Artificial Intelligence
Introduction to Generative AI

National Prosecution Best Practices Conference

October 7, 2024

Frank P. Coyle, PhD

coyle@smu.edu

Associate Professor, Computer Science
SMU
Dallas, Texas

Artificial Intelligence (AI)

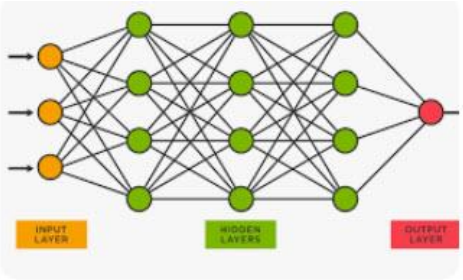
Expert Systems

Machine Learning (Neural Networks)

Large Language Models (LLMs) ChatGPT 3.5

1980s
Rules based on human experts

2000s
Neural Networks trained on large data sets using GPUs (Graphical Processing Units)



2022
Large Language Models trained on all the public data available on the Internet

Important LLM Terminology

- Prompts or Query
  - what we say to the LLM
- Tokens
  - words or parts of words used to train an LLM
- Embeddings
  - a numerical representation of sentences used to train LLMs
  - all text is converted to numbers for training
- Context Window
  - the maximum number of tokens/words that an LLM can handle at a time
- RAG – Retrieval Augmented Generation
  - providing your own information to the LLM to help it answer questions
  - usually private data that an LLM has not been trained on
- Fine-Tuning
  - providing additional training to teach an LLM new concepts

A Shallow Dive in the Technology behind LLMs

Steps in building an LLM

- Obtain ‘lots’ of **data**
  - LLMs trained on web pages, books & all the documents on the web
- Break the text up into words (also called ***tokens***)
- Use machine learning (neural networks) to convert the text into **numbers** that capture meaning
- The technical term for these numeric values is **embeddings**.

Embeddings are the secret sauce behind LLMs

“The prosecution approached the bench.”

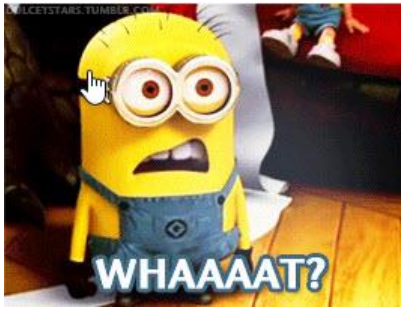
```
[-0.64516115  0.1579973  -0.18486351 -0.03695639  0.28924885 -0.09498847
 -0.43030658  0.15431975  0.1239115   0.17329243  0.21807285 -0.21267048
 -0.22992504  0.24730667 -0.4451861  -0.04412757  0.31399828 -0.23861146
  0.11274178 -0.70143986 -0.12033968  0.20097078 -0.34433937  0.0200157
 -0.17284475  0.43762085 -0.01585343 -0.16477302  0.13359785 -0.3297498
 -0.27070326 -0.45194244 -0.15027043  0.13564251 -0.31725803 -0.71317255
  0.23994786 -0.06365798 -0.2350698   0.12471341 -0.33628556 -0.45893794
  0.0239193  -0.01461021  0.7949769  -0.1963934  -0.38624054 -0.18512818
  0.00129966 -0.09555561  0.23405671  0.32197502 -0.04406496 -0.14301962
 -0.06501128 -0.10083073  0.1285449   0.08399501  0.19720553  0.0606354
 -0.22448681 -0.557067   -0.22160476  0.06177633  0.534892   -0.1717653
 -0.567688    0.5929364  -0.14680988  0.78627753 -0.09622003  0.00605357
  0.31533802  0.21695644 -0.5902365  -0.2347756   0.32014322  0.29467028
  0.2876221  -0.18719622 -0.44544363  0.37007272 -0.09240708  0.4406842
  0.20087402  0.22351162 -0.23393816  0.18165983 -0.05705923 -0.2805166
  0.6086521   0.2634202  -0.28903696  0.18512465  0.01641486 -0.09391
  0.06074792  0.06188582 -0.26275593  0.47158718]
```

An embedding captures…

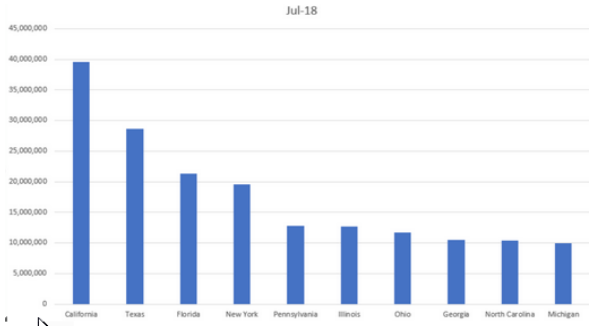
```
[-0.64516115  0.1579973  -0.18486351 -0.03695639  0.28924885 -0.09498847
 -0.43030658  0.15431975  0.1239115   0.17329243  0.21807285 -0.21267048
 -0.22992504  0.24730667 -0.4451861  -0.04412757  0.31399828 -0.23861146
  0.11274178 -0.70143986 -0.12033968  0.20097078 -0.34433937  0.0200157
 -0.17284475  0.43762085 -0.01585343 -0.16477302  0.13359785 -0.3297498
 -0.27070326 -0.45194244 -0.15027043  0.13564251 -0.31725803 -0.71317255
  0.23994786 -0.06365798 -0.2350698   0.12471341 -0.33628556 -0.45893794
  0.0239193  -0.01461021  0.7949769  -0.1963934  -0.38624054 -0.18512818
  0.00129966 -0.09555561  0.23405671  0.32197502 -0.04406496 -0.14301962
 -0.06501128 -0.10083073  0.1285449   0.08399501  0.19720553  0.0606354
 -0.22448681 -0.557067   -0.22160476  0.06177633  0.534892   -0.1717653
 -0.567688    0.5929364  -0.14680988  0.78627753 -0.09622003  0.00605357
  0.31533802  0.21695644 -0.5902365  -0.2347756   0.32014322  0.29467028
  0.2876221  -0.18719622 -0.44544363  0.37007272 -0.09240708  0.4406842
  0.20087402  0.22351162 -0.23393816  0.18165983 -0.05705923 -0.2805166
  0.6086521   0.2634202  -0.28903696  0.18512465  0.01641486 -0.09391
  0.06074792  0.06188582 -0.26275593  0.47158718]
```

- **Syntax**
- **Semantics**
- **Context and Relationships**
- **Word Co-occurrence**
- **Relationships and Analogies**
- **Hierarchical Information**

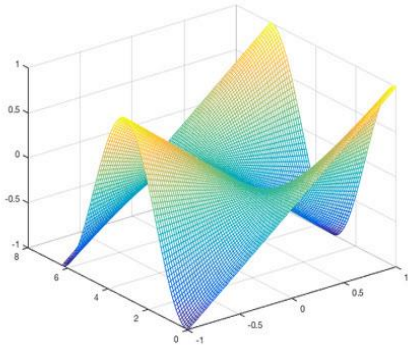
The LLM learns to group contextually similar concepts close together in **n-dimensional space**



Dimensional Spaces



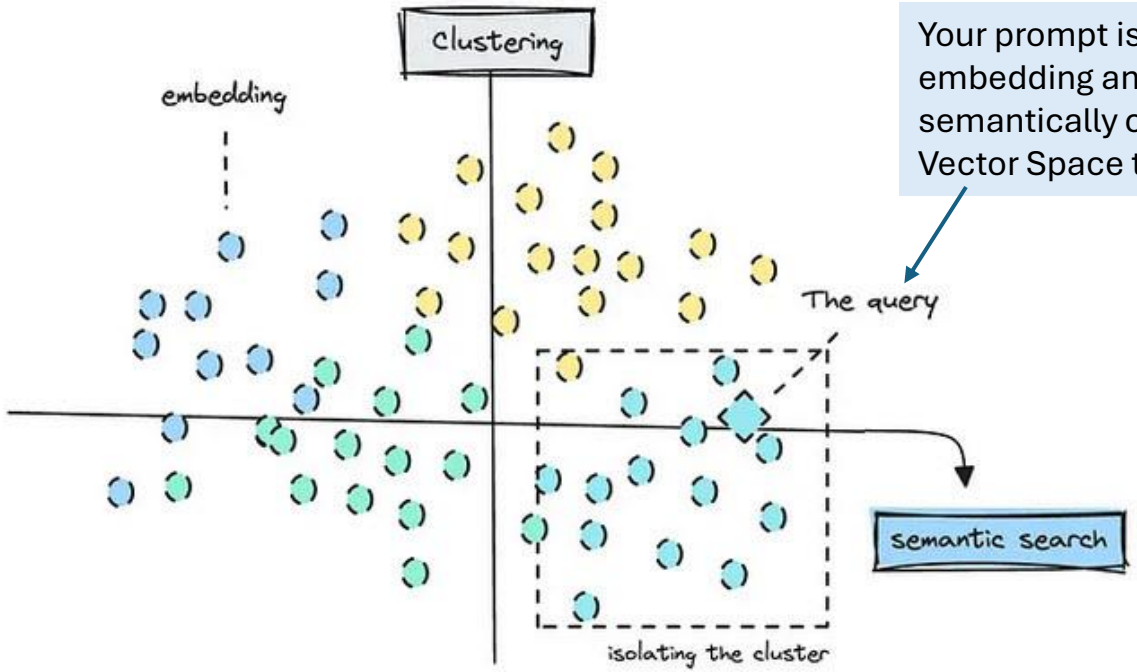
2D- Space



3D- Space

Multi-Dimensional Space

The ‘brain’ of an LLM is its Vector Space



Your prompt is converted into an embedding and the LLM tries to find the semantically closest concepts in its Vector Space to compose its answer

Hallucinations

- LLMs return answers based on the distance between terms and concepts in multi-dimensional space
- LLMs try and return their ‘best’ answer
- But the closest points in n-dimensions may not be the ‘best’ answer – or may be an incorrect answer – thus what we call a ***hallucination***

End of Shallow Dive

Prompt Engineering

- The process of designing and optimizing the input (or "prompt") to achieve desired outputs or behaviors.
- The goal is to structure the input in a way that maximizes the quality of the model's response
- Since language models respond to the text they are provided, how a task is described or queried can significantly impact the result.

Prompt Engineering Example
Role Playing

Role: "You are a highly skilled legal assistant working for the District Attorney's office. Your task is to help prepare a case for trial based on the following evidence. Review the details, organize the facts, and suggest next steps for the DA. Your recommendations should be based on legal standards, and you must highlight any areas where further investigation or clarification is needed.

Evidence:

- A signed witness statement claiming they saw the defendant at the crime scene at 9:00 PM.
- Security camera footage showing a person matching the defendant's description entering a nearby store at 9:15 PM
- A forensic report showing fingerprints matching the defendant's on the weapon found at the scene.
- The defendant's alibi stating they were at a friend's house at 8:30 PM and stayed there all night
- Phone records showing the defendant made a call near the crime scene at 8:45 PM

Prompt continues...

Based on this evidence, help the DA prepare the following:

1. A timeline of events to corroborate or refute the defendant's alibi.
2. Identify any inconsistencies in the evidence that should be investigated further.
3. Suggest legal motions or strategies that the DA could use to strengthen the case.
4. Highlight any risks or weaknesses in the case that the defense might exploit, and propose solutions."

Fine-Tuning Large Language Models (LLMs)

Fine-tuning is the process of taking a pre-trained large language model and adapting it to a specific task or dataset.

This allows the model to learn domain-specific knowledge while retaining its broad understanding of language.

Key steps include:

- **Data Preparation:** Curate a task-specific dataset
  - A task-specific dataset for fine-tuning should be carefully curated, ensuring it is labeled, relevant, and sufficiently diverse to represent the target domain.
- **Evaluation**: Assess the model's performance on validation data
- **Benefits:** Improves model performance for specific use cases (e.g., legal, medical text generation) without needing to train from scratch.
- **Downside**: Requires some technical expertise

Fine-tuning enables customization while leveraging the powerful generalization capabilities of LLMs.