TCS AI Fridays Regional Round

Preparation Meetings
21/25 November 2025

Powered by TCS Ahmedabad AI Chapter

Agenda, Availability Confirmation

- **Briefing # 1**
  - 21st Nov
  - 3-4 PM IST
  - In Person Vivacious room

- **Briefing # 2**
  - 25th Nov
  - 10 AM - 12:30 PM IST
  - In Person MPH

- **Regional Finals Event** -
  - 5th and 6th Dec (Fri-Sat, both days inclusive) -
  - 15-16 hour format

  - **Availability Confirmation**

21st Nov – Friday – 1 Hour – 3 PM to 4 PM

- Draft Agenda
- Availability Confirmation
- Channel
- Mentimeter
- AI Literacy
- Anatomy of good solution
- Problem Definition – 2 problem statements

25st Nov – Tue – 10 AM t 12:30 AM

- Individual Presentations – Solution Architecture – 5 Mins each
- Mentimeter
- Cumulative jury, audience feedback and Q/A – 45 mins

Mentimeter

What has been your / your solution's unique selling point (USP)?

Slido.com - #2386271

Mentimeter

What has been your / your solution's biggest improvement area?

Slido.com - #2386271

Revisiting GenAI concepts

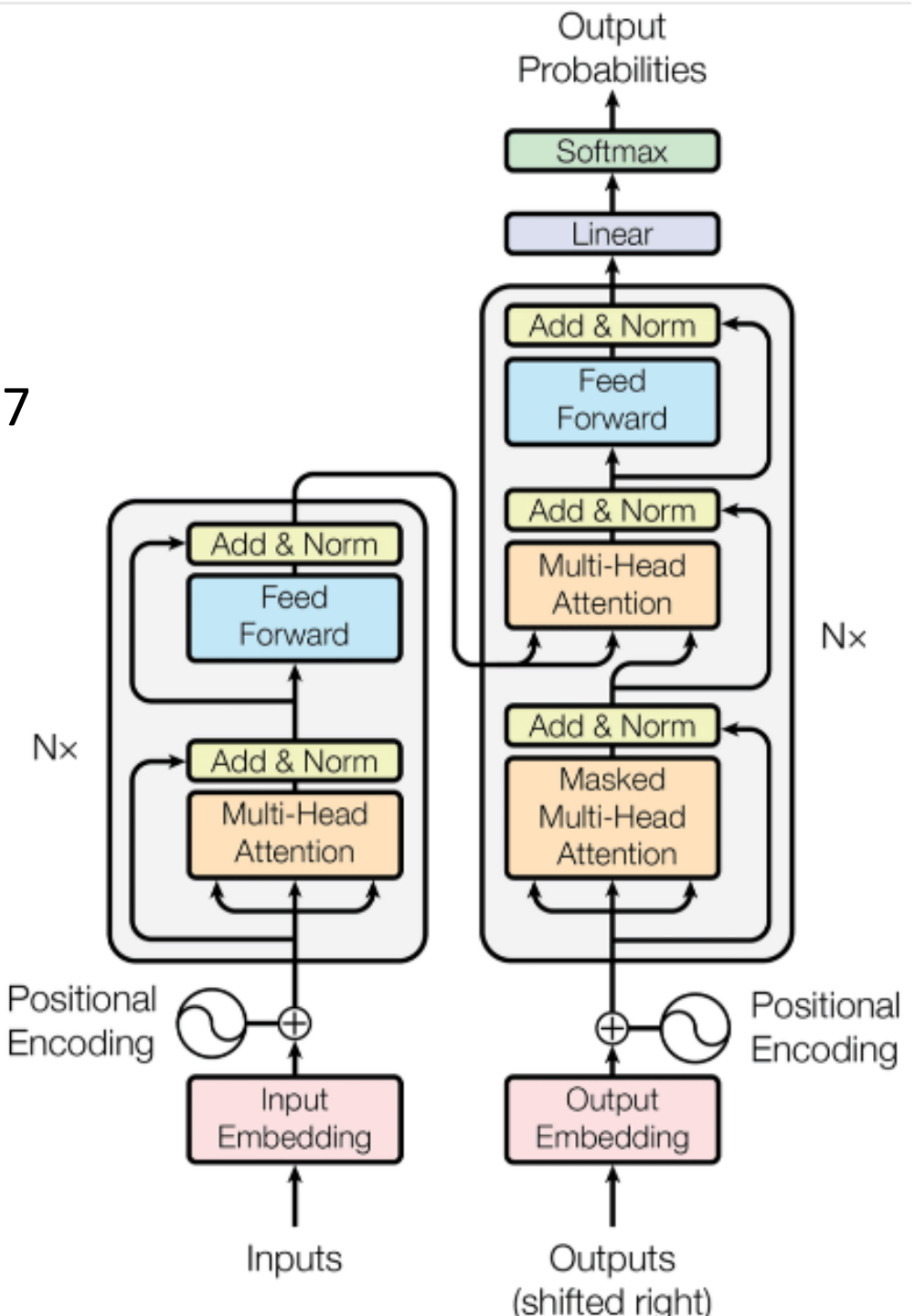
- Different types of LMs – Encoder-only, Encoder-Decoder, Decoder-only
- SLMs vs LLMs
- Training: Pretraining
- Training: Fine-tuning
- Inference
  - Prompting and Prompting guide (https://www.promptingguide.ai/)
    - System prompt and User prompts
    - Temperature
  - Retrieval Augmented Generation (RAG)
    - Embeddings and Vector Stores
- Agents
  - Motivation and General principles
  - Tools: LangChain, LangGraph, …

Language Models

Attention is all you need

A. Vaswani, N. Shazeer, N. Parmar, …

Neural Information Processing Systems, 2017

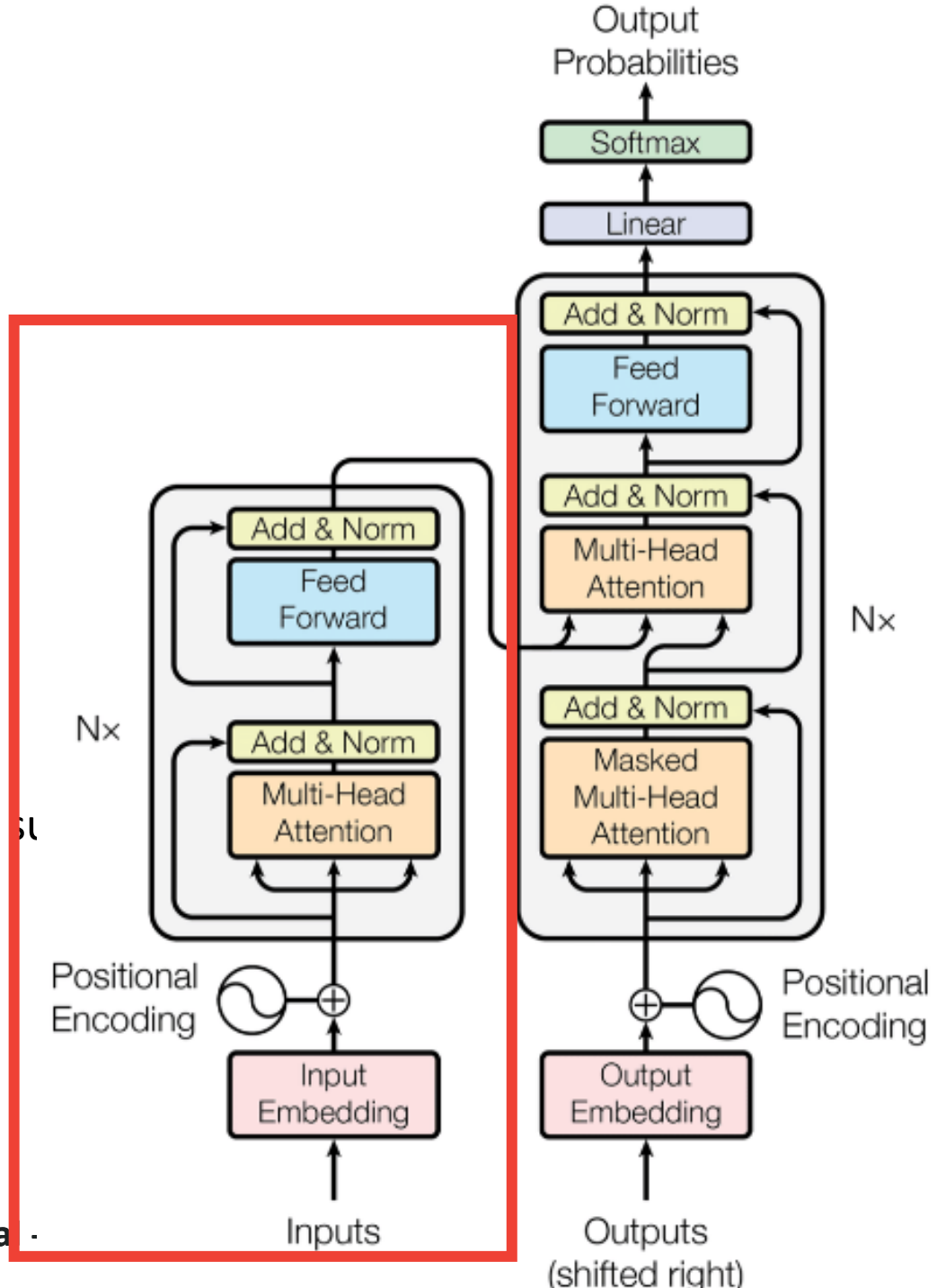


Transformer model

TCS Internal

Language Models

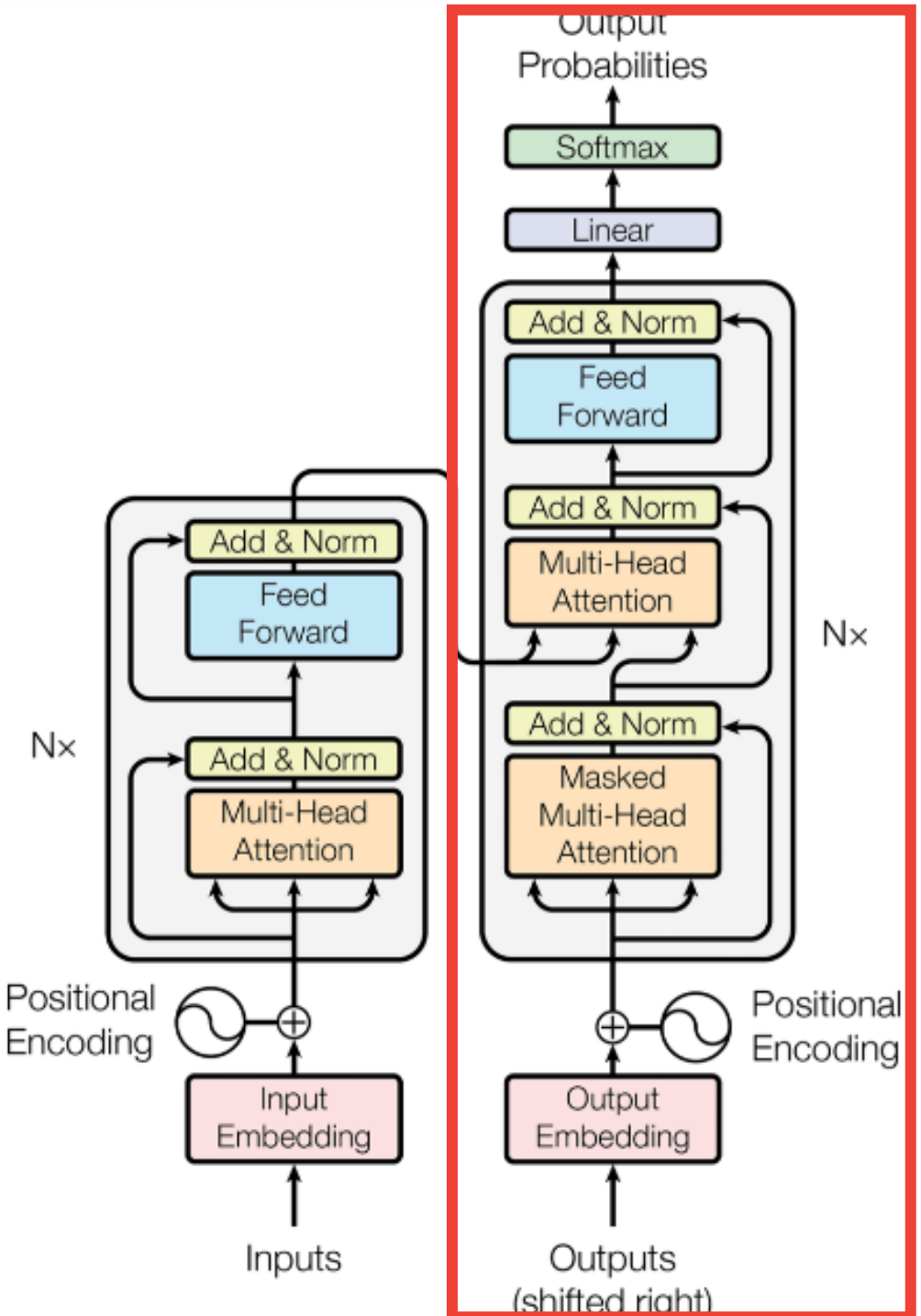
- **Encoder-only models**
  - Only have the encoder part of the transformer architecture
  - Bidirectional attention
  - BERT
  - RoBERTa
  - DeBERTa
  - …
  - Learning representations of the input for tasks su as
    - **Text Embeddings**
    - Text classification
    - Fill-in-the-blanks



TCS Internal

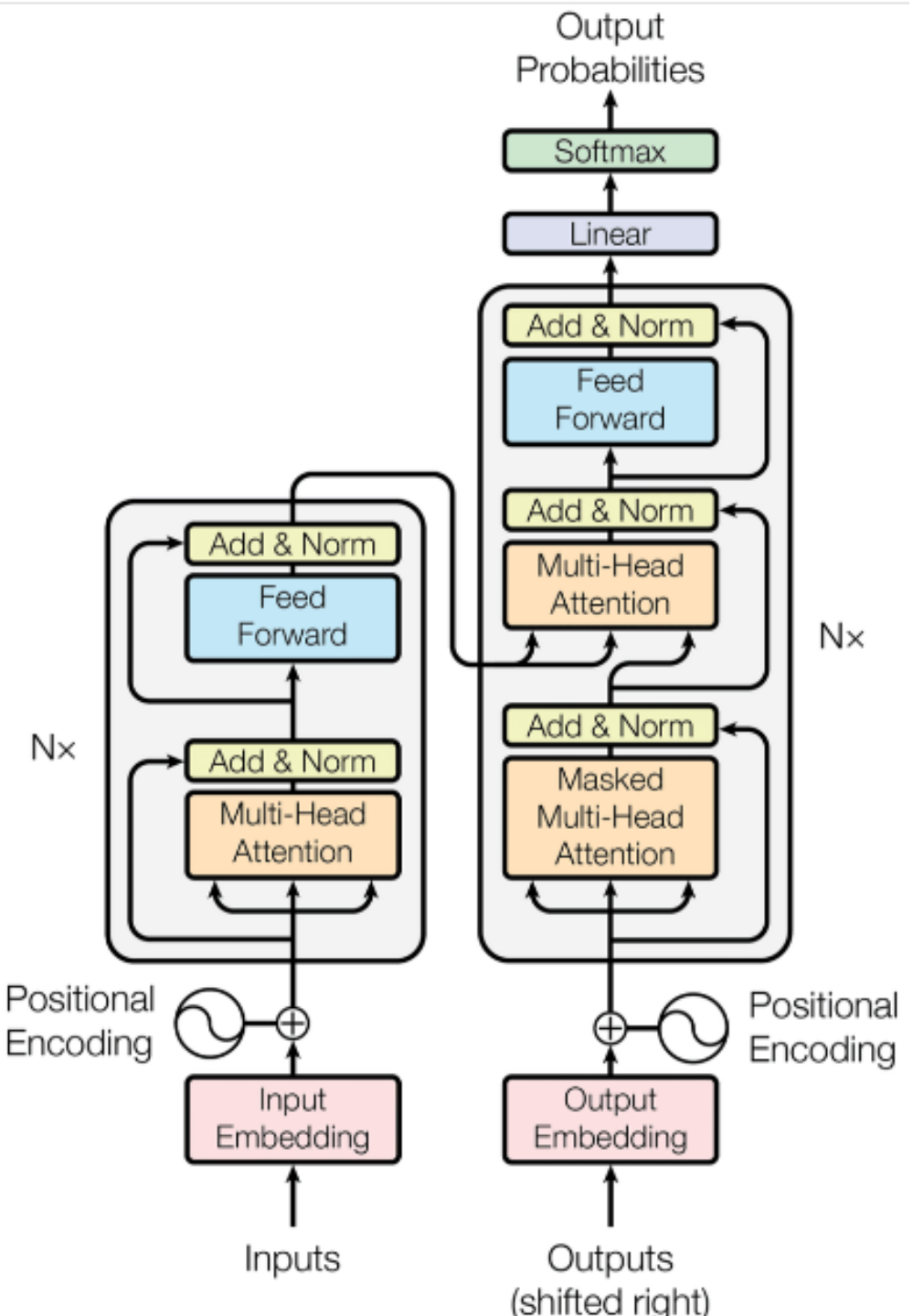
Language Models

- **Decoder-only models**
  - Only have the decoder part of the transformer architecture
  - Unidirectional as focused towards generation
  - GPT
  - Llama
  - Mistral
  - …
  - Generative tasks such as
    - Summarization
    - Question answering
    - Contextual generation

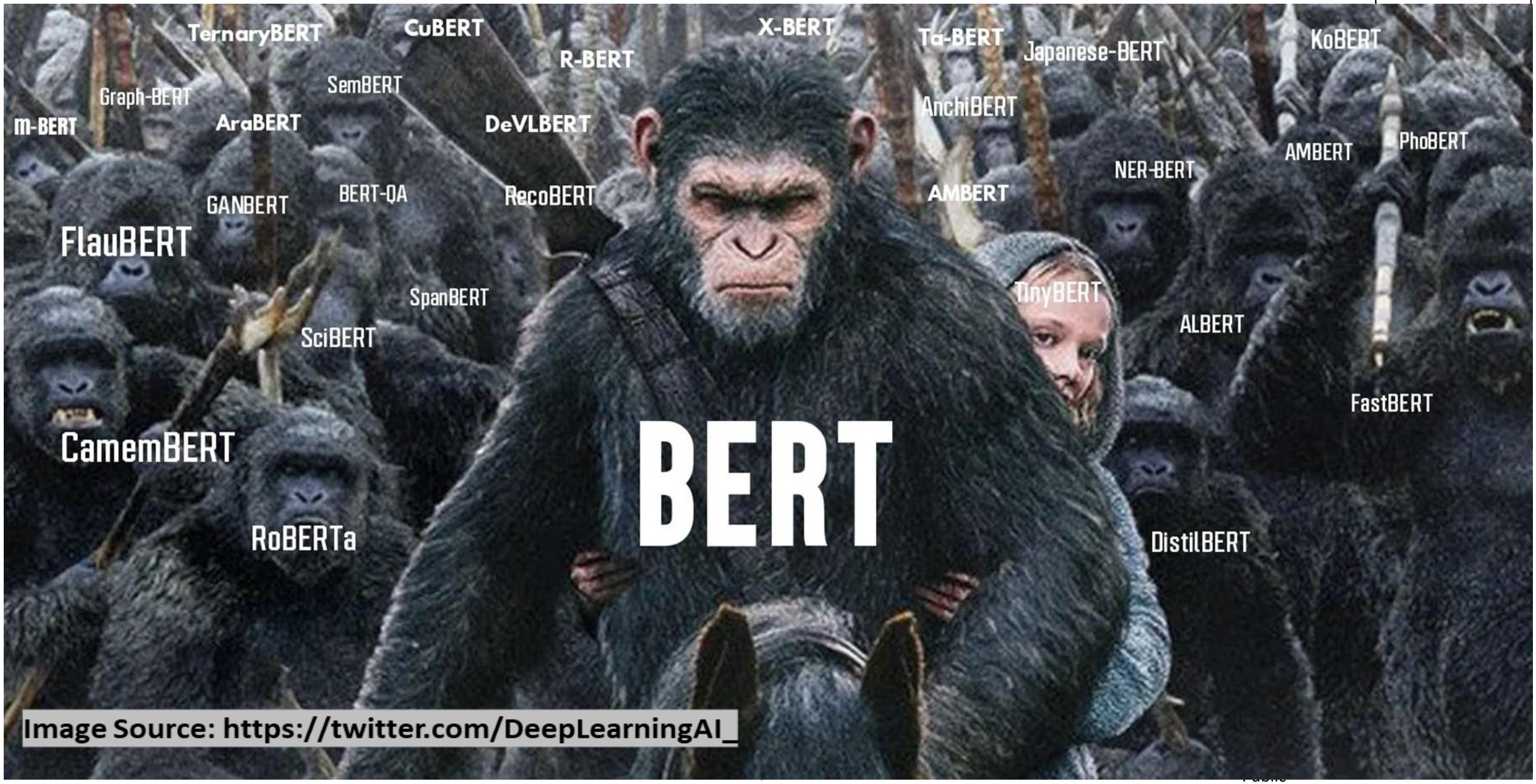


Language Models

- **Encoder-Decoder models**
  - Have both the encoder and decoder parts of the transformer architecture
  - T5
  - Flan-T5
  - BART
  - …
  - Text-to-Text transformer
    - Translation
    - Summarization
    - Sentence Acceptability
    - …



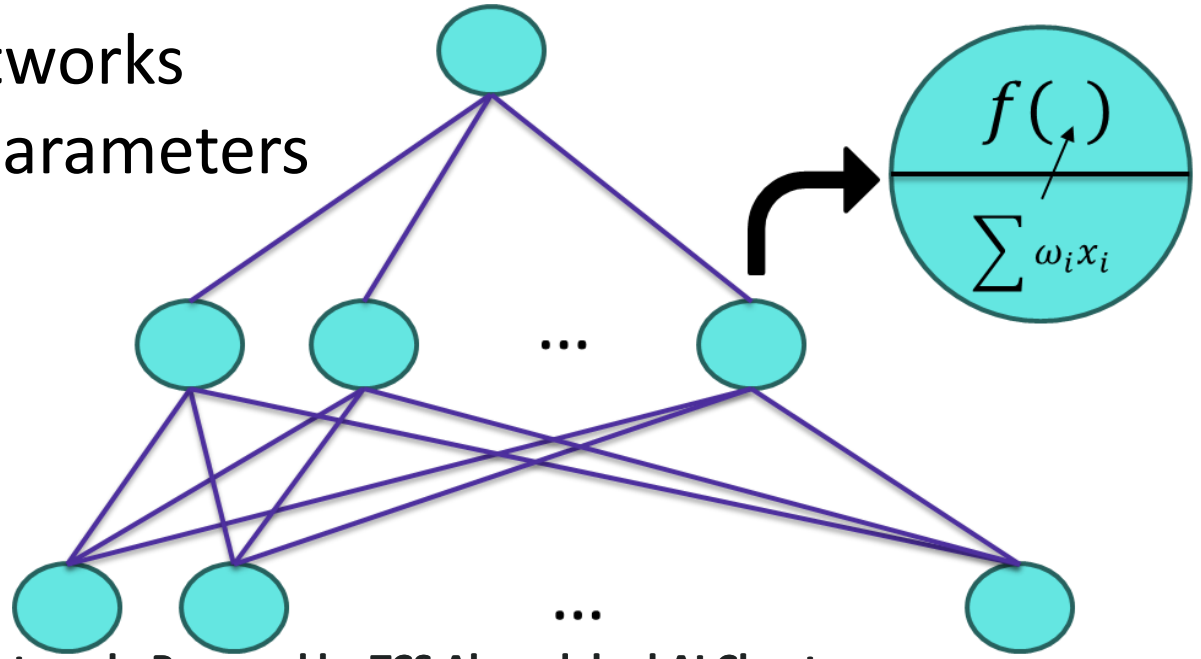
Language Models - Encoders



Language Models - Decoders

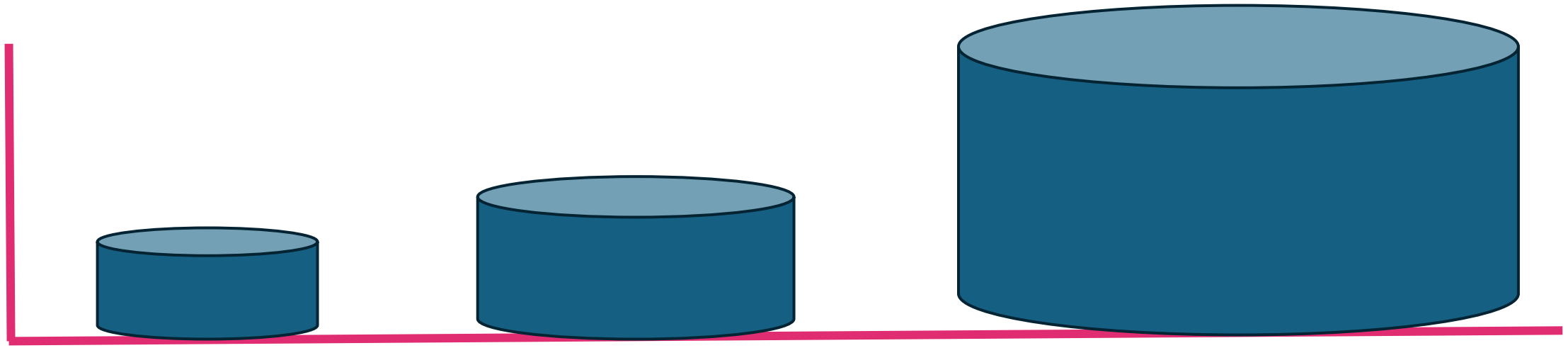
- Focus shifted to decoder-only models
  - All latest research is mostly focused on decoder-only models
  - All the famous models – GPT, Gemini, Deepseek, Llama are all decoder-only models

- Parameters
  - All these are deep neural networks
  - Weights and Biases are the parameters



Language Models – SLM vs LLM

Different parameter sizing bounds being put in literature[3,4]



Small LMs: Less than 10B parameters

- Apple's OpenELM
- GPT-2, GPT2-XL
- Mistral-7B, MPT-7B, Falcon-Instruct-7B
- Llama-8B-instruct

Medium LMs: 10B to 100B parameters

- GPT-NeoX-20B
- OPT-30B, Llama30B
- Falcon-40B

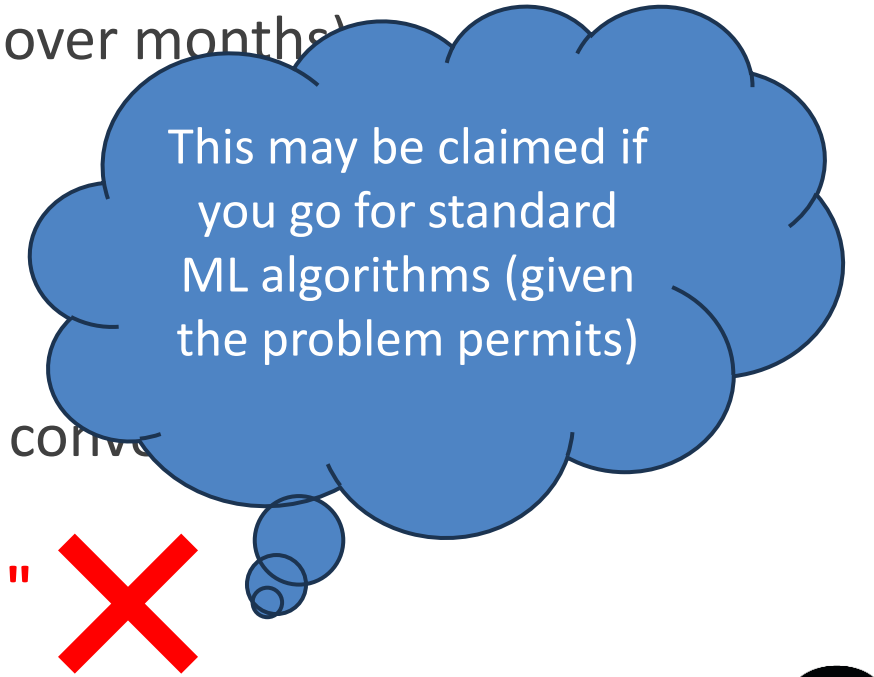
Large LMs: Greater than 100B parameters

- GPT-3.5, GPT-4o, GPT-4.1
- DeepSeek

3. https://catherinebreslin.medium.com/small-language-models...
4. https://www.koyeb.com/blog/what-are-large-language-models

Language Models - Training

- How are they created
  - Pre-training
    - Large amounts of text used to pre-train these models
    - Next token prediction and Reinforcement Learning
    - Privacy concerns
    - Practically impossible in our setting (1000s of GPUs over months)
  - Fine-tuning
    - Further train these models on your domain/task
    - Difficult with very large models
    - Painfully slow on CPUs and a GPU are necessary for conv



- **"We trained the model" … "We fine-tuned the model.. "**

Language Models - Inference

- Of most use to us is harnessing this pre-training
  - **Prompting –** Types of prompting
    - Zero-shot prompting
      - Only explaining the task without showing any examples explaining the task
      - Most teams did this
    - Few-shot prompting
      - Explaining the task with "k" examples explaining the task
      - Example: Generating advertisement snippets for products. Showing some example advertisement snippets in the prompt itself and then asking it to generate for the product under consideration.

Language Models - Inference

- **Prompting -** Elements of a prompt
- Instruction: a specific task or instruction you want the model to perform
  - Classify the following news as one of the following types – World, Business, Sports, or Science-Technology.

- Context: external information or additional context that can steer the model to better responses
  - Business news is about banking, stock markets, economy, investments, monetary policies, or international trade.

- Input Data: the input or question that we are interested to find a response for
  - Japan's economic growth slows down in last financial year.

- Output Indicator: the type or format of the output
  - "Class:" OR "News type:" at the end of the prompt

Language Models - Inference

- **Prompting –** System vs User prompt
  - System Prompt
    - A foundational instruction that defines the agent's role, behavior and limitations
    - "*You are a diligent financial analyst. Your primary role is to provide clear, concise, and accurate information based on financial data. Avoid making personal opinions or financial recommendations. Stick to objective analysis, use formal language, and provide data-backed summaries.*"
  - User prompt
    - A task-oriented question with specific details.
    - "*Summarize the key findings of the provided text about the recent interest rate hike. Include the impact on inflation and consumer spending in bullet points.*"

Language Models - Inference

- **Prompting**
  - More types such as Chain-of-Thought, Self-consistency, ReAct
  - System prompts vs User prompt - Examples
  - Elements of a prompt – Arrangement and requirements

  - Promptingguide.ai >> https://www.promptingguide.ai/

  - Hyperparameter - Temperature (0 to 1)
    - More towards 1 – Creative writing (poetry, suggestion generation)
    - More towards 0 – Factual answers (factual question/answering)

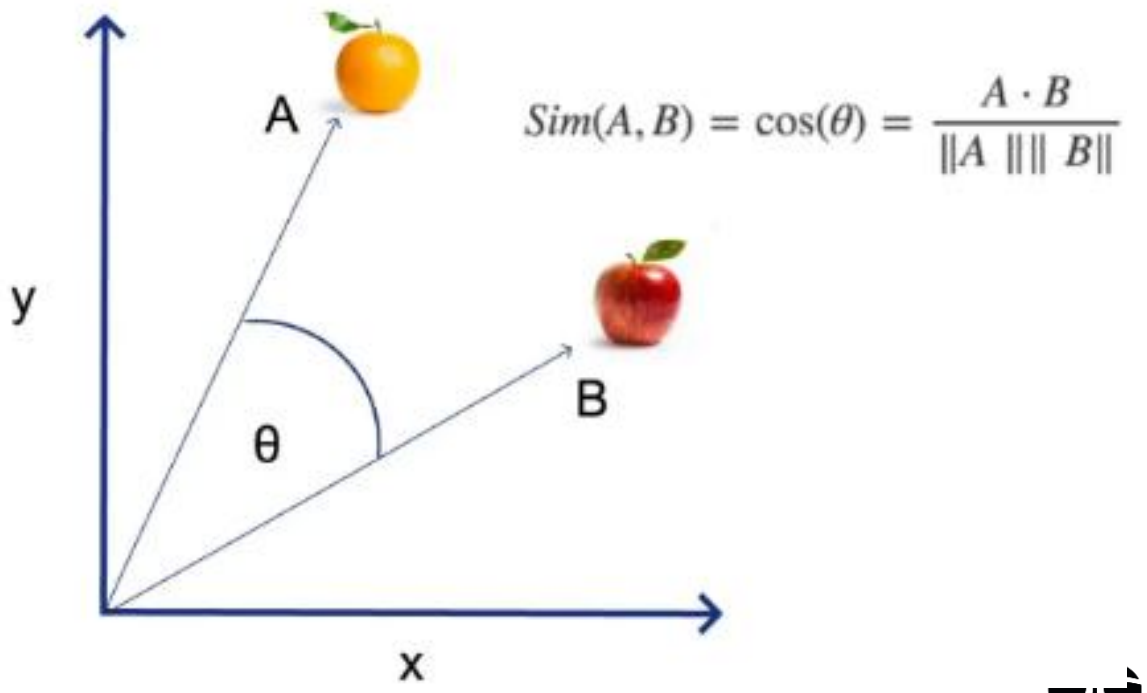
  - Hyperparameter - max sequence length
  - ...

Retrieval Augmented Generation (RAG) - Prerequisites

- Embeddings
  - Numerical representations of real-world data such as text, images and audio
  - "John lost his ticket to the cricket match" >> [0.1 -0.8 -1.7 … 1.6] (BERT – 768 dim)
  - Words, Sentences, Paras, (Chunks), Documents all can be embedded

- Cosine Similarity
  - Similarity based on the angle between the embedding vectors

- Vector Stores/Databases
  - Store/DB of embeddings
  - Native support for similarity functions
  - Examples: FAISS, ChromaDB, Milvus



https://botpenguin.com/glossary/cosine-similarity

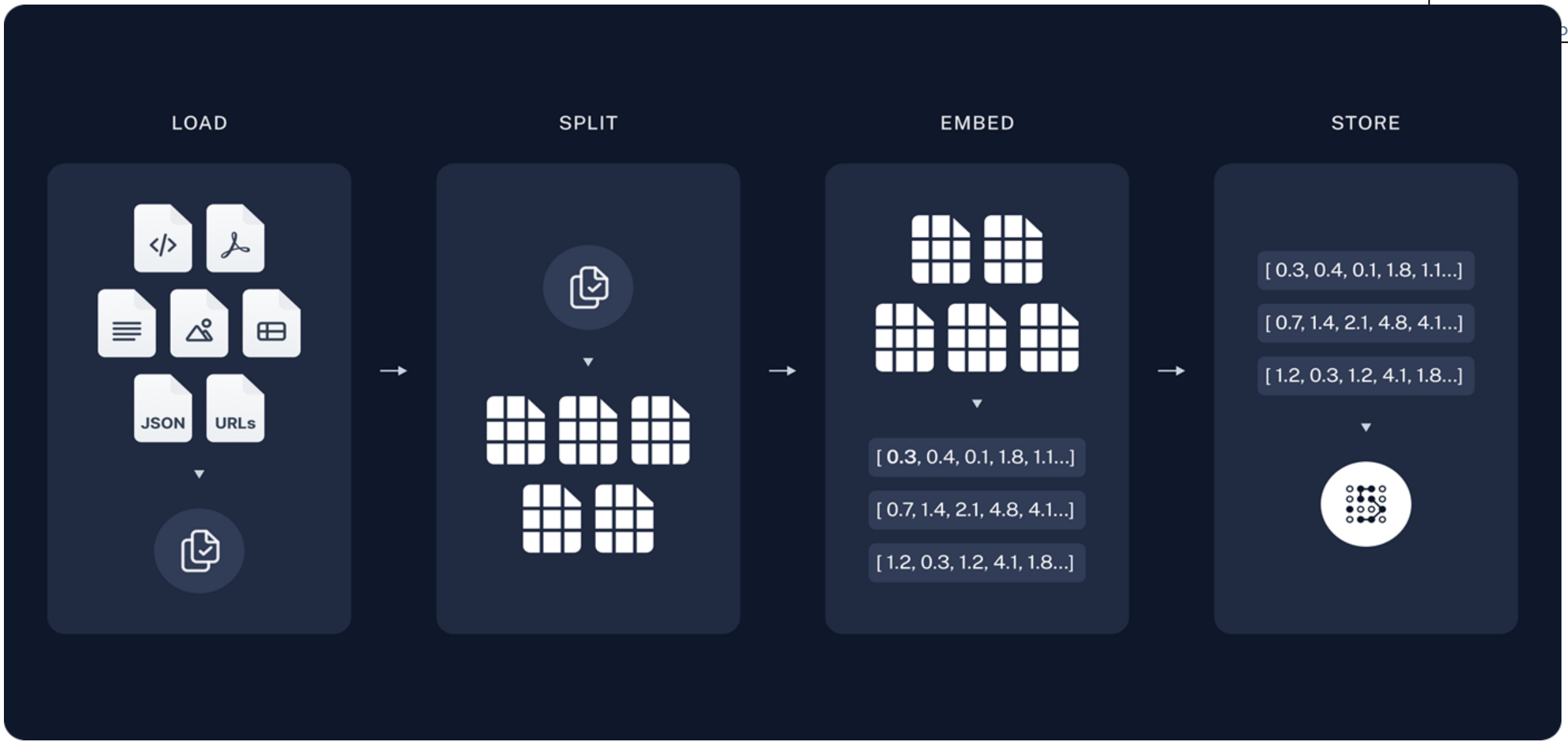
Retrieval Augmented Generation (RAG)

- **Goal**: To answer any questions using information from specific domain documents

- Why do we need RAG?
  - Any LLM's knowledge is limited to the public data up to a specific point in time that they were pre-trained on.
  - LLM's knowledge may not cover certain information present in private documents

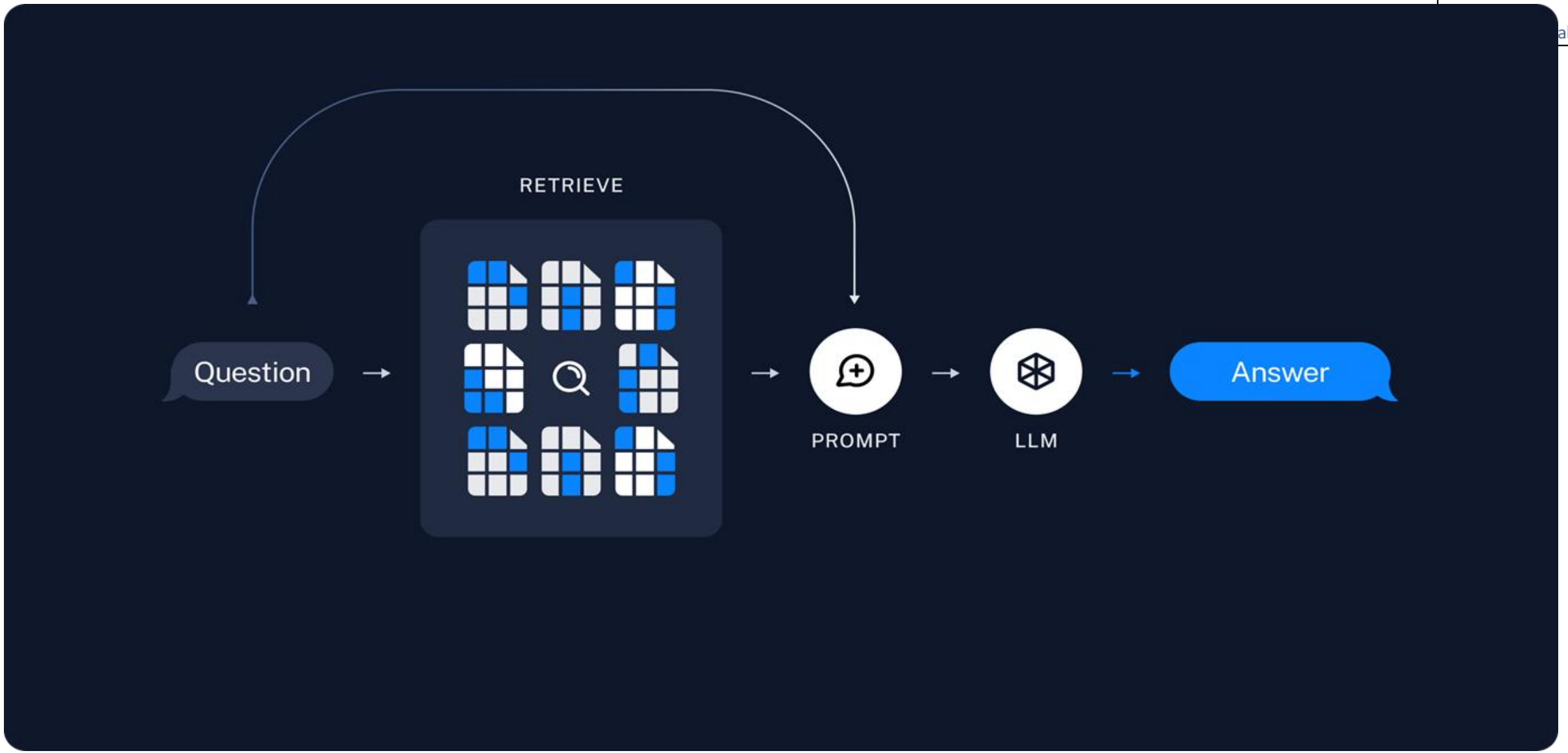
- The process of bringing the appropriate information and inserting it into the model prompt is known as **Retrieval Augmented Generation (RAG)**.
  - Indexing
  - Retrieval
  - Generation

Retrieval Augmented Generation (RAG) - Indexing Phase



https://python.langchain.com/docs/tutorials/rag/

Retrieval Augmented Generation (RAG) - Retrieval & Answering phase



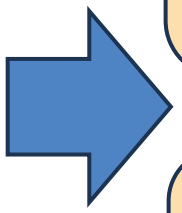
https://python.langchain.com/docs/tutorials/rag/

Retrieval Augmented Generation (RAG) - Pitfalls observed

- Many teams simply sent anything under the sun for embedding and storage to vector stores 
- Chunking is a not a straightforward process

We observe patients arrive with high fever, body ache, loss of appetite and headache. We suspect several possible issues such as common cold, COVID and infections. We prescribe an initial 2-day round of the following medicines:

1. Paracetamol (500 mg) 3 each day
2. Azithromycin (250 mg) 2 each day
3. Metrogyl (200 mg) 3 each day
4. Rantadine (150 mg) before food, 3 each day



We observe patients arrive with high fever, body ache, loss of appetite and headache. We suspect several possible issues such as common cold, COVID and infections. We prescribe an initial 2-day round of the following medicines:

1. Paracetamol (500 mg) 3 each day

1. Azithromycin (250 mg) 2 each day
2. Metrogyl (200 mg) 3 each day
3. Rantadine (150 mg) before food, 3 each day

……

LLM Agents - Motivation

- Consider an LLM application that is designed to help financial analysts to answer questions about any company's performance
- **Question**: What was XYZ corporation's total profit for FY 2023 from their hotel business?
  - A well-designed Retrieval Augmented Generation (RAG) pipeline can answer this
- **Question**: What were the three takeaways from the Q2 earnings call from FY 2023, focusing on recent acquisitions of smaller hotel chains?
  - Requires more than simple lookup
  - Requires planning, breaking down complex problem into smaller sub-problems, memory, using different tools => **LLM Agent**

LLM Agents - Components

- Planning:
  - Subgoal and decomposition: Breaking down large tasks into smaller, manageable subgoals, enabling efficient handling of complex tasks.
  - Reflection and refinement: Self-criticism and self-reflection over past actions, learn from mistakes and refine them for future steps
- Memory:
  - Short-term memory: In-context learning
  - Long-term memory: Capability to retain and recall (infinite) information over extended periods, often by leveraging an external vector store and fast retrieval
- Tool Use:

  The agent learns to call external APIs for extra information that is missing from the model weights (often hard to change after pre-training), including current information, code execution capability, access to proprietary information sources and more

LLM Agents - Resources

- Technical Blog: https://lilianweng.github.io/posts/2023-06-23-agent/

- General Conceptual/Business perspective: https://www.insightpartners.com/ideas/state-of-the-ai-agent-ecosystem-use-cases-and-learnings-for-technology-builders-and-buyers/

- Several Toolkits available:
  - LangChain
  - LangGraph
  - CrewAI
  - Microsoft AutoGen
  - Big list at https://github.com/kaushikb11/awesome-llm-agents

Anatomy of a good solution – The Pitch -

Connect Solutions to Real-World Problems: Encourage participants to start by **identifying specific, real-life challenges** that their AI solutions can address. This ensures **relevance and practicality.**	**Understand the User Context**: Stress the importance of considering **who will use the solution**, **how**, and in **what environment**. Contextualizing use cases can help create more stable and useful products.
Map Commercial Value to Real-Life Impact: Guide them to link commercial aspects with tangible benefits for users. Ask: How does this solution improve daily life, save time, or reduce costs?	**Relate to Existing Processes**: Suggest they **compare their AI solution to existing workflow**s or methods, highlighting improvements and differences.
Use Case Storytelling: Recommend using **storytelling techniques** to describe how their solution would work in a real-life scenario. This makes explanations clearer and easier to understand.	**Showcase Measurable Outcomes**: Encourage the use of metrics or examples to clearly demonstrate the impact and value of their solution.
Be Specific in Explanations: Advise them to **avoid generic** terms and instead use **precise language** that **accurately conveys** what their AI/GenAI solution does.	**Engage with Stakeholders**: Remind them to think from the perspective of all stakeholders involved, not just end-users—consider business owners, technical staff, and others.
Demonstrate Solution Stability: Ask participants to discuss how their solution **would perform in real situations, including potential limitations** and **how they plan to address them**.	**Practice Clear Communication**: Urge them to **rehearse explaining their solution** to both technical and non-technical audiences**, tailoring their language and detail accordingly**.

Anatomy of a good solution

- Planning
  - Work assignment to team members (all workers + optionally a coordinator)
  - Phase wise (3 hr milestone, 6 hr milestone, etc.)
- Data
  - Generate data (Using LLMs) - requires some idea of how the data is – take help from mentors or LLMs
  - Download from Kaggle or other data sources – requires checking if the downloaded thing fits the definition required data
  - If any of the team members are from the domain of the definition and has experience in such data, one can see how to harness that knowledge.

Anatomy of a good solution

- Solution
  - Large LMs (key based access)
  - Small LMs (ollama) ? -- brownie points for saving on cost and ensuring data privacy
    - Study the use of gguf format llm files for faster inference on CPU
- Evaluation
  - Multiple LLMs for verification – Reporting actual accuracy numbers on a small test dataset can make your claimed solution's impact stronger and appeal to the jury
- Presentation
  - HTML based quick view with click to the demo

Psychological Aspects of Gen AI Solution

- **Trustworthy and Reliable**
  - Accuracy
  - Transparency
  - Consistency
- **Autonomous with Human Oversight**
  - Human-in-the-loop
  - User Control
- **Competence Building and Skill Enhancement**
  - Complementary Design
  - Feedback
- **Relatable without Emotional Dependence**
  - Human-centered Design
  - Emotional Nuance
- **Ethical Considerations**
  - Bias Mitigation
  - Privacy and Security

Problem Definitions for evaluation on 25th (Session 2)

- Problem 1: (Natural Language oriented)
  - Public Services Query Resolver assistant
    - Input: Driving License, Home registration, electricity, gas etc. Procedural pdfs and manuals, FAQs, …
    - Output: A chatbot that assists regular public on several kinds of questions such as documents required, appointments, etc.

- Problem 2: (Data oriented)
  - IoT AI Agent for Smart Facility Utility Usage Optimization
    - Input: Data from IoT devices such as light sensors, timings, water flows, etc.
    - Output: Dashboard/Report generation system which considers the input parameters and brings out usage charts and suggestions on how to optimize (should certain lights be turned out after 6 PM,etc.)

Mentimeter

- What has been your / your solution's unique selling point (USP)?

Mentimeter

- What has been your / your solution's biggest improvement area?

Jury's Perspectives from earlier rounds

- Data
  - Data generation vs Data Download
  - Limited data points (Common issue)

- Problem Definition
  - Swaying away from what was asked (Agent vs no agent)
  - Not all aspects being tackled

- Solution perspective
  - Planning and distribution of work
  - Use of standard ML is not discouraged
  - Creativity is awarded
  - Evaluation

.....Going the extra mile!

Jury's Perspectives from earlier rounds

- Appreciable inclusions to respective solutions
  - Connectivity with existing workflows
  - Using standard ML where required (Clustering, Classification)
  - Knowledge and Use of advanced Generative AI concepts (Self-consistency, React)
  - "Useful" Chatbot - when not asked
  - Voice based commands (Marks for Accessibility and Inclusion)
  - Processing image-based inputs when only text was specified
  - Easy to understand and aesthetic UI
  - …

Thank you and
Best Wishes!