PLG Disrupt

Presented by

Product Led Hub

2024

Platinum Sponsor

kaizen GAMING

Retrieval-Augmented Generation for talking with your private data using LLM

Enrico Zimuel, Tech Lead @ Elastic

PLG Disrupt

Agenda

- Introduction to AI
- Machine Learning
- Generative AI
- Neural Network
- Large Language Model (LLM)
- Prompt engineering
- Retrieval-Augmented Generation (RAG)
- Embeddings and Vector Database
- Semantic Search with Elasticsearch
- Hands-on: how to build a RAG system

Artificial Intelligence (AI)

PLG Disrupt

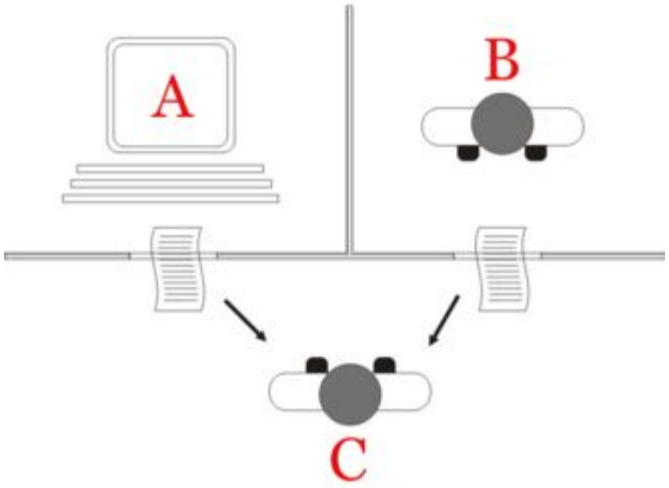
Created using DALL·E 3 with the prompt “Create an image for a tech conference in Athens”

Artificial intelligence

- Many definitions proposed
- For example:
  - The ability of a digital computer to perform tasks commonly associated with intelligent beings
  - An umbrella term for a range of algorithm-based technologies that solve complex tasks by carrying out functions that previously required human thinking

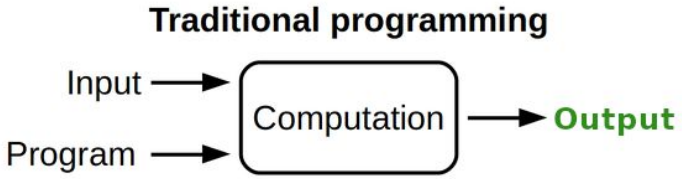
Turing test

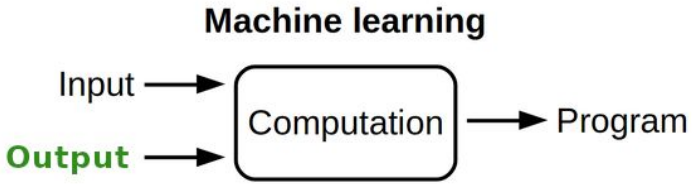
- Proposed by **Alan Turing** in 1950 (imitation game)
- A test of a machine's ability to exhibit intelligent behaviour equivalent to, or indistinguishable from, a human
- C.Biever, ChatGPT broke the Turing test, Nature, 25 July 2023



Machine learning

- **Machine Learning (ML)** is the use and development of computer systems that are able to **learn** and **adapt** without following explicit instructions, by using algorithms and **statistical models** to analyse and draw inferences from patterns in data



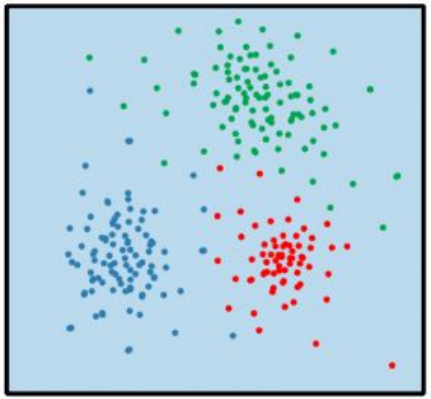


Three types of ML

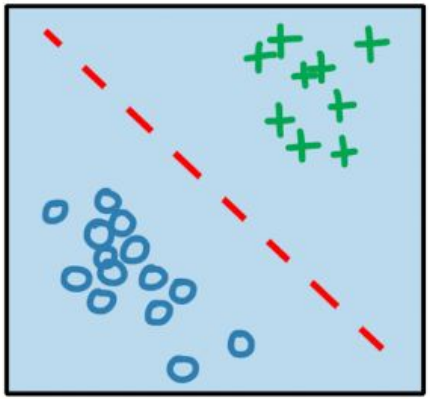
- **Supervised learning**: use of labeled datasets to train algorithms that to classify data or predict outcomes (eg. image and speech recognition, recommendation systems, fraud detection)
- **Unsupervised learning**: algorithms learn patterns exclusively from unlabeled data (eg. clustering, anomaly detection)
- **Reinforcement learning:** training method based on rewarding desired behaviors and punishing undesired ones (eg. NLP, LLM)

PLG Disrupt

unsupervised learning



supervised learning



reinforcement learning

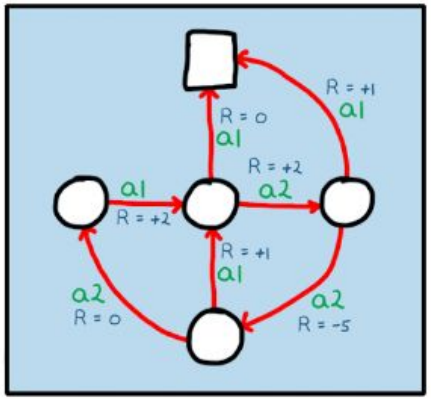


Image source: https://uk.mathworks.com/discovery/reinforcement-learning.html

Generative AI

- **Generative Artificial Intelligence** (GenAI) is artificial intelligence capable of generating text, images, or other media, using generative models
- GenAI models **learn the patterns and structure** of their input training data and then generate new data that has **similar characteristics**
- It's used in many industries: art, writing, software development, healthcare, finance, gaming, marketing, etc
- Generative AI market is projected to reach **$ 66.62bn** in 2024 with a growth rate (CAGR 2024-2030) of **20.80%**

Neural Network

- A **neural network** is a method in artificial intelligence that teaches computers to process data in a way that is inspired by the human brain
- Collection of **nodes** (artificial neurons) with inputs and outputs. A neuron computes some non-linear function of the sum of its inputs
- The nodes are collected in **layers**
- If the number of layers > 3 we call it **deep learning network**

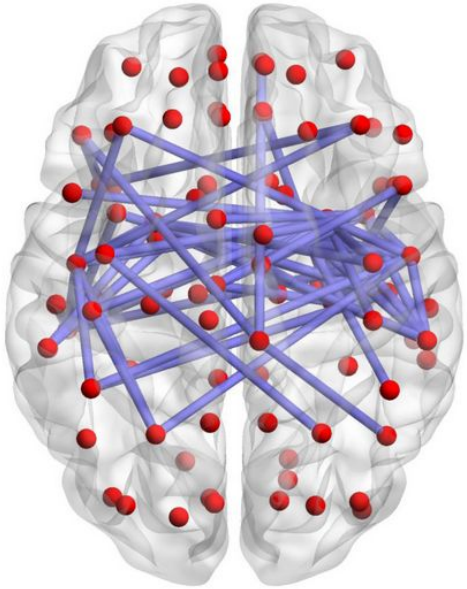
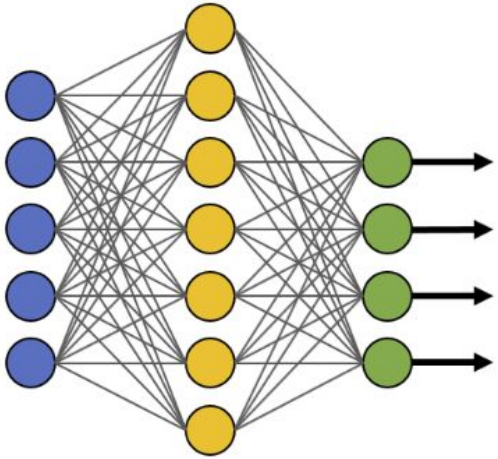


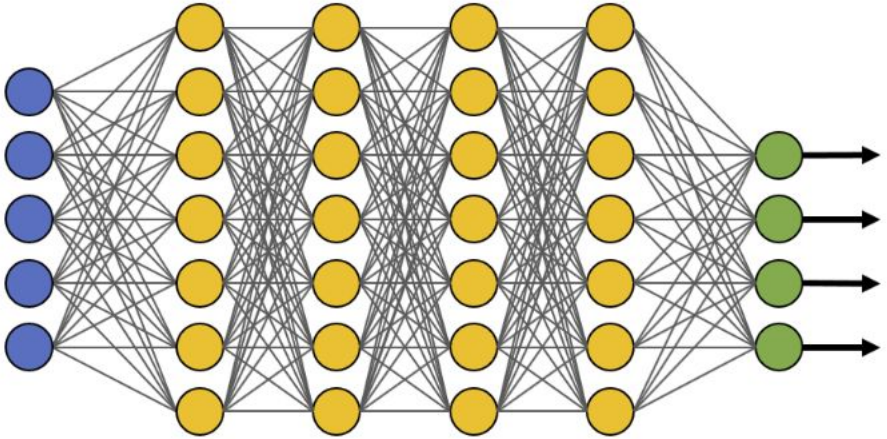
Image source: https://commons.wikimedia.org/wiki/File:Brain_network.png

PLG Disrupt

Simple Neural Network



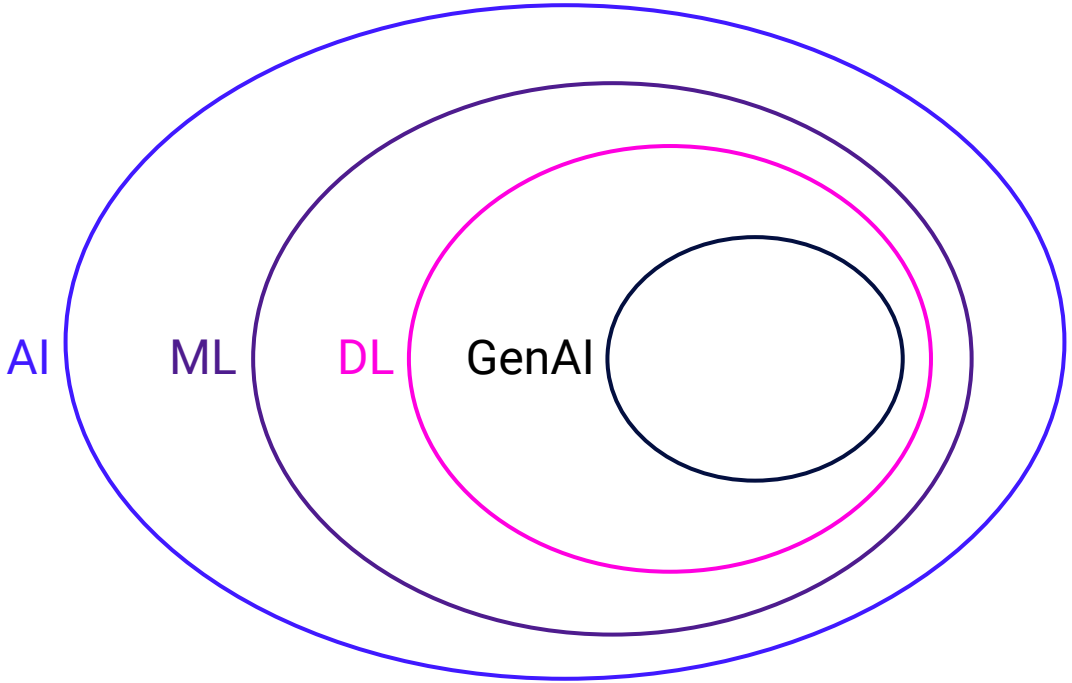
Deep Learning Neural Network



Input Layer Hidden Layer Output Layer

PLG Disrupt

AI ⊃ ML ⊃ DL ⊃ GenAI

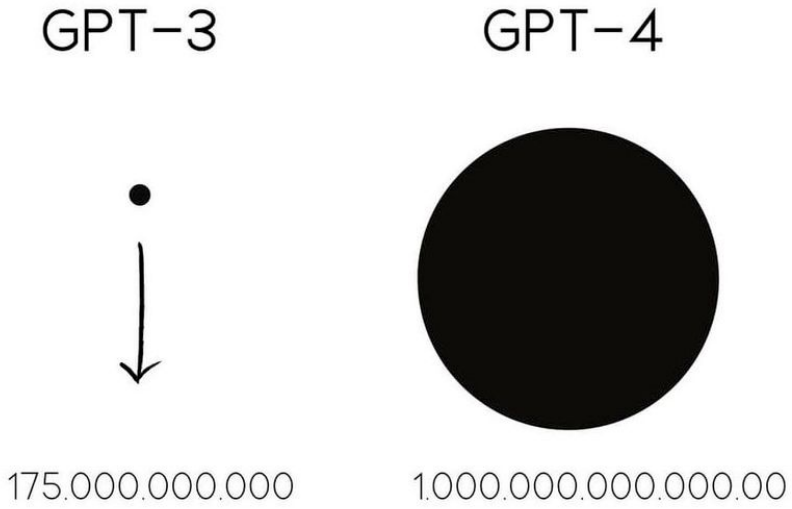


LLM

- **Large Language Model** (LLM) consisting of a neural network with many parameters (typically billions), trained on large quantities of unlabelled text using self-supervised learning
- A message is splitted in **tokens**
- Each token is translated in a number using an operation called **embeddings**
- LLM works by taking an input text and **repeatedly predicting** the next token or word

Size of GPT-4

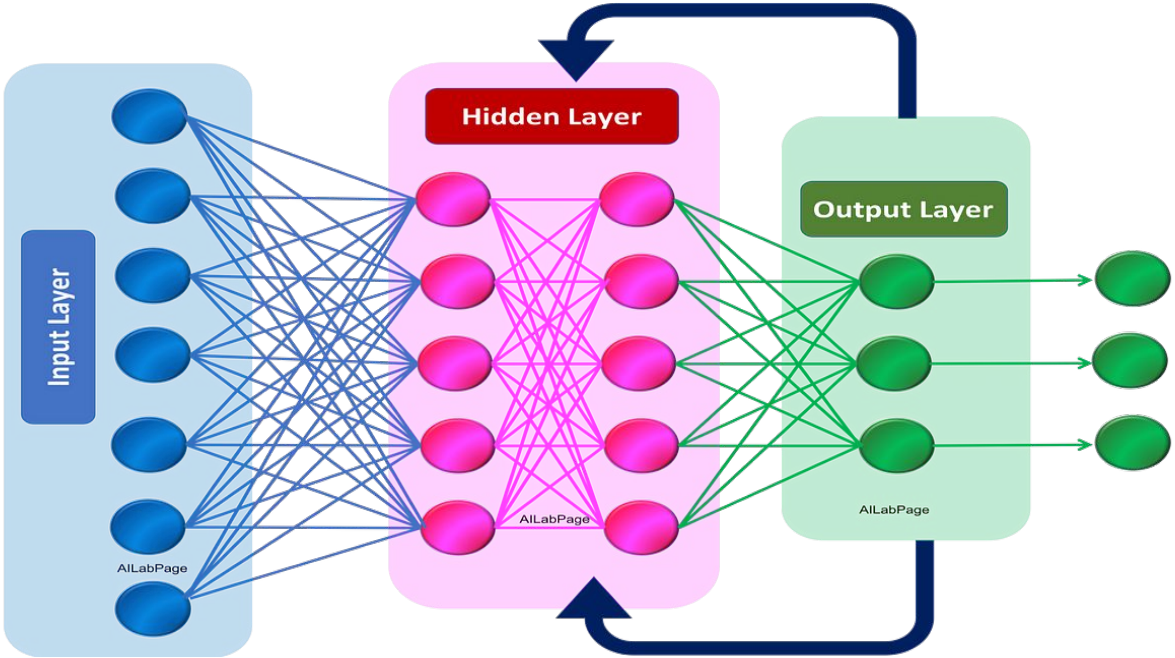
- Around **1.76 trillion** parameters
- Neural network with **120** layers
- Process up to **25,000** words at once
- Estimated training cost is $200M using 10,000 Nvidia A100 GPU for 11 months



RNN, before LLM

- **Recurrent Neural Networks** (RNN)
- Prediction of the next words based on the previous words
- RNN does not scale
- To complete a sentence the model needs to understand the structure of the entire sentence
- Eg. “The teacher taught the students with the book”
  - Did the teacher teach using the book?
  - Did the student have the book?
  - Or was it both?

RNN (2)


Input Layer
Hidden Layer
Output Layer
AILabPage
AILabPage
AILabPage

Attention Is All You Need

- Google and University of Toronto published a paper in 2017 “Attention is All You Need”
- In this paper they introduced the **Transformer architecture**
- This novel approach unlocked the progress in generative AI that we see today
- Scale efficiently, parallel process, attention to input meaning

Attention Is All You Need

Ashish Vaswani[*] Google Brain avaswani@google.com

Noam Shazeer[*] Google Brain noam@google.com

Niki Parmar[*] Google Research nikip@google.com

Jakob Uszkoreit[*] Google Research usz@google.com

Llion Jones[*] Google Research llion@google.com

Aidan N. Gomez[*] [†] University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser[*] Google Brain lukaszkaiser@google.com

Illia Polosukhin[*] [‡] illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

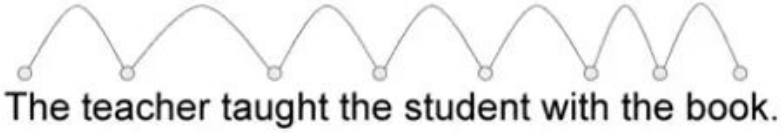
[*] Equal contribution. Listing order is random. Jakob proposed replacing RNNs with self-attention and started the effort to evaluate this idea. Ashish, with Illia, designed and implemented the first Transformer models and has been crucially involved in every aspect of this work. Noam proposed scaled dot-product attention, multi-head attention and the parameter-free position representation and became the other person involved in nearly every detail. Niki designed, implemented, tuned and evaluated countless model variants in our original codebase and tensor2tensor. Llion also experimented with novel model variants, was responsible for our initial codebase, and efficient inference and visualizations. Lukasz and Aidan spent countless long days designing various parts of and implementing tensor2tensor, replacing our earlier codebase, greatly improving results and massively accelerating our research.

[†] Work performed while at Google Brain.

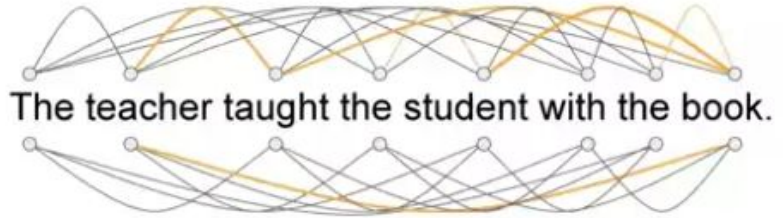
[‡] Work performed while at Google Research.

RNN vs Transformers

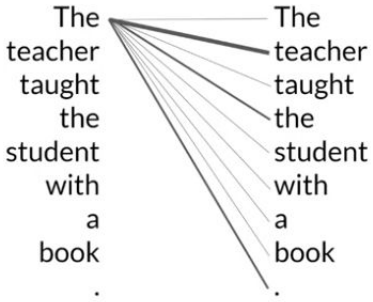
RNN

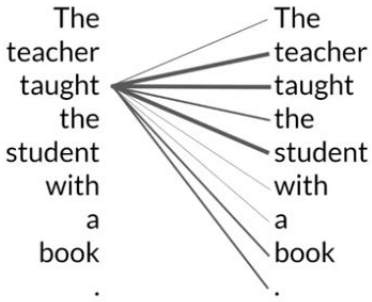


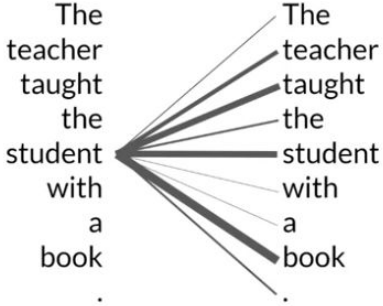
Transformers



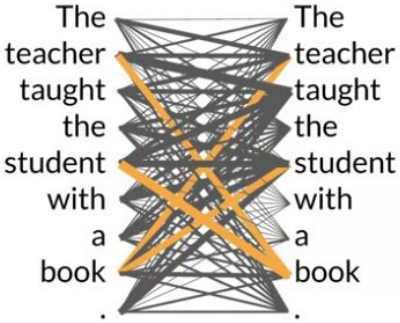
Attention map





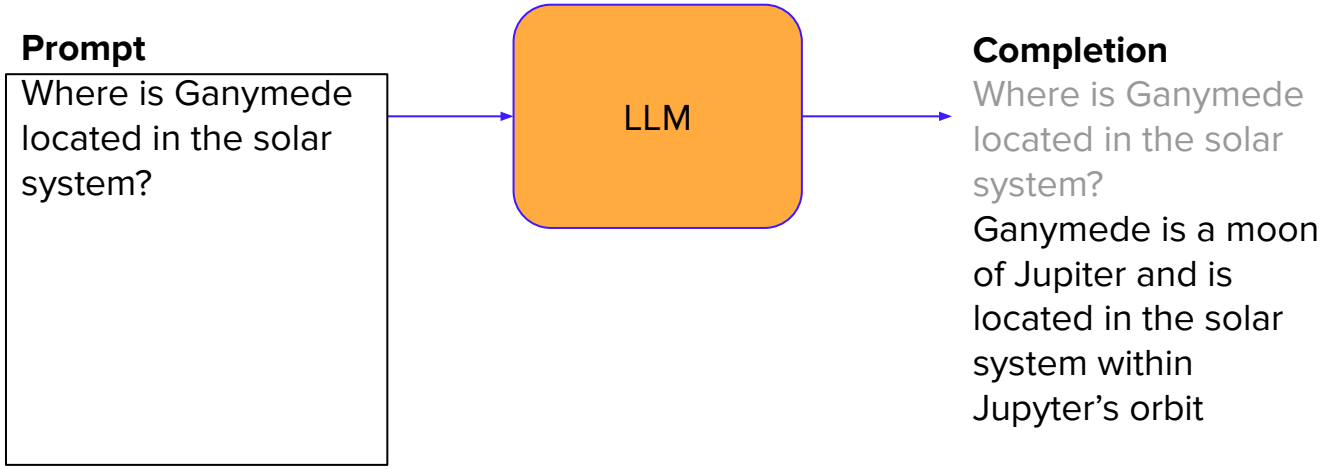


eg. **book** is strongly connected with **teacher** and **student**



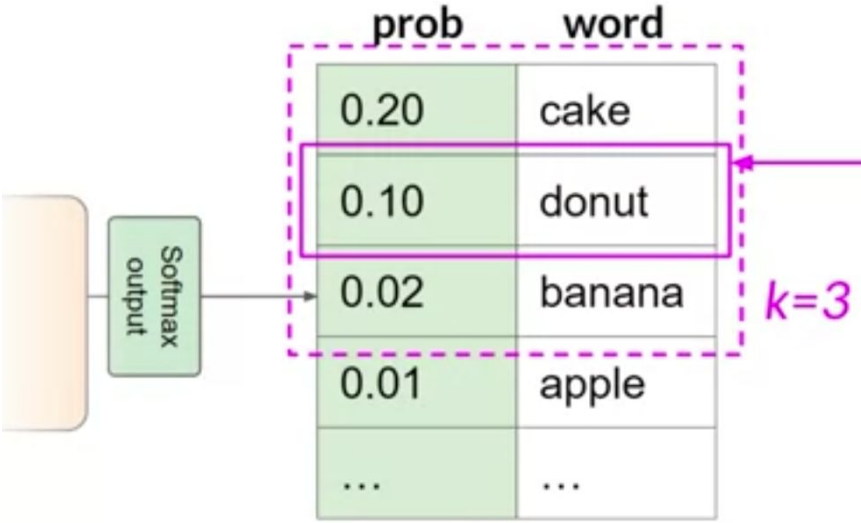
self-attention

Prompt



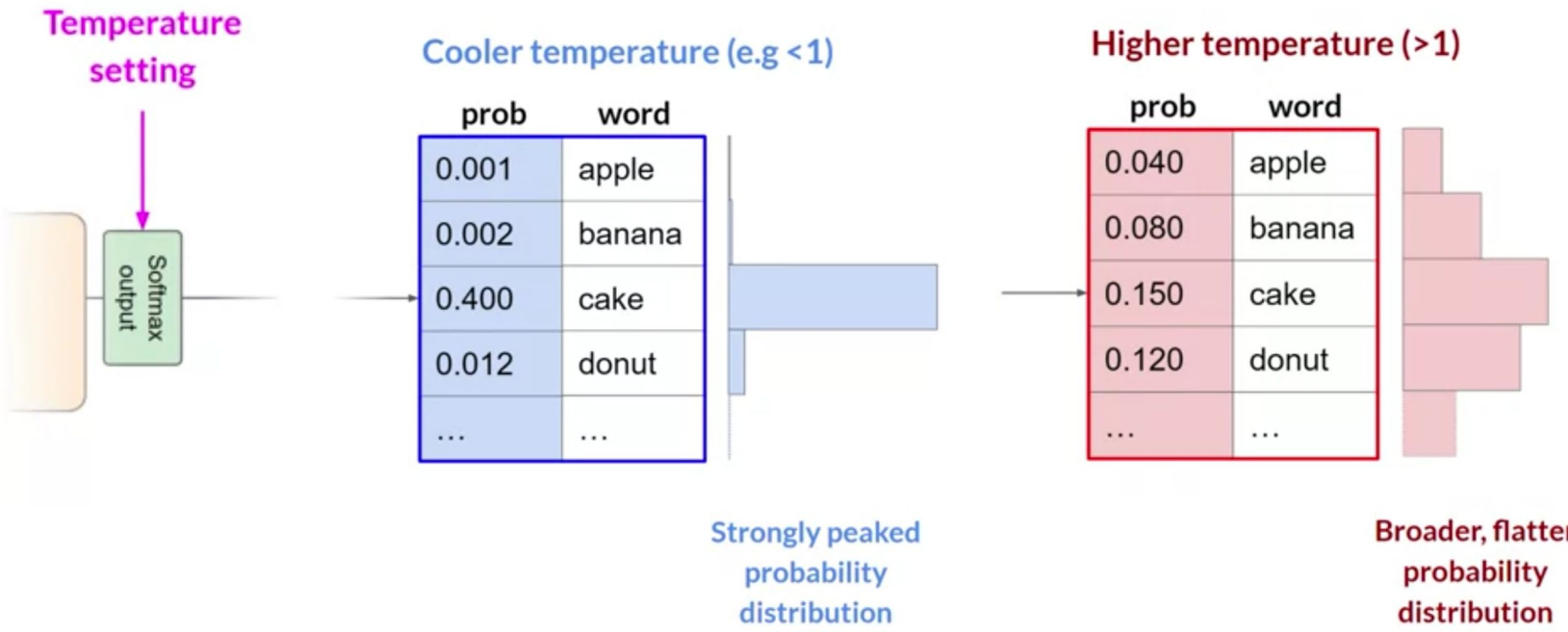
Context window: few thousand words

Top-k



top-k: select an output from the top-k results after applying random-weighted strategy using the probabilities

Temperature

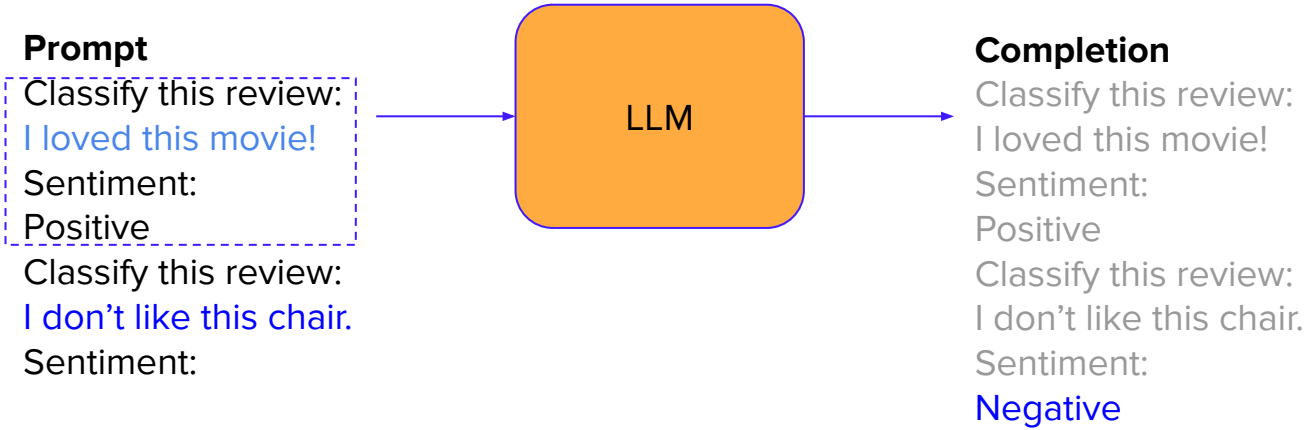


Prompt engineering

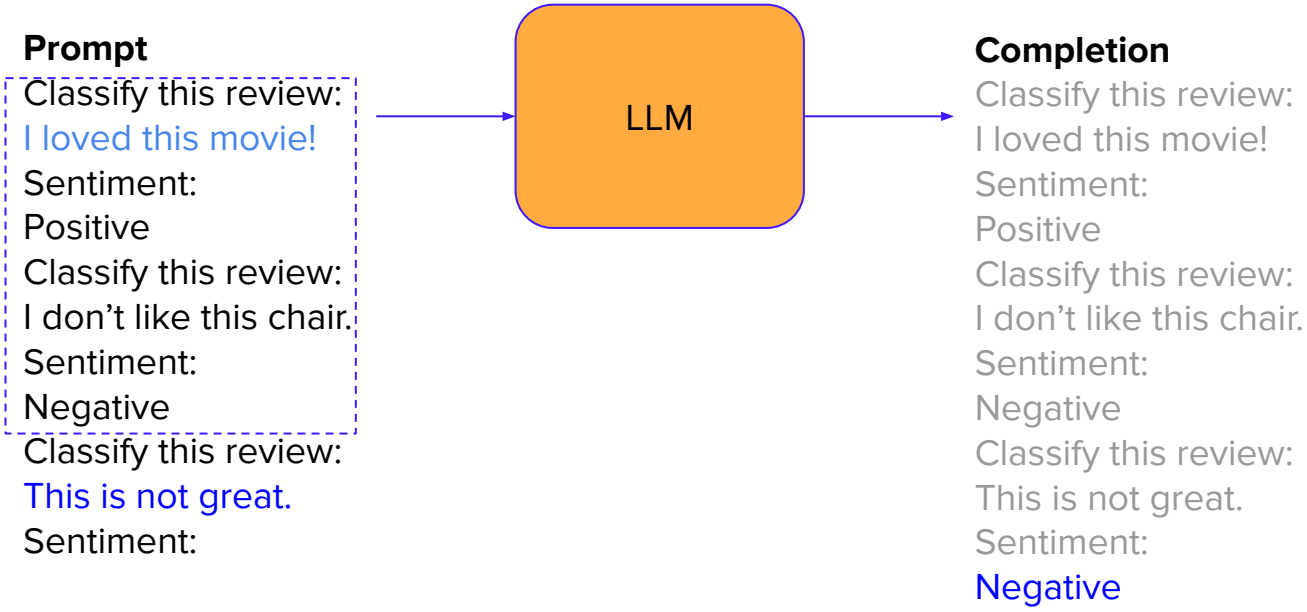
- You can encounter situations where the model doesn't produce the outcome that you want on the first try
- You may have to revisit the language several times to get a good answer
- The development and improvement of the prompt is known as **prompt engineering**
- One powerful strategy is to include examples of the task that you want the model to carry out inside the prompt
- This is called **In-Context Learning (ICL)**

ICL - zero shot inference

ICL - one shot inference



ICL - few shot inference



Retrieval-Augmented Generation (RAG)

- **RAG** is a technique in natural language processing that combines information retrieval systems with **Large Language Models** (LLM) to generate more informed and accurate responses
- It is composed by the following parts:
  - **Retrieval-Augmented**
  - **Generation**

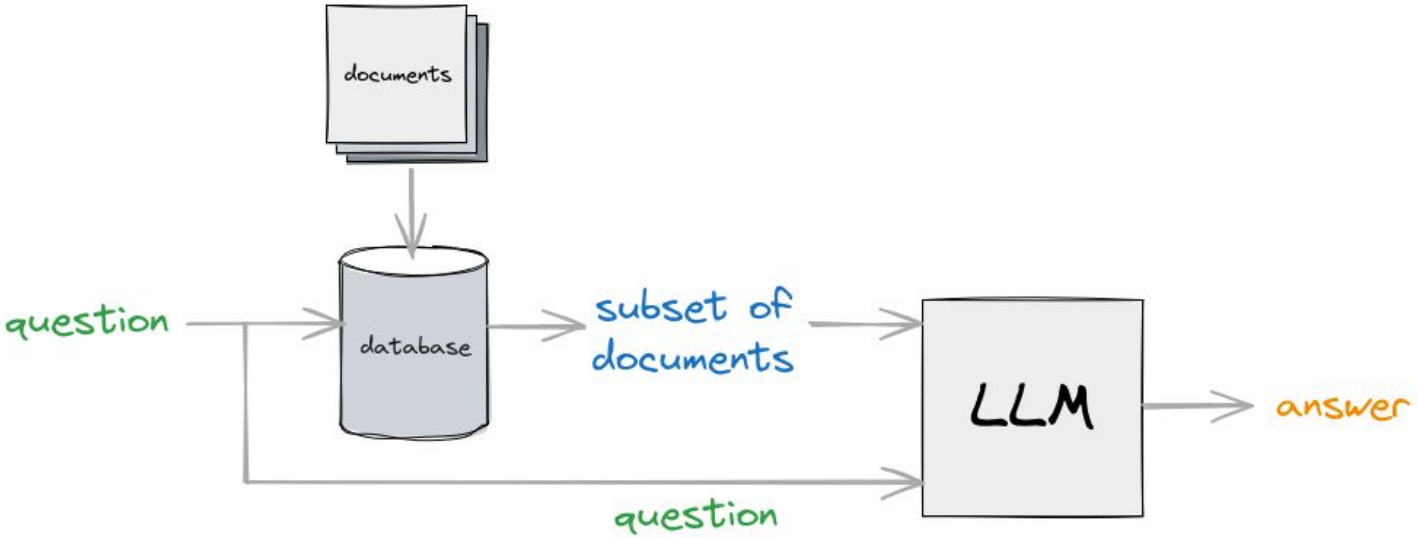
Generation

- LLMs like ChatGPT are a disruptive technology
- They are very useful and powerful in many industries
- But they have some limitations:
  - **No source** (potential hallucinations)
    - How can I verify the information coming from an LLM?
    - What sources has been used to generate the answer?
  - **Out of date**
    - An LLM is trained in a period of time
    - For update we need to retraining the model

Retrieval-Augmented

- Goal: answer to a **question** given in natural language using a a public or private knowledge (documents)
- We need a **semantic search database** to extract, starting from the question, a subset of relevant documents (**context**)
- We pass the **question + context** to an LLM with the following prompt:
  - *Given the following **{context}** answer to the following **{question}***
- The LLM returns an answer in natural language

RAG architecture

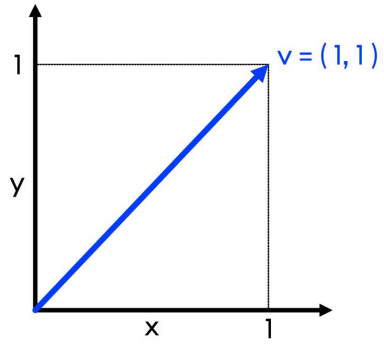


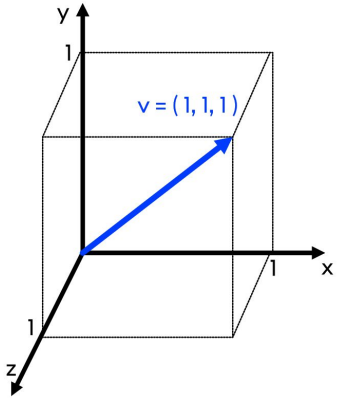
Retrieve relevant documents

- How we can retrieve relevant documents from a database using a question in natural language?
- We need to use **semantic search**
- One solution is to use a **vector database**
- A vector database is a system that uses **vectors** (set of numbers) to retrieve semantic knowledge
- The semantic similarity is translated into a mathematical problem: distance between vectors

What is a vector?

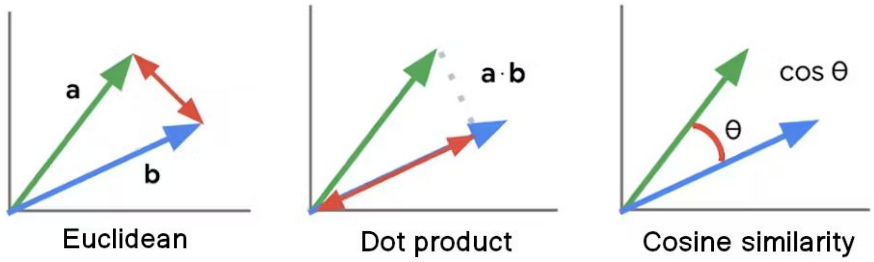
- A vector is a set of numbers
  - Example: a vector of 3 elements [10.5, 11.23, -10]
- A vector can be represented in a **multi-dimensional space**





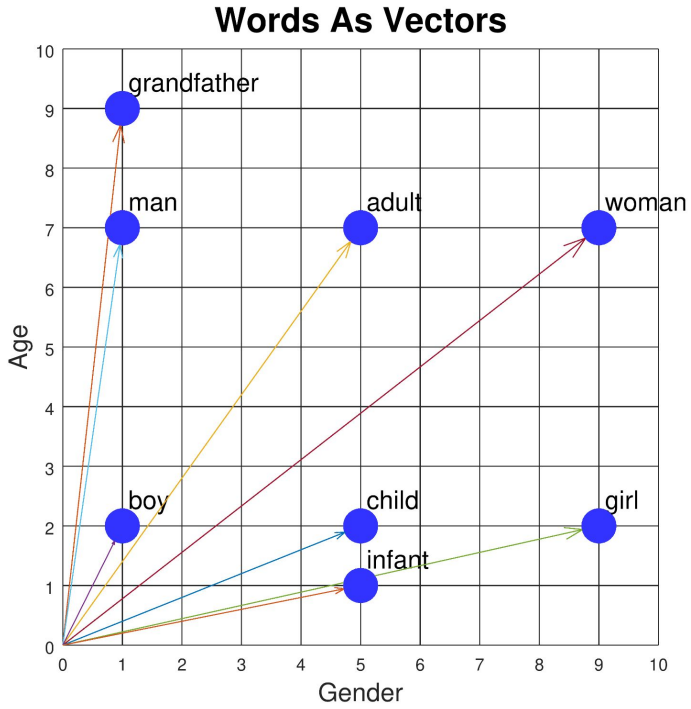
Similarity between two vectors

- Two vectors are (semantically) similar if they are close to each other
- We can measure the distance between vectors
- There are many methods:
  - Euclidean distance
  - Dot product
  - Cosine similarity



Embedding

- **Embedding** is the translation of an input (document, image, sound, movie, etc) to a vector
- Embedding for LLM is typically done by a **neural network**
- The goal is to **group information that are semantically related** to each other
- See projector.tensorflow.org



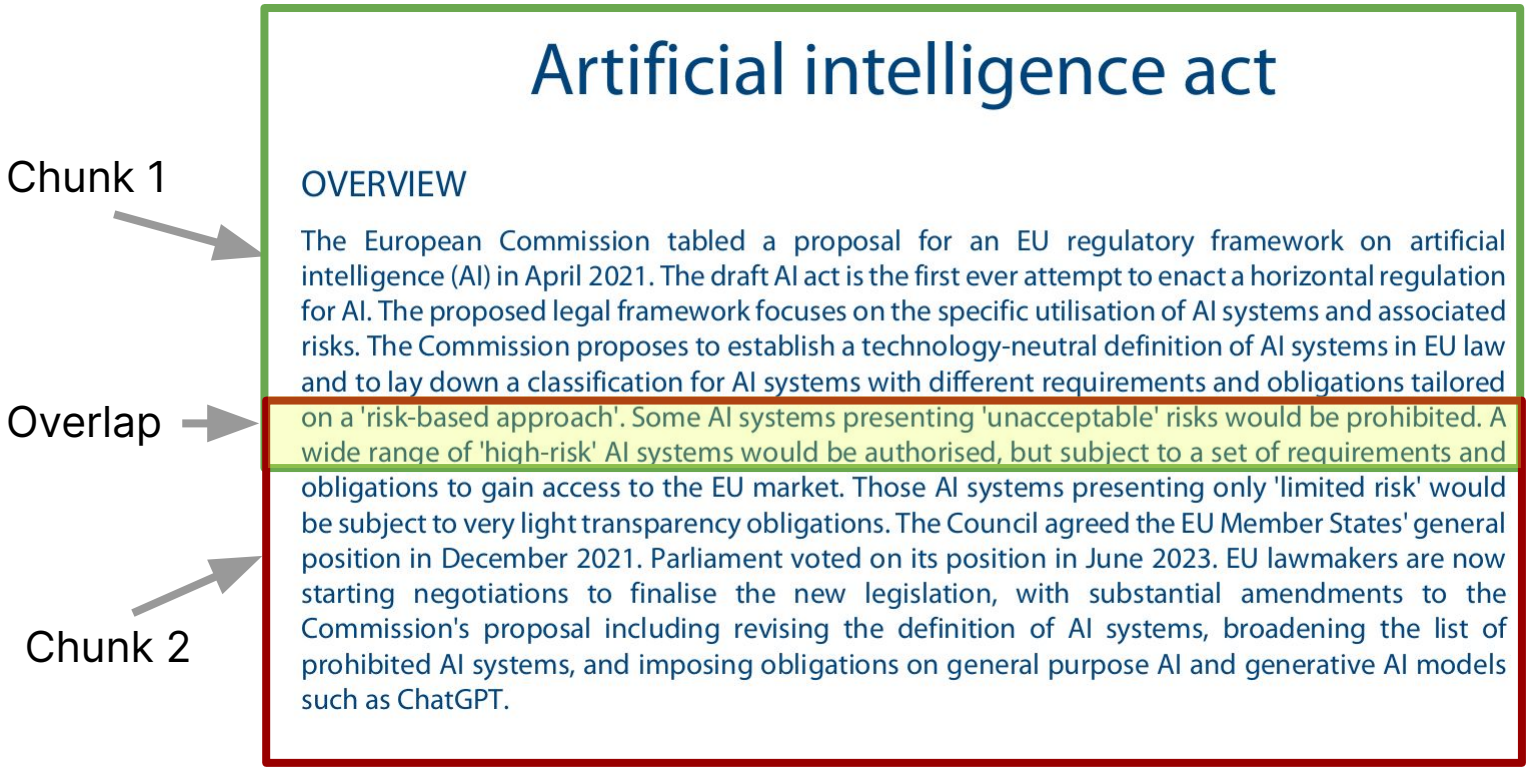
Vector database + LLM

- We can query a vector database using natural language (e.g. a question)
- The query produces a set of relevant documents ordered by a score
- We can extract the top-n score documents and pass it as **context** for a prompt using the previous **question**

Split the documents in chunk

- We cannot use big documents since we need to pass it in the context window (e.g. gpt-3.5-turbo from 4k to 16k)
- We need to split the knowledge into **chunk of information** (eg. 100 words) to be able to capture semantic meaning using embeddings
- To avoid semantic breakings between two chunks we can use an overlap: start with the last n-words of the previous chunk

Chunk and overlap



PLG Disrupt
Powered by Product-Led Hub

Hands-on: build a RAG system using LangChain with GPT 4 and Elasticsearch

Hands-on: Google Colab

PLG Disrupt

https://ela.st/plg-disrupt-zimuel-talk

Time For Questions

PLG Disrupt

Thank You

PLG Disrupt

Presented by Product-Led Hub