Lecture 3: Stages of Building Large Language Models

1. The Two-Stage Pipeline

Building an industrial-strength LLM is not a single step; it involves two distinct phases:

1. **Pre-training:** Training on a massive, diverse dataset to create a "Foundational Model".
2. **Fine-tuning:** Refining that model on a narrower, specific dataset to optimize it for a specific task or domain,.

---

2. Stage One: Pre-training (Creating the Foundation)

This stage answers the question: *How does an LLM know so much about the world and grammar?*

- **The Data (Scale & Diversity):**
  - Pre-training requires a massive corpus of **unlabeled data** (raw text).
  - **GPT-3 Case Study:** GPT-3 (175 billion parameters) was trained on approximately **300 billion tokens** (where 1 token $\approx$ 1 word),.
  - **Data Sources for GPT-3:**
    - **Common Crawl:** 60% of the data (410 billion words scraped from the open internet).
    - **WebText2:** 22% (19 billion words derived from Reddit links, blog posts, etc.).
    - **Books:** ~16% (67 billion words).
    - **Wikipedia:** ~3% (3 billion words).
- **The Training Task (Next Word Prediction):**
  - The model is trained on a simple objective called **"Word Completion"** or **Auto-regression**,.
  - *Example:* Given the input *"The lion is in the..."*, the model learns to predict *"forest"*.
- **The "Magic" Discovery (Emergent Capabilities):**
  - OpenAI (2018) discovered that even though the model is *only* trained to predict the next word, it unintentionally learns to perform complex tasks like translation, sentiment analysis, and summarization without being explicitly trained for them,.
- **The Cost:**
  - Pre-training is computationally expensive. The estimated cost for pre-training GPT-3 was **$4.6 million** due to the GPU power required.
- **Outcome:**
  - This stage produces a **"Foundational Model"** (or Base Model). It is a generalist that knows a lot but isn't specialized,.

---

3. Stage Two: Fine-tuning (Specialization)

This stage answers the question: *How do we make the model work for a specific company or specific job?*

- **Why is it needed?**
    - A pre-trained model (like base GPT-4) is generic. It does not know private company data or specific industry terminologies.
    - **The "Manager" Problem:** If an airline CEO wants a chatbot to answer questions about specific flight prices, a generic model might hallucinate or give general answers. It needs to be trained on the airline's specific data.
- **The Data (Labeled):**
    - Unlike pre-training, fine-tuning usually requires **labeled data** (Inputs paired with correct Outputs).
- **Real-World Success Stories:**
    - **SK Telecom:** Fine-tuned a model specifically for Korean telecom customer service. This resulted in a **35% improvement** in conversation quality compared to using the base model,.
    - **Harvey:** An AI tool for lawyers. A general model lacks knowledge of specific legal case history, so Harvey was fine-tuned on legal documents to assist attorneys reliably,.
    - **JP Morgan:** Developed an internal LLM suite fine-tuned on proprietary banking data to assist analysts.

---

4. Types of Fine-tuning

The lecture categorizes fine-tuning into two primary types based on the goal:

1. **Instruction Fine-tuning:**
    - Teaching the model to follow specific instructions by providing "Instruction-Answer" pairs.
    - *Example:* Training a translator by feeding pairs of {English Sentence} $\rightarrow$ {French Sentence}.
2. **Classification Fine-tuning:**
    - Teaching the model to sort or label text.
    - *Example:* A spam detector. You feed the model 10,000 emails labeled as "Spam" or "Not Spam" so it learns the specific patterns of your email system.

---

5. Summary Schematic

Dr. Dander summarizes the entire building process in three steps-:

Step	Action	Data Type	Outcome
1	**Data Collection**	**Raw Text** (Billions of words)	The Training Corpus
2	**Pre-training**	**Unlabeled Data** (Self-supervised)	**Foundational Model** (Generalist, High Cost)
3	**Fine-tuning**	**Labeled Data**	**Specialized Application** (e.g., Legal Bot, Spam Classifier)