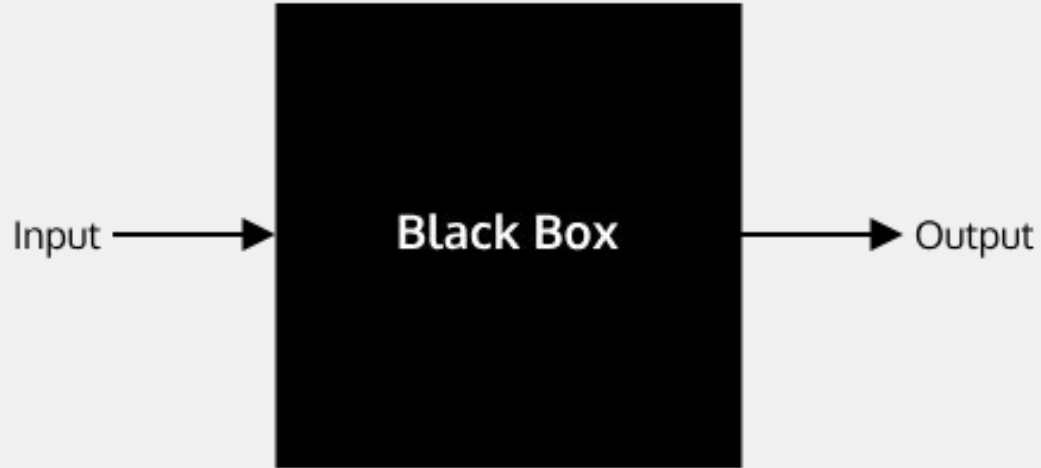


Retrieval Augmented Generative AI

Solving the black box problem

October 3 2023 6:30 PM | SLC AI Tinkerers Meetup

In LLMs we do not trust...



In LLMs we do not trust...



Example of black box in action



Who is taylor swift dating

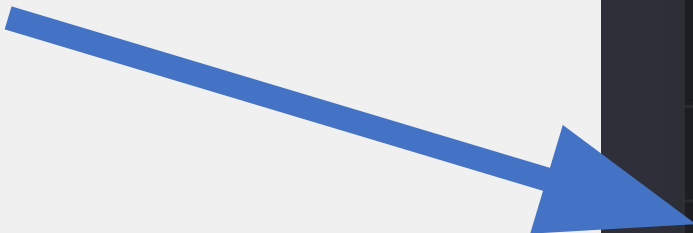


As of my last update in January 2022, Taylor Swift was dating British actor Joe Alwyn. They had reportedly been in a relationship since 2016 and have kept their relationship largely private. However, details beyond this point could have changed, and I cannot provide the most up-to-date information. If you're interested in the latest news about Taylor Swift's personal life, I recommend checking out recent news articles or her social media channels for the most current information.

False!



But wait, what's this!?




⚡ GPT-3.5 🌐 GPT-4

Our most capable model, great for tasks that require creativity and advanced reasoning.

Available exclusively to Plus users

GPT-4 currently has a cap of 50 messages every 3 hours.

✦ Default

🌐 Browse with  Bing Beta ✓

📊 Advanced Data Analysis Beta

⚙️ Plugins Beta

Correct answer! This is retrieval augmentation in action



Who is Taylor Swift dating as of september 2023



Finished browsing



I am unable to access the detailed pages from my search results due to certain restrictions. However, it seems from the snippets available that there are rumors about Taylor Swift dating Travis Kelce as of September 2023. For more accurate and detailed information, you might want to visit the pages from the search results directly or search for the latest updates on Taylor Swift's dating status on news or entertainment websites.

What is Retrieval Augmented Generation?

- Retrieve data relevant to query (Search for news on Taylor Swift)
- Inject into prompt (Who is Taylor Swift dating? *insert search results*)
- Response is more accurate, and can cite source (Travis Kelce, see this article)

Advantages

- Users can take a foundational model and augment it
- This means no fine-tuning required
- **Cheaper** (fine tuning is expensive!)
- **Faster** (fine tuning takes a long time!)
- **Easier** (fine tuning has a learning curve!)
- **Scales well** (continuously add new data)
- **Biggest advantage: sources can be cited!**
- **No more black box problem**



That's cool! How do I build a RAG AI with my own data?

Enter the vector database!

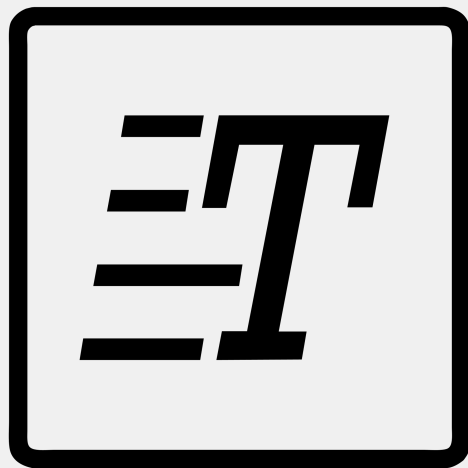


Vector

(Despicable Me)

What is a vector database?

It stores embeddings (which are vectors)



Text input (or other mediums)



Pass to neural network embedding model (easiest way is hit the OpenAI api)



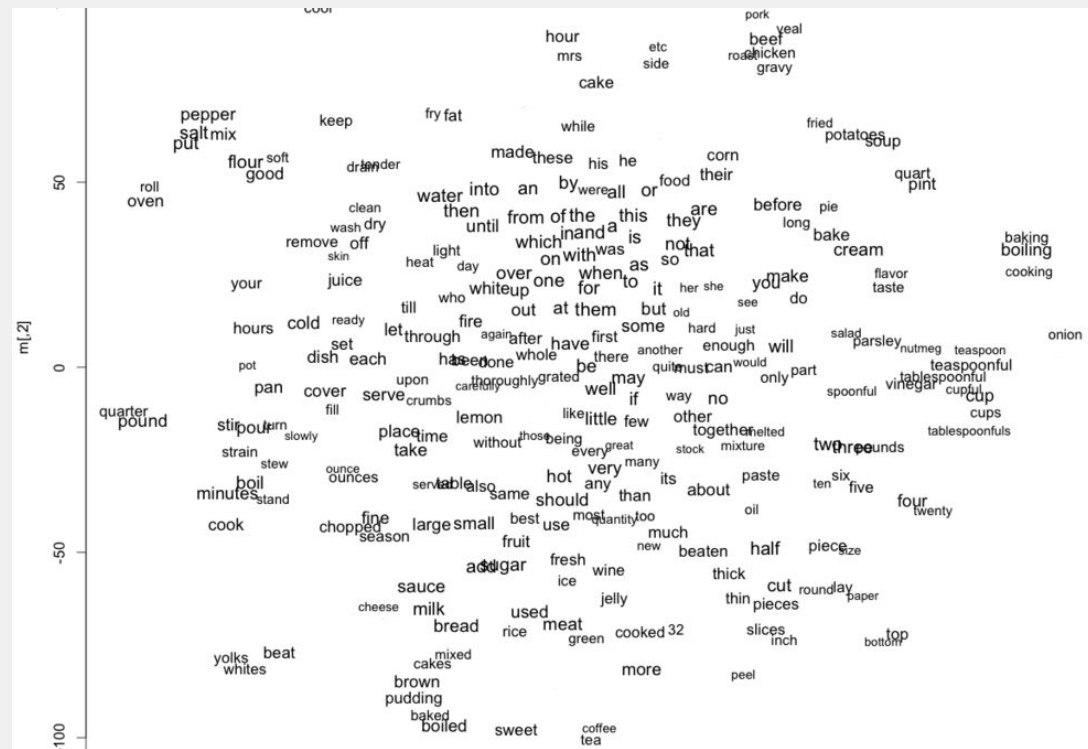
Embedding vectors

Get back an embedding!

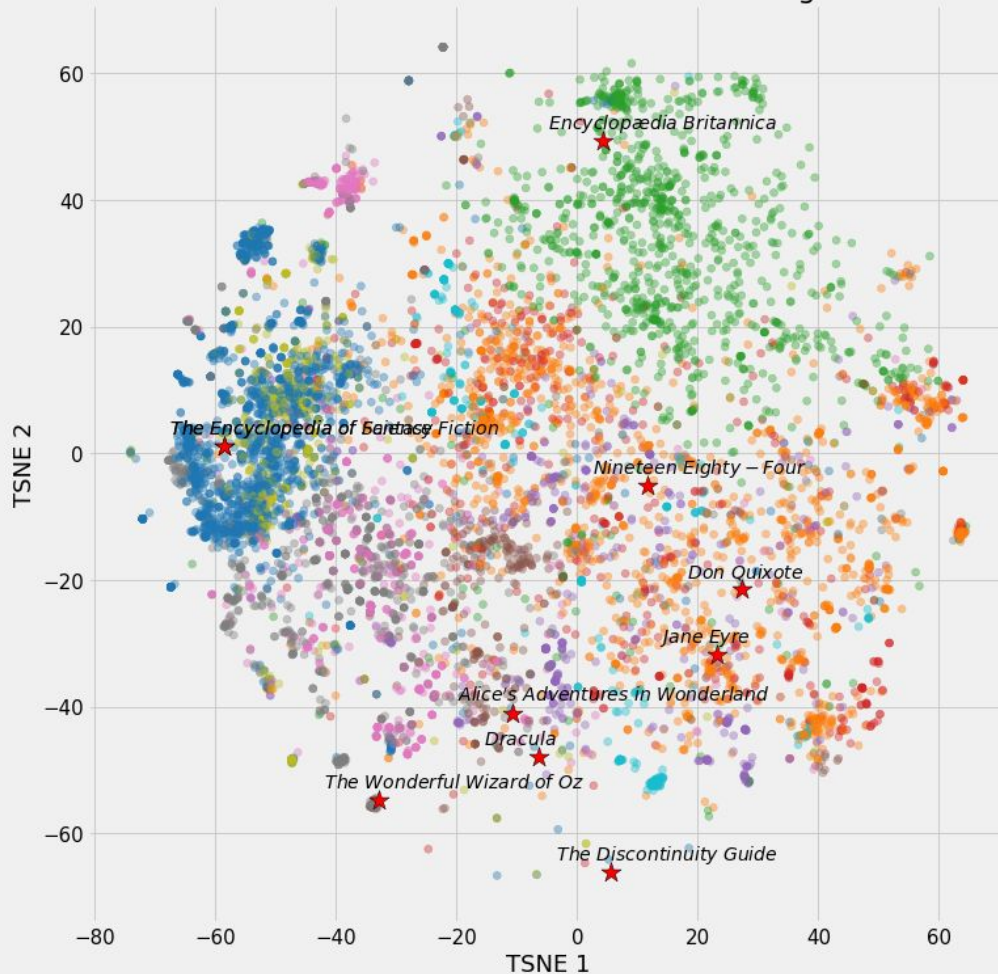
These embeddings vectors 'map' the words.

Similar words are close to each other

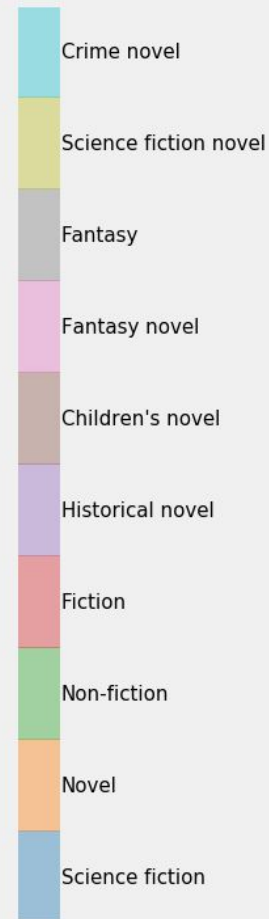
Seems simple when mapping individual words.
But embeddings also scale...



TSNE Visualization of Book Embeddings



Genre

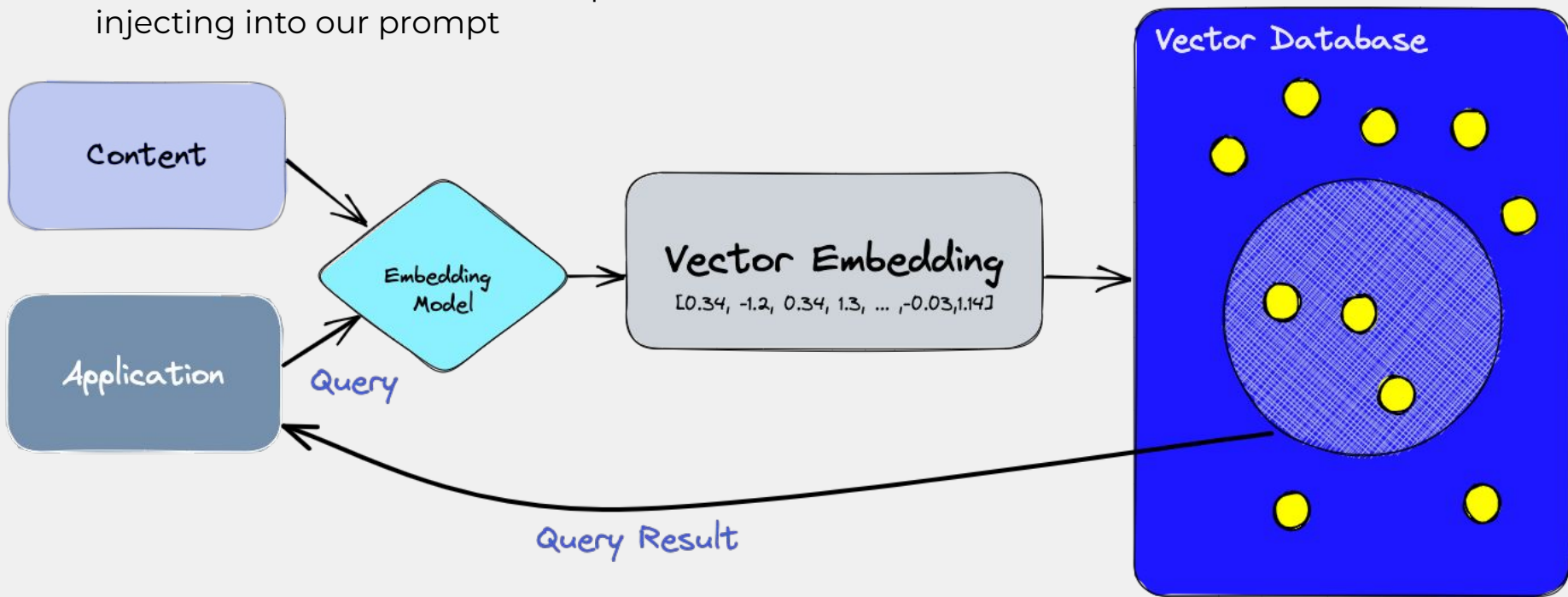


We can embed entire books and see how 'similar' they are!

These embeddings have been mapped to two dimensions using TSNE

Now we can store and retrieve our data!

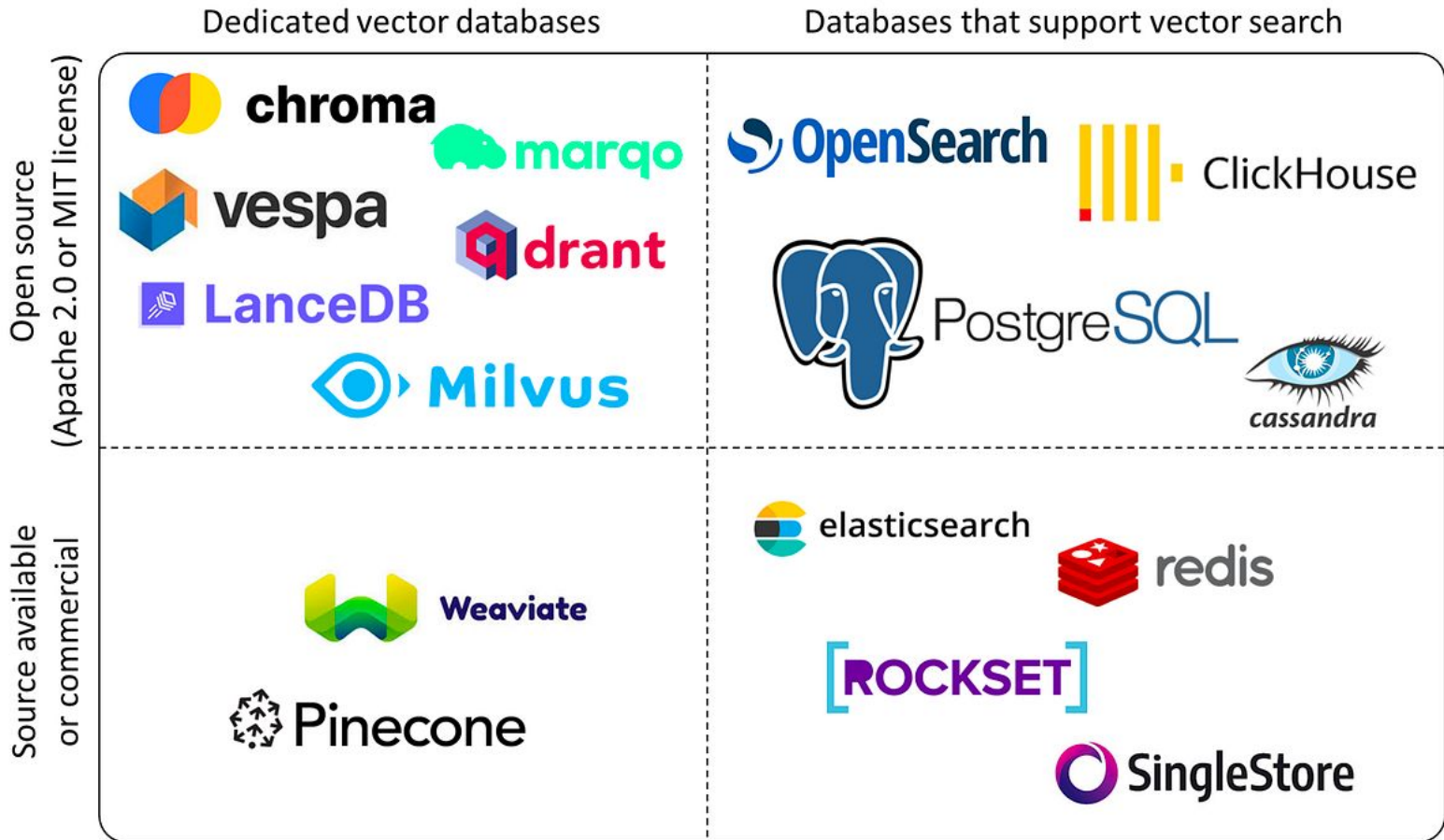
1. Store lots of chunks of text as embeddings
2. Take your query and search it's embedded value in your database
3. Get back a list of the most comparable results for injecting into our prompt



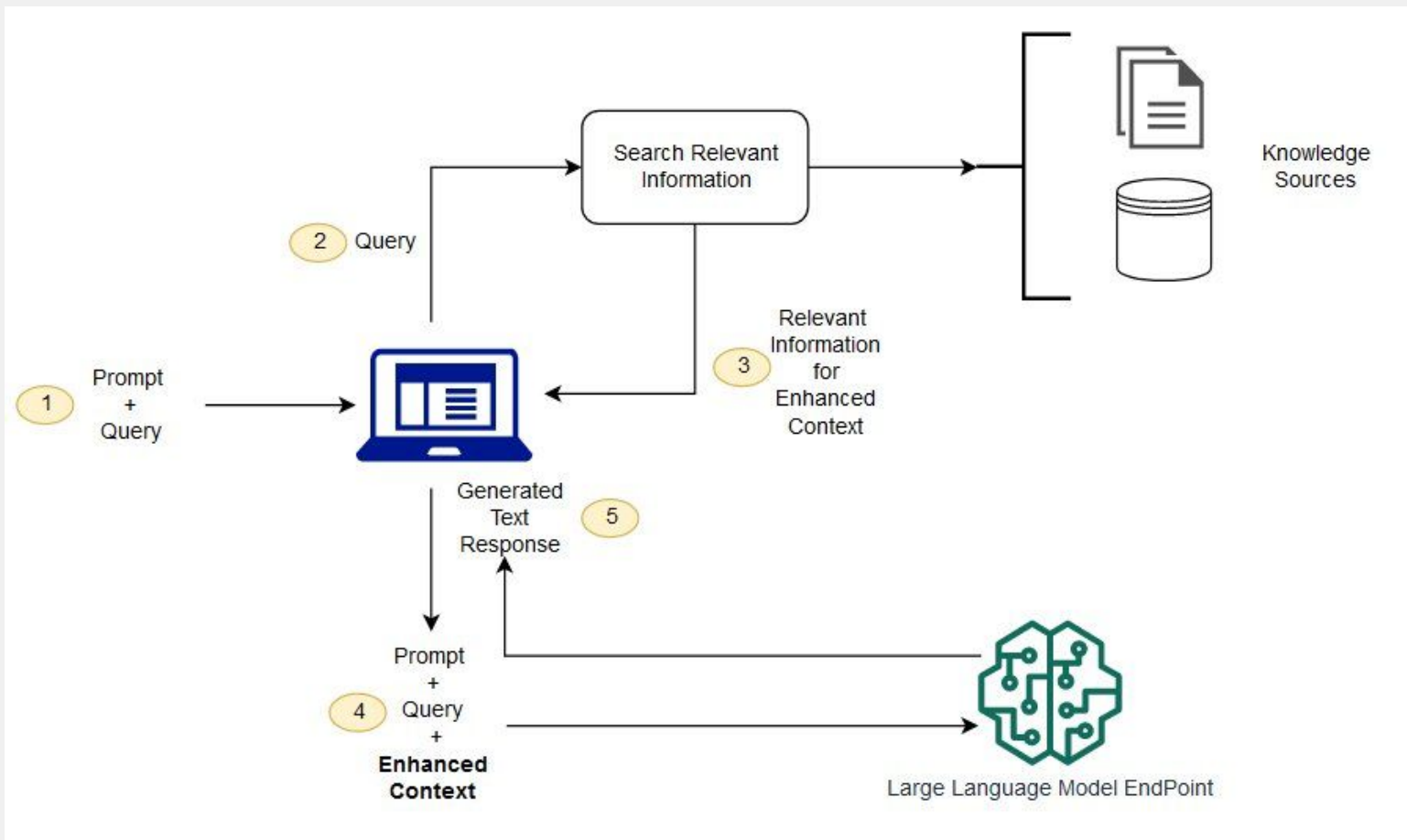
What can I use for a vector database?

Are there many options available for me to build such a thing?

Yes!



Putting Together the Whole Picture



See RAG in action (courtesy of Pinecone)

<https://colab.research.google.com/github/pinecone-io/examples/blob/master/learn/generation/langchain/handbook/05-langchain-retrieval-augmentation.ipynb#scrollTo=DiRWzKh0mMGv>

Tips for setup

1. Add `!pip install --upgrade datasets multiprocessing` to end of first box before running anything
2. Have an OpenAI API key handy (will cost a couple cents to use)
<https://platform.openai.com/account/api-keys>
3. Have a pinecone API key handy (free trial) pinecone.io
4. Indexing step takes about 5 minutes


```
[24] 1 from langchain.chains import RetrievalQAWithSourcesChain
      2
      3 qa_with_sources = RetrievalQAWithSourcesChain.from_chain_type(
      4     llm=llm,
      5     chain_type="stuff",
      6     retriever=vectorstore.as_retriever()
      7 )
```

```
4s 1 qa_with_sources(query)
```

```
{'question': 'who was Benito Mussolini?',
  'answer': "Benito Mussolini was an Italian politician and journalist who served as the Prime Minister of Italy from 1922 until 1943. He was the leader of the National Fascist Party and played a significant role in the rise of fascism in Italy. Mussolini sought to establish Italy as a new Roman Empire and pursued expansionist policies, including the occupation of Abyssinia (Ethiopia) and Albania. He formed an alliance with Adolf Hitler and entered World War II as part of the Axis Powers. However, Mussolini's regime eventually fell, and he was captured and executed by partisans in 1945. \n",
  'sources': 'https://simple.wikipedia.org/wiki/Benito%20Mussolini'}
```

Now we answer the question being asked, *and* return the source of this information being used by the LLM.

Thank you!

Any questions,
comments, and/or
concerns?