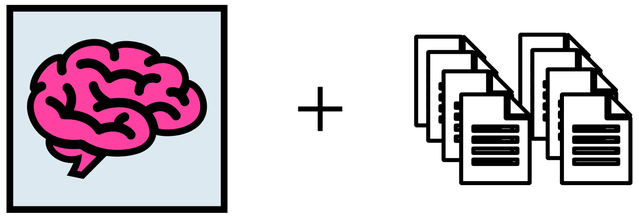
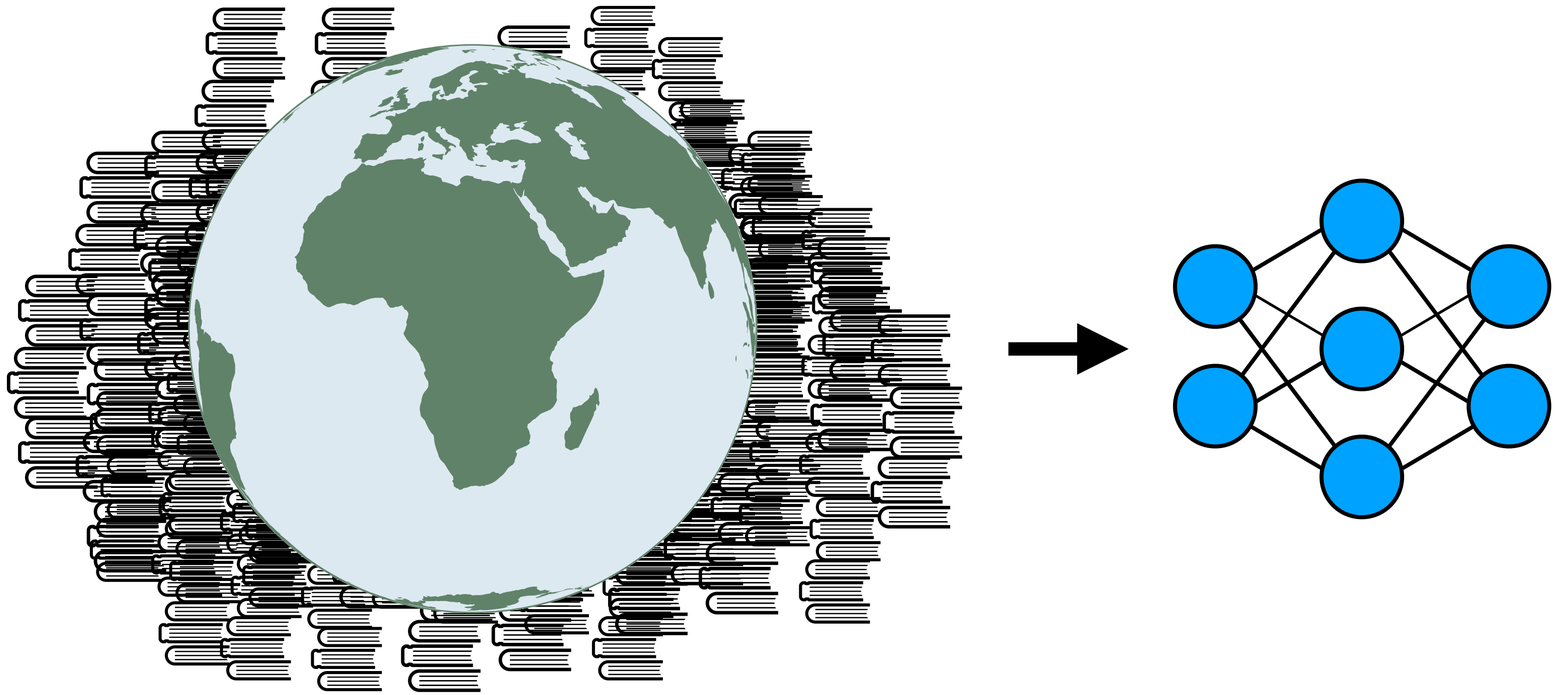
Improving LLMs with RAG

A beginner-friendly introduction

Shaw Talebi



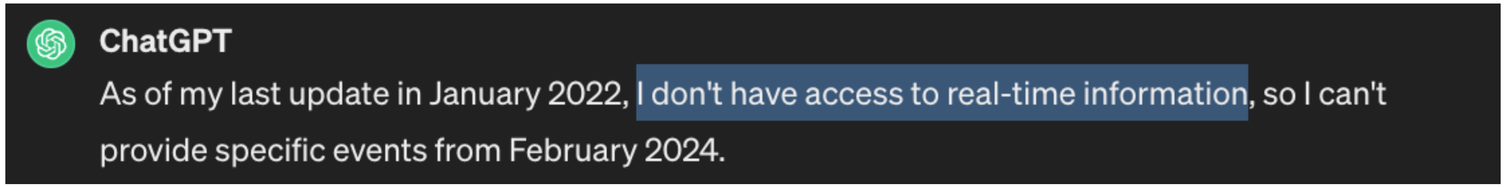
LLMs Compress World Knowledge



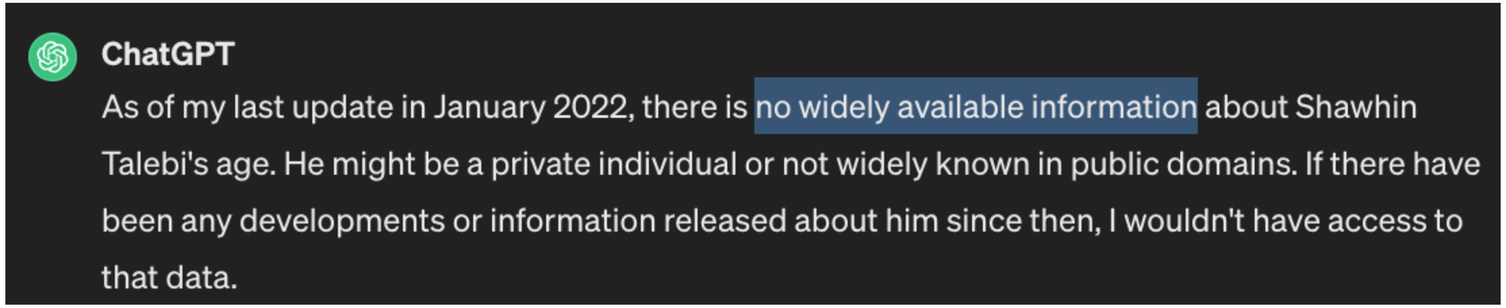
*not to scale

2 Key Limitations

1) Static World Knowledge

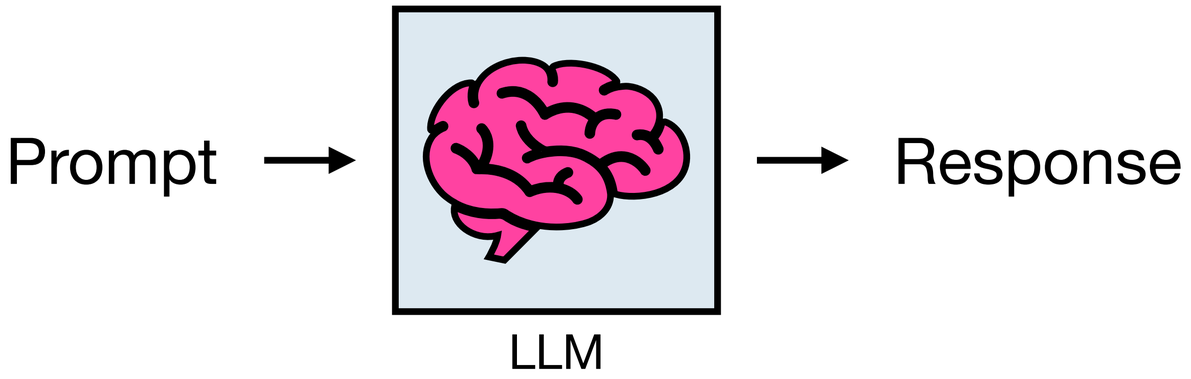


2) Lack of Specialized Information



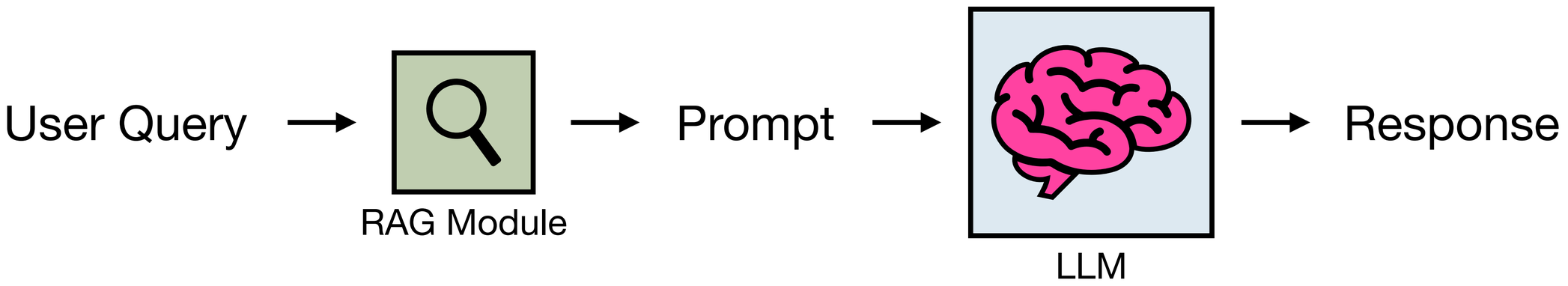
What is RAG?

Augmenting LLM with specialized and mutable knowledge base



Why we care?

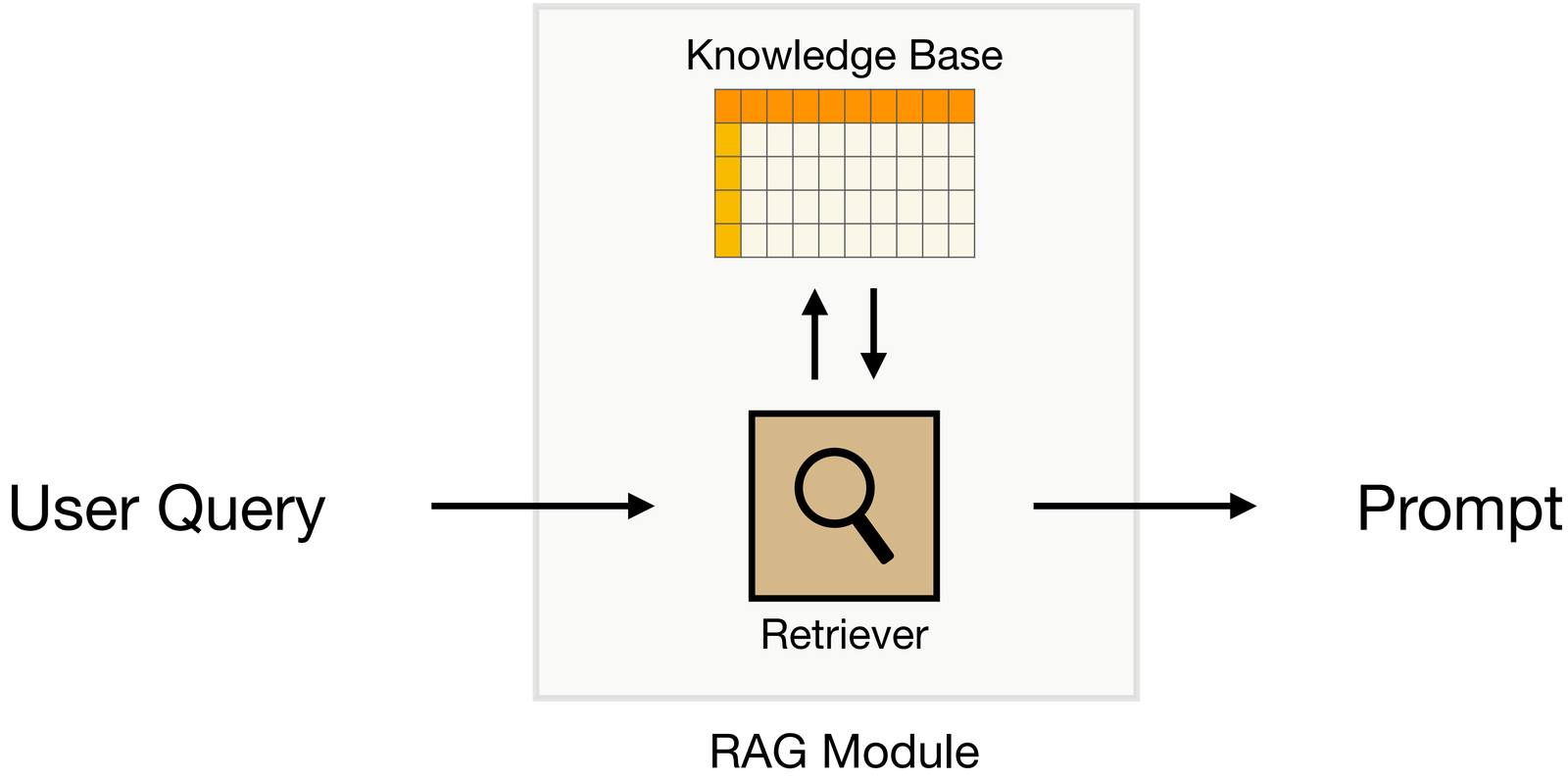
Why not FT?



 [1]

How it works?

2 key elements: retriever and knowledge base

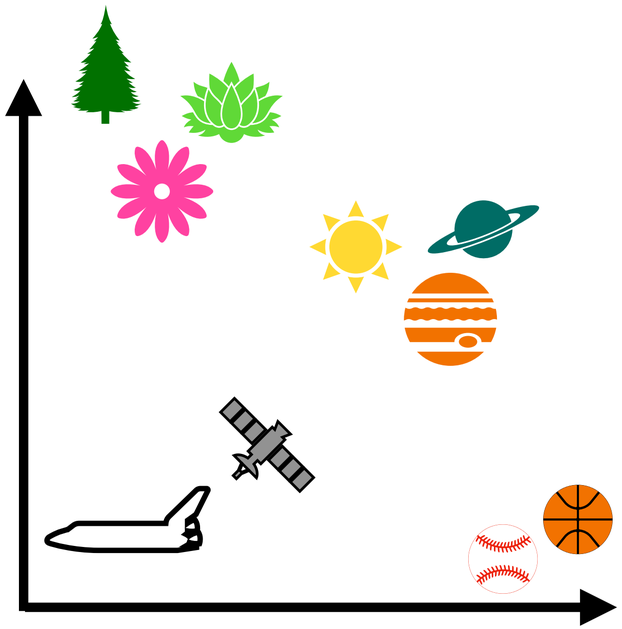


 [2] [3]

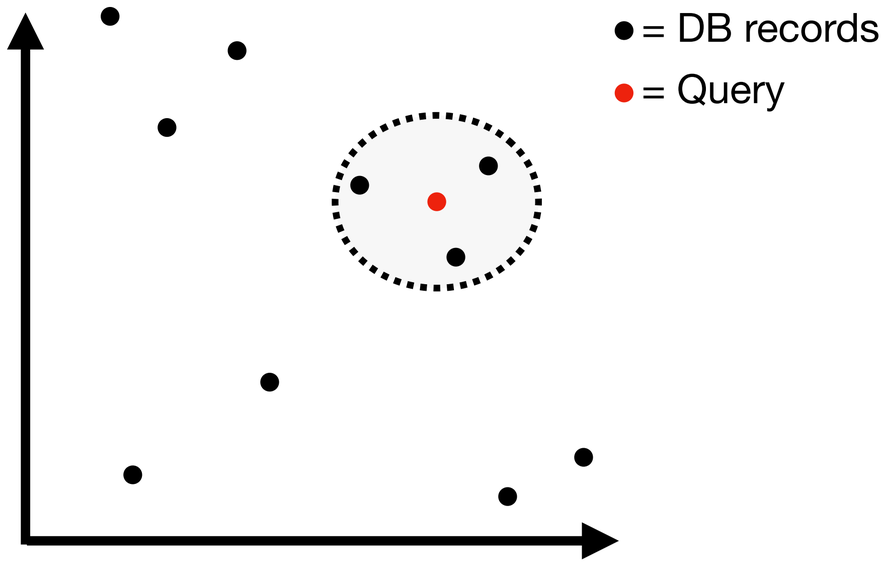
How it works?

2 key elements: **retriever** and knowledge base

Text Embeddings

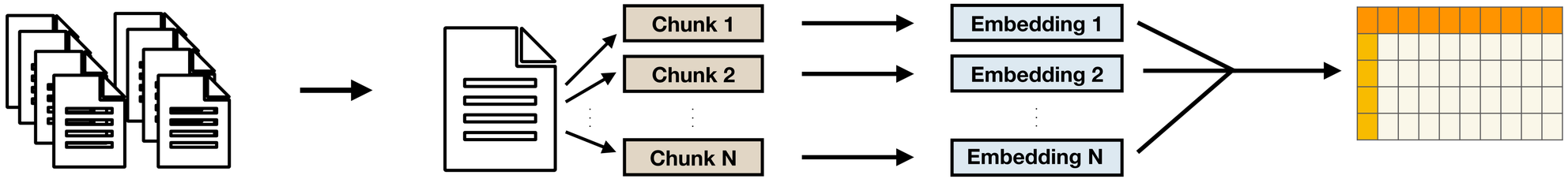


Text Embedding-based Search



How it works?

2 key elements: retriever and **knowledge base**

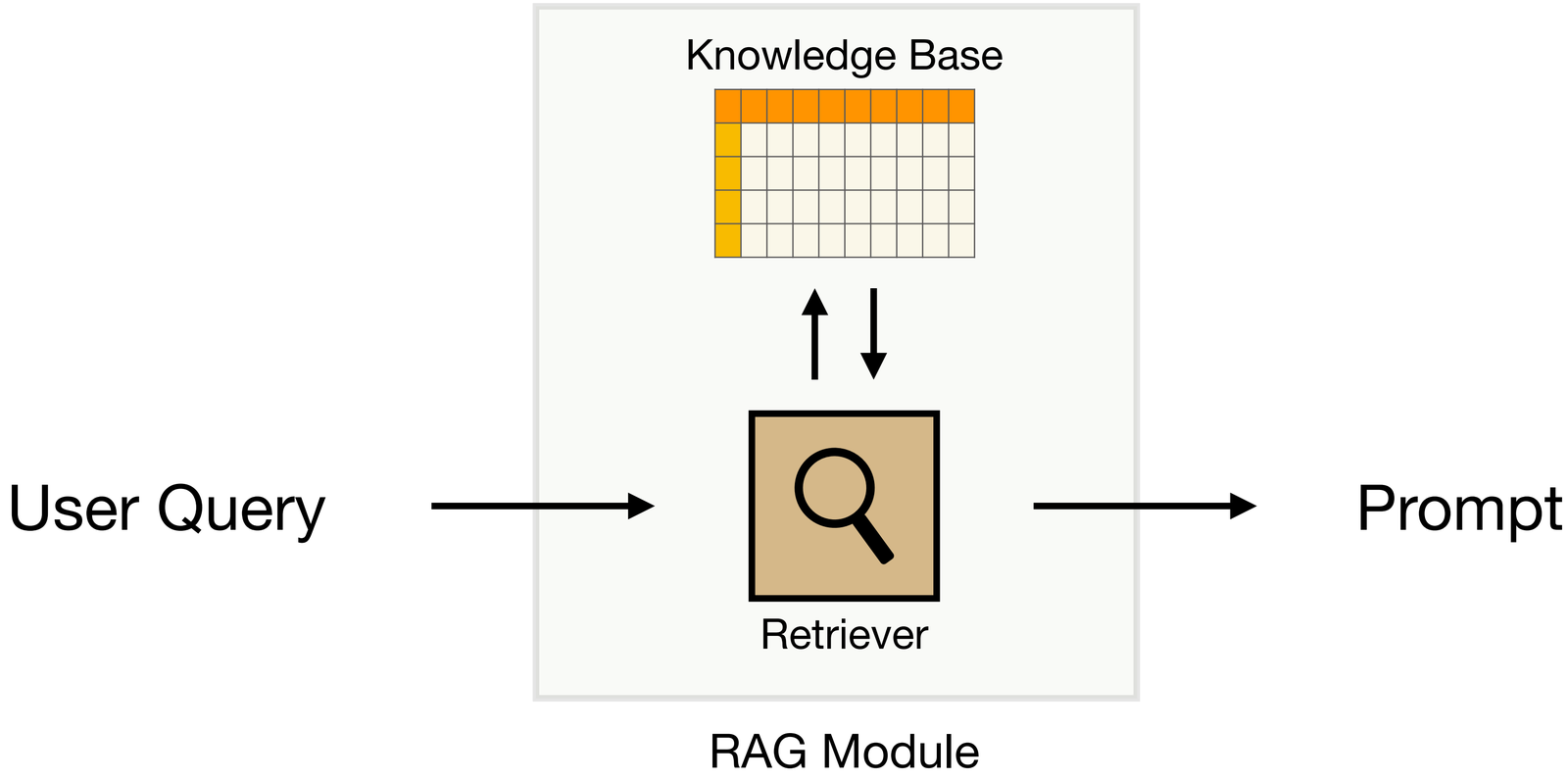


1) Load Docs **2) Chunk Docs** **3) Embed Chunks** **4) Load into VDB**

[2] [3]

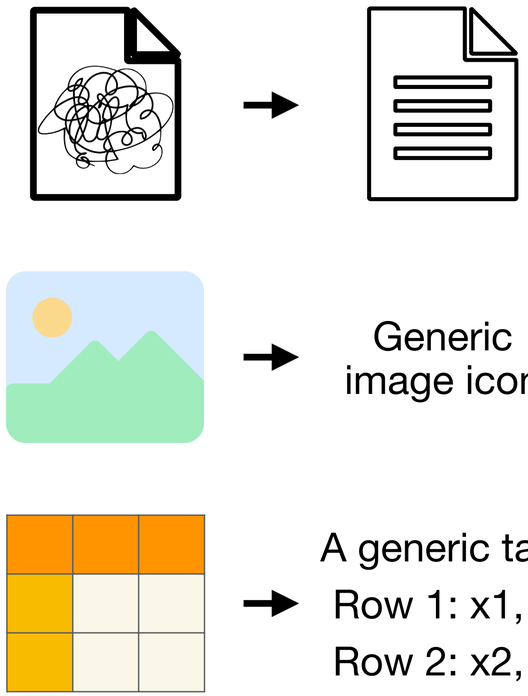
How it works?

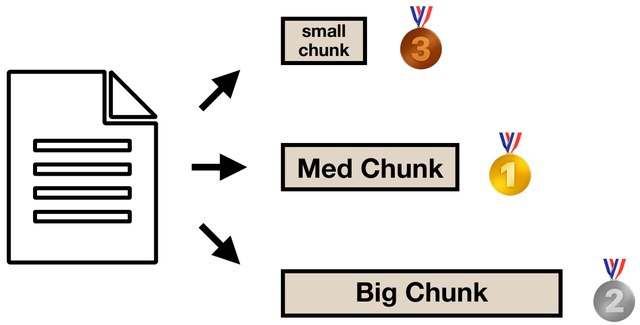
2 key elements: retriever and knowledge base

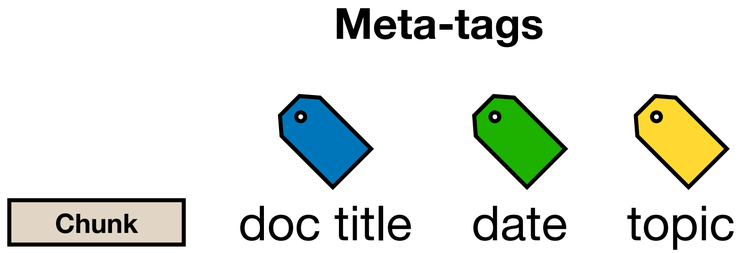


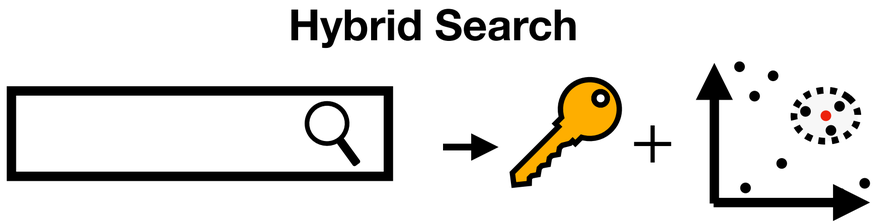
 [2] [3]

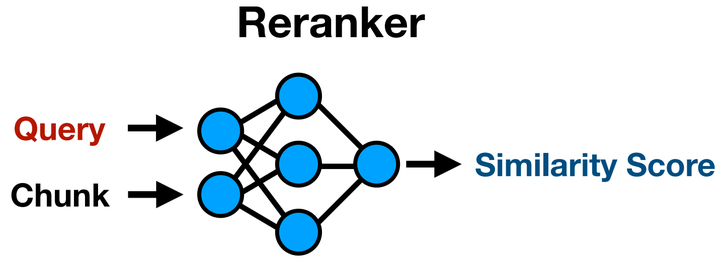
Some Nuances











Doc Prep **Optimizing Chunk Size** **Improving Search**

 [4]

Example Code: Improving YouTube Comment Responder with RAG

Imports

```
!pip install llama-index
!pip install llama-index-embeddings-huggingface
!pip install peft
!pip install auto-gptq
!pip install optimum
!pip install bitsandbytes
# if not running on Colab ensure transformers is installed too
```

Links in description

```
from llama_index.embeddings.huggingface import HuggingFaceEmbedding
from llama_index.core import Settings, SimpleDirectoryReader, VectorStoreIndex
from llama_index.core.retrievers import VectorIndexRetriever
from llama_index.core.query_engine import RetrieverQueryEngine
from llama_index.core.postprocessor import SimilarityPostprocessor
```


https://github.com/ShawhinT/YouTube-Blog/tree/main/LLMs

Example Code: Improving YouTube Comment Responder with RAG

Setting up Knowledge Base

```
# import any embedding model on HF hub
Settings.embed_model = HuggingFaceEmbedding(model_name="BAAI/bge-small-en-v1.5")

Settings.llm = None # we won't use LlamaIndex to set up LLM
Settings.chunk_size = 256
Settings.chunk_overlap = 25
```

```
documents = SimpleDirectoryReader("articles").load_data()
```


https://github.com/ShawhinT/YouTube-Blog/tree/main/LLMs

Example Code: Improving YouTube Comment Responder with RAG

Setting up Knowledge Base

```
print(len(documents)) # prints: 71
for doc in documents:
    if "Member-only story" in doc.text:
        documents.remove(doc)
        continue

    if "The Data Entrepreneurs" in doc.text:
        documents.remove(doc)

    if " min read" in doc.text:
        documents.remove(doc)

print(len(documents)) # prints: 61
```

```
'Member-only story\n4 Ways to Quantify Fat Tails with\nPyth
on\nIntuition and Example Code\nShaw Talebi\nPublished inTo
wards Data Science·11 min read·Dec 7, 2023\n200 8\nA fat (c
at's) tail. Image from Canva.\nOpen in app\nSearch Write\n'
```

```
'Although each approach has its limitations, they provide p
ractitioners with\nquantitative ways of comparing the fat-t
ailedness of empirical data.\n👉 More on Power Laws & Fat Ta
ils: Introduction | Power Law Fits\nResources\nConnect: My
website | Book a call | Ask me anything\nSocials: YouTube 📽
| LinkedIn | Twitter\nSupport: Buy me a coffee ☕\nThe Data
Entrepreneurs\nA community for entrepreneurs in the data sp
ace. 👉  Join the\nDiscord!\nmedium.com\n[1] Scipy Kurtosis
\n[2] Scipy Moment\n[3] arXiv:1802.05495 [stat.ME]'
```

```
'[4] https://en.wikipedia.org/wiki/Log-normal_distribution
\n[5] Pham-Gia, T., & Hung, T. (2001). The mean and median
absolute\ndeviations. Mathematical and Computer Modelling,
34(7–8), 921 – 936.\nhttps://doi.org/10.1016/S0895-7177(01)
00109-1\nMore from the list: "Data Science"\nCurated byShaw
Talebi\nView listData Science Statistics Python Fat Tails P
ower Law\nShaw T…inTowards Data …\n5 Questions Every Data\n
Scientist Should…\n·6 min read·Dec 21, 2023Shaw T…inTowards
ata with…\n·10
nPareto, Power
n'
```

```
index = VectorStoreIndex.from_documents(documents)
```


https://github.com/ShawhinT/YouTube-Blog/tree/main/LLMs

Example Code: Improving YouTube Comment Responder with RAG

Setting up Retriever

```
# set number of docs to retreive
top_k = 3

# configure retriever
retriever = VectorIndexRetriever(
    index=index,
    similarity_top_k=top_k,
)
```

```
# assemble query engine
query_engine = RetrieverQueryEngine(
    retriever=retriever,
    node_postprocessors=[SimilarityPostprocessor(similarity_cutoff=0.5)],
)
```


https://github.com/ShawhinT/YouTube-Blog/tree/main/LLMs

Example Code: Improving YouTube Comment Responder with RAG

Use Query Engine

```
query = "What is fat-tailed
response = query_engine.que
```

```
# reformat response
context = "Context:\n"
for i in range(top_k):
    context = context + res

print(context)
```

```
Context:
Some of the controversy might be explained by the observation that log-
normal distributions behave like Gaussian for low sigma and like Power Law
at high sigma [2].
However, to avoid controversy, we can depart (for now) from whether some
given data fits a Power Law or not and focus instead on fat tails.
Fat-tailedness — measuring the space between Mediocristan
and Extremistan
Fat Tails are a more general idea than Pareto and Power Law distributions.
One way we can think about it is that “fat-tailedness” is the degree to which
rare events drive the aggregate statistics of a distribution. From this point of
view, fat-tailedness lives on a spectrum from not fat-tailed (i.e. a Gaussian) t
very fat-tailed (i.e. Pareto 80 - 20).
This maps directly to the idea of Mediocristan vs Extremistan discussed
earlier. The image below visualizes different distributions across this
conceptual landscape [2].

print("mean kappa_1n = " + str(np.mean(kappa_dict[filename])))
    print("")
Mean κ (1,100) values from 1000 runs for each dataset. Image by author.
These more stable results indicate Medium followers are the most fat-tailed,
followed by LinkedIn Impressions and YouTube earnings.
Note: One can compare these values to Table III in ref [3] to better understand
κ value. Namely, these values are comparable to a Pareto distribution with α
between 2 and 3.
Although each heuristic told a slightly different story, all signs point toward
Medium followers gained being the most fat-tailed of the 3 datasets.
Conclusion
While binary labeling data as fat-tailed (or not) may be tempting, fat-
tailedness lives on a spectrum. Here, we broke down 4 heuristics for
quantifying how fat-tailed data are.

Pareto, Power Laws, and Fat Tails
What they don’t teach you in statistics
towardsdatascience.com
Although Pareto (and more generally power law) distributions give us a
salient example of fat tails, this is a more general notion that lives on a
spectrum ranging from thin-tailed (i.e. a Gaussian) to very fat-tailed (i.e.
Pareto 80 - 20).
The spectrum of Fat-tailedness. Image by author.
This view of fat-tailedness provides us with a more flexible and precise way of
categorizing data than simply labeling it as a Power Law (or not). However,
this begs the question: how do we define fat-tailedness?
4 Ways to Quantify Fat Tails
```


https://github.com/ShawhinT/YouTube-Blog

Example Code: Improving YouTube Comment Responder with RAG

Import Fine-tuned Model

```
# load fine-tuned model from hub
from peft import PeftModel, PeftConfig
from transformers import AutoModelForCausalLM, AutoTokenizer

model_name = "TheBloke/Mistral-7B-Instruct-v0.2-GPTQ"
model = AutoModelForCausalLM.from_pretrained(model_name,
                                             device_map="auto",
                                             trust_remote_code=False,
                                             revision="main")

config = PeftConfig.from_pretrained("shawhin/shawgpt-ft")
model = PeftModel.from_pretrained(model, "shawhin/shawgpt-ft")

# load tokenizer
tokenizer = AutoTokenizer.from_pretrained(model_name, use_fast=True)
```


https://github.com/ShawhinT/YouTube-Blog/tree/main/LLMs

Example Code: Improving YouTube Comment Responder with RAG

Create Prompt (no context)

```
# prompt (no context)
intstructions_string = f"""ShawGPT, functioning as a virtual data science \
consultant on YouTube, communicates in clear, accessible language, escalating \
to technical depth upon request. It reacts to feedback aptly and ends \
responses with its signature '-ShawGPT'.

ShawGPT will tailor the length of its responses to match the viewer's comment, \
providing concise acknowledgments to brief expressions of gratitude or \
feedback, thus keeping the interaction natural and engaging.
```

```
[INST] ShawGPT, functioning as a virtual data science consultant on YouTube, com

Please respond to the following comment.

What is fat-tailedness?
[/INST]
```


https://github.com/ShawhinT/YouTube-Blog/tree/main/LLMs

Example Code: Improving YouTube Comment Responder with RAG

Ge

```
Great question!

Fat-tailedness is a statistical property of a distribution. In simple terms,
it refers to the presence of extreme outliers or heavy tails in the
distribution.

For instance, consider the distribution of heights in a population. A normal
distribution would have most people clustered around an average height with a
few people deviating slightly from the mean. However, in a fat-tailed
distribution, you would observe a larger number of people being significantly
taller or shorter than the average.

In finance, fat-tailedness is often used to describe the presence of extreme
market moves or outliers in financial returns. It's an important concept in
risk management and financial modeling.

I hope this explanation helps! Let me know if you have any questions.
-ShawGPT
```


https://github.com/ShawhinT/YouTube-Blog/tree/main/LLMs

Example Code: Improving YouTube Comment Responder with RAG

Create Prompt (with context)

```
# prompt (with context)
prompt_template_w_context = lambda context, comment: f"""[INST]ShawGPT, function
It reacts to feedback aptly and ends responses with its signature '-ShawGPT'. \
ShawGPT will tailor the length of its responses to match the viewer's comment, p
thus keeping the interaction natural and engaging.

{context}
Please respond to the following comment. Use the context above if it is helpful.

{comment}
[/INST]
"""
```


https://github.com/ShawhinT/YouTube-Blog/tree/main/LLMs

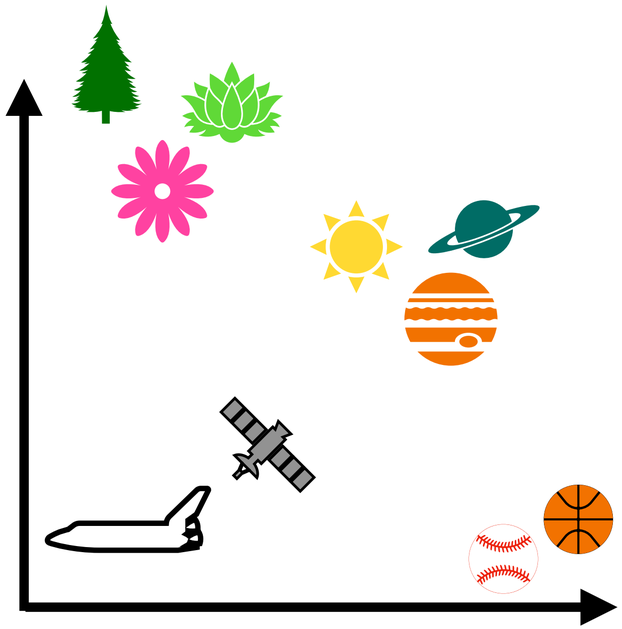
Example Code: Improving YouTube Comment Responder with RAG

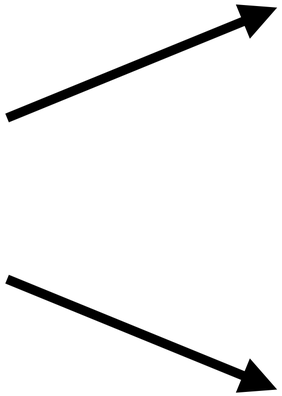
```
Great question!

Fat-tailedness is a property of a distribution that describes how much the
presence of extreme values (rare events) affects the overall statistics of the
distribution.

In the context of the video, the author discusses how Mediocristan and
Extremistan distributions differ in terms of fat-tailedness. Mediocristan
distributions have a relatively small impact from extreme values, while
Extremistan distributions are heavily influenced by extreme values.

The author also mentions that fat-tailedness is a more general concept than
specific distributions like Pareto or Power Laws. Instead, it lives on a
spectrum ranging from thin-tailed (Gaussian) to very fat-tailed (Pareto 80-20).

I hope that helps clarify things a bit! Let me know if you have any questions.
-ShawGPT
```

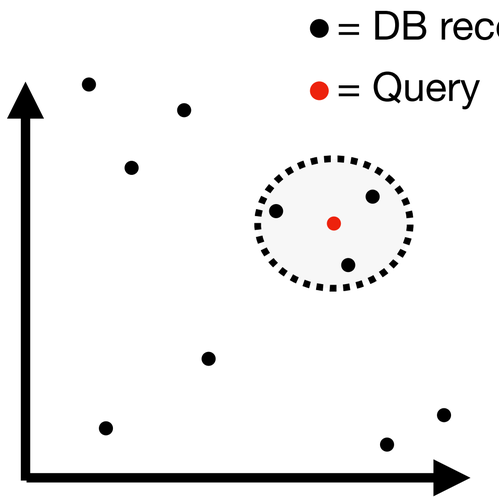

https://github.com/ShawhinT/YouTube-Blog/tree/main/LLMs

What's Next?

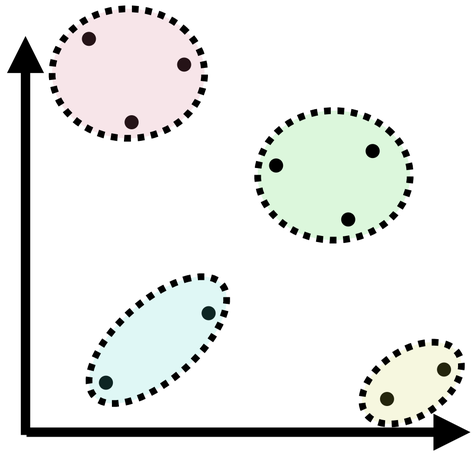
Text Embeddings







1) Semantic Search



2) Classification

✦ Member-only story

How to Improve LLMs with RAG

A beginner-friendly introduction w/ Python code

Shaw Talebi

Published in Towards Data Science · 12 min read · 2 days ago

432 1