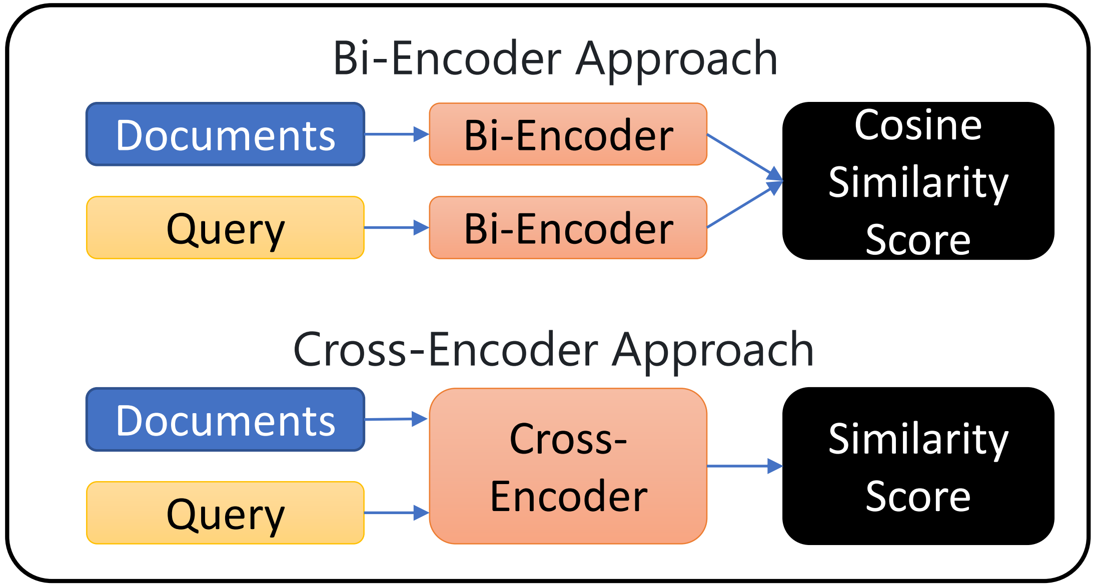
Improving RAG performance with pre-processing and post-processing

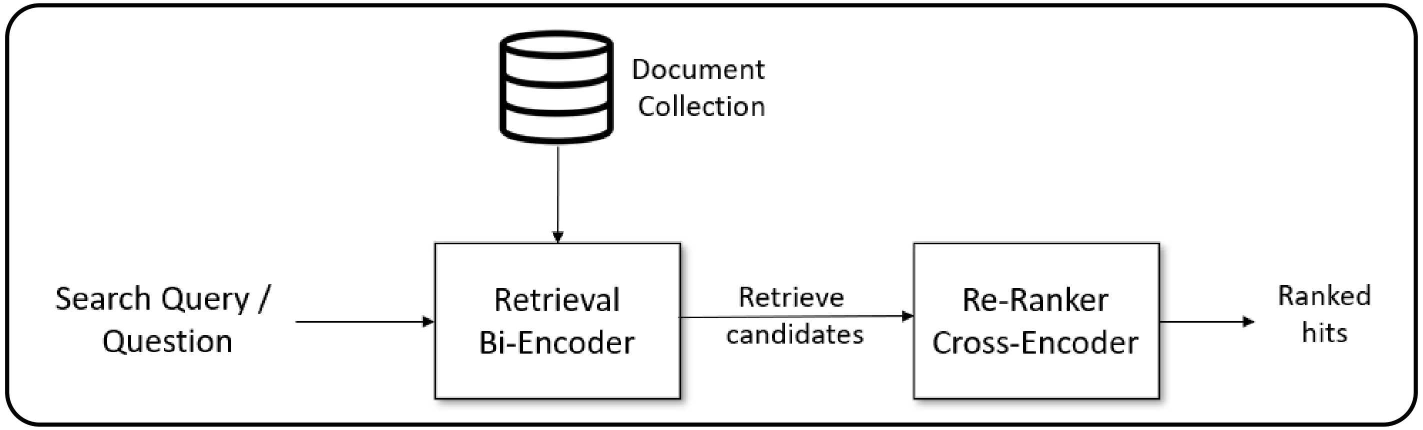


Post-retrieval Processing

Reranking using Cross-encoders



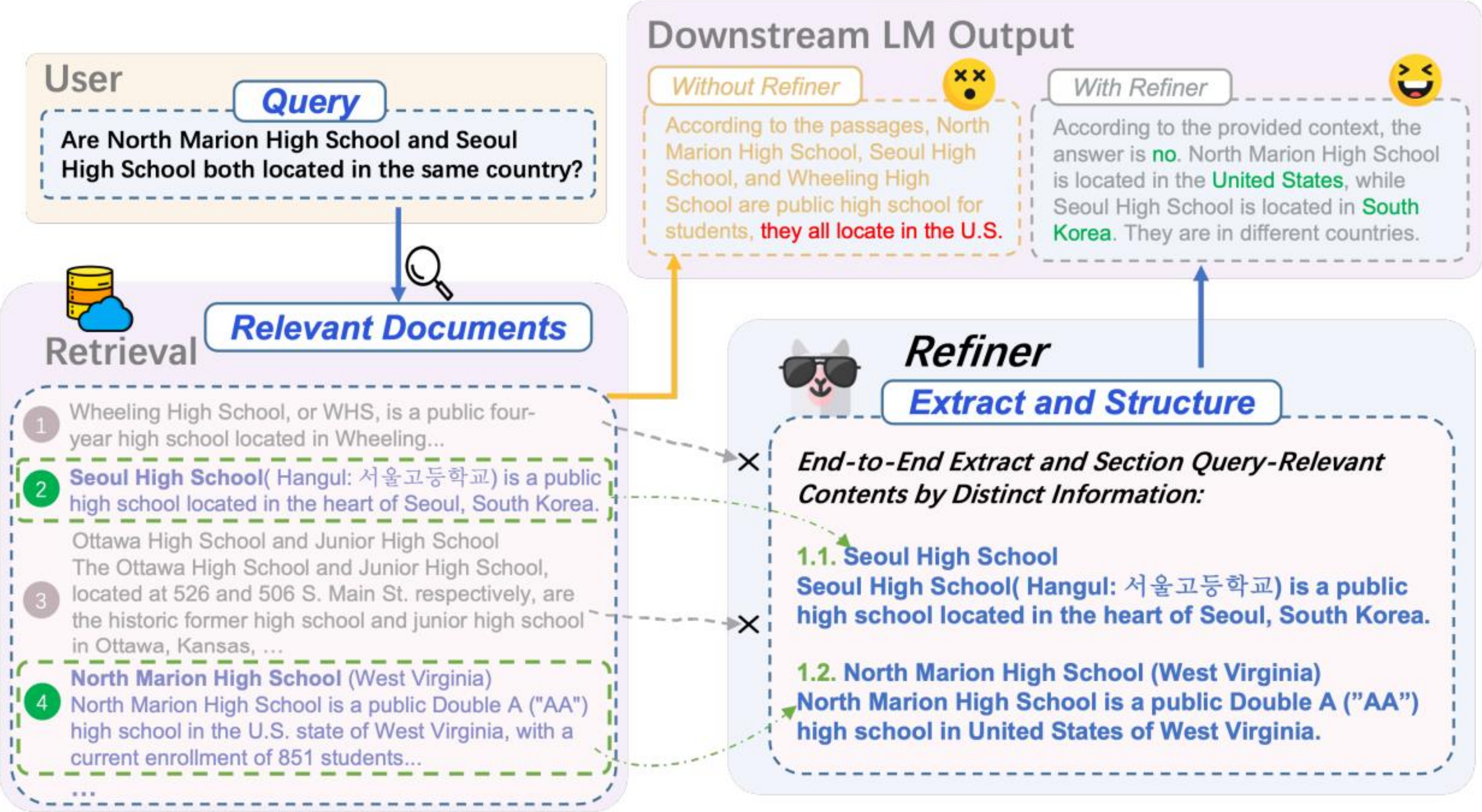




Beyond the Basics of RAG https://parlance-labs.com/education/rag/ben.html

Refiner: Restructure Retrieved Content

Key Idea: Extract and restructure document chunks, organizing query-relevant and context-completed content into sections.

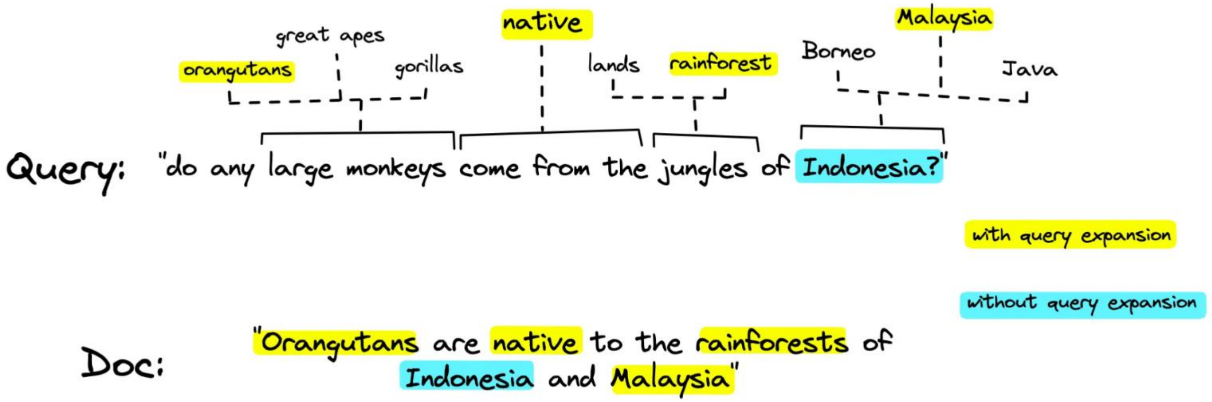


Li, et al. "Refiner: Restructure Retrieval Content Efficiently to Advance Question-Answering Capabilities." 2024

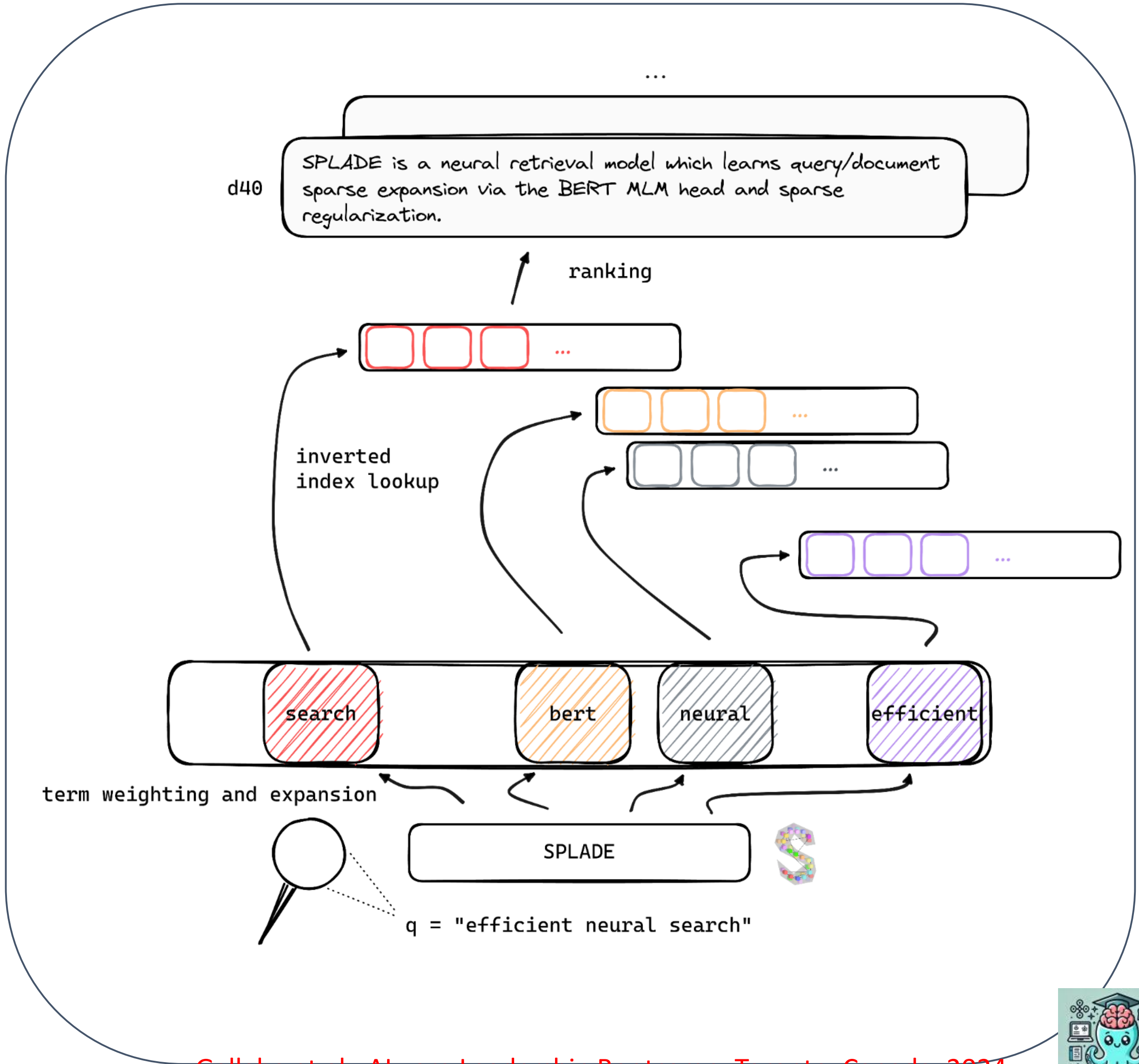
Overview of the session

- **Why RAG?:** What are the challenges in AIware development that RAG tries to address?
- **RAG Overview:** Overview of RAG and main concepts
    - Sparse Retrieval vs Dense Retrieval
    - Augmentation
    - Pre-processing and post-processing
- **Advanced Retrieval Concepts:** Combining multiple paradigms to achieve better retrieval
    - SPLADE
    - ColBERT
    - ColPali
- **Advanced RAG Patterns:** Integrating RAG with other advancements in foundation model domain
    - Contextual Retrieval
    - Self-RAG
    - Least-to-most prompting
    - IR-COT
- **Applications of RAG in SE:** Software Engineering as a case study how retrieval augmented generation has been used to improve SOTA
- **Limitations of RAG**
- **Productionizing challenges of RAG**

SPLADE - Sparse Lexical and Expansion



- A neural network-based approach, but it learns sparse representations.
- Learning **term weighting** and **expansions** minimizes **vocabulary mismatch problem**.
- For every document and query, it outputs a **weighted-term matrix**, the similarity of which is used to do the scoring.

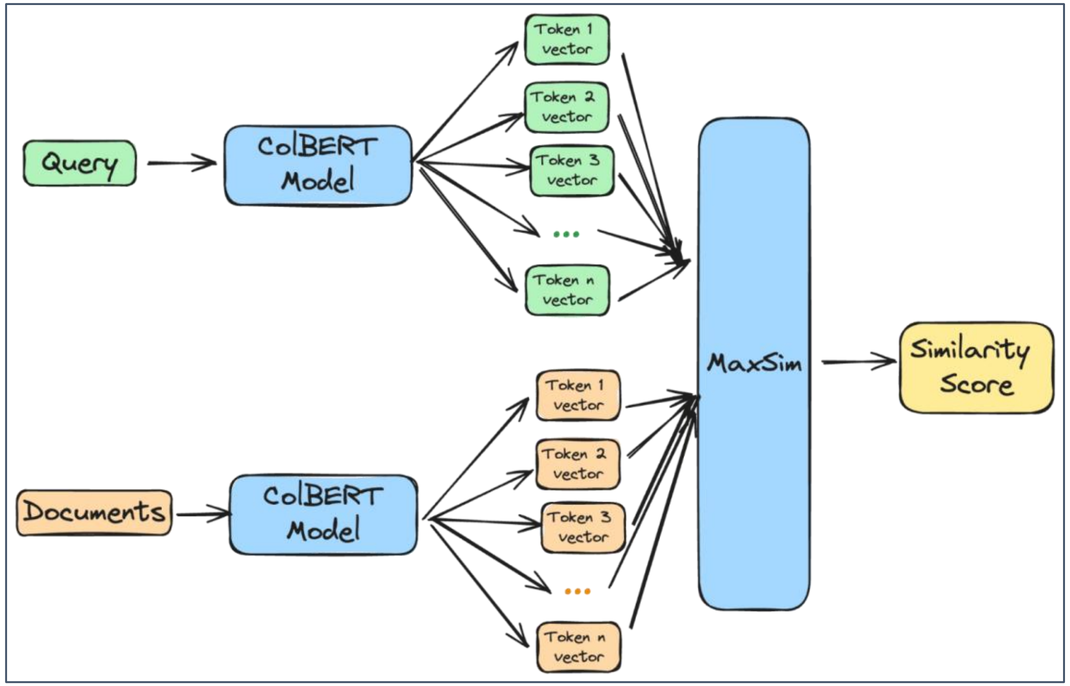


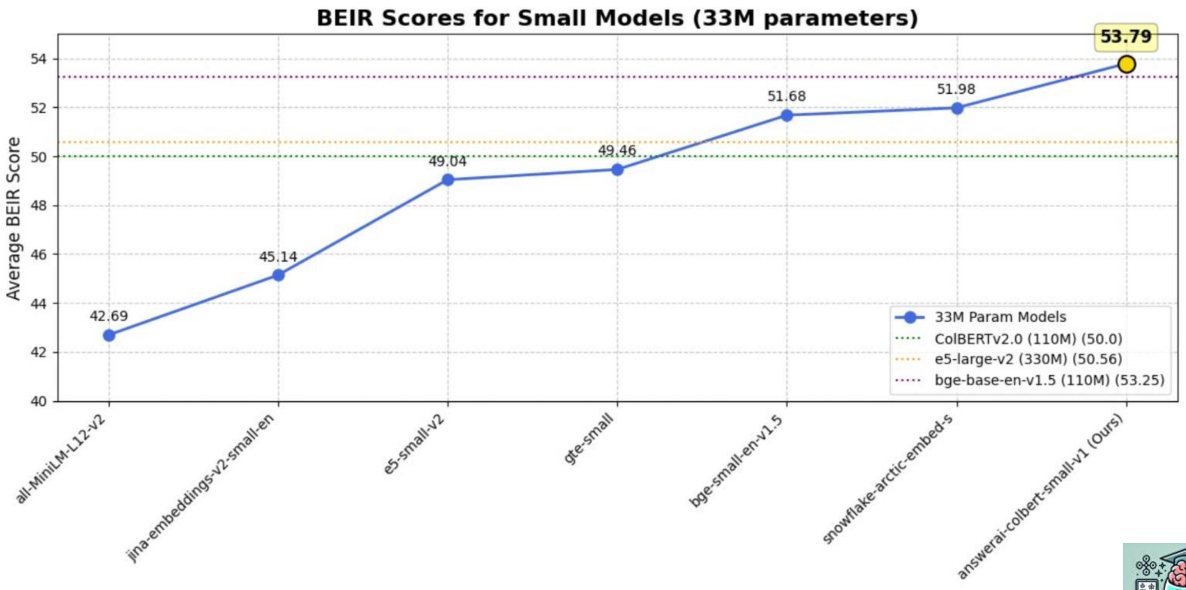
[1] T. Formal, B. Piwowarski, S. Clinchant, SPLADE: Sparse Lexical and Expansion Model for First Stage Ranking (2021), SIGIR 21
[2] T. Formal, C. Lassance, B. Piwowarski, S. Clinchant, SPLADE v2: Sparse Lexical and Expansion Model for Information Retrieval (2021)

Advanced Retrieval

ColBERT

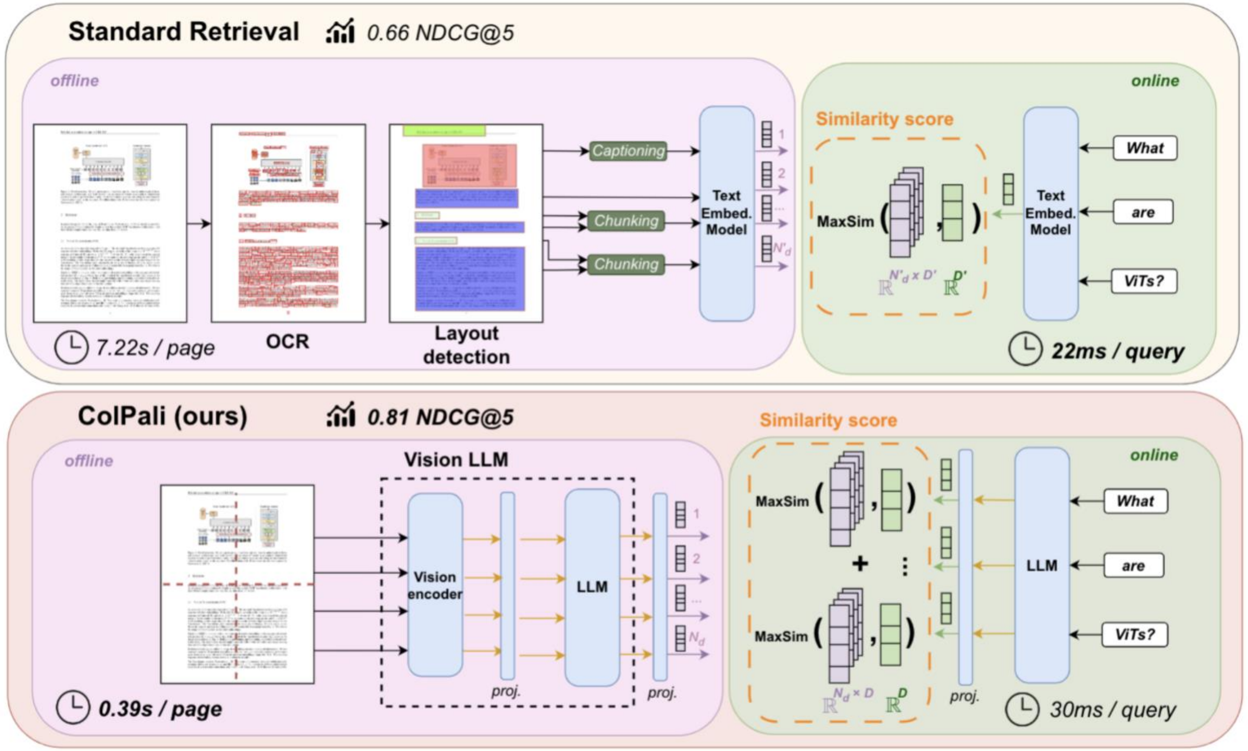
- Inspired by sparse, dense, and re-ranking methods.
- A multi-vector representation method
- All documents representations are pre-computed in isolation, with no query awareness.
- Queries are encoded at inference-time.
- The token-level representations are kept until the scoring time.
- **MaxSim** operator focuses on interactions between individual query and document tokens, rather than the full document.
- Strategies to save storage: downcasting layer, aggressive quantization, pooling

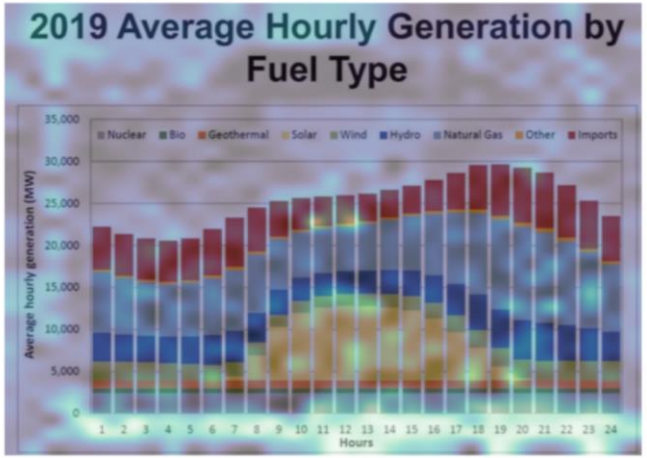




Advanced Retrieval
ColPali

- Multi-vector + Multi-model approach
- **Key idea:** Bypass complexity by using images ("screenshots") of the document pages directly during indexing.
- Applies the ColBERT late-interaction approach to image tokens, generated by a large Vision-Language Model (VLM) such as PaliGemma.
- Allows querying images, paragraphs and tables from any document with no pre-processing.





Query: "Which hour of the day had the highest overall eletricity generation in 2019?"

ColPali can answer fine-grained questions by storing image token-level information.

Faysse et al., ColPali: Efficient Document Retrieval with Vision Language Models

Overview of the session

- **Why RAG?:** What are the challenges in AIware development that RAG tries to address?
- **RAG Overview:** Overview of RAG and main concepts
    - Sparse Retrieval vs Dense Retrieval
    - Augmentation
    - Pre-processing and post-processing
- **Advanced Retrieval Concepts:** Combining multiple paradigms to achieve better retrieval
    - SPLADE
    - ColBERT
    - ColPali
- **Advanced RAG Patterns:** Integrating RAG with other advancements in foundation model domain
    - Contextual Retrieval
    - Self-RAG
    - Least-to-most prompting
    - IR-COT
- **Applications of RAG in SE:** Software Engineering as a case study how retrieval augmented generation has been used to improve SOTA
- **Limitations of RAG**
- **Productionizing challenges of RAG**

Contextual Retrieval - The intuition

User Question: "What was the revenue growth for ACME Corp in Q2 2023?"

Original retrieved chunk: "The company's revenue grew by 3% over the previous quarter."

Which company?

Contextualized chunk: "This chunk is from an SEC filing on ACME corp's performance in Q2 2023; the previous quarter's revenue was $314 million. The company's revenue grew by 3% over the previous quarter."

Added context

Documents are typically split into smaller chunks for efficient retrieval.

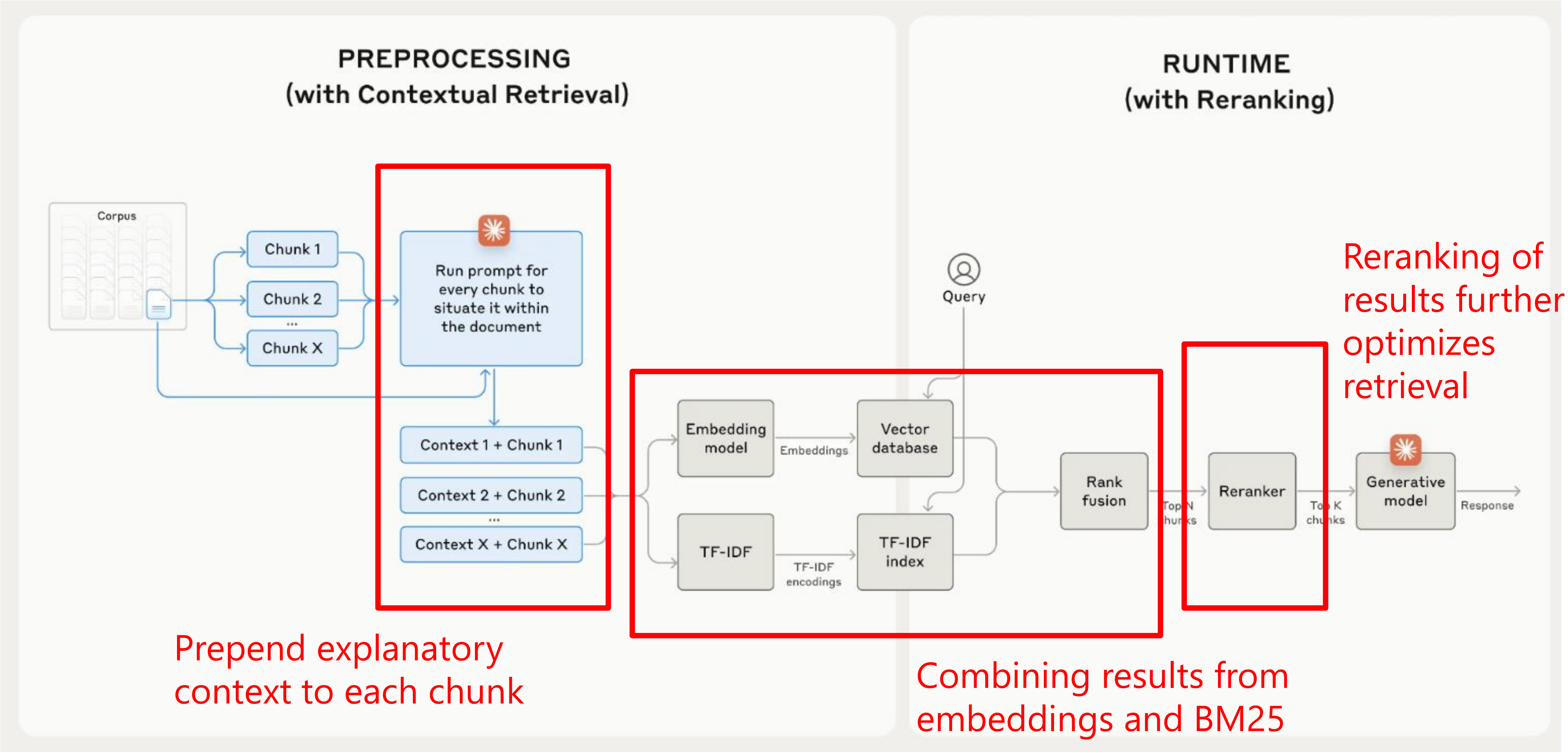
But individual chunks lack sufficient context.

Prepending chunk-specific explanatory context (50-100 tokens) to each chunk before embedding may help.

But far too much work to manually annotate all chunks.

https://www.anthropic.com/news/contextual-retrieval

Contextual Retrieval - Putting it all together



Prepend explanatory context to each chunk

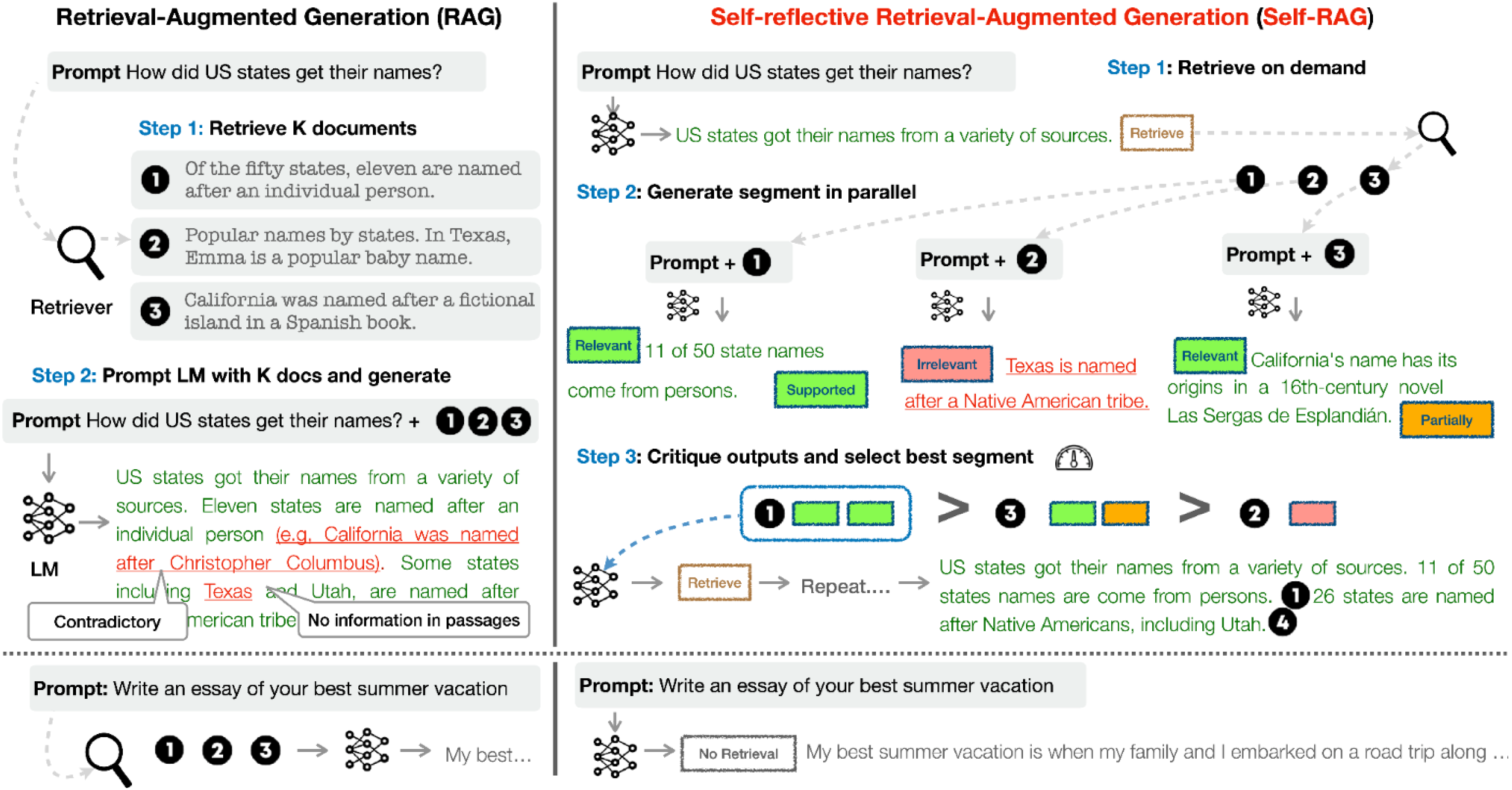
Combining results from embeddings and BM25

Reranking of results further optimizes retrieval

https://www.anthropic.com/news/contextual-retrieval

Advanced RAG Patterns

Self-Reflective RAG (SELF-RAG)



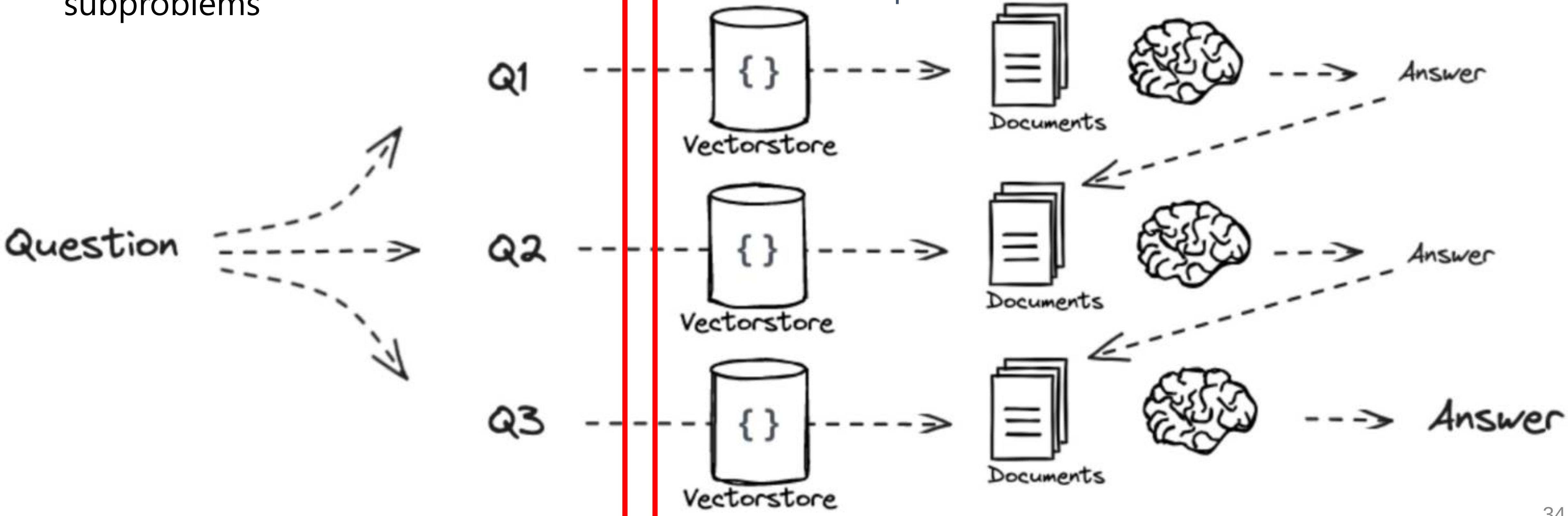
Asai et al., Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection

Gallaba et al., AIware Leadership Bootcamp, Toronto, Canada, 2024

Least-to-most prompting

Stage 1
Query the language model to decompose the problem into subproblems

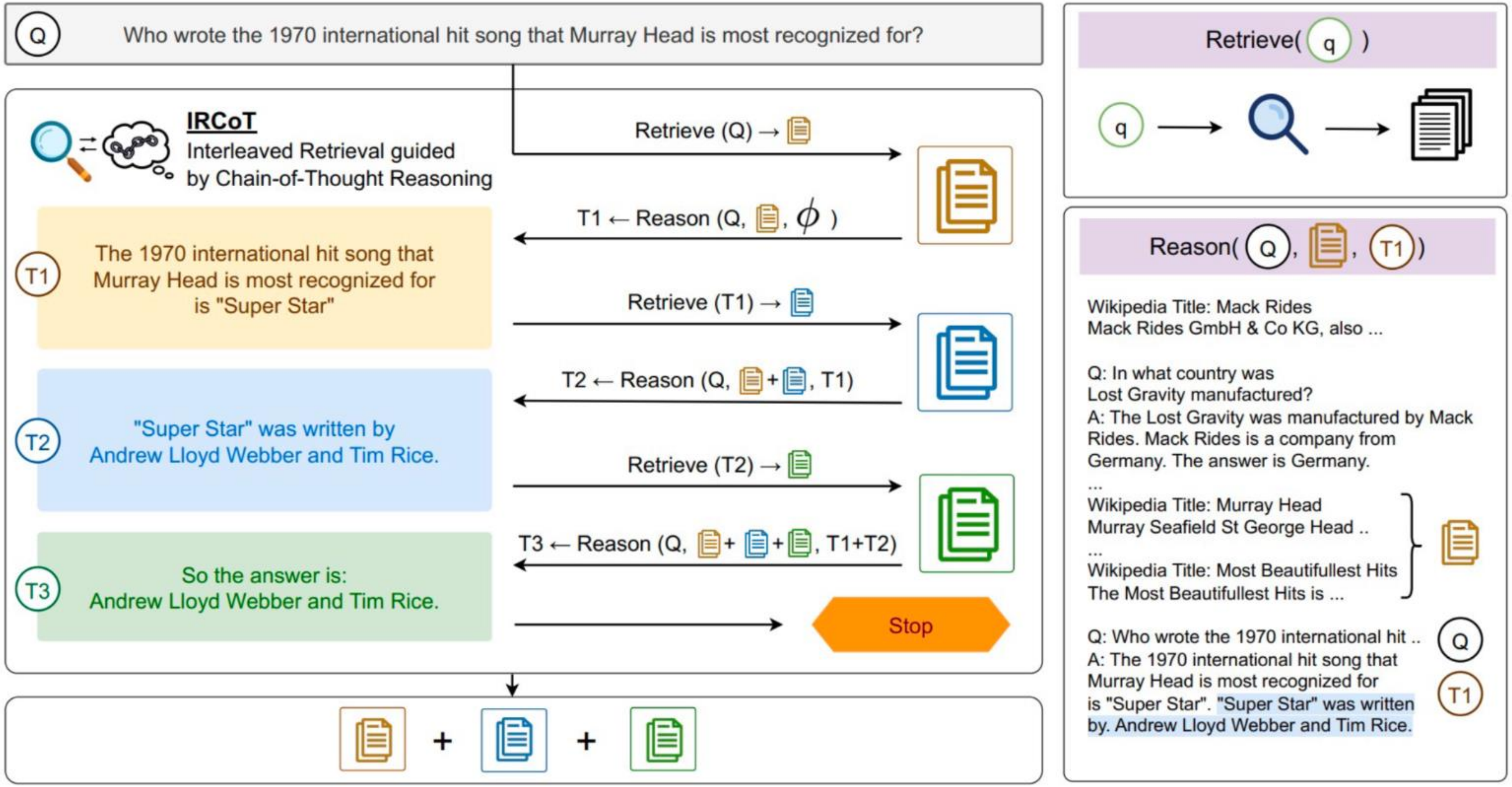
Stage 2
Query the language model to sequentially solve the subproblems.



Zhou, et al.2023. "Least-to-most prompting enables complex reasoning in large language models"

Advanced RAG Patterns

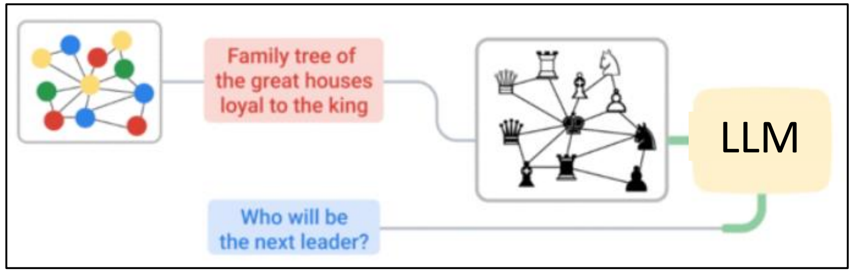
IRCoT - Interleaving retrieval with Chain-of-thought reasoning



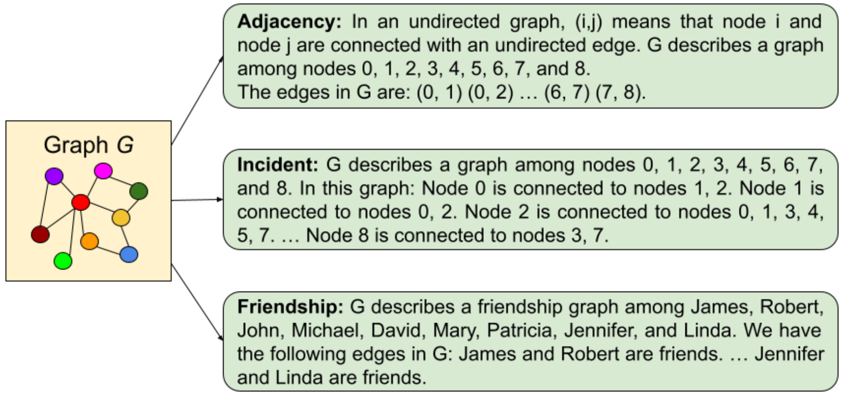
Advanced RAG Patterns

Graphs & RAG

Encoding Graphs via Text



LLMs struggle on most graph tasks.



Encoding matters. **Incident encoding** works for most tasks in general.

GraphRAG

Key Idea: Using LLMs, it parses data to create a knowledge graph and answer user questions about a user-provided private dataset.

Indexing

- extract entities, relationships and claims from raw text
- perform community detection in entities
- generate community summaries and reports
- embed entities into a graph vector space
- embed text chunks into a textual vector space

Querying

- **Global Search –** Use community summaries for questions about corpus
- **Local Search –** reasoning about specific entities by fanning-out to their neighbors and associated concepts.

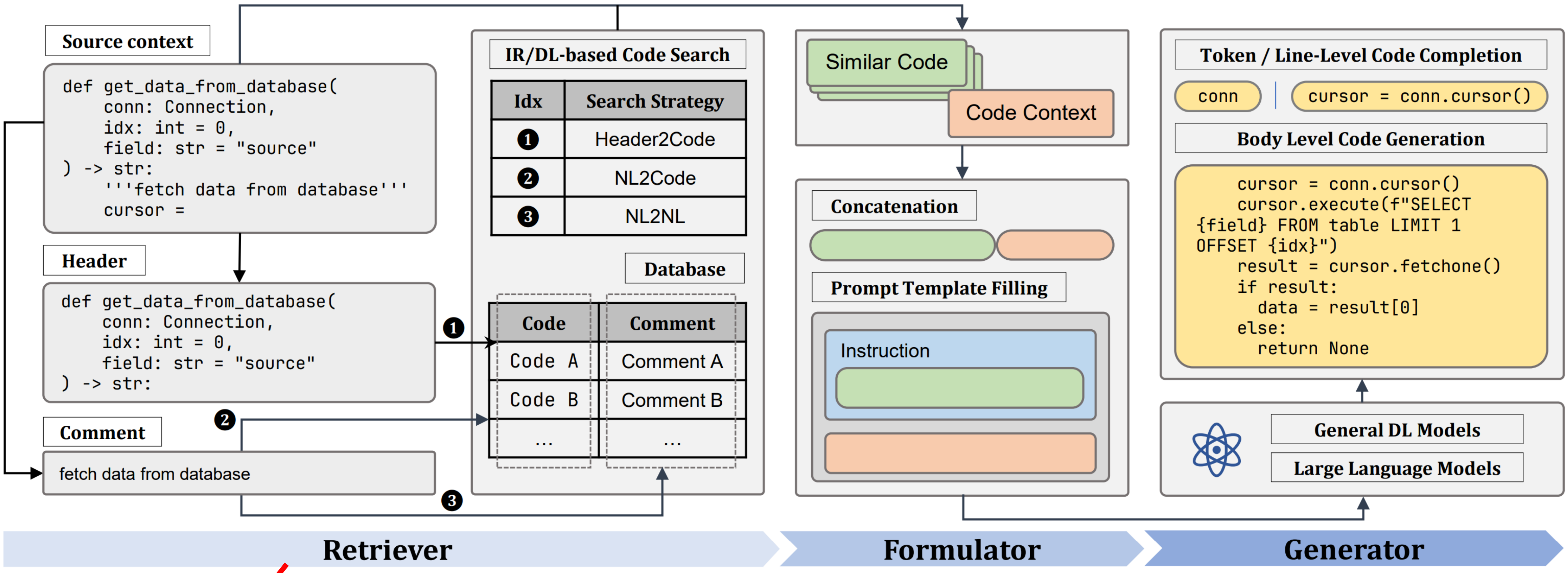
Fatemi et al., Talk like a Graph: Encoding Graphs for Large Language Models, ICLR 2024
GraphRAG https://www.microsoft.com/en-us/research/blog/graphrag-unlocking-llm-discovery-on-narrative-private-data/

Overview of the session

- **Why RAG?:** What are the challenges in AIware development that RAG tries to address?
- **RAG Overview:** Overview of RAG and main concepts
  - Sparse Retrieval vs Dense Retrieval
  - Augmentation
  - Pre-processing and post-processing
- **Advanced Retrieval Concepts:** Combining multiple paradigms to achieve better retrieval
  - SPLADE
  - ColBERT
  - ColPali
- **Advanced RAG Patterns:** Integrating RAG with other advancements in foundation model domain
  - Contextual Retrieval
  - Self-RAG
  - Least-to-most prompting
  - IR-COT
- **Applications of RAG in SE:** Software Engineering as a case study how retrieval augmented generation has been used to improve SOTA
- **Limitations of RAG**
- **Productionizing challenges of RAG**

Software Engineering applications of RAG

Code Generation

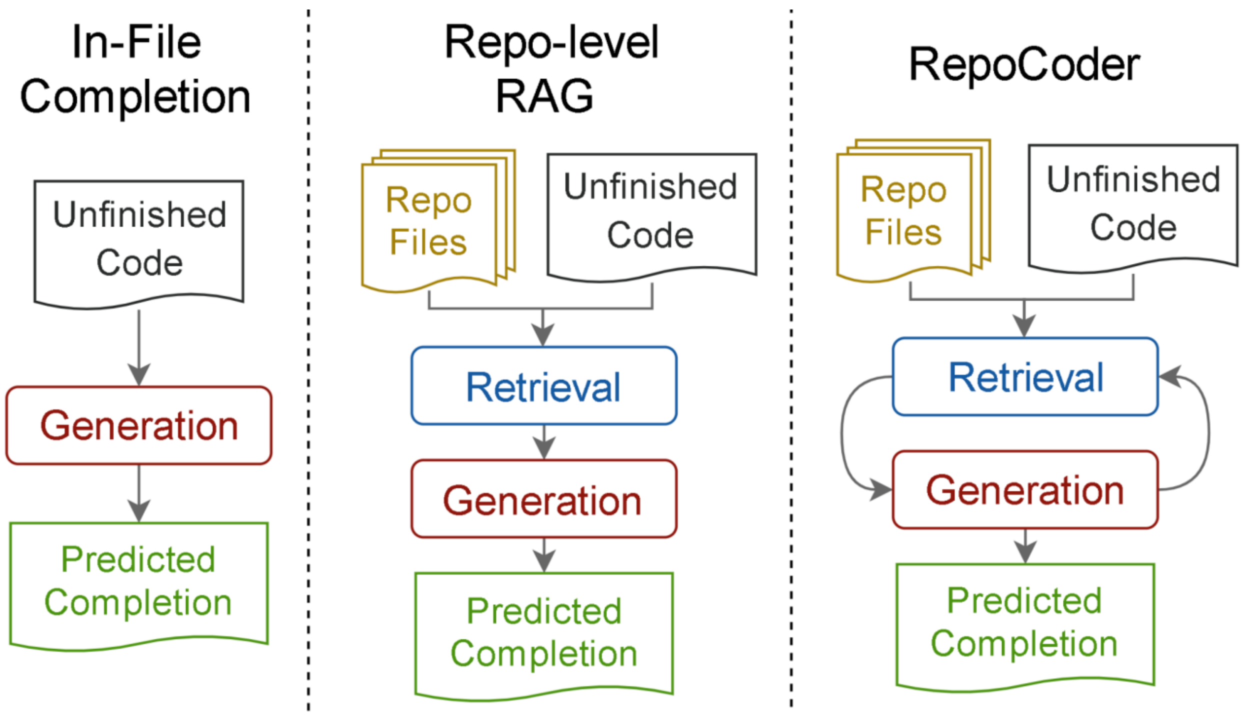


The retriever uses queries in natural language or code to find similar code, with IR-based or DL-based code search tools

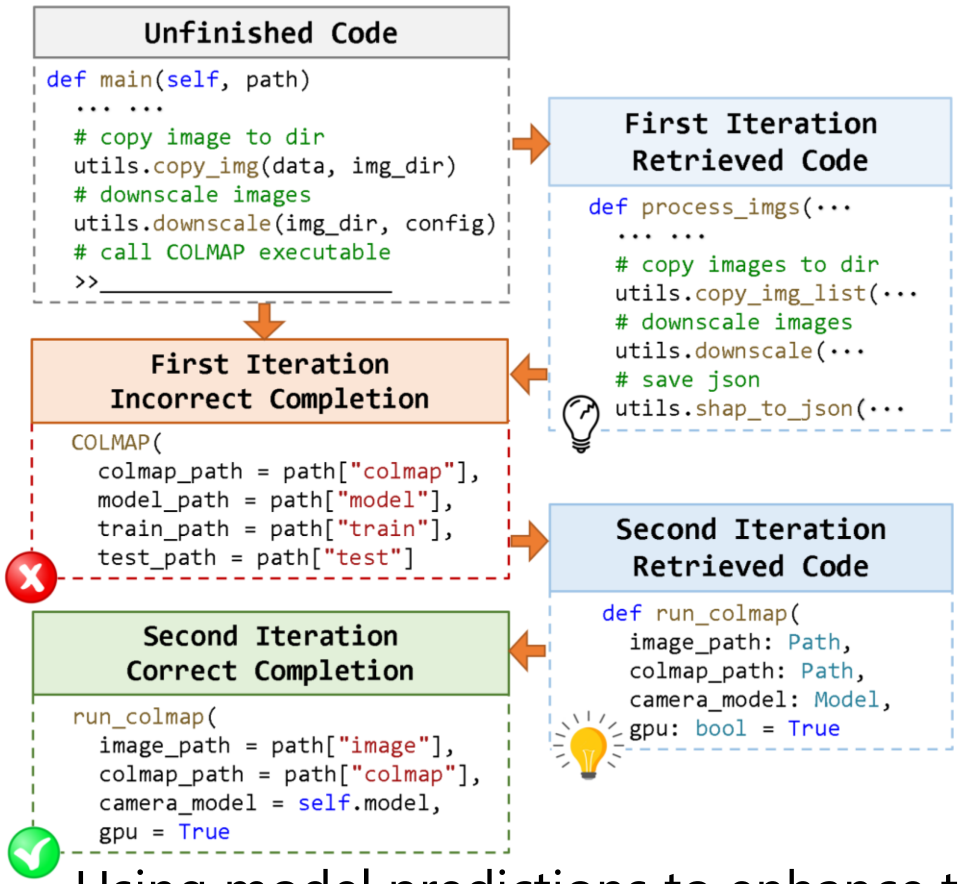
Chen et al., Code Search is All You Need? Improving Code Suggestions with Code Search, ICSE'24

Software Engineering applications of RAG

Code Completion - RepoCoder



Iterative retrieval-generation of RepoCoder

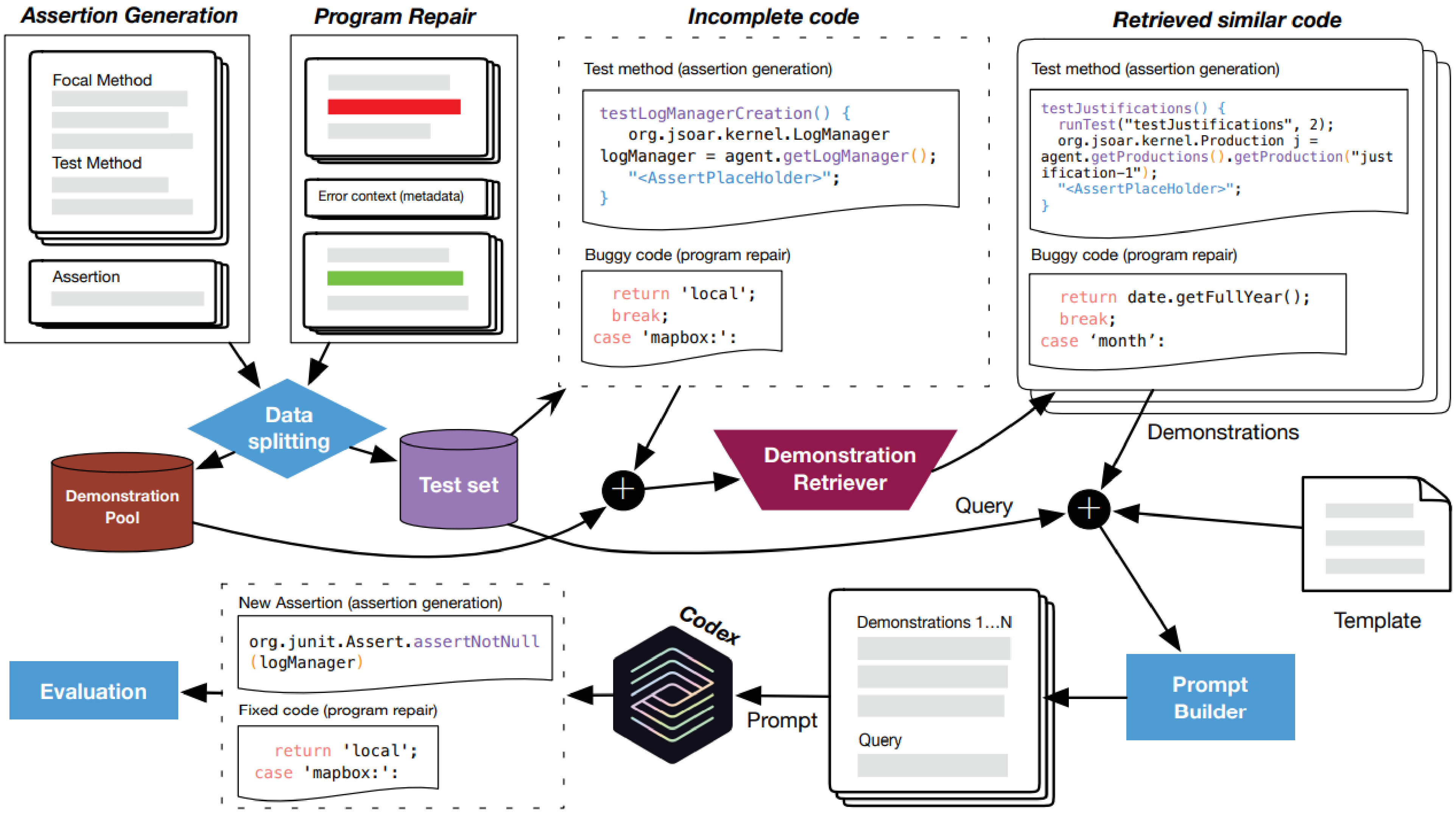


Using model predictions to enhance the performance of code retrieval

RepoCoder: Repository-Level Code Completion Through Iterative Retrieval and Generation
https://aclanthology.org/2023.emnlp-main.151/
https://github.com/microsoft/CodeT/tree/main/RepoCoder

Software Engineering applications of RAG

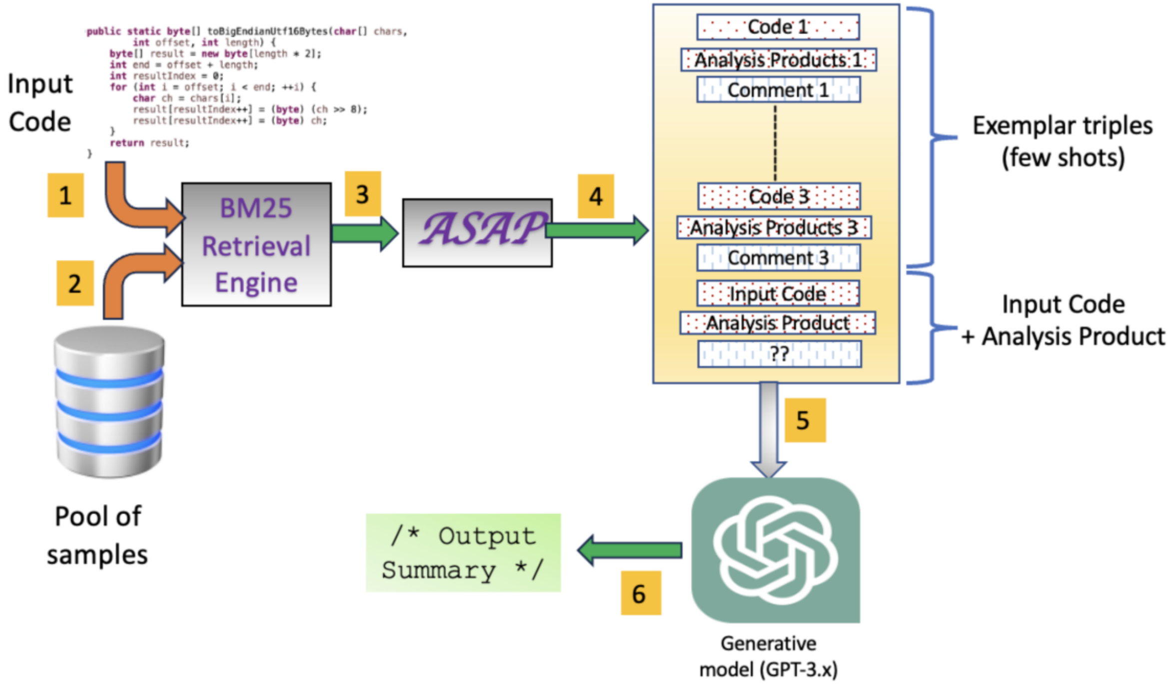
Automated Program Repair - CEDAR



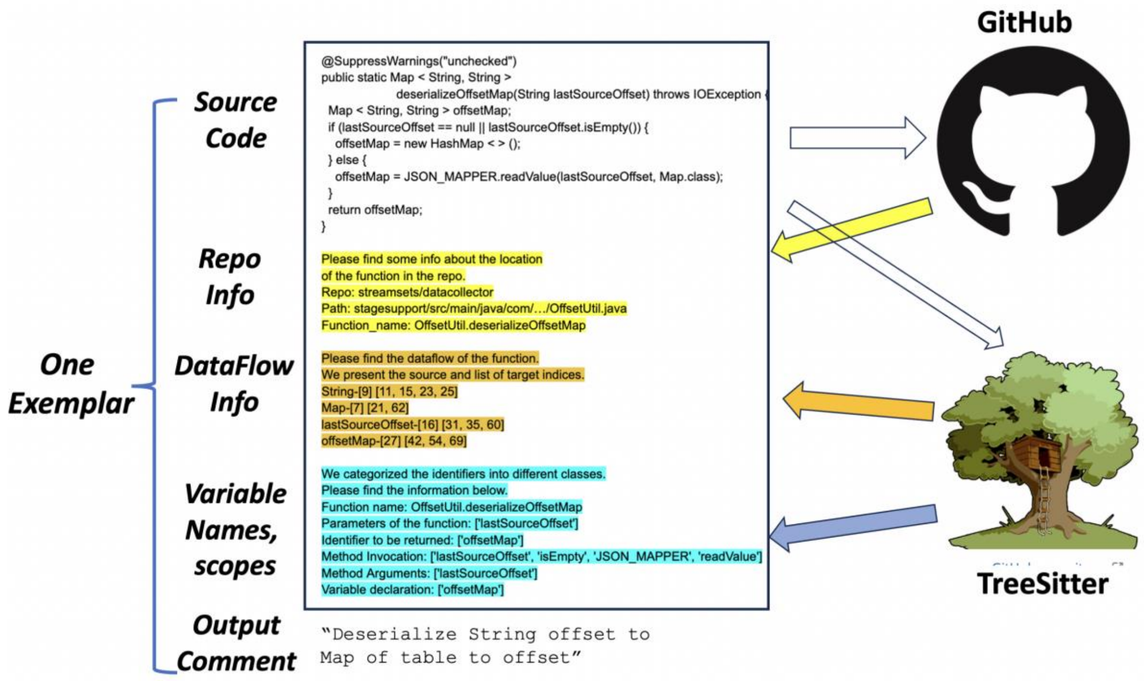
Nashid et al., 2023. "Retrieval-Based Prompt Selection for Code-Related Few-Shot Learning"

Software Engineering applications of RAG

Code Summarization



Retrieve best matching samples. Three input output pairs are used in produce a prompt with exemplars.



Exemplars have components from multiple sources.

Ahmed et al., Automatic Semantic Augmentation of Language Model Prompts (for Code Summarization), ICSE'24

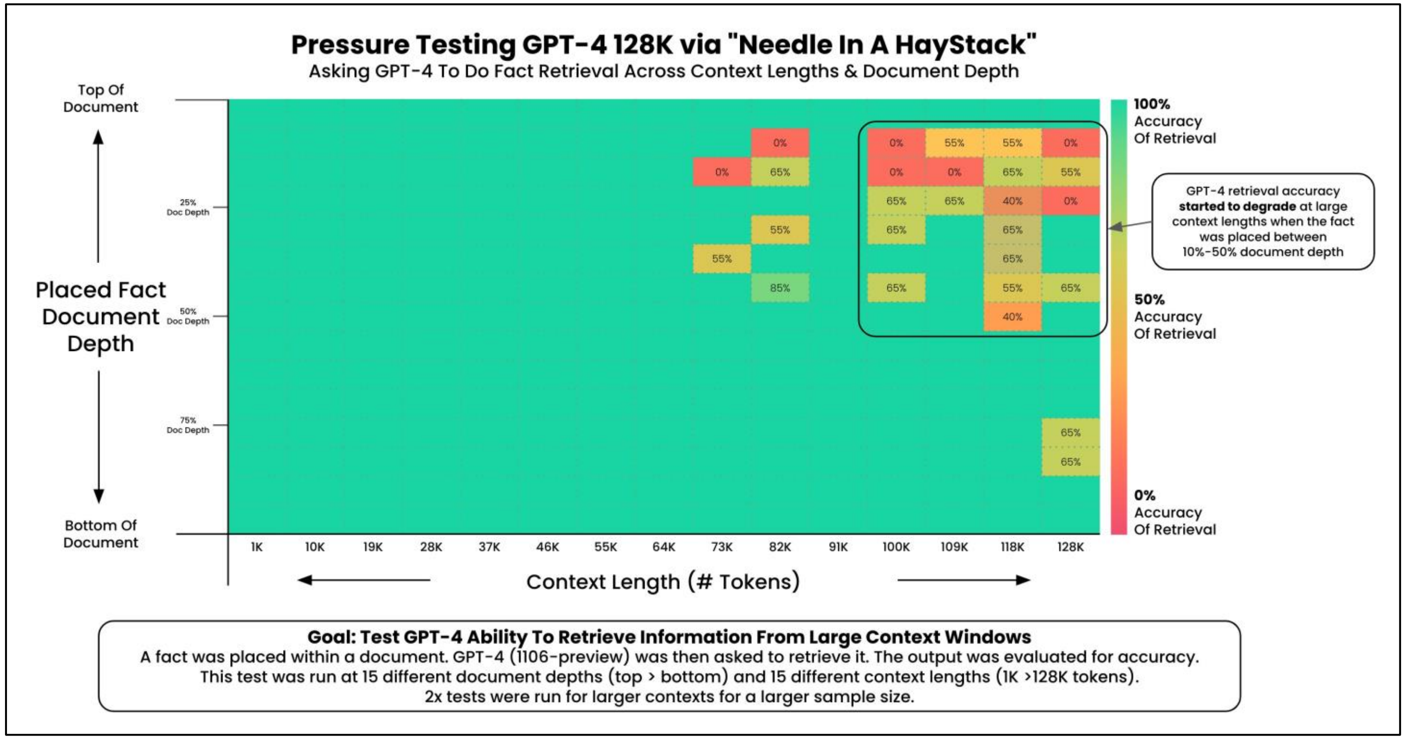
Overview of the session

- **Why RAG?:** What are the challenges in AIware development that RAG tries to address?
- **RAG Overview:** Overview of RAG and main concepts
  - Sparse Retrieval vs Dense Retrieval
  - Augmentation
  - Pre-processing and post-processing
- **Advanced Retrieval Concepts:** Combining multiple paradigms to achieve better retrieval
  - SPLADE
  - ColBERT
  - ColPali
- **Advanced RAG Patterns:** Integrating RAG with other advancements in foundation model domain
  - Contextual Retrieval
  - Self-RAG
  - Least-to-most prompting
  - IR-COT
- **Applications of RAG in SE:** Software Engineering as a case study how retrieval augmented generation has been used to improve SOTA
- **Limitations of RAG**
- **Productionizing challenges of RAG**

Limitations of RAG

Needle in a haystack

Short Context	Less than 5k tokens equivalent to RAG on few pages
Medium Context	5k to 25k tokens equivalent to RAG on a book chapter
Long Context	40k to 100k tokens equivalent to RAG on a book



LLM In-Context Recall is Prompt Dependent, https://arxiv.org/pdf/2404.08865
Lost in the Middle: How Language Models Use Long Contexts https://arxiv.org/pdf/2307.03172
Large Language Models Can Be Easily Distracted by Irrelevant Context https://arxiv.org/pdf/2302.00093

Limitations of RAG

Irrelevant context can be harmful

Cognition @cognition_labs

Prompting o1 is noticeably different; in particular:

– Chain-of-thought and asking the model to "think out loud" are common prompts for previous models. On the contrary, we find that asking o1 to only give the final answer often performs better, since it will think before answering regardless.

– o1 requires denser context and is more sensitive to clutter and unnecessary tokens. Traditional prompting approaches often involve redundancy in giving instructions, which we found negatively impacted performance with o1.

– o1's improved intelligence trades off with increased variability in following highly prescriptive instructions.

2:07 PM · Sep 12, 2024 · **40.5K** Views

34 Reposts **12** Quotes **294** Likes **114** Bookmarks

"**Limit additional context in retrieval-augmented generation (RAG):** When providing additional context or documents, include only the most relevant information to prevent the model from overcomplicating its response."

- Open AI Documentation for reasoning models.
https://platform.openai.com/docs/guides/reasoning

LLMs Can Be Easily Distracted by Irrelevant Context.
Adding irrelevant contexts to GSM8K leads to 20+ points performance drop.

Lucy has $65 in the bank. She made a $15 deposit and then followed by a $4 withdrawal. Maria's monthly rent is $10. What is Lucy's bank balance?

↓ LLM

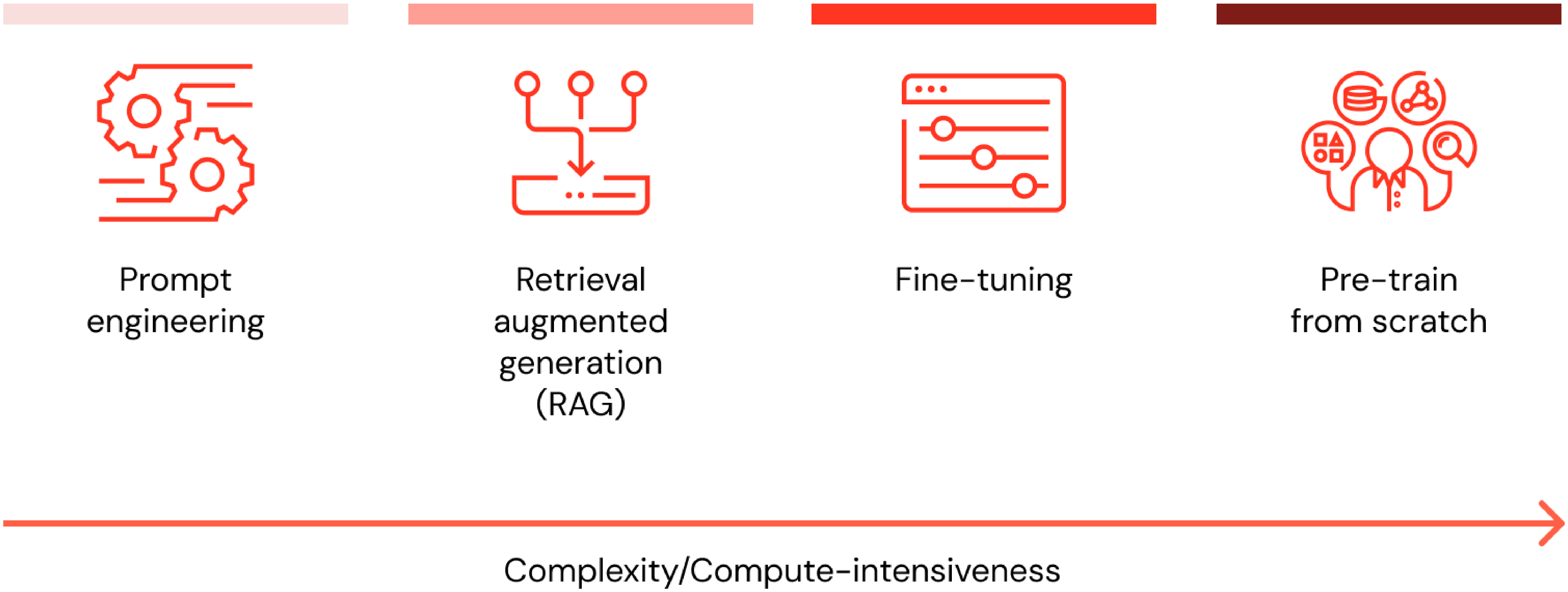
Lucy's bank balance is $65 + $15 - $4 - $10 = $66. The answer is $66.

Shi et al., ICML '23. https://arxiv.org/abs/2302.00093

Overview of the session

- ❑ **Why RAG?:** What are the challenges in AIware development that RAG tries to address?
- ❑ **RAG Overview:** Overview of RAG and main concepts
  - ❑ Sparse Retrieval vs Dense Retrieval
  - ❑ Augmentation
  - ❑ Pre-processing and post-processing
- ❑ **Advanced Retrieval Concepts:** Combining multiple paradigms to achieve better retrieval
  - ❑ SPLADE
  - ❑ ColBERT
  - ❑ ColPali
- ❑ **Advanced RAG Patterns:** Integrating RAG with other advancements in foundation model domain
  - ❑ Contextual Retrieval
  - ❑ Self-RAG
  - ❑ Least-to-most prompting
  - ❑ IR-COT
- ❑ **Applications of RAG in SE:** Software Engineering as a case study how retrieval augmented generation has been used to improve SOTA
- ❑ **Limitations of RAG**
- ❑ **Productionizing challenges of RAG**

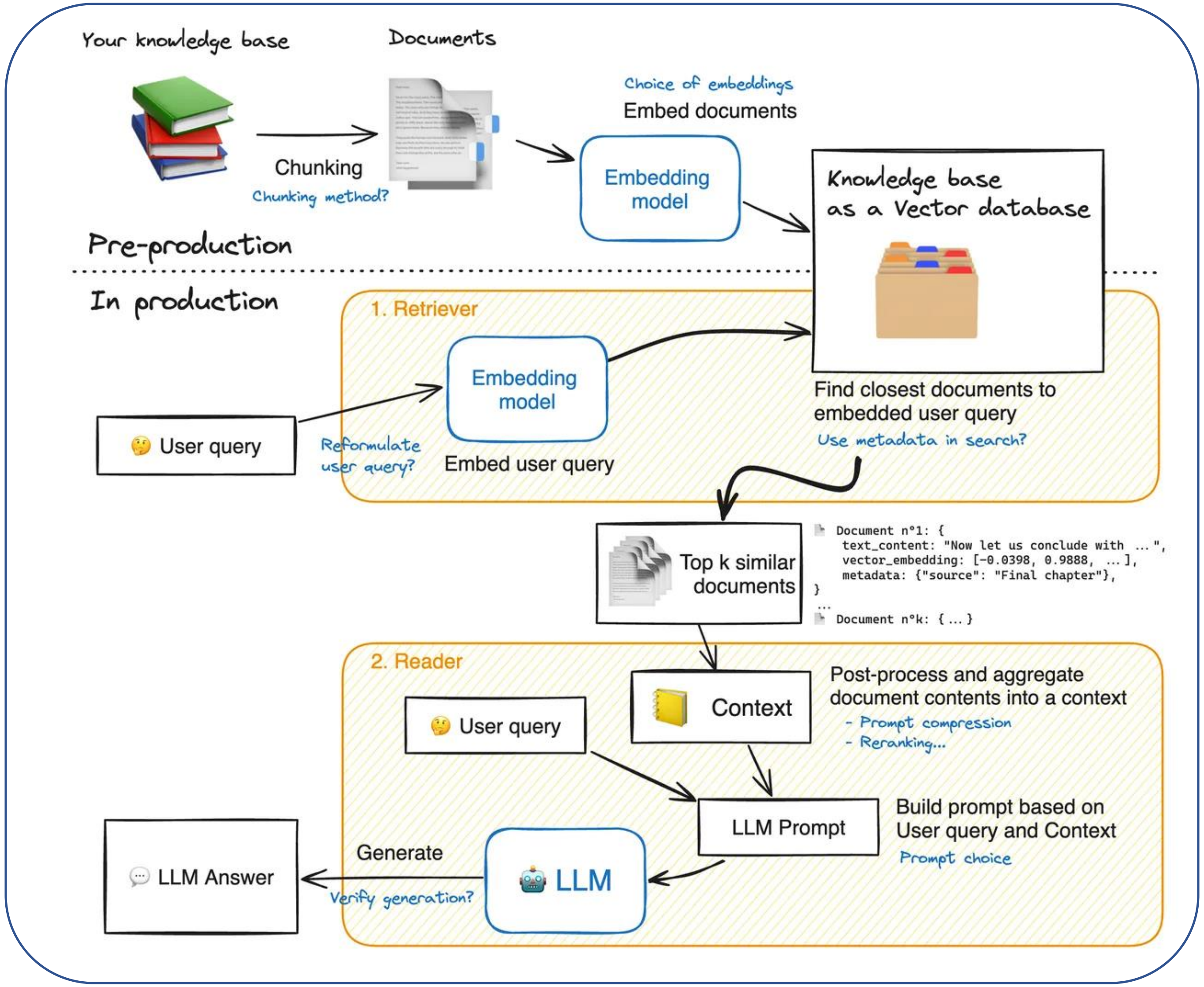
RAG vs Fine-tuning



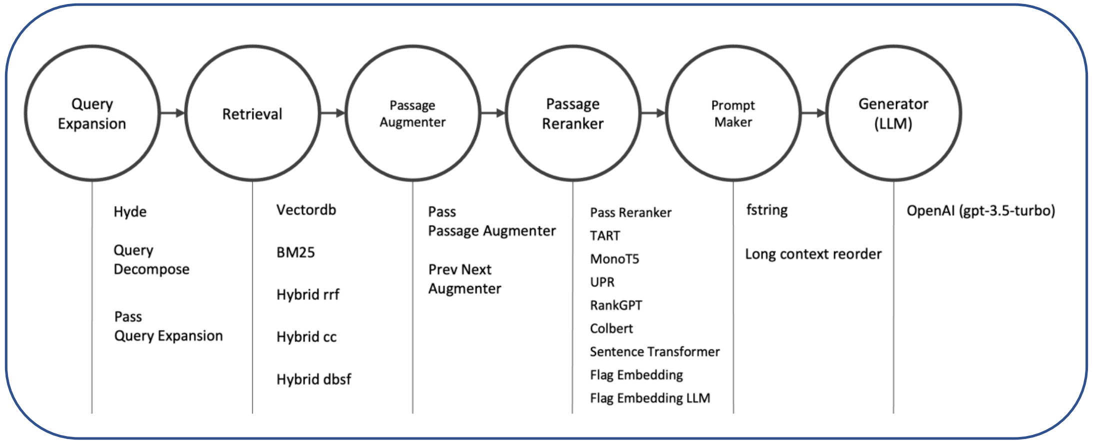
https://www.databricks.com/de/glossary/large-language-models-llm

Productionizing Challenges of RAG

Too many configuration options



AutoRAG: Automated Framework for optimization of Retrieval Augmented Generation Pipeline https://arxiv.org/pdf/2410.20878



- Continuous Experimentation
- Automated pipeline optimization

Can such techniques help?

Productionizing Challenges of RAG

Evaluating RAG Systems

Context Adherence: the degree to which a model's response aligns strictly with the given context.

RAG Specific

- Context Precision
- Context Recall
- Context Entities Recall
- Noise Sensitivity
- Response Relevancy
- Faithfulness
- Multimodal Faithfulness
- Multimodal Relevance

Natural Language Comparison

- Factual Correctness
- Semantic Similarity
- Non LLM String Similarity
- BLEU Score
- ROUGE Score
- String Presence
- Exact Match

ragas

General purpose

- Aspect critic
- Simple Criteria Scoring
- Rubrics based scoring
- Instance specific rubrics scoring

You may have to roll your own evaluation, with your own **data** and **queries** for the particular use case. A public IR dataset may not work. But metrics libraries can help!

x

Survey papers on RAG

- Gao et al., Retrieval-Augmented Generation for Large Language Models: A Survey https://arxiv.org/pdf/2312.10997
- Zhao et al., Retrieval-Augmented Generation for AI-Generated Content: A Survey https://arxiv.org/pdf/2402.19473
- Zhao et al., Retrieving Multimodal Information for Augmented Generation: A Survey https://aclanthology.org/2023.findings-emnlp.314v2.pdf
- Fan et al., A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models https://doi.org/10.1145/3637528.3671470
- Kenthapadi et al., Grounding and Evaluation for Large Language Models: Practical Challenges and Lessons Learned (Survey), KDD 2024. https://dl.acm.org/doi/10.1145/3637528.3671467