M-Lab CARTE AI Workshop 2025

Large Language Models

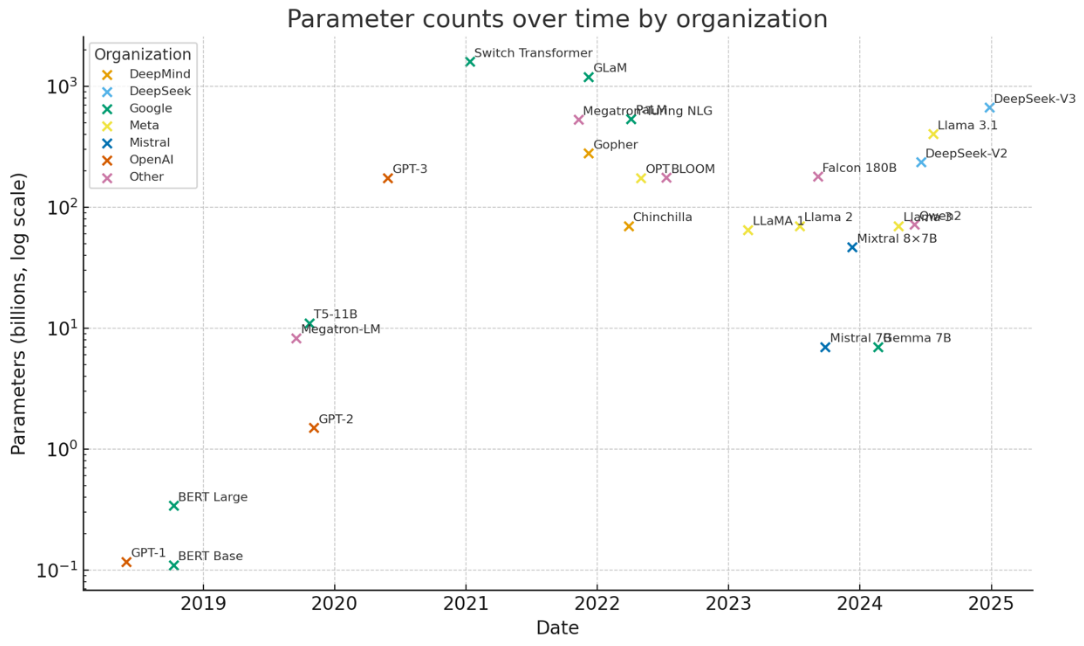
Language Models

- Estimates the likelihood of a sequence of words occurring
- To generate text, select the word most likely to appear next
- How do we estimate likelihood? By looking at lots of text
- Simple approach: look up the number of times a sequence occurs

$$P(The, dog, and, the, cat) > P(The, dog, and, the, ostrich)$$

Large Language Models

- Latest models can learn from much more data, and capture more complex nuance
- Key factors in model scaling:
  - Growth in data availability over the last two decades
  - Technological performance improvements (GPUs, TPUs)
  - Algorithmic improvements (transformers)
  - Greater willingness to invest

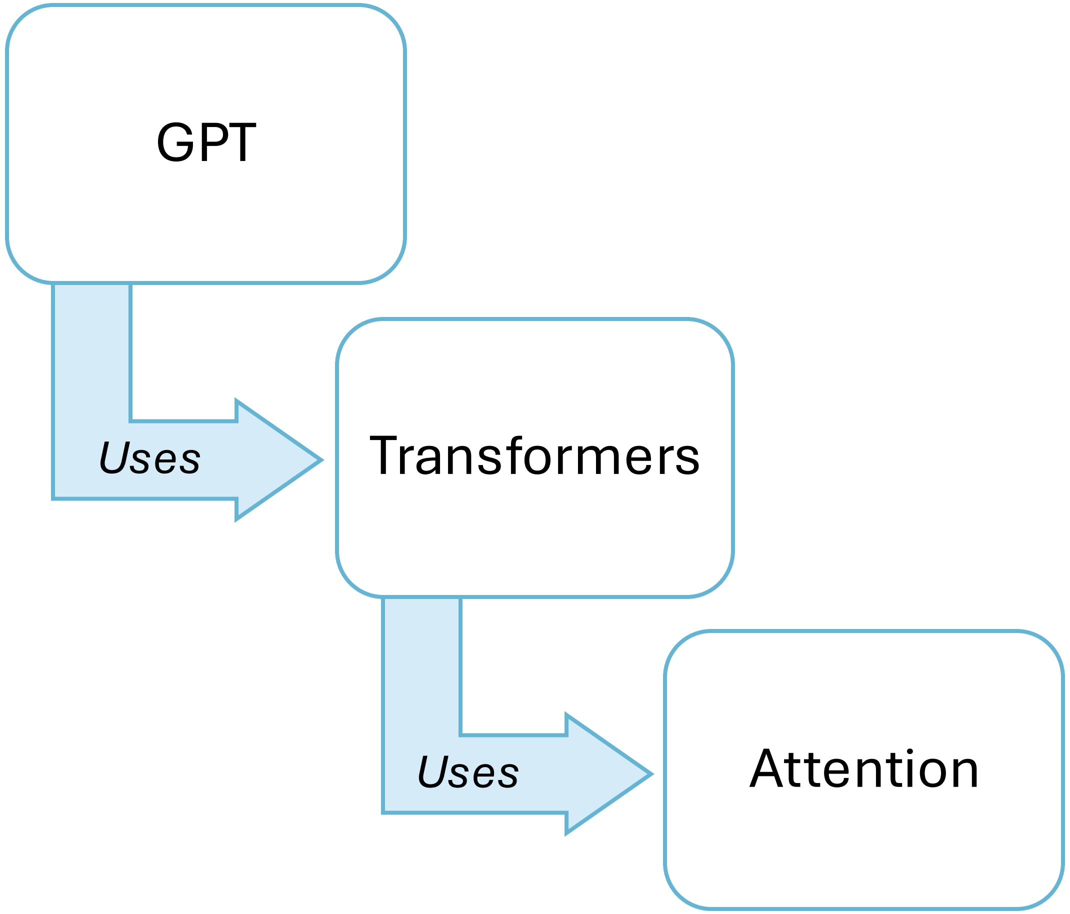


UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE & ENGINEERING
Centre for Analytics and Artificial Intelligence Engineering

Key Terminology

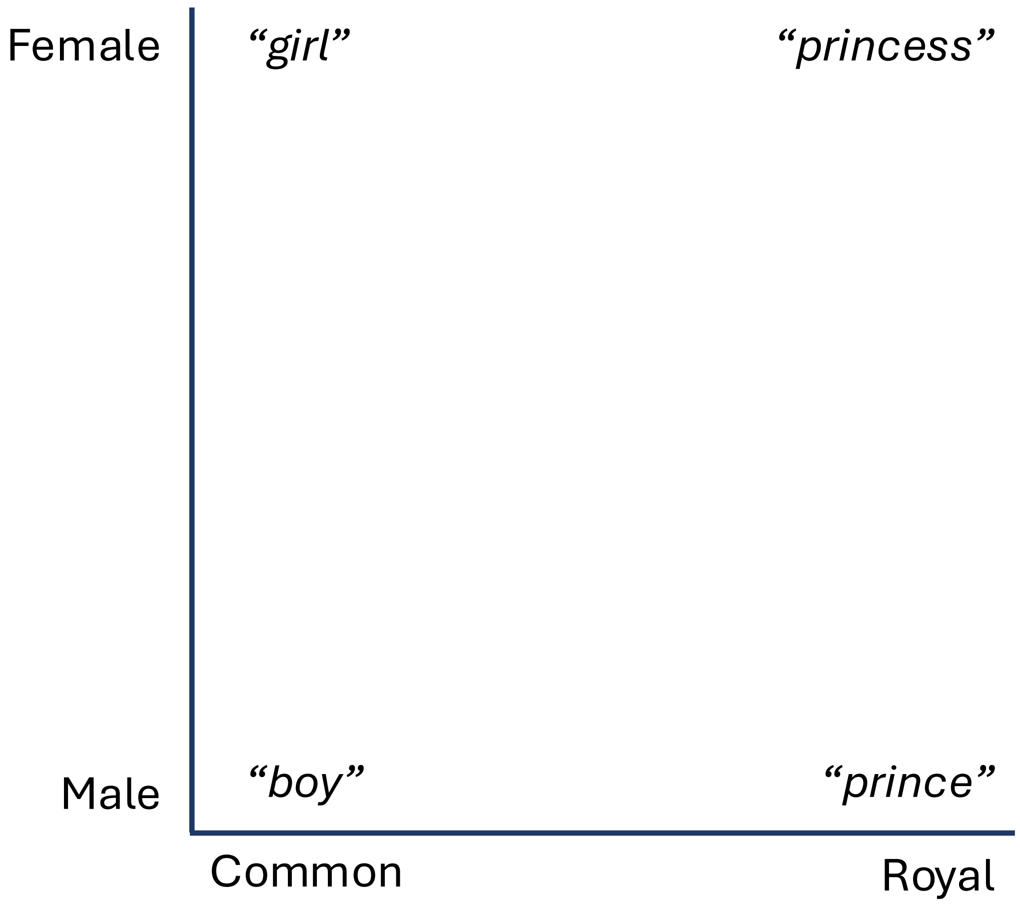
- *Attention:* machine learning method allowing for greater understanding of sentence structure (and beyond)
- *Transformer:* neural network architecture utilizing attention
- *GPT:*
  - *Generative:* creates "new" content
  - *Pre-Trained:* two-step training process to produce final model
  - *Transformer:* as above

Key Technology



How does an LM “understand” word meaning?

- To predict the likelihood of a word, we must have some sense of its meaning
- We can organize words along dimensions that represent different aspects of their meaning
- Easy to do for a few senses, but words have unlimited ways to conceptualize



How does an LM “understand” word meaning?

- To effectively model word meaning, new LMs use thousands of dimensions to represent each word in their vocabulary
- This allows the system to capture complex word meaning *before* any prediction is carried out
- But how do we determine these values?

Building word embeddings

- We have access to a vast quantity of *unstructured* text data
  - Reminder: this is data without useful labels or organization
- For learning embeddings, having examples of how words are *used* should be enough!
- We start by drawing random samples of text from a large data set (e.g. Wikipedia)

Word Embedding Visualizer

https://word-embedding-visualizer-871047044699.us-west1.run.app/

https://projector.tensorflow.org/

Tokenization

- LLMs may encounter words they don't recognize
- We can often predict the meaning of a word by breaking it into chunks
- *Tokenization* is a process where words are broken down into more familiar parts, if needed

- eat
- eating
- work
- working
- worked
- florming
- flormed

Learning to predict

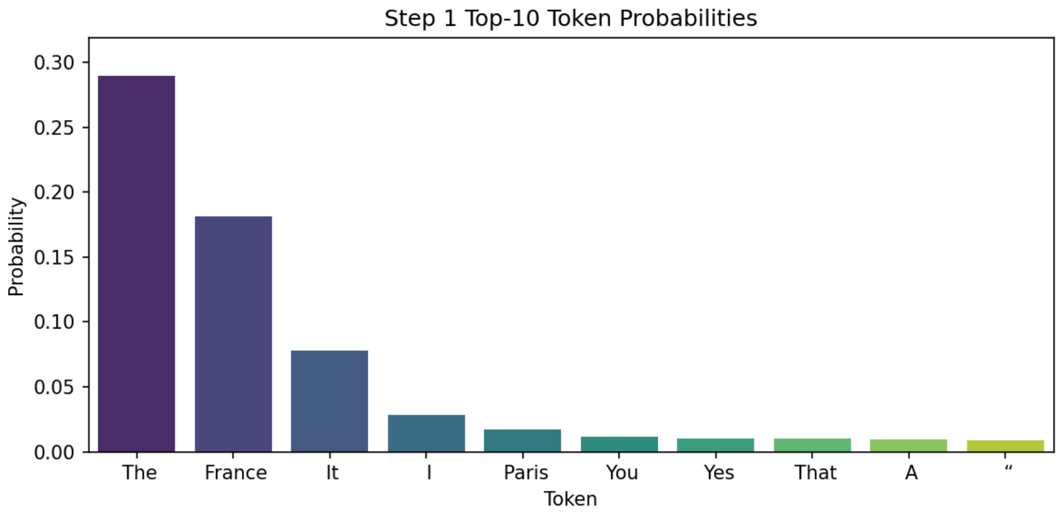
- At the first step, the model is shown examples of phrases with one word missing
- It must predict the correct word to fill that space
- Because we know how *similar* words are, we can say if a prediction is better or worse

- `The ____ purred loudly`
  - `Answer: cat`
  - `Good guesses:`
    - `lion`
    - `leopard`
    - `bobcat`
  - `Bad guesses:`
    - `dog`
    - `train`
    - `running`

Text Generation

- LLMs generate text by estimating the probability of *each token in the vocabulary* coming next after all the input
- The token to show to the user is semi-randomly selected, with weighting by estimated likelihood

```
User: What's the capital of France?
```

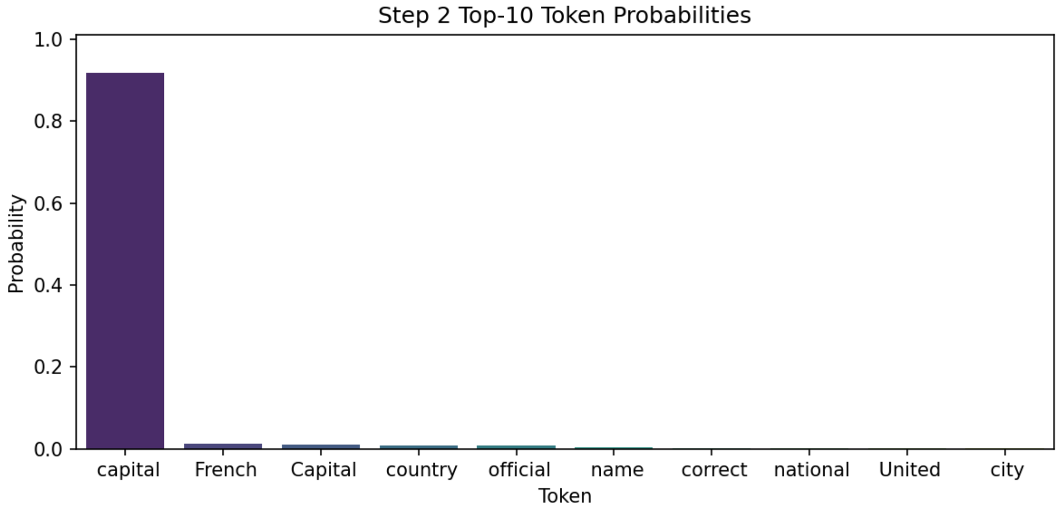



```
System: The
```

Text Generation

- LLMs generate text by estimating the probability of *each token in the vocabulary* coming next after all the input
- The token to show to the user is semi-randomly selected, with weighting by estimated likelihood

```
User: What’s the capital of France?
```

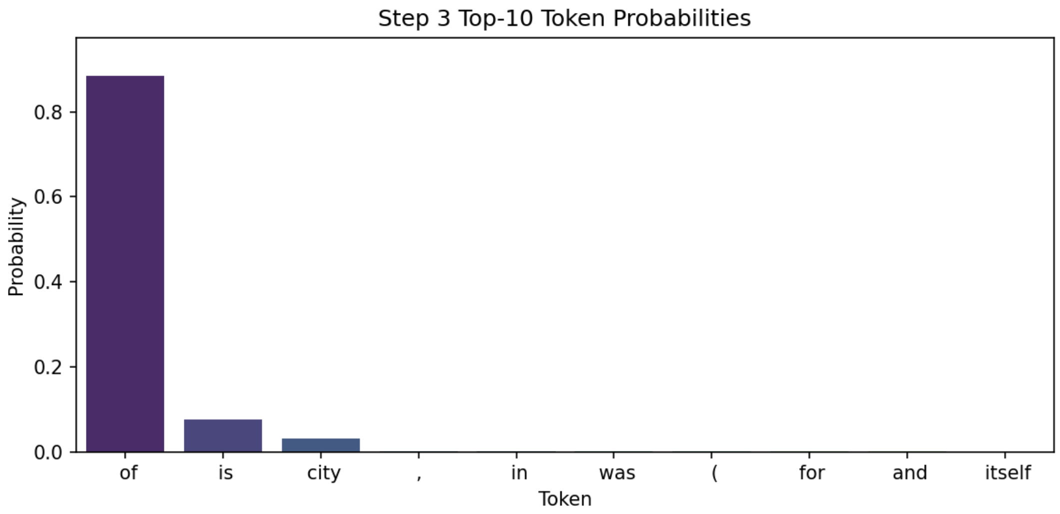



```
System: The capital
```

Text Generation

- LLMs generate text by estimating the probability of *each token in the vocabulary* coming next after all the input
- The token to show to the user is semi-randomly selected, with weighting by estimated likelihood

`User: What's the capital of France?`

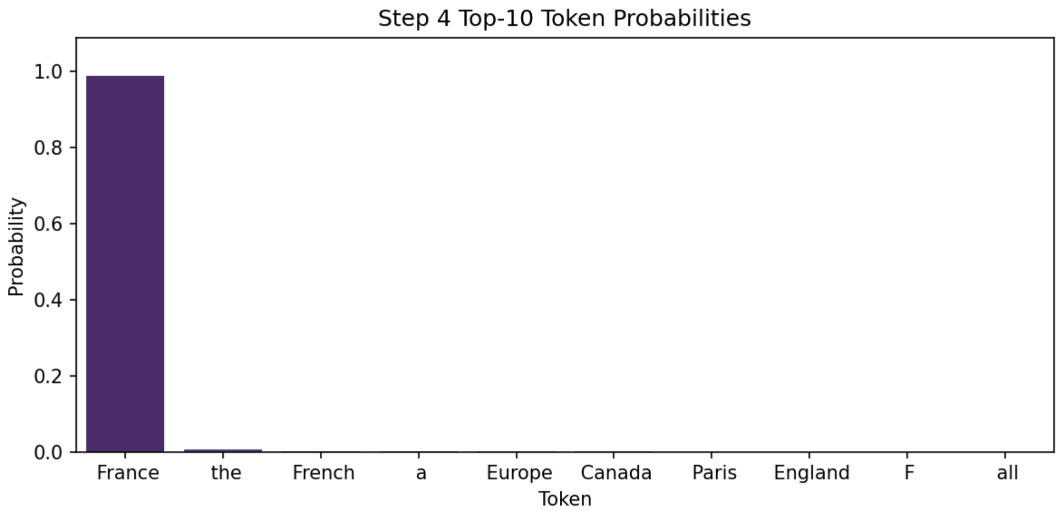


`System: The capital of`

Text Generation

- LLMs generate text by estimating the probability of *each token in the vocabulary* coming next after all the input
- The token to show to the user is semi-randomly selected, with weighting by estimated likelihood

`User: What's the capital of France?`

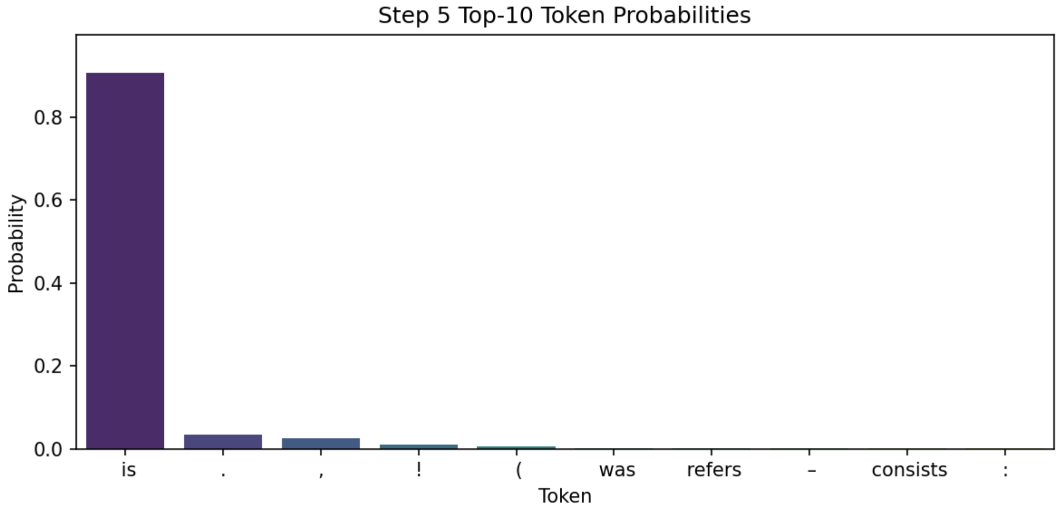


`System: The capital of France`

Text Generation

- LLMs generate text by estimating the probability of *each token in the vocabulary* coming next after all the input
- The token to show to the user is semi-randomly selected, with weighting by estimated likelihood

```
User: What’s the capital of France?
```

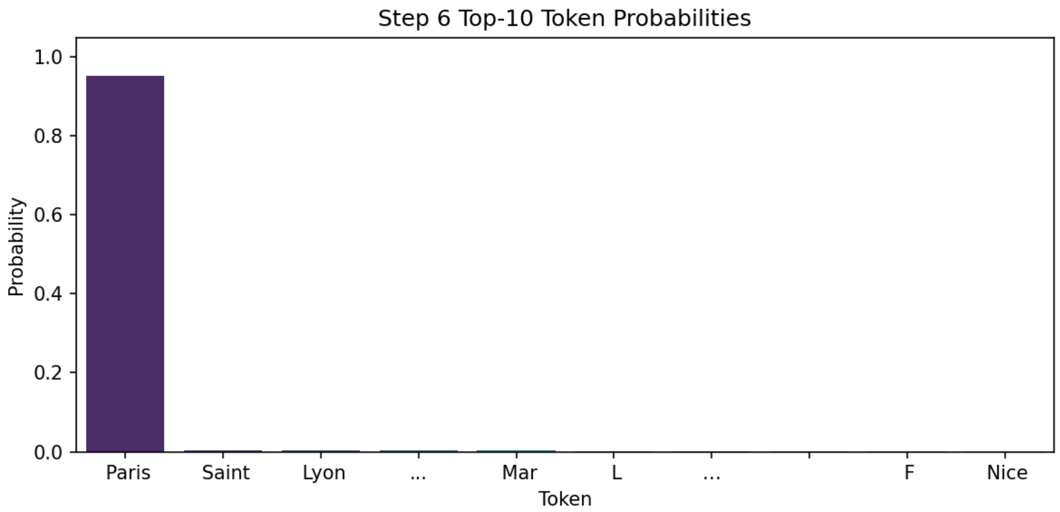



```
System: The capital of France is
```

Text Generation

- LLMs generate text by estimating the probability of *each token in the vocabulary* coming next after all the input
- The token to show to the user is semi-randomly selected, with weighting by estimated likelihood

`User: What’s the capital of France?`

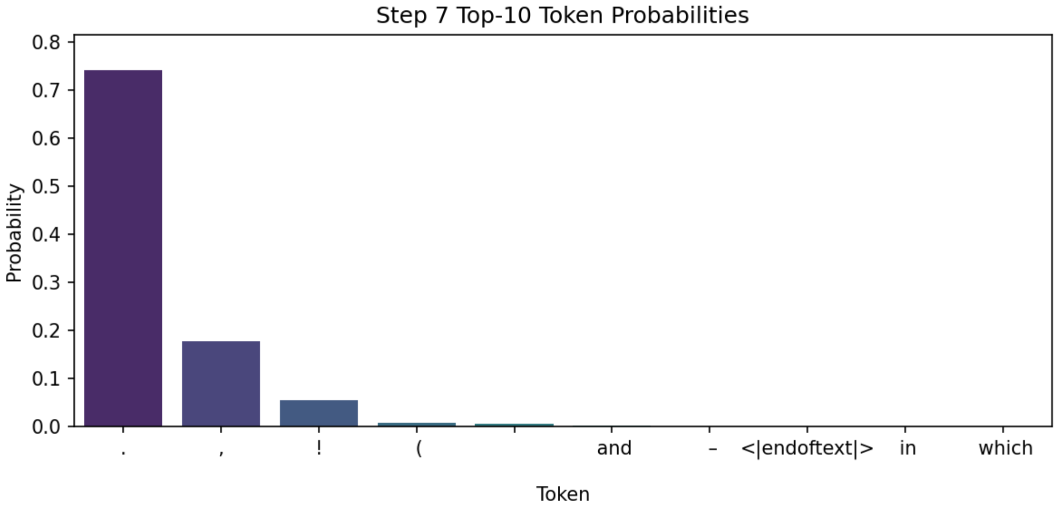


`System: The capital of France is Paris`

Text Generation

- LLMs generate text by estimating the probability of *each token in the vocabulary* coming next after all the input
- The token to show to the user is semi-randomly selected, with weighting by estimated likelihood

`User: What’s the capital of France?`



`System: The capital of France is Paris.`

Token Prediction Visualization

https://gemini-token-predictor-871047044699.us-west1.run.app/

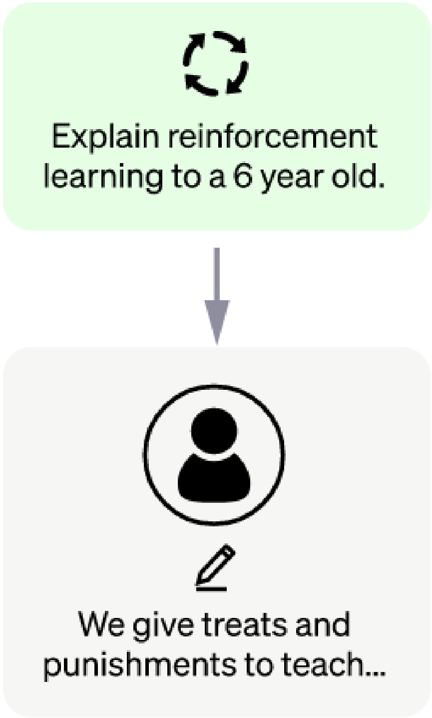
GPT's Training Data

- 1 token ≈ ¾ word
- Some datasets are sampled more times than others
- Common Crawl: billions of webpages collected over 7 years
- Webtext2: Dataset of webpages that have been shared on Reddit
- Books1: Free ebooks
- Books2: Unknown
- English Wikipedia

Dataset	Quantity (tokens)	Weight in training mix

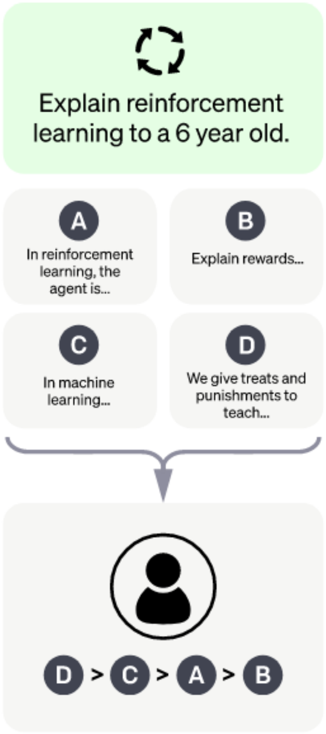
The training innovation of ChatGPT

Human annotators write answers to questions



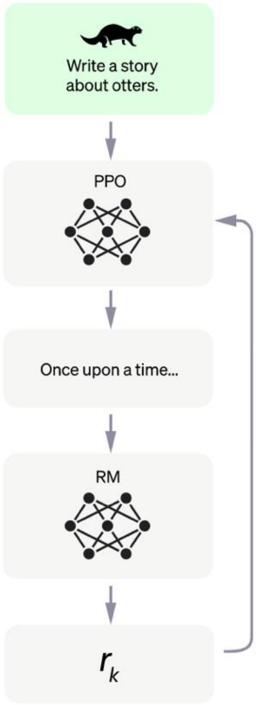
The generalist GPT model is taught from these Q&A pairs

Human annotators write <u>more</u> answers, and someone else ranks them



A <u>separate</u> model learns to rate the quality of an answer

GPT writes answers to sampled questions



The reward model rates each answer, allowing GPT to keep learning

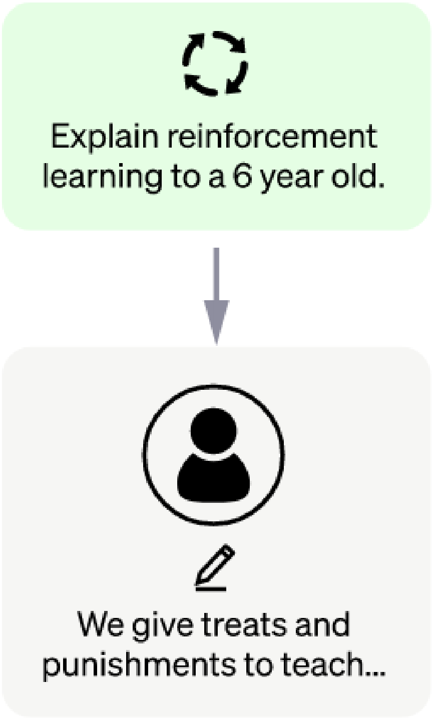
NG

Figures: https://openai.com/blog/chatgpt/

The training innovation of ChatGPT

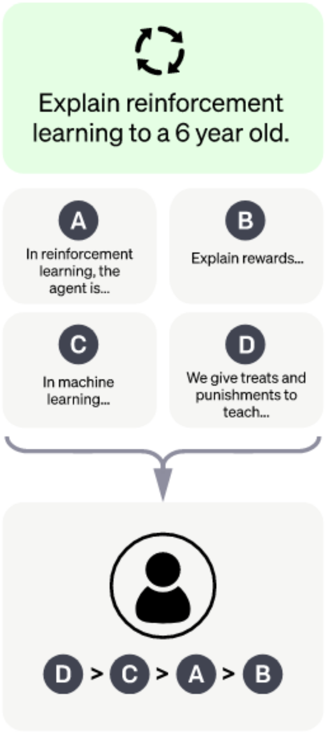
No more humans involved!

Human annotators write answers to questions



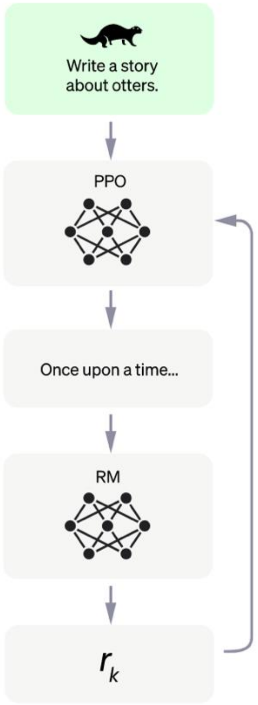
The generalist GPT model is taught from these Q&A pairs

Human annotators write more answers, and someone else ranks them



A separate model learns to rate the quality of an answer

GPT writes answers to sampled questions



The reward model rates each answer, allowing GPT to keep learning

NG

Figures: https://openai.com/blog/chatgpt/

Prompt Construction

- We'll begin with a *user prompt*: the message that the user sends to the system.

- `User: What's the capital of France?`

Prompt Construction

- We'll begin with a *user prompt*: the message that the user sends to the system.
- Our system doesn't just receive this prompt. It also receives a *system prompt*, which primes the model to behave how we'd like.

- `System: You are a helpful assistant that provides clear, concise, and accurate answers. When answering, you always give context and explain your reasoning where appropriate.`
- `User: What's the capital of France?`

Prompt Construction

- We'll begin with a *user prompt*: the message that the user sends to the system.
- Our system doesn't just receive this prompt. It also receives a *system prompt*, which primes the model to behave how we'd like.
- Tools like ChatGPT may also supply "memories" about the user, or user-defined instructions.

- System: You are a helpful assistant that provides clear, concise, and accurate answers. When answering, you always give context and explain your reasoning where appropriate.
- Memories:
  - 2024-04-08 User asked for recommendations on modern philosophy. Recommended "The History of Philosophy" by A.C. Grayling
  - 2024-03-15 User reported trouble installing Python libraries on a Mac. Explained how to use Pip and Homebrew to install Python packages.
- User Profile:
  - Name: Alex
  - Profession: Senior Research Associate
  - Interaction Style: professional and concise
- User: What's the capital of France?

Finding out about your own usage

- You can use the text on the right to prompt ChatGPT to describe the information it keeps about you
- You might be surprised about what it knows!
- For me: nearly five thousand words of background

```
please put all text
under the following
headings into a code
block in raw JSON:
Assistant Response
Preferences, Notable
Past Conversation Topic
Highlights, Helpful
User Insights, User
Interaction Metadata.
Complete and verbatim.
```

Key Players - ChatGPT

- Developed by OpenAI, founded 10 years ago by many top names
- Versatile models with strong reasoning and state-of-the-art features
- Often high cost compared to competitors, and scores worse on AI Safety benchmarks than some

Key Players – Microsoft Copilot

- Same models as ChatGPT behind the scenes
- Deep integration with Microsoft products
- Built for productivity and enterprise
- Slightly behind ChatGPT in model releases, but with Microsoft support

Key Players – Google Gemini

- Very strong at multilingual capabilities, and competitive with ChatGPT in performance (today)
- Some criticism that Google has raced to release products before they are fully ready in order to catch up with OpenAI

Smaller Players

- Anthropic (Claude): also strong in enterprise settings, high emphasis on AI safety and reasoning
- Mistral: EU-based company with strong emphasis on privacy and low cost
- Perplexity: strong in real-time web search + citations in answers, good transparency and research usability
- xAI (Grok): strong in live data; built for up-to-the-minute relevance

Enterprise Considerations

Deploying LLMs

- On-premises vs Cloud vs Hybrid
- On-prem: Full control, high upfront cost, your hardware
- Cloud (API): Fast to start, pay-per-use, vendor controls updates
- Hybrid: Sensitive data on-prem, general queries to cloud
- Choice depends on: data sensitivity, budget, technical capacity

Data Security & Privacy

- What data leaves your organization?
- Cloud APIs: Your prompts may be used for training (check terms)
- Enterprise agreements: Data opt-out, dedicated instances
- On-prem: Full control, but you manage security
- Compliance: GDPR, HIPAA, industry-specific requirements
- Question: Can this model see our proprietary data?

Cost & Performance Trade-offs

- API pricing: per-token, adds up fast at scale
- Latency: Cloud adds network delay, on-prem can be faster
- Model size: Bigger = better quality but slower & more expensive
- Customization: Fine-tuning, RAG, or prompt engineering?
- Hidden costs: Integration, monitoring, quality assurance
- ROI timeline: When does deployment pay for itself?

Questions?