**DAY 1**

**What is tokenization   
what is tiktoken  
what is transformer**

**What is vector embeddings**

**What is positional Encodings**

**What is Self Attention & multi-head Attention**

**What is RAG  
What is Synthetic data  
What AI Jargons**

**1. Tokenization**

Tokenization is the process of converting a sequence of text (like a sentence or a paragraph) into smaller units, called tokens. These tokens are typically words, subwords, or even characters. In NLP (Natural Language Processing), tokenization is a crucial step because machine learning models like transformers require input to be in tokenized form.

* **Word-level tokenization**: Splitting the text into individual words.
* **Subword-level tokenization**: Breaking down words into smaller units (subwords). This is especially useful for rare words or misspellings.
* **Character-level tokenization**: Breaking down text into individual characters.

For example:

* Text: "ChatGPT is amazing"
* Tokens: ["ChatGPT", "is", "amazing"]

Tokenization allows a model to better process text, understanding the structure and relationships of words.

**2. Tiktoken**

Tiktoken is a Python library designed to efficiently handle tokenization specifically for OpenAI's language models like GPT. It's optimized for encoding and decoding text into tokens in a way that is computationally efficient and memory-friendly.

* **Purpose**: It's a specialized tool used to manage the tokenization process for large-scale language models like GPT-3, GPT-4, etc.
* **Efficiency**: It reduces overhead by providing fast encoding and decoding of text inputs and outputs, making it highly suitable for applications that require real-time or large-scale processing.

**3. Transformer**

A transformer is a type of neural network architecture introduced in the paper "Attention is All You Need" by Vaswani et al. (2017). Transformers have revolutionized the field of NLP and are the foundation of many state-of-the-art models like BERT, GPT, T5, and others.

* **Key Components**:
  + **Self-Attention Mechanism**: Allows the model to weigh the importance of different words in a sequence relative to each other, helping it understand context.
  + **Positional Encodings**: Since transformers don't process input sequentially like RNNs or LSTMs, positional encodings are used to provide information about the position of words in a sequence.

Transformers have significantly improved performance on NLP tasks because they can capture long-range dependencies in text better than traditional models.

**4. Vector Embeddings**

Vector embeddings are dense representations of text, words, or sentences in continuous vector space. These vectors capture the semantic meaning of the text they represent, allowing models to understand relationships between words and phrases based on their proximity in vector space.

* **Example**: In a word embedding space, the words "king" and "queen" would be represented as vectors that are closer together than "king" and "car".
* **Use**: Embeddings are used in various NLP tasks like machine translation, text classification, sentiment analysis, etc.

Popular embedding techniques include:

* **Word2Vec**
* **GloVe**
* **BERT Embeddings**

**5. Positional Encodings**

Since transformers don’t have an inherent sense of order (unlike RNNs and LSTMs), positional encodings are added to give the model information about the position of words in a sequence.

* **Why needed**: Unlike RNNs, which process text sequentially and inherently have access to word order, transformers treat all words in a sequence at the same time. To account for the order, positional encodings are added to each token's embedding to tell the model its position in the sequence.

Positional encodings are typically vectors added to token embeddings, and they can be learned or fixed. Sinusoidal positional encodings are often used, which use a combination of sine and cosine functions.

**6. Self-Attention & Multi-Head Attention**

**Self-Attention** is a mechanism in which each word (or token) in a sentence is compared to every other word in the sentence to determine their relationship or relevance to each other.

* **Self-Attention**: Helps a model to focus on relevant parts of the input sequence when making predictions, regardless of their distance in the sequence. For instance, in the sentence "The cat sat on the mat", the word "cat" should pay more attention to the word "sat" than to the word "mat".

**Multi-Head Attention** extends self-attention by running multiple self-attention processes in parallel. Each "head" learns a different set of relationships, allowing the model to capture multiple types of relationships and patterns in the data.

* **Benefit**: It helps capture different aspects of the sentence simultaneously, improving the model’s ability to learn diverse information from the context.

**7. RAG (Retrieval-Augmented Generation)**

RAG is a hybrid model architecture that combines the power of retrieval-based methods (retrieving relevant documents from a database) with generation-based models (like transformers).

* **How it works**: RAG first retrieves relevant documents or passages from a large corpus based on the query. Then, it uses a generative model (like GPT) to synthesize an answer using the retrieved information. This allows RAG models to provide more accurate and contextually relevant responses, especially in knowledge-intensive tasks.
* **Use cases**: It’s useful in tasks where the model needs to answer questions based on external knowledge, like question answering or summarization, without having to store all knowledge in the model’s parameters.

**8. Synthetic Data**

Synthetic data refers to artificially generated data that is used to train machine learning models. Instead of using real-world data, synthetic data is created by algorithms to mimic real-world data.

* **Why use synthetic data**: It helps overcome data scarcity or privacy concerns, particularly in areas where obtaining real-world data is expensive, sensitive, or time-consuming (e.g., medical or financial datasets).
* **Applications**: It's used in training models for image recognition, NLP, autonomous vehicles, and more.

Synthetic data is generated using techniques like:

* **Simulation-based generation**
* **Generative models like GANs (Generative Adversarial Networks)**

**9. AI Jargon**

AI jargon refers to the specialized terminology used in the field of artificial intelligence and machine learning. Some common examples include:

* **Model**: The algorithm or system used to process input and make predictions or decisions.
* **Training**: The process of feeding data to a model to help it learn patterns.
* **Inference**: The process of using a trained model to make predictions on new, unseen data.
* **Loss Function**: A mathematical function used to evaluate the performance of a model.
* **Overfitting**: When a model learns too much from the training data and performs poorly on new data.
* **Underfitting**: When a model fails to capture the underlying patterns of the training data and performs poorly.

These terms are fundamental to understanding how machine learning models are built, trained, and deployed.

**DAY 2**

**What is Prompts?**

**What is Alpaca Prompts?**

**What is LLAMA Prompting?**

**What is System Prompt?**

**What is Zero and Few Shot Prompting?**

**What is Self-Consistency Prompting?**

**What is Persona Based Prompting?**

**What is Role-Play Prompting?**

**What is Conceptual Prompting?**

**What is Multimodal Prompting?**

**🧠 1. What is a Prompt?**

A **prompt** is the input or instruction you give to a language model (like GPT, Gemini, etc.) to get a desired output.

* It can be a **question**, a **command**, or a **structured format**.
* Think of it as a conversation starter or a set of instructions.

**Example:**  
Prompt → *"Write a poem about the ocean."*  
Output → A poem about the ocean 🌊

**🐑 2. What is Alpaca Prompting?**

**Alpaca Prompts** come from **Stanford’s Alpaca**, a fine-tuned version of LLaMA using **instruction-following** data.

* Alpaca prompts are **instruction-tuned** prompts like:  
  *“Explain how photosynthesis works.”*  
  *“Write a Python script to calculate factorial.”*

**Goal:**  
Make the model follow specific, **human-style instructions** better (like ChatGPT).

**🦙 3. What is LLaMA Prompting?**

**LLaMA (Large Language Model Meta AI)** is Meta's open-source LLM.

**LLaMA Prompting** refers to the **style of prompting** used for LLaMA models, often requiring:

* Precise input formatting
* Special tokens like <s>, </s> in older versions
* Instructions tailored for **factual, step-by-step** responses

LLaMA + fine-tuning = base for models like Alpaca, Vicuna, etc.

**⚙️ 4. What is a System Prompt?**

A **system prompt** is a **hidden or initial prompt** that sets the behavior of the AI for the session.

* It's often used **under the hood** (like in ChatGPT).
* It defines how the model should behave — tone, style, goals.

**Example:**

*"You are a helpful assistant that explains things in simple terms."*

All your responses will then follow this guideline.

**🎯 5. What is Zero-shot and Few-shot Prompting?**

**🟢 Zero-shot prompting:**

* **No examples** are given.
* Just the instruction.

**Example:**

*"Translate 'Hello' to French."*  
Model → "Bonjour"

**🟡 Few-shot prompting:**

* A few **examples** are provided to help the model understand the task.

**Example:**

diff

CopyEdit

Translate English to French:

- Hello → Bonjour

- Good morning → Bonjour

- Thank you → Merci

Now translate: How are you?

Model → Comment ça va?

**🔁 6. What is Self-Consistency Prompting?**

Instead of just **one output**, the model generates **multiple responses** and selects the **most consistent** answer.

* Increases **accuracy** and **reliability**.
* Often used in **reasoning tasks** or **math problems**.

**Example:**  
Ask the model to solve a math problem 5 times → Choose the most common correct answer.

**👤 7. What is Persona-based Prompting?**

You tell the model to **take on a specific persona** or identity.

**Examples:**

* "Act like Elon Musk and explain Mars colonization."
* "You are a fitness trainer. Give me a home workout."

It influences **tone, language, and personality** in the response.

**🎭 8. What is Role-Play Prompting?**

Similar to persona prompting, but more **interactive** and **scenario-driven**.

**Example:**

You are a doctor. I’m a patient with a headache. Let’s have a consultation.

Model replies like an actual doctor. Used in:

* Chatbots
* Simulations
* Education

**🧠💡 9. What is Conceptual Prompting?**

This is about **abstract thinking** and helping models reason through **ideas, logic, and mental models**.

**Example:**

"Explain democracy as if I’m a 5-year-old."

The model simplifies complex **concepts** into digestible ideas. Great for:

* Teaching
* Analogies
* Simplifying knowledge

**🖼️🔤 10. What is Multimodal Prompting?**

Using **multiple modes** of input (e.g., text, image, audio) to ask a question or get an answer.

**Example:**

[Image of a plant] + “What kind of plant is this?”

Used in:

* Gemini
* GPT-4 with vision
* CLIP models

**Real-world Uses:** AI tutors, visual question answering, image captioning.

**✨ Summary Table:**

| **Prompt Type** | **Description** |
| --- | --- |
| **Prompt** | Instruction to guide LLMs |
| **Alpaca Prompting** | Instruction-following for LLaMA-style models |
| **LLaMA Prompting** | Prompt format and style used for Meta’s LLaMA |
| **System Prompt** | Initial instruction to set AI behavior |
| **Zero-shot Prompting** | No examples, just the task |
| **Few-shot Prompting** | Give a few examples for better context |
| **Self-Consistency** | Generate multiple outputs and pick the best |
| **Persona-based Prompting** | Model acts like a specific character |
| **Role-play Prompting** | Interactive scenario-driven conversations |
| **Conceptual Prompting** | Explaining abstract or deep concepts clearly |
| **Multimodal Prompting** | Use of text + image/audio/video as input |