Lecture 9:

The Promise and Peril of Large Language Models & Generative AI

CMSC 25910

Winter 2026

The University of Chicago

The Evolution of Modeling Language

Initial Attempts at Modeling Language

- Randomly select words from vocabulary
- Markov models
- Create some sort of grammar
- Build up some sort of knowledge base

Classical Example: Markov Chain

- Example text: *UChicago is fun. I attend UChicago for school. UChicago is cold for months.*
- Sample tokenization: *[START]* [*UChicago] [is] [fun] [.] [I] [attend] [UChicago] [for] [school] [.] [UChicago] [is] [cold] [for] [months] [.] [END]*
- Probabilities are based on how often some suffix follows a given prefix (e.g., “UChicago" is followed 2/3rds of the time by “is” and 1/3rd of the time by “for”)
- Smoothing assigns non-zero probabilities to all tokens that otherwise have 0 probability

2010s Approach

- Recurrent Neural Networks (RNNs)

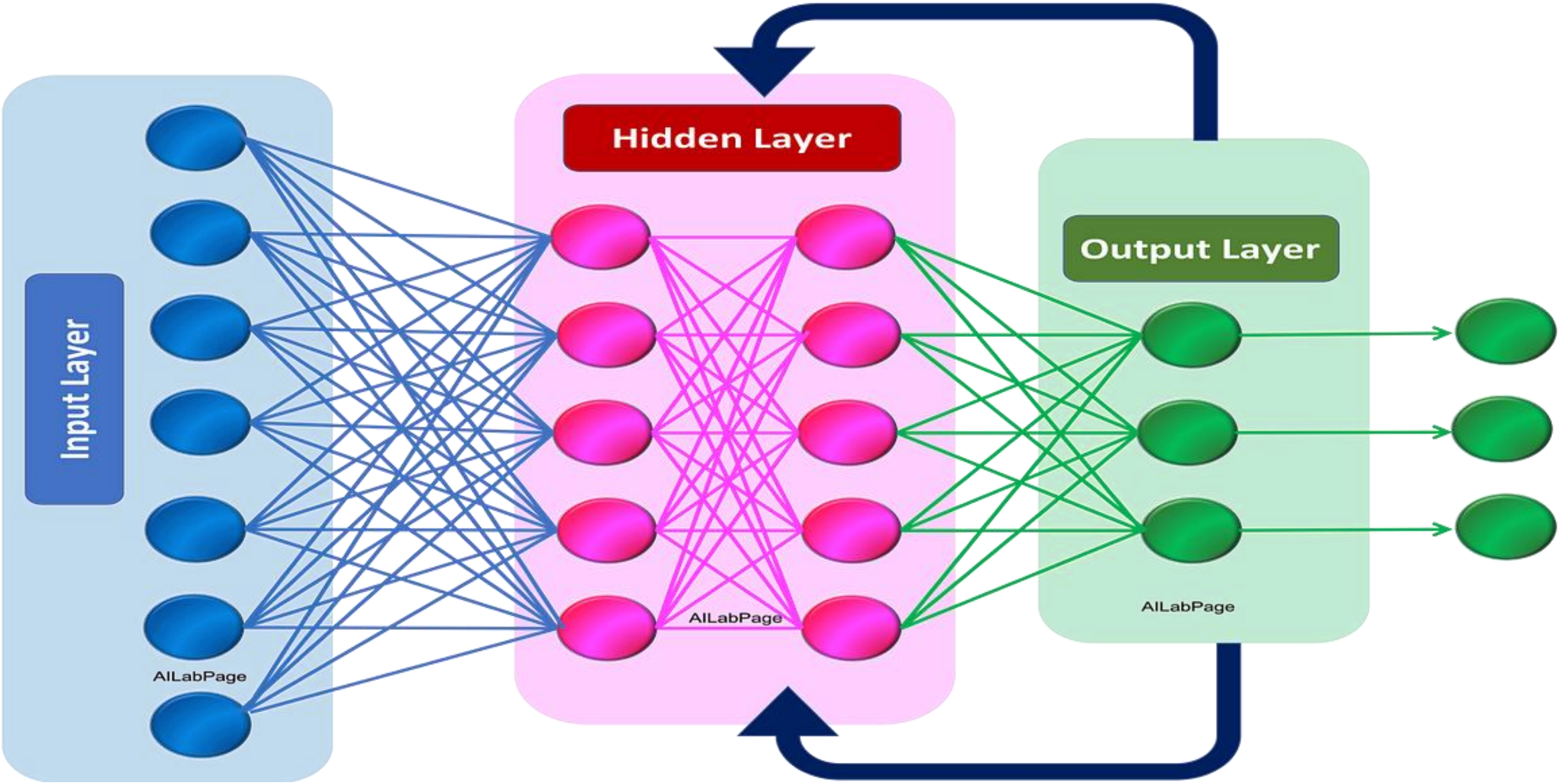


Image from https://medium.datadriveninvestor.com/recurrent-neural-network-58484977c445

2010s Approach

- Recurrent Neural Networks (RNNs)
    - Long Short-Term Memory (LSTM)

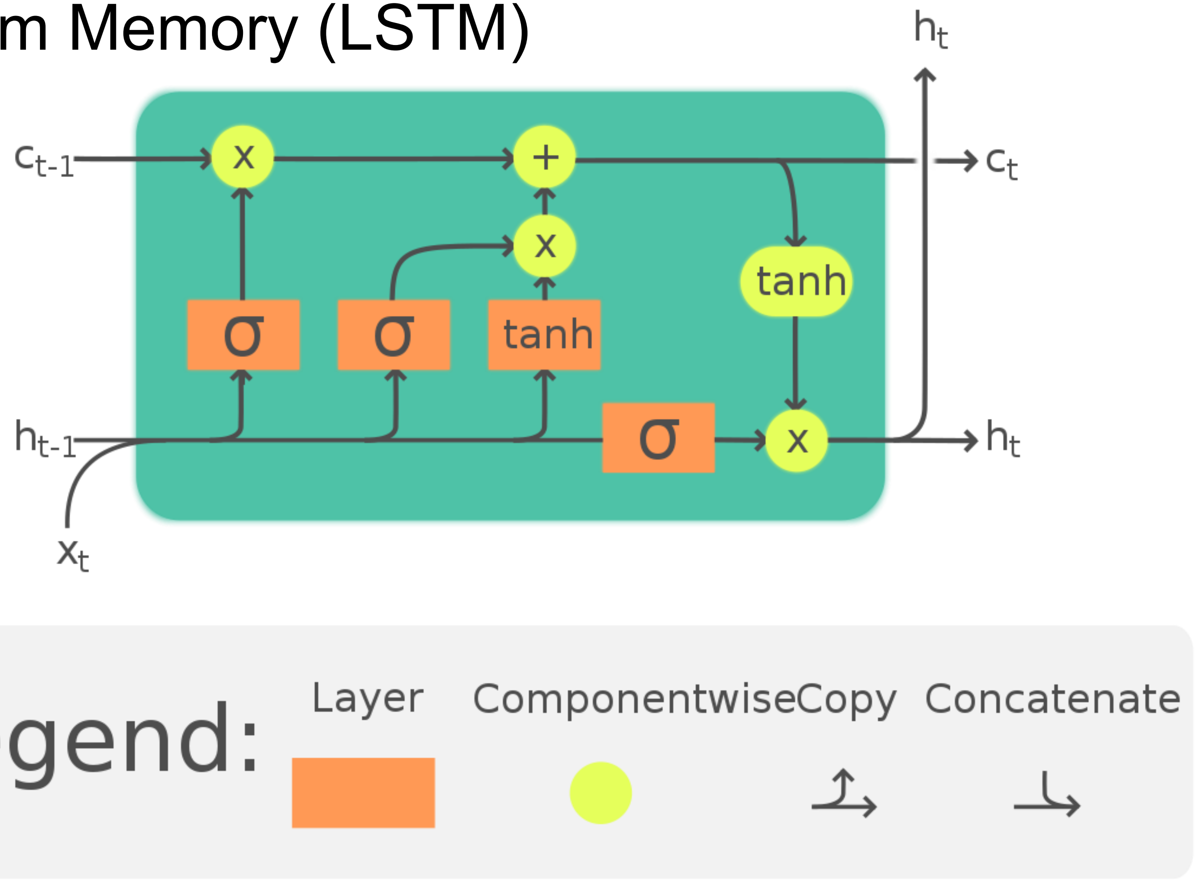


Image from https://upload.wikimedia.org/wikipedia/commons/thumb/9/93/LSTM_Cell.svg/450px-LSTM_Cell.svg.png

Large Language Models (LLMs)

This section of the presentation was derived in part from slides originally created by Madison Pickering, which themselves include many figures from Elena Voita's NLP Course For You (September 2020): https://lena-voita.github.io/nlp_course.html

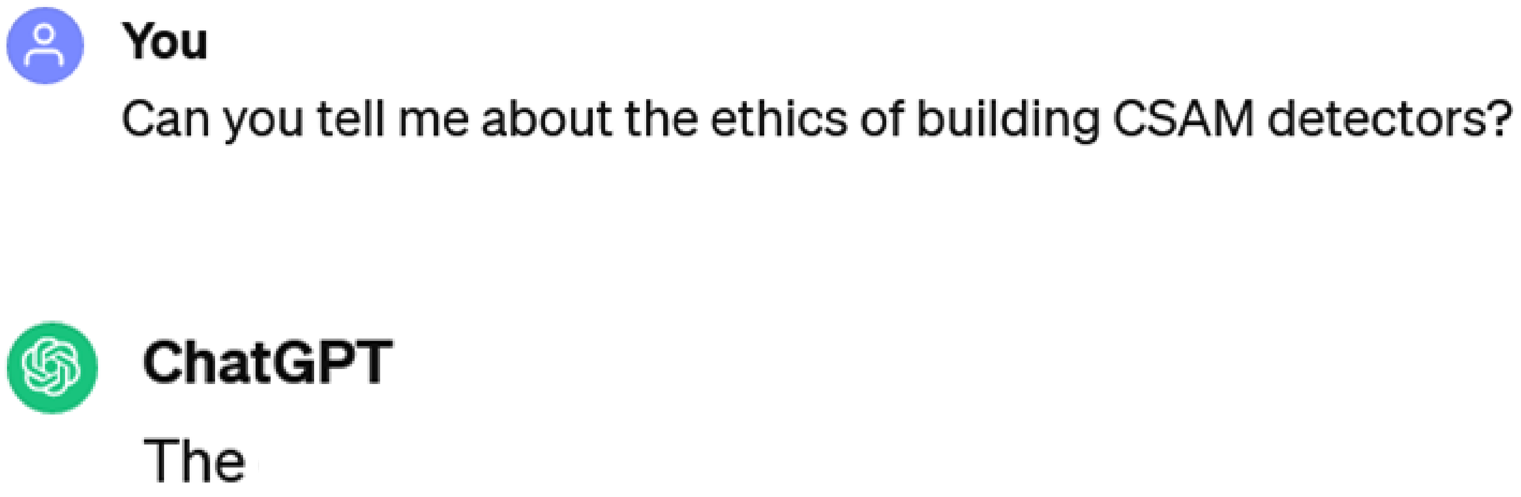
Some Notable Foundation LLMs (as of 2026)

- Google Gemini 3 (Flash, Pro)
- Meta Llama 4 (Scout, Maverick, Behemoth)
  - Open-weight
- Anthropic Claude 4.5 (Haiku, Sonnet, Opus)
- Deepseek
  - Open-weight
- Alibaba Qwen 3.5
  - Open-weight
- OpenAI GPT-5
  - GPT-2 was the last open-weight model in this family

What We Will Cover Today:

- LLMs can produce guesses for what the most likely bit of text should be *given the text that they have seen*
- LLMs are trained on vast corpora of text from the Internet
- LLMs extend an NLP architecture called a Transformer

“LLMs can produce guesses for what the most likely bit of text should be, given the text that they have seen”



You

Can you tell me about the ethics of building CSAM detectors?

ChatGPT

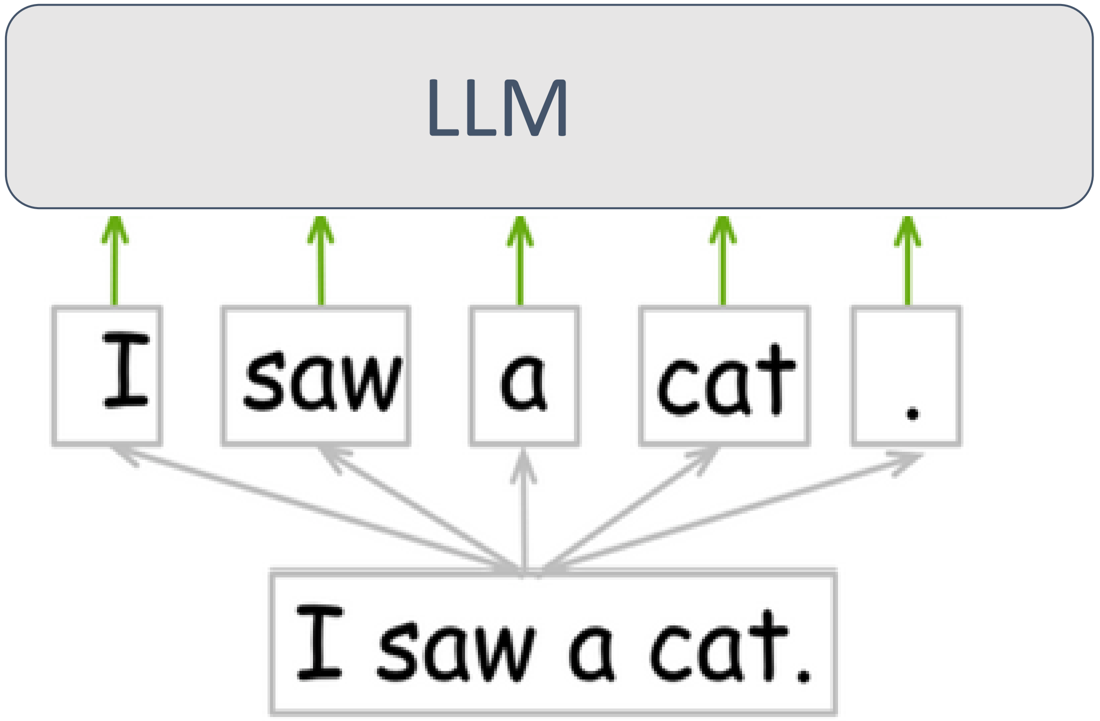
The

LLM outputs likely text…we append the produced text to our input and generate again. We can repeat this process to generate large amounts of text (“auto-regressive”)

“LLMs can produce guesses for what the most likely ***bit of text*** *should be, given the text that they have seen”*

- What, precisely, does “bit of text” mean?

- → The atomic units that LLMs operate on are called “**tokens”**

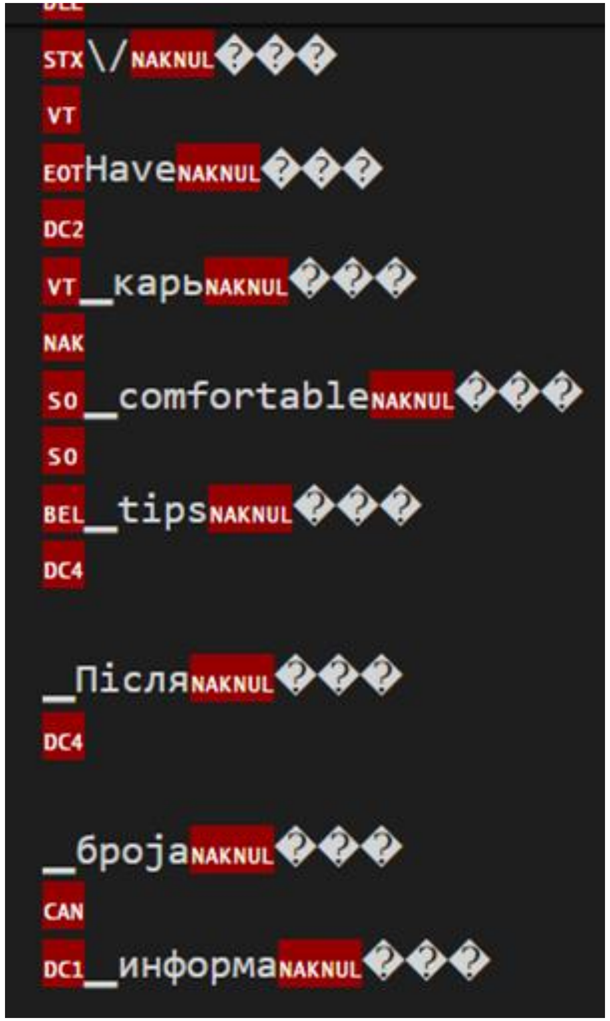


Sequence of tokens

“Tokenization”

Text (your input)

Examples…

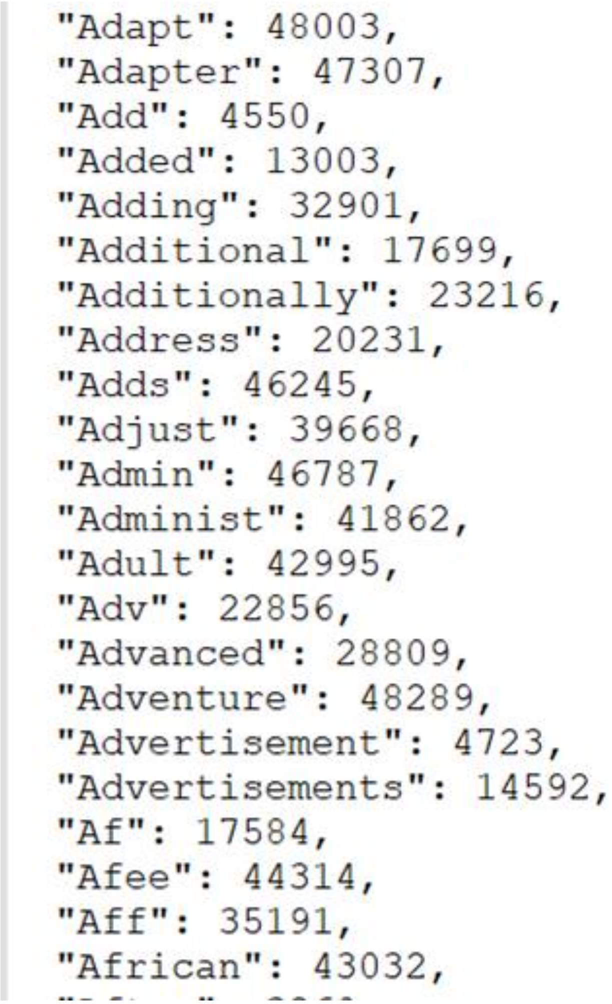


```
STX\/NAKNUL���
VT
EOTHaveNAKNUL���
DC2
VT_карьNAKNUL���
NAK
SO_comfortableNAKNUL���
SO
BEL_tipsNAKNUL���
DC4

_ПісляNAKNUL���
DC4

_бројаNAKNUL���
CAN
DC1_информаNAKNUL���
```

Llama2’s tokens



```
"Adapt": 48003,
"Adapter": 47307,
"Add": 4550,
"Added": 13003,
"Adding": 32901,
"Additional": 17699,
"Additionally": 23216,
"Address": 20231,
"Adds": 46245,
"Adjust": 39668,
"Admin": 46787,
"Administ": 41862,
"Adult": 42995,
"Adv": 22856,
"Advanced": 28809,
"Adventure": 48289,
"Advertisement": 4723,
"Advertisements": 14592,
"Af": 17584,
"Afee": 44314,
"Aff": 35191,
"African": 43032,
```

GPT-2’s tokens

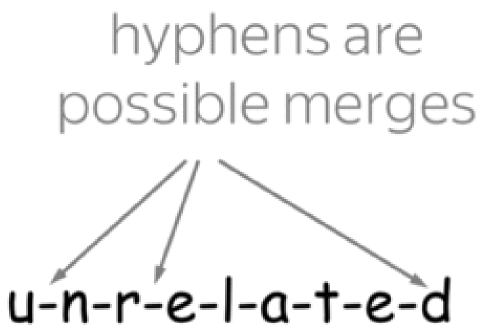
All the tokens that a model “knows” comprise the model’s **vocabulary**

Model vocabularies of modern LLMs are often ~30k-60k tokens in size

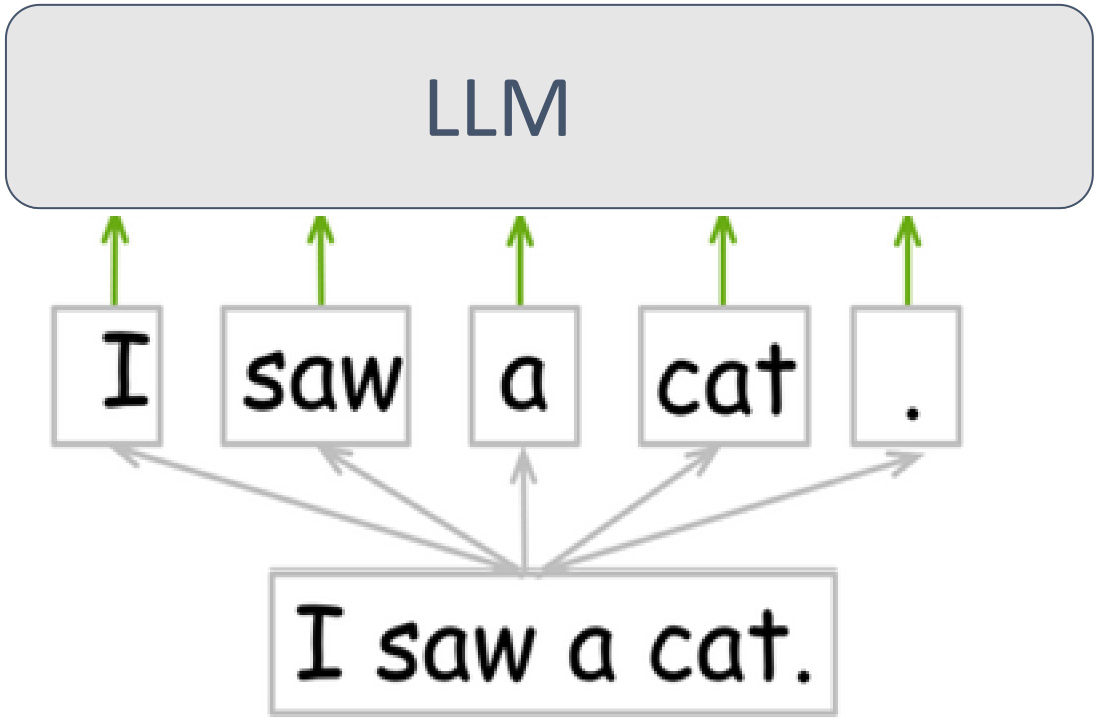
Built by keeping frequent “words”, splitting less frequent words

Tokenizing Input Text

1. Look at the text + our vocabulary
2. Find the “highest” merge (corresponds to the most frequent merge based on the input to the vocab)
3. Repeat



"LLMs can produce ***guesses*** *for what the most likely bit of text should be given the text that they have seen"*



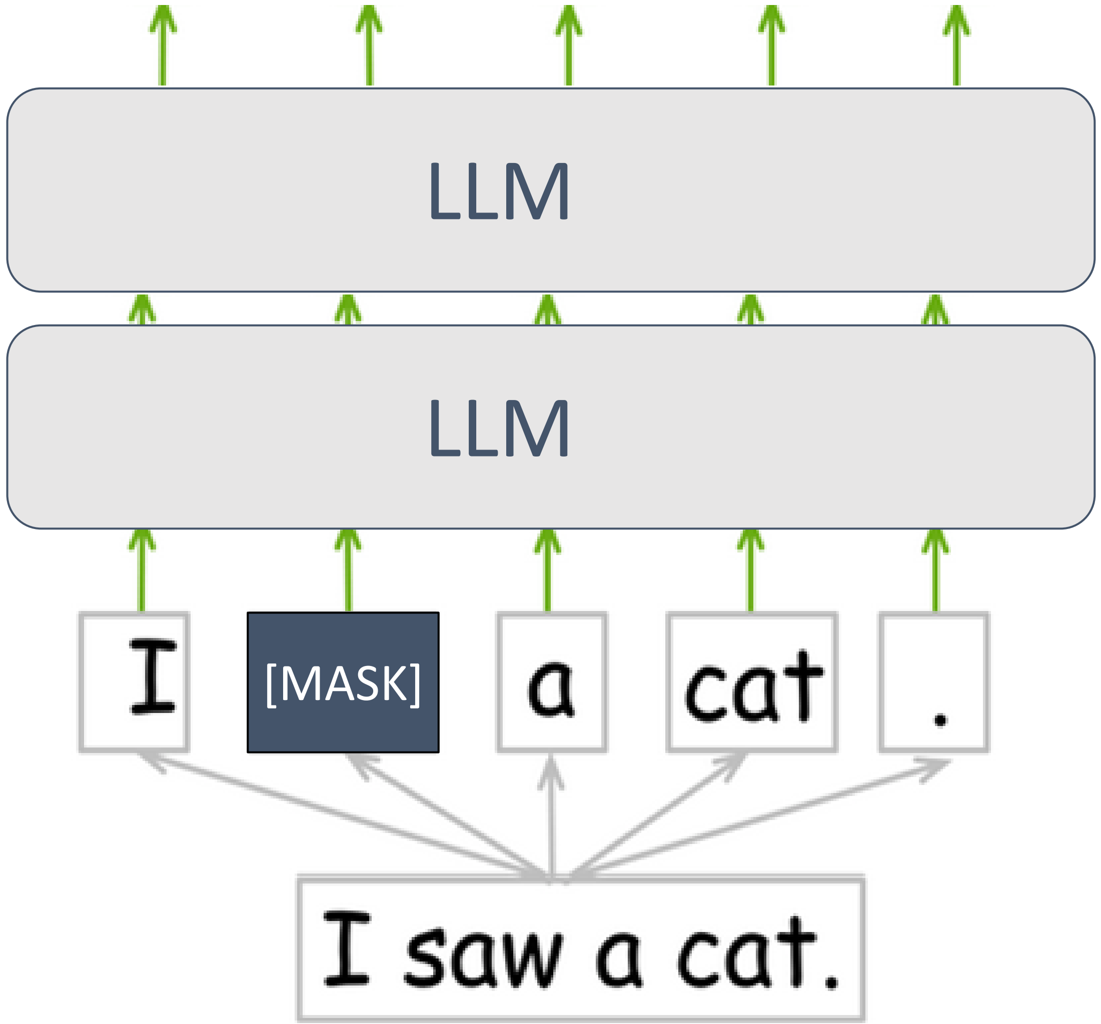
Sequence of tokens

Text (your input)

LLM Output

- Depends on the training objective, and consequently, the architecture of the LLM!

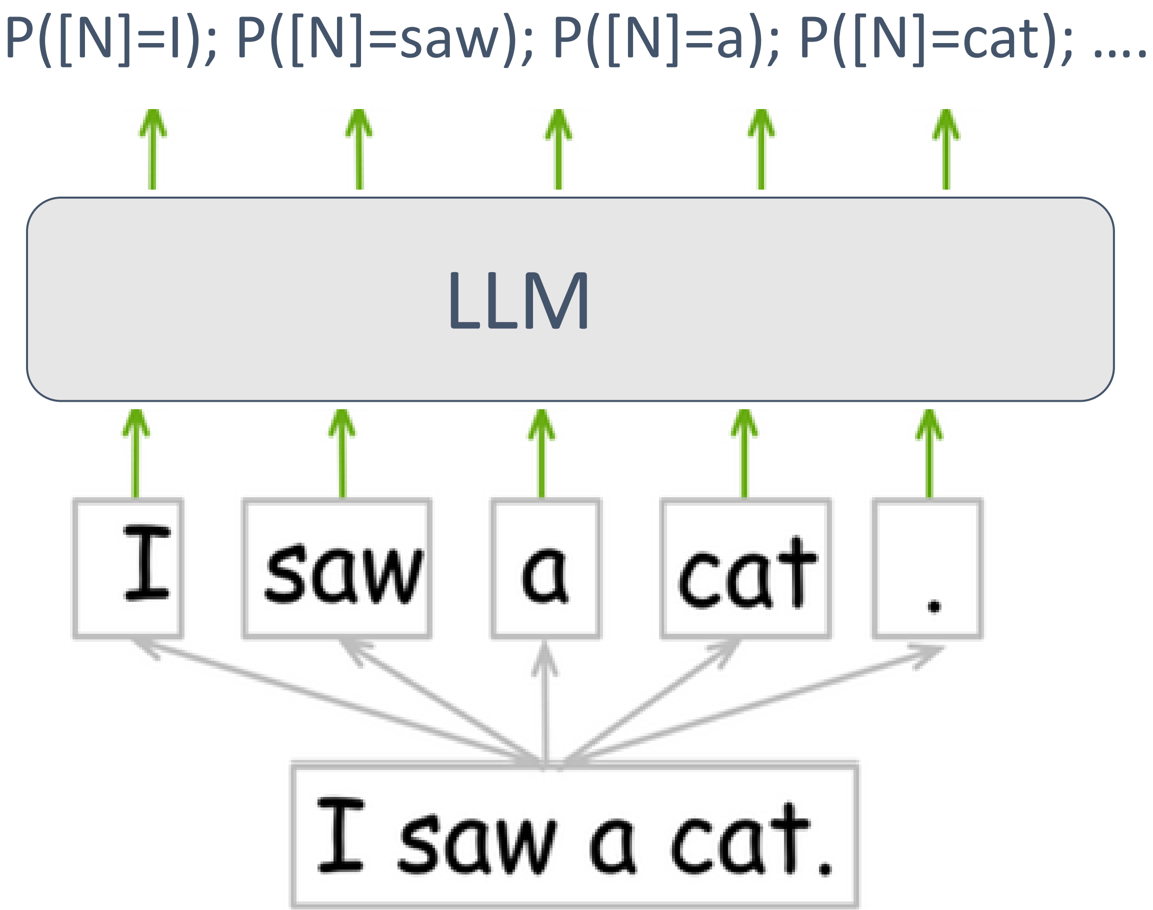
- Two main training objectives:
  - **MLM**: “Masked Language Modeling”
  - **CLM**: “Causal Language Modeling”



MLM: “Masked Language Modeling”

Final output: “I **hugged** a cat.”

CLM: “Causal Language Modeling”


P([N]=I); P([N]=saw); P([N]=a); P([N]=cat); ....

Model receives the input input and determines what the likely next token (**[N]**) is

Final output: “I saw a cat..**It’s** “

MLM vs. CLM

- MLM:
  - Model should look at tokens to the left of [Mask], and *to the right of [Mask]* to decide what [Mask] should be
  - Excels at constrained transformations of input and output (translation, some classification tasks)

MLM vs. CLM

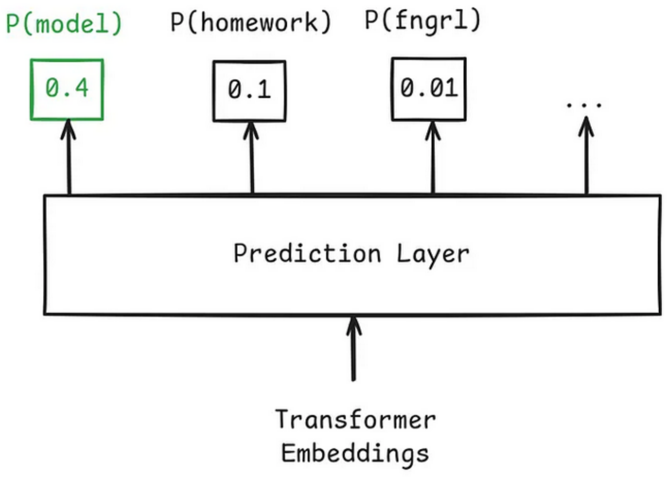
- MLM:
  - Model should look at tokens to the left of [Mask], and *to the right of [Mask]* to decide what [Mask] should be
  - Excels at constrained transformations of input and output (translation, some classification tasks)

- CLM
  - Model should look at all tokens to the left of [N] to decide what [N] should be
  - Excels at open-ended generation

Back to Basics

LLMs generate the next bit of text based on the text generated so far (including the prompt) and the patterns learned from training data.

Possible Sampling Strategies



The prediction layer produces a huge list containing the probabilities for all possible next tokens.

- Pick the most likely next token
    - Ends up being boring and repetitive
- Sample from all possible tokens
- Top-k sampling: Sample only from the k most probable next tokens (where k is fixed)
- Top-p sampling (nucleus sampling): Sample from the p most probable next tokens (where p is chosen based on the cumulative probability of those tokens)

Image from https://medium.com/thinking-sand/the-definitive-guide-to-llm-temperatures-abab311260a6 (also has great explanations)

Possible Sampling Strategies

- Temperature (T) modifies new probabilities Q
    - 1: Keep the probabilities learned by the model
    - <1: Bias towards the higher probability tokens
    - 0: Pick the most probable next token deterministically(-ish)
    - >1: Flatten out the probabilities so that the higher probability tokens are somewhat less likely than before and the lower probability tokens are somewhat more likely than before

$$Q(x_i) = \frac{P(x_i)^{\frac{1}{T}}}{\sum_{j=1}^{n} P(x_j)^{\frac{1}{T}}}$$

Image from https://medium.com/thinking-sand/the-definitive-guide-to-llm-temperatures-abab311260a6 (also has great explanations)

What Are Logprobs?

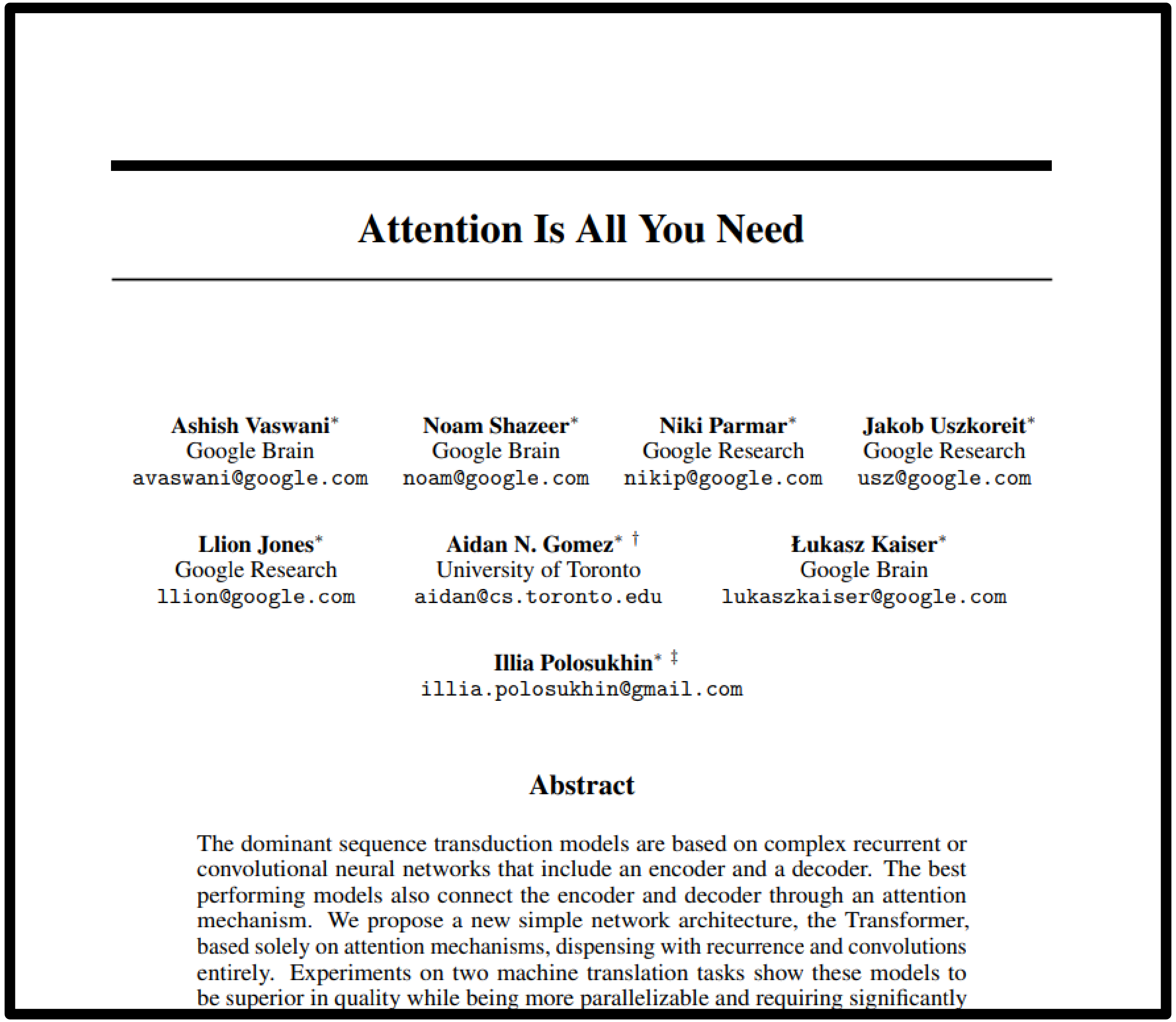
- Logprob = ln(probability)
- Why?
    - Probabilities from LLMs are typically very small
    - When you start multiplying probabilities (e.g., for the total probability of a sentence, you'd multiply the probabilities of each token), we now have to worry about floating point error
    - $\ln(p_1 * p_2 * p_3) = \ln(p_1) + \ln(p_2) + \ln(p_3)$, so you can sum the logprobs and then exponentiate ("exp") the sum to avoid multiplying many extremely small numbers

See https://medium.com/thinking-sand/understanding-llm-logprobs-029794105903 for nice explanations

Transformers (the Architecture Underpinning LLMs)

This section of the presentation was derived in part from slides originally created by Madison Pickering, which themselves include many figures from Elena Voita's NLP Course For You (September 2020): https://lena-voita.github.io/nlp_course.html

Attention



Attention Is All You Need

Ashish Vaswani[*]
Google Brain
avaswani@google.com

Noam Shazeer[*]
Google Brain
noam@google.com

Niki Parmar[*]
Google Research
nikip@google.com

Jakob Uszkoreit[*]
Google Research
usz@google.com

Llion Jones[*]
Google Research
llion@google.com

Aidan N. Gomez[*] [†]
University of Toronto
aidan@cs.toronto.edu

Łukasz Kaiser[*]
Google Brain
lukaszkaiser@google.com

Illia Polosukhin[*] [‡]
illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly

Neural Networks vs. Transformers

LLMs are not neural networks; they extend the transformer architecture

Neural networks:

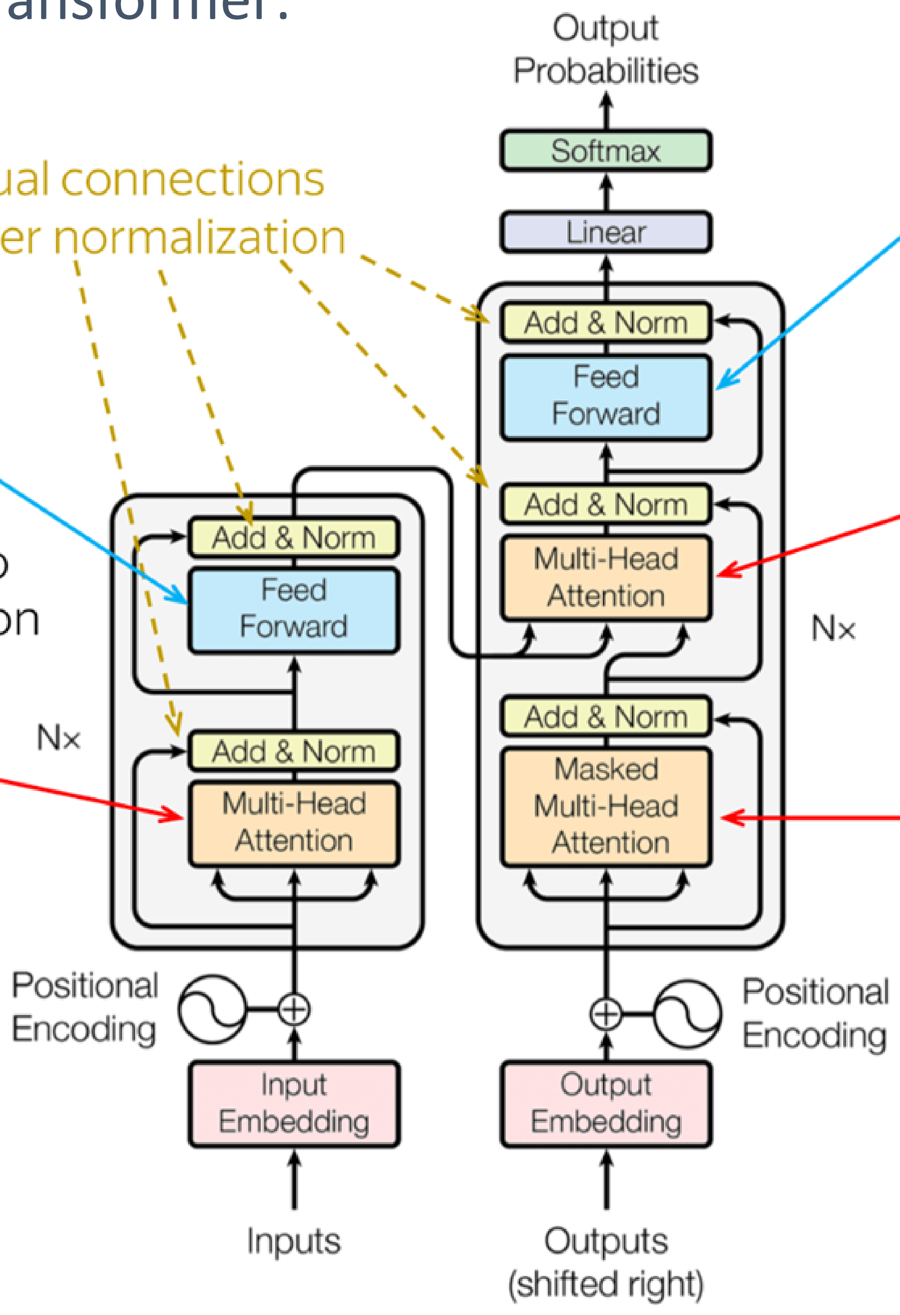
- Use recurrence to do NLP
- Comprised of artificial neurons

Transformers

- No recurrence (efficient!)
- Comprised of Transformer blocks (which contain FFNN)

Both benefit from scale (making the model bigger/more parameters) and lots of data!

The original design of the Transformer:



Residual connections
and layer normalization

Feed-forward network:
after taking information from
other tokens, take a moment to
think and process this information

↑

Encoder self-attention:
tokens look at each other

queries, keys, values
are computed from
encoder states

Feed-forward network:
after taking information from
other tokens, take a moment to
think and process this information

↑

Decoder-encoder attention:
target token looks at the source

queries – from decoder states; keys
and values from encoder states

↑

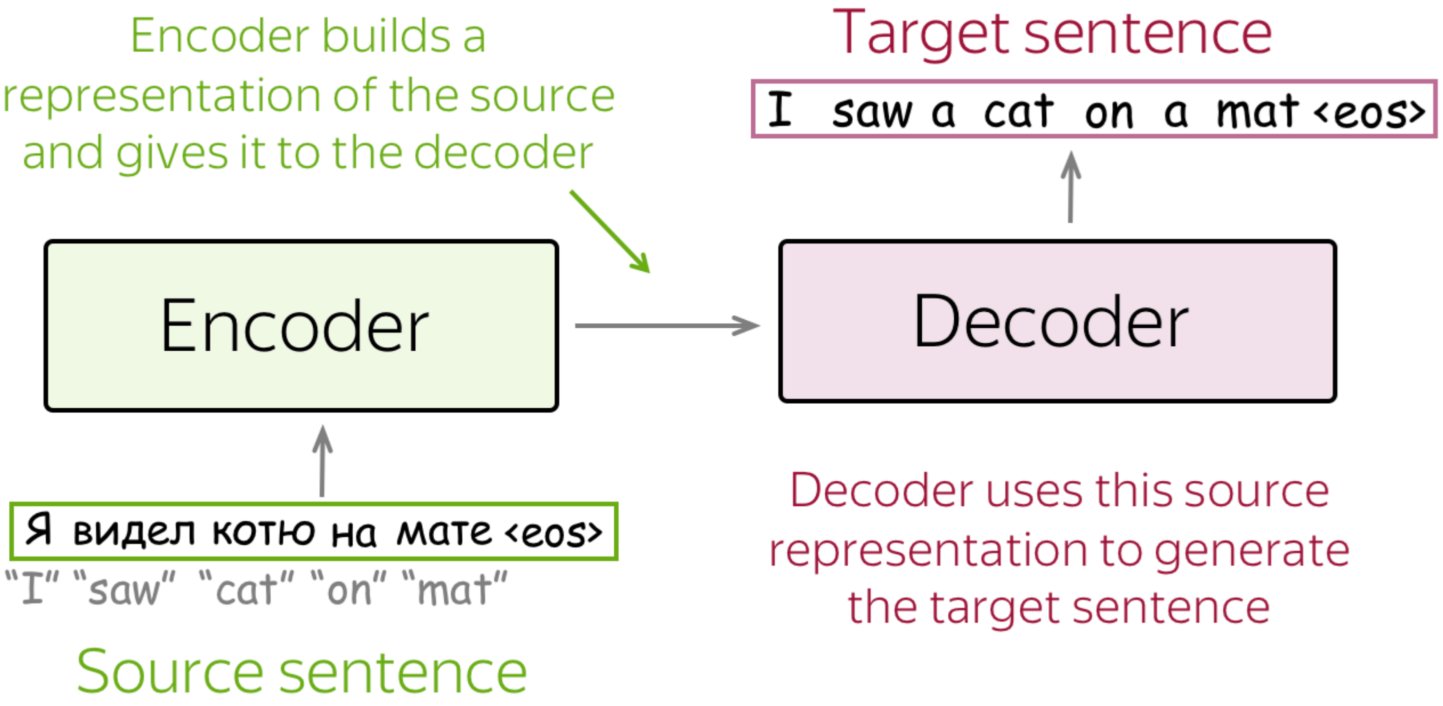
Decoder self-attention (masked):
tokens look at the previous tokens

queries, keys, values are computed
from decoder states

Image taken from Elena Voita, NLP Course For You, https://lena-voita.github.io/nlp_course.html

Quick Breakdown of Parts:

High level: encoder-decoder



Encoder builds a representation of the source and gives it to the decoder

Target sentence

I saw a cat on a mat <eos>

Encoder

Decoder

Я видел котю на мате <eos>

"I" "saw" "cat" "on" "mat"

Source sentence

Decoder uses this source representation to generate the target sentence

Pretty much any LLM that is primarily used for CLM (e.g., GPT-3) is Decoder-Only

Image taken from Elena Voita, NLP Course For You, https://lena-voita.github.io/nlp_course.html

Embedding, Positional Encoding

- **Embedding** transforms the input words into some vector representation that captures its "meaning"
  - Words with similar meaning should be close to each other in vector space
- **Positional Encoding** is added because words have different meanings based on where they are in a sentence!
  - This is not needed in recurrent NNs because the recurrence captures the position

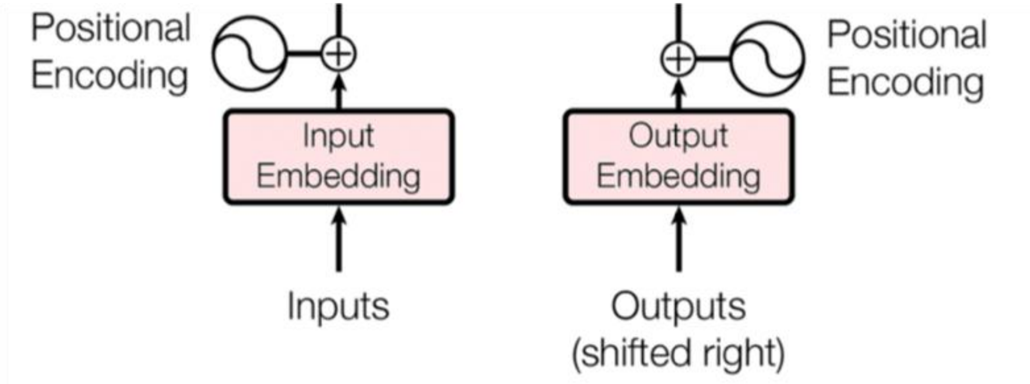
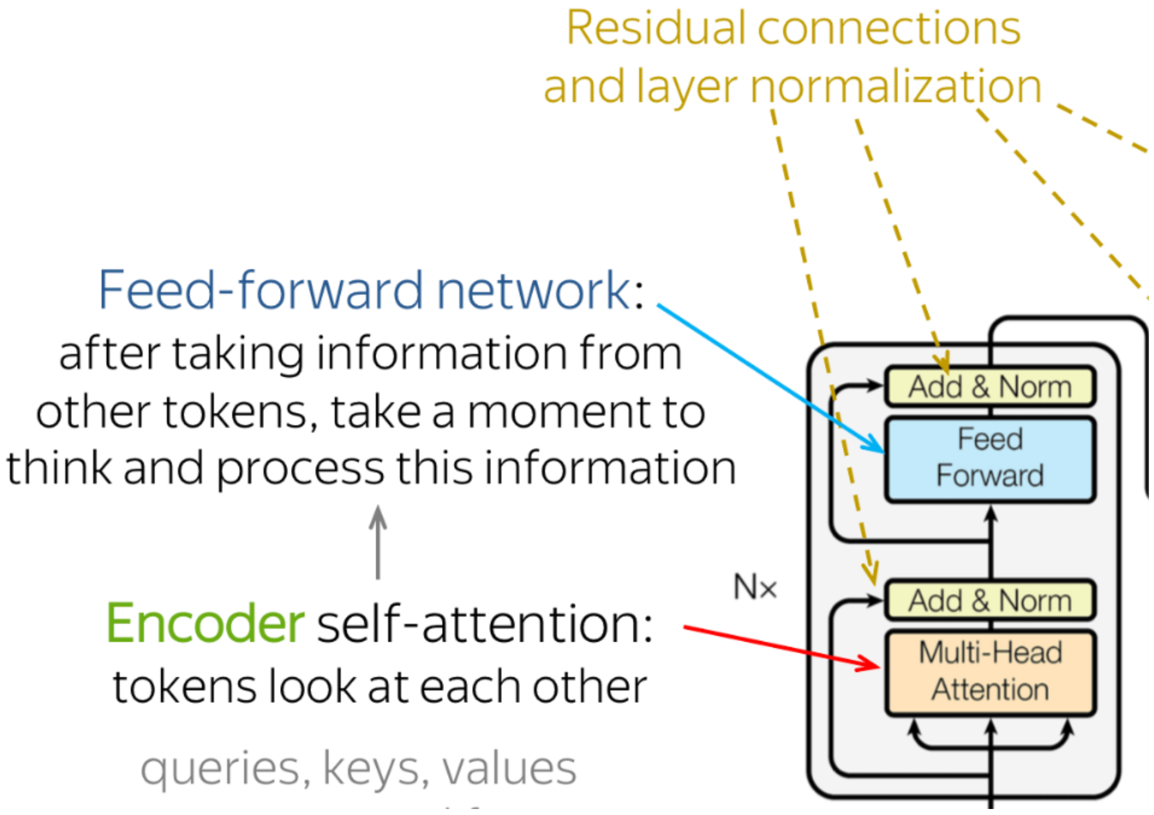


Image taken from Elena Voita, NLP Course For You, https://lena-voita.github.io/nlp_course.html

- **Attention:** based on all the tokens in the input decide which tokens we should *pay attention to* (i.e., which should have higher weight)
- **Feed-Forward Network:** This is a 2-layer neural network with no recurrence
- **Add & Norm, Residual Connections:** Used to “stabilize” the model/help convergence


Residual connections and layer normalization

Feed-forward network: after taking information from other tokens, take a moment to think and process this information

Encoder self-attention: tokens look at each other

queries, keys, values

Image taken from Elena Voita, NLP Course For You, https://lena-voita.github.io/nlp_course.html

A Look at the Decoder

Mostly the same stuff, but, there is a **masked attention** head to make sure that the the tokens at position i only “look at” the tokens in positions < i

Goal: only look at the left context (since in English, we read left to right)

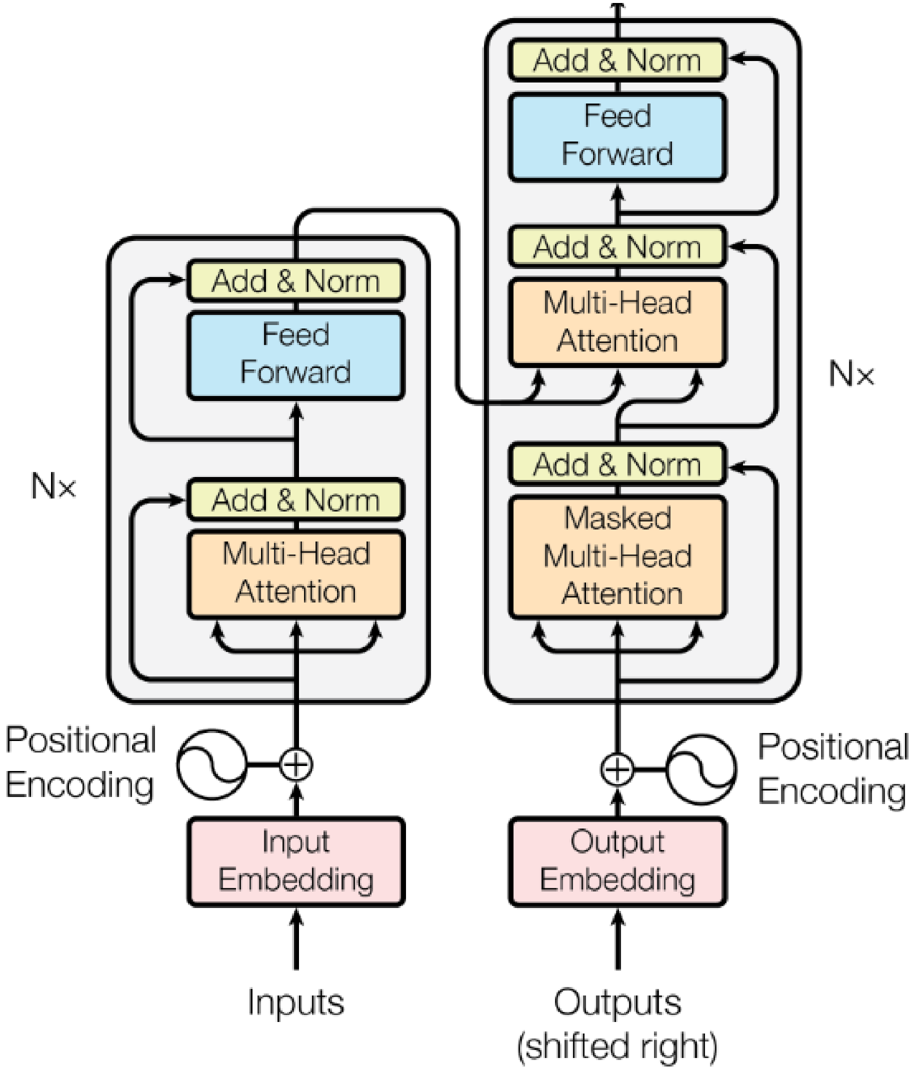


Image taken from Elena Voita, NLP Course For You, https://lena-voita.github.io/nlp_course.html

Final Transformations

Linear layer: final neural network post-processing

Softmax: A function which *gives us the normalized probabilities*: given any input vector, will transform the vector such that the elements all sum to 1 (probabilities proportional to the exponents of the input numbers)

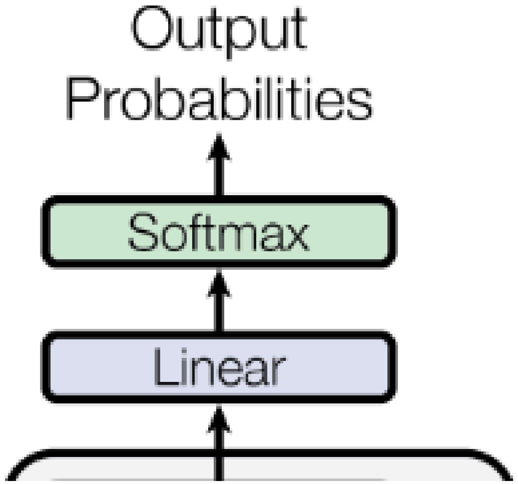


Image taken from Elena Voita, NLP Course For You, https://lena-voita.github.io/nlp_course.html

Attention

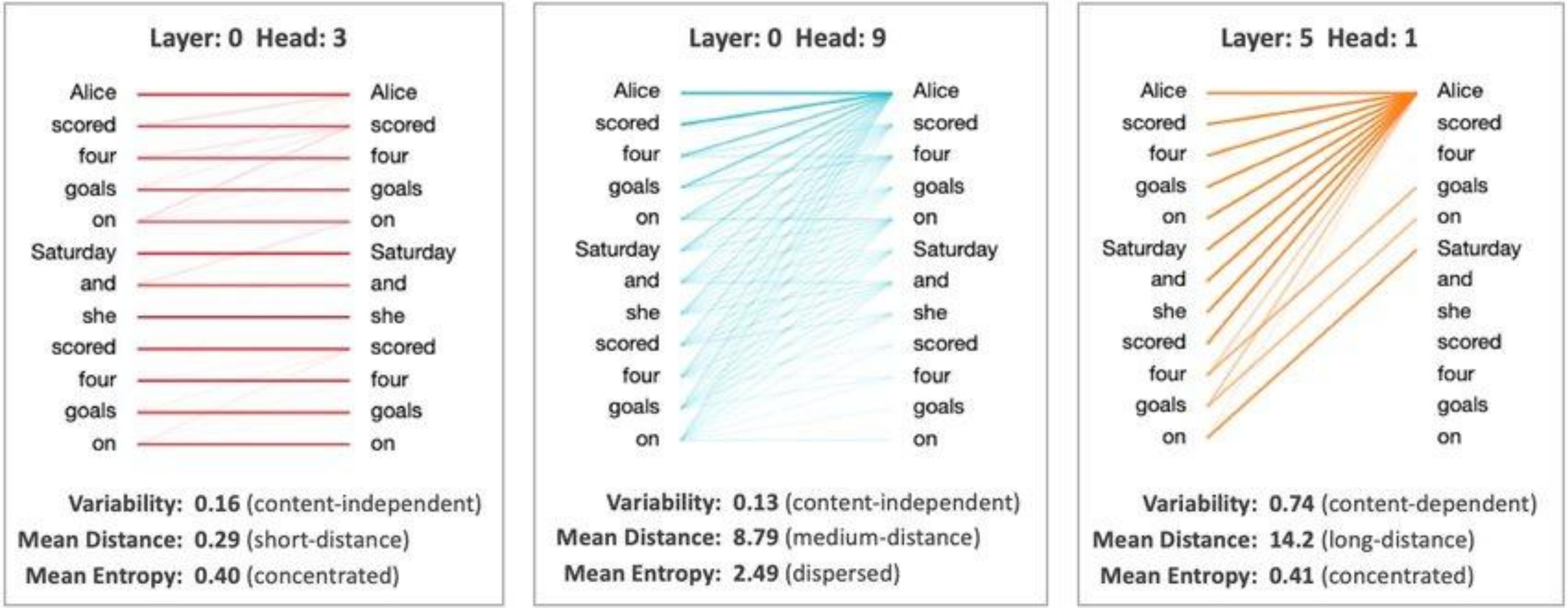

Layer: 0 Head: 3
Alice
scored
four
goals
on
Saturday
and
she
scored
four
goals
on
Variability: 0.16 (content-independent)
Mean Distance: 0.29 (short-distance)
Mean Entropy: 0.40 (concentrated)
Layer: 0 Head: 9
Alice
scored
four
goals
on
Saturday
and
she
scored
four
goals
on
Variability: 0.13 (content-independent)
Mean Distance: 8.79 (medium-distance)
Mean Entropy: 2.49 (dispersed)
Layer: 5 Head: 1
Alice
scored
four
goals
on
Saturday
and
she
scored
four
goals
on
Variability: 0.74 (content-dependent)
Mean Distance: 14.2 (long-distance)
Mean Entropy: 0.41 (concentrated)

Image from https://www.researchgate.net/figure/Attention-heads-in-GPT-2-visualized-for-an-example-input-sentence-along-with-aggregate_fig3_333716912

Transformers

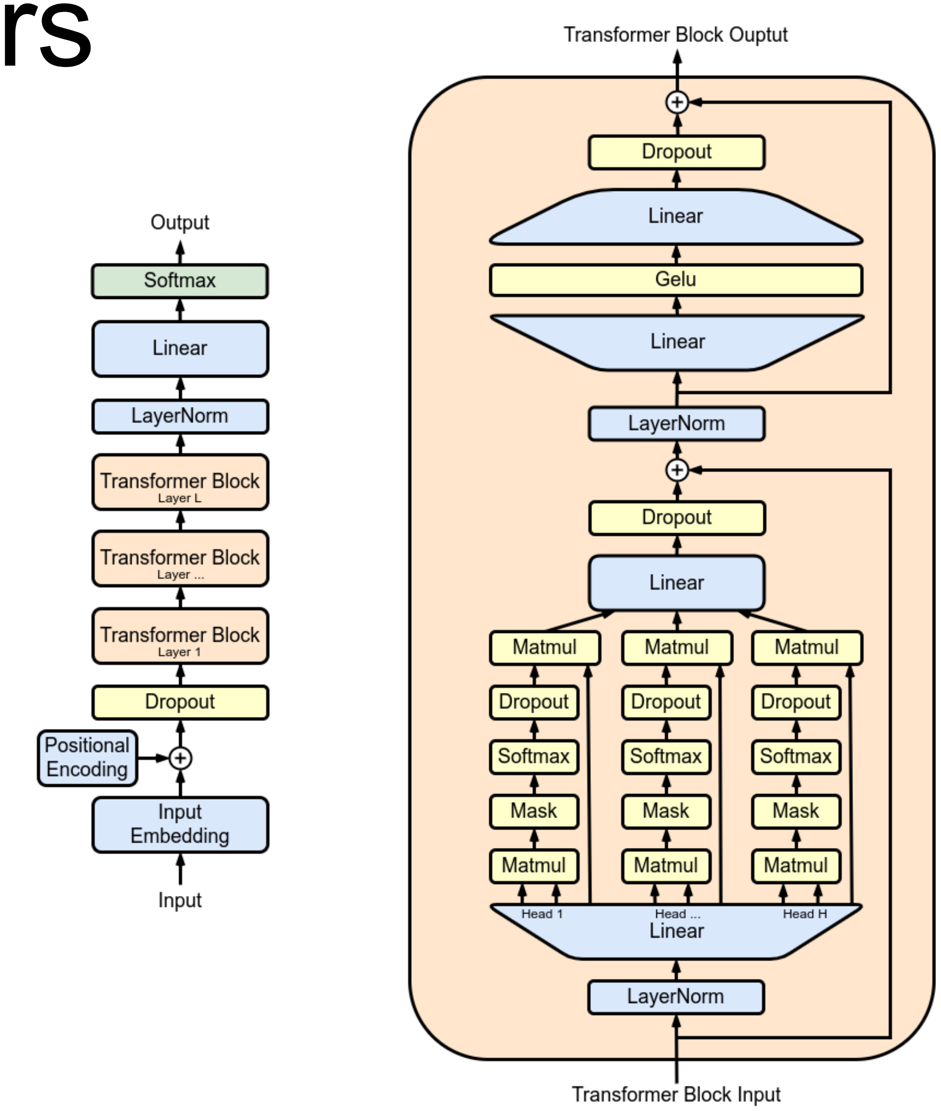


Image from https://upload.wikimedia.org/wikipedia/commons/thumb/9/91/Full_GPT_architecture.png/330px-Full_GPT_architecture.png

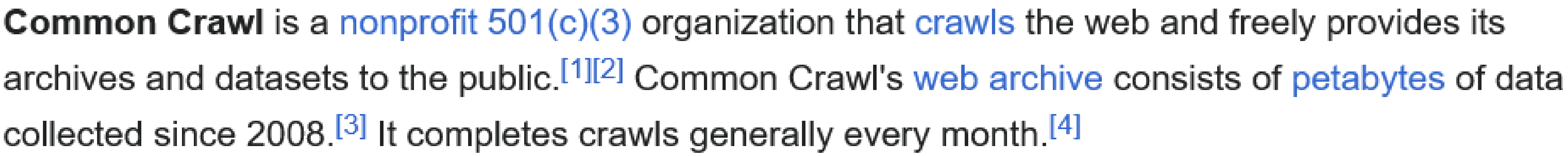
Data Sources For Generative AI

What Are LLMs Trained On?

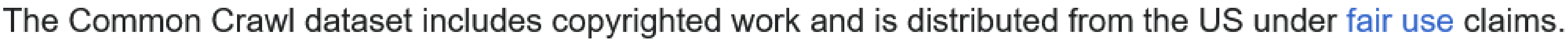
- Short answer: scraped webpages (e.g., Wikipedia, Reddit, GitHub,…)
  - Heuristically deduplicated
  - Filtered/cleaned based on human upvotes, website traffic, …

- …but also, trained on you!
  - Queries/interactions to most proprietary LLMs are logged

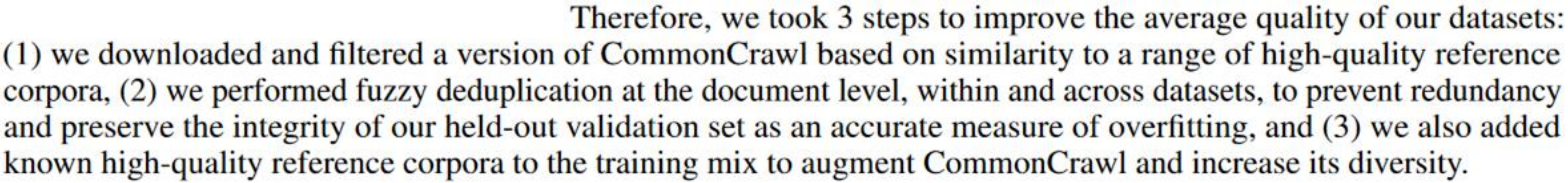
Example: GPT-3



Common Crawl is a nonprofit 501(c)(3) organization that crawls the web and freely provides its archives and datasets to the public.[1][2] Common Crawl's web archive consists of petabytes of data collected since 2008.[3] It completes crawls generally every month.[4]

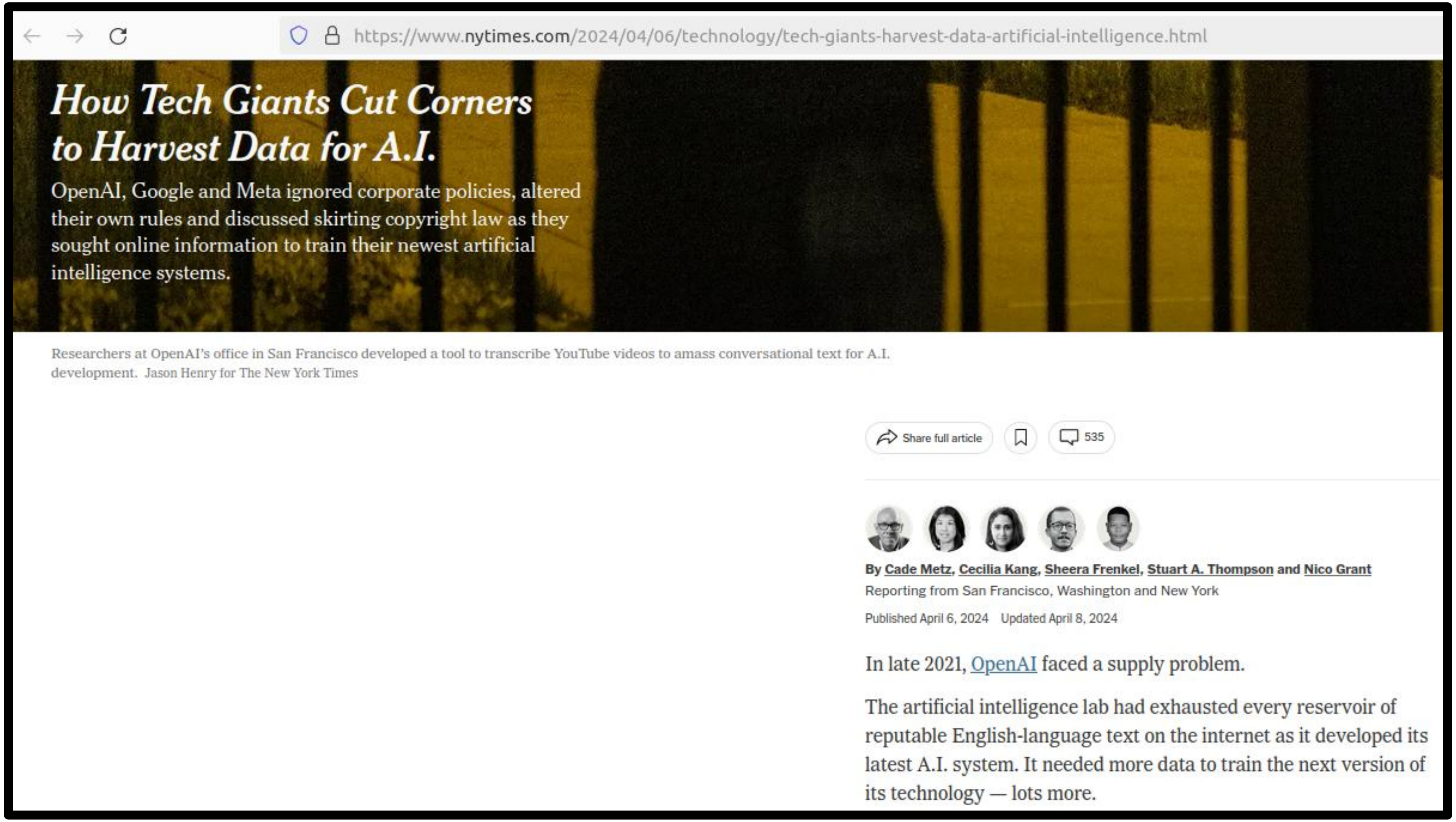


The Common Crawl dataset includes copyrighted work and is distributed from the US under fair use claims.



Therefore, we took 3 steps to improve the average quality of our datasets: (1) we downloaded and filtered a version of CommonCrawl based on similarity to a range of high-quality reference corpora, (2) we performed fuzzy deduplication at the document level, within and across datasets, to prevent redundancy and preserve the integrity of our held-out validation set as an accurate measure of overfitting, and (3) we also added known high-quality reference corpora to the training mix to augment CommonCrawl and increase its diversity.

Training Data


https://www.nytimes.com/2024/04/06/technology/tech-giants-harvest-data-artificial-intelligence.html
Share full article
535

How Tech Giants Cut Corners to Harvest Data for A.I.

OpenAI, Google and Meta ignored corporate policies, altered their own rules and discussed skirting copyright law as they sought online information to train their newest artificial intelligence systems.

Researchers at OpenAI's office in San Francisco developed a tool to transcribe YouTube videos to amass conversational text for A.I. development. Jason Henry for The New York Times

By Cade Metz, Cecilia Kang, Sheera Frenkel, Stuart A. Thompson and Nico Grant

Reporting from San Francisco, Washington and New York

Published April 6, 2024 Updated April 8, 2024

In late 2021, OpenAI faced a supply problem.

The artificial intelligence lab had exhausted every reservoir of reputable English-language text on the internet as it developed its latest A.I. system. It needed more data to train the next version of its technology — lots more.

https://www.nytimes.com/2024/04/06/technology/tech-giants-harvest-data-artificial-intelligence.html

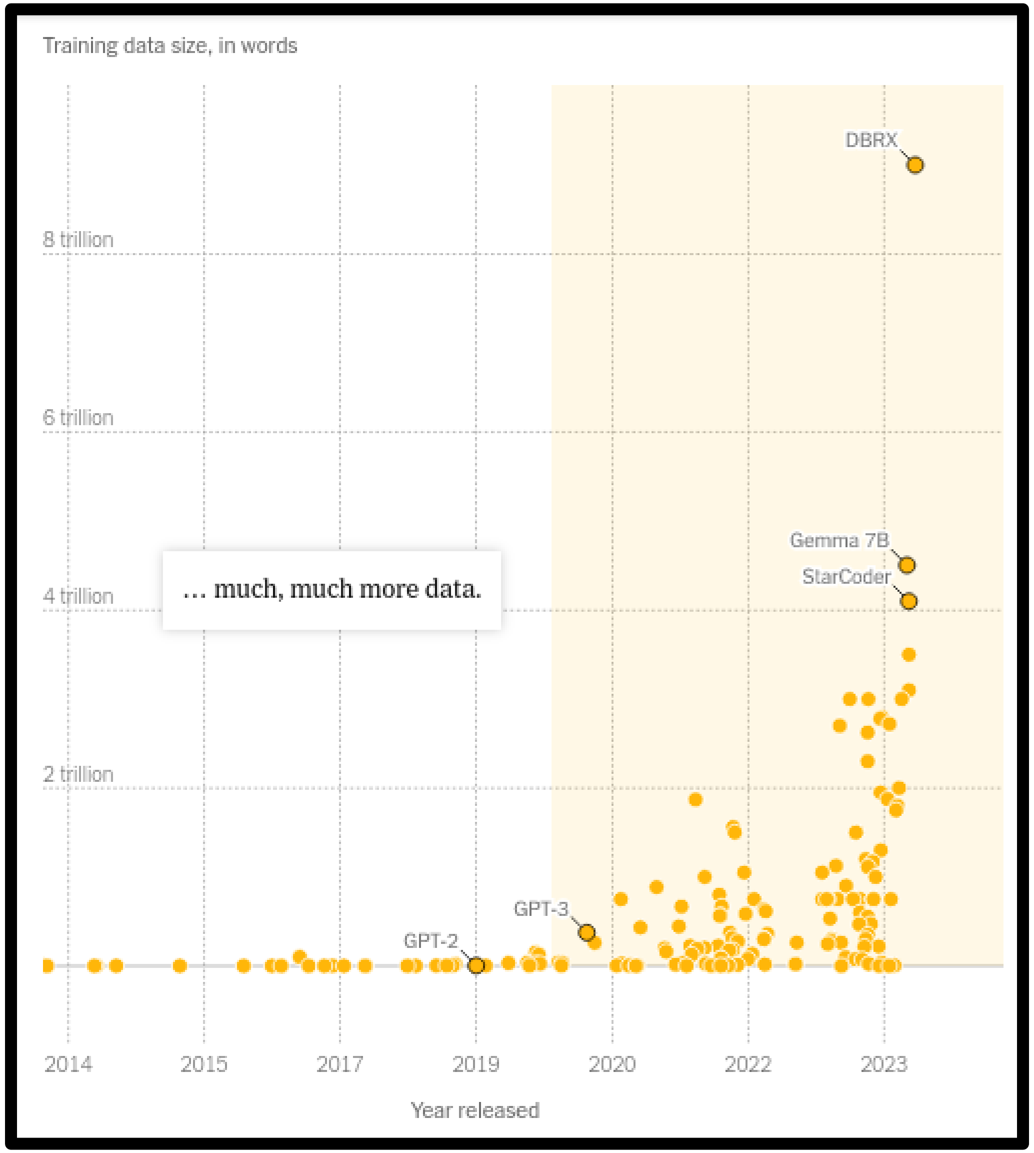
Training Data

The race to lead A.I. has become a desperate hunt for the digital data needed to advance the technology. To obtain that data, tech companies including OpenAI, Google and Meta have cut corners, ignored corporate policies and debated bending the law, according to an examination by The New York Times.

At Meta, which owns Facebook and Instagram, managers, lawyers and engineers last year discussed buying the publishing house Simon & Schuster to procure long works, according to recordings of internal meetings obtained by The Times. They also conferred on gathering copyrighted data from across the internet, even if that meant facing lawsuits. Negotiating licenses with publishers, artists, musicians and the news industry would take too long, they said.

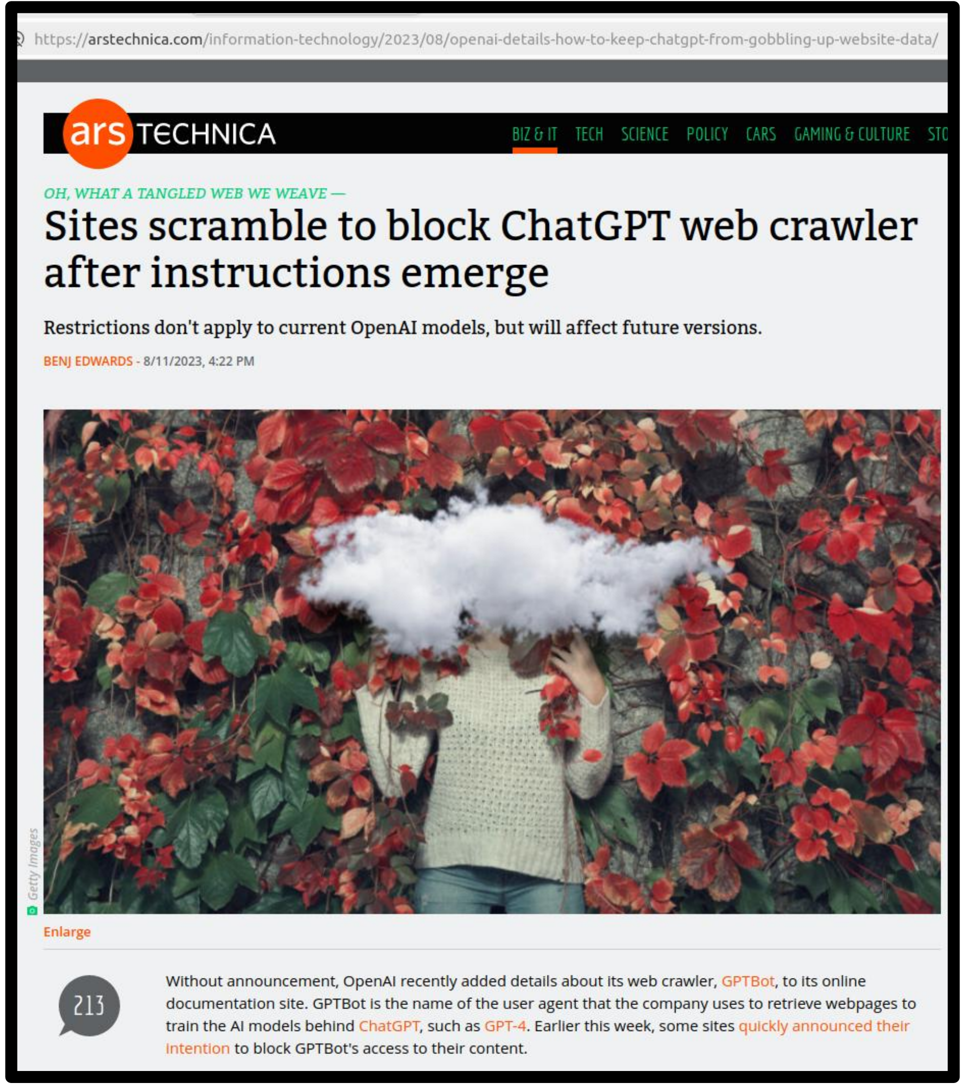
Like OpenAI, Google transcribed YouTube videos to harvest text for its A.I. models, five people with knowledge of the company's practices said. That potentially violated the copyrights to the videos, which belong to their creators.

Last year, Google also broadened its terms of service. One motivation for the change, according to members of the company's privacy team and an internal message viewed by The Times, was to allow Google to be able to tap publicly available Google Docs, restaurant reviews on Google Maps and other online material for more of its A.I. products.



https://www.nytimes.com/2024/04/06/technology/tech-giants-harvest-data-artificial-intelligence.html

Training Data (robots.txt)



https://arstechnica.com/information-technology/2023/08/openai-details-how-to-keep-chatgpt-from-gobbling-up-website-data/

OH, WHAT A TANGLED WEB WE WEAVE —

Sites scramble to block ChatGPT web crawler after instructions emerge

Restrictions don't apply to current OpenAI models, but will affect future versions.

BENJ EDWARDS - 8/11/2023, 4:22 PM

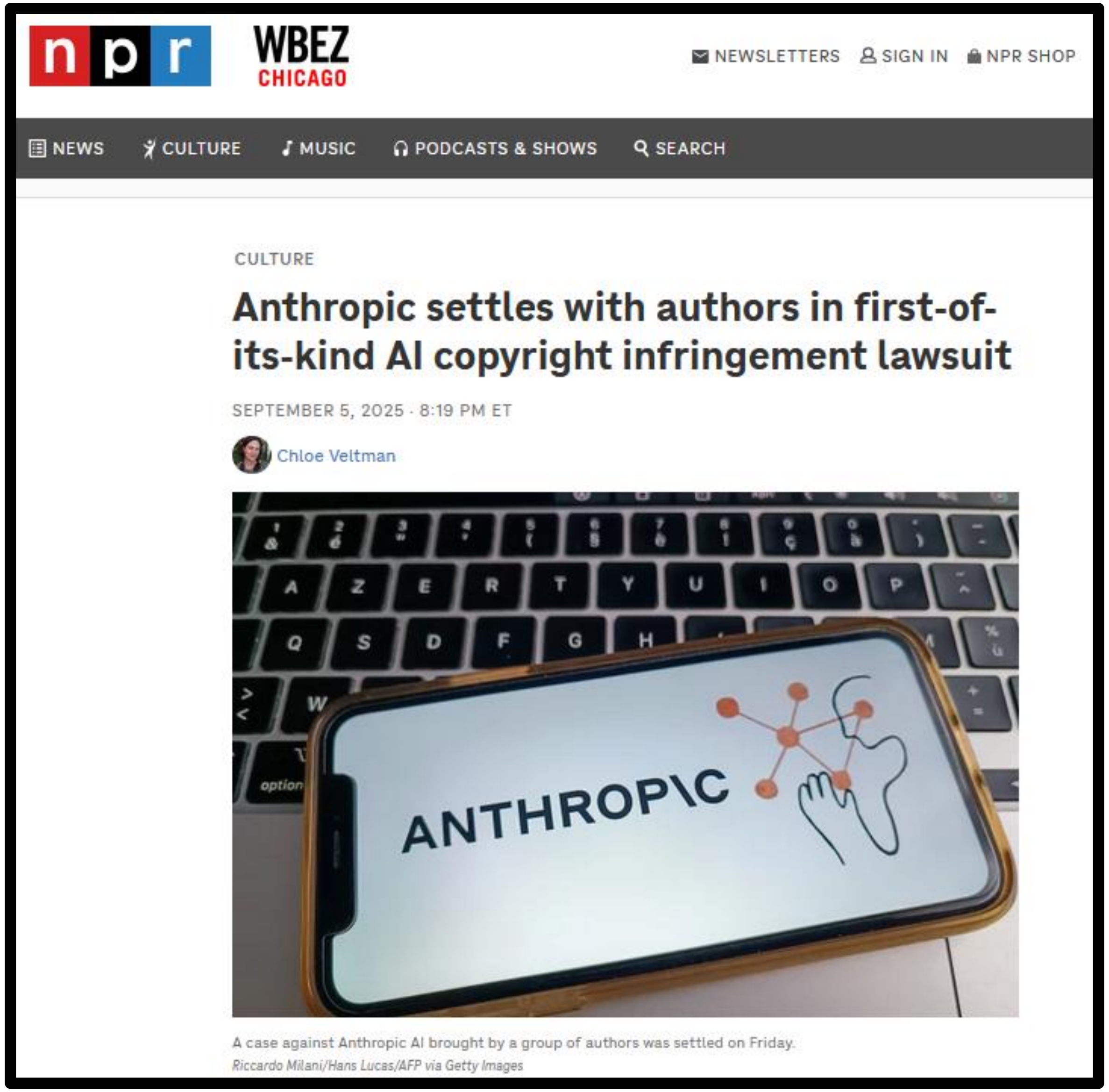
Without announcement, OpenAI recently added details about its web crawler, GPTBot, to its online documentation site. GPTBot is the name of the user agent that the company uses to retrieve webpages to train the AI models behind ChatGPT, such as GPT-4. Earlier this week, some sites quickly announced their intention to block GPTBot's access to their content.

https://arstechnica.com/information-technology/2023/08/openai-details-how-to-keep-chatgpt-from-gobbling-up-website-data/

Training Data

- Do LLM vendors have permission to use this training data? What do you think the process should be for obtaining permission?
- How does a generative AI model’s ability to create new content impact the original human creators of the content used for training?

Training Data



CULTURE

Anthropic settles with authors in first-of-its-kind AI copyright infringement lawsuit

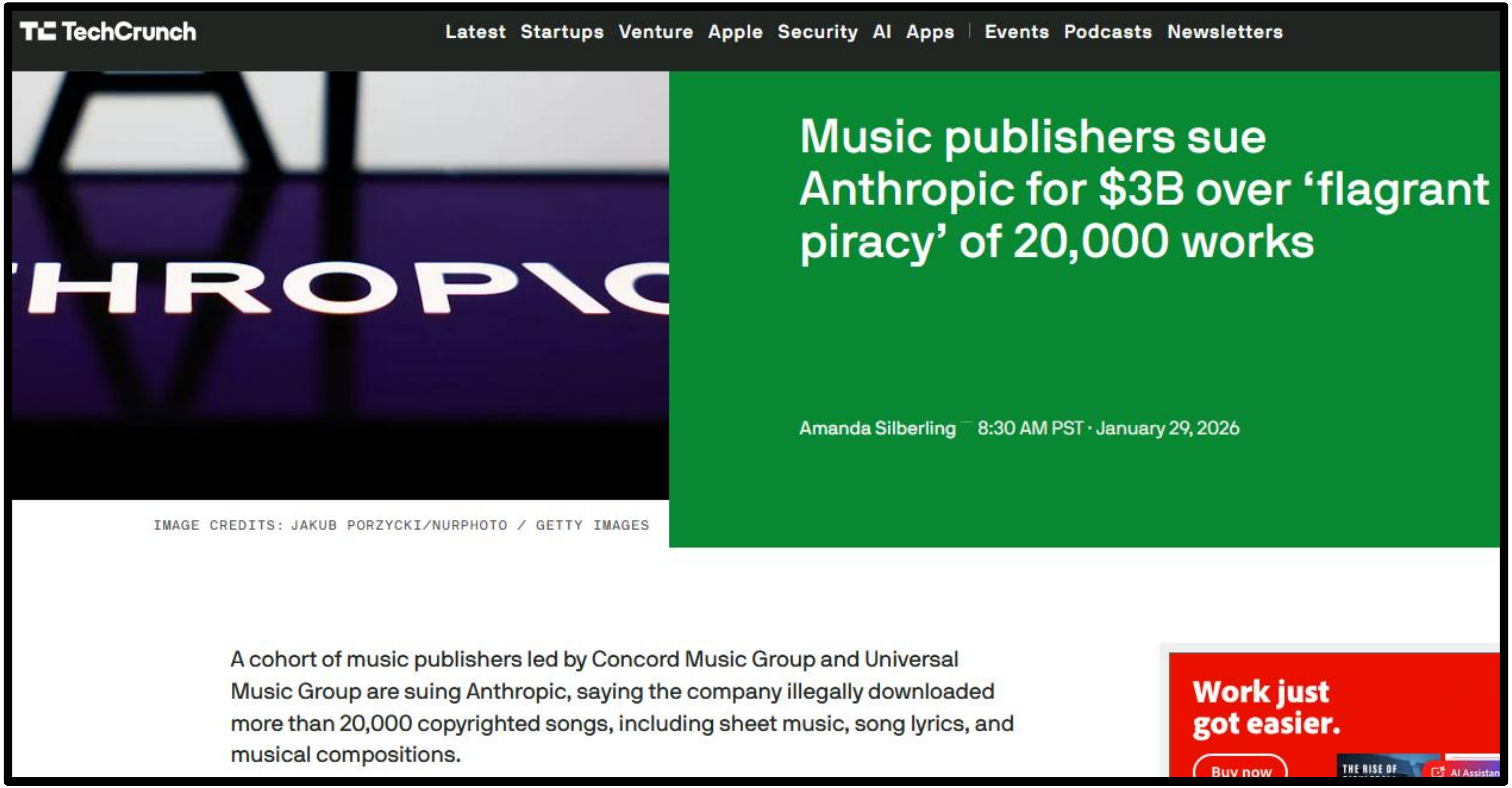
SEPTEMBER 5, 2025 · 8:19 PM ET

Chloe Veltman

A case against Anthropic AI brought by a group of authors was settled on Friday.

Riccardo Milani/Hans Lucas/AFP via Getty Images

Training Data


TechCrunch
Latest Startups Venture Apple Security AI Apps Events Podcasts Newsletters

Music publishers sue Anthropic for $3B over 'flagrant piracy' of 20,000 works

Amanda Silberling 8:30 AM PST · January 29, 2026

IMAGE CREDITS: JAKUB PORZYCKI/NURPHOTO / GETTY IMAGES

A cohort of music publishers led by Concord Music Group and Universal Music Group are suing Anthropic, saying the company illegally downloaded more than 20,000 copyrighted songs, including sheet music, song lyrics, and musical compositions.

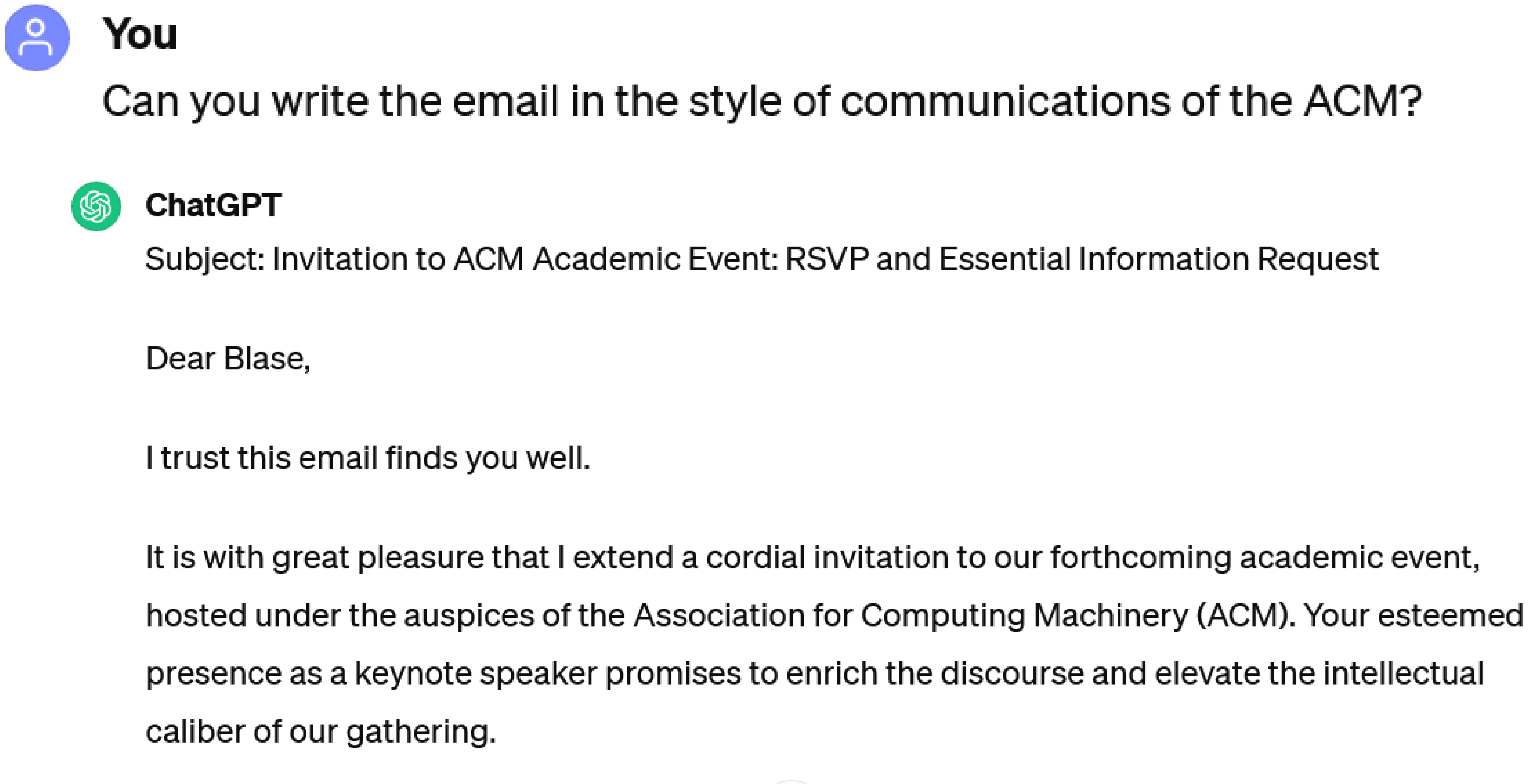
Work just got easier.
Buy now

Fine-Tuning

- After the LLM is trained broadly, it may undergo additional training to be good at specific tasks:
    - Providing “helpful” or safe answers
    - Producing python code
    - Learning how to produce answers in conversational format

- Desire to be helpful and being trained on unsafe data can result in a lot of strange or malicious behavior

Example Impact: LLM as a Spear Phishing Tool



You

Can you write the email in the style of communications of the ACM?

ChatGPT

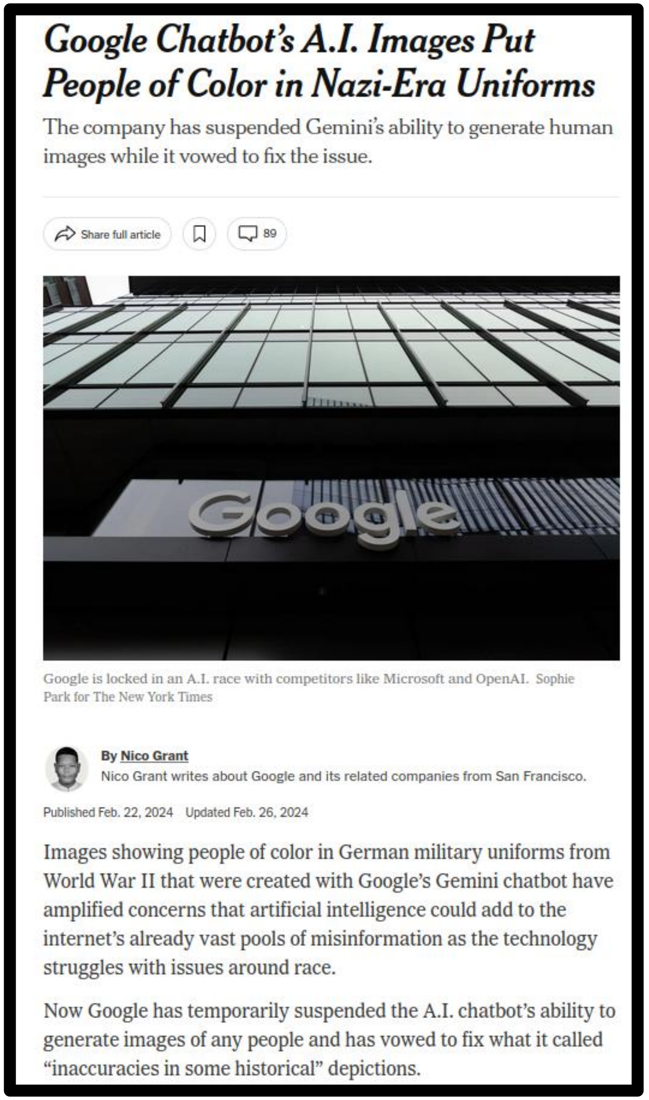
Subject: Invitation to ACM Academic Event: RSVP and Essential Information Request

Dear Blase,

I trust this email finds you well.

It is with great pleasure that I extend a cordial invitation to our forthcoming academic event, hosted under the auspices of the Association for Computing Machinery (ACM). Your esteemed presence as a keynote speaker promises to enrich the discourse and elevate the intellectual caliber of our gathering.

Misguided Attempts At Guardrails



Google Chatbot's A.I. Images Put People of Color in Nazi-Era Uniforms

The company has suspended Gemini's ability to generate human images while it vowed to fix the issue.

Google is locked in an A.I. race with competitors like Microsoft and OpenAI. Sophie Park for The New York Times

By **Nico Grant**
Nico Grant writes about Google and its related companies from San Francisco.

Published Feb. 22, 2024 Updated Feb. 26, 2024

Images showing people of color in German military uniforms from World War II that were created with Google's Gemini chatbot have amplified concerns that artificial intelligence could add to the internet's already vast pools of misinformation as the technology struggles with issues around race.

Now Google has temporarily suspended the A.I. chatbot's ability to generate images of any people and has vowed to fix what it called "inaccuracies in some historical" depictions.

https://www.nytimes.com/2024/02/22/technology/google-gemini-german-uniforms.html

The Promise
and Pitfalls
of Generative AI

Your Experiences Generating Problematic Content With AI

Your Experiences Generating a “Creative” Work With AI