Lecture 5

How does GPT-3 really work?

Generative pre-training is the process of training a model on large amounts of text data to predict the next word in a sequence. This helps the model learn general language patterns, grammar, and context. It is called “generative” because the model learns to generate text, and “pre-training” because it happens before fine-tuning on specific tasks.

It’s done in unsupervised manner.

GPT-3 has been inspired by the transformer architecture but **it doesn’t use the encoder.**

Zero-shot learning: The model performs a task it hasn’t been explicitly trained on, using only its general knowledge.

Few-shot learning: The model learns a new task from a small number of examples or demonstrations provided during inference.

Open-source LLMs:

Their code, model weights, and training details are publicly available. Developers can modify, retrain, or deploy them freely (e.g., LLaMA, Mistral).

Closed-source LLMs:

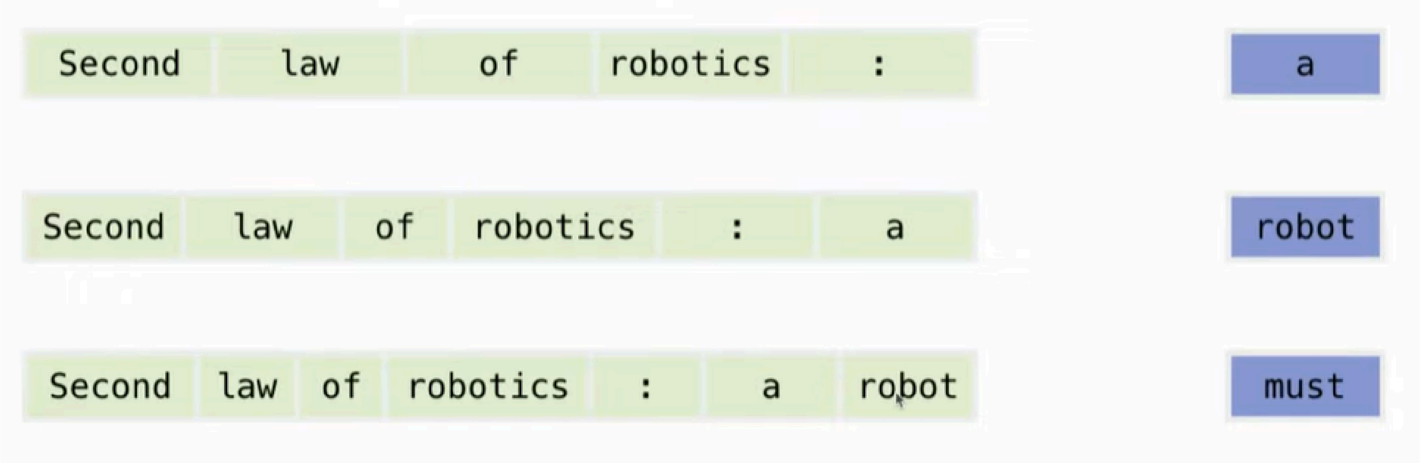
Their internal architecture, data, and weights are kept private by the company. They’re accessible only through APIs or limited interfaces (e.g., GPT, Claude).

Auto-regressive models:

These models generate text by predicting the next word based on all previous words — one token at a time. Each new word depends on what came before it.

Why GPTs are auto-regressive:

GPTs (Generative Pre-trained Transformers) are trained to predict the next token in a sequence. **They use the previous outputs as inputs for the future predictions.** This makes them naturally auto-regressive, as they build sentences step-by-step while maintaining context from earlier tokens.



Second	law	of	robotics	:			a
Second	law	of	robotics	:	a		robot
Second	law	of	robotics	:	a	robot	must

When the model (like GPT) is being **trained**, it sees tons of sentences from books, articles, websites, etc. During training, it *does* get to see the correct next word (the “label”) — that’s how it learns. It predicts a word, compares it to the true word, and adjusts its internal parameters slightly to get better next time. This happens billions of times until it becomes really good at predicting what “sounds right.”

But — and this is the key part — once training is done, the model **doesn’t store sentences or data**. It only stores **patterns** and **statistical relationships** between words, concepts, and contexts inside its parameters.

So if you give it something completely new — say, a sentence or question that no one has ever written before — it doesn’t “look it up.” Instead, it **uses what it has learned about language and reasoning** to *generate a likely continuation or answer*.

So, we don't collect labels for training data, instead we use the structure of the data itself. The next word in the sentence is used as the label.

The original transformer architecture has 6 encoders and decoders, and the GPT-3 model has 96 transformer layers, 175B parameters.

Emergent behavior refers to new abilities or skills that a model begins to show **spontaneously** as it grows larger or more capable — even though it was **never explicitly trained** for those tasks.

As language models become more powerful through generative pre-training, they start performing tasks like question answering or reasoning *without being specifically taught to do so*. These capabilities "emerge" naturally from the model's understanding of language and world knowledge learned during training.

In short, **emergent behavior** is when complex, higher-level abilities appear unexpectedly from the model's scale and training, not from direct supervision.