Large Language Models

Learning from Humans

Agenda

- Chapter 4: How LLMs learn from humans
- (If time) Chapter 5: What LLMs can (and cannot) do
- Key debates: prompting, emergence, hallucinations

Three Forms of LLMs

- **Base / Foundation Models**
  - Pretrained on massive text corpora
  - Good at prediction or continuation, not obedience
- **Fine-Tuned Models**
  - Trained on instruction-response pairs
  - Capable of instruction following
- **AI Assistants**
  - Optimized with human feedback (RLHF)
  - Display *preference-aligned* behavior

The LLM Education Pipeline

- Base:
  Prompt: Write a poem about quantum mechanics.
  Response: ...said the physics teacher to his class.

The LLM Education Pipeline

- Base:
  Prompt: Write a poem about quantum mechanics.
  Response: ...said the physics teacher to his class.
- Fine tuning:
  Prompt: Write a poem about quantum mechanics.
  Response:

  > *Much to their lament / after running the double-slit experiment / the physicist finally understood / that quantum is no good.*

Size of Model

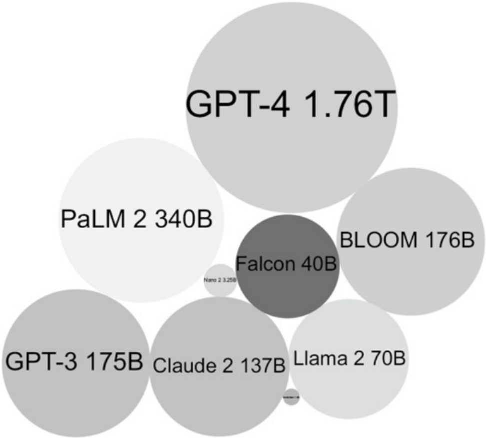

Figure 22 Relative sizes of LLMs, based on (log-)scaled numbers of parameters (weights). The tiny models are the Nano 2 (3.25B parameters) and Gemini Nano 1 (1.8B parameters) models. B = billion; T = trillion.

Bond Hall, WWU

Will models get larger and larger?

Cost vs Capability

- Moving from base to fine-tuned to RLHF increases cost.
  - Computational cost
  - Human labor
  - Model usefulness
- Fine-tuning and RLHF are expensive
- Not all applications require AI assistants

Continued Pretraining

- Reintroducing a model to raw language data
- Used to:
  - Add new languages
  - Strengthen weak domains
- Produces a *new base model*, not alignment
- Computationally intensive

In-Context Learning (Prompting)

- Learning *without* updating model weights
- Uses the context window as working memory
- Types:
  - Zero-shot
  - One-shot
  - n-shot
- Core mechanism behind prompt engineering

Example from Text

Prompt:

```
Sentiment: This is a great poem
Output: bla1
Sentiment: This book is awkward
Output: bla2
Sentiment: An amazing movie
Output: ?
Sentiment: I loved this book
Output: ?
```

Answer:

Example from Text

Prompt:

```
Sentiment: This is a great poem
Output: bla1
Sentiment: This book is awkward
Output: bla2
Sentiment: An amazing movie
Output: ?
Sentiment: I loved this book
Output: ?
```

Answer:

```
bla3
bla4
```

Gemini

Sentiment: This is a great poem.
output: bla1
Sentiment: This book is awkward.
output: bla2
Sentiment: An amazing movie....

Show thinking

Based on the pattern you've established, it looks like **bla1** represents positive sentiment and **bla2** represents negative sentiment.

Since both of your new examples are glowing reviews, here is the completion:

Sentiment: An amazing movie. output: **bla1**

Sentiment: I loved this book. output: **bla1**

Would you like me to generate a few "bla2" examples for you to test, or perhaps try a more complex sentiment category?

Gemini

1. The "Medium" Pattern

In this version, the code doesn't care about feelings—it only cares about what is being reviewed. If **Poem = bla1** and **Book = bla2**, then a new medium (Movie) would get a new code.

- **Logic:** Assign a unique ID to each type of media.
- **An amazing movie:** `bla3` (A new category)
- **I loved this book:** `bla2` (Matches the "Book" category)

2. The "Word Count" Pattern

If we look at the number of words in each sentence, we see a correlation between length and the number in the output.

- **Logic:** "This is a great poem" (**5 words**) = `bla1`. "This book is awkward" (**4 words**) = `bla2`. The pattern is: *Lower word count = Higher "bla" number.*
- **An amazing movie (3 words):** `bla3`
- **I loved this book (4 words):** `bla2`

---

3. The "Keyword" Trigger

Sometimes these patterns are binary based on a single word. In your examples, the only sentence with the word "book" resulted in `bla2`.

- **Logic:** If the string contains "book", output `bla2`. Otherwise, output `bla1`.
- **An amazing movie:** `bla1`
- **I loved this book:** `bla2`

Why Prompting Works

- Does *not* teach new knowledge
- Teaches how to *use* existing knowledge
- Possible explanations:
  - Emergent behavior at scale
  - Bayesian/statistical inference
  - Semantic indexing during pretraining

Advanced Prompting Techniques: Chain of Thought

Example: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 balls. How many balls does he have now? Think through this step-by-step.

Advanced Prompting: Zero-Shot "Think step by step"

Simply adding the phrase "Let's think step by step" to a prompt can dramatically increase an AI's accuracy on complex tasks, even if you don't give it any examples of how to solve the problem.

Advanced Prompting: Zero-Shot "Think step by step"

Simply adding the phrase "Let's think step by step" to a prompt can dramatically increase an AI's accuracy on complex tasks, even if you don't give it any examples of how to solve the problem.

In newer models,"step-by-step" process is often baked into the model's architecture.

Advanced Prompting: Socratic Prompting

"I want you to evaluate the feasibility of opening a cat-themed coffee shop in downtown Seattle. Before giving me your final analysis, follow these steps:

- **Sub-questions:** Identify 3 critical questions you need to answer to determine the feasibility.
- **Investigation:** Answer those 3 questions one by one.
- **The 'Devil's Advocate'** Look at your own answers and identify one major assumption you made that might be wrong.
- **Synthesis**: Provide your final recommendation based on this internal dialogue."

Prompting Is Unstable

- Sensitive to phrasing and punctuation
- Order effects in n-shot prompting
- Temperature controls randomness
- Evaluation becomes difficult

Supervised Fine-Tuning (SFT)

- Makes instruction learning permanent
- Trains on curated prompt–response datasets
- Generalization depends on:
  - Task novelty
  - Prompt clarity
  - Prior exposure

Efficient Fine-Tuning

- Parameter-efficient methods:
  - Pruning
  - Low-rank adaptation
  - Quantization (e.g., 4-bit models)
- Reduce memory and compute cost

Reinforcement Learning from Human Feedback

- Humans rank and evaluate model outputs
- Reward model guides optimization
- Produces preference-aligned behavior
- Variant: Direct Preference Optimization (DPO)

End of Chapter 4

From language models to aligned AI assistants

Emergent Abilities

- Abilities appearing only at large scale
- Not explicitly trained
- Analogy: ant colonies

Defining Emergence

An ability is emergent if it:

1. Performs better than random guessing
2. Was not explicitly trained
3. Does not appear at smaller scale

Examples of Emergent Abilities

- Toxicity detection
- Truthful question answering
- Arithmetic reasoning
- Multistep reasoning (StrategyQA)

Multilingual Transfer

- Multilingual LLMs outperform monolingual ones
- Even tiny language fractions matter
- Transfer learning across languages
- Code-switching remains difficult

Is Emergence Real?

- Performance curves resemble power laws
- Apparent phase transitions
- Stanford critique:
  - Wrong metrics exaggerate emergence
  - Smoother progress with better metrics

Why Emergence Matters

- Control and safety concerns
- Unexpected behavior at scale
- Need for reliable capability assessment

Key Limitations of LLMs

- Numerical imprecision
- Weak deductive reasoning
- Hallucinations and fabrication

Hallucinations

- Fabricated facts
- Confident but false explanations
- Influenced by context and titles
- Psychological analogies:
  - Confabulation
  - Source amnesia

Factual Knowledge Limits

- LLMs are not databases
- Training data is static
- Latency and outdated information
- Partial fix: tool use and web search

Looking Ahead

- Can LLMs be creative?
- Can they generate genuinely new knowledge?
- Chapter 6: Creativity

Questions for Discussion