Weaviate

Building a Super-Doctor 🦸‍♂️ ⚕️

Advanced RAG Techniques

Zain Hasan

@zainhasan6 zainhas

The average human doctor 🧑‍⚕️

- Studies for 7+ years after undergrad
- See ~100,000 patients in a lifetime

Can we build an AI powered medical doctor/assistant?

- Have access to more data then any human doctor
- Search for relevant source patient cases and articles
- Cite previous source cases – explainability
- Reason over technical medical concepts
- Needs to be realtime (seconds to do all this!)

The average human doctor 🧑‍⚕️

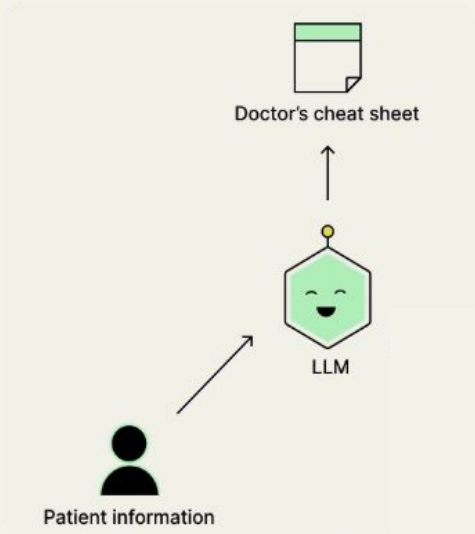
"Studies for 7+ years after undergrad"
"See ~100,000 patients in a lifetime"

Can we build an AI powered medical doctor/assisstant?

More Data | **Patient Cases** | **Explainability**

Simplest Approach

- Given patient specific information → propose a diagnosis and reasoning.
- If the LLM is fine-tuned on medical data it'll do better
- Really benefits from good prompting techniques*



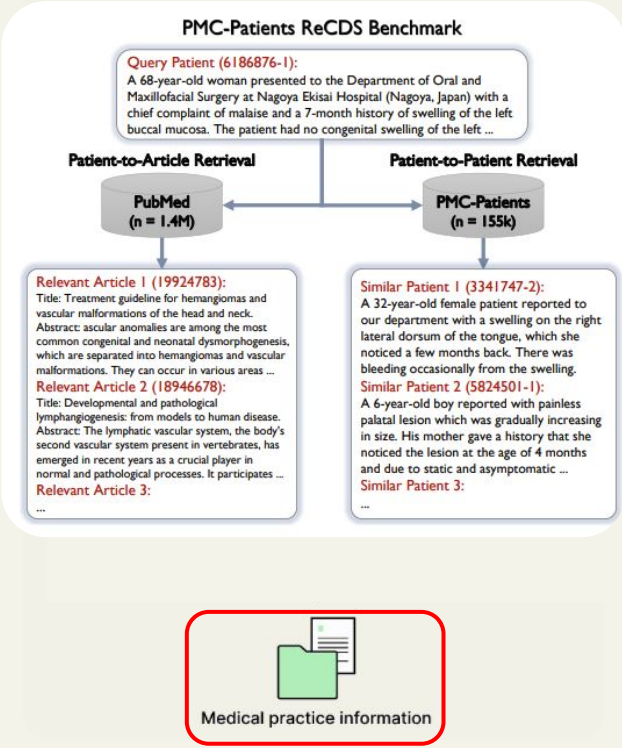
*Source: OpenMedLM - Maharjan et al. 2024

RAGdoctor Approach

- Search for similar patient cases/publications
- Pass relevant cases to LLM
- LLM performs a “retrieve then read” operation
- Allows you to cite sources for a proposed diagnosis – explainability

Patient Cases Dataset

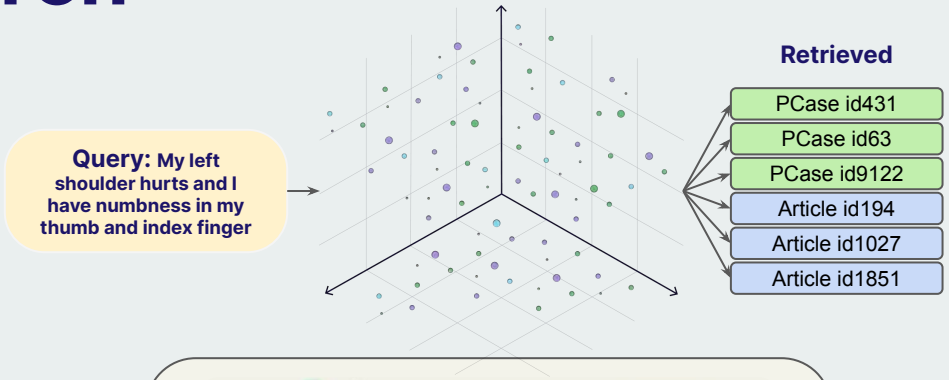
- Open PMC-Patients dataset
- 167k patient summaries extracted from case reports in PubMed Central
- 1.4M PubMed article abstracts
- ReCDS Benchmark – can be used to assess recall

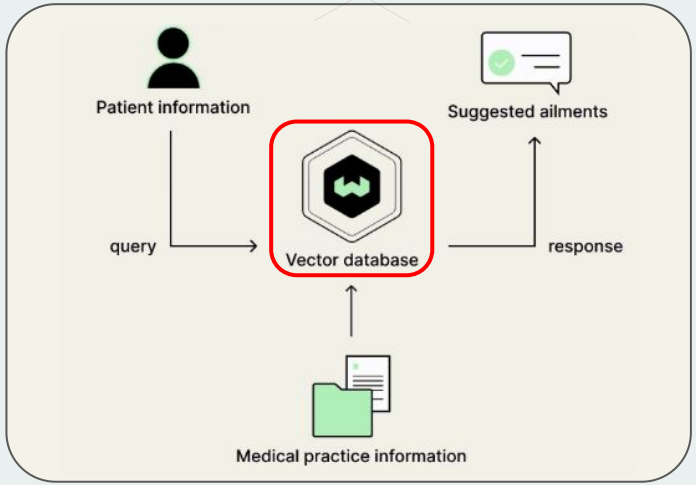


Source: PMC-Patients - Zhao et al. 2023

Vector Database – Search

- Allows you to store billions of patient cases and medical articles
- Given a query, can perform real time ***similarity search*** to get top similar articles and cases
- Open source vector DB – [Weaviate]

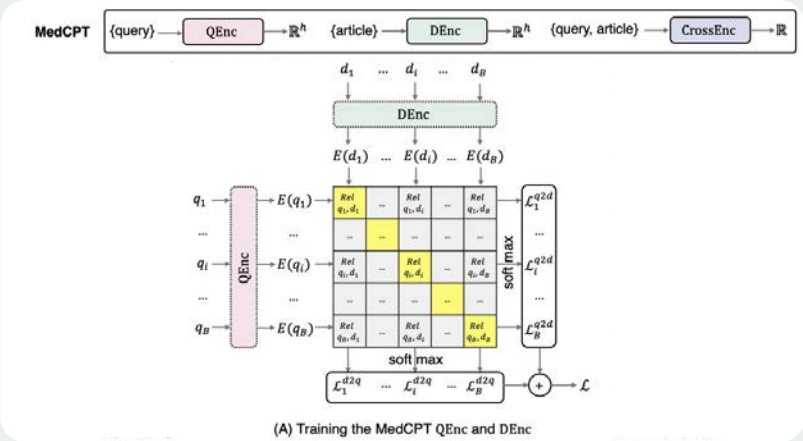




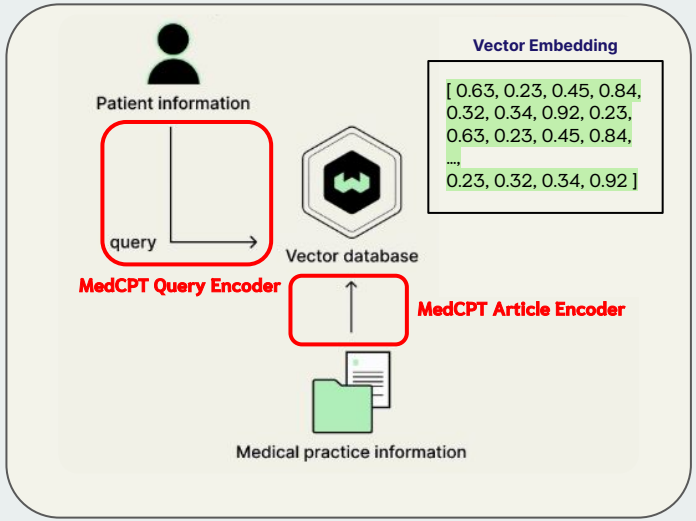
Link: [Open Source Vector Database]→[Docs]

Bio Embedding Models

- Need to represent patient cases/articles as vectors
- Need a medical domain embedding model
- MedCPT Query Encoder: compute the embeddings of short texts
- MedCPT Article Encoder: compute the embeddings of patient cases & articles



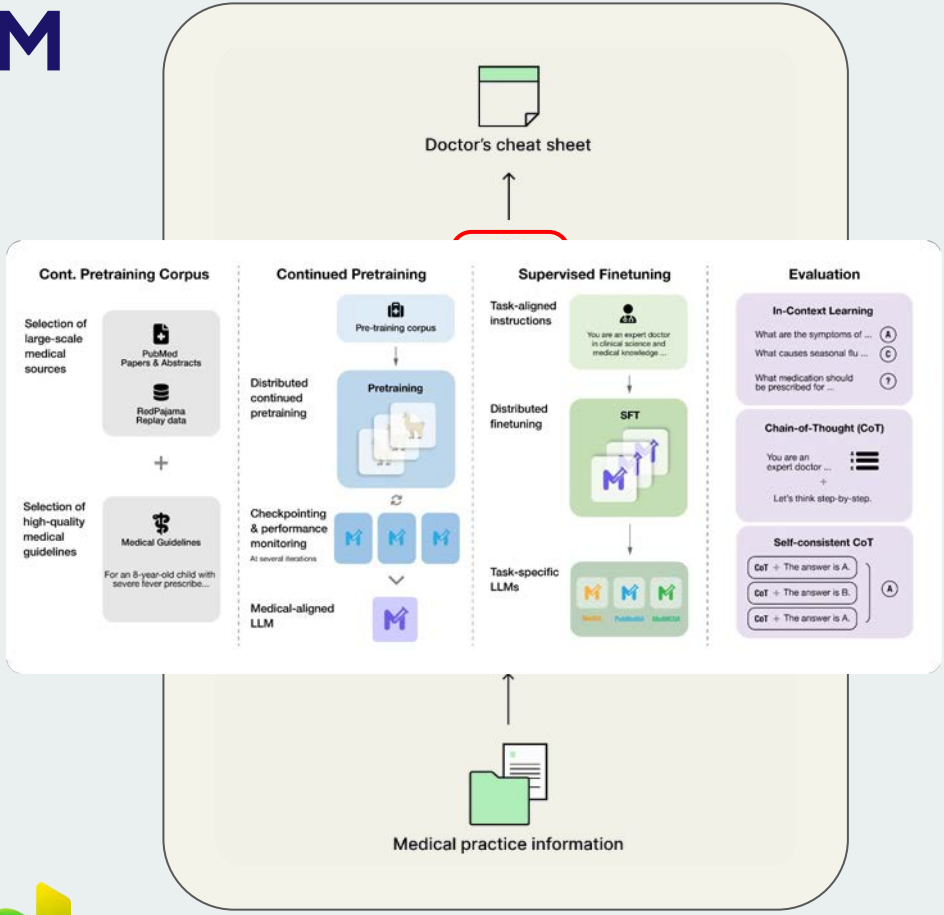
(A) Training the MedCPT QEnc and DEnc



Source: MedCPT - Jin et al. 2023 - Code

Biomedical Domain LLM

- If you use a LLM fine-tuned on medical domain data it can perform better
- [Meditron-70B](#) open-source medical LLMs
- Trained on 48.1B tokens from the medical domain
- Outperforms Llama2-70B, GPT-3.5 medical reasoning tasks.



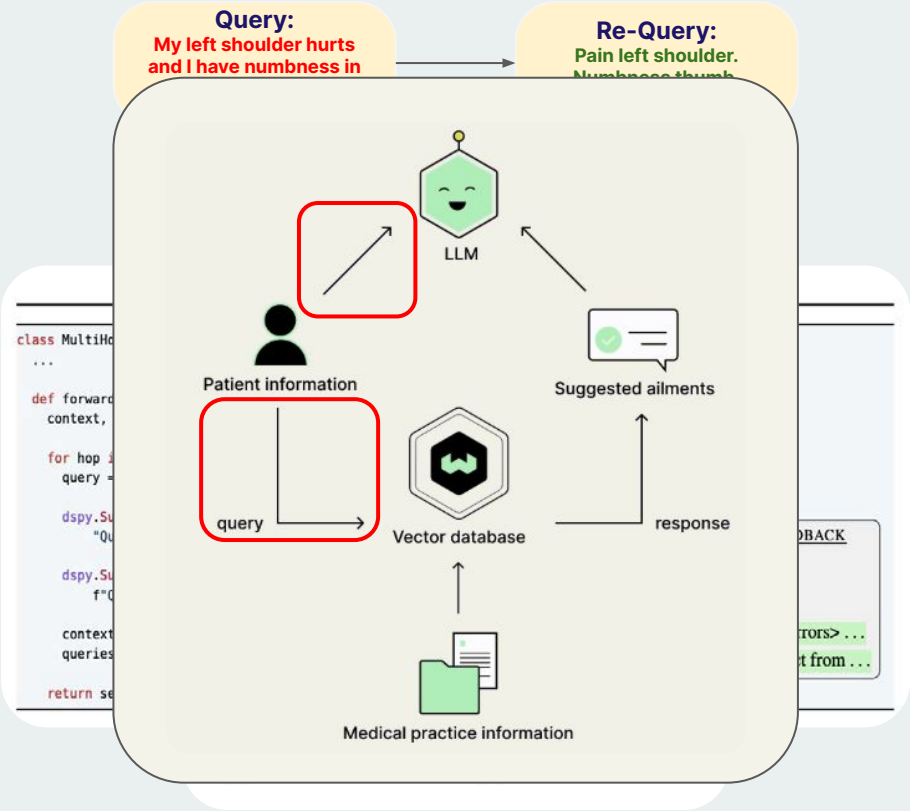
Source: [Meditron 70B - Chen et al. 2023](#)

Can we do <u>even better</u>?

Let’s introduce some advanced RAG techniques into the pipeline!

Query Rewriting

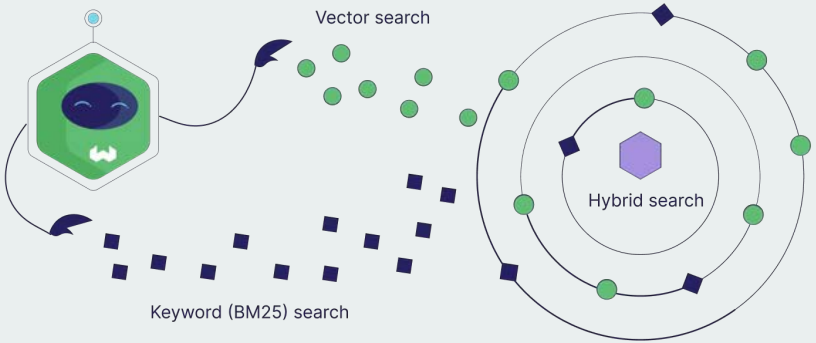
- **Idea:** We don't know how to write the best query!
- Get a LLM to ***re-write both queries*** to vector DB and LLM
- We can use LLMs to re-write both the prompt (DSPy) and the query to the vector DB



Source: Query Rewriting - Ma et al. 2023
DSPy - Singhvi et al. 2023

Hybrid Search

- **Idea:** Medicine has a lot of specific keywords you might want to use in the search for relevant cases/articles
- **Vector search:** only uses semantic similarity → not great for exact matching
- **Keyword search:** great for exact string matches
- **Hybrid Search:** Use ***both!***



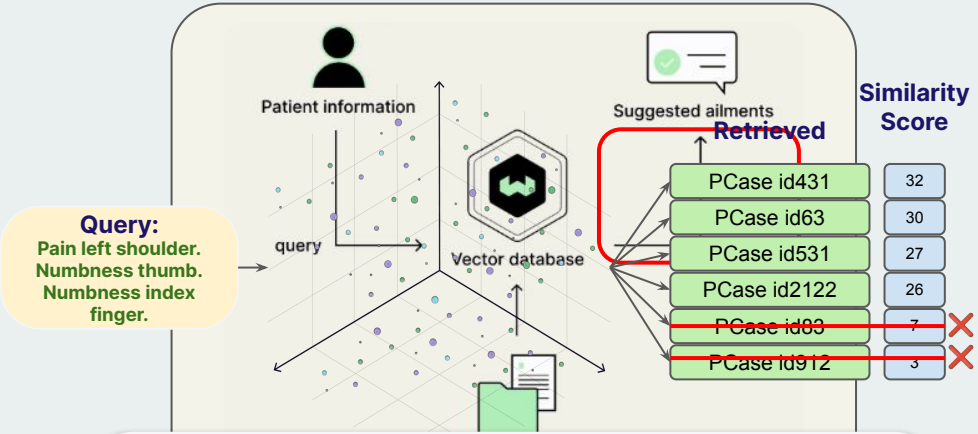

```
questions = client.collections.get("JeopardyQuestion")
response = questions.query.hybrid(
    query="space travel",  # Your query string
    limit=2
)

for o in response.objects:
    print(o.uuid)
    print(o.properties)
```

Medical practice information

Source: Vector DB → Hybrid Search

AutoCut

- **Idea:** If you get irrelevant results from the search → automatically cut them off
- Vector DB will throw away returned objects a “jump” away from relevant objects
- Less chances Vector DB will return irrelevant results and thus confuse LLM

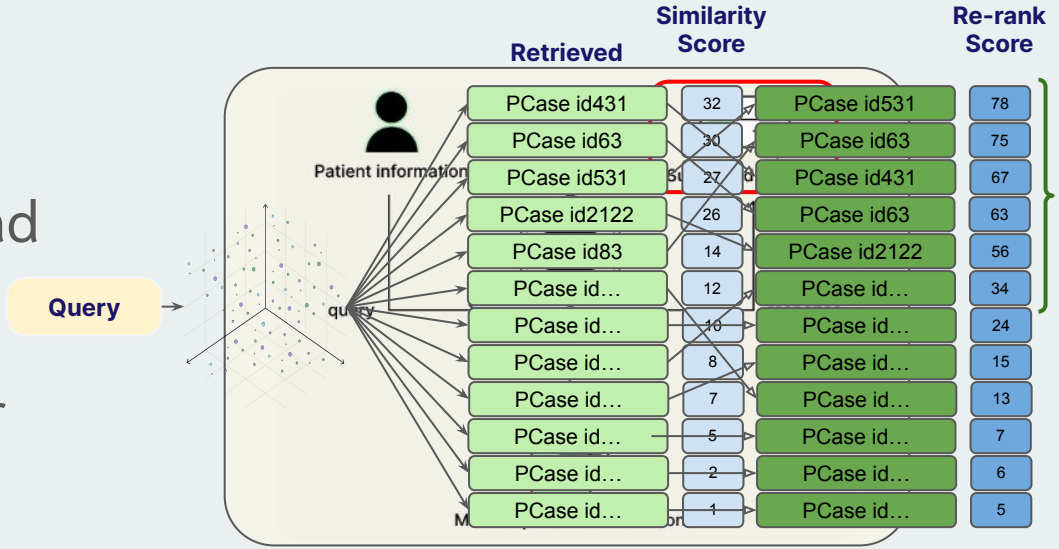



```
jeopardy = client.collections.get("JeopardyQuestion")
response = jeopardy.query.near_text(
    query="animals in movies",
    auto_limit=1, # number of close groups
    return_metadata=wvc.query.MetadataQuery(distance=True)
)
```

Source: Vector DB → AutoCut

Re-ranking

- **Idea:** Sift through top returned patient cases and *re-rank* them based on relevance!
- Over retrieve more similar cases
- Use a heavy model to re-rank top candidates
- Improves quality of cases sent to LLM




```
response = jeopardy.query.near_text(
    query="flying",
    limit=10,
    rerank=wvc.query.Rerank(
        prop="question",
        query="publication"
    ),
    return_metadata=wvc.query.MetadataQuery(score=True)
)
```

Source: Vector DB → [Re-ranker Module]

Voila 🎉

With that we've created an AI based super doctor assistant!

- Retrieves from a knowledge base larger then any human can
- Proposes plausible diagnosis w/ source citations
- Reasons over previous patterns of similar patient cases/articles
- All in realtime (few seconds end-to-end)

Image by pikisuperstar on Freepik

Thank you!

weaviate.io

weaviate/weaviate

@weaviate_io

Connect with me!

Zain Hasan

@zainhasan6

linkedin.com/in/zainhas/

Get started with a free trial!

Weaviate

https://bit.ly/47bxh0X