Workshop: RAG to Basic with One MongoDB

Piti Champeethong (Fyi/ฟี่)

PyLanna Co-founder (Python User Group)
MongoDB Thailand User Group Buddy
https://github.com/ninefyi

Agenda

- Evolution of the recommender system
- Introduction to Hybrid Search.
- Introduction to the Fusion algorithm
- Proposed RAG pipeline
- Application Demo
- Key takeaway
- Q&A

Prerequisites

- MongoDB Atlas (https://cloud.mongodb.com/)
- Voyage AI (https://www.voyageai.com/)
- Python Environment (https://docs.python.org/3/library/venv.html)

Evolution of the recommender system

The Early Days (1990s): Collaborative & Content-Based Filtering

- "People like you liked this!" (User-to-User)
- "If you liked this movie, here are similar ones." (Item-to-Item)
- Simple but limited by sparse data.

Evolution of the recommender system

The Matrix Era (2000s): Matrix Factorization & SVD

- Uncovering hidden patterns ("latent factors") in ratings.
- Powerful for predicting unknown preferences.

Evolution of the recommender system

The Deep Learning Shift (2010s): Neural Networks

- Complex pattern recognition, personalized sequences.

- Handling massive datasets and subtle signals.

Evolution of the recommender system

The Generative Turn (2020s+): Retrieval Augmented Generation (RAG)

- Moving beyond just recommending items to generating contextual, informed responses.

- The focus shifts to understanding and answering questions based on your data.

Agenda

- Evolution of the recommender system
- Introduction to Hybrid Search.
- Introduction to the Fusion algorithm
- Proposed RAG pipeline
- Application Demo
- Key takeaway
- Q&A

Introduction to Hybrid Search.

The Problem with Single Search Types:

- Keyword Search (BM25/Full-Text Search):
- Pros: Excellent for exact matches, specific terms, IDs, proper nouns.
- Cons: Fails at semantic understanding ("movies about space that feel lonely" vs. "Star Wars"). Doesn't handle synonyms or nuanced meaning well.

Introduction to Hybrid Search.

Vector Search (kNN/ANN/ENN/Semantic Search):

- Pros: Brilliant at understanding meaning, context, and similarity based on embeddings. Finds "lonely space movies" even if the words aren't exact.
- Cons: Can miss highly specific keywords or exact IDs if their vector representation isn't perfectly aligned. Might return semantically similar but factually incorrect results.

Introduction to Hybrid Search.

The Hybrid Solution:

- Combining them: Run both a keyword search and a vector search simultaneously.
- The Goal: Leverage the precision of keywords and the recall/semantic understanding of vectors.
- Result: A more robust, comprehensive, and accurate search experience that truly understands user intent.

Introduction to Hybrid Search.

- MongoDB Atlas Advantage:

- Native integration of both Full-Text Search and Vector Search within a single database.
- Simplifies architecture, reduces latency, and keeps your data unified.

Introduction to the Fusion algorithm

The Challenge:

- You have two lists of results: one from keyword search, one from vector search.
- How do you combine them when their scoring mechanisms are completely different?
- A keyword score of 'X' means something different than a vector distance of 'Y'.

Reciprocal Rank Fusion (RRF)

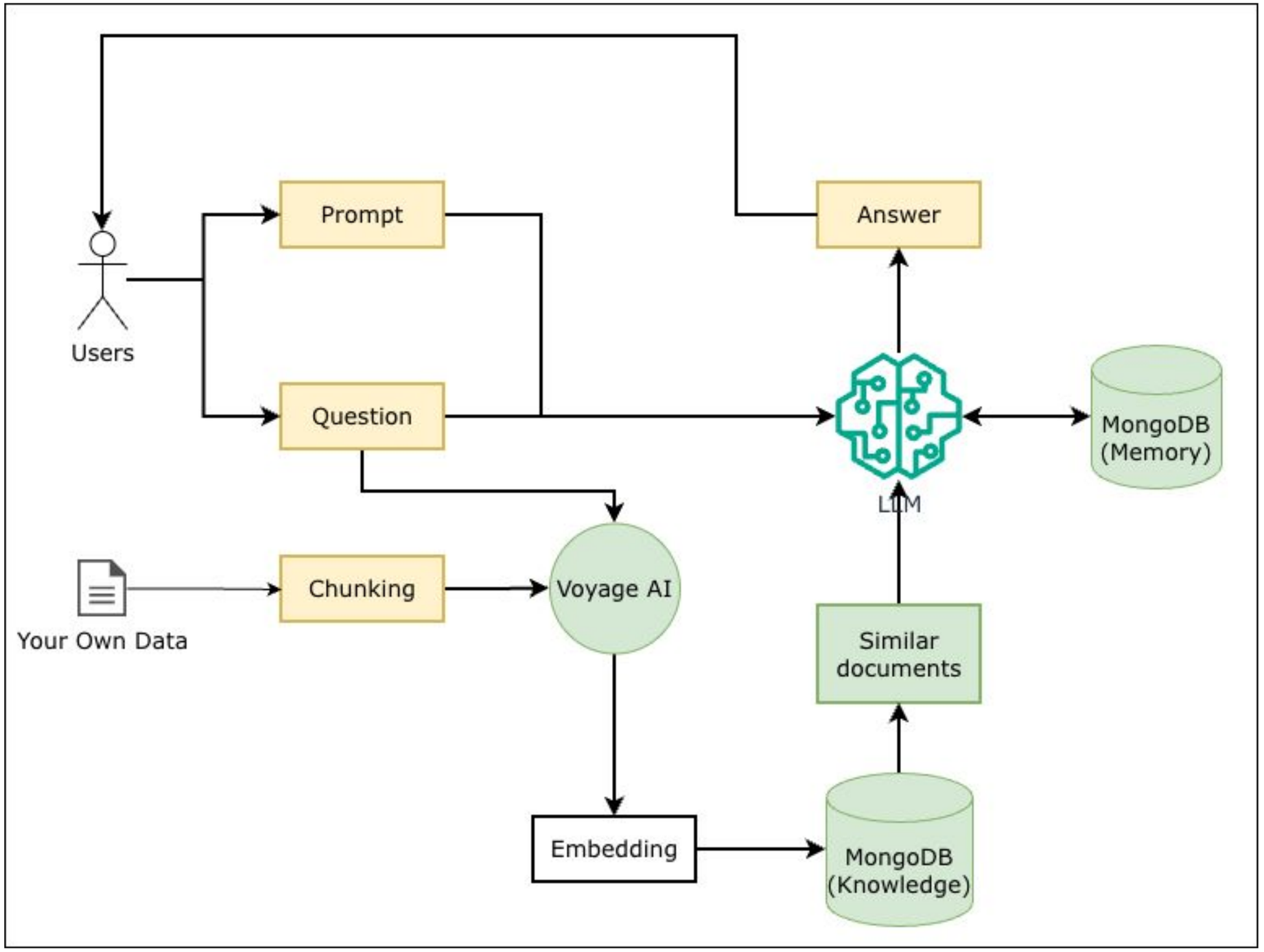
Reciprocal Rank Fusion (RRF)

- If a document appears high on both the keyword and vector search lists, it gets a very high RRF score.
- If a document appears high on one list but low or not at all on the other, it still gets a good score, but less so.
- This prioritizes documents that are strongly relevant to both exact terms and semantic meaning.

Reciprocal Rank Fusion (RRF)

$$score(d) = \sum_{r \in R} \frac{1}{k + r(d)}$$

- score(d): The final RRF score for document 'd'.
- R: The set of all ranked lists (e.g., keyword search, vector search).
- r(d): The rank of document 'd' in a specific list.
- k: A constant (typically 60) to prevent division by zero for rank 0 and to control the impact of lower ranks.


Prompt
Answer
Users
Question
MongoDB
(Memory)
LLM
Chunking
Voyage AI
Your Own Data
Similar
documents
Embedding
MongoDB
(Knowledge)

Key takeaway

- Consolidation is King:
- Hybrid Search (Keyword + Vector) combined with Reciprocal Rank Fusion delivers superior, more accurate, and contextually rich results than either search type alone.
- Building RAG applications doesn't have to be complex. MongoDB Atlas provides the robust, scalable, and intuitive foundation you need to focus on innovation, not infrastructure.

References

- https://github.com/ninefyi/tech-on-the-rock-2025
- https://www.mongodb.com/resources/basics/ann-search
- https://docs.voyageai.com/docs/embeddings
- https://www.mongodb.com/resources/basics/chunking-explained

THANK YOU!