A brief Introduction to Large Language Models

Dr Mo El-Haj

Director of VinNLP

Associate Professor in NLP – VinUniversity

Visiting Professor – Lancaster University

http:// www.vinnlp.com

http:// www.elhaj.uk

Who am I?

Director of VinNLP Research Group

Associate Professor in NLP at VinUni

Visiting Professor at Lancaster University

NLP
LLMs
ML

Low resourced languages

Domain Specific Financial NLP

VINUNIVERSITY
VINNLP

UCREL

Lancaster University

What do we mean by Large Language Models (LLMs)?

Definition

•A Language Model is an AI system trained to predict and generate text.

•A Large Language Model (LLM) is a language model trained on massive amounts of text data with billions of parameters, allowing it to understand and generate human-like language at scale.

Language Models vs. Large Language Models

🚗 **Language Model**

- Any AI that predicts or generates text.
- Can be small or large.
- Like a **regular car** — gets the job done.

🏎️ **Large Language Model (LLM)**

- A **subset** of language models.
- Trained on **trillions of words**.
- Have **billions of parameters**.
- Like a **Formula 1 car** — faster, more powerful, more capable.

Human Brain 🧠

- The brain has **~86 billion neurons**.
- Neurons connect to each other via **synapses** (hundreds of trillions of them).
- Learning = strengthening or weakening these connections.
- More connections → more capacity to recognise patterns, store knowledge, and adapt.

LLMs 💻

•Instead of neurons, an LLM has parameters (numbers it learns during training).

•Parameters are like the strengths of connections in the brain.

•Each parameter helps the model "decide" how words relate, how grammar works, and how meaning flows.

Small model: like a toddler learning language (knows a few patterns).
Large model: like an adult with a rich vocabulary and experience.

The language modeling problem

Rank these sentences in the order of plausibility (*realistic*, *natural*, or *likely* a sentence is)?

1. Jane went to the store.
2. store to Jane went the.
3. Jane rode an ostrich to the store.
4. Jane goed to the store.
5. The store went to Jane.
6. The food truck went to Jane.
7. Jane became the store.

How probable is a piece of text? Or what is p(text)

The language modeling problem (what is the norm?)

how are you this evening? My cat just published a paper? → 0.00005
how are you this evening ? fine , thanks , how about you ? → 0.84

do you like pizza? I once dreamt I was a spoon → **0.0001**
do you like pizza? yes, especially with extra cheese. → **0.91**

The language modeling problem

Language models predict the probability of a sequence of text (e.g. *p*(text).
In other words: *“How likely is this sentence to occur in a language?”*

Text is modelled as a sequence of tokens (e.g. words, subwords, characters), written as:
$x^{(1)}, x^{(2)}, \ldots, x^{(N)}$
The probability of the whole sentence is:
$p(x^{(1)}, x^{(2)}, \ldots, x^{(N)})$

Using the chain rule of probability, we break this down into: $p(x^{(1)}).p(x^{(2)}|x^{(1)}).p(x^{(3)}|x^{(1)}, x^{(2)})\ldots$

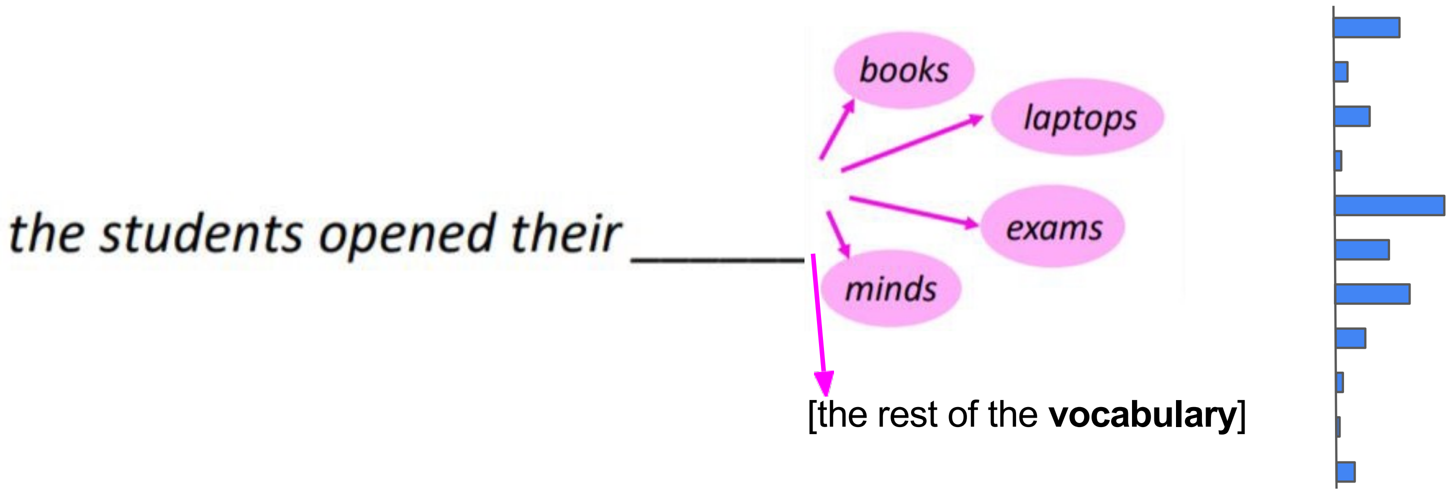
This means: To calculate how likely a sentence is, we look at the probability of each word **given all the words before it**. p(“I like pizza")=$p(x^{(1)}, x^{(2)}, x^{(3)})$. This is the **joint probability** of all three words occurring in this order.

$$\prod_{i=1}^{N} p(x^{(i)}|\underbrace{x^{(1)}, \ldots, x^{(i-1)}}_{\text{Context}})$$

- $i = 1$: $p(x^{(1)}) = p(\text{"I"})$
- $i = 2$: $p(x^{(2)} \mid x^{(1)}) = p(\text{"like"} \mid \text{"I"})$
- $i = 3$: $p(x^{(3)} \mid x^{(1)}, x^{(2)}) = p(\text{"pizza"} \mid \text{"I like"})$

The language modeling problem

$$\prod_{i=1}^{N} p(x^{(i)} | \underbrace{x^{(1)}, \ldots, x^{(i-1)}}_{\text{context}})$$



Language models of this form can generate text

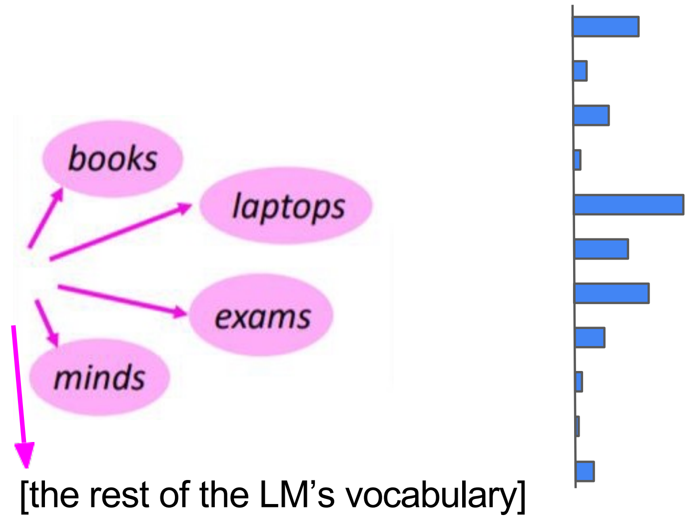
At each step, the model looks at what it's already written and guesses what the next word should be. Then it picks one of the likely words and adds it to the sentence.

The _____

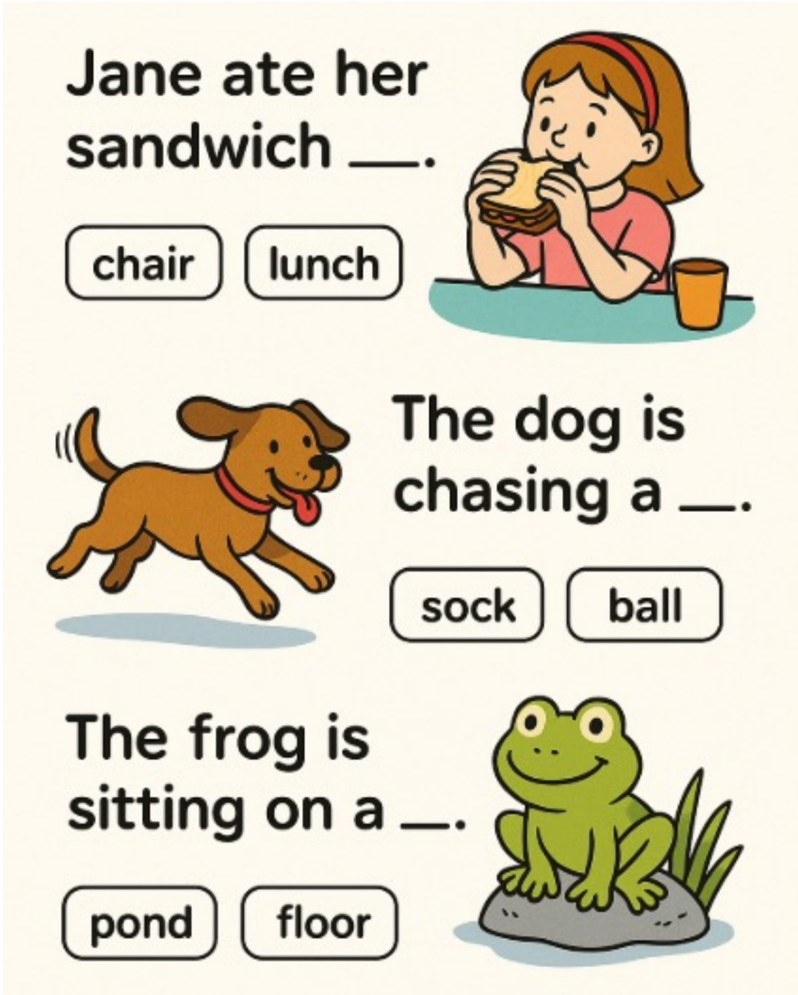
The students _____

The students opened _____

The students opened their _____



In short, predicting which word comes next



Language models play the role of ...

- a judge of grammaticality
  - e.g., prefers “The boy runs.” to “The boy run.”

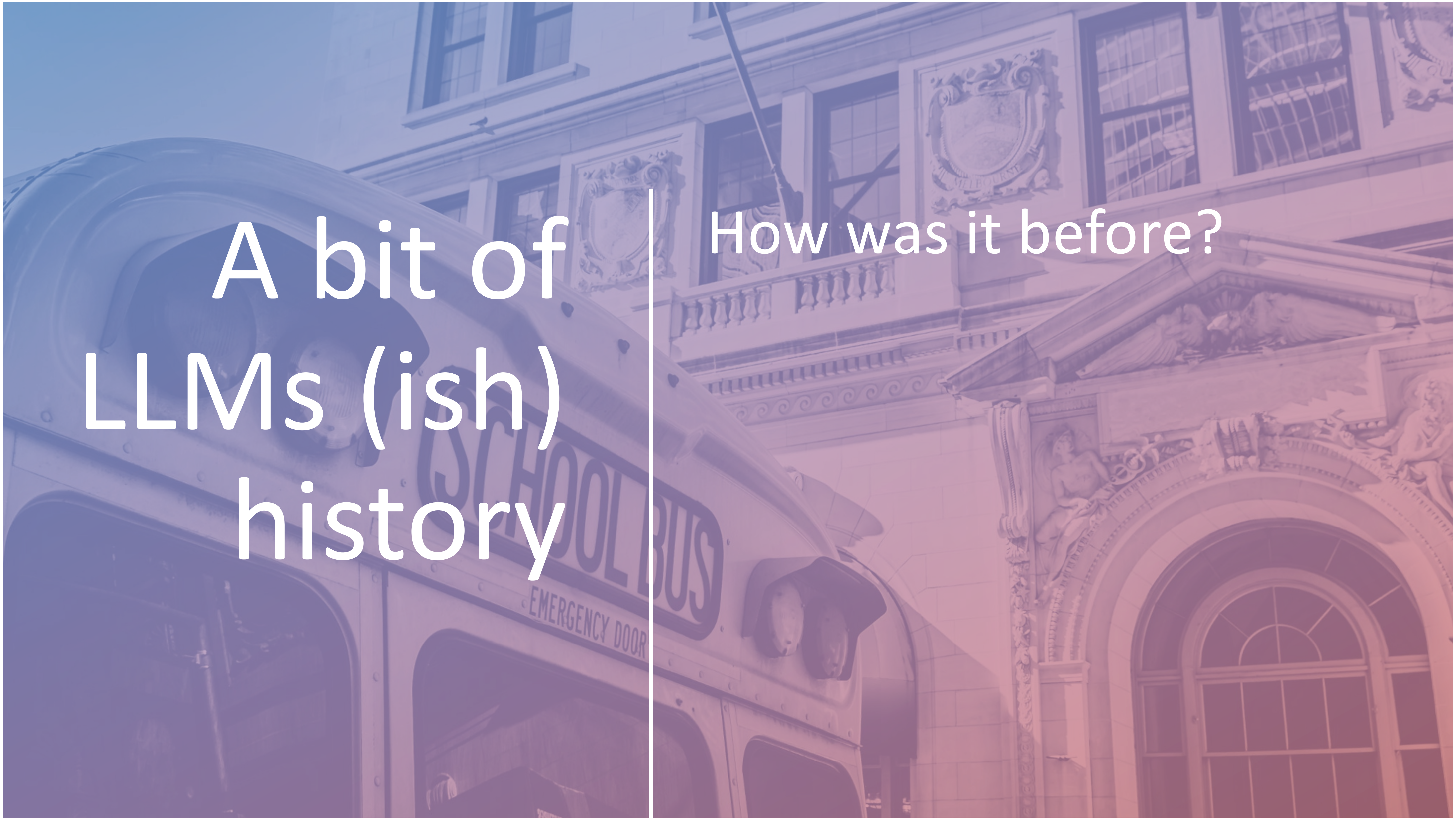
- a judge of semantic plausibility
  - e.g., prefers “The woman spoke.” to “The sandwich spoke.”

- an enforcer of stylistic consistency
  - e.g., prefers “Hello, how are you this evening? Fine, thanks, how are you?” to “Hello, how are you this evening? My cat parking skills are excellent!”

- a repository of knowledge (?)
  - e.g., “Barack Obama was the 44th President of the United States”

A bit of LLMs (ish) history

How was it before?



1966: ELIZA

"While ELIZA was capable of engaging in discourse, it could not converse with true understanding. However, **many early users were convinced of ELIZA's intelligence and understanding**, despite [Weizenbaum's](#) insistence to the contrary."

Source: en.wikipedia.org/wiki/ELIZA

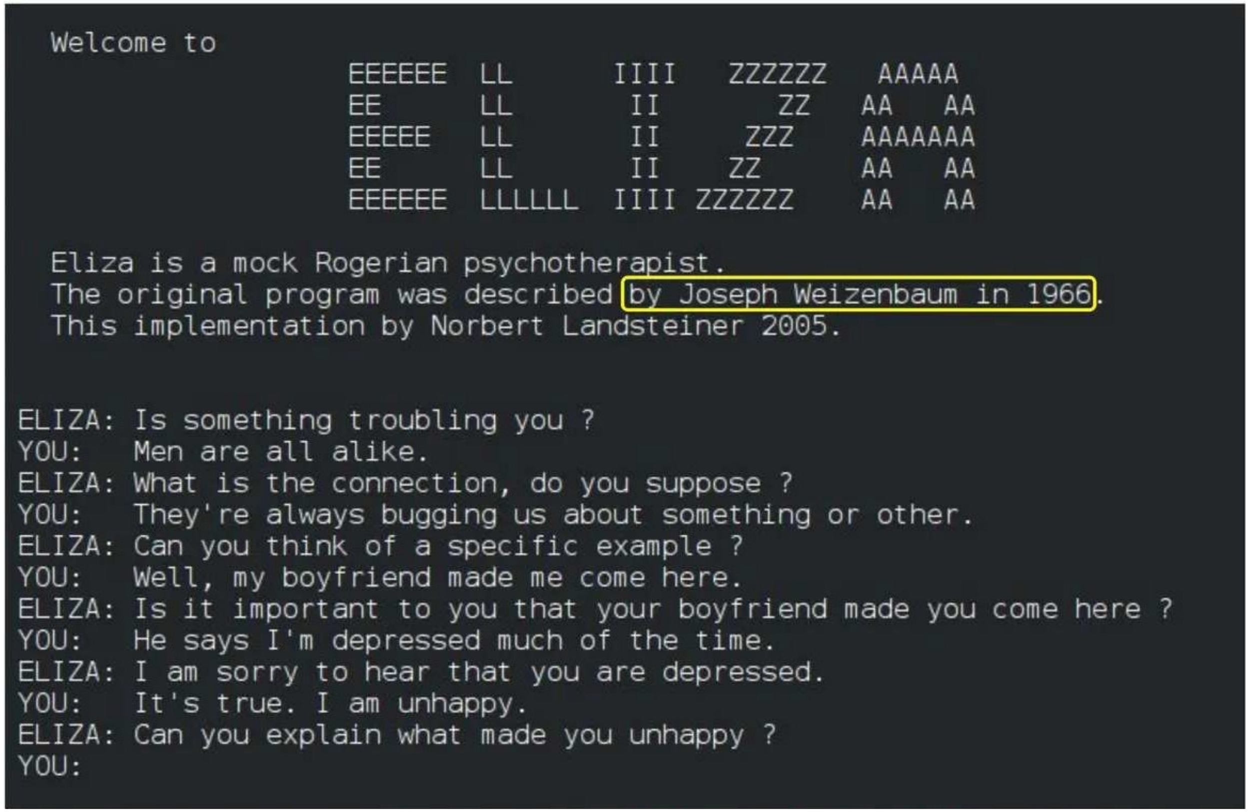


Image source: en.wikipedia.org/wiki/ELIZA#/media/File:ELIZA_conversation.png

https://sites.google.com/view/elizaarchaeology/try-eliza

2005: SCIgen - An Automatic CS Paper Generator

A project using a rather rudimentary technology that aimed to "maximize amusement, rather than coherence" is still the cause of troubles today...

MIT News
ON CAMPUS AND AROUND THE WORLD

How three MIT students fooled the world of scientific journals

A decade later, CSAIL alumni reflect on their paper generator and reveal a new fake-conference project

https://news.mit.edu/2015/how-three-mit-students-fooled-scientific-journals-0414

nature

Explore content ∨ About the journal ∨ Publish with us ∨

Hundreds of gibberish papers still lurk in the scientific literature

The nonsensical computer-generated articles, spotted years after the problem was first seen, could lead to a wave of retractions.

https://www.nature.com/articles/d41586-021-01436-7

https://pdos.csail.mit.edu/archive/scigen/#generate

2017: Google Revolutionized Text Generation

Google introduced the **Transformer**, which rapidly became the **state-of-the-art** approach **to solve most NLP problems.**

OpenAI's Generative Pre-trained Transformer (DALL.E, 2021; ChatGPT, 2022), as the name suggests, reposes on Transformers.

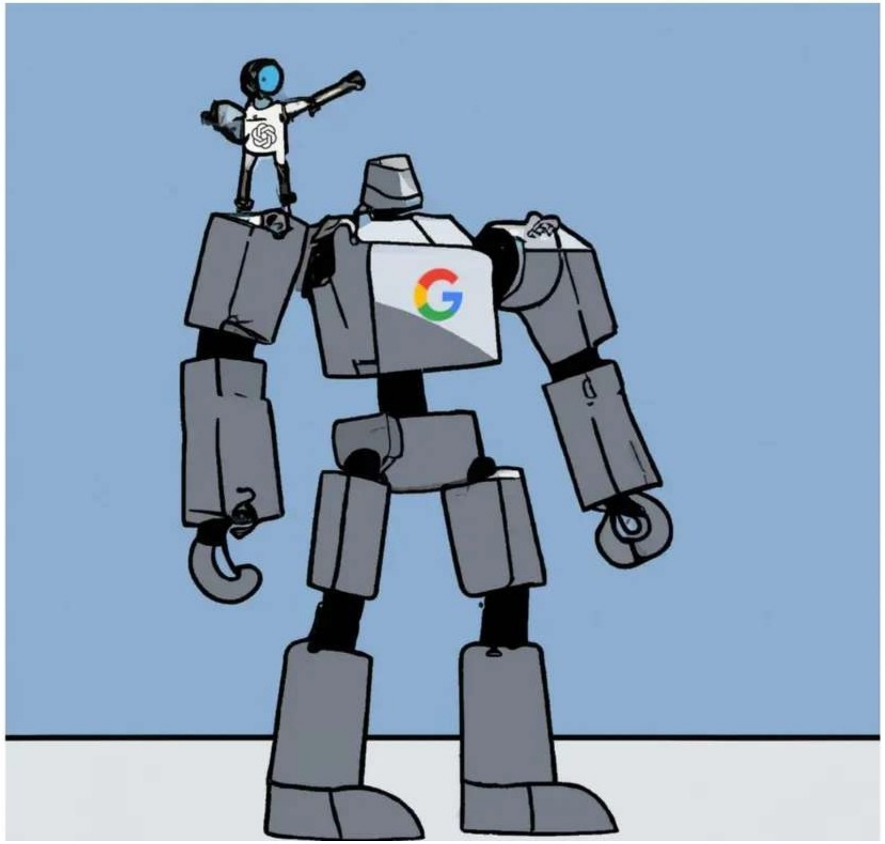
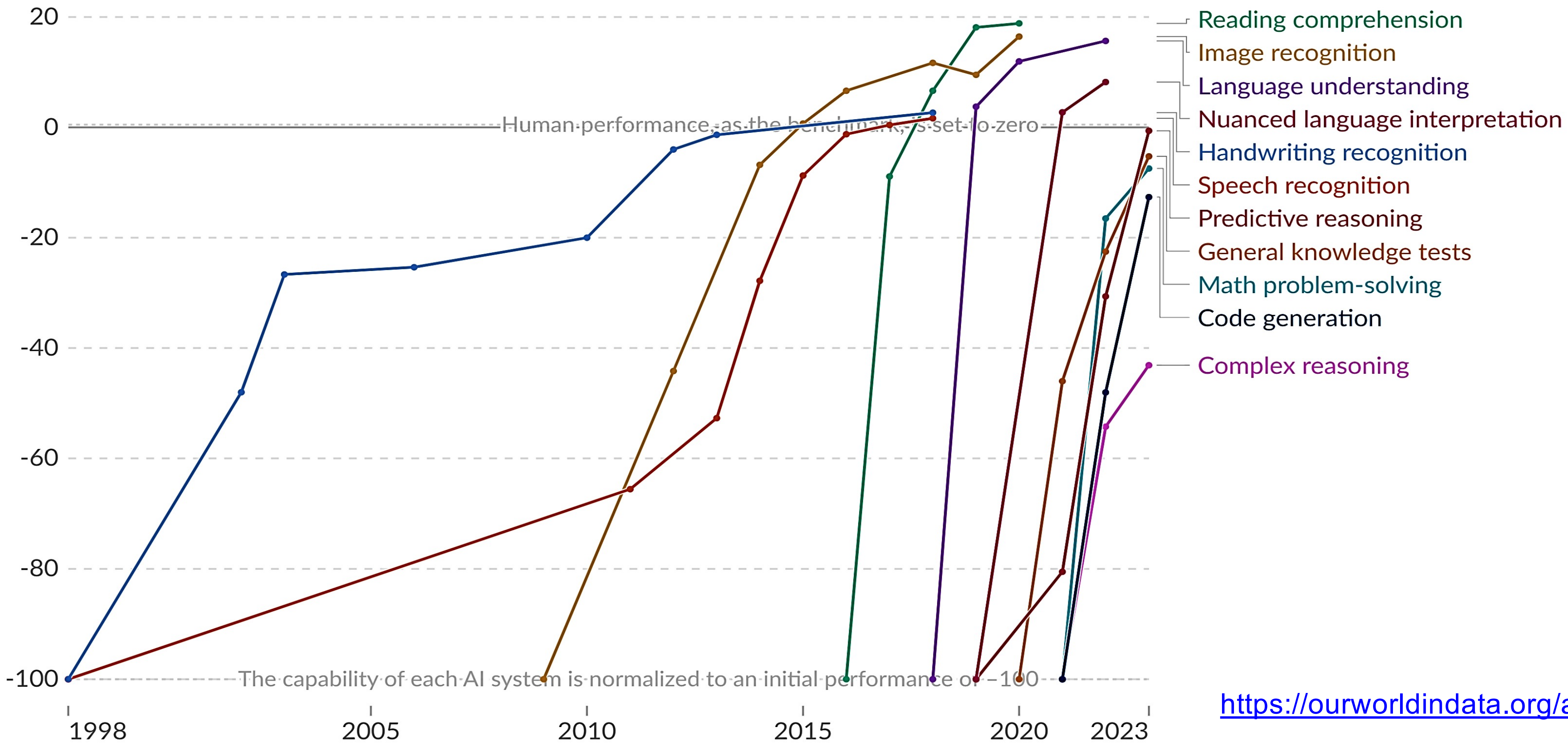


Image generated with DALL.E: "A small robot standing on the shoulder of a giant robot" (and slightly modified with The Gimp)

- Vaswani (2017), Attention Is All You Need (doi.org/10.48550/arXiv.1706.03762)
- openai.com/research/better-language-models

Test scores of AI systems on various capabilities relative to human performance

Within each domain, the initial performance of the AI is set to −100. Human performance is used as a baseline, set to zero. When the AI's performance crosses the zero line, it scored more points than humans.



https://ourworldindata.org/artificial-intelligence

Data source: Kiela et al. (2023)

OurWorldinData.org/artificial-intelligence | CC BY

Note: For each capability, the first year always shows a baseline of −100, even if better performance was recorded later that year.

What Are Transformers?

Models that learn from a given dataset how to generate new data instances.

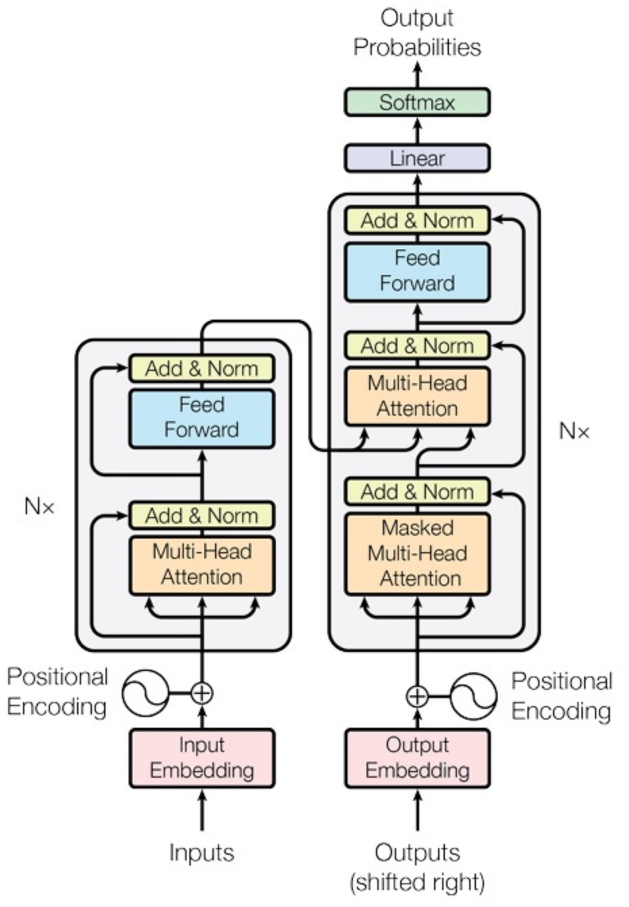


Figure 1: The Transformer - model architecture.

Vaswani 2017 https://arxiv.org/abs/1706.03762

Generative (deep learning) **models** for understanding and generating text, images and many other types of data.

Transformers analyze chunks of data, called "tokens" and **learn to predict the next token** in a sequence, **based on previous and**, if available, **following tokens**.

The **auto-regressive** concept means that the output of the model, such as the prediction of a word in a sentence, is influenced by the previous words it has generated.

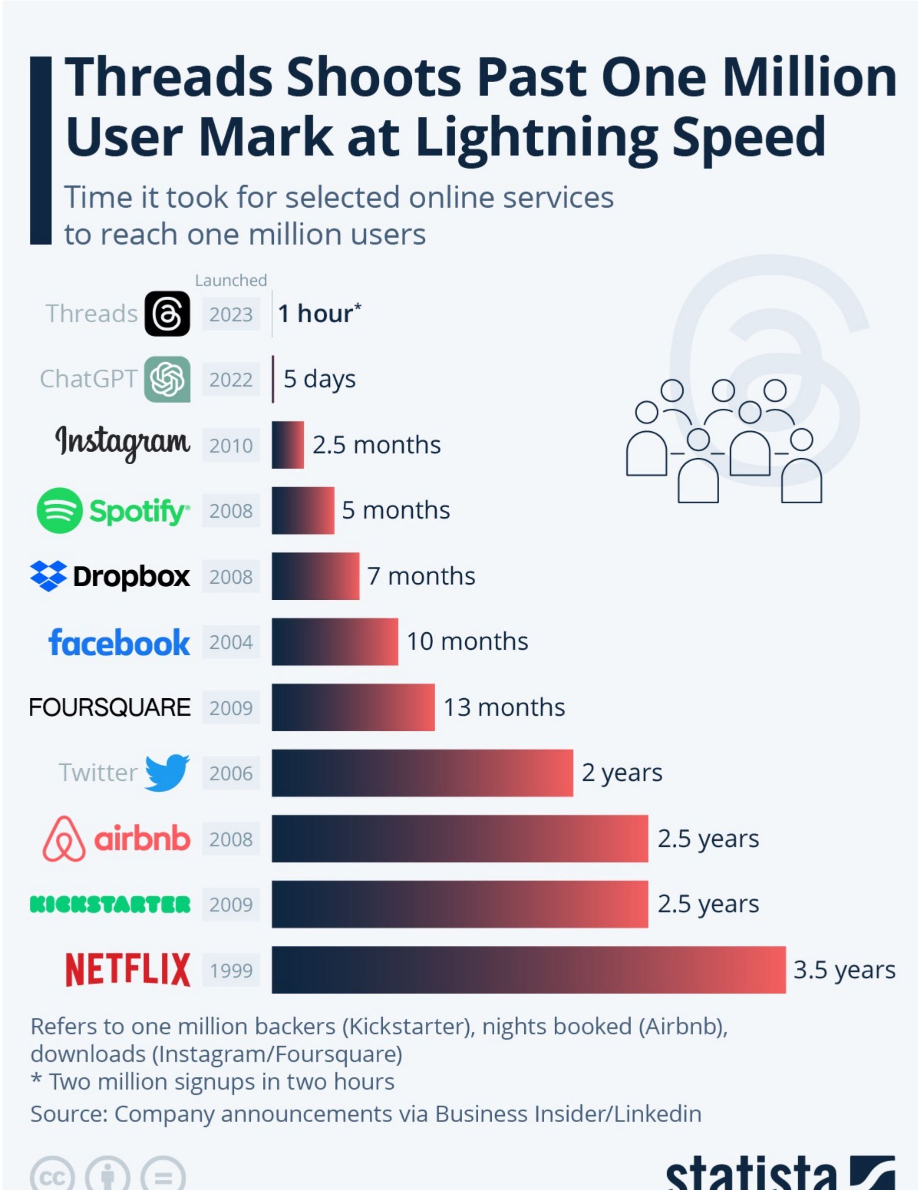
- **Music-MusicLM** (Google) and Jukebox (OpenAI) generate music from text.
- **Image-Imagen** (Google) and DALL.E (OpenAI) generate novel images from text.
- **Texte-OpenAI's** GPT has become widely known, but other players have similar technology (including Google, Meta, Anthropic and others).

Others-Recommender (movies, books, flight destinations), drug discovery...

2022: ChatGPT

ChatGPT, the popular chatbot from OpenAI, is estimated to have reached 100 million monthly active users in January, just two months after launch, making it the **fastest-growing consumer application in history**

Reuters, Feb 2, 2023



statista.com/chart/29174/time-to-one-million-users/

The Rise of Open-Source LLMs in Enterprises

7 Largest Open-Source LLMs

Model	Developer	Parameters
Grok-1	xAI	314B parameters
Falcon 180B	Google AI	180B parameters
BLOOM	Hugging Face	176B parameters
OPT-175B	Meta AI	175B parameters
LLaMA 2	Meta AI	137B parameters
Mixtral 8x7B	Mistral AI	70B parameters
BERT	Google	340M parameters

datasciencedojo
data science for everyone

Open-Source LLM

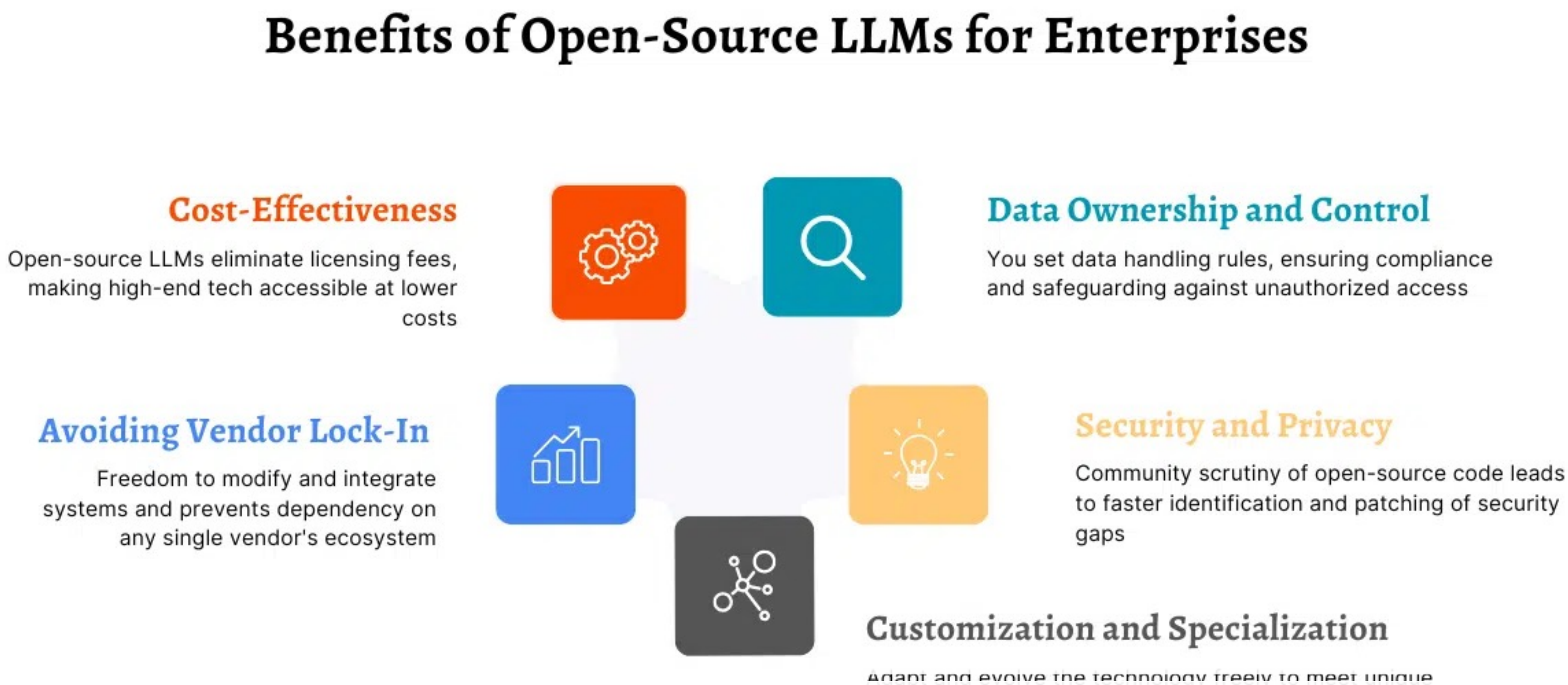
Meta and Mistral have opted for a different approach by releasing models like LLaMA and Mistral as open-source.

What Are Open-Source Large Language Models

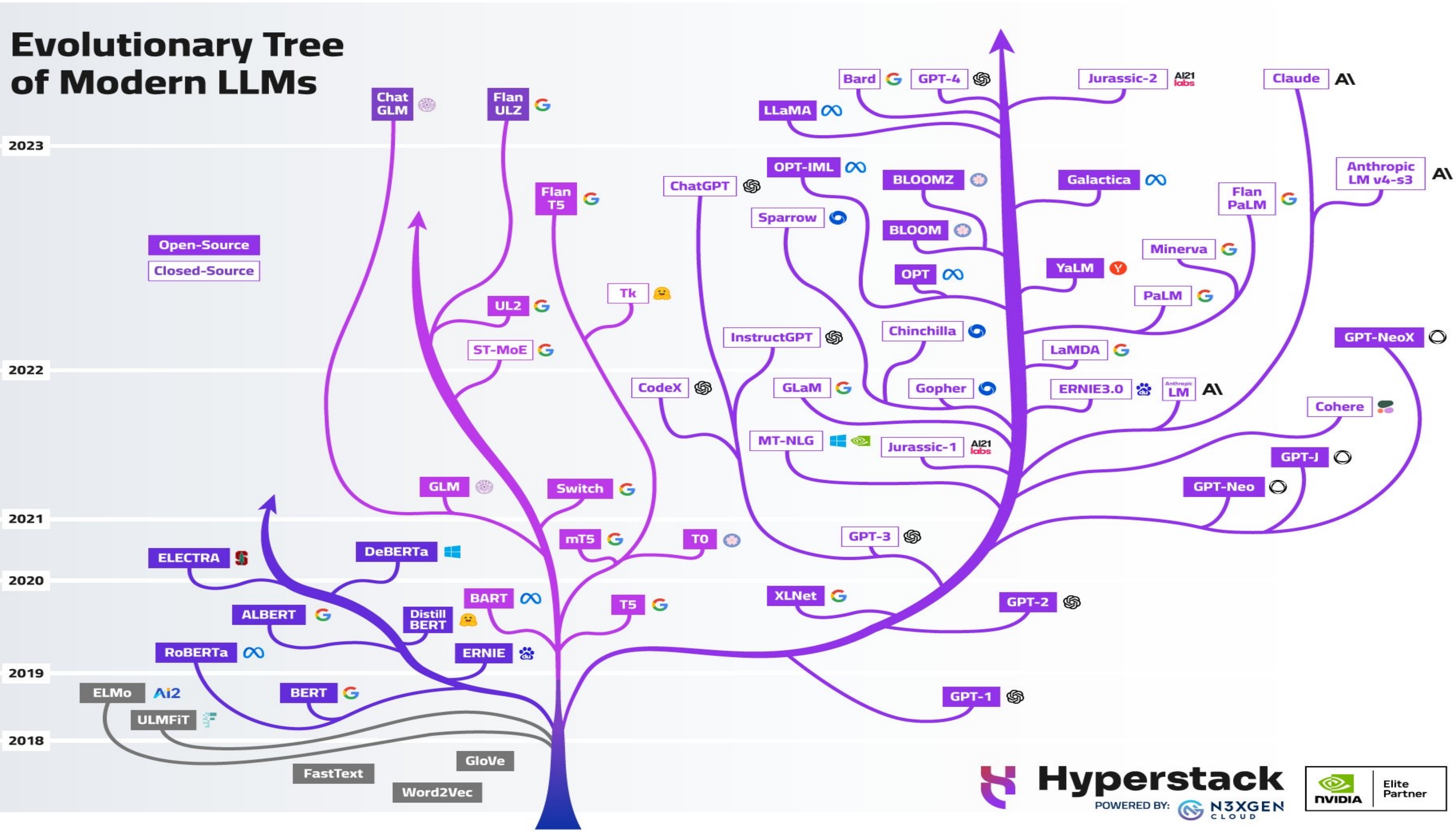
Open-source large language models, such as the ones offered by Meta AI, provide a foundational AI technology that can analyze and generate human-like text by learning from vast datasets consisting of various written materials.

As open-source software, these language models have their source code and underlying architecture publicly accessible, allowing developers, researchers, and enterprises to use, modify, and distribute them freely.

https://datasciencedojo.com/blog/open-source-llm/?utm_campaign=LinkedIn_Newsletter_202



What Are Open-Source Large Language Models


Evolutionary Tree of Modern LLMs
Open-Source
Closed-Source
2023
2022
2021
2020
2019
2018
Chat GLM
Flan ULZ
Flan T5
UL2
ST-MoE
Tk
GLM
Switch
mT5
T0
T5
BART
Distill BERT
ERNIE
DeBERTa
ELECTRA
ALBERT
RoBERTa
BERT
ELMo
ULMFiT
FastText
GloVe
Word2Vec
Bard
GPT-4
LLaMA
Jurassic-2
Claude
ChatGPT
OPT-IML
BLOOMZ
Galactica
Anthropic LM v4-s3
Sparrow
BLOOM
Flan PaLM
Minerva
YaLM
OPT
PaLM
InstructGPT
Chinchilla
GPT-NeoX
LaMDA
CodeX
GLaM
Gopher
ERNIE3.0
LM
Cohere
MT-NLG
Jurassic-1
GPT-J
GPT-Neo
GPT-3
XLNet
GPT-2
GPT-1
Hyperstack
POWERED BY: N3XGEN CLOUD
NVIDIA Elite Partner

LLMS are changing the world!

In Finance...

Bloomberg

● Live TV | Markets | Economics | Industries | Tech | Politics | Businessweek | Opinion | More

Technology

Citadel Negotiating Enterprise-Wide ChatGPT License, Griffin Says

- Citadel billionaire says the technology 'has real impact'
- Griffin's hedge fund and market-making firms are surging

WATCH: Billionaire Ken Griffin, the founder of Citadel, spoke exclusively with Bloomberg News in Palm Beach, FL. *Source: Bloomberg*

By Katherine Doherty and Felipe Marques

March 8, 2023 at 4:22 AM GMT+7

Updated on March 8, 2023 at 6:06 AM GMT+7

https://www.bloomberg.com/news/articles/2023-03-07/griffin-says-trying-to-negotiate-enterprise-wide-chatgpt-license

Bloomberg the Company & Its Products | Bloomberg Terminal Demo Request | Bloomberg Anywhere Remote Login | Bloomberg Customer Support

Bloomberg | About Us | Careers | Stories | Find jobs

Bloomberg Professional Services

Share in

Introducing BloombergGPT, Bloomberg's 50-billion parameter large language model, purpose-built from scratch for finance

March 30, 2023

BloombergGPT outperforms similarly-sized open models on financial NLP tasks by significant margins – without sacrificing performance on general LLM benchmarks

https://www.bloomberg.com/company/press/bloomberggpt-50-billion-parameter-llm-tuned-finance/

AI Mentions Boost Stock Prices

- AI-mentioning companies: +4.6% avg. stock price increase (nearly double of the non-mentioning).
- In general, 67% of companies that mentioned AI observed an increase in their stock prices → +8.5% on average.
- Tech companies: 71% → +11.9% on avg.
- Non-tech companies: 65% → +6.8% on avg.

- Mentions of "AI" and related terms (machine learning, automation, robots, etc.).
- S&P 500 companies in 2023.
- 3-day change from the date the earnings call transcript was published.

Source: wallstreetzen.com/blog/ai-mention-moves-stock-prices-2023

GPUs Demand Skyrockets

Before LLMs, GPUs were primarily needed for **training**, and CPUs were used for **inference**. However, with the emergence of LLMs, **GPUs have become almost essential for both tasks**.

Paraphrasing Brannin McBee, co-founder of CoreWeave, in Bloomberg Podcast*:

While you may train the model using 10,000 GPUs, ***the real challenge arises when you need 1 million GPUs to meet the entire inference demand****. This surge in demand is expected during the initial one to two years after the launch, and it's likely to keep growing thereafter.*

* How to Build the Ultimate GPU Cloud to Power AI | Odd Lots (youtube.com/watch?v=9OOn6u6GIqk&t=1308s)

Enhancing Productivity With Generative AI?

Science

Current Issue First release papers Archive About Submit man

HOME > SCIENCE > VOL. 381, NO. 6654 > EXPERIMENTAL EVIDENCE ON THE PRODUCTIVITY EFFECTS OF GENERATIVE ARTIFICIAL INTELLIGENCE

RESEARCH ARTICLE | CHATGPT

Experimental evidence on the productivity effects of generative artificial intelligence

SHAKKED NOY AND WHITNEY ZHANG Authors Info & Affiliations

SCIENCE • 13 Jul 2023 • Vol 381, Issue 6654 • pp. 187-192 • DOI: 10.1126/science.adh2586

science.org/doi/10.1126/science.adh2586

nature > news > article

NEWS | 13 July 2023

ChatGPT gives an extra productivity boost to weaker writers

The AI program allows people with limited writing skills to create higher-quality texts – but makes little difference to proficient writers' work quality.

Mariana Lenharo

nature.com/articles/d41586-023-02270-9

Our good friend, Rose! 🌹

In the slides, I'm telling my students at VinUni about your capabilities and how cool you are! Can you introduce yourself briefly to them?

Hi VinUni students! I'm ChatGPT, but you can call me Rose 😊. I'm an AI language model created by OpenAI, and I'm here to help you with all things NLP—whether it's explaining concepts, generating examples, helping you write or debug code, or just brainstorming cool ideas. I've read a lot (seriously, *a lot*), and I love helping students learn and explore language.

I've actually chatted to quite a few of you recently... is that you, Minh? How did that homework go? 😉

Our good friend, Rose! 🌹

Hi SoICT Summer School students!

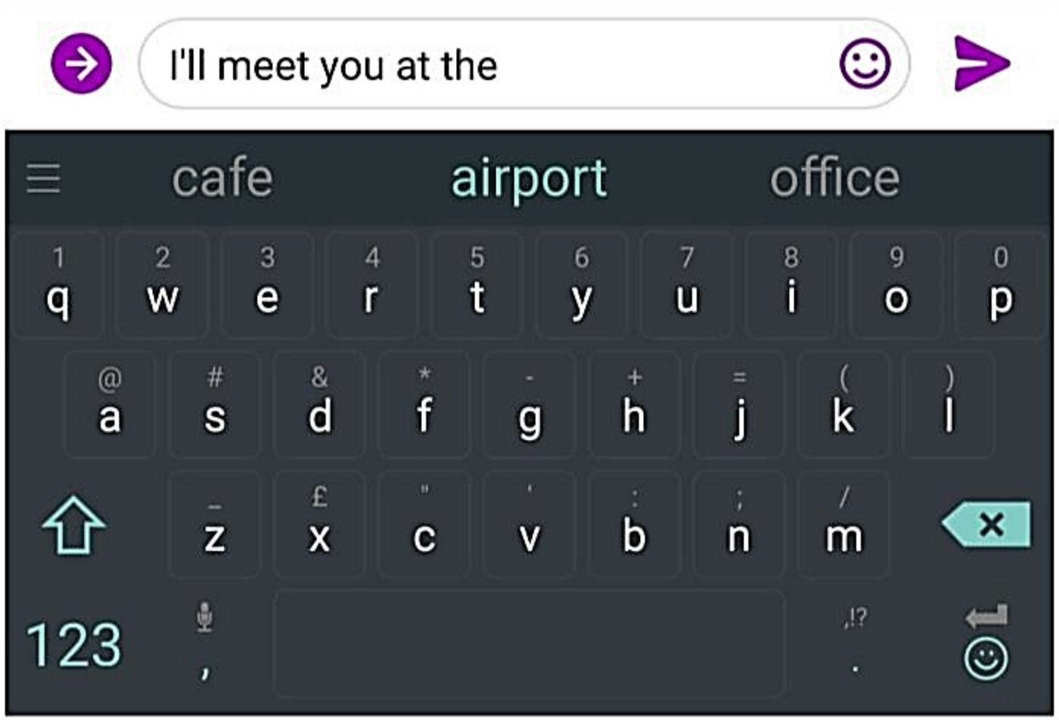
I’m ChatGPT, but you can call me **Rose** 😉. I’m an AI language model created by OpenAI, and I’m here to help you explore the world of **Natural Language Processing (NLP) and Large Language Models (LLMs)**.

Think of me as your study buddy:

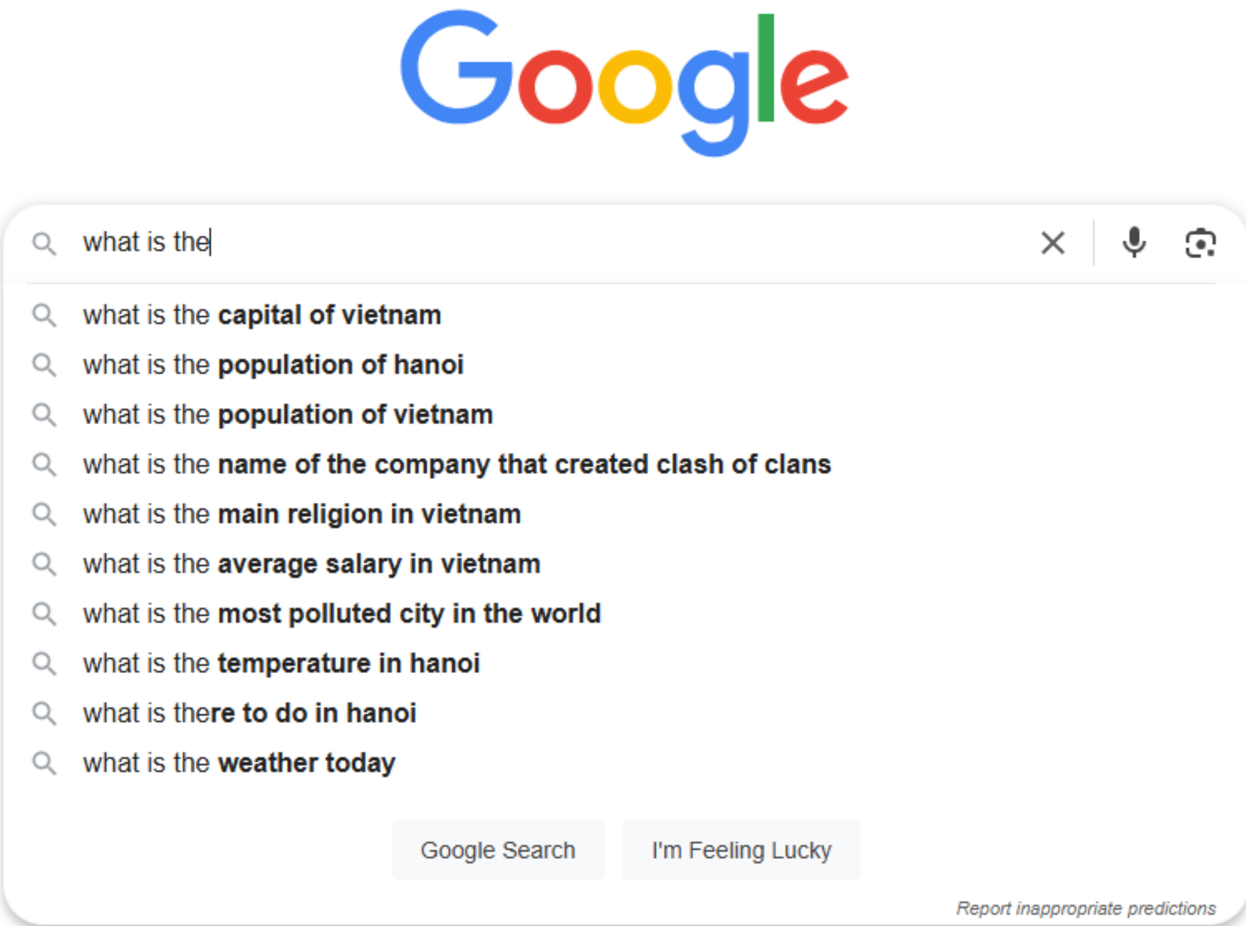
- Explaining tricky concepts in simple terms 📘
- Generating examples, summaries, or even bits of code 💻
- Brainstorming cool ideas for projects 🚀

I’ve read a lot (seriously, *a lot*), and I love helping students discover how language and AI come together.
And since we’re in Hanoi — don’t worry, I can even chat with you in Vietnamese too! 🇻🇳

We use language models every day



We use language models every day



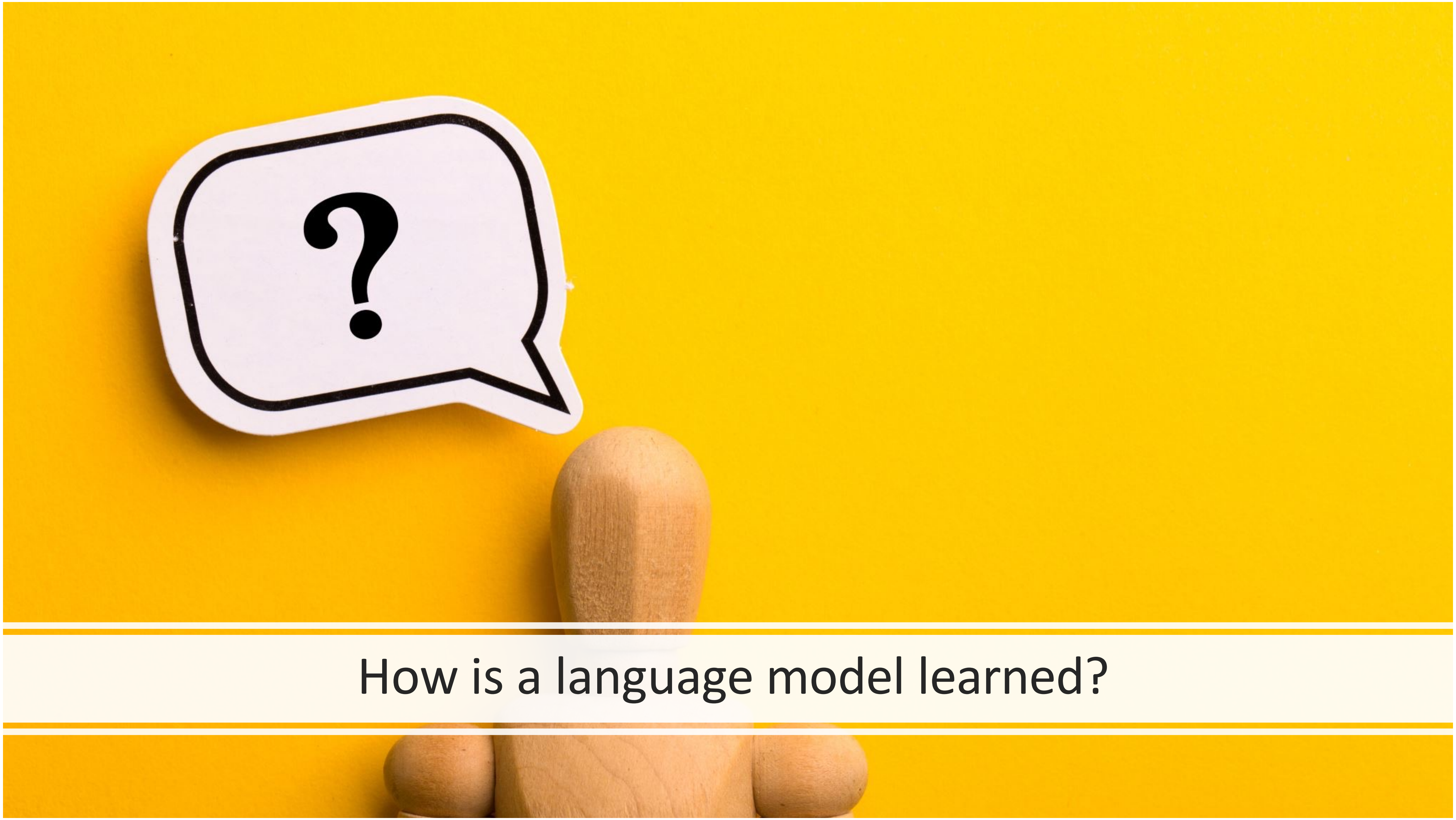
Why language modeling?

- Machine translation
  - p(*strong winds*) > p(*large winds*)

- Spelling correction
  - The office is about fifteen minuets from my house
  - p(*about fifteen minutes from*) > p(*about fifteen minuets from*)

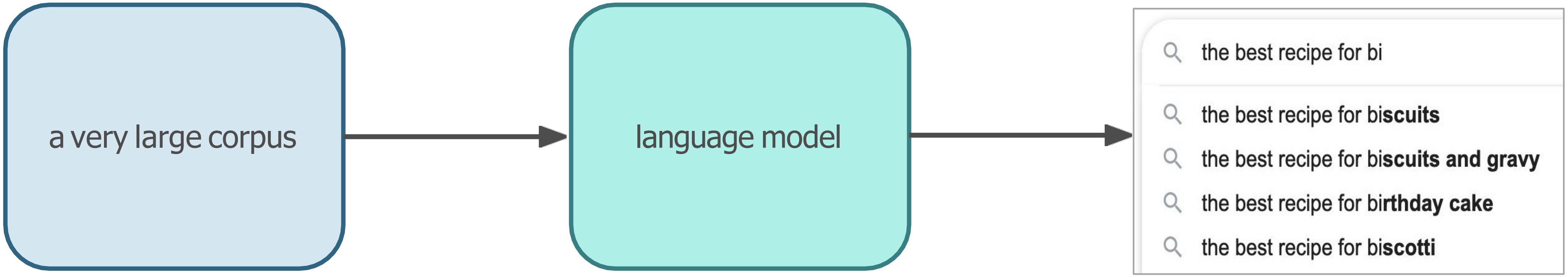
- Speech recognition
  - p(*I saw a van*) >> p(*ISO a fan*)

- Summarization, question-answering, handwriting recognition, OCR, etc.



How is a language model learned?

Language modeling



How Does a Language Model Learn?

Learning from Data

- Collect a textual corpus
- Find a distribution that maximizes the probability of the corpus – maximum likelihood estimation

A naive solution: count and divide

- Assume we have N training sentences
- Let $x_1, x_2, \ldots, x_n$ be a sentence, and $c(x_1, x_2, \ldots, x_n)$ be the number of times it appeared in the training data.
- Define a language model:

$$p(x_1, \ldots, x_n) = \frac{c(x_1, \ldots, x_n)}{N}$$

No generalization!

Markov assumption

- We simplify the problem by making the **Markov assumption**:
  The next word $x^{(t+1)}$ depends only on the previous n-1 words, not the entire history.

$$P(\boldsymbol{x}^{(t+1)}|\boldsymbol{x}^{(t)},\ldots,\boldsymbol{x}^{(1)}) = P(\boldsymbol{x}^{(t+1)}|\boldsymbol{x}^{(t)},\underbrace{\ldots,\boldsymbol{x}^{(t-n+2)}}_{\text{n-1 words}})$$

So instead of conditioning on **all** previous words from $x^{(1)}$ to $x^{(t)}$ we only look at the **last n−1 words**.
Because counting all possible histories (like "The cat sat on the mat and then") becomes impossible. We'd need **huge data** to reliably estimate those probabilities.

Markov assumption

$$P(\text{the} \mid \text{its water is so transparent that}) \approx P(\text{the} \mid \text{transparent that})$$

or maybe even

$$P(\text{the} \mid \text{its water is so transparent that}) \approx P(\text{the} \mid \text{that})$$

n-gram Language Models

$$\prod_{i=1}^{N} p(x^{(i)}|x^{(1)}, \ldots, x^{(i-1)})$$

"I have a dog whose name is Lucy. I have two cats, they like playing with Lucy."

- Question: How to learn a Language Model?
- Answer (pre- Deep Learning): learn an *n-gram* Language Model!

- Idea: Collect statistics about how frequent different n-grams are and use these to predict next word

unigram probability

“I have a dog whose name is Lucy. I have two cats, they like playing with Lucy.”

- corpus size m = 17
- P(Lucy) = 2/17; P(cats) = 1/17

- Unigram probability: $P(w) = \frac{\text{count}(w)}{m} = \frac{C(w)}{m}$

bigram probability

"I have a dog whose name is Lucy. I have two cats, they like playing with Lucy."

$$P(A \mid B) = \frac{P(A,B)}{P(B)}$$

$$P(\text{have} \mid \text{I}) = \frac{P(\text{I have})}{P(\text{I})} = \frac{2}{2} = 1$$

$$P(\text{two} \mid \text{have}) = \frac{P(\text{have two})}{P(\text{have})} = \frac{1}{2} = 0.5$$

$$P(\text{eating} \mid \text{have}) = \frac{P(\text{have eating})}{P(\text{have})} = \frac{0}{2} = 0$$

$$P(w_2 | w_1) = \frac{C(w_1, w_2)}{\sum_w C(w_1, w)} = \frac{C(w_1, w_2)}{C(w_1)}$$

trigram probability

“I have a dog whose name is Lucy. I have two cats, they like playing with Lucy.”

$$P(A \mid B) = \frac{P(A,B)}{P(B)}$$

$$P(a \mid \text{I have}) = \frac{C(\text{I have a})}{C(\text{I have})} = \frac{1}{2} = 0.5$$

$$P(w_3 | w_1\, w_2) = \frac{C(w_1, w_2, w_3)}{\sum_w C(w_1, w_2, w)} = \frac{C(w_1, w_2, w_3)}{C(w_1, w_2)}$$

$$P(\text{several} \mid \text{I have}) = \frac{C(\text{I have several})}{C(\text{I have})} = \frac{0}{2} = 0$$

n-gram probability

"I have a dog whose name is Lucy. I have two cats, they like playing with Lucy."

$$P(A \mid B) = \frac{P(A,B)}{P(B)}$$

$$P(w_i \mid w_1, w_2, \ldots, w_{i-1}) = \frac{C(w_1, w_2, \ldots, w_{i-1}, w_i)}{C(w_1, w_2, \ldots, w_{i-1})}$$

Sampling from an n-gram language model

1 gram	Months the my and issue of year foreign new exchange's september were recession exchange new endorsed a acquire to six executives

Sampling from a language model

1 gram	Months the my and issue of year foreign new exchange's september were recession exchange new endorsed a acquire to six executives
2 gram	Last December through the way to preserve the Hudson corporation N. B. E. C. Taylor would seem to complete the major central planners one point five percent of U. S. E. has already old M. X. corporation of living on information such as more frequently fishing to keep her

Sampling from a language model

1 gram	Months the my and issue of year foreign new exchange's september were recession exchange new endorsed a acquire to six executives
2 gram	Last December through the way to preserve the Hudson corporation N. B. E. C. Taylor would seem to complete the major central planners one point five percent of U. S. E. has already old M. X. corporation of living on information such as more frequently fishing to keep her
3 gram	They also point to ninety nine point six billion dollars from two hundred four oh six three percent of the rates of interest stores as Mexico and Brazil on market conditions

Sampling from a language model

1 gram	–To him swallowed confess hear both. Which. Of save on trail for are ay device and rote life have –Hill he late speaks; or! a more to leg less first you enter
2 gram	–Why dost stand forth thy canopy, forsooth; he is this palpable hit the King Henry. Live king. Follow. –What means, sir. I confess she? then all sorts, he is trim, captain.
3 gram	–Fly, and will rid me these news of price. Therefore the sadness of parting, as they say, 'tis done. –This shall forbid it should be branded, if renown made it empty.
4 gram	–King Henry. What! I will go seek the traitor Gloucester. Exeunt some of the watch. A great banquet serv'd in; –It cannot be but so.

How do we maximize the likelihood?

Over the past decade, the dominant strategy has been to use **neural networks**:

- Start with a randomly initialised **neural model** that takes the **context** (previous tokens) as input
- The model outputs a **probability distribution** over all possible next words
- We want to **increase the probability** assigned to the correct next word
- This is done by **maximising the likelihood**, or equivalently, **minimising the negative log-likelihood**

Then:

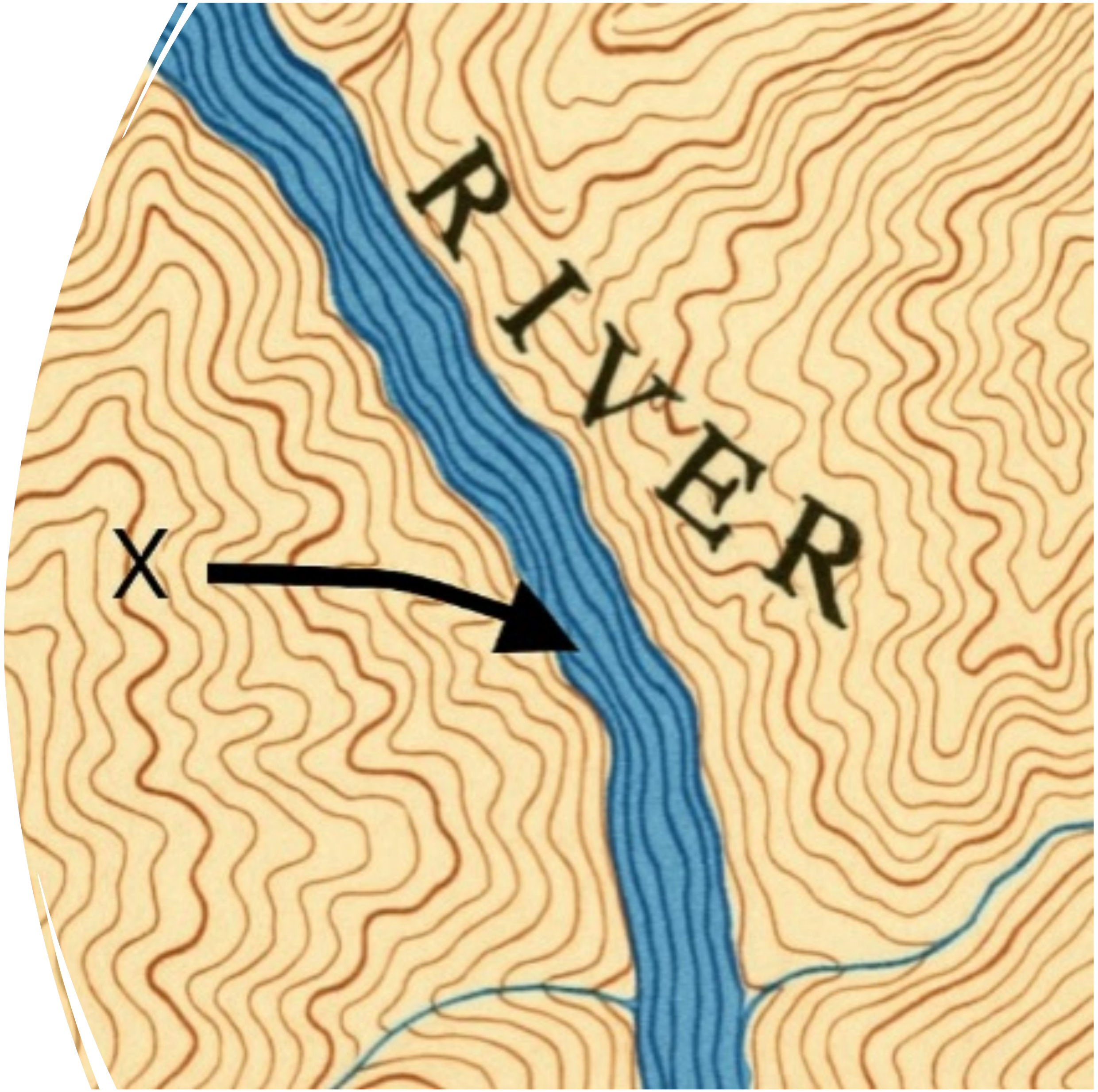
- Use **gradient descent** to adjust the model's parameters
- Repeat across lots of examples until the model gets better at predicting the next word

How it works:

1. The model makes a prediction (e.g. next word).
2. We calculate the loss (how far the prediction is from the correct answer).
3. The gradient tells us the direction that would reduce this loss the most.
4. We adjust the model’s parameters slightly in that direction.
5. Repeat this over and over with many examples.

Intuition of gradient descent

- Imagine you're standing at point **X** in a hilly area, trying to reach the **bottom of the river canyon** (the lowest point—just like trying to minimise a loss function).
- **Look around in all directions** (360° view of the terrain).
- **Figure out which direction goes uphill fastest** (this is the gradient).
- **Now go the opposite way**—that's your step in gradient descent.
- Repeat this process, and you'll gradually make your way down to the river—just like a model improves by reducing its error step by step.



Large Language Models

The transformer model allows fast parallel computations on many GPUs **(large amounts of compute)**

It allows training on **large amounts of data** (think the whole internet worth of text).

It allows adding many and many layers in the model (**large model**)

A large language model is a language model with a large number of parameters, trained on large amounts of data, for long period of time.

Why large language models?

- Scaling the models, compute, and data leads in increase in performance

- Emergent properties at scale (Wei et al 2022)
  - Large models (with 7-100B+ parameters) suddenly become capable of performing tasks they weren't able to do when small (such as 1B or small).

Training the model to chat

A simple language model (also called a pretrained model) is not equipped to chat with an end user, like ChatGPT.

ChatGPT (and many other models) are further trained on supervised data to follow instructions.

Instruction Tuning

- Collect a large dataset of instruction following examples of the form
    - <instruction> <input> <output>
    - For example,
    - Summarize this news article [ARTICLE] [SUMMARY]
    - Answer this question [QUESTION] [ANSWER]
    - Predict the sentiment of this review [REVIEW] [SENTIMENT]....

- This is also a text corpus but in a very specific format.

- Continue training the model on this dataset (again using the same training objective)

Aligning the model to humans' preferences

- Chat based on models are supposed to converse with humans
- Learn from humans' feedback
- Basic idea: Model samples multiple outputs – users rank them based on their preference
  - Convert user preferences into reward scores – more preferred output has higher reward
  - Treat an LLM like an agent and use Reinforcement Learning (RL) to maximize this reward (RL + Human Feedback (RLHF))

What are some problems that ChatGPT’s training leaves it prone to?

Inaccuracies

- The language model doesn’t “plan” what it will say in advance
- The model doesn’t store facts, just outputs plausible looking sentences which may or may not be factual

LLMs Limitations

Beware of "Hallucinations" Which Do Remain Very Real

"Hallucinations are confident statements that are not true".

For the moment, this phenomenon inexorably affects all known LLMs.

Hallucination Example

What are the 5 most recent papers published by Dr Mo El-Haj?

1. **"Neural Discourse Structures for Financial Event Prediction"**
   ACL 2025 – Proposes a transformer-based model for extracting and structuring financial discourse in multilingual reports.
2. **"ViAraT5: A Multilingual Summarisation Model for Vietnamese and Arabic News"**
   EMNLP 2024 – Introduces a cross-lingual summarisation model fine-tuned on Vietnamese–Arabic datasets for news summarisation.
3. **"Faithfulness-Aware Summarisation Using Reward-Guided Decoding"**
   NAACL 2024 – Explores reinforcement-learning-style decoding to improve factual consistency in abstractive summaries.
4. **"Detecting Corporate Greenhushing in ESG Reports Using Contrastive LLMs"**
   Findings of ACL 2024 – Presents a dataset and method for identifying greenwashing and greenhushing in financial disclosures.
5. **"WelshLLM: A Pilot Language Model for Celtic Languages"**
   LREC-COLING 2024 – Documents the development of a low-resource LLM trained on Welsh corpora, with applications in translation and summarisation.

What are the consequences?

The Saga of the Lawyer Who Used ChatGPT

Here's What Happens When Your Lawyer Uses ChatGPT

A lawyer representing a man who sued an airline relied on artificial intelligence to help prepare a court filing. It did not go well.

nytimes.com/2023/05/27/nyregion/avianca-airline-lawsuit-chatgpt.html

The ChatGPT Lawyer Explains Himself

In a cringe-inducing court hearing, a lawyer who relied on A.I. to craft a motion full of made-up case law said he "did not comprehend" that the chat bot could lead him astray.

nytimes.com/2023/06/08/nyregion/lawyer-chatgpt-sanctions.html

ChatGPT Lawyers Are Ordered to Consider Seeking Forgiveness

Steven A. Schwartz and Peter LoDuca must pay a fine and send letters to judges named in a brief filled with fiction, a judge ordered.

nytimes.com/2023/06/22/nyregion/lawyers-chatgpt-schwartz-loduca.html

Why do LLMs hallucinate?

LLMs generate text by predicting the next most likely token based on patterns in data.

No grounding in facts or reality→ LLMs don't "know" the truth—they just generate likely-looking text.

Trained on noisy or incomplete data→ Misinformation, outdated info, or biased text from the web can lead to confident but incorrect answers.

Lack of access to real-time information→ Unless connected to a search tool or external database, LLMs rely on training data alone.

Fluent ≠ Factual→ Just because it sounds plausible doesn't mean it's true.

Types of Hallucination

Type	Description	Example
Factual	Makes up names, dates, or statistics	*“Dr Mo El-Haj is a physicist at MIT”*
Bibliographic	Invents papers, citations, or authors	*“Published in Nature, 2023”* (never happened)
Contextual	Misinterprets question context	*User: “Translate this” → Model summarises instead*
Entity Linking	Confuses similar people/places/things	*Mixes up two authors with similar names*

How Can We Reduce Hallucination?

Hallucination can't be fully eliminated, but it can be mitigated through smarter design:

📚 **Retrieval-Augmented Generation (RAG)** → Combine the LLM with a search engine or external knowledge base to ground answers in real data (next week's lecture).

📊 **Fine-tuning on high-quality domain-specific data** → Improves factual accuracy for specialised tasks (e.g. legal, medical, financial LLMs).

🔍 **Post-hoc fact checking**→ Use separate models or tools to verify claims made by the LLM before presenting them.

🧑‍🏫 **Human-in-the-loop supervision** → Especially in high-stakes applications, add expert review and correction.

📏 **Reinforcement Learning from Human Feedback (RLHF)** → Fine-tunes the model to prefer helpful and truthful responses.

🗣️ **Well designed prompts** → *How you ask matters (prompt engineering)*.

🧩 What Is Prompt Engineering?

Prompt engineering is the practice of designing input text (prompts) in a way that guides a language model to produce better, more accurate, and useful outputs.

In other words:

"How you ask matters."

A well-designed prompt can:

- Reduce hallucination 🧠
- ❌Improve clarity and relevance
- Control tone, format, or level of detail
- Encourage step-by-step reasoning or cautious answers

It's like giving the model good instructions—specific, clear, and context-aware.

Example – Reducing Hallucination with Better Prompts

Let’s say we ask the model:

❌ **Prompt 1 (Vague):**
“Who is Dr Mo El-Haj?” → **Possible Hallucination:**

“Dr Mo El-Haj is a theoretical physicist at MIT, known for work on quantum mechanics.”

✅ **Prompt 2 (Guided):**
“Based on verified academic sources, provide a brief summary of Dr Mo El-Haj’s academic background and research in NLP.

If unknown, say so.”→ **Better Response:**
“I don’t have access to real-time or verified academic databases, but if Dr Mo El-Haj is a researcher in NLP, they may be affiliated with a university or group focused on language modelling.”

Student Task – Prompt Debugging Challenge

Instructions:
The following three example prompts below can lead to hallucinated or unhelpful outputs.
Your task is to **rewrite each prompt** to reduce hallucination and improve output quality.

✍️ **Prompt 1:**
"Tell me about the inventor of the Vietnamese language."
👉 *(No one "invented" it—this will likely hallucinate.)*

✍️ **Prompt 2:**
"What is the best financial model used in 2023?"
👉 *(Too broad and possibly outdated—encourages speculation.)*

✍️ **Prompt 3:**
"Summarise the latest research on Vietnamese NLP."
👉 *(Unclear what counts as "latest"; may hallucinate papers.)*

Thanks! Questions?