Embeddings and Encodings

Preprocessing for LLM Fine Tuning

Attention is all you need

Bio

Name : Jayanthi Suryanarayana

https://www.linkedin.com/in/jayanthi-suryanarayana-2313841/

- Principal AI engineer with expertise in building transactional, data, ML engineering, LLM enterprise solutions.
- Curious and constantly learning in this fast evolving AI space.
- Hold an electronics engineering degree.

Built data platforms for enterprise, Passionate about the use of **synthetic data** to enable people who work with data (analysts, engineers, scientists anyone who wants to get value out of data)

Activities I enjoy:
Time with family, cooking and practicing yoga

Agenda

- Attention Here Please- Solve some math problems
- AI - Historical understanding
- Default algorithm and architecture
- Concepts - Embedding and Encodings
- Development paradigms
- Framework - Emerging architecture for LLM based solutions
- SENTENCE TRANSFORMER
- Questions
- Hands on - Build a Question Answer Application

High School Math Student

Cosine similarity and Dot Product

Homework - Math Problem Set 9th - Dot product

Find the dot product between two vectors:

Problem #1 : Vec A = [3,4,5]   Vec B = [3,4,5]

Problem #2 : Vec A = [3,4,5] Vec B = [5,5,-7]

Hint : Use this calculator : https://www.omnicalculator.com/math/dot-product

Homework - Math Problem Set 9th - Dot product

Find the dot product between two vectors:

Problem #1 : Vec A = [3,4,5]   Vec B = [3,4,5]

3*3 + 4*4 + 5*5 = 9+16+25 = 50 hint : angle between the vector = 0

Problem #2 : Vec A = [3,4,5] Vec B = [5,5,-7]

3*5+4*5+5*-7=15+20-35 = 0 hint : Orthogonal vectors

Hint : Use this calculator : https://www.omnicalculator.com/math/dot-product

Homework - Math Problem Set - Cosine similarity

Find the cosine similarity between two vectors: a.b/|a|*|b|

Problem #1 : Vec A = [3,4,5]   Vec B = [5,5,-7]

Problem #2 : Vec A = [3,4,5] Vec B = [-3,-4,-5]

Problem #3 : Vec A = [3,4,5] Vec B = [3,4,5]

Hint : Use this calculator : https://www.omnicalculator.com/math/cosine-similarity

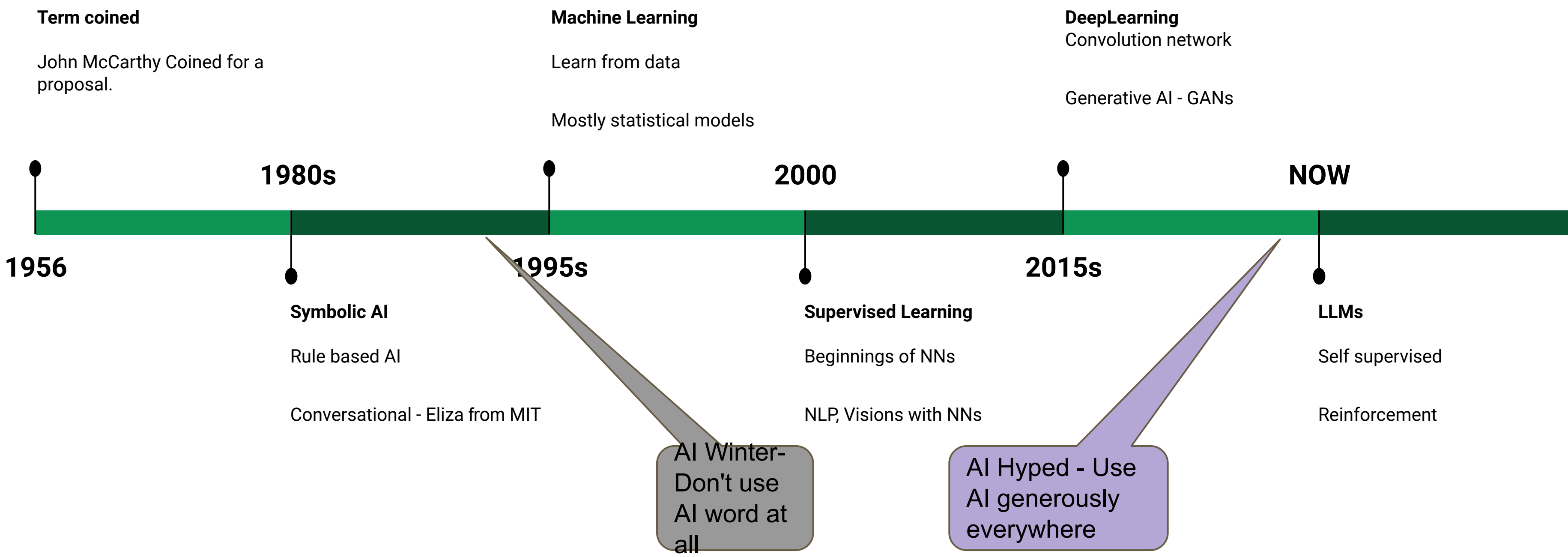
Ans : 0,-1,1

Curious Grandma

What is chatGPT and AI

credit/disclaimer : Image generated with Duet AI

AI - when did it start - a timeline view



1956

Term coined

John McCarthy Coined for a proposal.

1980s

Symbolic AI

Rule based AI

Conversational - Eliza from MIT

1995s

Machine Learning

Learn from data

Mostly statistical models

AI Winter- Don't use AI word at all

2000

Supervised Learning

Beginnings of NNs

NLP, Visions with NNs

2015s

DeepLearning
Convolution network

Generative AI - GANs

NOW

LLMs

Self supervised

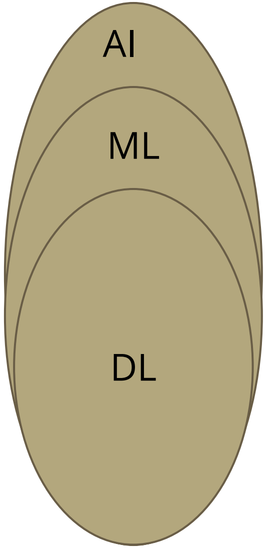
Reinforcement

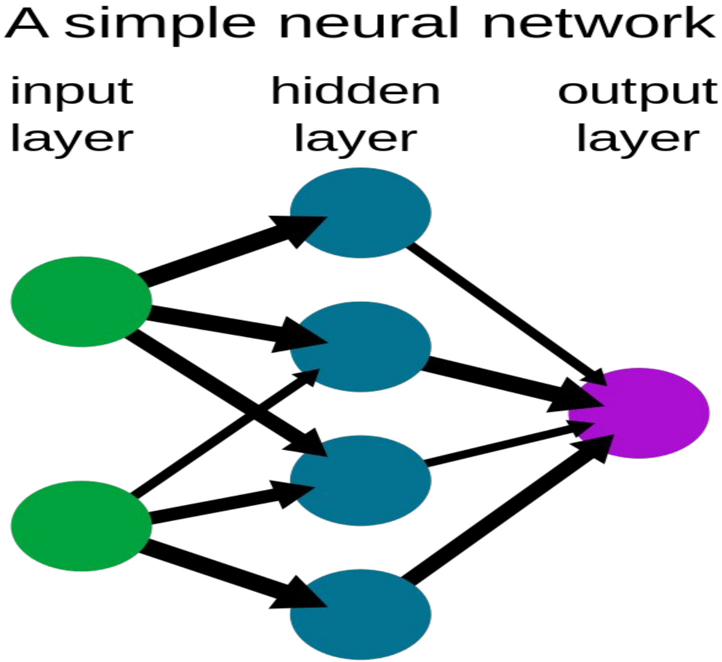
AI Hyped - Use AI generously everywhere

State of the art….

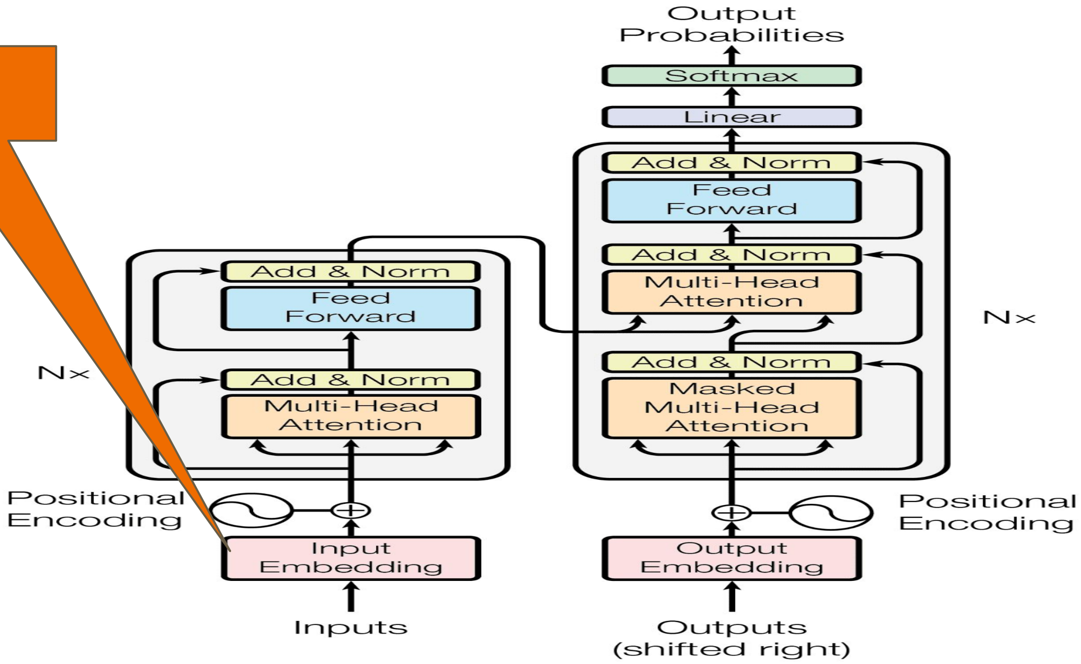
Nonlinear function and Gen AI architecture

Default Algorithm - Nonlinear Function approximation





Transformer architecture : Attention is all you need



Our Focus Today

Embedding vs Encoding

Embeddings are the A.I-native way to represent any kind of data, making them the perfect fit for working with all kinds of A.I-powered tools and algorithms. They can represent text, images, and soon audio and video. There are many options for creating embeddings, whether locally using an installed library, or by calling an API.

A.I Native way = Numbers

the vector representation in the high school math.

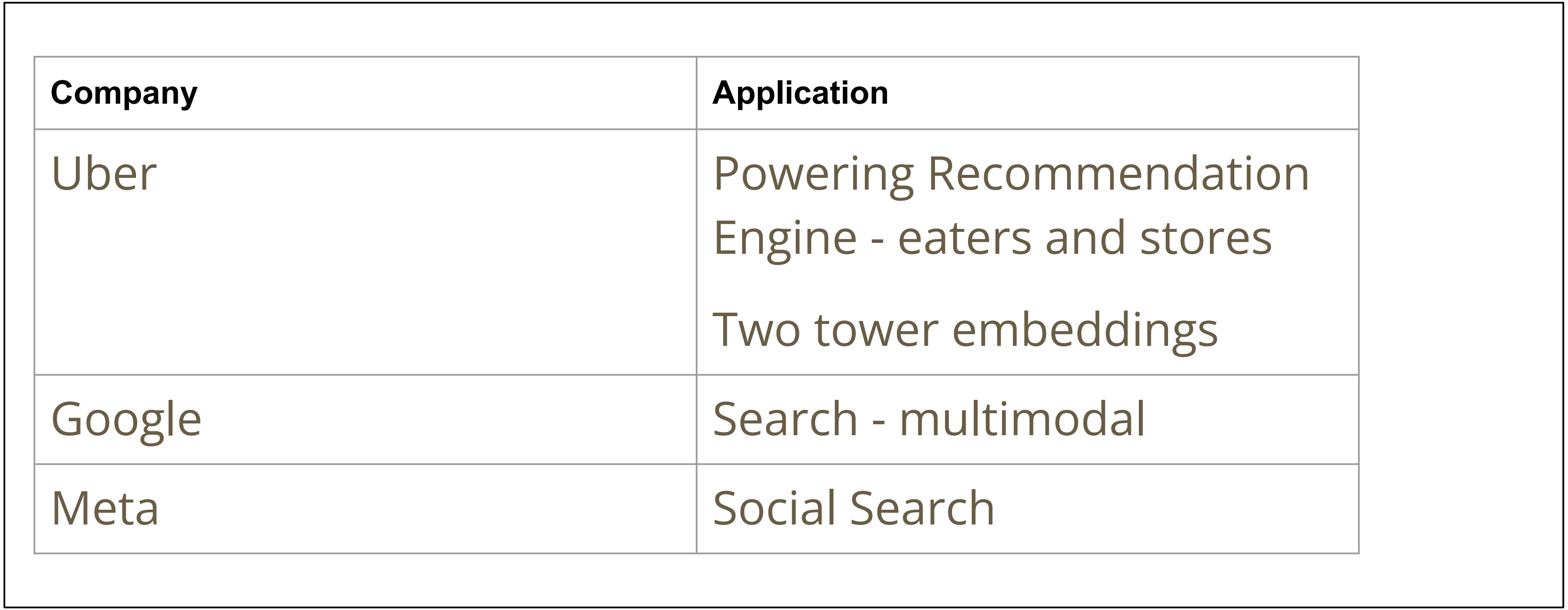
Encoding

Transformation but has reverse process to decode

e.g. one hot encoding

Sometimes people use interchangeably, but one should understand the context

Industrial strength Embeddings Applications



Company	**Application**
Uber	Powering Recommendation Engine - eaters and stores Two tower embeddings
Google	Search - multimodal
Meta	Social Search

AI Tasks

Information retrieval

Clustering

Classification, requiring minimal additional feature engineering

Semantic Search, nn search

Multi modal applications

RAGs - Retrieval Augmented Generation

Multimodal approach convergence

Vision - CNN

Language - RNN era then transformer era

Default algorithm for any function approximation - Deep Learning

Default Generative architecture - Transformer architecture

What is common : Learning representation as number for the algorithm to process

Key Takeaways thus far

- Attention is all you need :
    - Common Architecture and Common Compute framework
- Multi model convergence:

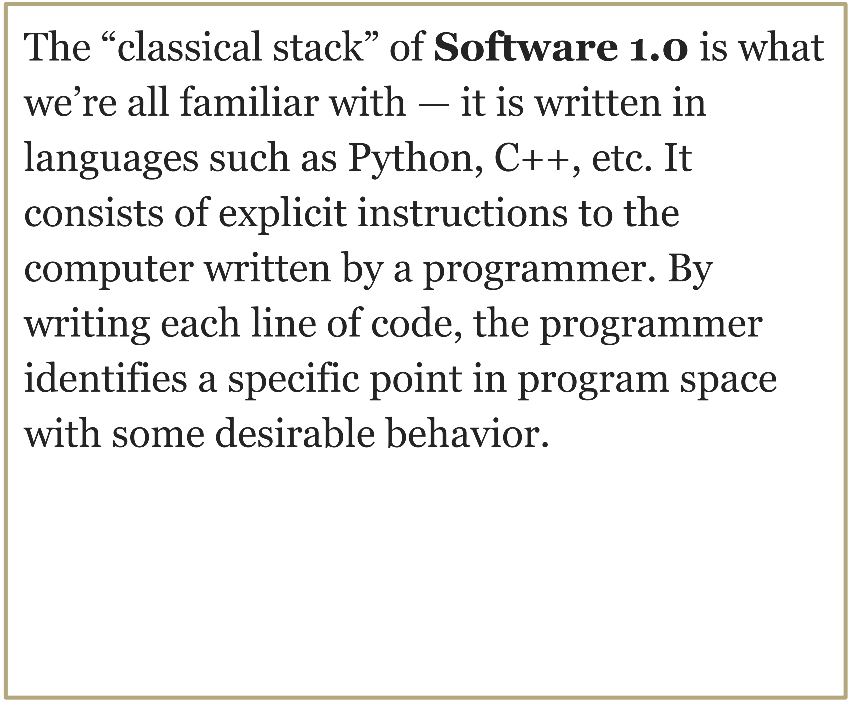
  Good learning representation of real world in numbers - EMBEDDINGS

- New representation and new way to retrieve, need to build the ecosystem

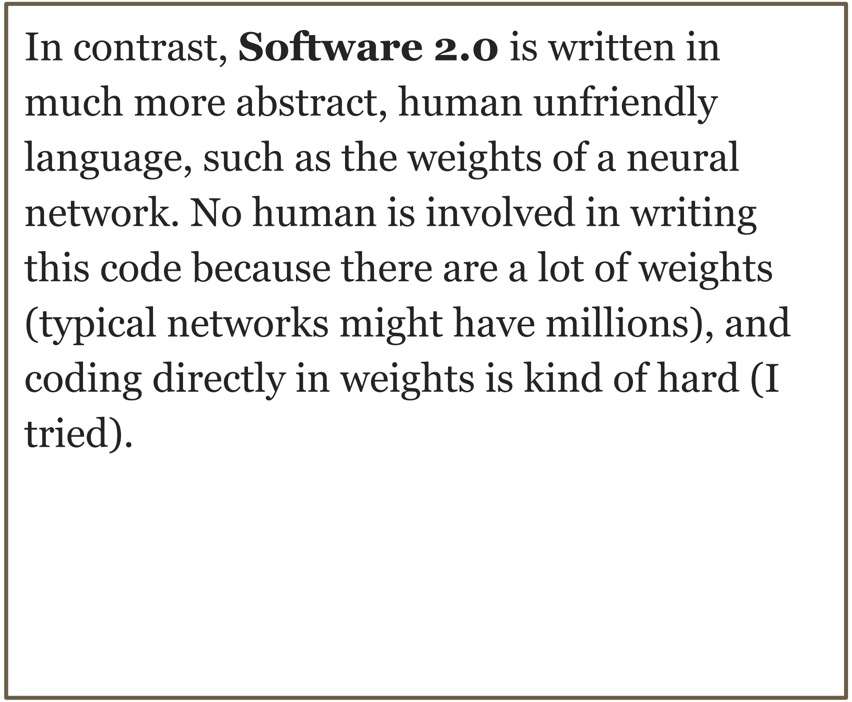
Developer

understand evolving paradigms

Programming Paradigms - Software 1.0 and 2.0



The “classical stack” of **Software 1.0** is what we’re all familiar with — it is written in languages such as Python, C++, etc. It consists of explicit instructions to the computer written by a programmer. By writing each line of code, the programmer identifies a specific point in program space with some desirable behavior.



In contrast, **Software 2.0** is written in much more abstract, human unfriendly language, such as the weights of a neural network. No human is involved in writing this code because there are a lot of weights (typical networks might have millions), and coding directly in weights is kind of hard (I tried).

https://karpathy.medium.com/software-2-0-a64152b37c35

Programming Paradigm Software 2.0

Andrej Karpathy:

www.youtube.com/watch?v=y57wwucbXR8

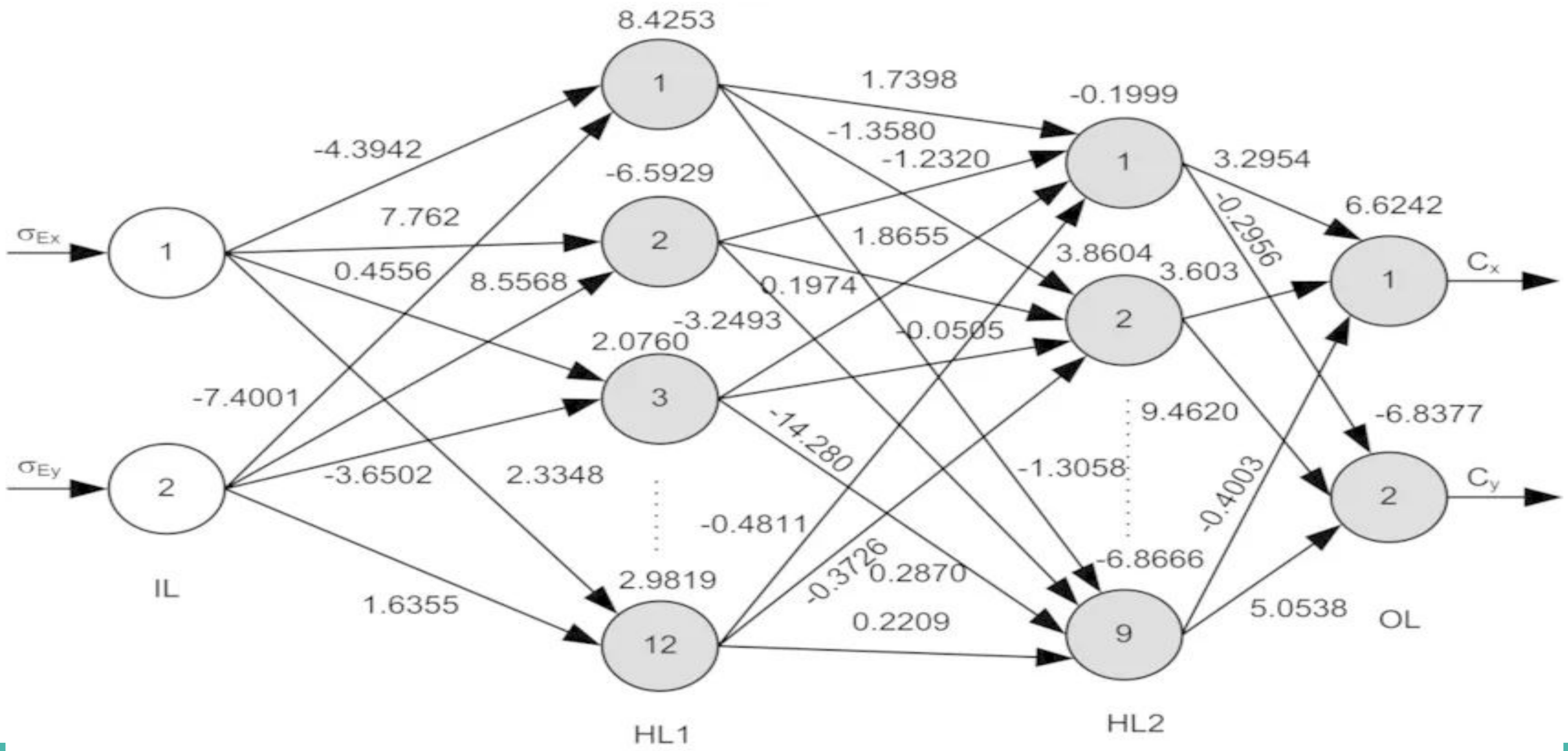
https://karpathy.medium.com/software-2-0-a64152b37c35

Software 1.0 : Explicit - Code only

Software 2.0: Abstract - Code + Data (Machine Learning mostly deep learning)

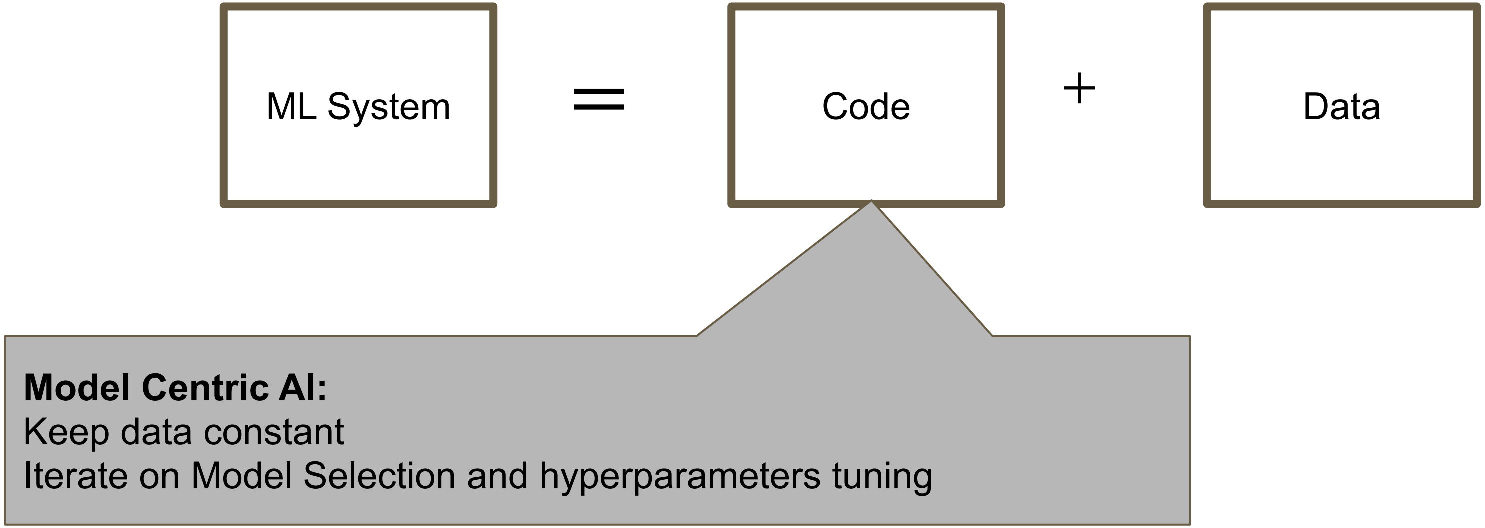
most of the active “software development” takes the form of curating, growing, massaging and cleaning labeled datasets. This is fundamentally altering the programming paradigm

2.0



σEx
σEy
1
2
IL
8.4253
-6.5929
2.0760
2.9819
1
2
3
12
HL1
-4.3942
7.762
0.4556
8.5568
-7.4001
-3.6502
2.3348
1.6355
1.7398
-1.3580
-1.2320
1.8655
0.1974
-3.2493
-0.0505
-14.280
-0.4811
-0.3726
0.2870
0.2209
-0.1999
3.8604
9.4620
-6.8666
1
2
9
HL2
3.2954
-0.2956
3.603
-1.3058
-0.4003
5.0538
6.6242
-6.8377
1
2
OL
Cx
Cy

Machine Learning Systems - Software 2.0



ML System = Code + Data

Model Centric AI:
Keep data constant
Iterate on Model Selection and hyperparameters tuning

Data Centric AI

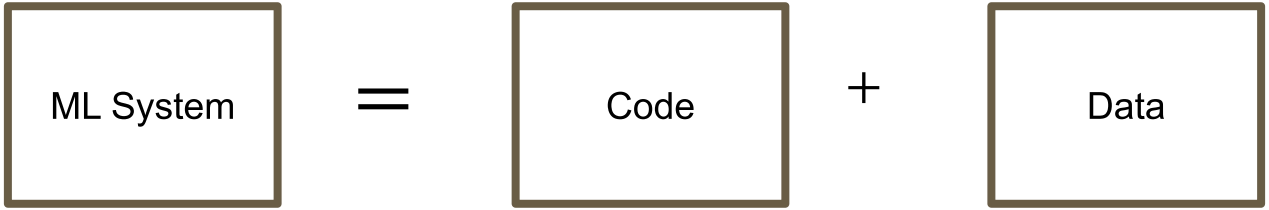
Andrew Ng:

Momentum Since couple of years

Data-Centric AI is the discipline of systematically engineering the data used to build an AI system.

https://landing.ai/data-centric-ai/#:~:text=Data%2DCentric%20AI%20is%20the,on%20data%20instead%20of%20code.

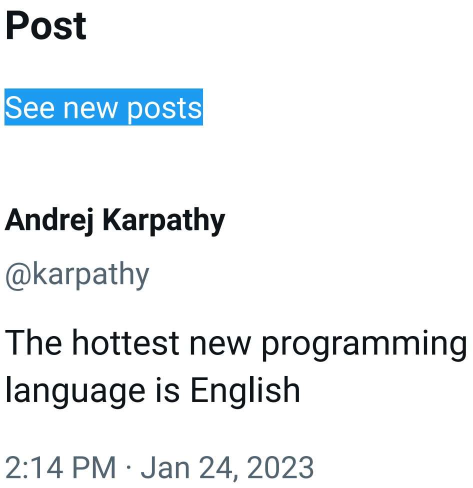
Data Centric AI - Software 2.0



Data Centric AI:

- ❏ **Systematic approach** understand/update data to improve ML System performance
- ❏ Use machine learning approaches, techniques

New Programming Language -- English is that all?



Post

See new posts

Andrej Karpathy

@karpathy

The hottest new programming language is English

2:14 PM · Jan 24, 2023

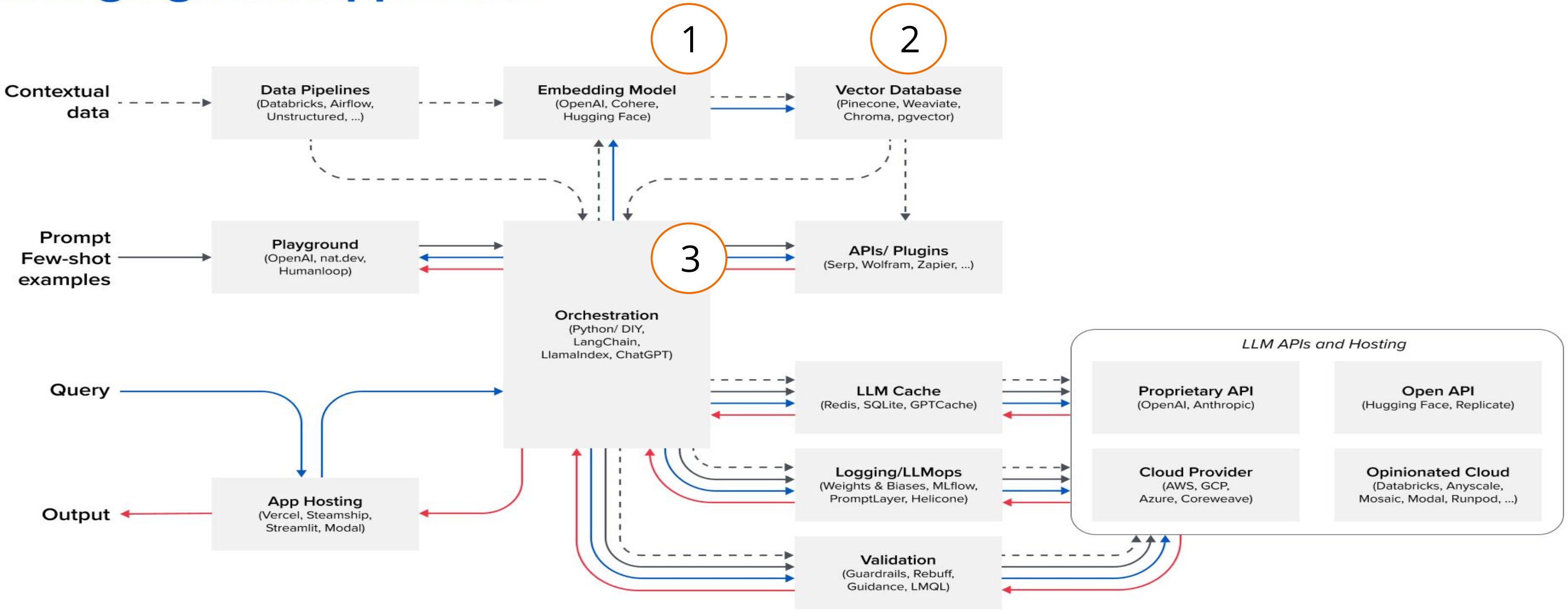
Software Enterprise Development - Software 1.0 - 2.x

Application Category	**Paradigm**	**Ownership**
Transactional/Fullstack	1.0	Code
Traditional ML apps	2.0	Code and Data (Train and inference time)
Prompt Engineered	English	Inference time data
RAG	2.0 for embedding 1.0 for CRUD of embeddings	New store, building Inference time data
Fine Tuning	2.x	Train time data

AI Engineer

building state of the art AI applications

Emerging LLM App Stack



1

2

3

Contextual data

Prompt Few-shot examples

Query

Output

Data Pipelines (Databricks, Airflow, Unstructured, ...)

Embedding Model (OpenAI, Cohere, Hugging Face)

Vector Database (Pinecone, Weaviate, Chroma, pgvector)

Playground (OpenAI, nat.dev, Humanloop)

Orchestration (Python/ DIY, LangChain, LlamaIndex, ChatGPT)

APIs/ Plugins (Serp, Wolfram, Zapier, ...)

App Hosting (Vercel, Steamship, Streamlit, Modal)

LLM Cache (Redis, SQLite, GPTCache)

Logging/LLMops (Weights & Biases, MLflow, PromptLayer, Helicone)

Validation (Guardrails, Rebuff, Guidance, LMQL)

LLM APIs and Hosting

Proprietary API (OpenAI, Anthropic)

Open API (Hugging Face, Replicate)

Cloud Provider (AWS, GCP, Azure, Coreweave)

Opinionated Cloud (Databricks, Anyscale, Mosaic, Modal, Runpod, ...)

LEGEND

Gray boxes show key components of the stack, with leading tools/systems listed

Arrows show the flow of data through the stack

- - - ▶ Contextual data provided by app developers to condition LLM outputs

—▶ Prompts and few-shot examples that are sent to the LLM

—▶ Queries submitted by users

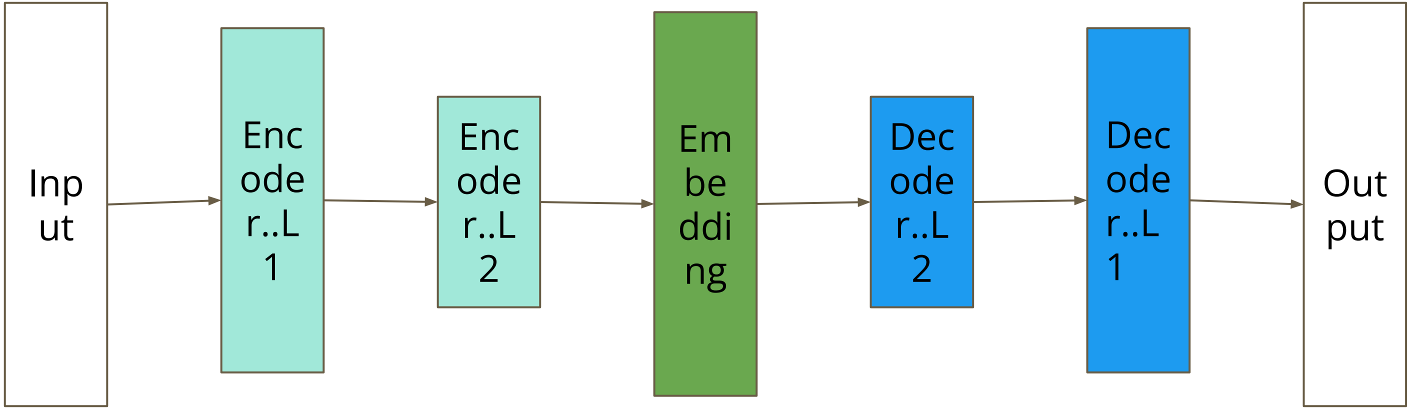
—▶ Output returned to users

a16z Enterprise

Credit : https://a16z.com/emerging-architectures-for-llm-applications/

Embeddings - Create embedding model

https://keras.io/api/layers/core_layers/embedding/



Persisting Embeddings

New Storage Type : Vector stores

Capability :

- Store embedding
- Retrieve them with various techniques and performance suited for applications (similarity search, distance search etc.)

e.g. Budding ecosystem Chromadb, pinecone and a lot of others in variety of models

Embeddings Retrieval - Vector search techniques

Distance Based

- L1 distance,
- L2 distance
- Tanimoto
- Jaccard distance

Similarity

- Cosine similarity
- Floating point vector similarity metrics
- Binary vector similarity metrics

Other

- Space partitioning
- k-dimensional trees, inverted file index

Storage and new retrieval framework for representations

Representation Type	Storage	Retrieval mechanism
Rows and columns - Structured	RDBMS	SQL - Set Theory
Columnar,key value	NOSQL (cassandra,HBASE,redis)	CQL/SQL Wrappers/APIs(set theory and custom ops)
Relationships	Graph	cypher,SPARQL (predicate logic)
Inverted index	Bag of Words - Elastic	APIs keyword search
Embeddings	Vector DB	Semantic search (Cosine similarity, distance measures)

Embedding Models - what to look for

It depends on the application. But generally...

Language : Words vs Sentences vs documents

Size of the model

Hosting models and costs

Embedding length

Architecture

Explore Embedding models

Sentence Transformer

https://www.sbert.net/docs/pretrained_models.html#model-overview

AI’s Share Of US Startup Funding Doubled In 2023

- Source

https://news.crunchbase.com/ai-robotics/us-startup-funding-doubled-openai-anthropic-2023/

New programming paradigm, New representation, New retrieval mechanisms….

- **Vector Stores**
- **Embeddings - Multi Model**
- **LLMs**
- **Fine tuning**
- **Inference management - RAGs, Prompt Engineering**

Resources/References

Industrial strength embedding papers:

https://research.facebook.com/publications/embedding-based-retrieval-in-facebook-search/

https://cloud.google.com/blog/topics/developers-practitioners/meet-ais-multitool-vector-embeddings

https://www.uber.com/blog/innovative-recommendation-applications-using-two-tower-embeddings/

https://engineering.linkedin.com/blog/2023/how-linkedin-is-using-embeddings-to-up-its-match-game-for-job-se

My inspiration - Women in AI

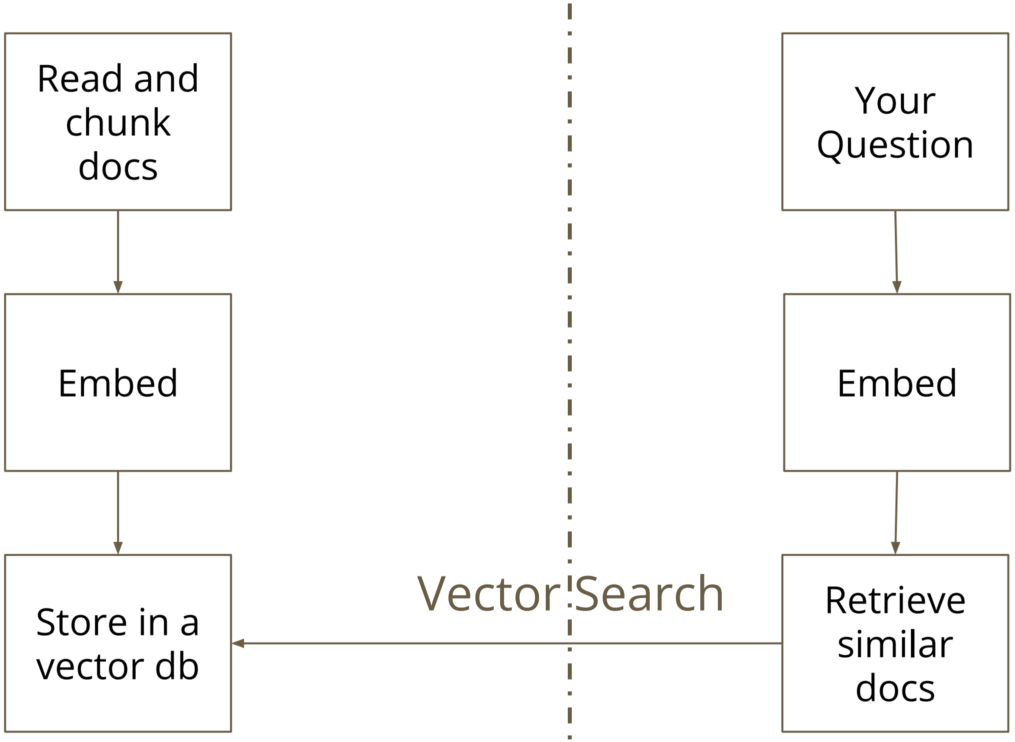
pioneering to modern - 1852 - current

- The Dawn of Computing and AI: Ada Lovelace
- The Emergence of AI: Elaine Rich
- The Advent of Social Robotics: Cynthia Breazeal
- Privacy-Preserving Data Analysis: Cynthia Dwork
- Computer Vision - Dr. Fei-Fei Li, a computer science professor at Stanford
- Ethical AI - Timnit Gebru
- Open AI - Mira Murati
    - and many many more………

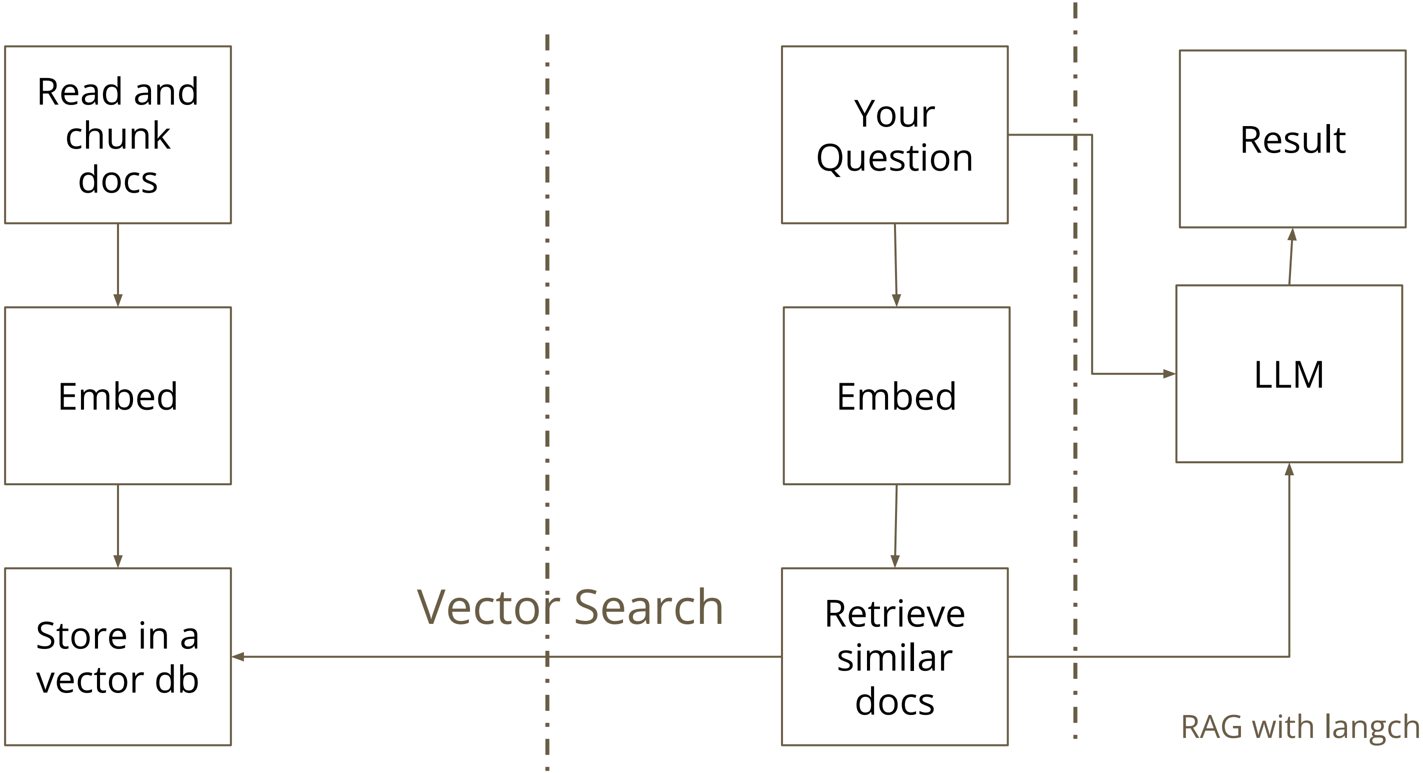
Hands on Workshop

Build a QA System - Vanila and with RAG

Building a Q/A application - Part I



Building a Q/A application RAG - Part II



RAG with langchain

Workshop

Build a QA app

Go to Collab

Follow along:

https://github.com/jaynetra/AppliedAIConf2023