Fine-Tuning LLMs With Your Data

Tutorial at URAI'24

Prof. Dr.-Ing. Janis Keuper

INSTITUTE FOR MACHINE
LEARNING AND ANALYTICS

HOCH
SCHULE
OFFEN
BURG

LLM Tutorial
Intro: Lecturer

INSTITUTE FOR MACHINE LEARNING AND ANALYTICS

https://imla.hs-offenburg.de/

https://www.keuper-labs.org/

Prof. Dr.-Ing. Janis Keuper

- Research Professor for “Data Science”
- Head of the Institute for Machine Learning and Analytics

Research Interests

- Machine Learning
  - Generative Models
  - Robustness
- Computer Vision / Pattern Recognition
  - Image Analysis
- ML Systems
- Application of ML+CV to Physical Problems
  - Geo- and Climate Physics

LLM Tutorial
Intro: Lecturer

INSTITUTE FOR MACHINE LEARNING AND ANALYTICS

https://imla.hs-offenburg.de/

https://www.keuper-labs.org/

LLM-Praxis

GEFÖRDERT VOM

Bundesministerium für Bildung und Forschung

Website des LLM-Praxis BMBF Projekts

View the Project on GitHub
LLM-Praxis/website

BMBF Projekt *LLM-Praxis*

Laufzeit: 10/24 - 9/28

KI generiertes Symbolbild: "A small team working with Large Language Models" [Dall-e 3]

Projektbeschreibung

Im Kontext von LLMs, wie bei der Einführung vieler anderer disruptiven Technologien, ergeben sich vielfältige wissenschaftliche, technologische, juristische und gesellschaftliche Fragestellungen von große Breite und Tiefe, welche unfraglich ausnahmslos von großer Bedeutung sind, aber unmöglich alle in einem Projekt behandelt werden können. Daher wird sich das vorgeschlagenen Projekt sowohl technologisch, als auch bei der Betrachtung der Technologiefolgen und Rahmenbedingungen auf wenige Kernthemen beschränken, welche in der Frühphase der Erprobung und Entwicklung von LLM Lösungen beim KMUs von hoher praktische Bedeutung sind und Risiken bezüglich anderer Aspekte minimieren.

Im Rahmen von LLMpraxis sollen explizit keine neuen GPT Algorithmen erforscht oder Modelle von Grund auf entworfen oder trainiert werden. Stattdessen soll auf das inzwischen breite Angebot an (unter offenen

https://www.llm-praxis.de/

LLM Tutorial

Intro: Lecturer

INSTITUTE FOR MACHINE LEARNING AND ANALYTICS

https://imla.hs-offenburg.de/

https://www.keuper-labs.org/

Can Visual Language Models Replace OCR-Based Visual Question Answering Pipelines in Production? A Case Study in Retail.

Bianca Lamm
Markant Services International GmbH
Offenburg, Germany
Bianca.Lamm@de.markant.com

Janis Keuper
Institute for Machine Learning and Analytics (IMLA)
Offenburg University, Germany
keuper@imla.ai

Abstract

Most production-level deployments for Visual Question Answering (VQA) tasks are still build as processing pipelines of independent steps including image pre-processing, object- and text detection, Optical Character Recognition (OCR) and (mostly supervised) object classification. However, the recent advances in vision Foundation Models [25] and Vision Language Models (VLMs) [23] raise the question if these custom trained, multi-step approaches can be replaced with pre-trained, single-step VLMs.
This paper analyzes the performance and limits of various VLMs in the context of VQA and OCR [5, 9, 12] tasks in a production-level scenario. Using data from the Retail-786k [10] dataset, we investigate the capabilities of pre-trained VLMs to answer detailed questions about advertised products in images. Our study includes two commercial models, GPT-4V [16] and GPT-4o [17], as well as four open-source models: InternVL [5], LLaVA 1.5 [12], LLaVA-NeXT [13], and CogAgent [9].
Our initial results show, that there is in general no big performance gap between open-source and commercial models. However, we observe a strong task dependent variance in VLM performance: while most models are able to answer questions regarding the product brand and price with high accuracy, they completely fail at the same time to correctly identity the specific product name or discount. This indicates the problem of VLMs to solve fine-grained classification tasks as well to model the more abstract concept of discounts.

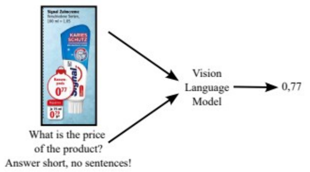


Figure 1. Illustration of the single-step process: input, model, output. The input consists of a product advertising image and a prompt querying specific product or advertising feature. A VLM is used as a model.

progress. The importance of handling multi-modal inputs is highlighted by the growing use of image analysis and image creation. Previous research has shown that VLMs can be effective in Visual Question Answering (VQA), Optical Character Recognition (OCR), or Image Captioning [5, 9, 12]. This study examines the transformation of a multi-step approach into a single-step process through the utilization of VLMs. The considered problem includes an OCR-based pipeline. Hence, the research question arises: *can we replace OCR-based VQA pipelines with VLMs at a production level?* We investigate this question on a use case derived from the retail domain. The basis of the case study is the dataset *Retail-786k* [10] that consists of images cropped from leaflets. Each image presents an advertisement of a

Are Vision Language Models Texture or Shape Biased and Can We Steer Them?

Paul Gavrikov[1] Jovita Lukasik[2] Steffen Jung[3,6] Robert Geirhos[4]
Bianca Lamm[1] Muhammad Jehanzeb Mirza[5] Margret Keuper[6,3] Janis Keuper[1,6]
[1] IMLA, Offenburg University [2] University of Siegen
[3] Max Planck Institute for Informatics, Saarland Informatics Campus
[4] Google DeepMind [5] ICG, Graz University of Technology [6] University of Mannheim

Abstract

Vision language models (VLMs) have drastically changed the computer vision model landscape in only a few years, opening an exciting array of new applications from zero-shot image classification, over to image captioning, and visual question answering. Unlike pure vision models, they offer an intuitive way to access visual content through language prompting. The wide applicability of such models encourages us to ask whether they also *align* with human vision — specifically, how far they adopt human-induced visual biases through multimodal fusion, or whether they simply inherit biases from pure vision models. One important visual bias is the texture vs. shape bias, or the dominance of local over global information. In this paper, we study this bias in a wide range of popular VLMs. Interestingly, we find that VLMs are often more shape-biased than their vision encoders, indicating that visual biases are modulated to some extent through text in multimodal models. If text does indeed influence visual biases, this suggests that we may be able to steer visual biases not just through visual input but also through language: a hypothesis that we confirm through extensive experiments. For instance, we are able to steer shape bias from as low as 49% to as high as 72% through prompting alone. For now, the strong human bias towards shape (96%) remains out of reach for all tested VLMs.

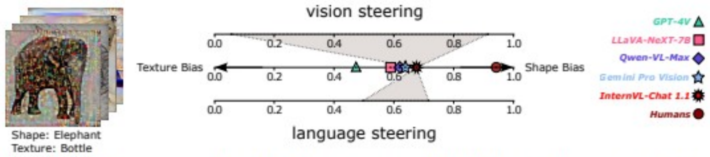


Figure 1: Unlike many unimodal models, vision language models (VLMs) prefer shape over texture for object recognition, but not to the same extent as humans. Further, we find that the (visual) texture/shape bias [1] can be steered through language alone, albeit not to the extent as through vision. Here we visualize the texture/shape bias of some exemplary VLMs, and highlight the steerability of InternVL-Chat 1.1 [2].

INSTITUTE FOR MACHINE LEARNING AND ANALYTICS

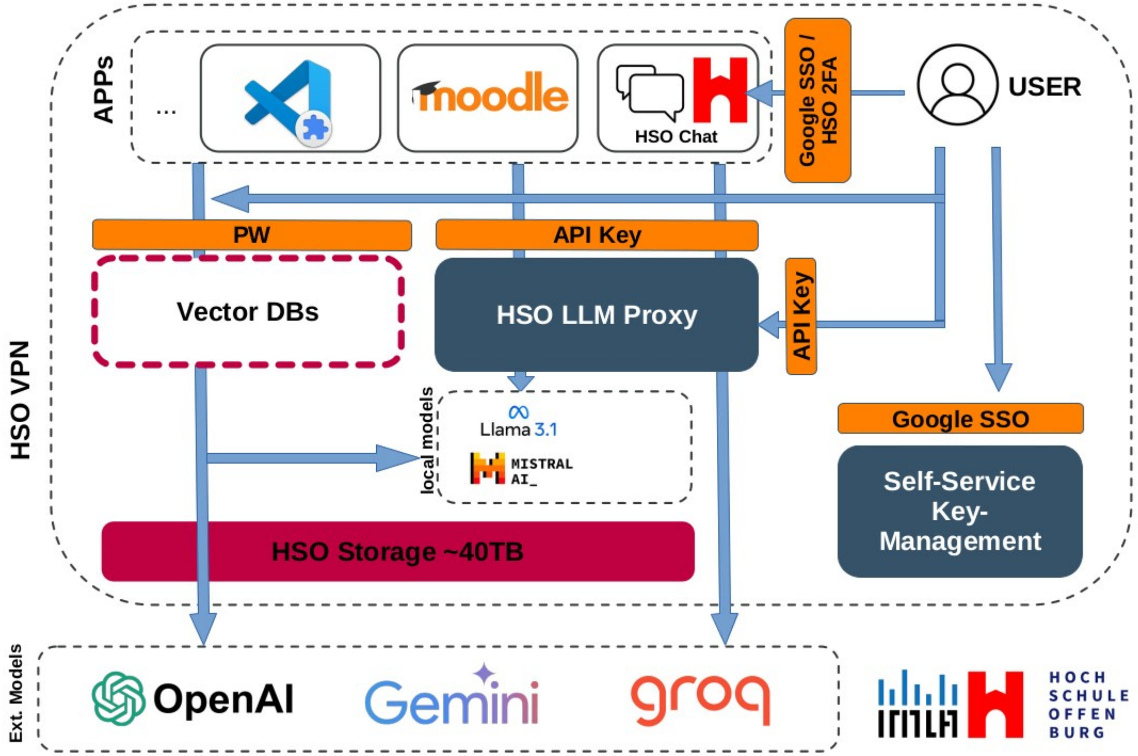
LLM Tutorial
Intro: Lecturer

INSTITUTE FOR MACHINE LEARNING AND ANALYTICS

https://imla.hs-offenburg.de/

https://www.keuper-labs.org/

HSO LLM Infrastructure:



https://llm-proxy.imla.hs-offenburg.de/info/index.html

LLM Tutorial
Organization

- Hands on part: use **your laptop** or the **pool computers** in front of you

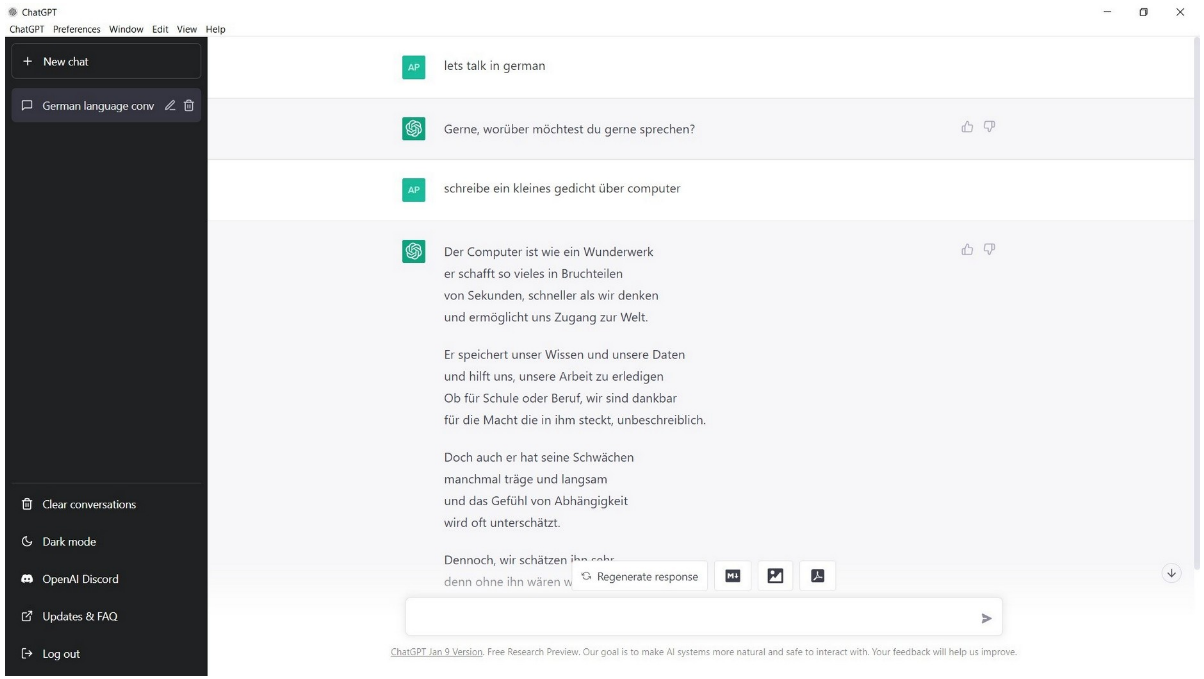
  - Pool Computer: boot any basic image you like (Windows or Ubuntu)
    - HSO users: use your HSO account
    - Login for external users:
      - User: ********
      - PW: ********

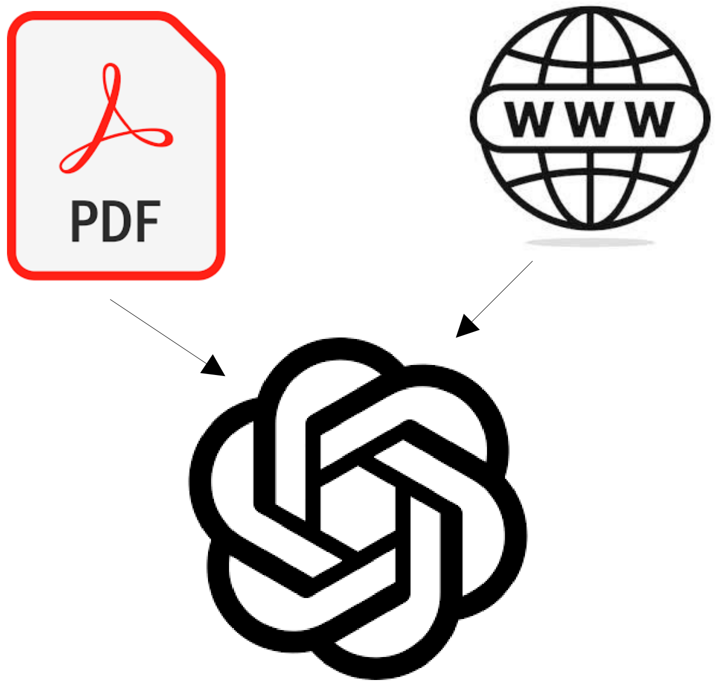
- Open Git-hub page in browser: ***https://github.com/keuperj/LLM_Tutorial_URAI24***

- **We use Colab for coding**
  - HSO users use HSO account
  - External users private Google account

LLM Tutorial
Goal

Get information from own data sources into a existing (trained) LLM so it can use it to answer questions





LLM Tutorial
Overview

Part I

Brief introduction to LLMs

Part II

Fine-Tuning OpenAI Models

Part III

A simple RAG Setup

LLM Tutorial
Some Notes Before We Start

- **Building LLM systems is complex task – a 3h tutorial can’t cover this**
    - We focus on basic understanding and first practical examples to get you started
- **We have participants with very different backgrounds**
    - The tutorial is necessarily a compromise between different needs

OpenAI

- **We will use the OpenAI Toolchain for practical reasons**
    - Basic principles directly apply to other providers and open source frameworks
    - HSO members can use this directly with the LLM-Proxy

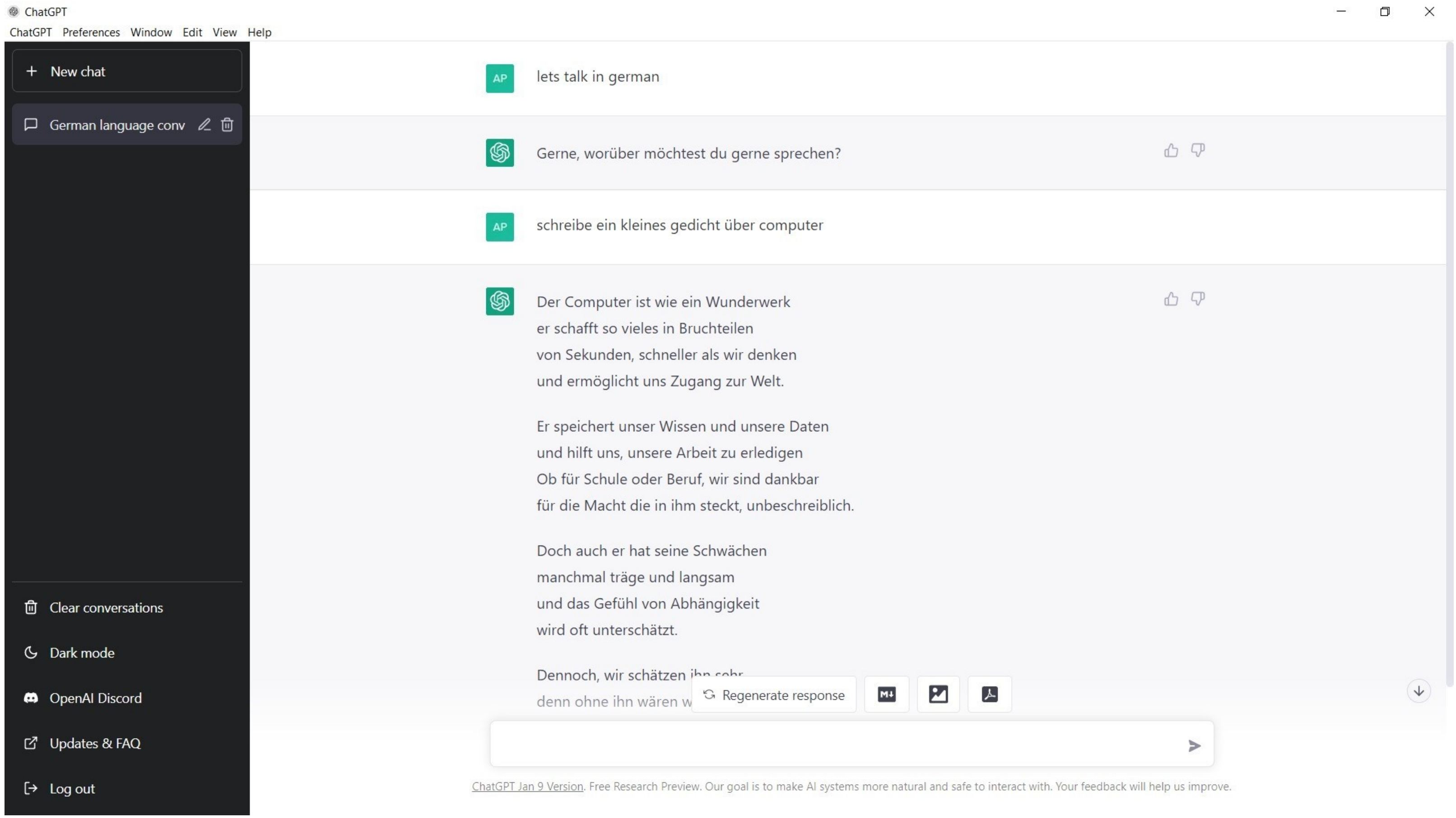
Part I: LLM Basics

Part I

Brief introduction
to LLMs

Part I: LLM Basics
GPT Models

GPT
Model



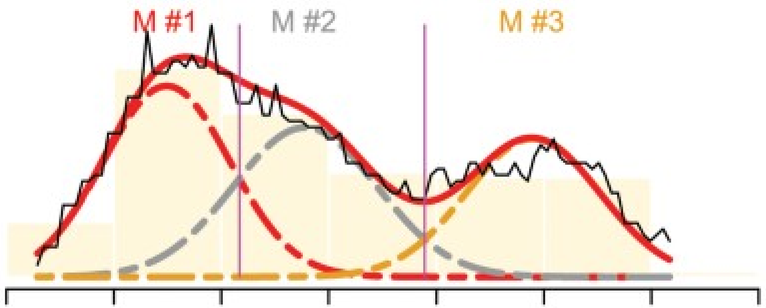
GPT → **G**enerative **P**re-Trained **T**ransformer

GPT
Model

GPT Models

GPT → Generative Pre-Trained Transformer

GPT Model



Learn complex distributions from data

Generation → sampling from distribution

Part I: LLM Basics
GPT Models

GPT → **G**enerative **P**re-Trained **T**ransformer

GPT Model

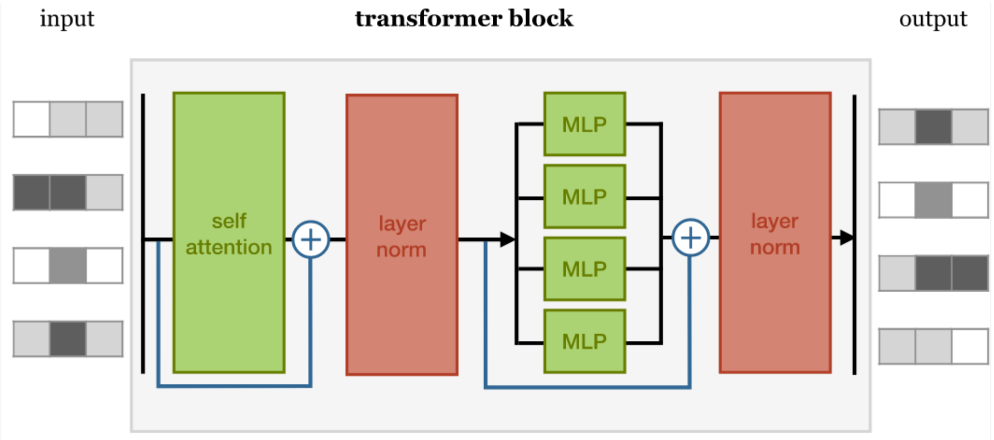
Optimal LLM Training Cost				
Model	**Size (# Parameters)**	**Tokens**	**GPU**	**Optimal Training Compute Cost**
MosaicML GPT-30B	30 Billion	610 Billion	A100	$ 325,855
Google LaMDA	137 Billion	168 Billion	A100	$ 368,846
Yandex YaLM	100 Billion	300 Billion	A100	$ 480,769
Tsinghua University Zhipu.AI GLM	130 Billion	400 Billion	A100	$ 833,333
Open AI GPT-3	175 Billion	300 Billion	A100	$ 841,346
AI21 Jurassic	178 Billion	300 Billion	A100	$ 855,769
Bloom	176 Billion	366 Billion	A100	$ 1,033,756
DeepMind Gopher	280 Billion	300 Billion	A100	$ 1,346,154
DeepMind Chinchilla	70 Billion	1,400 Billion	A100	$ 1,745,014
MosaicML GPT-70B	70 Billion	1,400 Billion	A100	$ 1,745,014
Nvidia Microsoft MT-NLG	530 Billion	270 Billion	A100	$ 2,293,269
Google PaLM	540 Billion	780 Billion	A100	$ 6,750,000

Huge Data Sets

Part I: LLM Basics
GPT Models

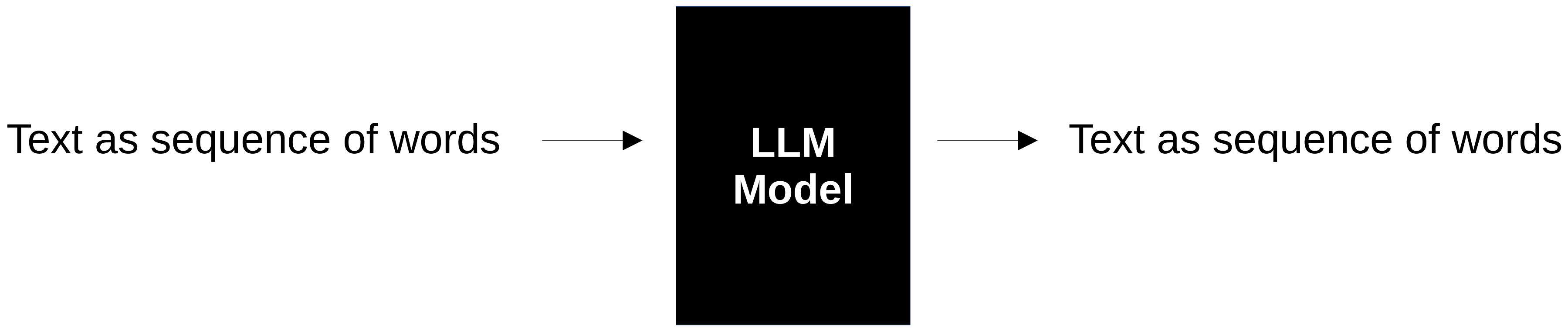
GPT → Generative Pre-Trained Transformer

GPT Model



Neural Network Architecture

Part I: LLM Basics
Text Generation: Sequence to Sequence Mapping



Part I: LLM Basics
How do LLMs learn to generate Text?

Basic idea: **use context**

Text example: **which words can we fill in the blank?**

“The ________ is climbing on the tree...“

Part I: LLM Basics
How do LLMs learn to generate Text?

Basic idea: **use context**

Text example: **which words can we fill in the blank?**

“The ________ is climbing on the tree…“

More Context: ***“...His sister is 10 years old“***

Part I: LLM Basics
How do LLMs learn to generate Text?

Basic idea: **use context**

Text example: **which words can we fill in the blank?**

Predict probability of words

youngster
lad
Guy
Boy

"The ________ is climbing on the tree…"

…

More Context: ***"…His sister is 10 years old"***

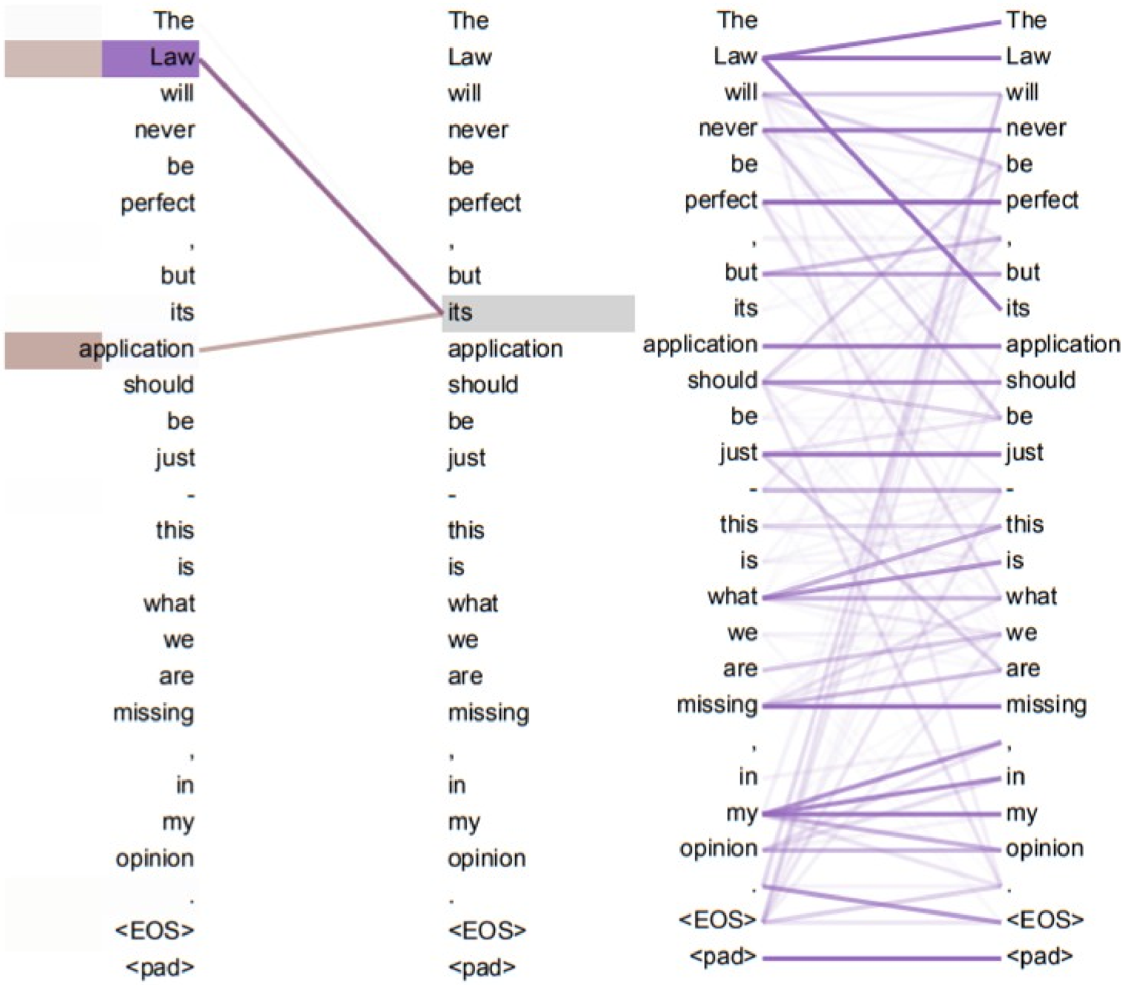
INSTITUTE FOR MACHINE LEARNING AND ANALYTICS

Part I: LLM Basics
How to model Context?

What is the meaning of A word?

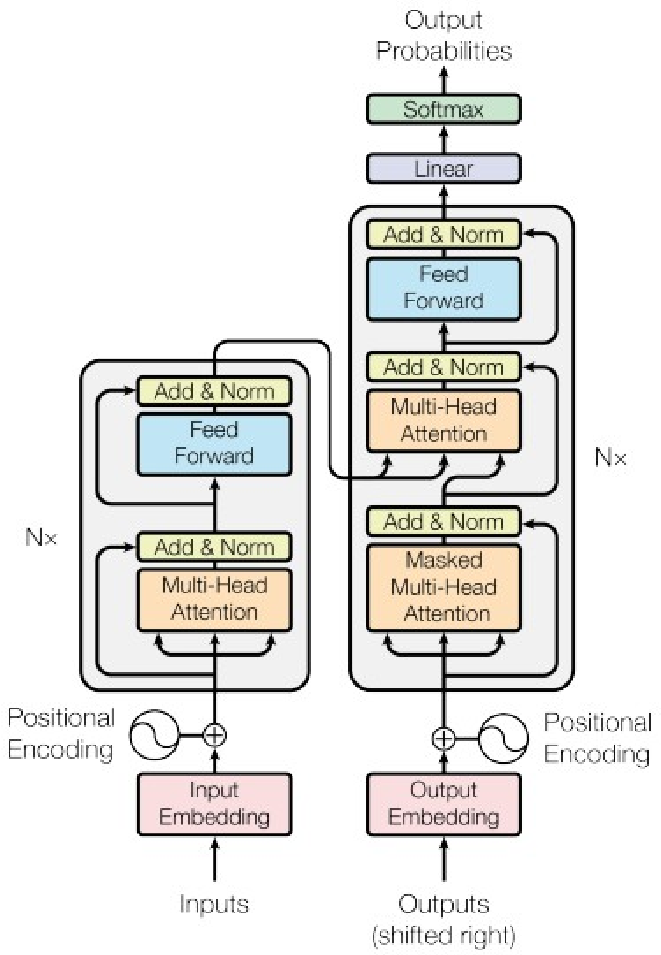
How is it related to other Words in its context?

→ better performance by modeling context relations



INSTITUTE FOR MACHINE LEARNING AND ANALYTICS

Part I: LLM Basics
Attention



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 41.0 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature.

1 Introduction

Recurrent neural networks, long short-term memory [12] and gated recurrent [7] neural networks in particular, have been firmly established as state of the art approaches in sequence modeling and transduction problems such as language modeling and machine translation [29, 2, 5]. Numerous efforts have since continued to push the boundaries of recurrent language models and encoder-decoder architectures [31, 21, 13].

Part I: LLM Basics
Attention

self-attention with key, query and value

Sequence-to-sequence mapping

$$x_1, x_2, \ldots, x_t \longrightarrow y_1, y_2, \ldots, y_t$$

$k \times k$ **Linear transform matrices**

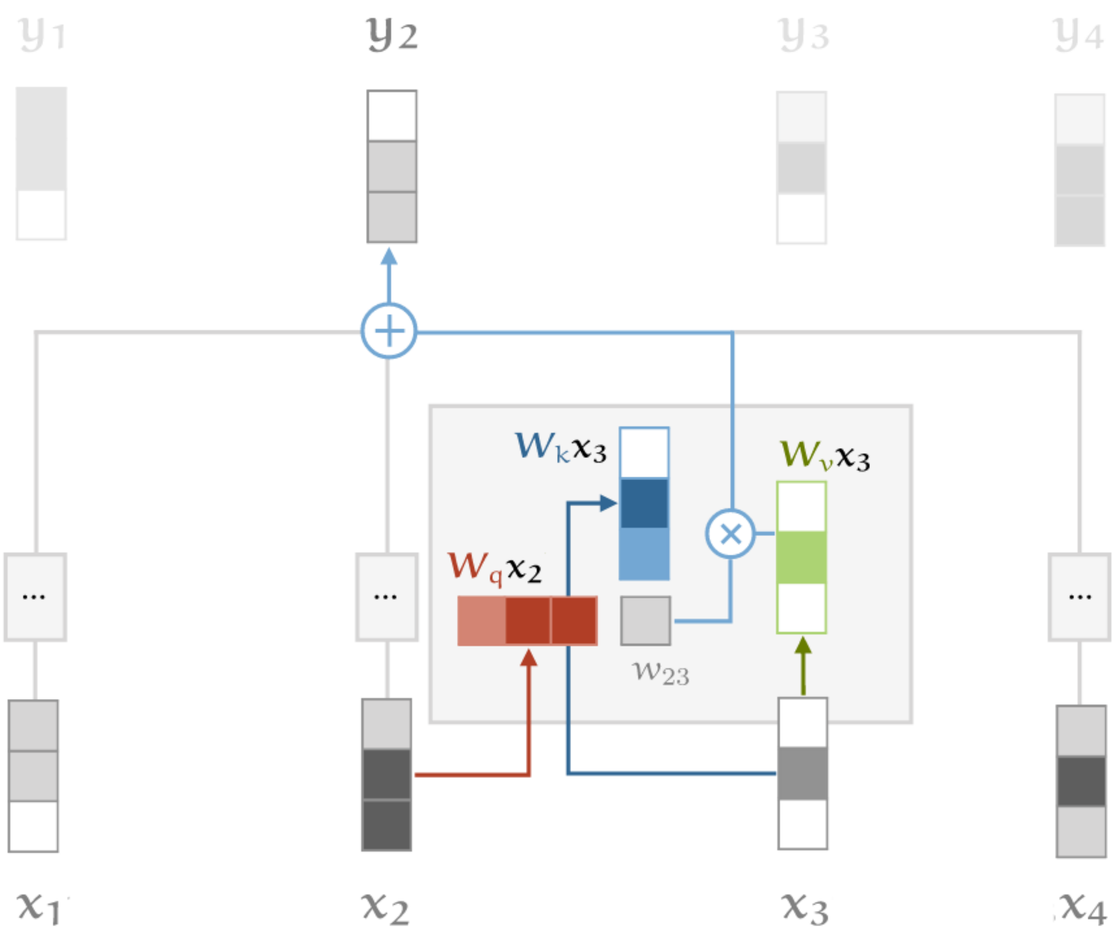
$$q_i = W_q x_i \qquad k_i = W_k x_i \qquad v_i = W_v x_i$$

$$w'_{ij} = {q_i}^T k_j$$

$$w_{ij} = \text{softmax}(w'_{ij})$$

$$y_i = \sum_j w_{ij} v_j \, .$$

Scaling $w'_{ij} = \dfrac{{q_i}^T k_j}{\sqrt{k}}$



Part I: LLM Basics

Attention

$$w'_{ij} = \mathbf{x}_i{}^T \mathbf{x}_j \,.$$

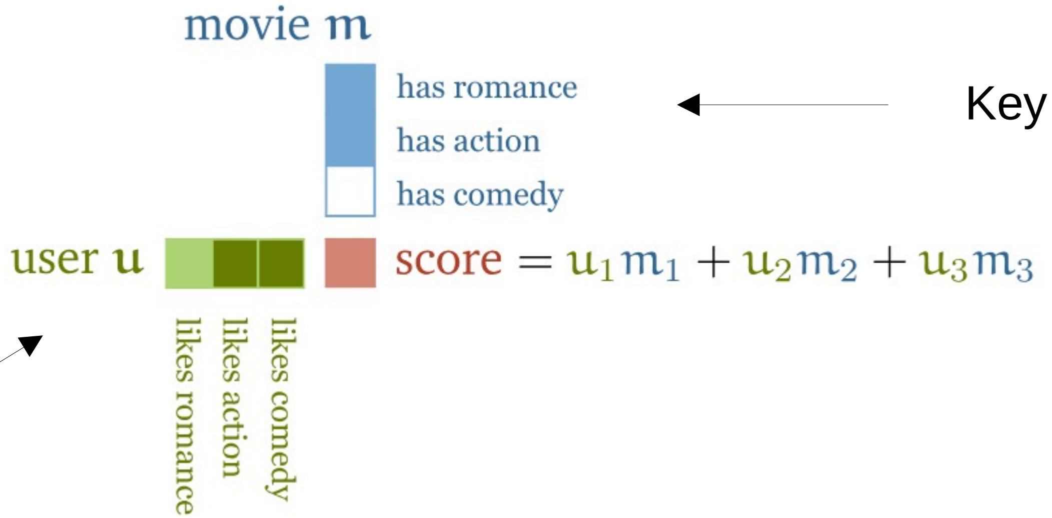
Dot-product as similarity measure

SOFTMAX

$$w_{ij} = \frac{\exp w'_{ij}}{\sum_j \exp w'_{ij}} \,. \longrightarrow [0, 1]$$

self-attention with key, query and value

Analogy to Recommender Systems



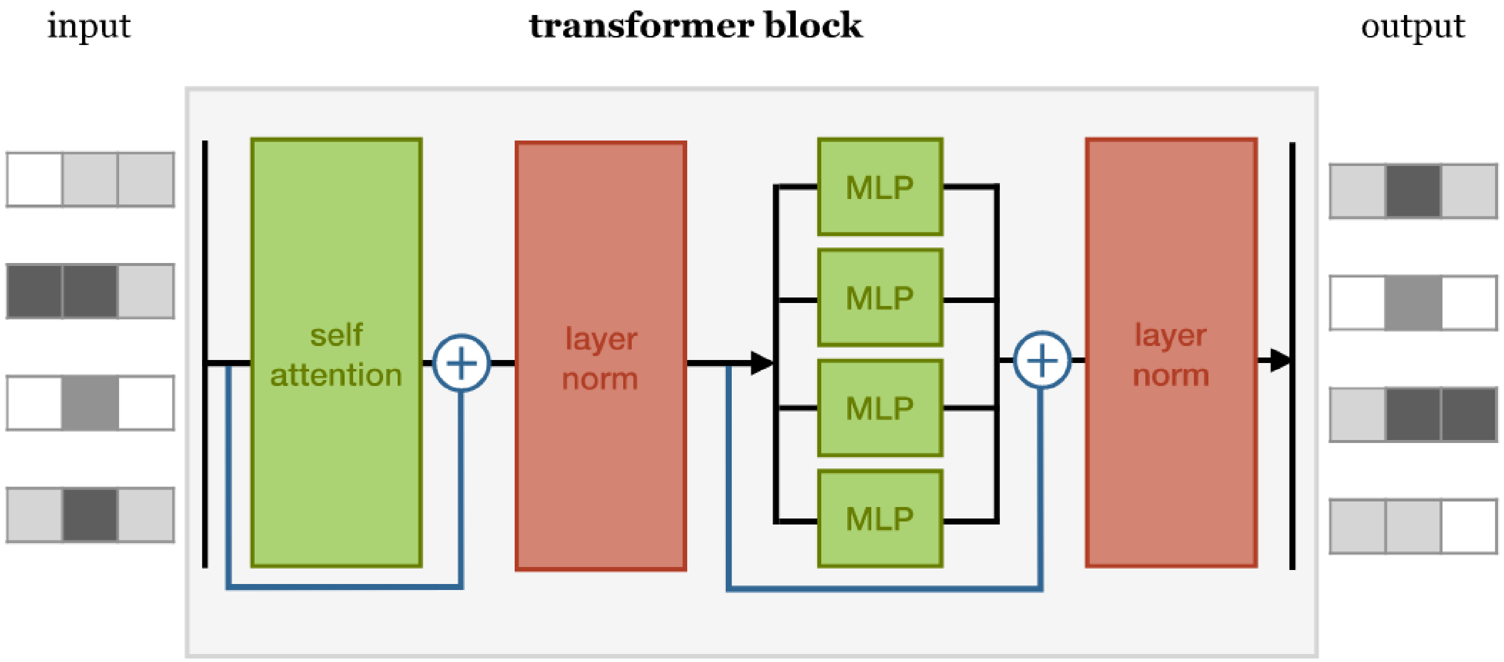
Key

Query

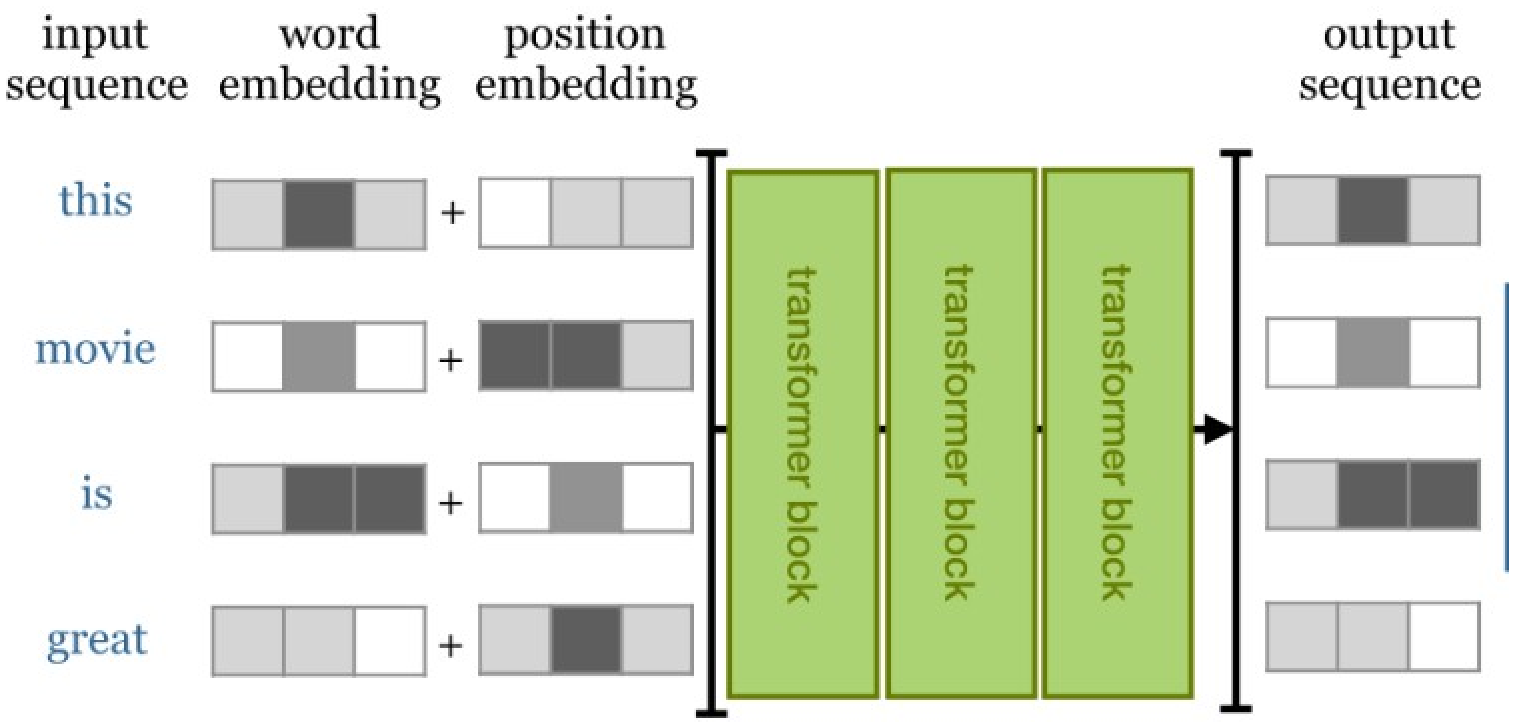
INSTITUTE FOR MACHINE LEARNING AND ANALYTICS

Part I: LLM Basics
Transformer Networks

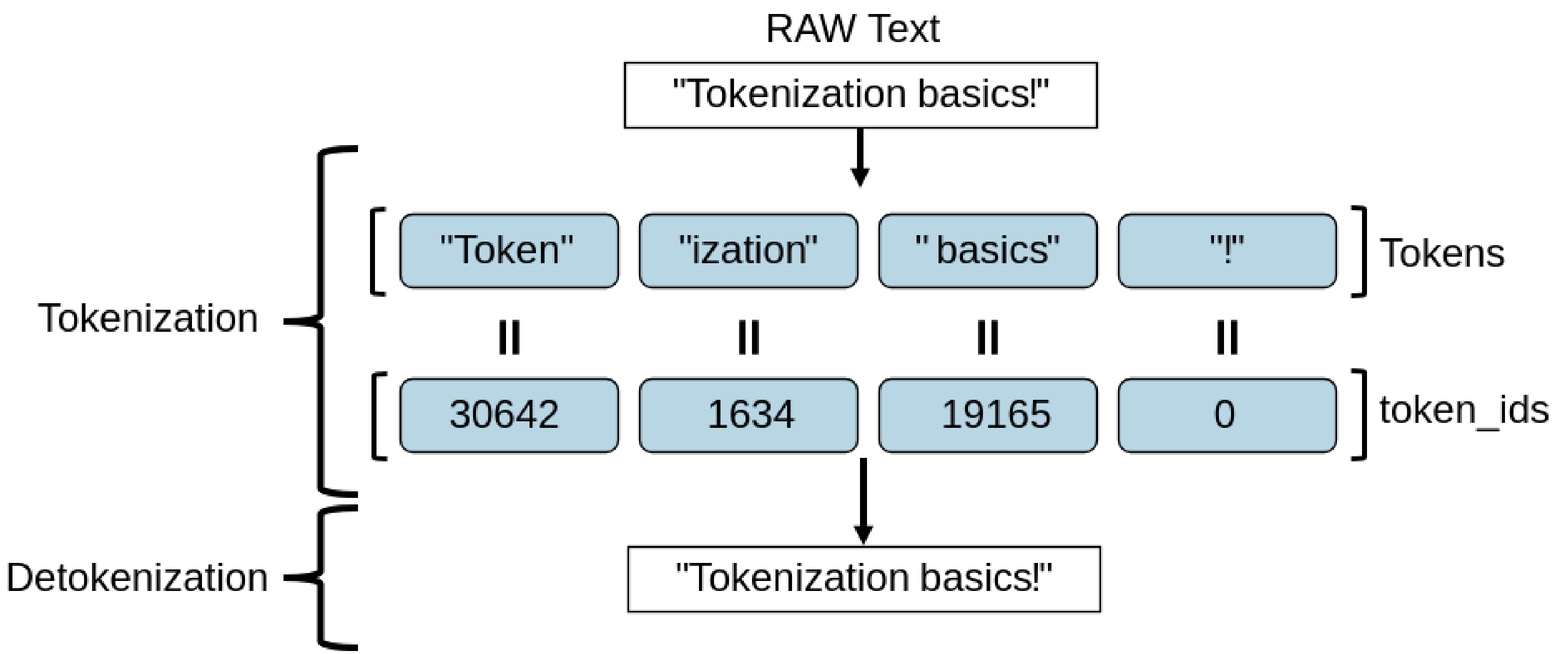
> "Any architecture designed to process a connected set of units—such as the tokens in a sequence or the pixels in an image—where the only interaction between units is through self-attention."



Part I: LLM Basics
Transformer Networks

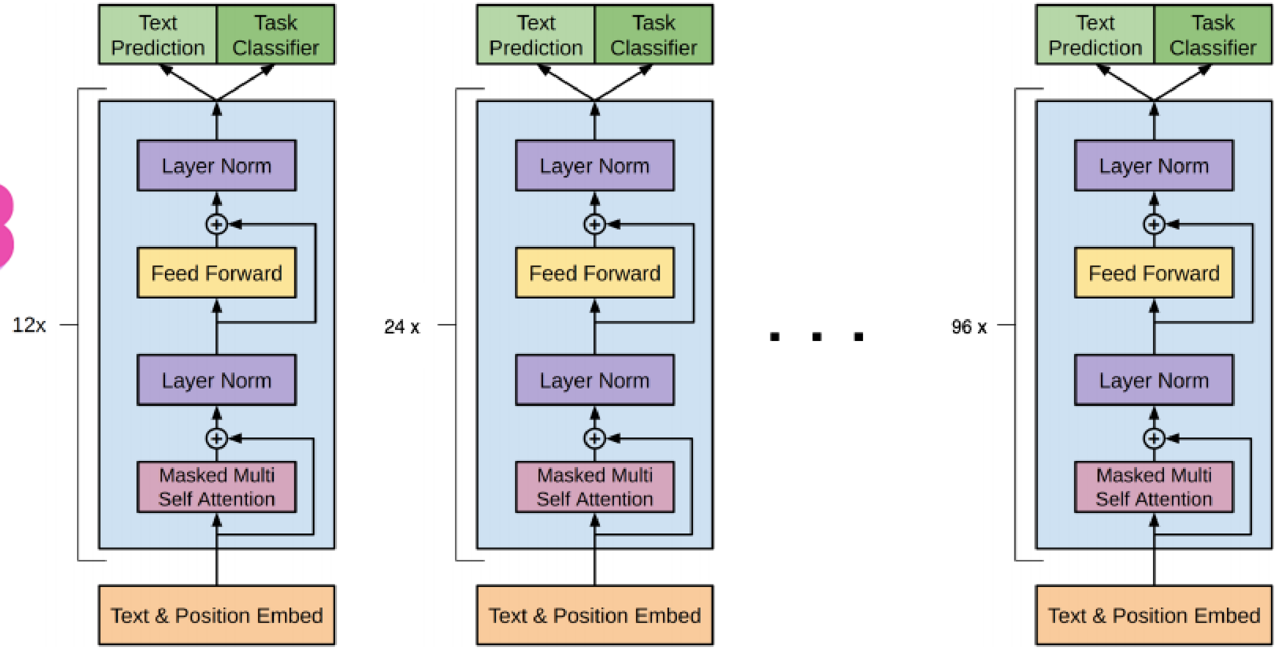


Part I: LLM Basics
Tokenization



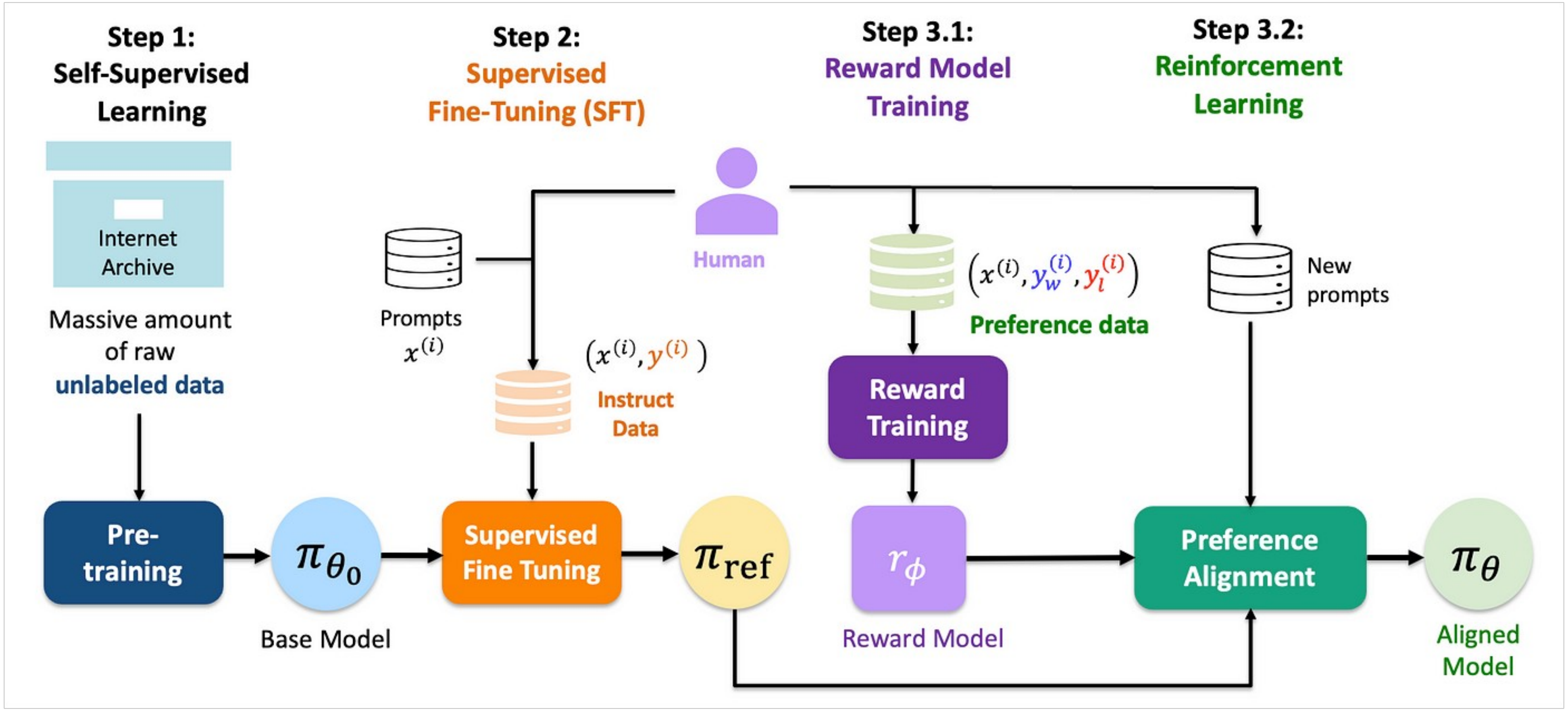
- Map text to sequence of vectors
- Token <= word
- Tokenizer is a own (statistical) model trained on data
- Unknown words are split into small tokens
- Tokenizer must fit model!

Part I: LLM Basics
Example: GPT-3 Architecture



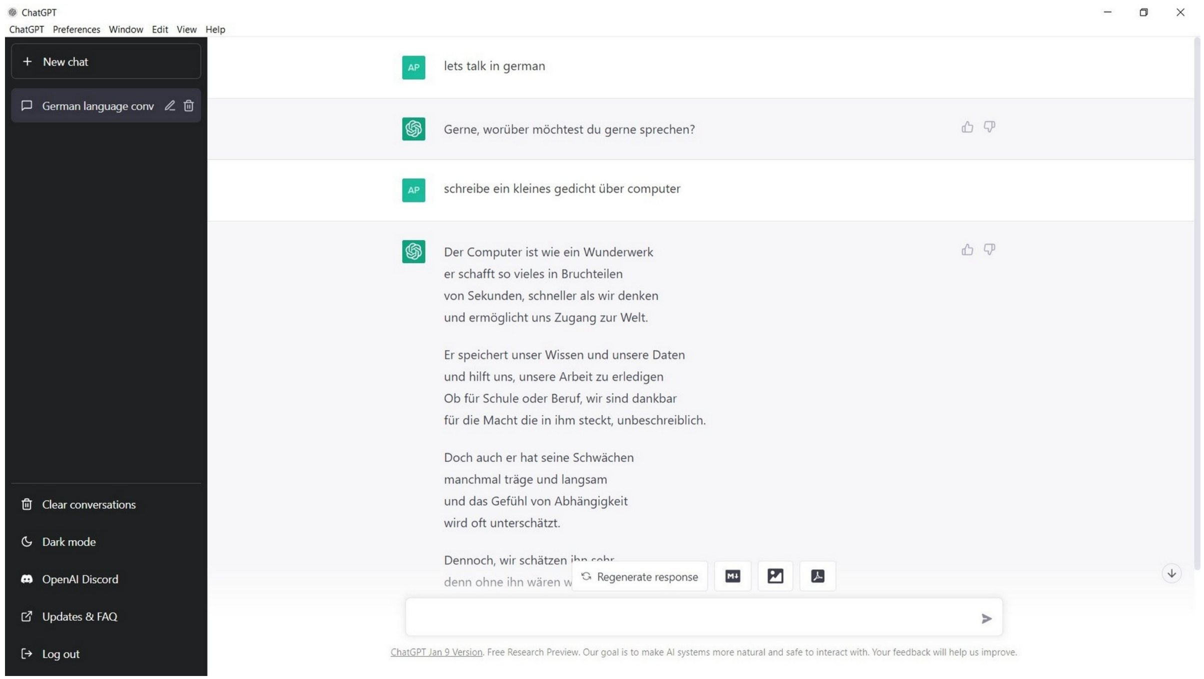
Model Name	n_{params}	n_{layers}	d_{model}	n_{heads}	d_{head}	Batch Size	Learning Rate
GPT-3 Small	125M	12	768	12	64	0.5M	6.0×10^{-4}
GPT-3 Medium	350M	24	1024	16	64	0.5M	3.0×10^{-4}
GPT-3 Large	760M	24	1536	16	96	0.5M	2.5×10^{-4}
GPT-3 XL	1.3B	24	2048	24	128	1M	2.0×10^{-4}
GPT-3 2.7B	2.7B	32	2560	32	80	1M	1.6×10^{-4}
GPT-3 6.7B	6.7B	32	4096	32	128	2M	1.2×10^{-4}
GPT-3 13B	13.0B	40	5140	40	128	2M	1.0×10^{-4}
GPT-3 175B or "GPT-3"	175.0B	96	12288	96	128	3.2M	0.6×10^{-4}

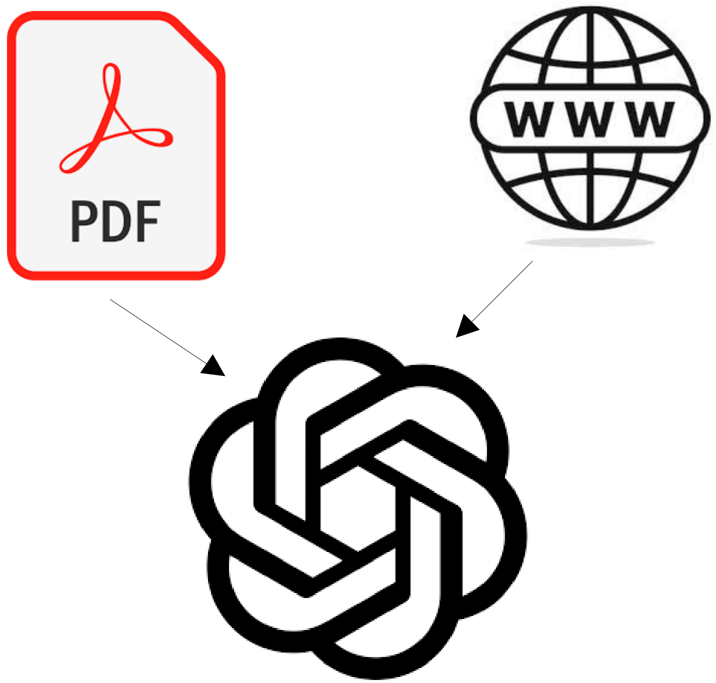
Part I: LLM Basics
LLM Training in a Nutshell



Part I: LLM Basics
Goal

Get information from own data sources into a existing (trained) LLM so it can use it to answer questions





Part I: LLM Basics
How to get your data into the LLM?

Train from Scratch

+ best performance
-- very expensive
-- very complicated

Fine-Tuning

Continue to train a Pre-trained model

+ good performance
+- cost depends on data size
- can harm model

Add Context

Add your data to the model input

+ model does not change
+ simple
- increase query cost
-- small context window

Part I: LLM Basics

How to get your data into the LLM?

Train from Scratch

+ best performance
-- very expensive
-- very complicated

Fine-Tuning

Continue to train a Pre-trained model

+ good performance
+- cost depends on data size
- can harm model

Add Context

Add your data to the model input

+ model does not change
+ simple
- increase query cost
-- small context window

Part II: Fine-Tuning

Part II

Fine-Tuning

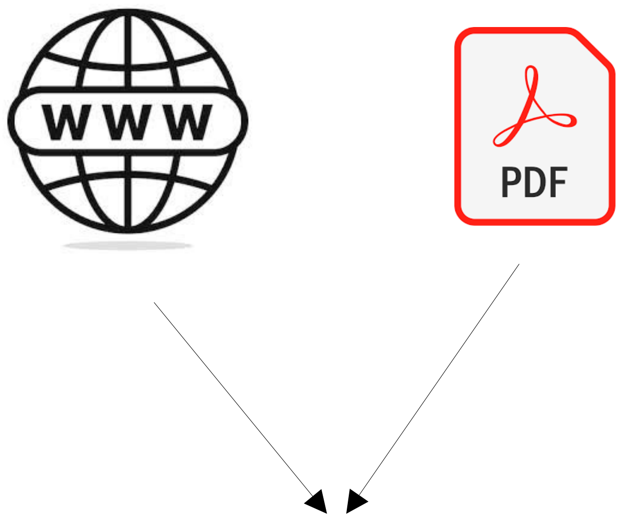
Part II: Fine-Tuning
What do we need?

Pre-trained LLM Model

GPT-4o mini

Criteria	GPT-4o mini	GPT-4o	GEMINI 1.5	CLAUDE 3.5 SONNET	LLAMA 3 (8B)	GPT-4 TURBO
Release Date	July 2024	May 2024	Feb. 2024	June 2024	Apr. 2024	Nov. 2023
Key Feature	Cost-effective, efficient AI	Multimodal (text, image, etc.)	Factual language updates	Best for creative works	Advanced NLP, complex queries	Larger, faster, more accurate
Output Cost	$0.15 / 1M Tokens	$15.00 / 1M Tokens	$1.05 / 1M Tokens	$75.00 / 1M Tokens	$0.1 / 1M Tokens	$30.00 / 1M Tokens
Prompt Inputs	Text, Images	Text, Image, Audio, and Video	Text, Images	Text, Images	Text, Images	Text, Images and Text-to-Speech
Quality Index	85	100	76	98	65	94
Speed	166 t/s	75 t/s	156 t/s	79 t/s	240 t/s	23 t/s

DATA



Extract + per-process

INSTITUTE FOR MACHINE LEARNING AND ANALYTICS

Part II: Fine-Tuning
Excursion: getting Web Data

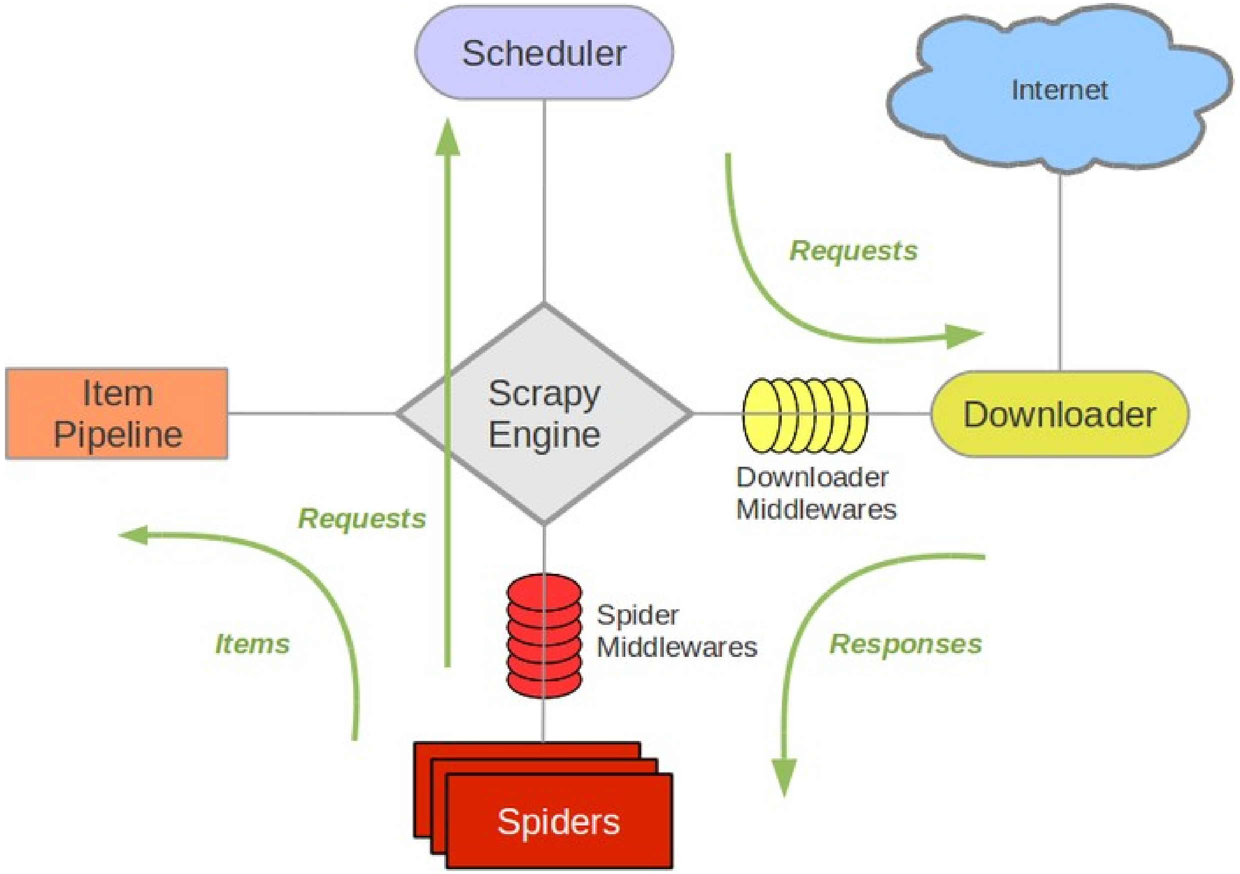
```python
from pathlib import Path

import scrapy


class QuotesSpider(scrapy.Spider):
    name = "quotes"

    def start_requests(self):
        urls = [
            "https://quotes.toscrape.com/page/1/",
            "https://quotes.toscrape.com/page/2/",
        ]
        for url in urls:
            yield scrapy.Request(url=url, callback=self.parse)

    def parse(self, response):
        page = response.url.split("/")[-2]
        filename = f"quotes-{page}.html"
        Path(filename).write_bytes(response.body)
        self.log(f"Saved file {filename}")
```



https://scrapy.org/

Part II: Fine-Tuning
Excursion: processing PDFs

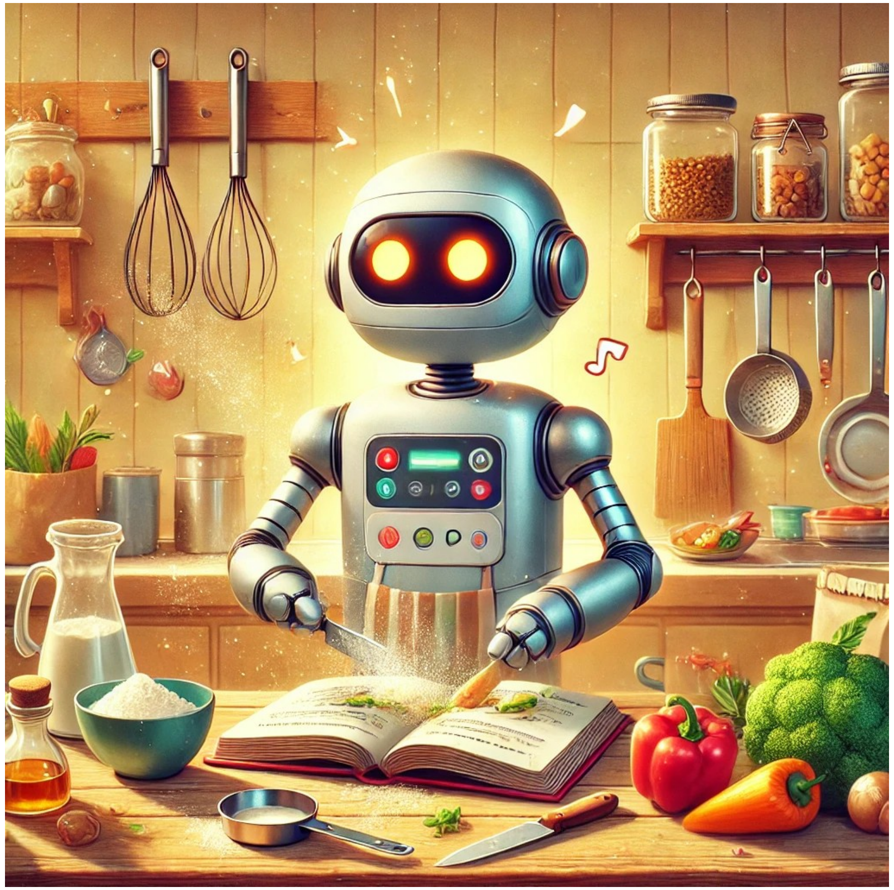
PyMuPDF

MuPDF with Python

```python
import pymupdf # imports the pymupdf library
doc = pymupdf.open("example.pdf") # open a document
for page in doc: # iterate the document pages
  text = page.get_text() # get plain text encoded as UTF-8
```

https://github.com/pymupdf/PyMuPDF

Part II: Fine-Tuning
Hands on...



Fine-Tuning GPT-4o-mini to learn recipes ...

Part III: RAG

How to get your data into the LLM?

Train from Scratch

+ best performance
-- very expensive
-- very complicated

Fine-Tuning

Continue to train a Pre-trained model

+ good performance
+- cost depends on data size
- can harm model

Add Context

Add your data to the model input

+ model does not change
+ simple
- increase query cost
-- small context window

Part III: RAG
How to add Context to your Query?

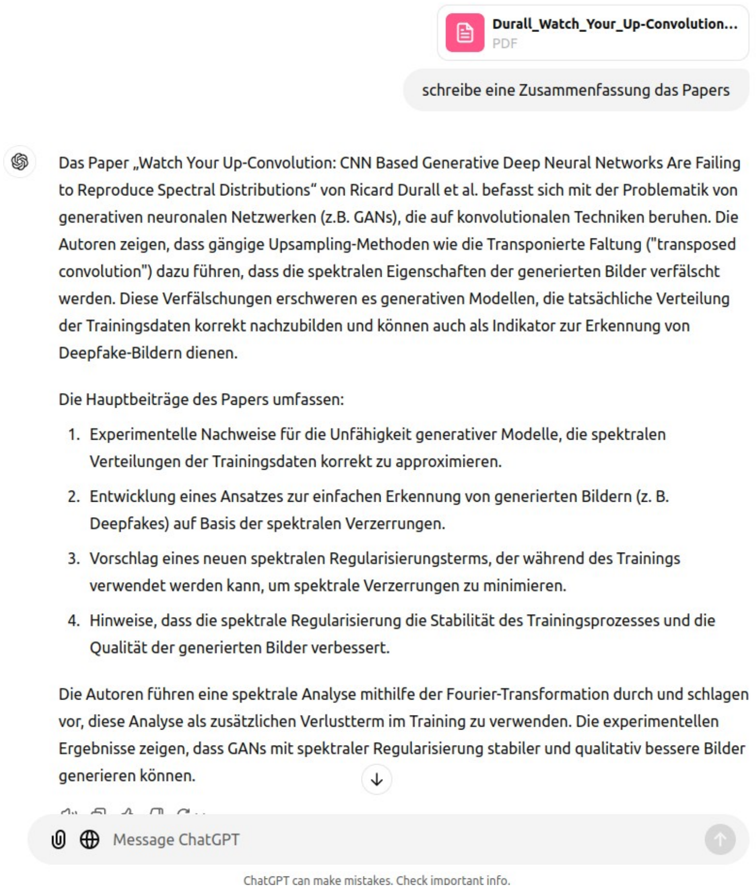
ChatGPT: File upload (pro version)

GPT-4o mini

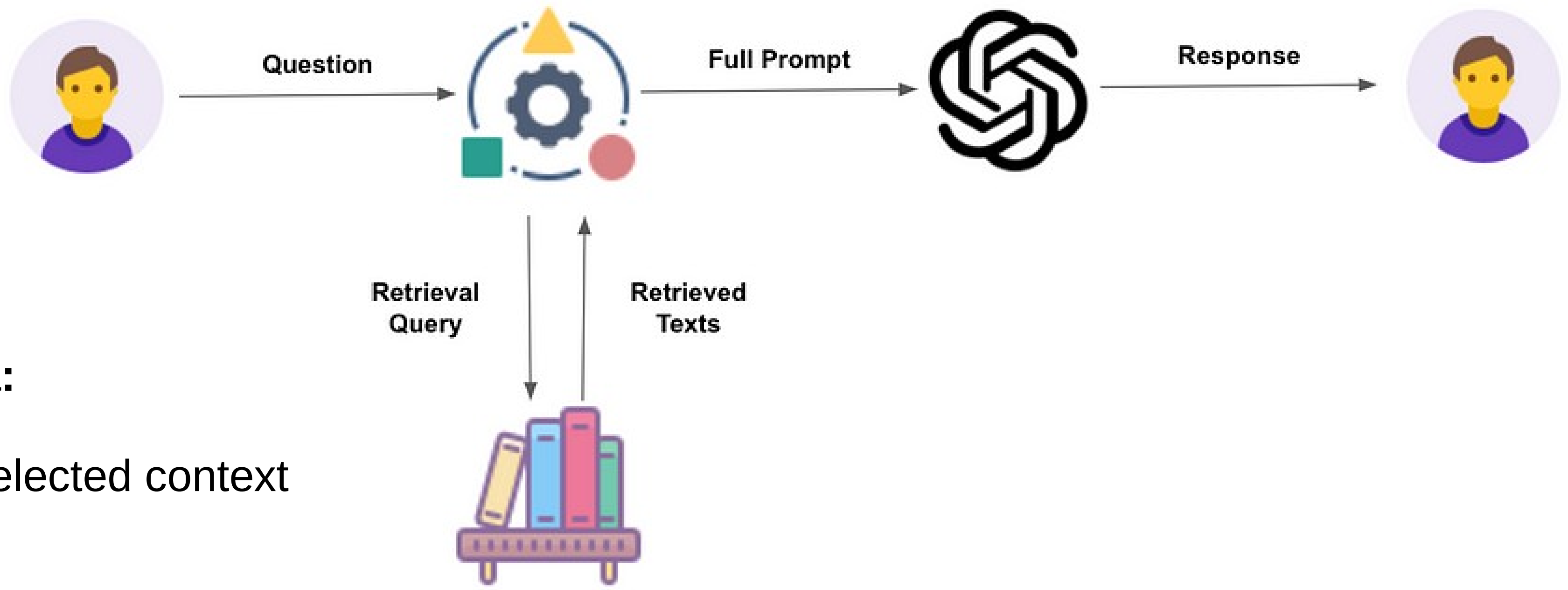
```
[{'content': 'You are a helpful recipe assistant. You are to extract the '
             'generic ingredients from each of the recipes provided.',
  'role': 'system'},
 {'content': 'Title: Beef Brisket\n'
             '\n'
             'Ingredients: ["4 lb. beef brisket", "1 c. catsup", "1 c. water", '
             '"1/2 onion, minced", "2 Tbsp. cider vinegar", "1 Tbsp. prepared '
             'horseradish", "1 Tbsp. prepared mustard", "1 tsp. salt", "1/2 '
             'tsp. pepper"]\n'
             '\n'
             'Generic ingredients: ',
  'role': 'user'}]
```

+ Simple
-- API does not support PDFs – just text
→ need to pre-process PDFs on your own

-- very expensive for large number of documents
-- limited by window size of LLM



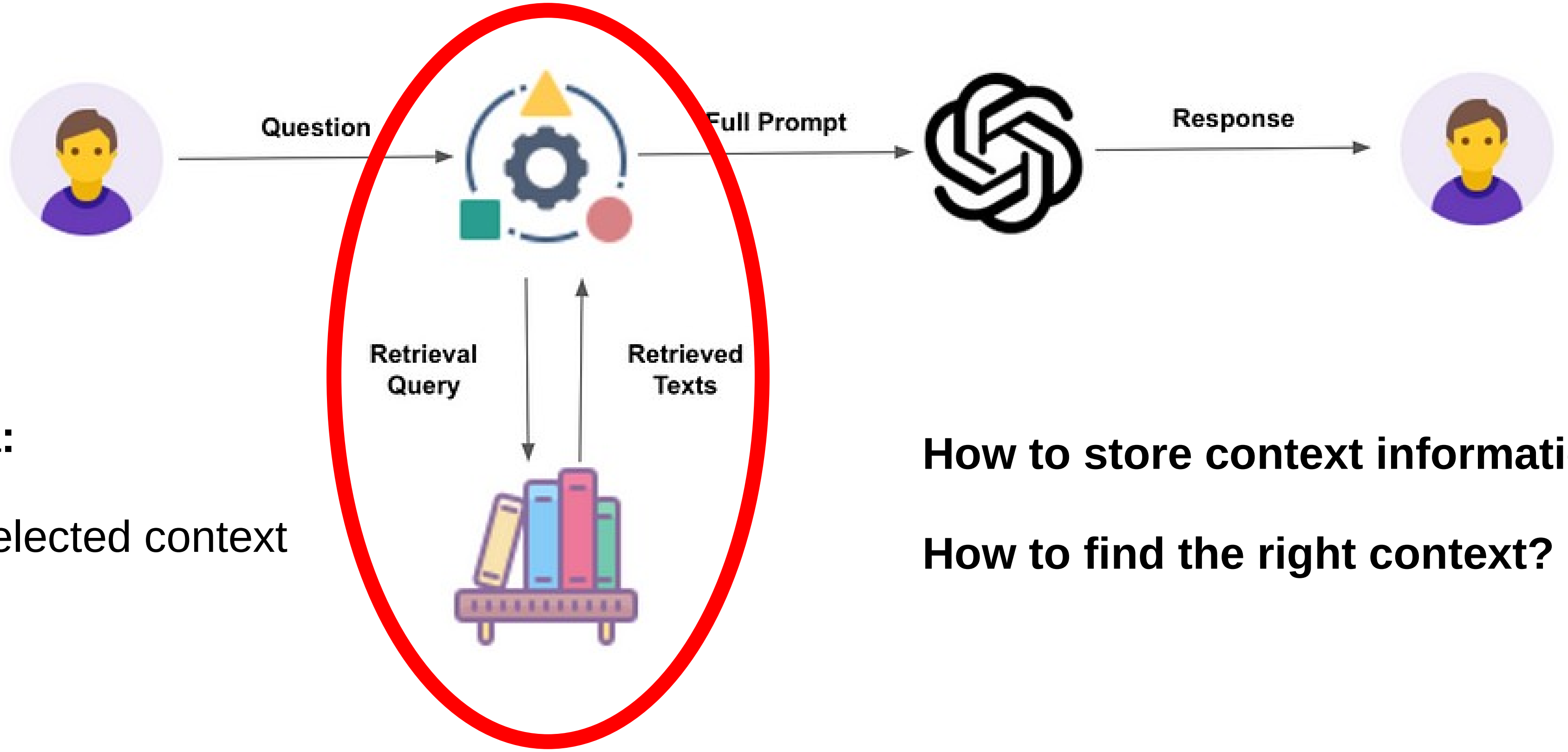
Part III: RAG
Retrieval Augmented Generation:



Basic Idea:

Add only selected context

Part III: RAG
Basics



Basic Idea:

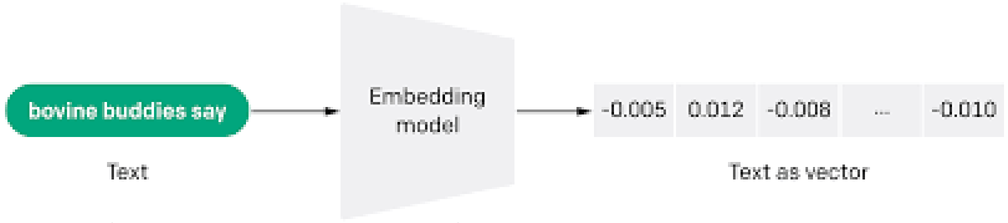
Add only selected context

How to store context information?

How to find the right context?

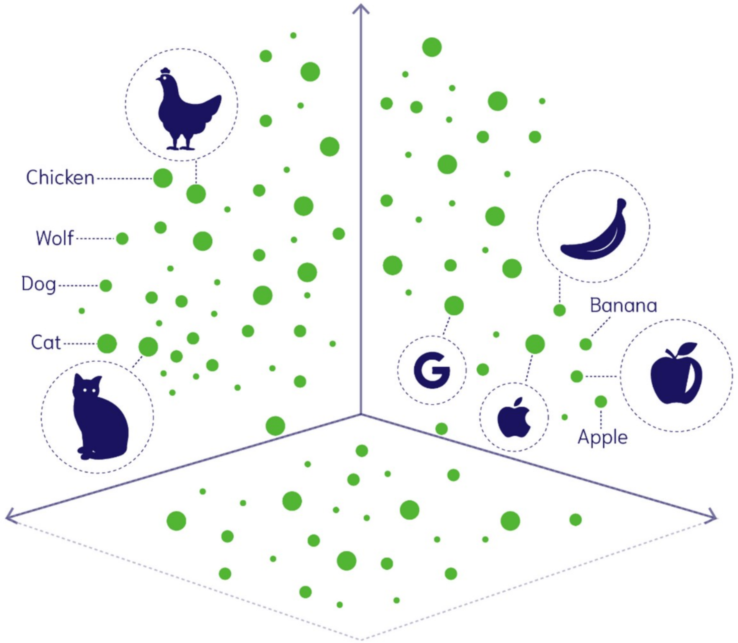
Part III: RAG
Embeddings

How to store context information?



Chunks of text

Yet another transformer



Text embedding vector space

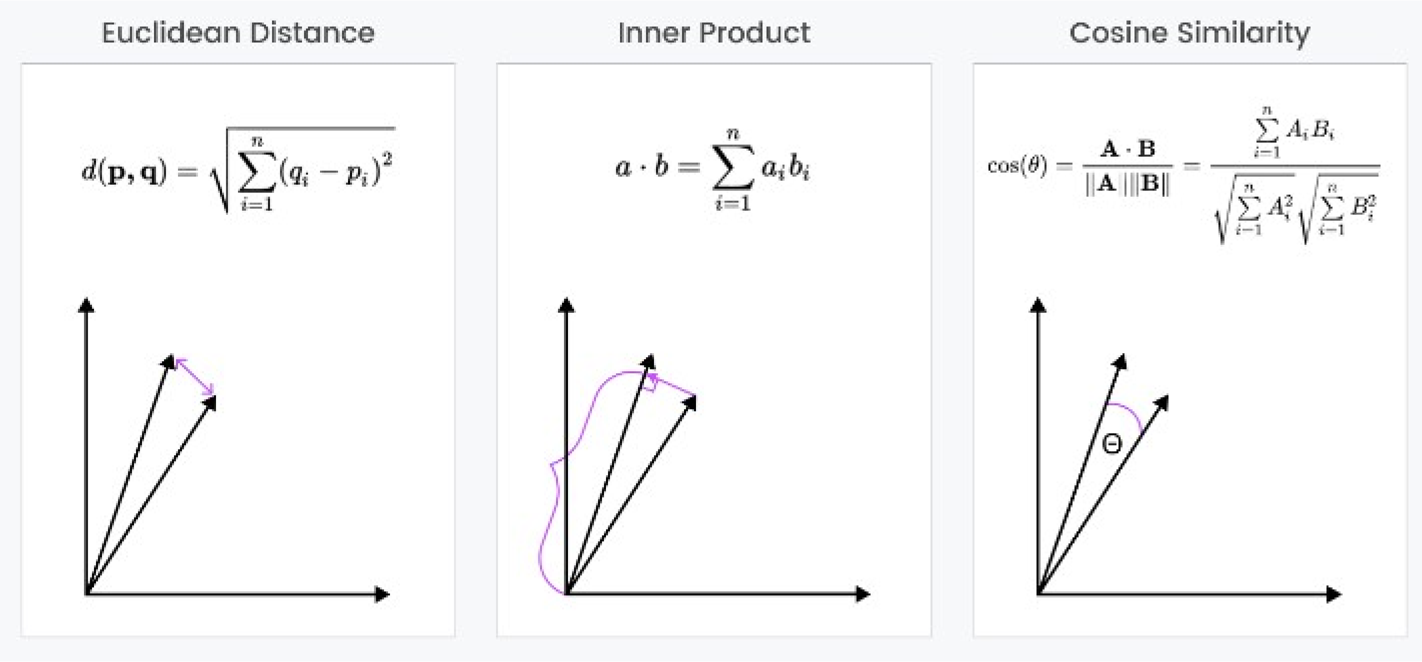
→ Vector Databases

Part III: RAG
Embeddings

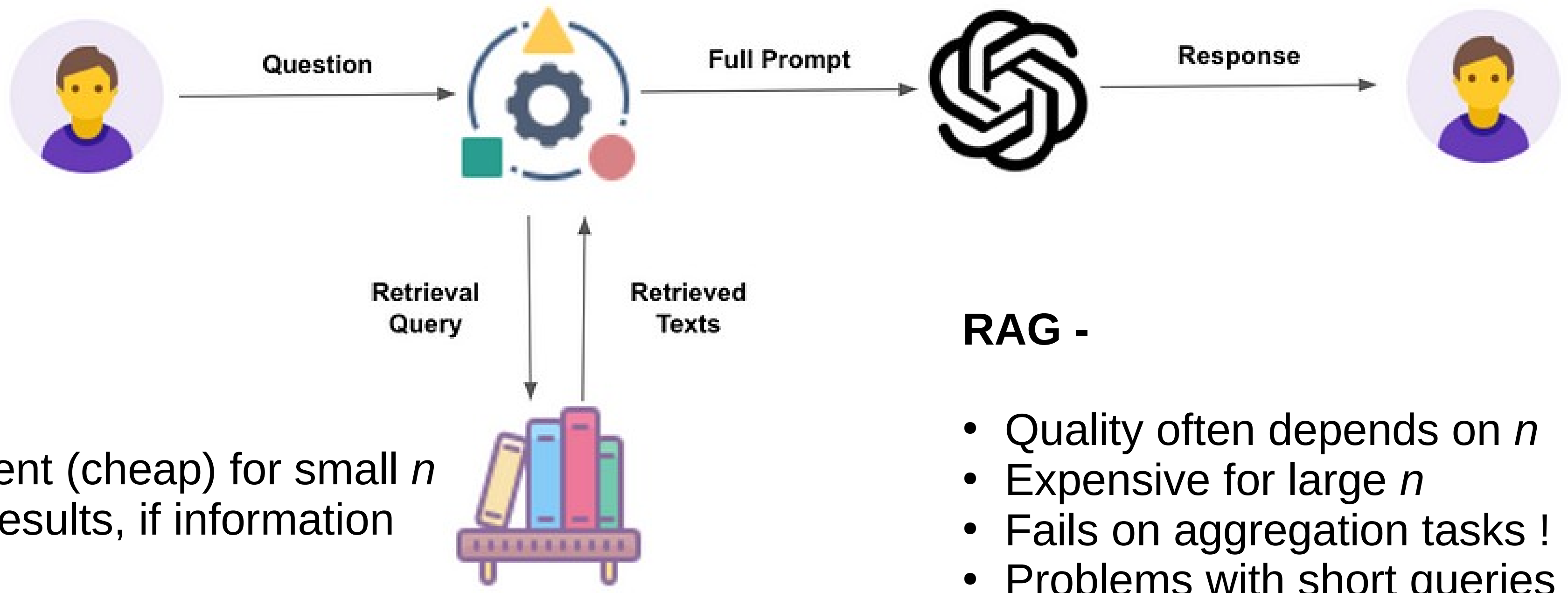
How to find the right context?

Basic Idea:

Find *n* nearest neighbors in vector space



Part III: RAG
Basics



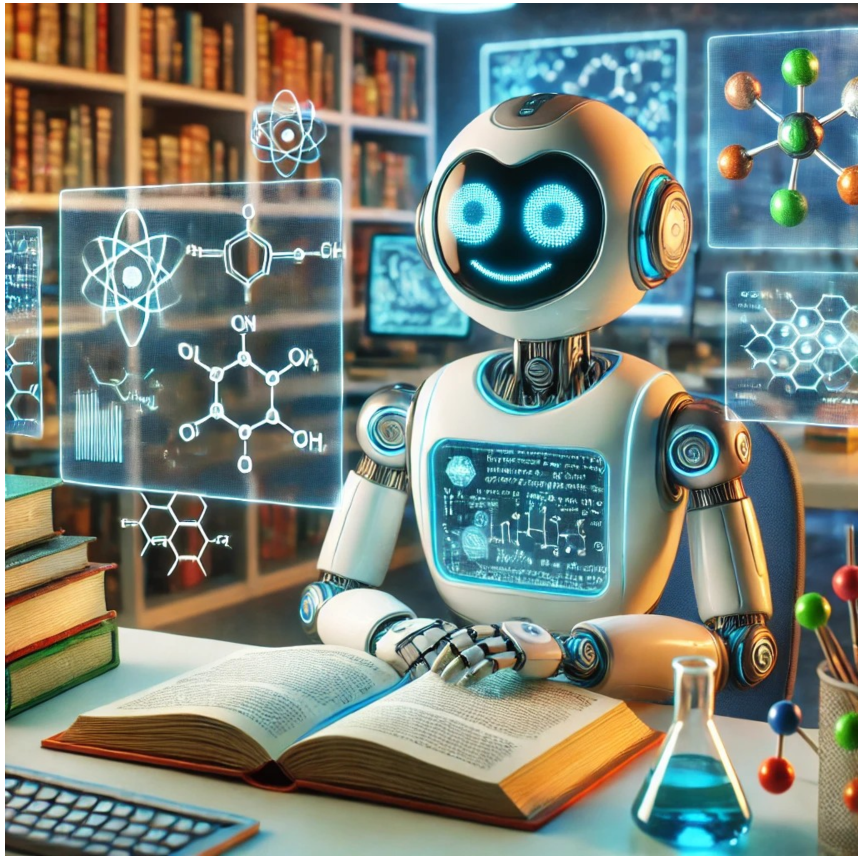
RAG +

- Simple
- Token efficient (cheap) for small *n*
- Very good results, if information is retrieved
- Source references via meta data
- Works well with small LLMs

RAG -

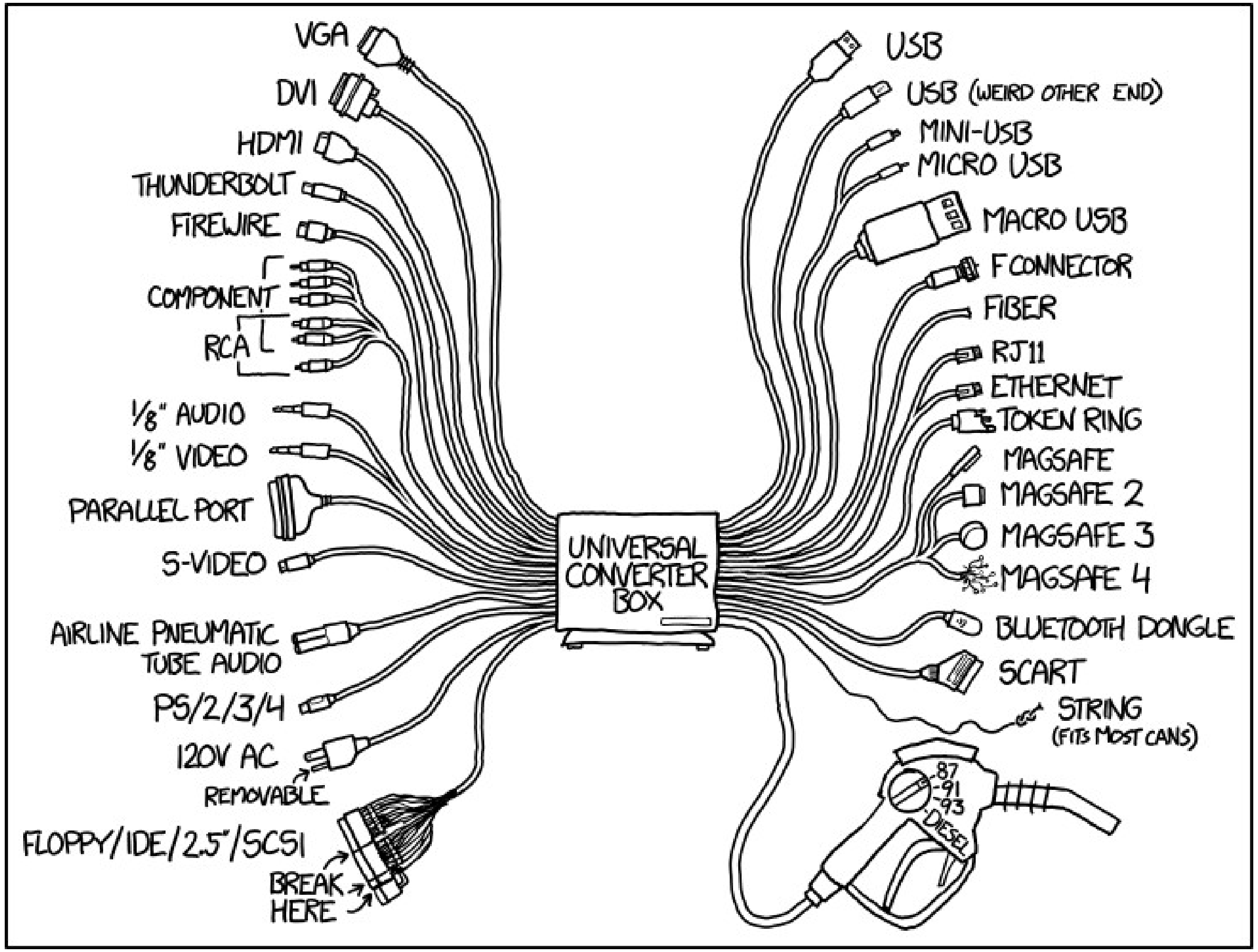
- Quality often depends on *n*
- Expensive for large *n*
- Fails on aggregation tasks !
- Problems with short queries → hybrid search

Part III: RAG
Hands on…



Augment GPT-3.5 with science facts

LLM Tutorial
Discussion



[xkcd]