


Building Robust Retrieval Augmented Generation(RAG): Advanced RAG Techniques



Zain Hasan

 @zainhasan6  zainhas

We  Open Source...



The average human doctor

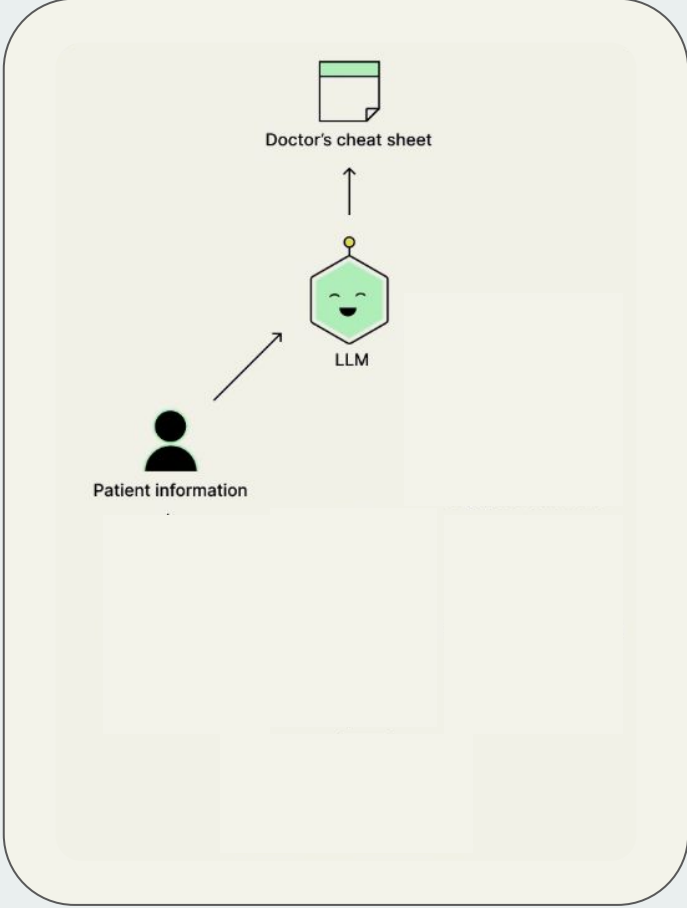


- Studies for 7+ years after undergrad
- See ~100,000 patients in a lifetime

Can we build an **AI powered medical doctor/assistant?**



R.A.G.doctor

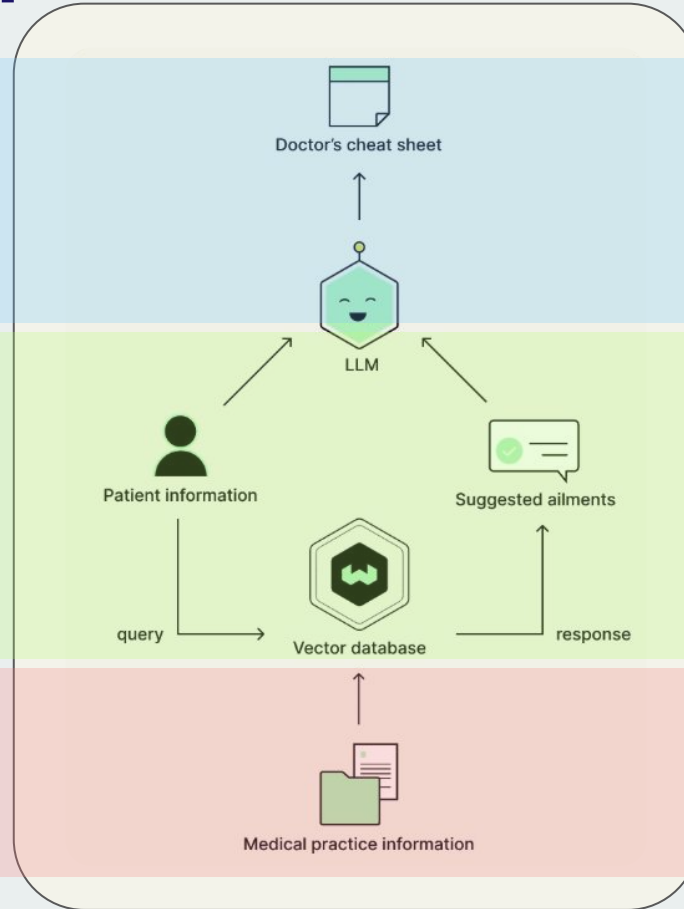


R.A.G.doctor

Generation

Retrieval

Indexing

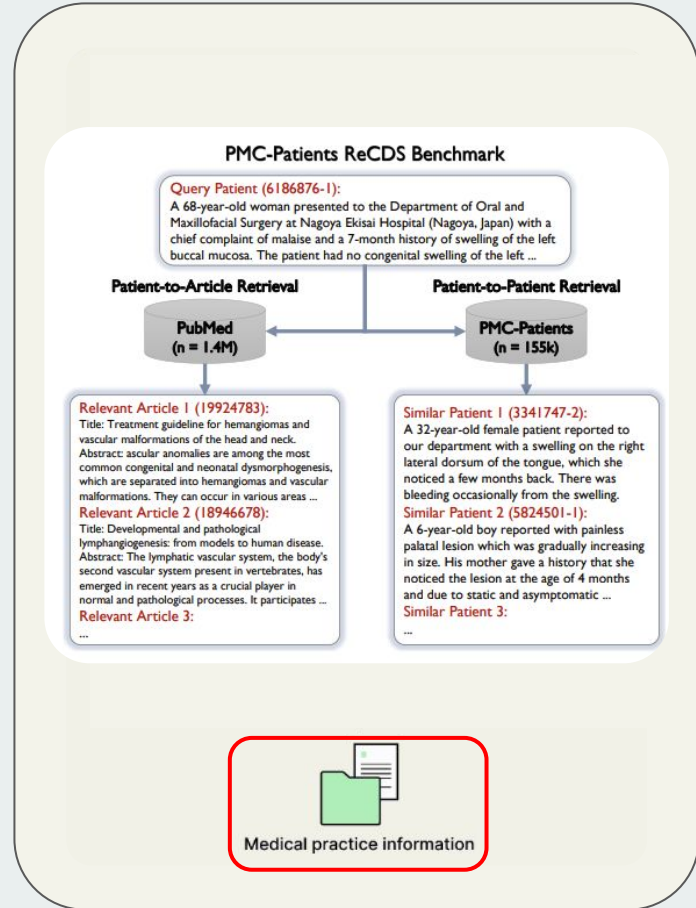


Retrieval
Augmented
Generation

Patient Cases Dataset

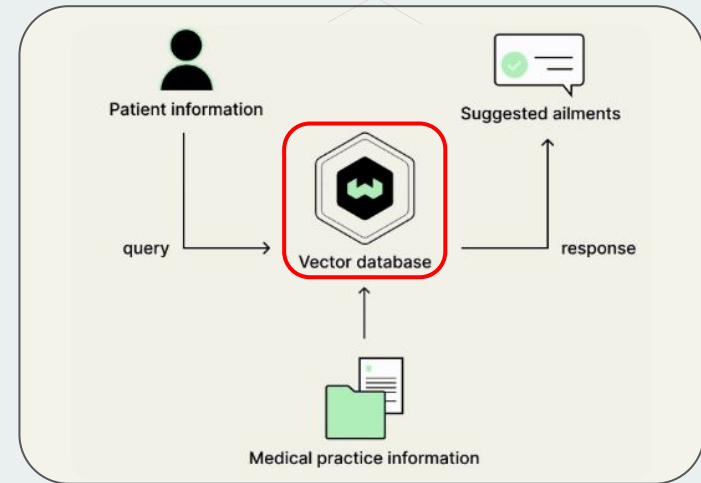
- Open [PMC-Patients](#) dataset
- 167k patient summaries + 1.4M PubMed abstracts
- Data = text, images, charts ...
- ReCDS Benchmark – can be used to assess recall

Source: PMC-Patients - [Zhao et al. 2023](#)



Vector Database – Search

- **Weaviate** – Open source vector DB – Allows you to store **billions** of patient cases and medical articles
- Given a query, responds with top **similar** articles and patient cases



Link: [Open Source Vector Database](#) → [Docs](#)

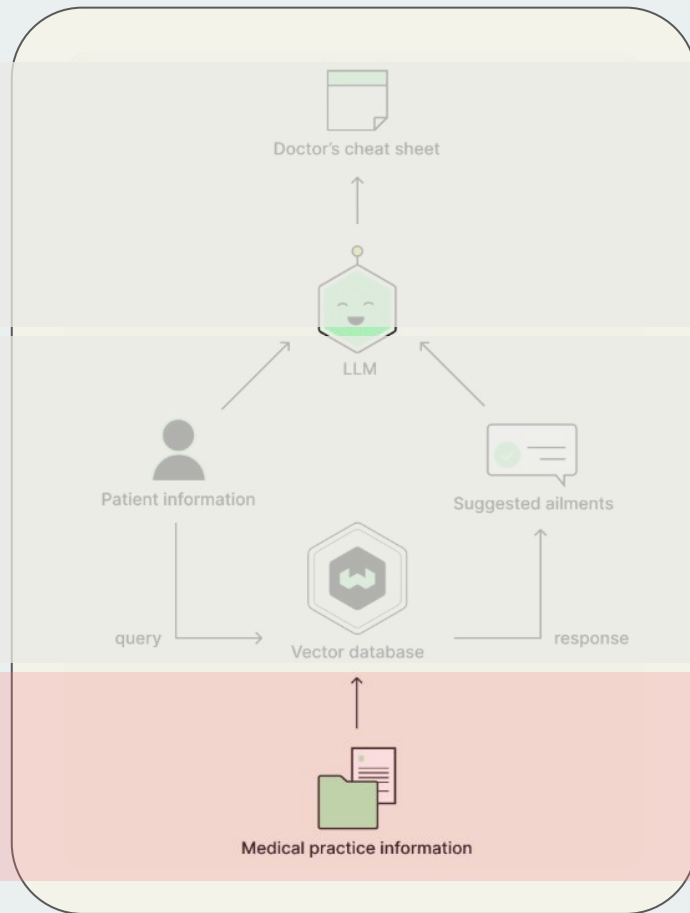


Can we do better ?

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

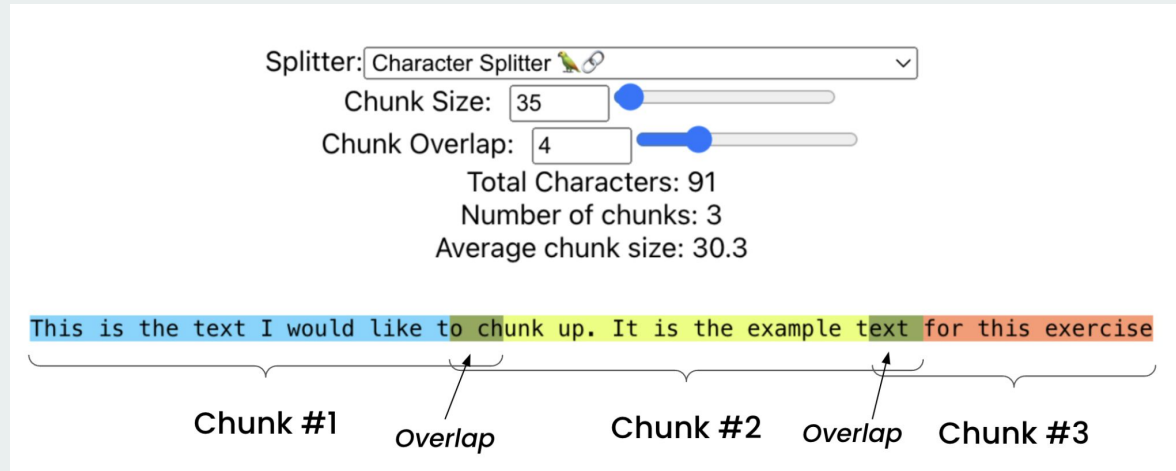


Better Indexing



Chunking Text


- 💡 Chunk our documents into semantically coherent text pieces
- Chunks too large or too small lose semantic meaning – [StackOverflow Blog](#)
- Fixed Size Chunking = Chunks + Overlap





Chunking Text

- 💡 Recursive Character Text Splitter – take into account the structure of our document

Upload .txt

Splitter: Recursive Character Text Splitter 

Chunk Size: 

Chunk Overlap: 

Total Characters: 905
Number of chunks: 3
Average chunk size: 301.7

One of the most important things I didn't understand about the world when I was a child is the degree to which the returns for performance are superlinear.

Teachers and coaches implicitly told us the returns were linear. "You get out," I heard a thousand times, "what you put in." They meant well, but this is rarely true. If your product is only half as good as your competitor's, you don't get half as many customers. You get no customers, and you go out of business.

It's obviously true that the returns for performance are superlinear in business. Some think this is a flaw of capitalism, and that if we changed the rules it would stop being true. But superlinear returns for performance are a feature of the world, not an artifact of rules we've invented. We see the same pattern in fame, power, military victories, knowledge, and even benefit to humanity. In all of these, the rich get richer. [1]



Chunking Text

- **Document Based Chunking**: create chunks using document specific structure
- **HTML** Separators: ["p", "h1", "h2", "h3", "h4", "h5", "h6", "li", "b", "i", "u", "section"]
- **Python** Separators: \nclass, \ndef, \n\tdef, \n\n, \n, " ", ""
- **JavaScript** Separators: \nfunction, \nconst, \nlet, \nvar, \nclass, \nif, \nfor

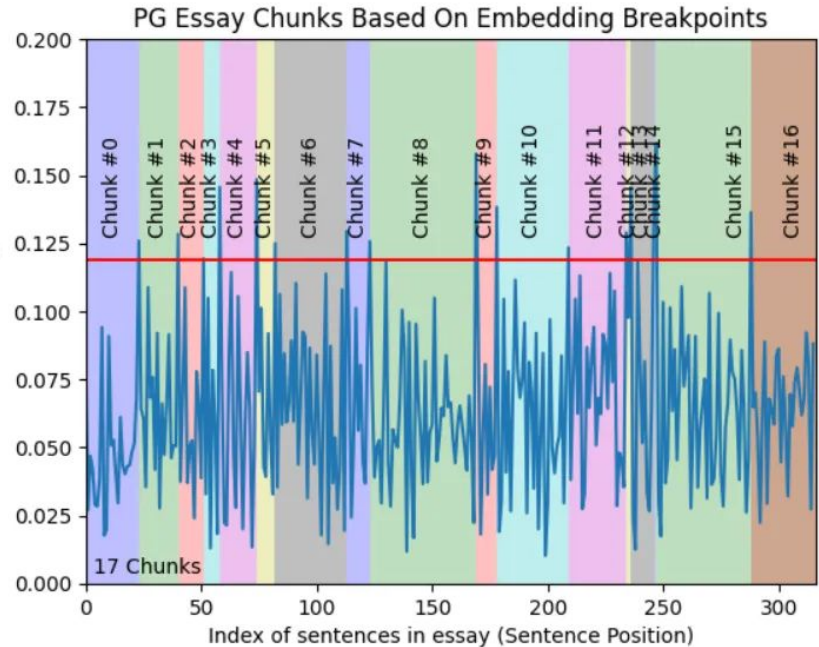


Chunking Text

- Semantic Chunking

I was puzzled by the 1401. I couldn't figure out what to do with it. And in retrospect there's not much I could have done with it. The only form of input to programs was data stored on punched cards...

cosine distance > 0.122



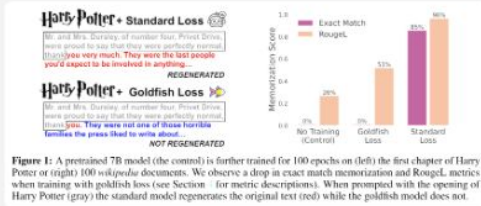
Weaviate Paper Reviews



GLINER: Generalist Model for Named Entity Recognition using Bidirectional Transformer

Using Metadata Filters to Improve Recall in RAG!

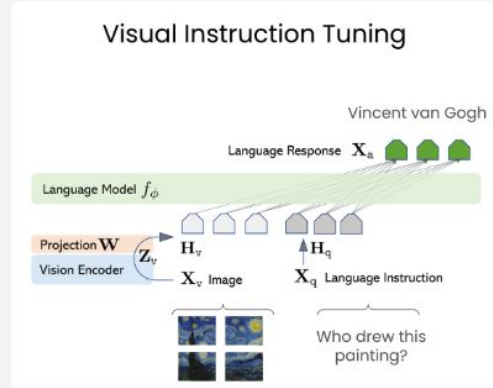
June 22, 2024 · 2 min read



Be like a Goldfish, Don't Memorize! Mitigating Memorization in Generative LLMs

Training LLMs without making them memorize!

June 18, 2024 · 2 min read



Visual Instruction Tuning

Training a LLM to understand images!

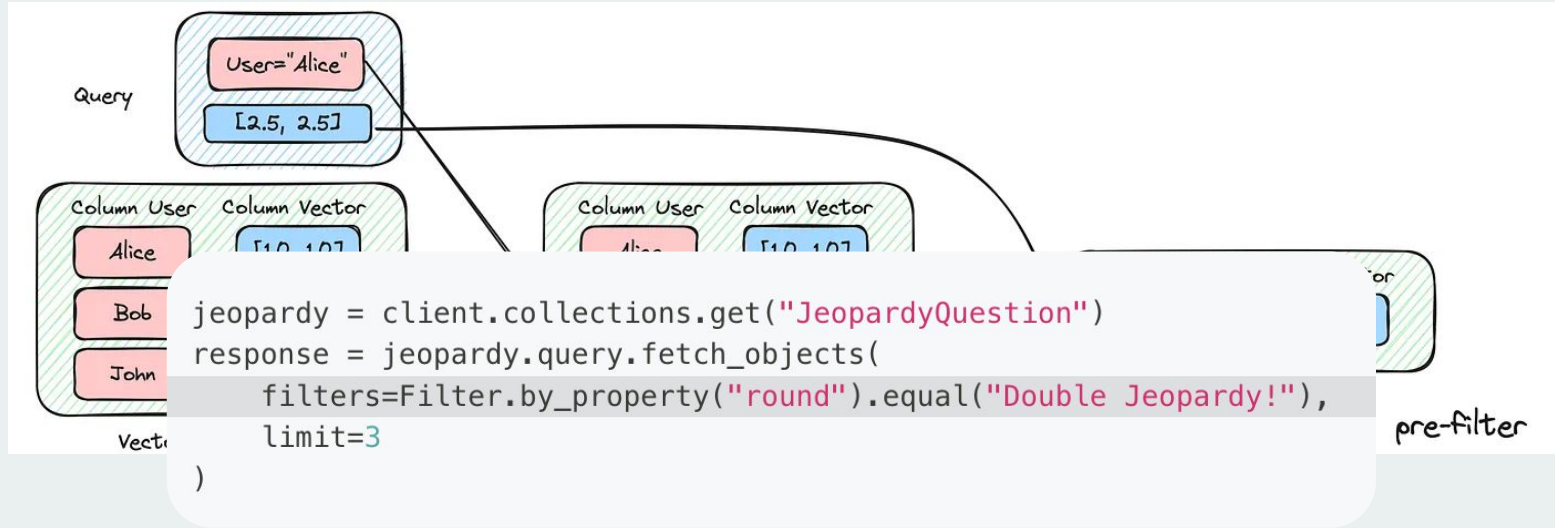
April 28, 2024 · 2 min read

[Dense X Retrieval: What Retrieval Granularity Should We Use? - https://arxiv.org/abs/2312.06648](https://arxiv.org/abs/2312.06648)

Model: <https://huggingface.co/chentong00/propositionizer-wiki-flan-t5-large>

Filtering with Metadata

- 💡 Use meta-data to ensure objects with irrelevant metadata don't even get searched



GLiNER: Create your own metadata while ingesting chunks...

-  Use a model to generate meta-data from text chunks

Text input

Libretto by Marius Petipa, based on the 1822 novella `` Trilby, ou Le Lutin d'Argail `` by Charles Nodier, first presented by the Ballet of the Moscow Imperial Bolshoi Theatre on January 25/February 6 (Julian/Gregorian calendar dates), 1870, in Moscow with Polina Karpakova as Trilby and Ludiia Geiten as Miranda and restaged by Petipa for the Imperial Ballet at the Imperial Bolshoi Kamenny Theatre on January 17-29, 1871 in St. Petersburg with Adèle Grantzow as Trilby and Lev Ivanov as Count Leopold.

Labels
person, book, location, date, actor, character

Threshold
Lower the threshold to increase how many entities get predicted. 0.3

Allow for nested NER?
 Nested NER

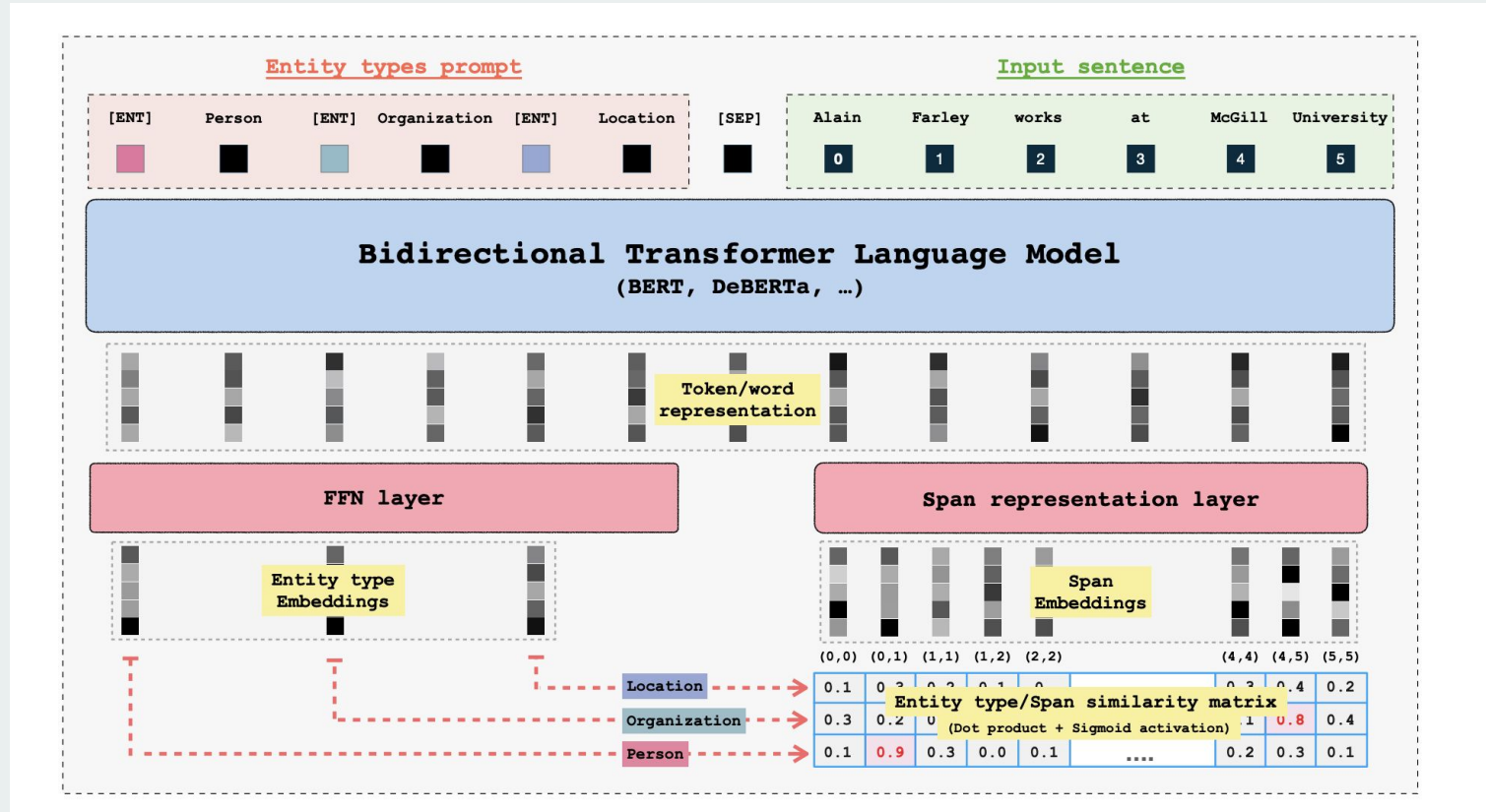
Predicted Entities

Libretto by **Marius Petipa** **PERSON**, based on the 1822 novella `` **Trilby** **CHARACTER**, ou Le Lutin d'Argail `` by **Charles Nodier** **BOOK**, first presented by the Ballet of the **Moscow** **LOCATION** **Moscow Imperial Bolshoi Theatre** **LOCATION** on **January 25/February 6** **DATE** **February 6** **DATE** (Julian/Gregorian calendar dates), **1870** **DATE**, in **Moscow** **LOCATION** with **Polina Karpakova** **ACTOR** as **Trilby** **CHARACTER** and **Ludiia Geiten** **ACTOR** as **Miranda** **CHARACTER** and restaged by Petipa for the Imperial Ballet at the Imperial Bolshoi Kamenny Theatre on **January 17-29, 1871** **DATE** in **St. Petersburg** **LOCATION** with **Adèle Grantzow** **ACTOR** as **Trilby** **CHARACTER** and **Lev Ivanov** **ACTOR** as **Count Leopold** **CHARACTER**.

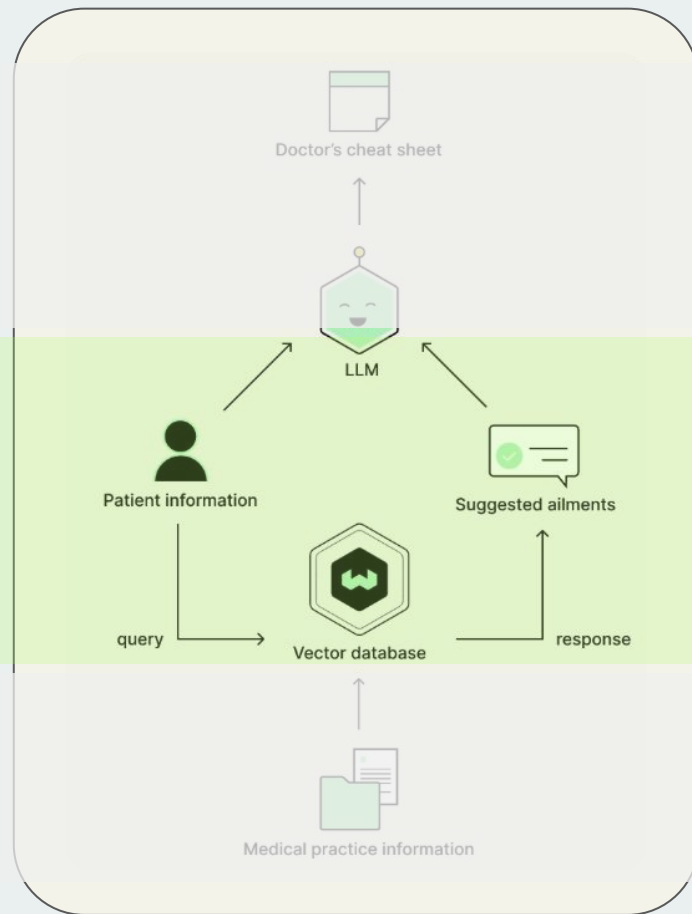
Source: <https://huggingface.co/spaces/tomaarsen/gliner-medium-v2.1>



GLiNER Architecture

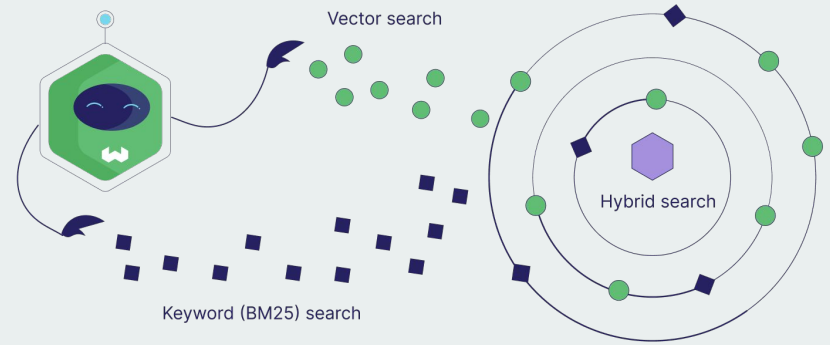


Better Retrieval



Hybrid Search

- 💡 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!*



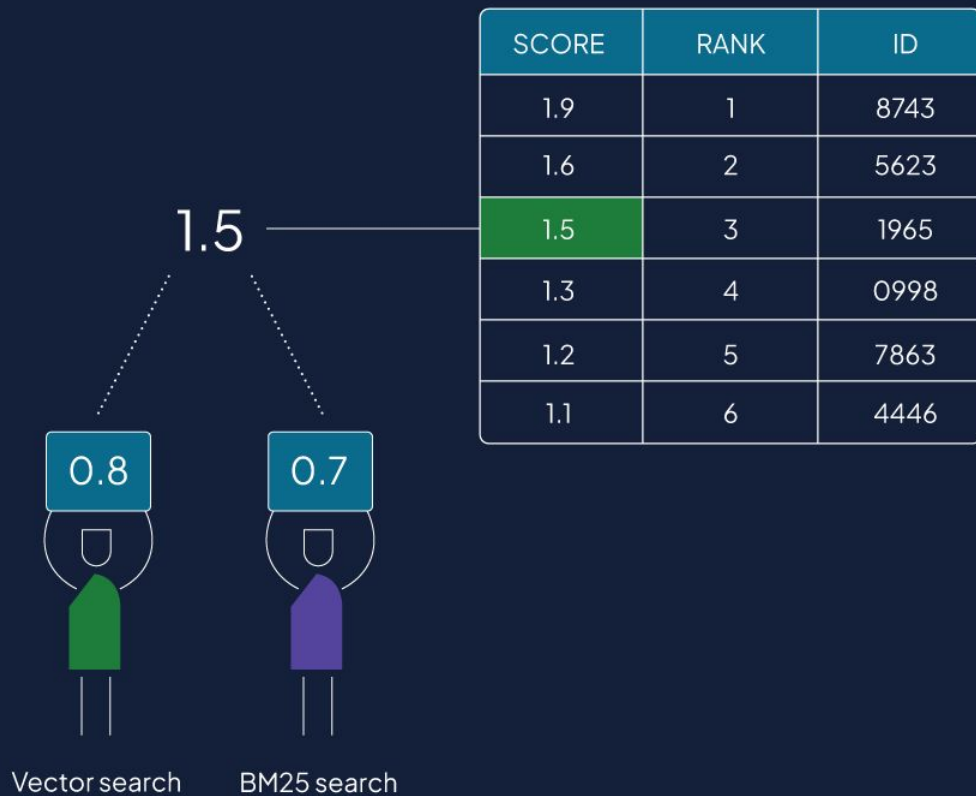
```
questions = client.collections.get("JeopardyQuestion")
response = questions.query.hybrid(
    query="space travel", # Your query string
    limit=2
)

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

Medical practice information

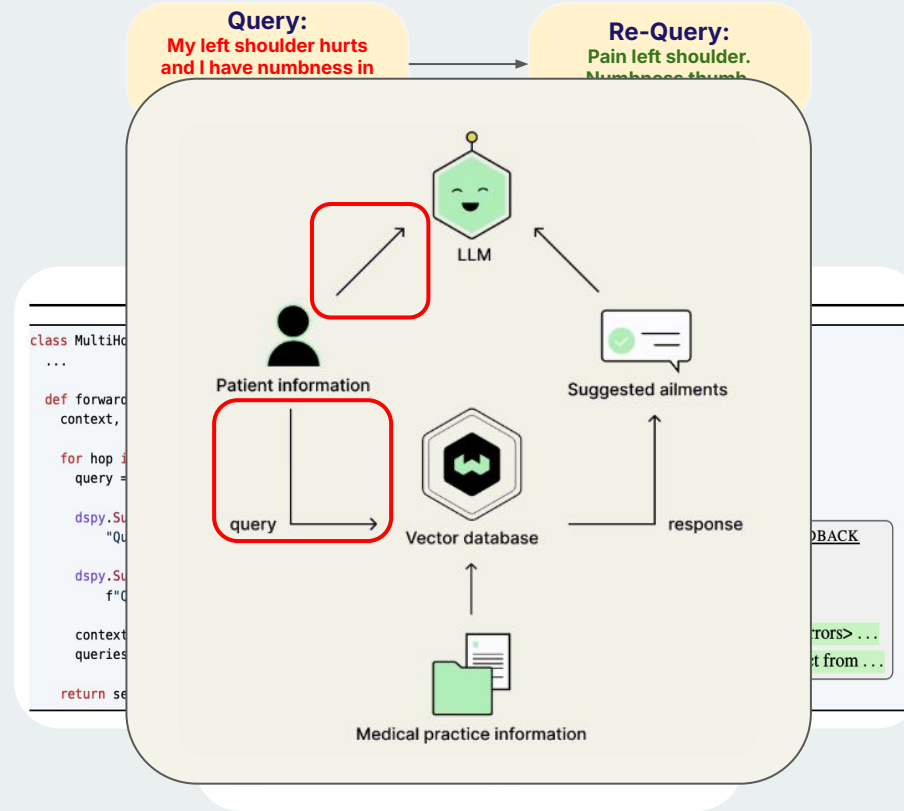
Source: Vector DB → [Hybrid Search](#)





Query Rewriting

- 💡 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



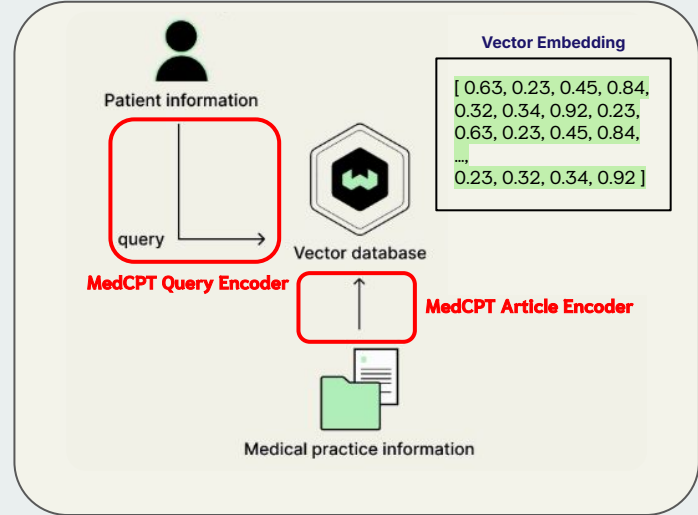
Source: [Query Rewriting - Ma et al. 2023](#)

[DSPy - Singhvi et al. 2023](#) - <https://github.com/weaviate/recipes/tree/main/integrations/llm-frameworks/dspy>

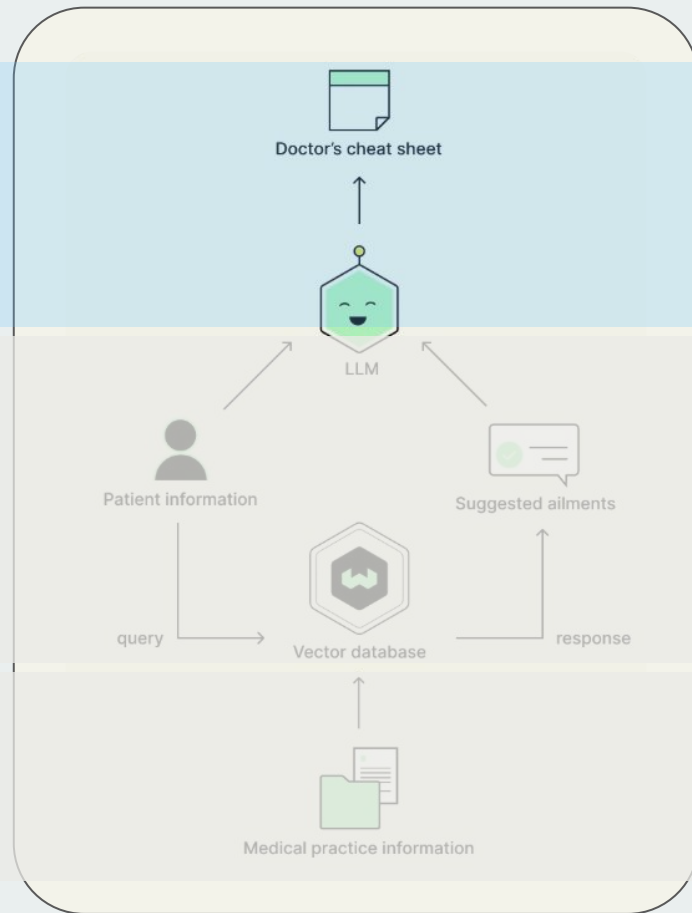


Fine-Tuned Embedding Models

- 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

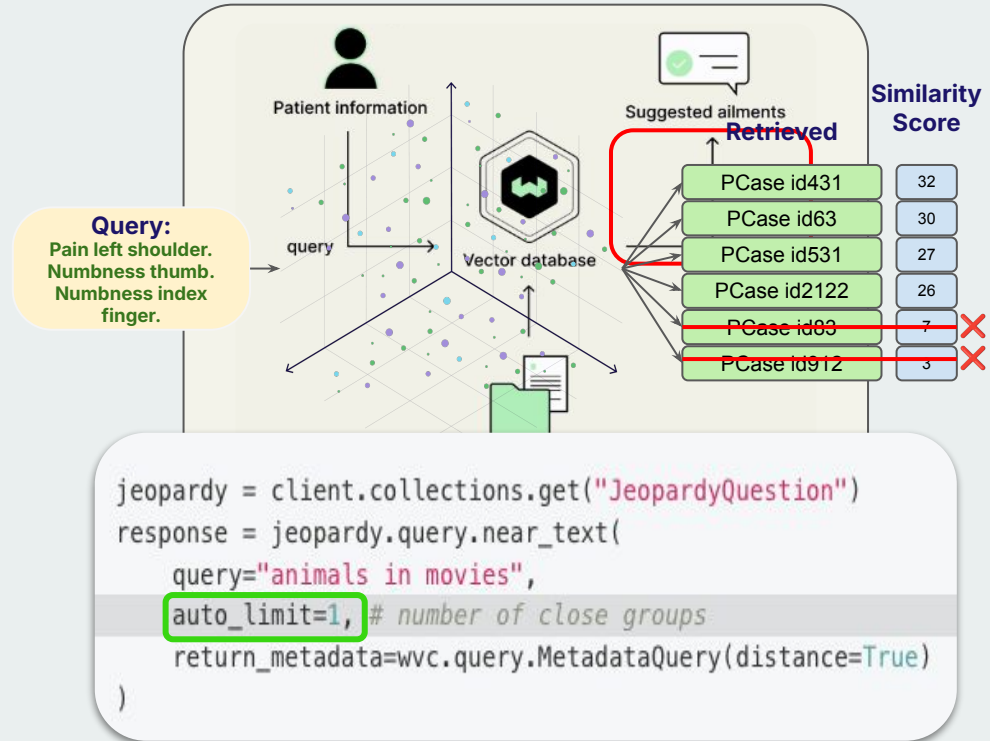


Better Generation




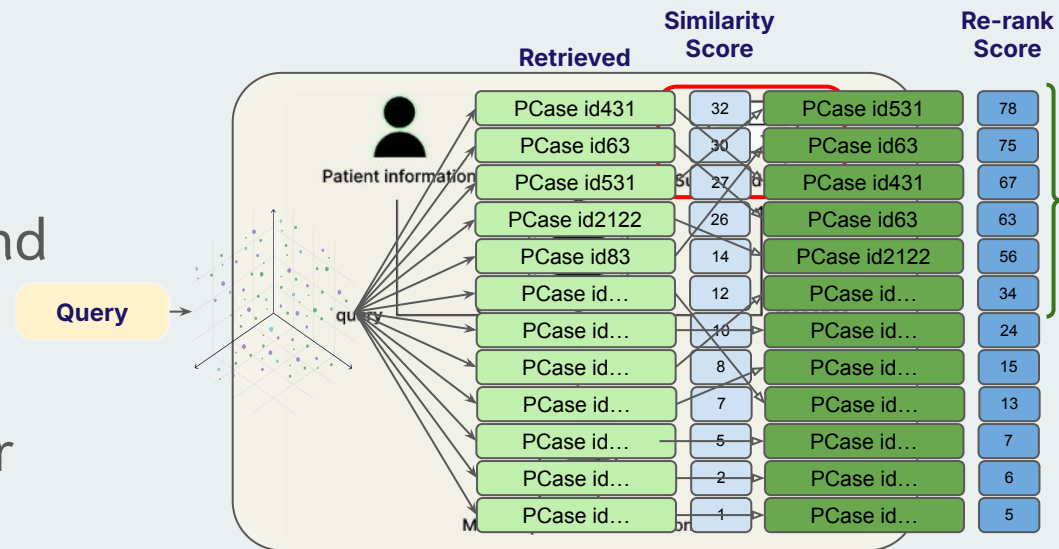
AutoCut

- 💡 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



Re-ranking

-  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

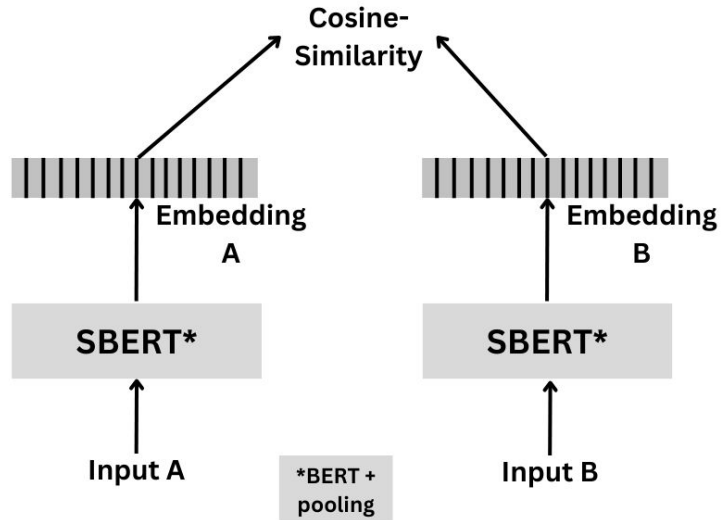


```
response = jeopardy.query.near_text(  
    query="flying",  
    limit=10,  
    rerank=wvc.query.Rerank(  
        prop="question",  
        query="publication"  
    ),  
    return_metadata=wvc.query.MetadataQuery(score=True)  
)
```


Source: Vector DB → [Re-ranker Module](#)

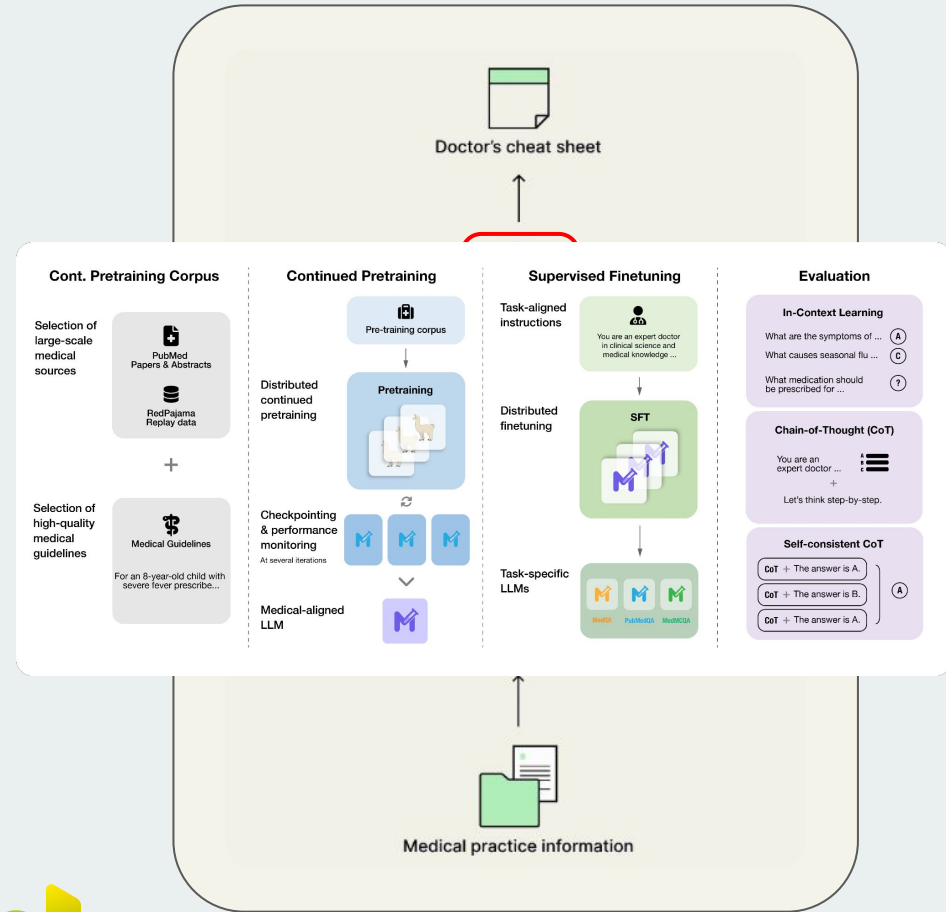


Bi-encoder



Fine-tuning LLMs

-  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](#)

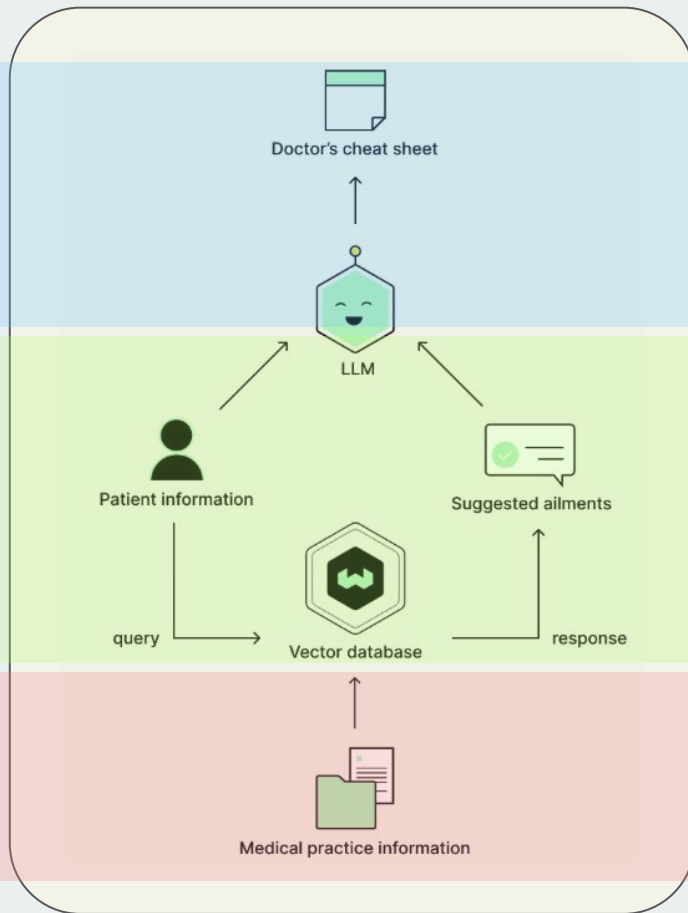


Overview

Generation

Retrieval

Indexing



- Auto-cut
- Re-ranking
- FT LLM

- Hybrid Search
- Query Rewriting
- FT Embedding Models

- Chunking
- Filtering

Code!

```
import weaviate, os, json
import weaviate.classes as wc
from weaviate.classes.query import Rerank, Filter
from wikipediaapi import Wikipedia
```

Query Overview

```
query = "What is Chihiro's new name given to her by the witch?"

#Run Hybrid query, pre-filtered on the category property and reranked and take results and pipe them into the LLM
response = wiki_collection.generate.hybrid(query = query,
                                          limit = 5,
                                          alpha = 0.7,
                                          filters = Filter.by_property('category').equal('film'),
                                          rerank = Rerank(prop="text", query=query),
                                          grouped_task=query
                                          )

print(response.generated) #Output = "Sen"
```



Great papers for even more Advanced RAG details:

- A Survey on Retrieval-Augmented Text Generation for Large Language Models - <https://arxiv.org/pdf/2404.10981v1>
- Retrieval-Augmented Generation for Large Language Models: A Survey - <https://arxiv.org/pdf/2312.10997>
- <https://www.oreilly.com/radar/what-we-learned-from-a-year-of-building-with-llms-part-i/> - Part 1, 2 and 3





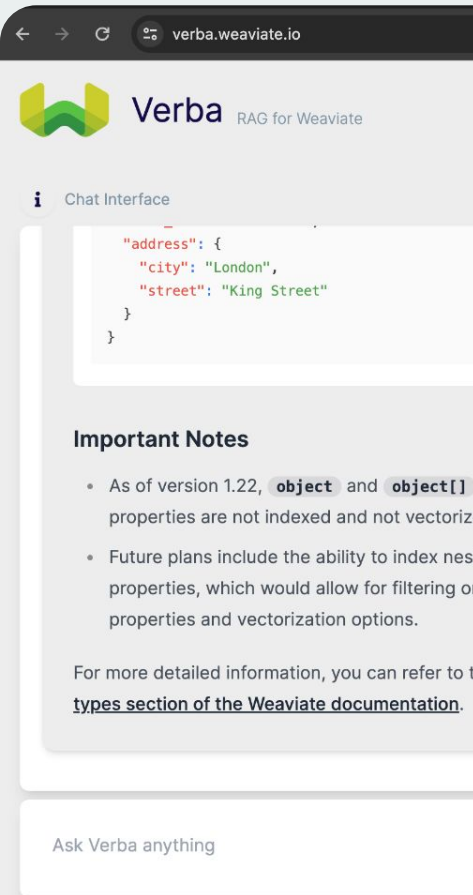
Community
RAG Corner

**Join the Discussion
in Slack**

#rag-corner



Verba – Open Source RAG App – verba.weaviate.io



verba.weaviate.io

Verba RAG for Weaviate

Chat Interface

```
"address": {  
  "city": "London",  
  "street": "King Street"  
}
```

Important Notes

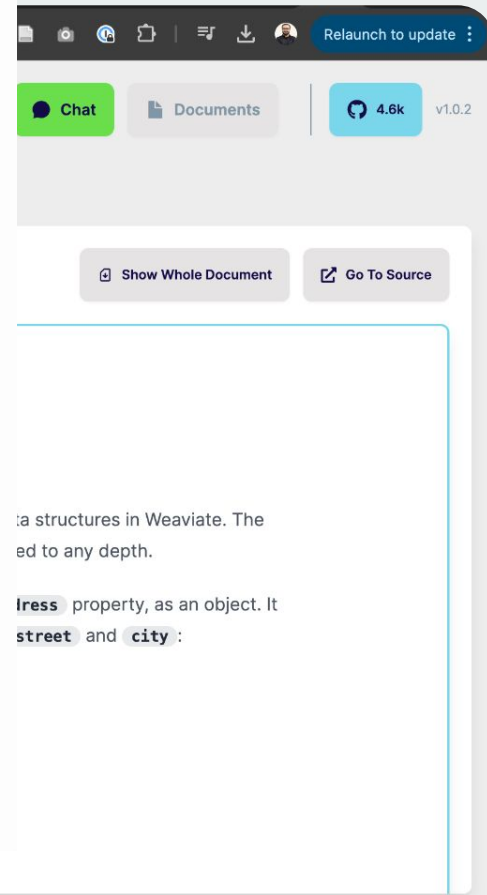
- As of version 1.22, `object` and `object[]` properties are not indexed and not vectorized.
- Future plans include the ability to index nested properties, which would allow for filtering on nested properties and vectorization options.

For more detailed information, you can refer to the [types section of the Weaviate documentation](#).

Ask Verba anything



<https://github.com/weaviate/Verba>



Relaunch to update

Chat Documents 4.6k v1.0.2

Show Whole Document Go To Source

ia structures in Weaviate. The
ed to any depth.

ress property, as an object. It
street and city :

"name" : "last_name" ,

Join our next Live Webinar

Thursday, August 22

You will learn

- Patterns and best practices for running **AI applications at scale**
- How to find the **sweet spot between cost and performance**
- How to **increase Developer Productivity** with Weaviate Workbench - including our new [Recommender service](#).



LIVE WEBINAR

Weaviate Product Update: Optimizing AI infrastructure for your use case

Thursday, August 22nd | 10am PT / 1pm ET / 7pm CEST

 Hot Memory	 Warm SSD Drives	 Cold Cloud Storage
Speed 	Speed 	Speed 
Cost 	Cost 	Cost 

Thank **you!**



weaviate.io



[weaviate/weaviate](https://github.com/weaviate/weaviate)



[@weaviate_io](https://twitter.com/weaviate_io)

Connect with me!

Zain Hasan



[@zainhasan6](https://twitter.com/zainhasan6)



linkedin.com/in/zainhas/

