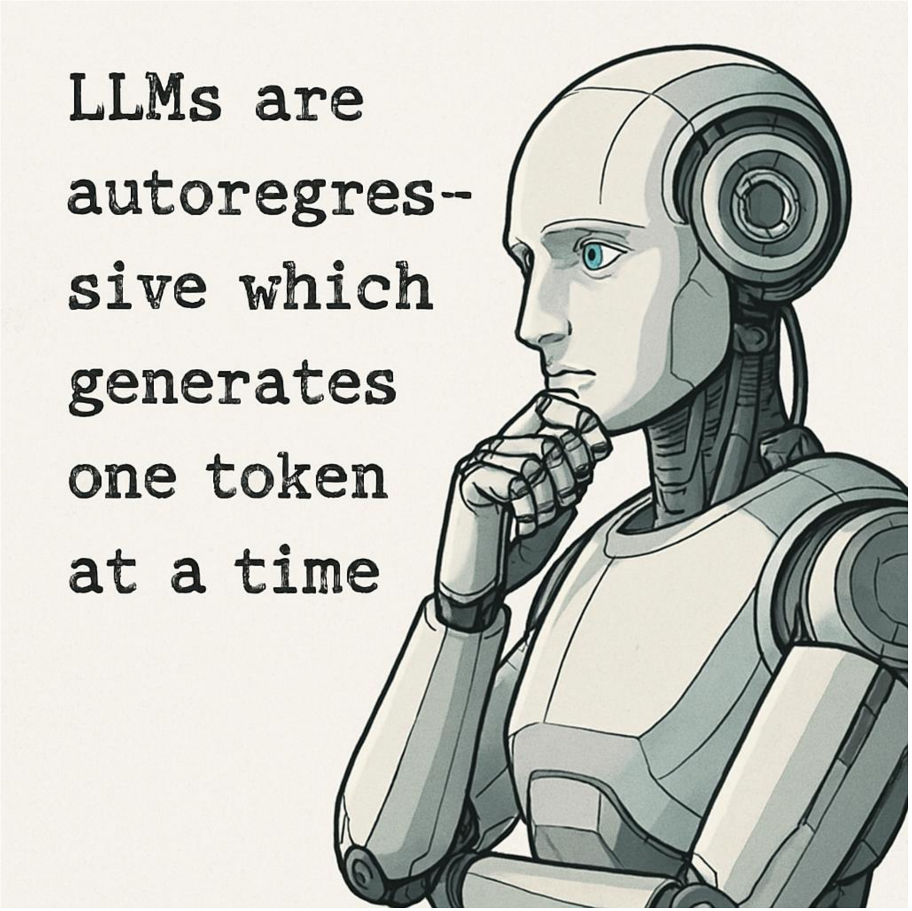
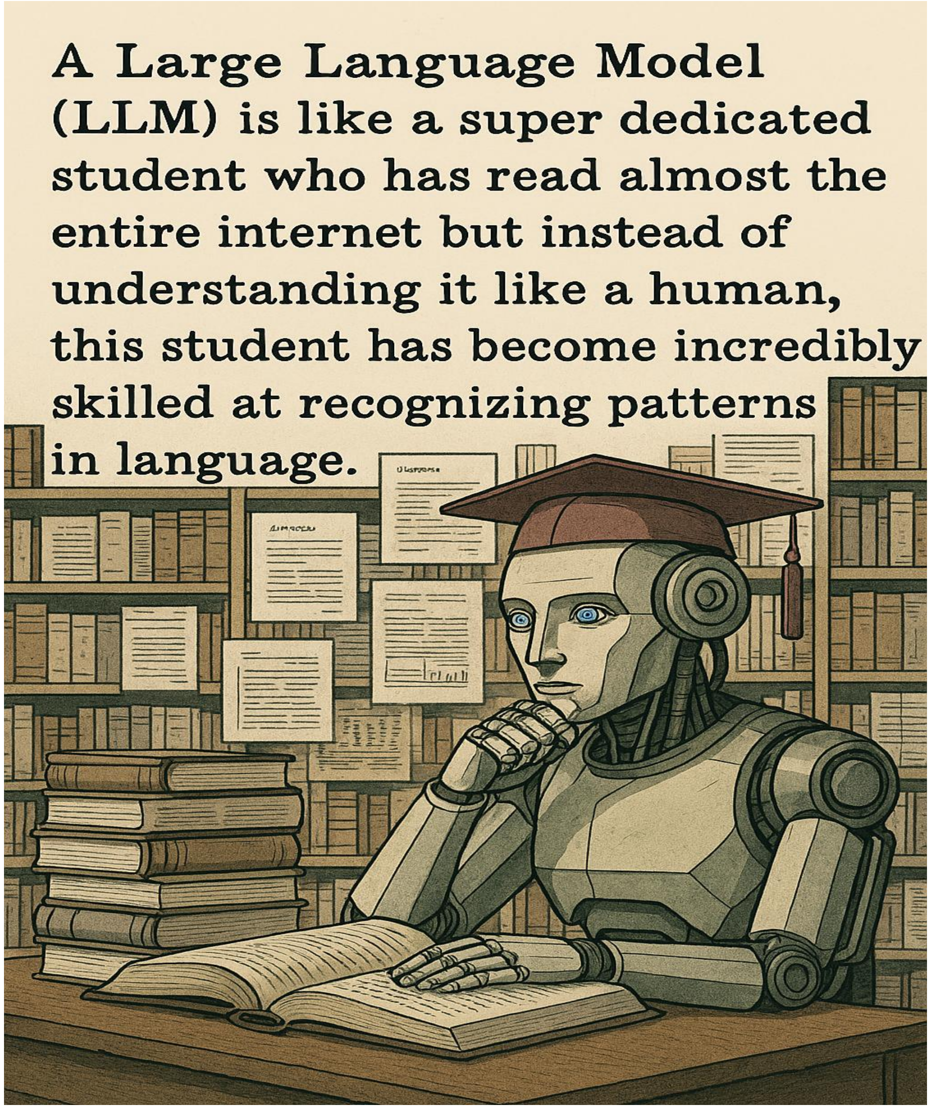
FUNDAMENTALS OF LLM



Presented By: Shreya Lal

LARGE LANGUAGE MODEL?



A Large Language Model (LLM) is like a super dedicated student who has read almost the entire internet but instead of understanding it like a human, this student has become incredibly skilled at recognizing patterns in language.

- **Definition**: LLM is a very sophisticated computer program that:
  - ➢ Is **trained** on a massive amount of text data.
  - ➢ Learns the statistical relationships between words, sentences, and concepts.
  - ➢ Its primary function is to **predict the next most likely word** in a sequence.

THE "LARGE" IN LLM ?

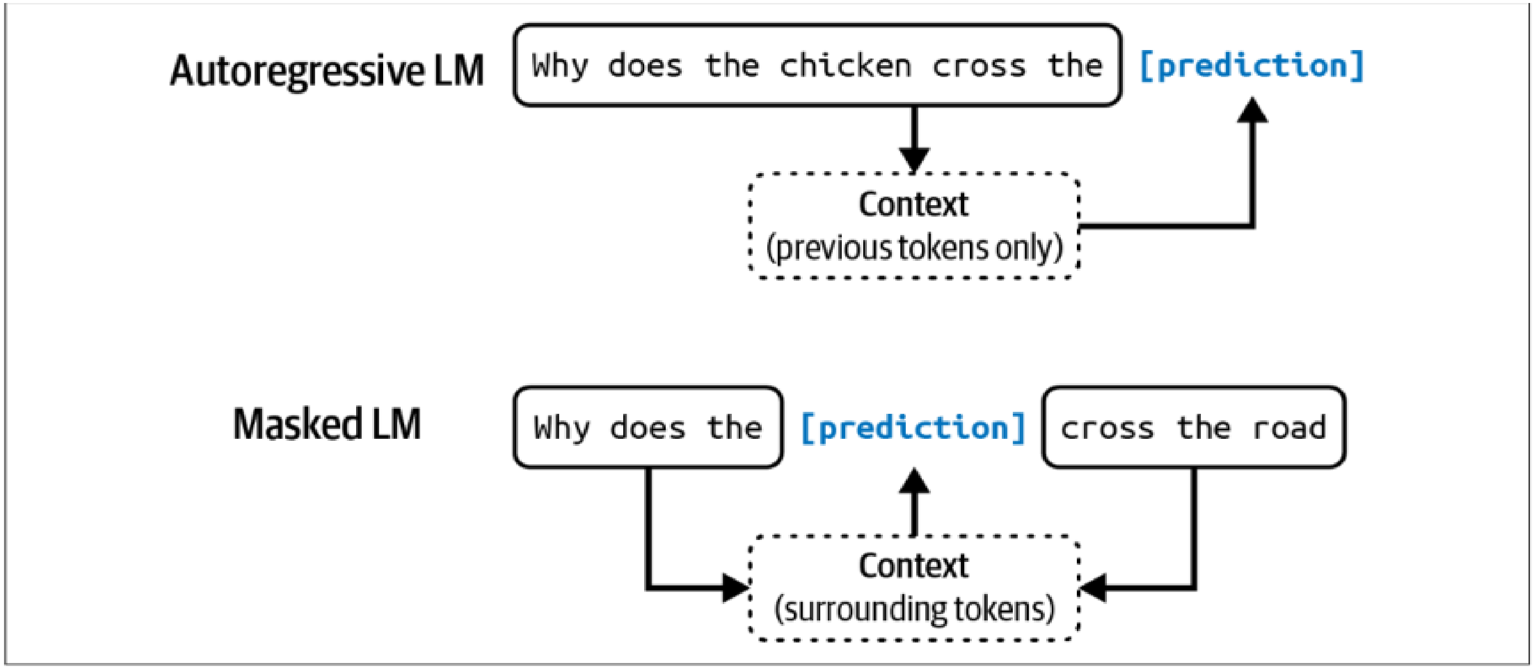
- What makes LLM’s so Powerful?

Three key ingredients make an LLM "large":

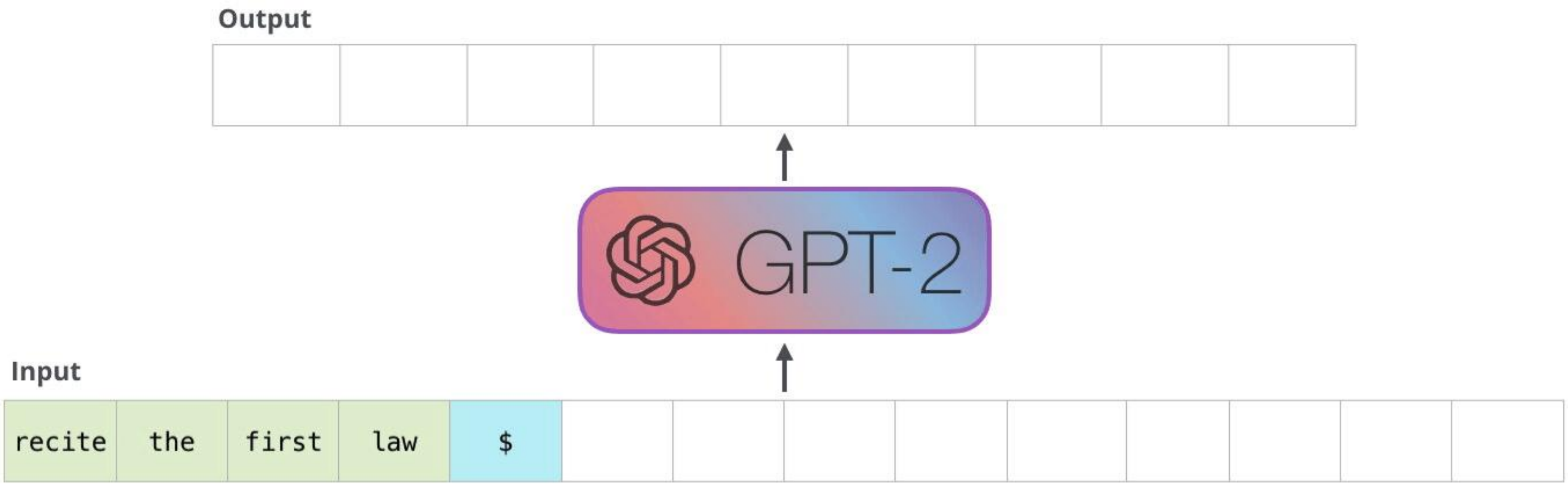
- **Massive Dataset:** The training data (the "textbooks" for our student) is enormous, often encompassing **terabytes** of text from diverse sources.
- **Immense Number of Parameters:** This is the most technical part, but think of **parameters** as the model's "knowledge nodes" or "synapses." They are the parts of the model that store the patterns it learns.
  - Early models - millions of parameters.
  - Modern LLMs like GPT-4 - *hundreds of billions*.

  More parameters allow for more nuanced and complex knowledge.
- **Computational Power:** Training these models requires supercomputers with thousands of powerful processors running for weeks or months.

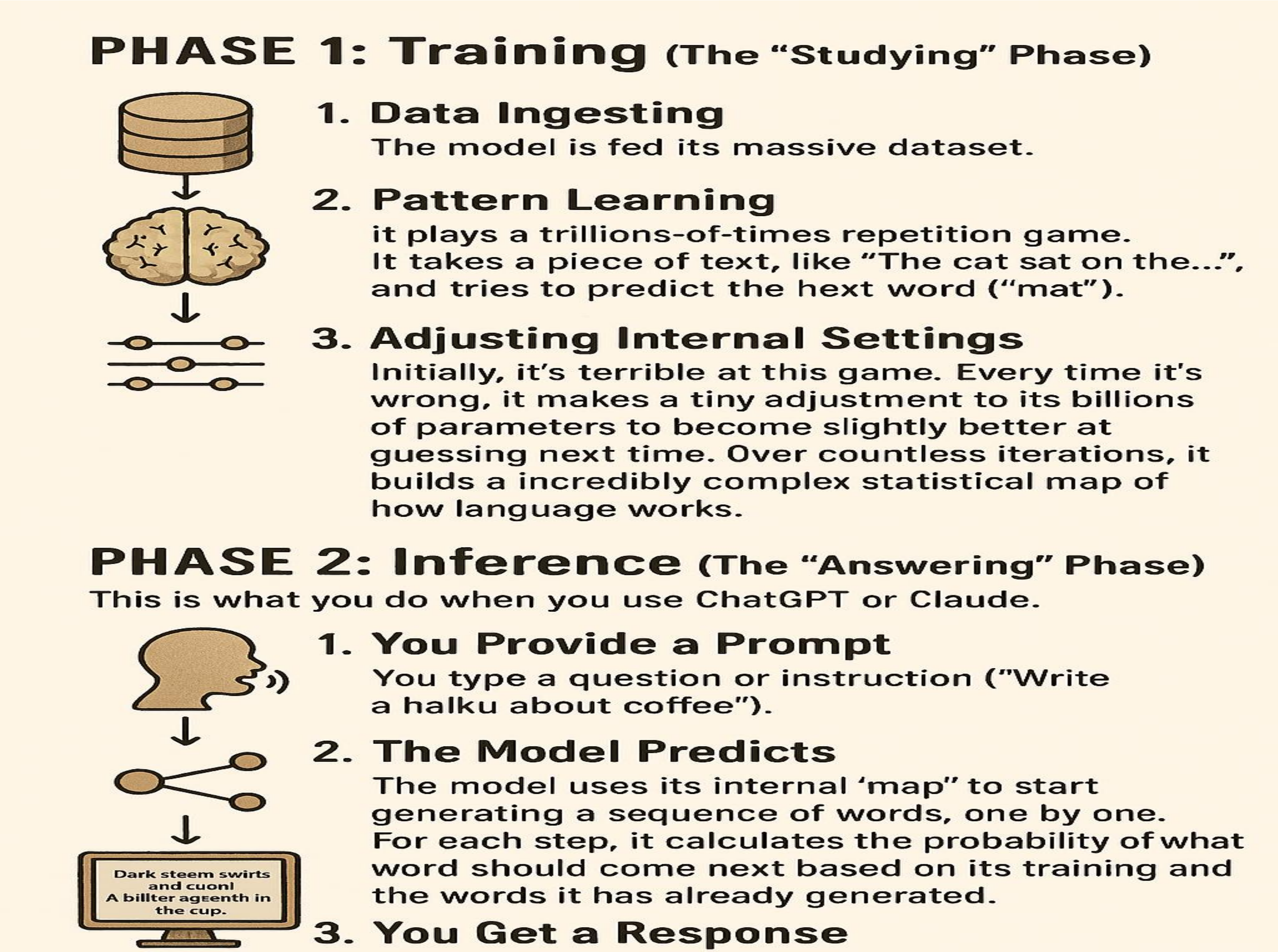
TYPES OF LLM


Autoregressive LM
Why does the chicken cross the
[prediction]
Context
(previous tokens only)
Masked LM
Why does the
[prediction]
cross the road
Context
(surrounding tokens)

AUTOREGRESSIVE MODEL



- **Analogy:** It's an ultra-powerful **autocomplete**.



PHASE 1: Training (The “Studying” Phase)

1. Data Ingesting

The model is fed its massive dataset.

2. Pattern Learning

it plays a trillions-of-times repetition game. It takes a piece of text, like “The cat sat on the...”, and tries to predict the hext word (“mat”).

3. Adjusting Internal Settings

Initially, it’s terrible at this game. Every time it's wrong, it makes a tiny adjustment to its billions of parameters to become slightly better at guessing next time. Over countless iterations, it builds a incredibly complex statistical map of how language works.

PHASE 2: Inference (The “Answering” Phase)

This is what you do when you use ChatGPT or Claude.

1. You Provide a Prompt

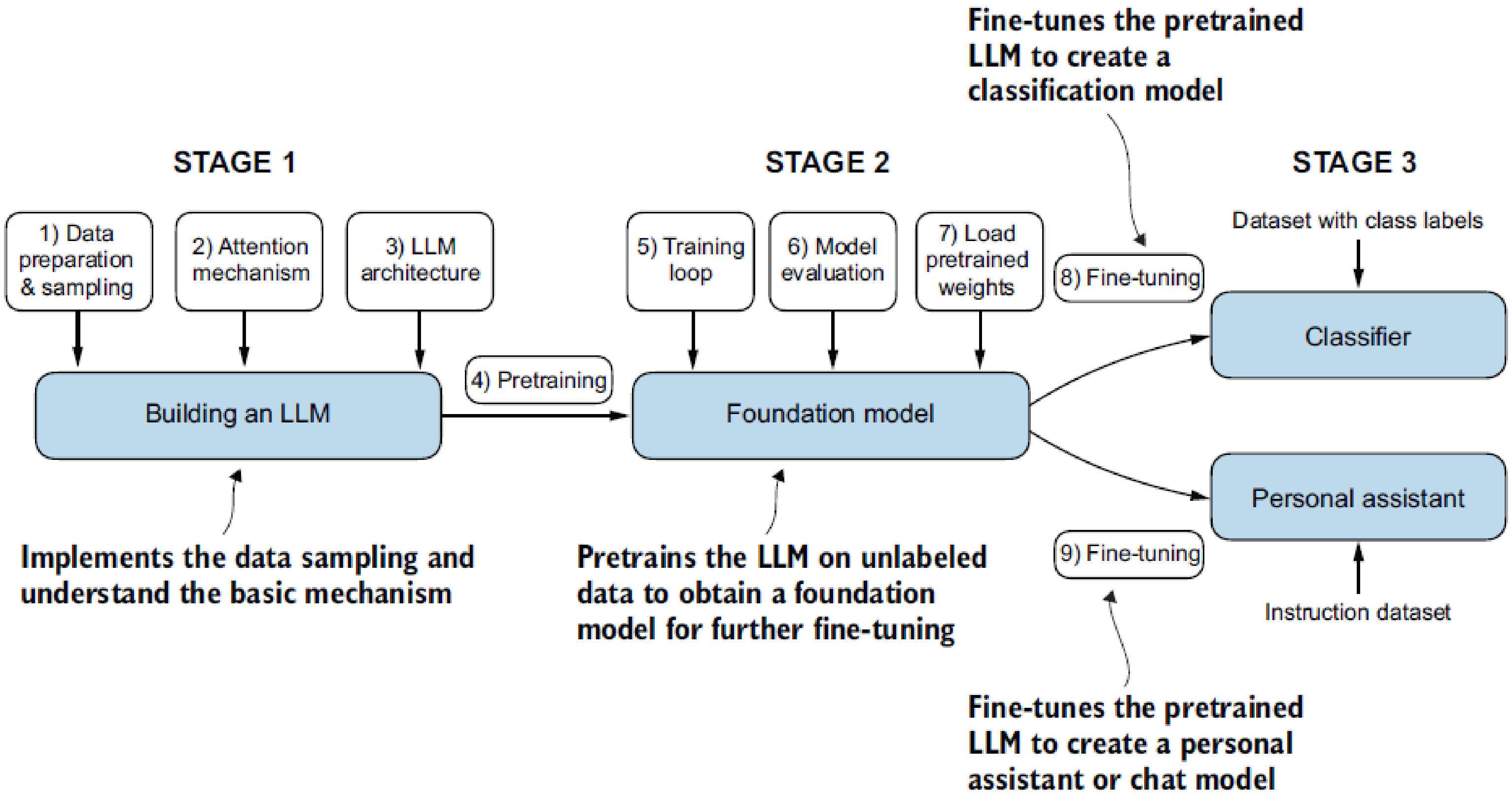
You type a question or instruction ("Write a halku about coffee").

2. The Model Predicts

The model uses its internal ‘map” to start generating a sequence of words, one by one. For each step, it calculates the probability of what word should come next based on its training and the words it has already generated.

3. You Get a Response

LLM LEARNING FROM SCRATCH



Fine-tunes the pretrained LLM to create a classification model
STAGE 1
STAGE 2
STAGE 3
1) Data preparation & sampling
2) Attention mechanism
3) LLM architecture
5) Training loop
6) Model evaluation
7) Load pretrained weights
8) Fine-tuning
Dataset with class labels
Classifier
4) Pretraining
Building an LLM
Foundation model
Personal assistant
9) Fine-tuning
Instruction dataset
Implements the data sampling and understand the basic mechanism
Pretrains the LLM on unlabeled data to obtain a foundation model for further fine-tuning
Fine-tunes the pretrained LLM to create a personal assistant or chat model

Key Concepts for Beginners to Know

- **Prompt:** The instruction or question you give to the LLM. The quality of your prompt directly influences the quality of the output. This leads to...

- **Prompt Engineering:** The art of crafting effective prompts to get the best results. It's like learning how to ask the smart student the right question to get the best answer.
  - *Bad prompt:* "Tell me about Napoleon."
  - *Good prompt:* "Explain the strategic reasons for Napoleon's defeat at the Battle of Waterloo in three simple bullet points for a high school student."

- **Hallucination:** This is when an LLM generates confident-sounding but incorrect or nonsensical information. Remember, it's a pattern predictor, not a truth-teller. It can invent facts, books, or URLs that don't exist. **Always fact-check important information from an LLM.**

- **Fine-Tuning:** After the initial massive training, a model can be further trained on a specific dataset (e.g., medical textbooks or legal documents) to become an expert in that niche.

What are LLMs Good and Bad at?

Index	What They're GOOD At	What They're BAD At
1.	**Generating text** (stories, emails, marketing copy)	**Factual accuracy & truthfulness** (They hallucinate)
2.	**Translating languages**	**Understanding complex, real-world context** (e.g., the emotional weight of a real event)
3.	**Answering questions** (based on their training data)	**Real-time reasoning** (Their knowledge is frozen at their last training date)
4.	**Summarizing long documents**	**Consistently performing logical or mathematical operations** (They approximate, don't calculate)
5.	**Writing and debugging code** (by recognizing code patterns)	**Having opinions, consciousness, or understanding** (They simulate conversation, they don't "think")

HOW YOU CAN START PLAYING WITH LLMS TODAY ?

The best way to learn is by doing! Here are some safe, free ways to experiment:

- **ChatGPT (OpenAI):** The most famous one. Great for conversation, brainstorming, and creative writing.
- **Claude (Anthropic):** Known for having a friendly "personality" and being good at handling long documents.
- **Gemini (Google):** Deeply integrated with Google's search knowledge (but still check for hallucinations!).
- **Hugging Face:** A website that hosts thousands of open-source LLMs you can try for free, often for more specific tasks.

Beginner's Exercise

Go to one of these tools and try these prompts:

- "Explain how a bicycle works to a 5-year-old."
- "Give me three ideas for a birthday party with a dinosaur theme."
- "Write a Python function to calculate the factorial of a number.“
- “The secret to a good life is _____”
- “The capital of France is”
- “for i in range(10):”
- “Artificial intelligence can be used to”

OLLAMA SETUP

Index	Step 1: Installation	Step 2: Verify Installation & Your First Command
1.	**Visit the Ollama Website:** Go to https://ollama.com	Open your terminal (Command Prompt, PowerShell, or any shell on Mac/Linux).
2.	**Download:** Click the download button for your operating system (macOS, Windows, or Linux).	**Pull your first model:** This downloads a medium-sized, very capable model to your machine.
3.	**Install:** Run the downloaded installer. It will set up the Ollama background service and command-line tool.	➢Bash commands ➢*ollama pull llama3.1* – **Download model** ❖ Once downloaded, you can start chatting with it directly in the terminal. ➢*ollama run llama3.1* - **Run model** ❖ You should see a >>> prompt. Type a hello message and press enter! **>>> Hello! How are you today?**

To exit the chat session, type /bye.

Note: Recent Chat interface is also good.(Discuss)

USEFUL OLLAMA COMMANDS

- ***ollama list*** - Shows all models you've downloaded.
- ***ollama run <model-name>*** - Start a continuous chat session with a model.
- ***ollama run <model-name> "Your prompt here"*** - Send a single prompt and get an answer.
- ***ollama pull <model-name>*** - Download a new model (e.g., ollama pull mistral, ollama pull phi3).
- ***ollama rm <model-name>*** - Delete a model from your machine to free up space.

Curated Links

- https://poloclub.github.io/transformer-explainer/ - Transformer Playground
- https://huggingface.co/spaces/ShreyaL/NLP_preprocessing_playground?logs=container – Text Pre-processing Playground
- https://huggingface.co/spaces/Xenova/the-tokenizer-playground - Tokenizer Playground
- https://www.promptingguide.ai/techniques/cot - Guide for Prompting
- https://jalammar.github.io/illustrated-transformer/ - Visual illustration of Transformer
- https://www.deeplearning.ai/short-courses/chatgpt-prompt-engineering-for-developers/ - Short Free Course on Prompt Engineering
- https://huggingface.co/blog/getting-started-with-embeddings - Embeddings article

Thank You!

Questions?