AI Explained

How AI Learns

By Dr Peter Fields

THE 2-MINUTE SUMMARY

A language model works with tokens, small pieces of text represented as numbers. During pre-training it learns to predict what comes next by adjusting its weights, the numbers that govern its predictions.

Post-training helps turn a text predictor into an assistant. It can teach the model to follow instructions, favour certain kinds of answers and, in some models, practise solving problems whose answers can be checked.

A correction can change the next reply without changing the model's weights. And a fluent answer can still be wrong, even when you give the model the right source.

What happens when you ask about Paris?

Ask a chatbot for the capital of France and it will probably say Paris. It feels like a lookup. A chatbot with a search tool can do a lookup, but a language model can also produce the answer from patterns it acquired during training. What did it learn to do before you ever typed the question?

Step 1 · Turning text into numbers

A tokeniser breaks text into units called tokens. Depending on the tokeniser, a token might be a whole word, part of a word or a punctuation mark. A word such as "unhappy" might be one token or split into pieces. The tokeniser gives each piece a numerical ID.[1]

Inside the model, each ID has a starting representation called an embedding, a list of numbers learnt during training. As the model processes a passage, a mechanism called attention lets other tokens in the available text shape how a token is represented. Think of "capital" in "the capital of France" and "capital invested in France". The same word plays a different part in each phrase. The article on Memory, Attention and Limits explains how this use of context affects a chat.[2]

Step 2 · Learning to predict what comes next

During pre-training, the model sees many examples of how text continues. The material varies by model and can include books, web pages and other sources. If a training passage reads "The capital of France is Paris", the model sees the preceding text and assigns probabilities to possible next tokens. The actual continuation supplies the target. No one needs to label that example by hand. That is why this method is called self-supervised learning.[3]

A loss function scores the prediction. The training software uses the score to work out how to adjust the weights so the model gets better at the task. Repeated across batches of text, that process can teach patterns of grammar and style, as well as associations such as France and Paris. The target is what the passage actually says. A work of fiction supplies its own target just as an atlas does.[3]

Geek Box · Can a model memorise what it reads?

The finished weights capture patterns that can be used in new combinations. They can also encode particular passages closely enough for the model to reproduce them word for word. Studies found that repeated passages were more likely to be reproduced in the models tested. Confidentiality and Privacy, Part 1, looks at what this means for personal information.[4]

Step 3 · Teaching the model to answer

The resulting base model is trained to continue text. Give it a question about France and it might answer it, or carry on with another question as if it were completing a quiz. Post-training can teach it how to respond as an assistant. Different teams use different combinations of methods.[3,5]

- **Demonstrations.** Supervised fine-tuning gives the model examples of how to respond. A demonstration might pair the capital question with a short answer in the form requested.[5]

- **Preferences.** If one reply answers directly and another goes off on a tangent, a reviewer can prefer the first. Reinforcement learning from human feedback (RLHF) can turn rankings into a reward signal for training. Direct preference optimisation (DPO) uses preferred and less preferred replies to adjust the model more directly. Tone, Personality and Alignment looks at how feedback can shape an assistant's tone.[5,6]

- **Checkable problems.** Some reasoning models also practise on tasks where software can score the result, such as a maths problem with a known answer or code that must pass tests. Reasoning, Speed and Effort looks at what you experience when a model spends more time on a reply.[7]

What a conversation can change

If you correct an answer in a chat, the app can give the model that correction as context for its next reply. In ordinary use, this leaves the model's weights as they were. In a new chat, the app may bring the correction across through a memory or saved instruction, depending on the product and settings. A provider's later use of chat data for training is a separate process. Check your service and account settings.[3,8]

How to use it better

- **Give it a source you can check.** Ask it to point to the passage that supports its answer. A model can still misread a source or add a claim the text cannot support. Check the passage yourself. If the material is confidential, establish whether you may share it with that service before uploading it. The Confidentiality and Privacy articles explain what to check.[9]

- **Ask it to say when the evidence runs out.** Try "If the source doesn't answer the question, say so." It may still answer with misplaced confidence. Researchers have found signals that can sometimes distinguish likely correct answers from errors, but the tone of a reply gives you no dependable measure. The Hallucinations series takes up that gap.[10]

- **Verify what you will rely on.** If the date, figure or legal citation matters, check it against an authoritative source. A plausible paragraph can point you towards the right place to look. The check is still yours to make.

Sources and notes

1. Rico Sennrich, Barry Haddow and Alexandra Birch (2016), Neural Machine Translation of Rare Words with Subword Units, ACL, pp. 1715–1725, §3. Subword tokenisation. Paper

2. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser and Illia Polosukhin (2017), Attention Is All You Need, NeurIPS 30, §§3.2 and 3.4. The Transformer's attention mechanism and learnt embeddings. Paper

3. Tom B. Brown and colleagues (2020), Language Models are Few-Shot Learners, NeurIPS 33, pp. 1877–1901, §§2.1–2.4. GPT-3's training data, model training and responses to context without weight updates. Paper

4. Milad Nasr, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A. Feder Cooper, Daphne Ippolito, Christopher A. Choquette-Choo, Eric Wallace, Florian Tramèr and Katherine Lee (2023), Scalable Extraction of Training Data from (Production) Language Models. See also Nicholas Carlini and colleagues (2023), Quantifying Memorization Across Neural Language Models, ICLR, on repetition and memorisation. Nasr and colleagues · Carlini and colleagues

5. Long Ouyang and colleagues (2022), Training language models to follow instructions with human feedback, NeurIPS 35, pp. 27730–27744, §3. Demonstrations, ranked replies and reinforcement learning. Paper

6. Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning and Chelsea Finn (2023), Direct Preference Optimization: Your Language Model is Secretly a Reward Model, NeurIPS 36, pp. 53728–53741, §4. Paper

7. DeepSeek-AI (2025, revised 2026), DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning, §2.2. Accuracy rewards for maths answers and code tests. Paper

8. Murray Shanahan (2024), Talking About Large Language Models, Communications of the ACM 67(2), pp. 68–79, on the distinction between a language model and a dialogue system. OpenAI's documentation gives examples of app memory and use of consumer data for model improvement. Shanahan · Memory · Consumer data

9. Cheng Niu, Yuanhao Wu, Juno Zhu, Siliang Xu, KaShun Shum, Randy Zhong, Juntong Song and Tong Zhang (2024), RAGTruth: A Hallucination Corpus for Developing Trustworthy Retrieval-Augmented Language Models, ACL, pp. 10862–10878. Unsupported or contradictory answers even with source passages. Paper

10. Saurav Kadavath and colleagues (2022), Language Models (Mostly) Know What They Know, arXiv:2207.05221. Experiments on models' estimates of whether they can answer correctly, including limits on new tasks. Paper