RAGe Against the keyword search

Extending GPT capabilities with RAG

www

who am i?

I'm a developer and Machine Learning Engineer

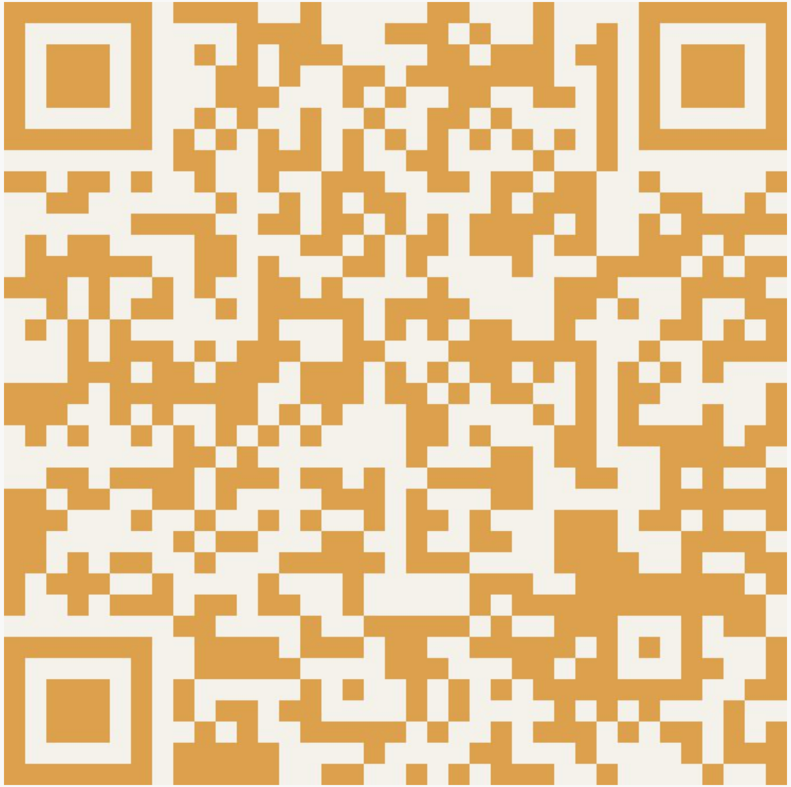
- **Data Science**: My career started with the title "Data Scientist." My capstone project in school was an RNN that predicted words, given the ending of words. I was hoping to do something like autocomplete for lawyers.
- **Machine Learning Engineer**: Through most of my career, mostly because I was mediocre at math, I gravitated towards engineering problems. I've been responsible for a number of applications that had some solid gravity.
- **AI Stuff**: The reason I'm here today is because of a friend who introduced me to Word2Vec during school. From I've always been interested in this way that we represent words with numbers and vectors.

how to get a bunch out of this preso

I talk fast and I speak mumble fluently. Call me on it.

Fonts	This is normal thoughts. `This is code font.`You can copy paste code font into apps
Notes	Take notes but the slides are there for you
Code	This is all in a GitHub repo
These slides	These slides are available. Next slide has a QR code
Should I code along?	Depending on your skill level, this may be the first time with these words, just let the words wash over you. I swear you could turn any of these slides into a thesis paper, so don't feel like you're ducking the wave if you gloss over something. Come back later. To this end, I've come up with a number system throughout. The next slide will explain more.
Contact?	For sure. I live here. Email me. Catch me in Safeway. I'll be around. I'll flash my email at the end, and you should 1000% contact me a lot. I'm also on the Hawaii Slack.

These slides are here:



meeting you where you’re at

1

“I have been impressed by AI and the use cases. I’m here to understand how I can place AI in a lattice of my understanding of tech. I want to employ AI in my business, but I need really solid use-cases.” OR “I’m here because tech is rad and I want to see Hawaii flourish as a place where bleeding edge tech happens.”

2

I have heard the term RAG before and I generally get that it’s an AI thing, but I am here to learn more about the theory behind it, I don’t really care about the code.

3

"I’m on AI LinkedIn / twitter | X / Substack daily and I have heard a lot about RAG RAG. I kiiinda code already and I’m here to learn more about RAG’s implementation at lower levels, i.e. with technical architecture."

4

"I’ve heard about RAG and want to see some code - I’m going to be responsible for implementing RAG architecture applications and I’m here to directly learn something to make me great at my job. Where are the XKCD references.”

table of contents

{01}

Introduction

{02}

Understanding RAG

{03}

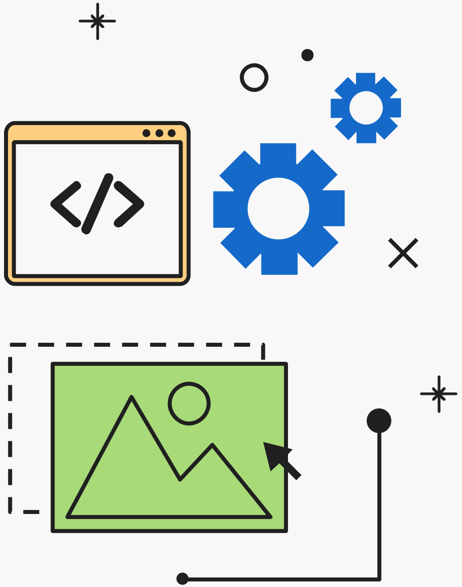
Key Tech Behind RAG

{04}

Implementing RAG w/ Python

{05}

Conclusion | Q&A



objectives

1 Understand the concept and importance of Retrieval-Augmented Generation in NLP.

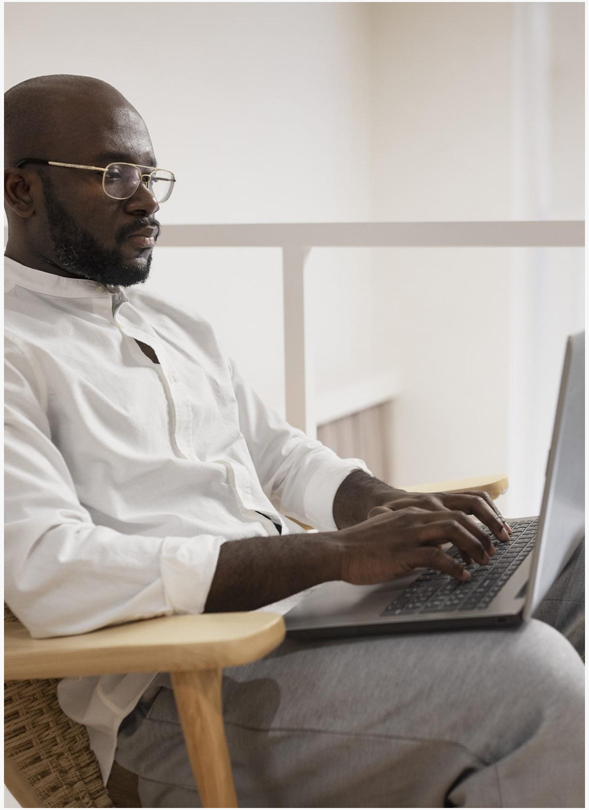
- Describe the evolution and basic concepts of NLP leading up to the development of RAG.
- Define Retrieval-Augmented Generation and its benefits over traditional generative models.

2 Identify the components and workflow of a RAG system.

- Outline the architecture of RAG including its retrieval and generation components.

3 Implement a basic RAG system using Python and associated libraries.

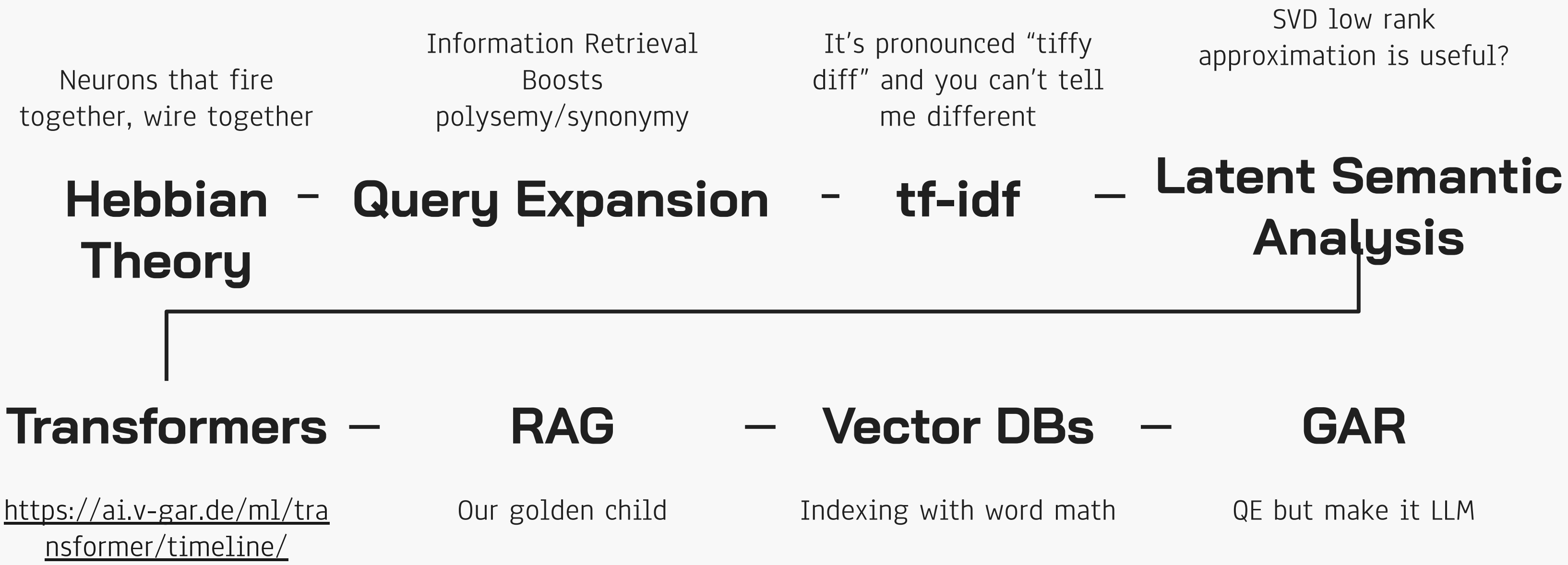
- Execute a basic implementation of RAG using Python libraries like Hugging Face Transformers and datasets.



{02}

Understanding RAG

Timeline of RAG



Neurons that fire together, wire together

Hebbian Theory –

Information Retrieval Boosts polysemy/synonymy

Query Expansion -

It's pronounced "tiffy diff" and you can't tell me different

tf-idf –

SVD low rank approximation is useful?

Latent Semantic Analysis

Transformers –

https://ai.v-gar.de/ml/transformer/timeline/

RAG –

Our golden child

Vector DBs –

Indexing with word math

GAR

QE but make it LLM

Benefits of RAG

Unseen Data

LLMs are always going to have cutoff dates, and it may benefit our system to have more up to date data. However, more importantly, we may also have proprietary data.

Unlimited Context

Context windows are like your brain's ability to hold a certain N of memories.

Embeddings… so hot rn

Embeddings are used widely throughout machine learning. LLM based embeddings are incredibly information dense. Embeddings are the cheapest recommender system ever

Benefits of RAG (cont’d)

Hallucinations

Hallucinations are a natural part of using LLMs with our synonym problems

Semantic Search

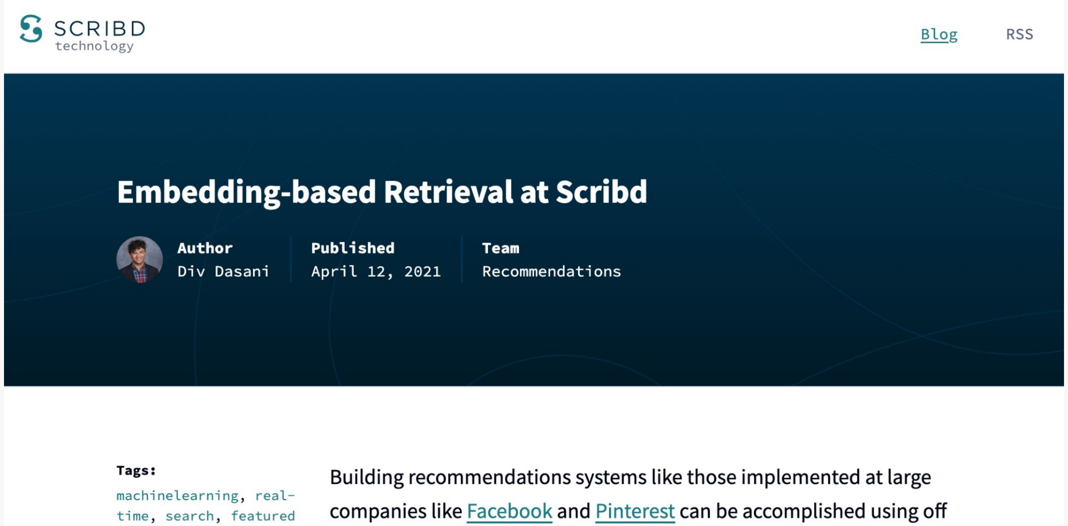
Super powered Ctrl + F

Today this 👆 is going to be our focus, but we’ll also hit on RAG + Chat

Use Cases

Embeddings for document retrieval

Embeddings are used at Scribd to find and recommend similar content



SCRIBD technology

Blog RSS

Embedding-based Retrieval at Scribd

Author
Div Dasani

Published
April 12, 2021

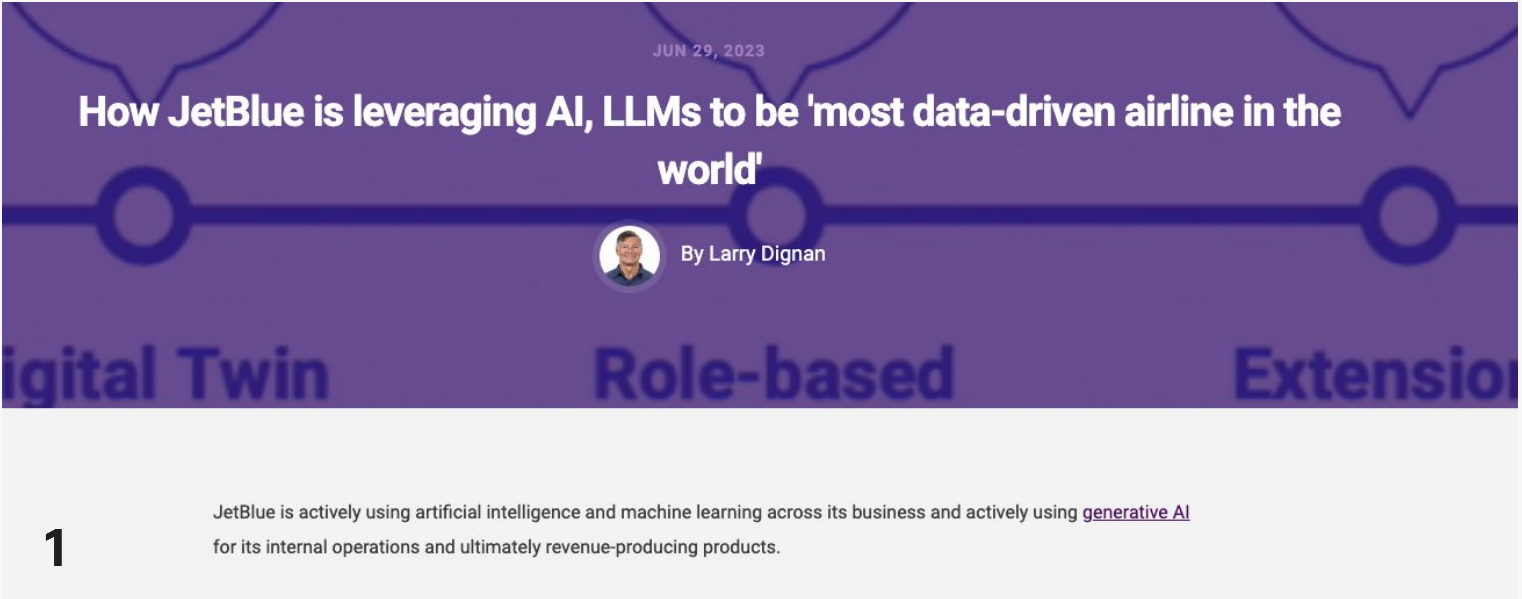
Team
Recommendations

Tags:
machinelearning, real-time, search, featured

Building recommendations systems like those implemented at large companies like Facebook and Pinterest can be accomplished using off

Use Cases

Better Chatbots



JUN 29, 2023

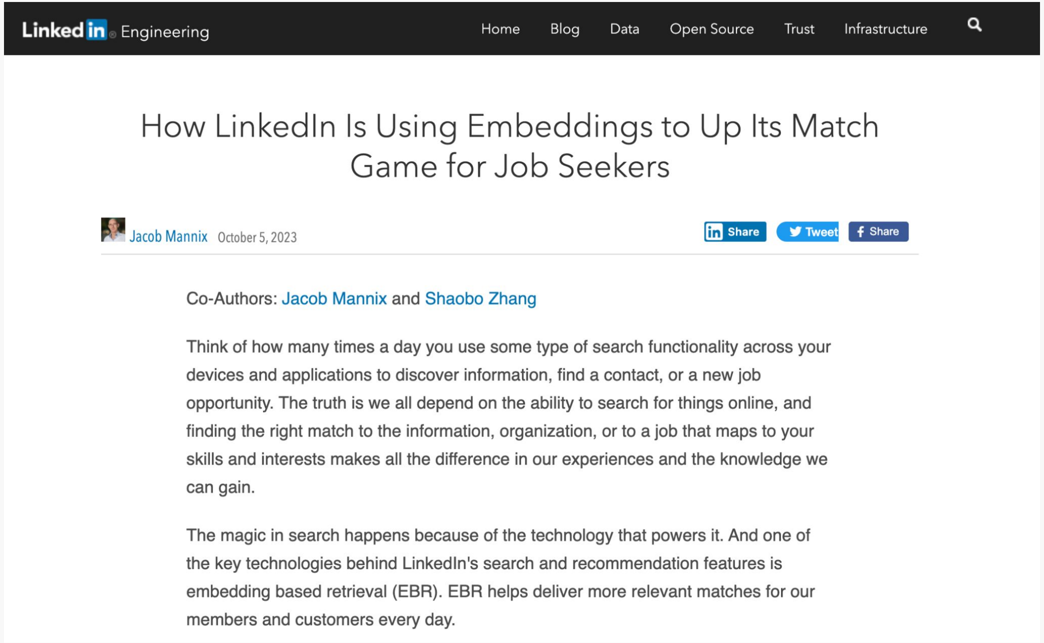
How JetBlue is leveraging AI, LLMs to be 'most data-driven airline in the world'

By Larry Dignan

JetBlue is actively using artificial intelligence and machine learning across its business and actively using generative AI for its internal operations and ultimately revenue-producing products.

Use Cases

Recommenders



How LinkedIn Is Using Embeddings to Up Its Match Game for Job Seekers

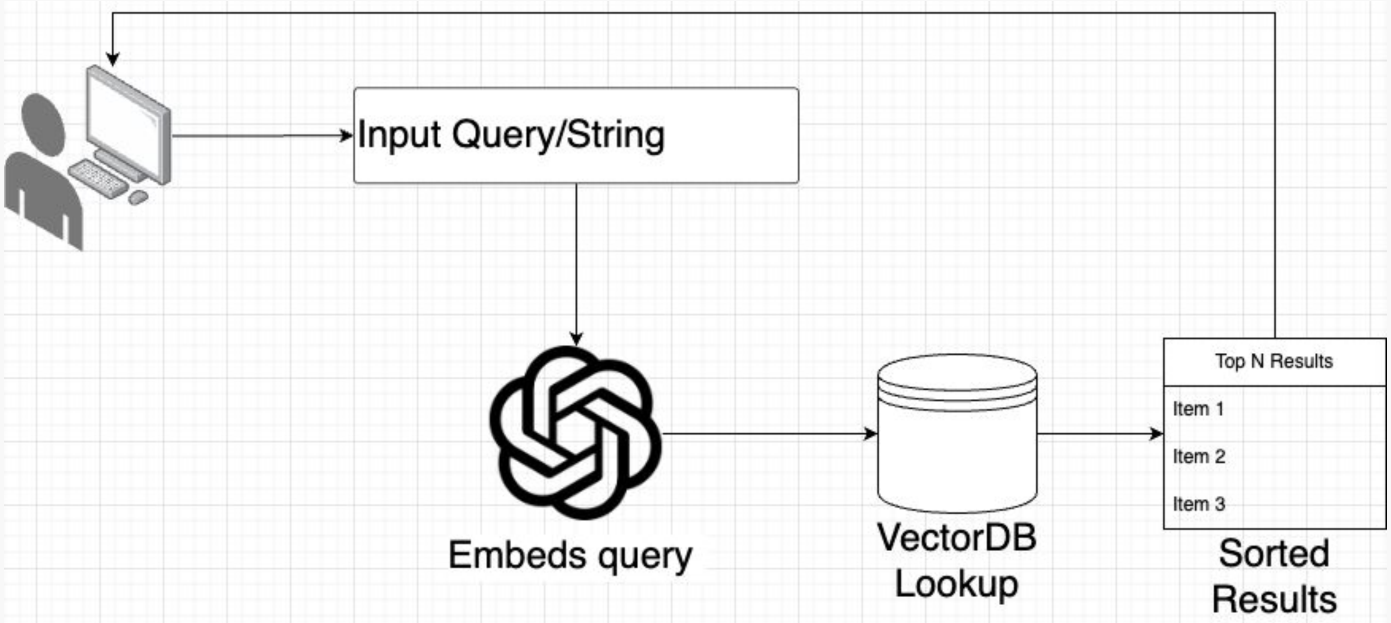
Jacob Mannix October 5, 2023

Co-Authors: Jacob Mannix and Shaobo Zhang

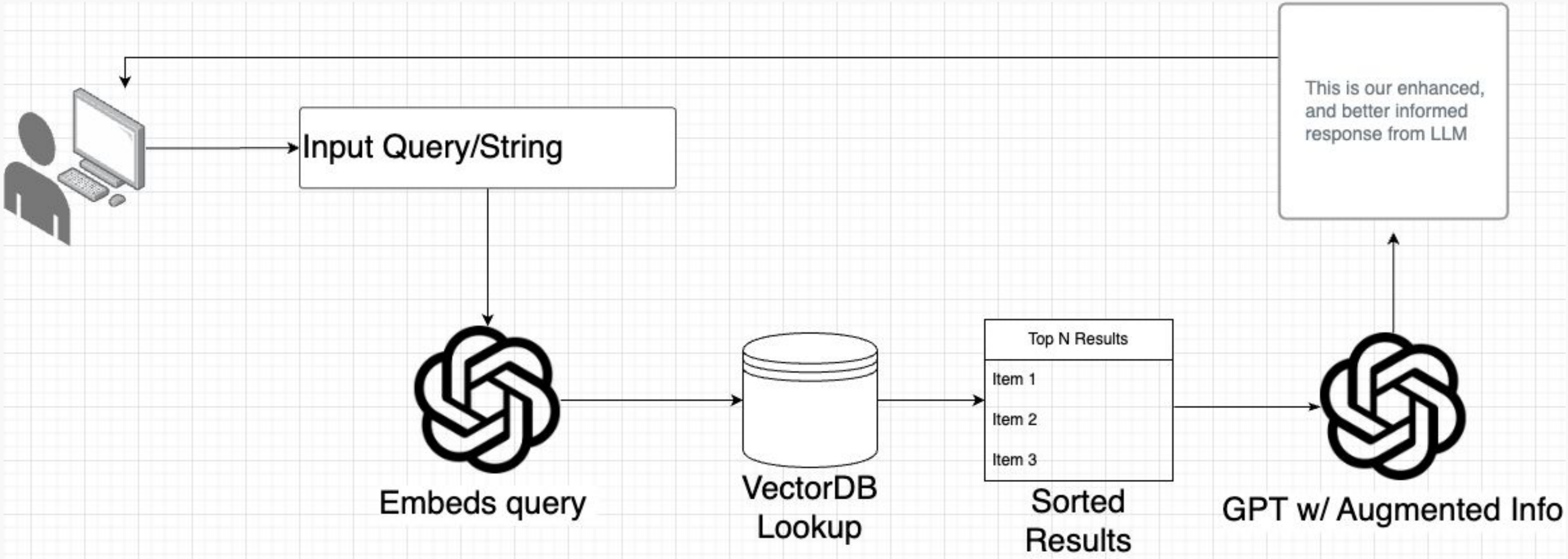
Think of how many times a day you use some type of search functionality across your devices and applications to discover information, find a contact, or a new job opportunity. The truth is we all depend on the ability to search for things online, and finding the right match to the information, organization, or to a job that maps to your skills and interests makes all the difference in our experiences and the knowledge we can gain.

The magic in search happens because of the technology that powers it. And one of the key technologies behind LinkedIn's search and recommendation features is embedding based retrieval (EBR). EBR helps deliver more relevant matches for our members and customers every day.

Simple Embedding Retrieval



Simple RAG Chat



{03}

Key Tech

Mathy Things

Starting from scratch. There's essentially two components and both are the fundamental blocks of RAG. At lowest levels, we're doing some vector math with inputs

Proximity Search

Cosine Similarity	**Fairly popular**	• Similarity • Measure distance
Nearest Neighbors	**Great scale**	• K-nearest or ANN • Used in prod
BM25	**Best Matching 25**	• Probabilistic • Used in traditional systems

Embeddings

Transformer

- Large vectors based on LLMs
- Predominantly created with API calls

Word2Vec

- First breakthroughs in embeddings
- Vectorize words

Traditional like SVD or PCA

- Sparse matrix work
- Well understood

Vector Databases

Scalable Lookups

Managed SaaS

Pinecone

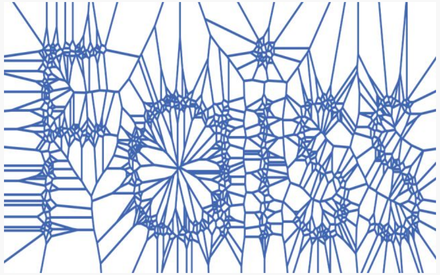
Weaviate

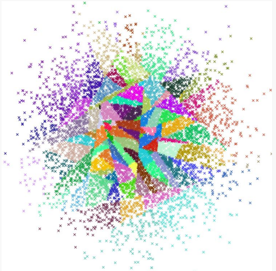
Open Source DB

Chroma

PostgreSQL

Open Source Vector Lookup





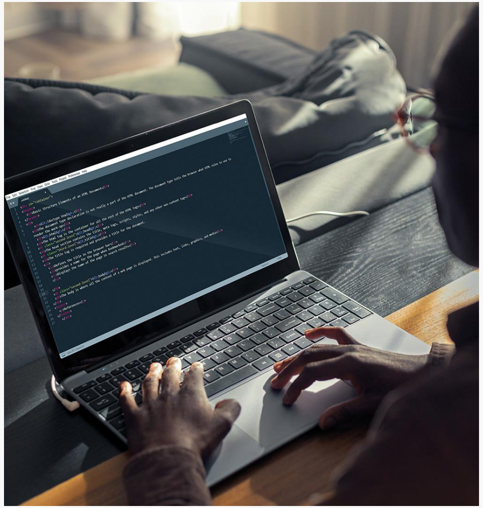
Python Libraries

More bells and whistles

🤗 Hugging Face

🦜🔗 LangChain

LlamaIndex



{04}

Implementing RAG

What’re we going to do?

Embedding retrieval

We’re going to first show embeddings can be used to retrieve semantically similar words/sentences.

RAG Chatbot

Then we’re going to show a simple Chat w/ GPT that includes documents not in GPT’s training base.

Pseudo Code

Embed Stuff:

```
def user_input(some_text):
    return openAI.get_embeddings(some_text)

my_sentence = user_input(“foo bar baz jumps the quick fox”)
```

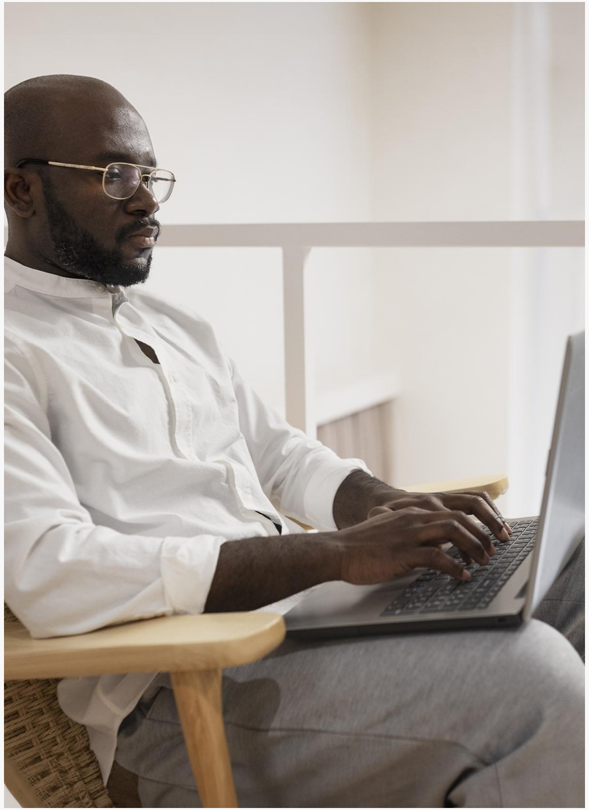
Cosine Similarity Stuff:

```
def cos_sin(embedded_text, stuff_in_vector_db):
    return openAI.cosine_similarty(embedded_text, stuff_in_vector_db, return_top_n=5)

Most_similar_docs = cos_sin(my_sentence, stuff_in_vector_db, 5)
```

Rank Stuff:

```
def rank(docs):
    return sorted(docs)
my_ranked_docs = rank(most_similar_docs)
```



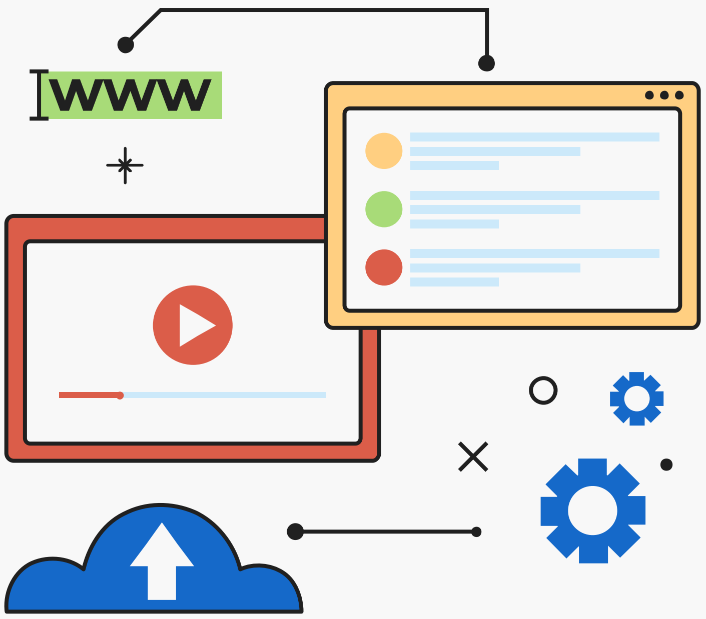
{05}

Conclusion + Q&A

Thanks!

Do you have any questions about RAG and/or think your company’s employee onboarding and training could be better?

kevin@chelle.ai



www.chelle.ai

Appendix

Stuff I said I'd fact check on, or provide links for:

- This is the article I referenced in answer to the question about reranking: https://postgresml.org/blog/personalize-embedding-results-with-application-data-in-your-database
- Additionally, here's some solid articles about this architecture of "candidate generation, followed by re-ranking:
  - ReRanking models / Learning to rank models on top of RAG systems: https://www.pinecone.io/learn/series/rag/rerankers/
  - Using Vector Databases/Embeddings alongside traditional recommender models like collaborative filtering: https://www.emno.io/posts/vector-databases-in-movie-recommenders
- How embeddings are created: https://arxiv.org/pdf/2201.10005
  - Creating your own from a foundation model: https://dagshub.com/blog/how-to-train-a-custom-llm-embedding-model/
  - What embeddings/vectors are, a deeper dive: https://www.youtube.com/watch?v=WumStBfoArc
  - Embeddings from scratch: https://towardsdatascience.com/contextual-transformer-embeddings-using-self-attention-explained-with-diagrams-and-python-code-d7a9f0f4d94e#cb34

References

- Original RAG paper: https://arxiv.org/pdf/2005.11401
- Chat with text
- Transformer timeline: https://ai.v-gar.de/ml/transformer/timeline/
- LSA: https://en.wikipedia.org/wiki/Latent_semantic_analysis
- Tf-idf: https://en.wikipedia.org/wiki/Tf%E2%80%93idf
- Survey paper on RAG: https://arxiv.org/pdf/2405.07437
- Generation Augmented Retrieval: https://aclanthology.org/2021.acl-long.316.pdf
- Query Expansion paper: https://arxiv.org/pdf/1708.00247
- Query Expansion from class: https://nlp.stanford.edu/IR-book/html/htmledition/query-expansion-1.html