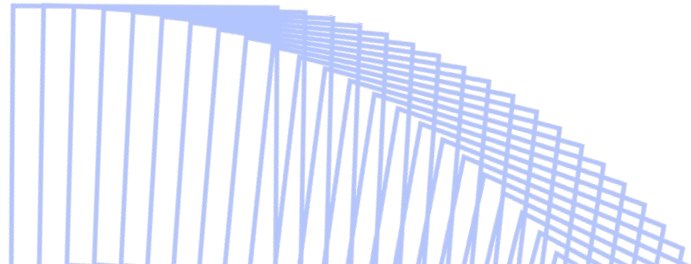
Haystack
by deepset

Advanced RAG



Hello 👋

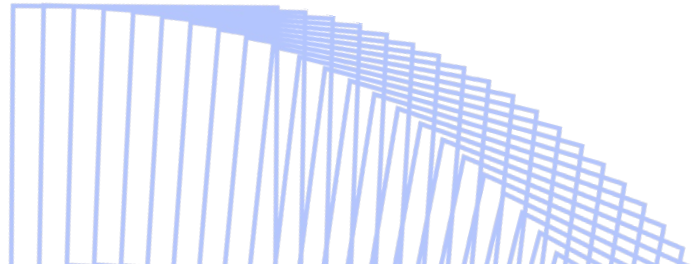
Bilge Yucel

- 🥑 Developer Relations Engineer at deepset 🇩🇪
- 🏗️ Open source LLM Framework: Haystack
- 🎓 Sabanci University B.Sc.
- 🧠 KU Leuven M.Sc.
- 📍 Istanbul, Turkey

@bilgeyucl **in/bilge-yucel**

bilgeyucel

Agenda

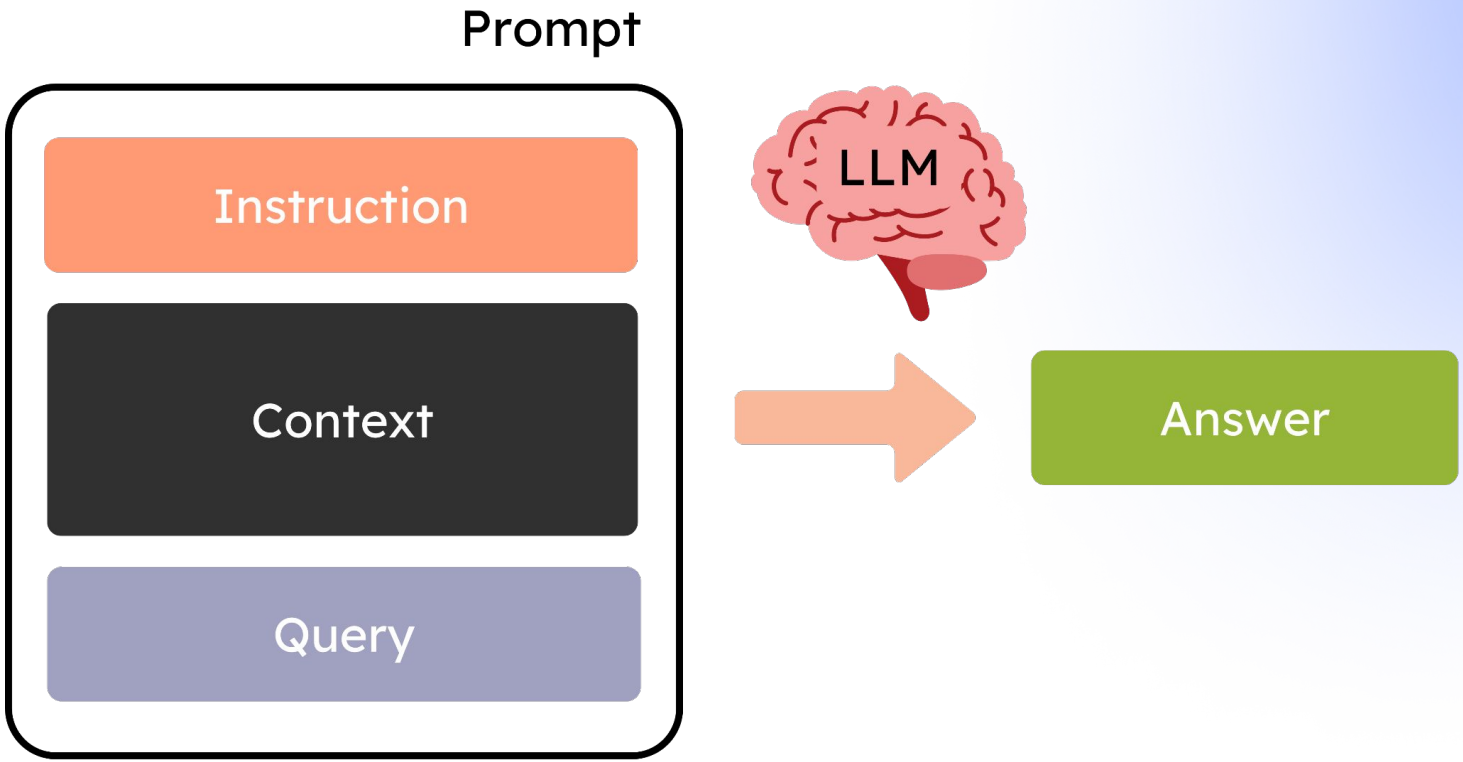


- LLM
- Retrieval Augmented Generation
- Advanced RAG with Examples

Large Language Models

- Fixed knowledge cutoff
- No access to internal data
- Hallucinations
- ⭐ RAG ⭐

Retrieval Augmented Generation (RAG)



Retrieval Augmented Generation (RAG)

- Use LLMs generative capabilities, not their knowledge
- LLM is “augmented” with a retrieval step
- Ground the generative model’s output in real-world data, so answers stay factual and relevant

Use Cases:

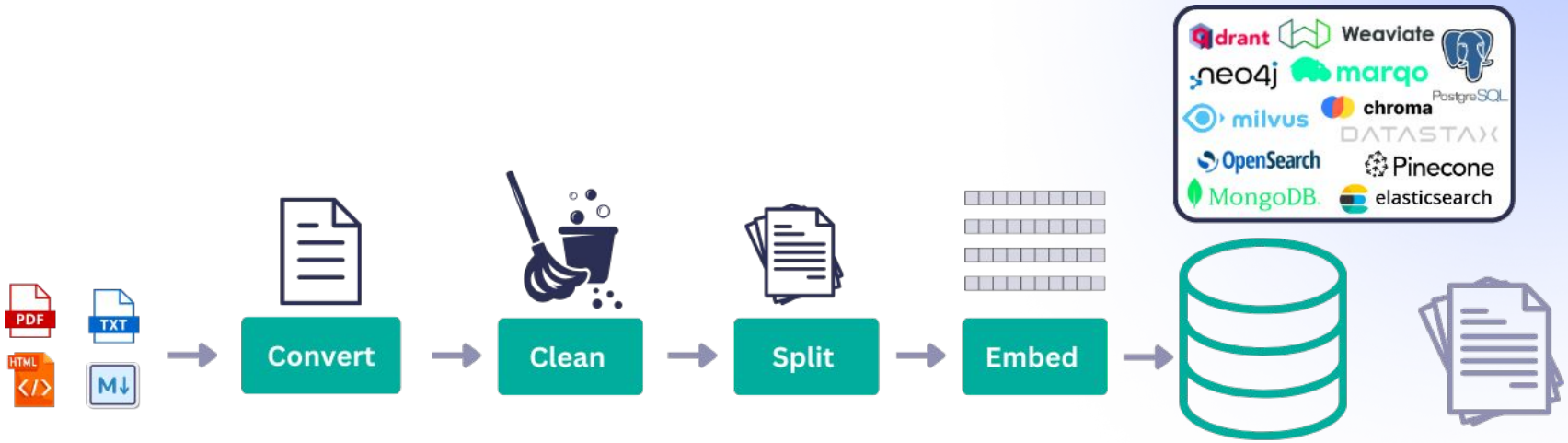
Customer Support FAQs,
Enterprise Knowledge Bases
Search Engines & QA

Benefits of RAG

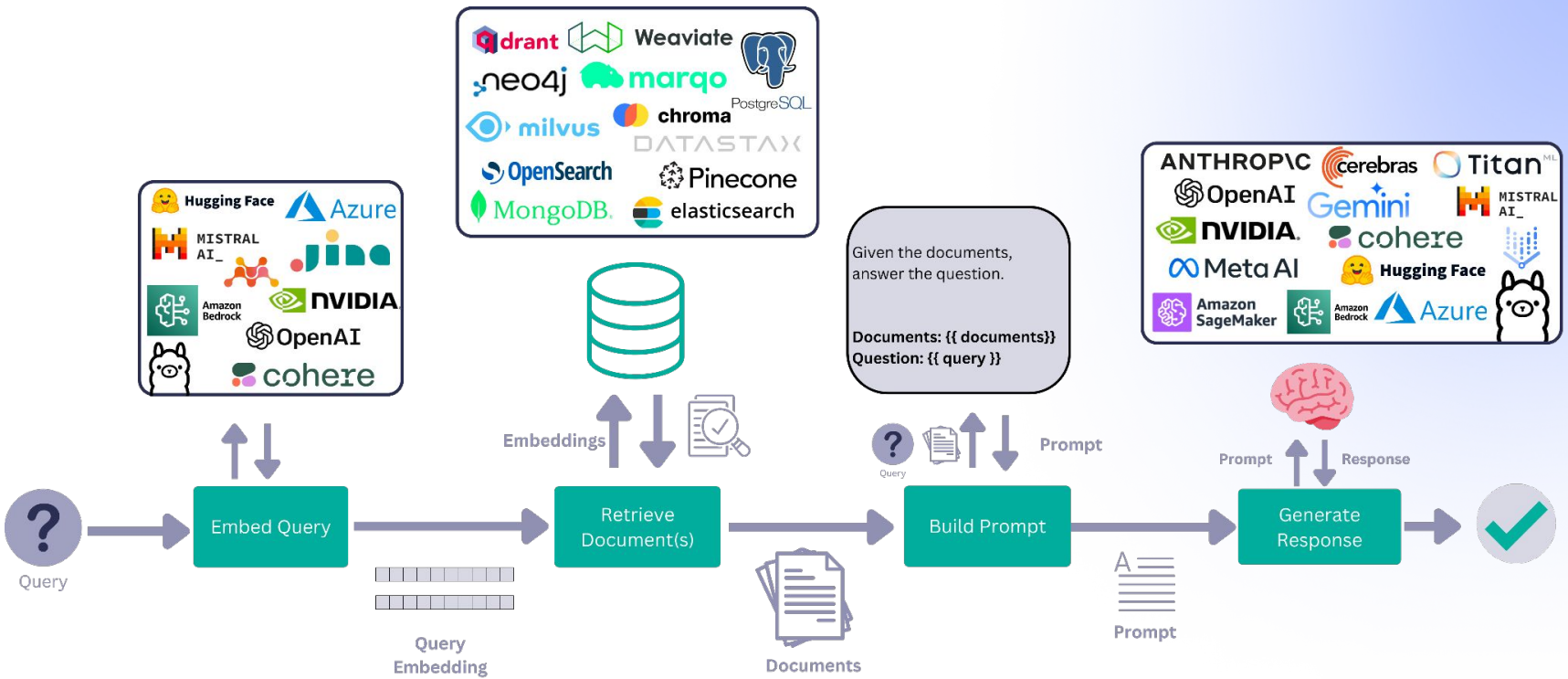
- Up-to-date information
- Private knowledge
- Reduced hallucination
- Cost-effective
- Transparency

Indexing for RAG

- Split by words, sentences, paragraphs → chunk
- Create embeddings for each chunk



Querying for RAG



Limitations of Standard RAG

- One type retrieval
- One-shot retrieval
- No feedback loop

Go Beyond Standard RAG

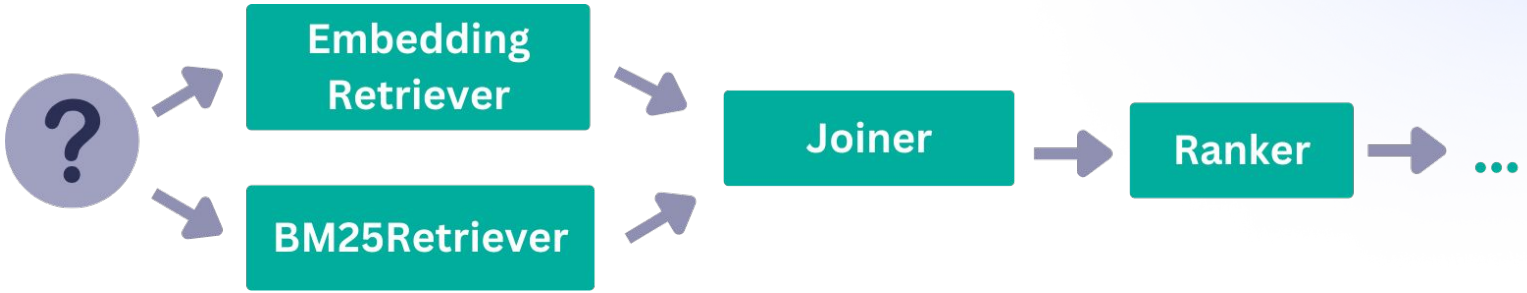
- Complex user queries often need multiple retrievals or deeper reasoning
- Sometimes, initial retrieval doesn’t provide enough context
- Need for dynamic and iterative search strategies to enhance accuracy

Retrieval Refinement

- Hybrid retrieval techniques (keyword + vector search) + Ranking
- Metadata filtering + Metadata extraction
- Query decomposition + Query expansion
- Multi-step retrieval
- Loops

Hybrid Retrieval/Search

- **Vector search** is powerful but sometimes retrieves less precise results (domain specific)
- **Keyword search** provides precision but lacks semantic understanding
- Keyword + vector = hybrid
- **Ranking** → Relevance, Lost in the middle (Source)



Metadata

- Extra info for documents like date, language, location, type...
- Document → content, embedding (dense), metadata

Metadata Filtering

- Narrow down the search space in pre-retrieval
- For user management

```
# Retrieval
{
  "query": "Why did the revenue increase?",
}

# Retrieval + Metadata Filtering
{
  "query": "Why did the revenue increase?",
  "filters": {"operator": "AND",
              "conditions": [
                    {"field": "meta.years", "operator": "==", "value": "2019"},
                    {"field": "meta.companies", "operator": "in", "value": ["BMW", "Mercedes"]}
              ]
             }
}
```

Extract Metadata Filters from a Query

- Get metadata filters from the query with an LLM

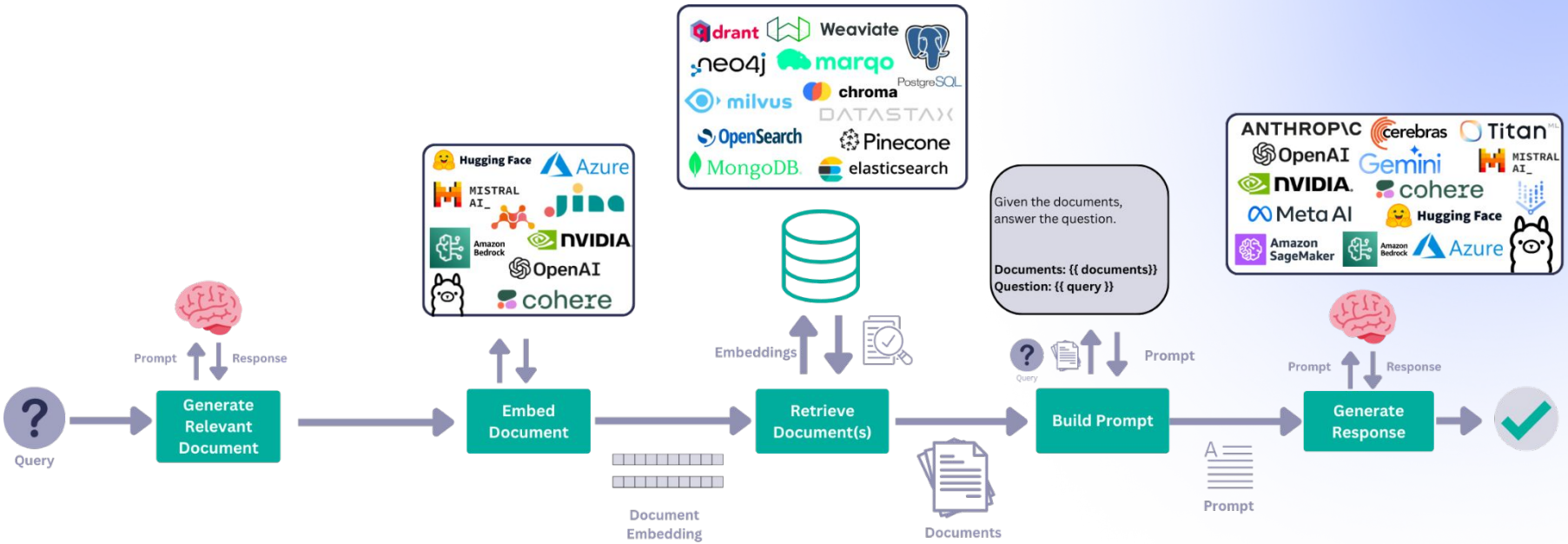
```
query = "What were the most influential publications in 2022 regarding Parkinson's disease?"
metadata_fields = {"disease", "year"}

# Metadata Filter
{'filters': {'operator': 'AND',
  'conditions': [
    {'field': 'meta.disease', 'operator': '==', 'value': 'Alzheimers'},
    {'field': 'meta.year', 'operator': '==', 'value': 2023}
  ]}
}
```

HyDE - Hypothetical Document Embeddings

- Queries are short compared to documents
- Use generated document(s) to retrieve data

HyDE - Hypothetical Document Embeddings



Query Rewriting

- Create alternatives of the query
  - “Green energy sources” → “renewable energy sources”, “sustainable energy options”...
- Add more context
  - “open source NLP frameworks” → “open-source natural language processing platforms”

Query Decomposition

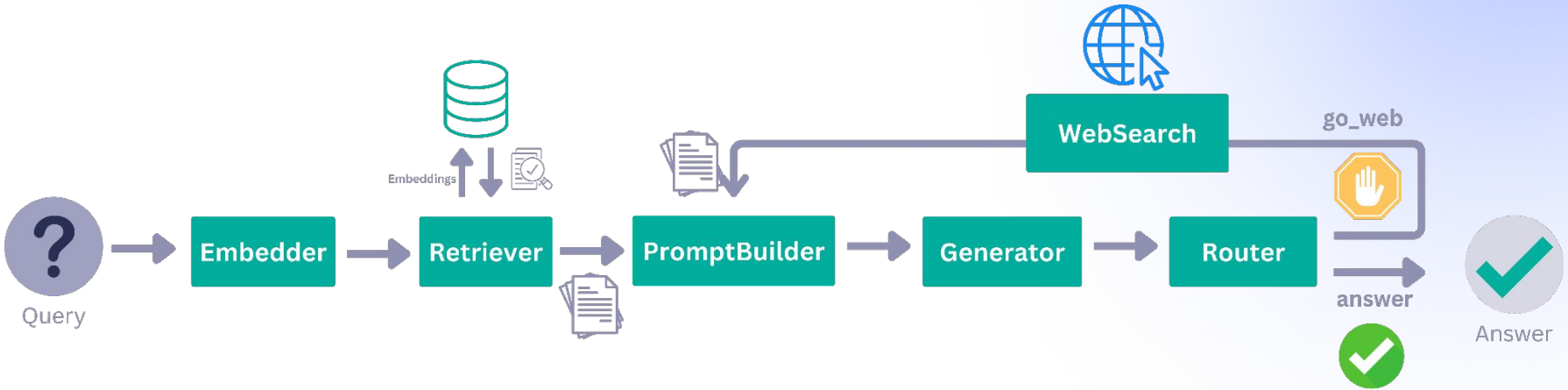
- Split the query into smaller sub-queries
- “Which model is better for reasoning, o1 or DeepSeek-R1?”
- “o1 reasoning capabilities”, “DeepSeek-R1 reasoning capabilities”
- Requires multi-step/multi-hop retrieval

Agentic RAG

- Deterministic → non-deterministic
- LLM as the brain and decides on the next action
- Can go to alternative resources: Web, another database
- Update the retrieval: query rewriting

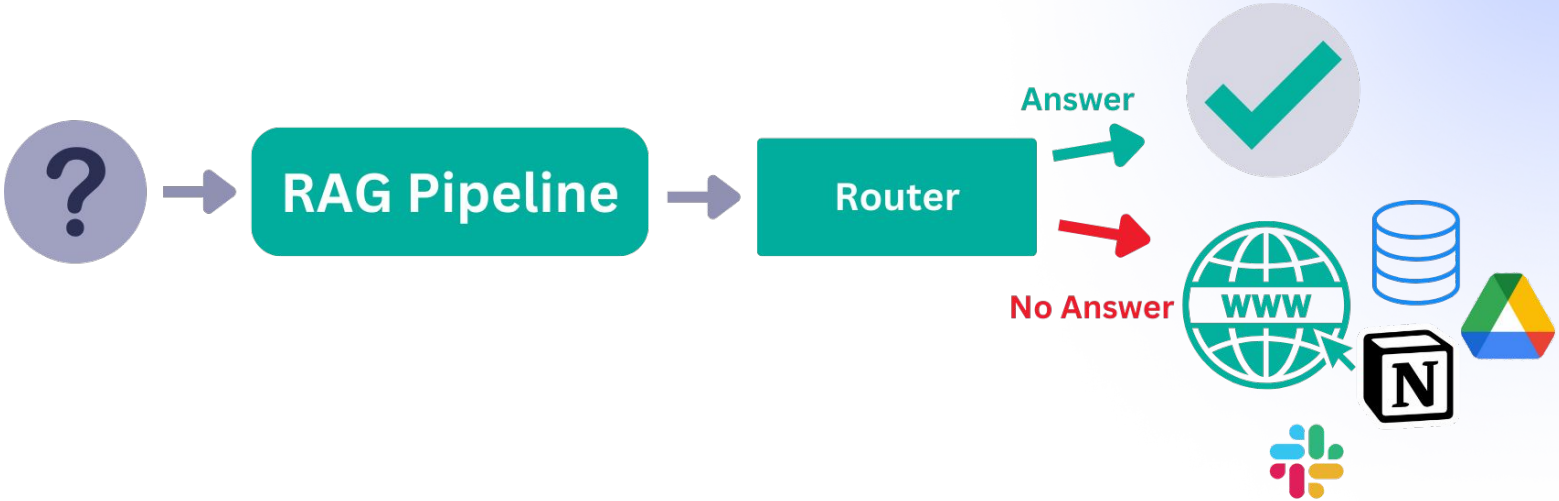
Self Reflection in Advanced RAG

- Go to alternative resources
- Looping incorporated



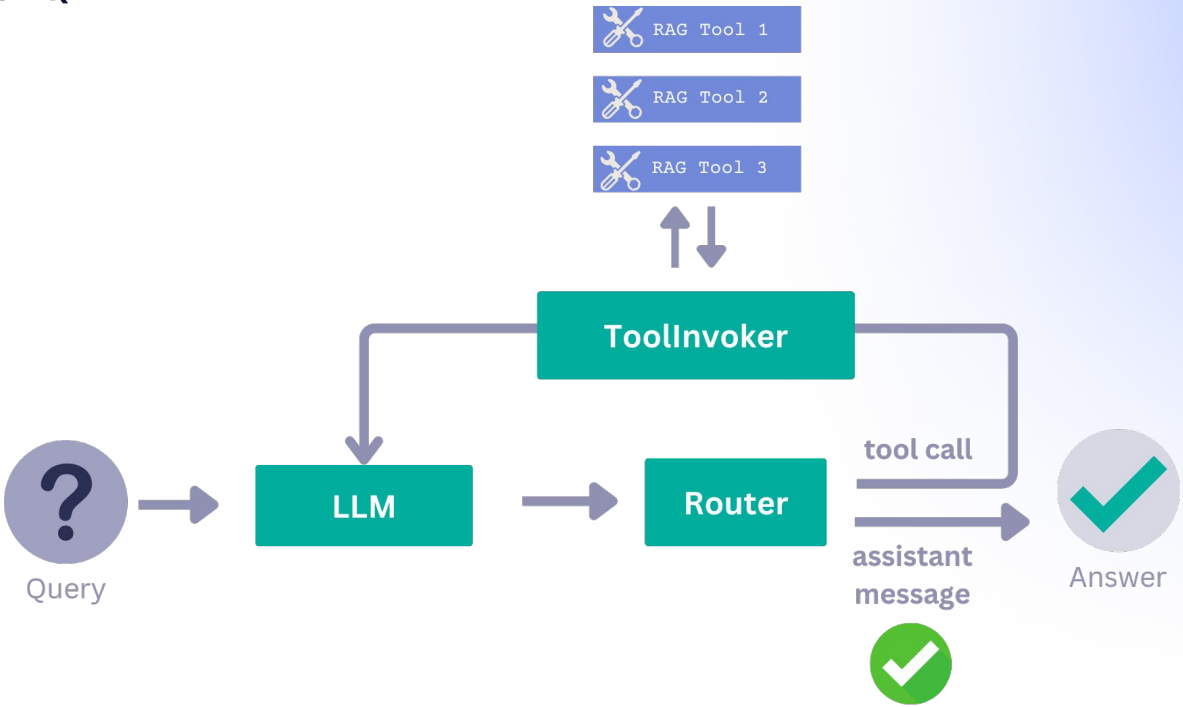
Fallback Mechanism in Advanced RAG

- Go to alternative resources to perform different actions
- No looping



Tool Calling RAG Agent

- Multi-hop QA



RAG Evaluation

	Retrieval Evaluation	Generation Evaluation	End-to-end Evaluation
Labeled data	DocumentMAPEvaluator, DocumentMRREvaluator, DocumentRecallEvaluator, DocumentNDCGEvaluator	-	AnswerExactMatchEvaluator, SASEvaluator
Unlabeled data (LLM-based)	ContextRelevanceEvaluator	FaithfulnessEvaluator	LLMEvaluator**

Evaluation Metrics

Answer Exact Match - ground-truth answers + predicted answers

Semantic Answer Similarity - ground-truth answers + predicted answers

Document Mean Average Precision (MAP) - ground-truth docs + retrieved docs

Document Recall (Multi hit, single hit) - ground-truth docs + retrieved docs

Document Mean Reciprocal Rank (MRR) - ground-truth docs + retrieved docs

Document Normalized Discounted Cumulative Gain (NDCG) - ground-truth docs + retrieved docs

Faithfulness - question + predicted docs + predicted answer

Context Relevance - question + predicted docs

LLM-based custom metrics

Ragas + FlowJudge + DeepEval

Summary

- Basic RAG is not enough to cover real life scenarios
- Retrieval is important for accurate RAG systems
- Enhance retrieval with some advanced techniques
- Incorporate agentic behavior if you need

Thank You! Any Questions?

Bilge Yucel

in/bilge-yucel @bilgeyucl bilgeyucel