WELCOME

Search and Retrieval: AI’s Most Successful Hack

Apoorva Joshi
Senior AI/ML Developer Advocate @MongoDB

whoami

```
{
  "name": "Apoorva Joshi",
  "MongoDB": {
    "positions": ["AI Developer Advocate"],
    "since": new Date("2024-01")
  },
  "Pre-MongoDB": {
    "positions": ["Security Data Scientist"],
    "companies": ["FireEye", "Elastic"]
  },
  "likes": ["reading", "yoga", "coffee shop hopping"]
}
```

Agenda

Search and Retrieval in AI

RAG = RecSys for LLMs

Building a 4-stage RAG system

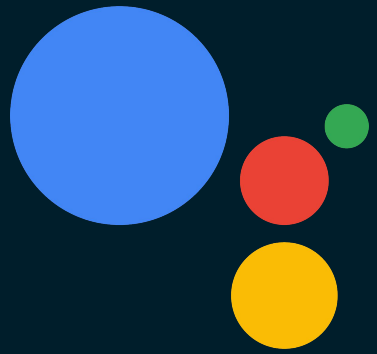
Conclusions

Q & A

Search and Retrieval in AI

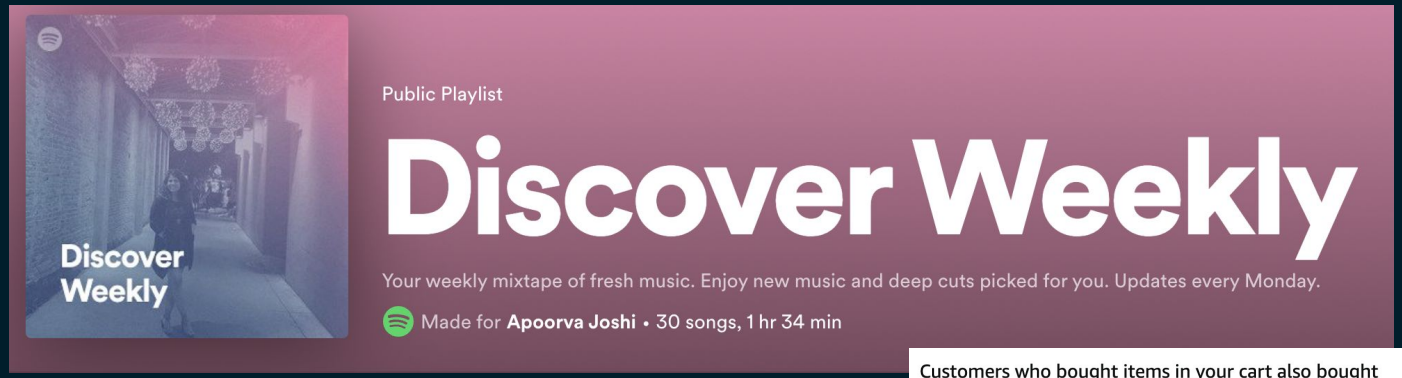
How it started…

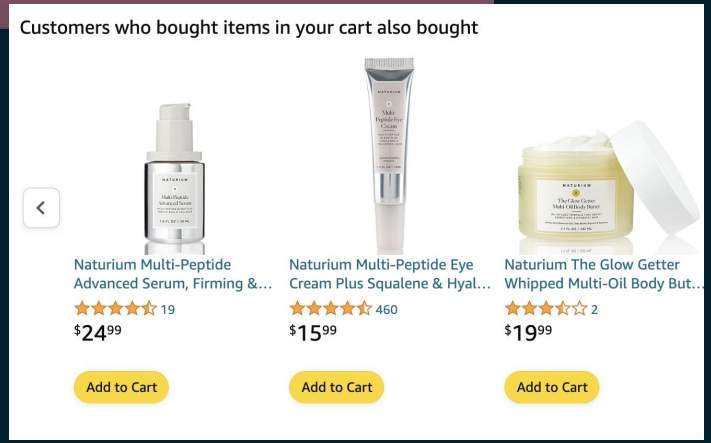
Virtual Assistants



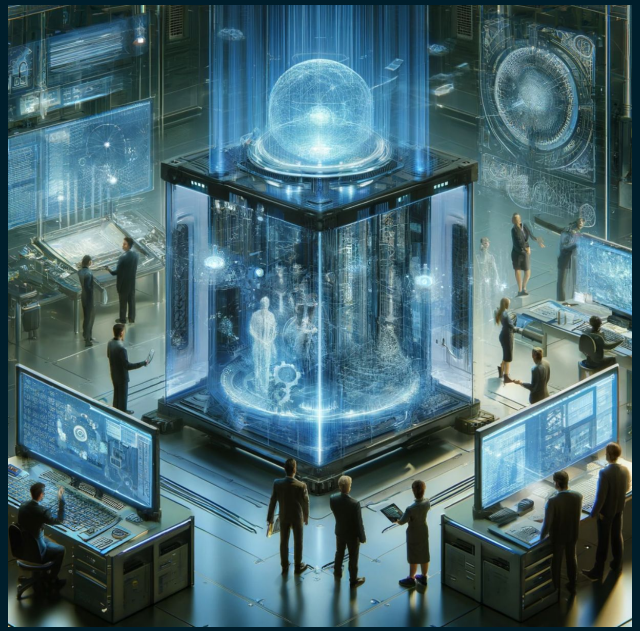
alexa

Recommender Systems



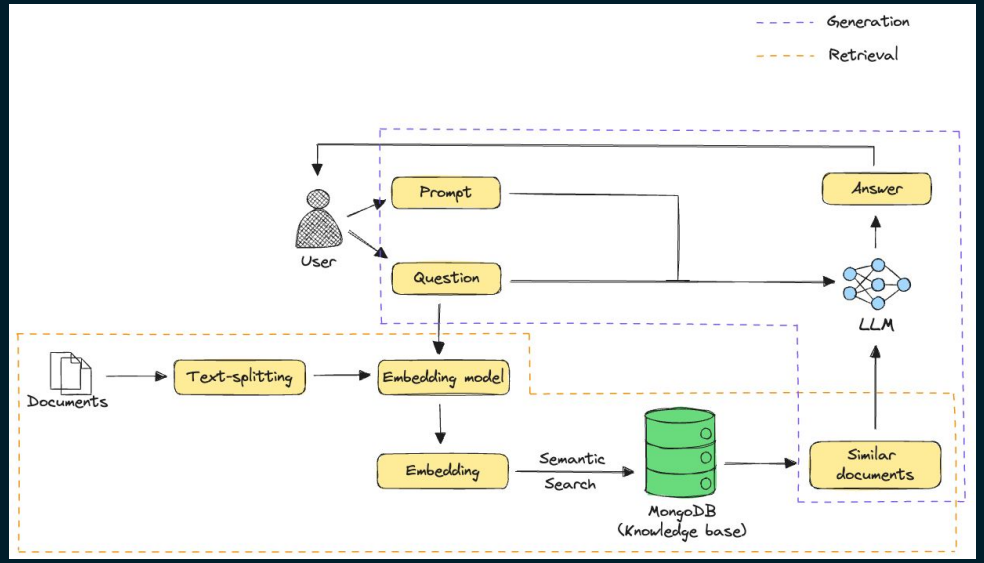


How it’s going…



RAG = RecSys for LLMs

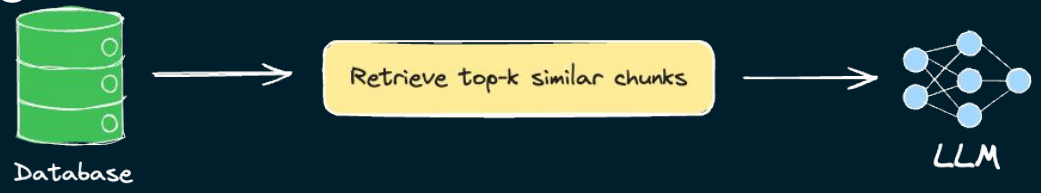
RAG — A quick refresher



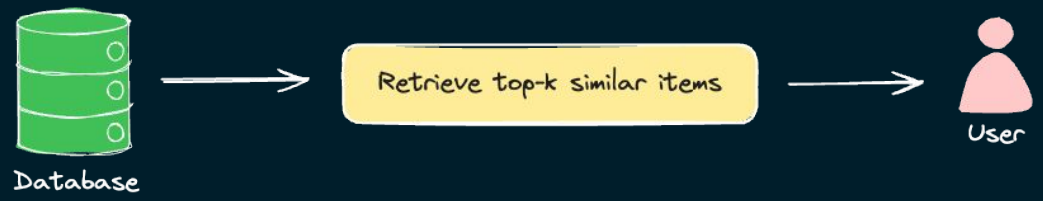
- Retrieve relevant documents from a knowledge base
  - Chunk up large documents
  - Embed chunks, store them in a database
  - Given a user query, embed it and retrieve semantically relevant chunks
- Pass all of this as context to an LLM to get an answer

Retrieving relevant chunks from the knowledge base

Is this enough?

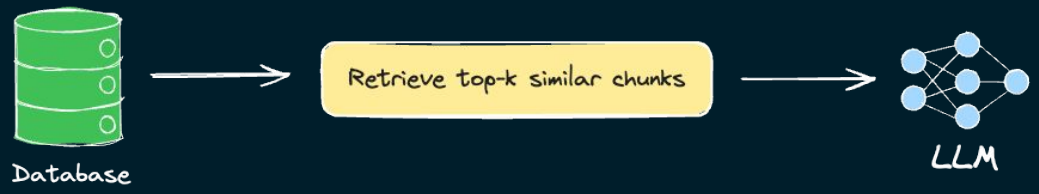


Think Recommender Systems



- ✅ Reduce the search space
- ❌ Drop out of stock items, already bought items
- ❌ Predict likelihood of the user purchasing the item
- ❌ Ordering the items by business logic

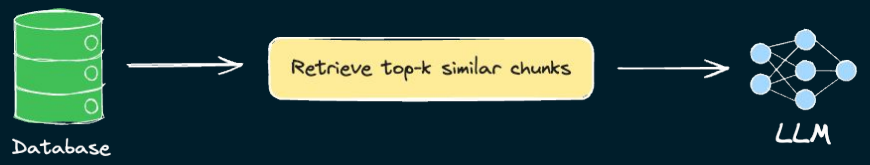
Same applies to RAG



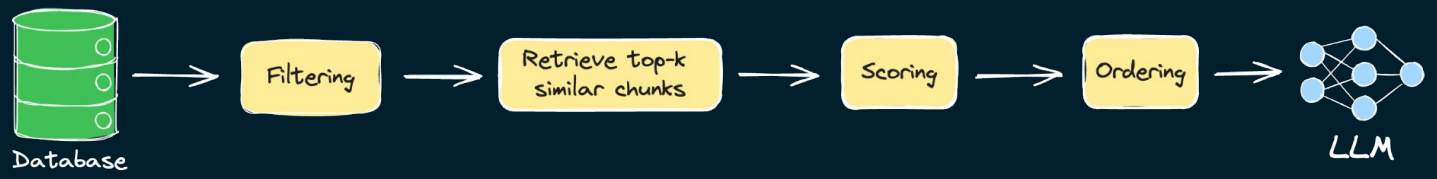
- ✅ Reduce the search space
- ❌ Filter by recency, topic, geographical location
- ❌ Incorporate user feedback
- ✅ Most relevant chunks at the top

Retrieving relevant chunks from the knowledge base

Naive approach



Improved approach



Building a 4-stage RAG system

Retrieving top-k similar chunks

What Search Methodology do I use? 🤔

Lexical Search

What?

- TF-IDF/BM25(f)
- Keyword Search

When?

- Your text corpus closely matches how users search
- Most traditional text search use cases.
- First pass at text-based relevancy.

Vector Search

What?

- ANN/kNN
- Contextual Similarities

When?

- Ambiguous user input
- ‘Vocabulary gap’ between corpus and how users search
- Text, image, audio, video search.

Query patterns for RAG can vary a lot

```
db.menus.aggregate([
  {
    $search: {
      text: {
        query: "pasta",
        path:"item"
      }
    }
  }
]);
```

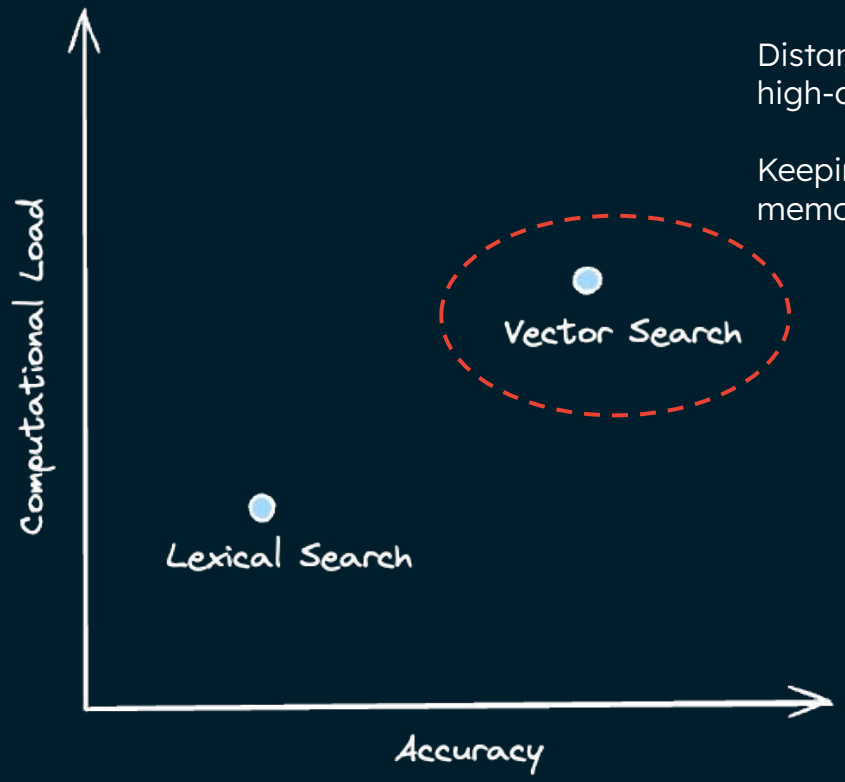
“Give me a recipe for Apple pie”

“Apple pie”

VECTOR SEARCH, DUH

(For now)

But…



Distance calculations on high-dimensional vectors

Keeping high-dimensional vectors in memory for faster retrieval

Making vector search more efficient

- Choosing the right embedding model
- Lower-precision vector representations

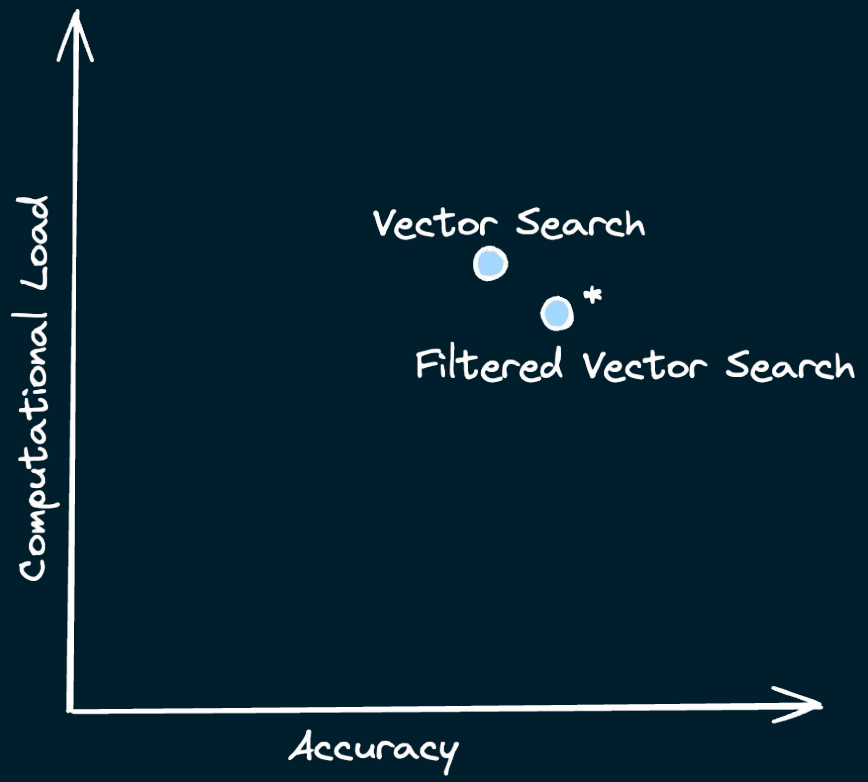
Capturing the complexity of data

vs

Operational efficiency

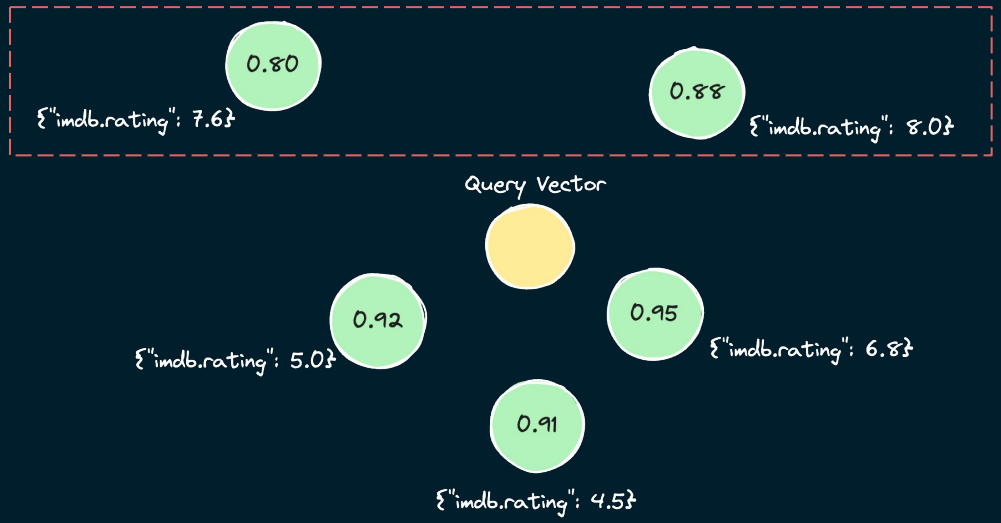
Filtering

Result of filtering



Pre-filtering vs post-filtering

filter : imdb.rating > 7.0, **k** : 2



Pre-filtering

Post-filtering

- No impact on vector search latency
- No results returned

Extracting metadata filters

"Give me movies with an IMDB rating greater than 7."

```
{"imdb.rating": {"gte": 7.0}}
```

Metadata filter

```
db.collection.aggregate([
  {
    "$vectorSearch": {
      "index": "vector_index",
      "path": "plot_embedding",
      "queryVector": [0.45, 0.63, 0.78,...],
      "limit": 10,
      "numCandidates": 100,
      "filter": {
        "imdb.rating": {"$gte": 7.0}
      }
    }
  },
  {
    "$project": {"plot_embedding": 0}
  }
])
```

Metadata filtering + Vector Search

Types of metadata filters

- Filtering by business logic
- Access control (filter by user id)

Easy

- Query dependent

Using LLMs

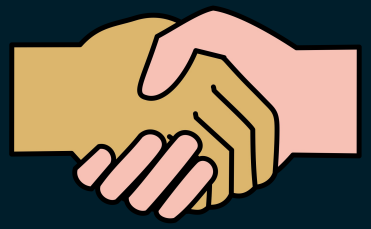
- Need metadata schema
- Need instructions to construct the filter in the query language of choice
- Not reliable

Scoring

Optimizing for important keywords

Lexical Search

Significance of keywords



Vector Search

Contextual similarity

Hybrid Search

Optimizing for important keywords

```
{
  "id": ObjectID("662043cfb084403cdcf5210a"),
  "title": "The Perils of Pauline",
  "plot_embedding": [0.43, 0.57, ...],
  "full_plot": "Young Pauline is left a lot of money when her wealthy
    uncle dies. However, her uncle's secretary has been named as her
    guardian until she marries, at which time she will officially take
    possession of her inheritance. Meanwhile, her \"guardian\" and his
    confederates constantly come up with schemes to get rid of Pauline
    so that he can get his hands on the money himself."
}
```

Define **vector search index** on **plot_embedding**

Define **lexical search index** on **full_plot**

Optimizing for important keywords

$vectorSearch

```
[
  {
  "id": ObjectID("662043cfb084403cdcf5210a"),
  "title": "The Great Mouse Detective",
  "plot_embedding": [0.43, 0.57, ...],
  "score": 0.95
  },
  {
  "id": ObjectID("662043cfb084403cdcf5210b"),
  "title": "The Heiress",
  "plot_embedding": [0.72, 0.63, ...],
  "score": 0.90
  },
  {
  "id": ObjectID("662043cfb084403cdcf5210b"),
  "title": "Crooked House",
  "plot_embedding": [0.12, 0.48, ...],
  "score": 0.67
  }
]
```

$search

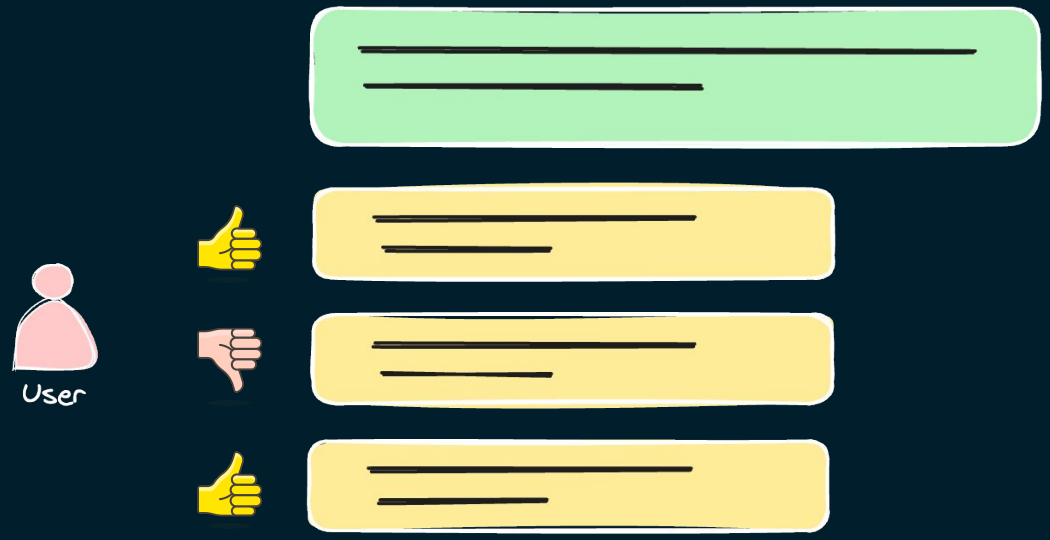
```
[
  {
  "id": ObjectID("662043cfb084403cdcf5210b"),
  "title": "The Heiress",
  "full_plot": "In 1840s New York, Catherine Sloper, a wealthy
    but plain-looking woman, falls for the charming Morris
    Townsend. Her emotionally distant father suspects Morris is
    only interested in her fortune and vows not to leave her
    anything if she marries him. The story deepens as Catherine
    must decide whether Morris's intentions are true or if her
    fortune is his true objective, leading to a poignant
    exploration of love, betrayal, and character strength",
  "plot_embedding": [0.72, 0.63, ...],
  "score": 0.91
  },
  {
  "id": ObjectID("662043cfb084403cdcf5210a"),
  "title": "The Great Mouse Detective",
  "full_plot": "Set in Victorian London, the film follows Basil
    of Baker Street, a rodent detective, as he attempts to
    thwart the evil Ratigan. When a toymaker is abducted,
    Basil's investigation leads him to Ratigan's plot against
    the Crown. Alongside his trusted companions, Basil races to
    save the Queen and prove his superiority to his nemesis,
    combining mystery and adventure in a whimsical animated
    setting.",
  "score": 0.83
  },
]
```

Optimizing for important keywords


```
[
  {
  "id": ObjectID("662043cfb084403cdcf5210b"),
  "title": "The Heiress",
  "full_plot": "In 1840s New York, Catherine Sloper, a wealthy
    but plain-looking woman, falls for the charming Morris
    Townsend. Her emotionally distant father suspects Morris is
    only interested in her fortune and vows not to leave her
    anything if she marries him. The story deepens as Catherine
    must decide whether Morris's intentions are true or if her
    fortune is his true objective, leading to a poignant
    exploration of love, betrayal, and character strength",
  "plot_embedding": [0.72, 0.63, ...],
  "score": 1.81
  },
  {
  "id": ObjectID("662043cfb084403cdcf5210a"),
  "title": "The Great Mouse Detective",
  "full_plot": "Set in Victorian London, the film follows Basil
    of Baker Street, a rodent detective, as he attempts to
    thwart the evil Ratigan. When a toymaker is abducted,
    Basil's investigation leads him to Ratigan's plot against
    the Crown. Alongside his trusted companions, Basil races to
    save the Queen and prove his superiority to his nemesis,
    combining mystery and adventure in a whimsical animated
    setting.",
  "plot_embedding": [0.43, 0.57, ...],
  "score": 1.78
  },
]
```


- $vectorSearch on embeddings
- $search for keywords in plot text
- group by title
- Combine vector and lexical search scores per group

Incorporating user feedback



Incorporating user feedback

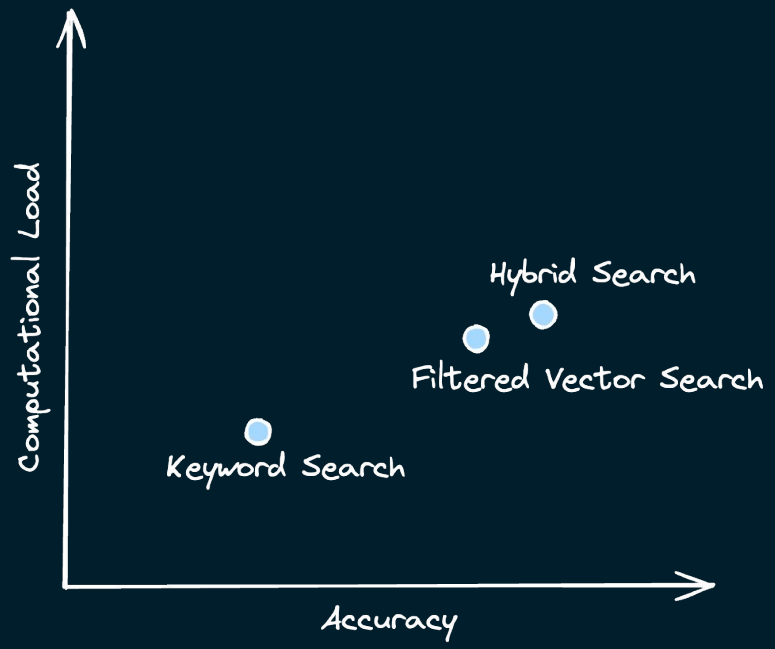

```
[
  {
  "id": ObjectID("662043cfb084403cdcf5210a"),
  "title": "The Great Mouse Detective",
  "full_plot": "Set in Victorian London, the film follows Basil of Baker
    Street, a rodent detective, as he attempts to thwart the evil Ratigan.
    When a toymaker is abducted, Basil's investigation leads him to
    Ratigan's plot against the Crown. Alongside his trusted companions,
    Basil races to save the Queen and prove his superiority to his nemesis,
    combining mystery and adventure in a whimsical animated setting.",
  "plot_embedding": [0.43, 0.57, ...],
  "upvotes": 20,
  "downvotes": 2,
  "score": 2.1
  },
  {
  "id": ObjectID("662043cfb084403cdcf5210b"),
  "title": "The Heiress",
  "full_plot": "In 1840s New York, Catherine Sloper, a wealthy but plain
    -looking woman, falls for the charming Morris Townsend. Her emotionally
    distant father suspects Morris is only interested in her fortune and
    vows not to leave her anything if she marries him. The story deepens as
    Catherine must decide whether Morris's intentions are true or if her
    fortune is his true objective, leading to a poignant exploration of
    love, betrayal, and character strength",
  "plot_embedding": [0.72, 0.63, ...],
  "upvotes": 2,
  "downvotes": 20,
  "score": 1.48
  }
]
```


hybrid_search_score

+ (upvotes * hybrid_search_score/100)

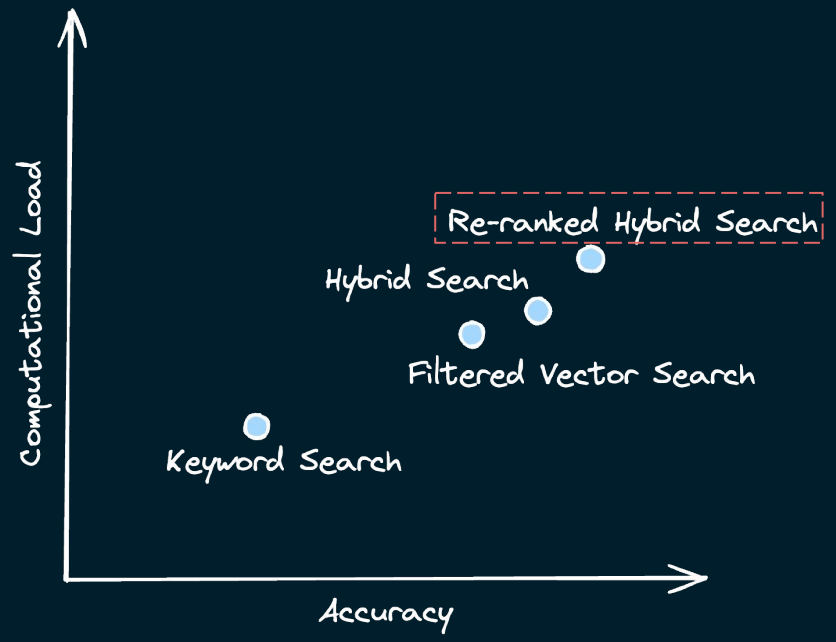
- (downvotes * hybrid_search_score/100)

Result of scoring

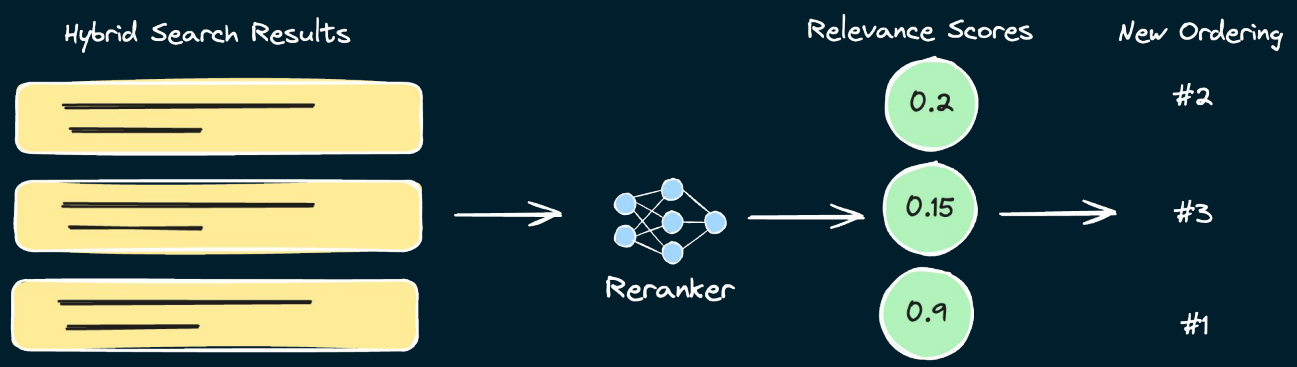


Ordering

Final push: Re-ranking



How does re-ranking work?



💡 Use smaller, specialized transformer models to score (query, document) pairs and generate a new ordering

Wrap up

Conclusions

- Retrieval for RAG goes beyond nearest neighbor search
- Choose the right embedding and data model to reduce the computational load from vector search
- Combine lexical search with vector search to get the best of both
- Follow up with re-ranking to have the most relevant results at the top

Shoutouts 🎉

- Henry Weller, Vector Search PM @ MongoDB
- Eugene Yan, Applied Scientist @ Amazon
- Jerry Liu, Co-founder/CEO @LlamaIndex

TUTORIAL

How to choose the right embedding model for RAG

More from me:
https://www.mongodb.com/developer/author/apoorva-joshi/

mdb.link/haystack

CONNECT

Apoorva Joshi
Senior AI/ML Developer Advocate @ MongoDB

Thank You!

Questions?