Introduction to GenAI and Systems for GenAI

August 28th, 2025

Jae-Won Chung

Jae-Won Chung

- Bio
  - 5th year PhD student working with Mosharaf
  - https://jaewonchung.me/about
- Background
  - 2017 – 2019: Machine learning, Computer vision, Meta-learning, Few-shot learning
  - 2019 – now: Systems for machine learning, Power & energy as first-class systems resources
- Three lectures
  - 08/29 (Thu) | Introduction to GenAI and Systems for GenAI
  - 09/02 (Tue) | Training basics
  - 10/02 (Thu) | Inference basics

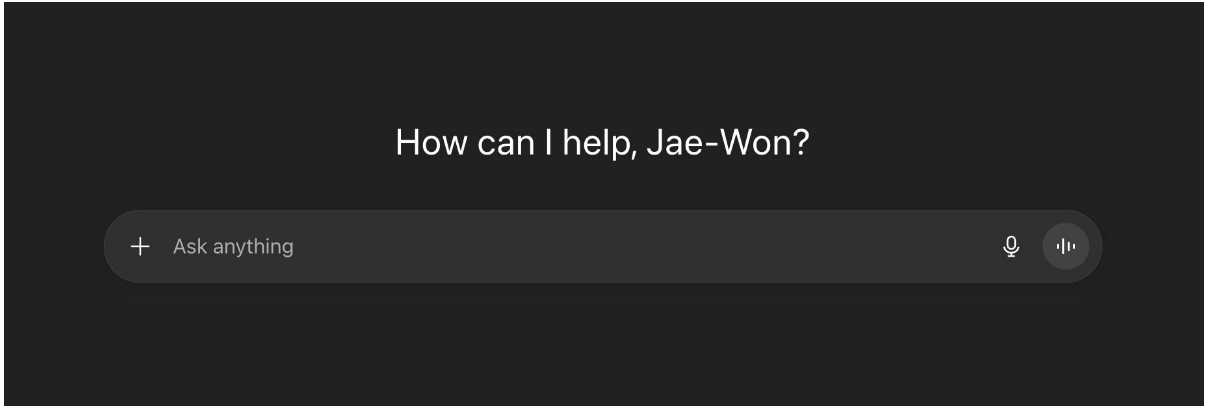
(Unofficial) Textbooks

- How to Scale Your Model (Google DeepMind)
  - Theoretical analysis of computations that are important to LLMs
  - Arithmetic intensity, compute- and memory-bound, roofline, back-of-the-envelope estimations
- The Ultra-Scale Playbook (Hugging Face)
  - Training LLMs on GPU Clusters
  - Various types of training parallelisms, implications on compute, memory, and communication
- Use as references
  - Each thing will take at least a week of full-time reading (it’s worth it, though)
  - It’s a fast-evolving field; the only way to be relevant is to continuously read

Today

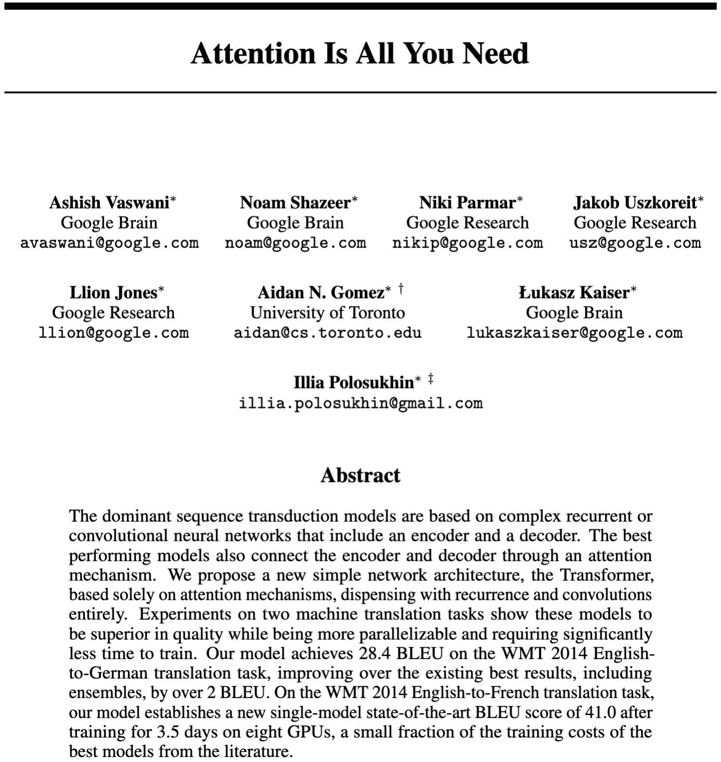
- Overview of GenAI
  - Essential ML-flavored knowledge and terminology
  - Aim to digest end-to-end and please ask questions if you don’t get something
- Overview of Systems for GenAI
  - A high-level landscape of systems challenges
  - This is what you’ll dive more deeply into this semester

How did we end up with this?



How can I help, Jae-Won?

Attention is All You Need (Google, 2017)



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 less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.0 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature.

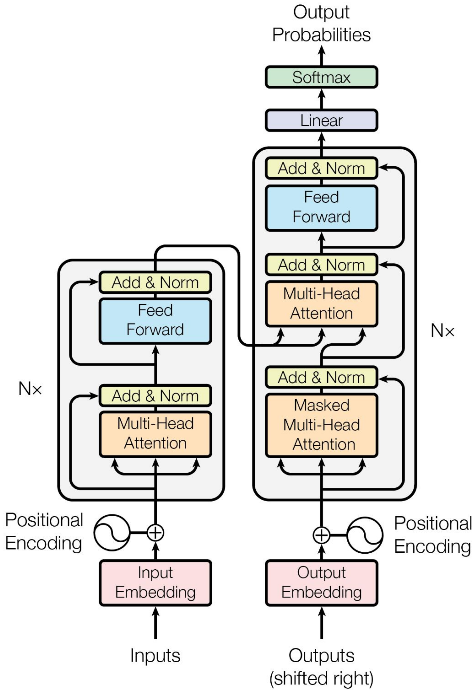
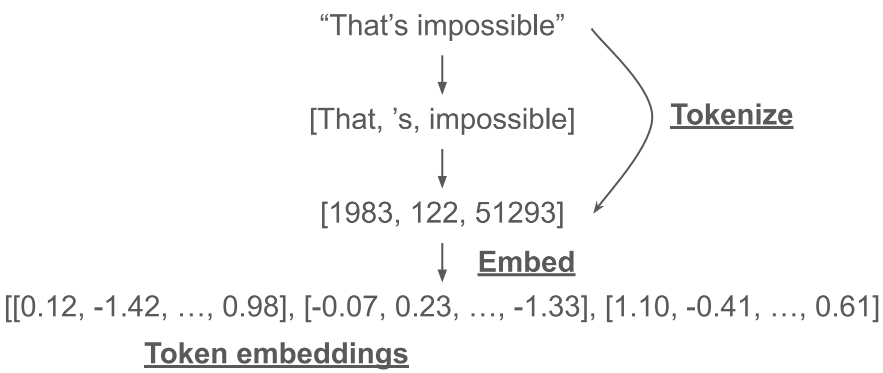


Figure 1: The Transformer - model architecture.

Attention is All You Need (Google, 2017)

- The **Transformer** model
  - It was a machine translation paper
  - Sequence-to-sequence learning
- Critical components
  - Tokenization & embedding
  - Attention
  - MLP



"That's impossible"

[That, 's, impossible]

[1983, 122, 51293]

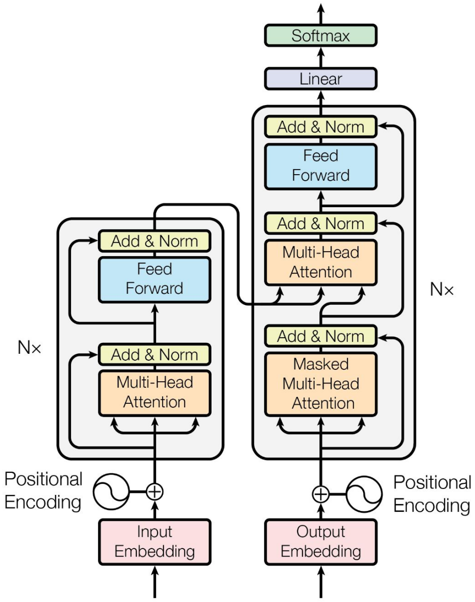
Tokenize

Embed

[[0.12, -1.42, …, 0.98], [-0.07, 0.23, …, -1.33], [1.10, -0.41, …, 0.61]]

Token embeddings

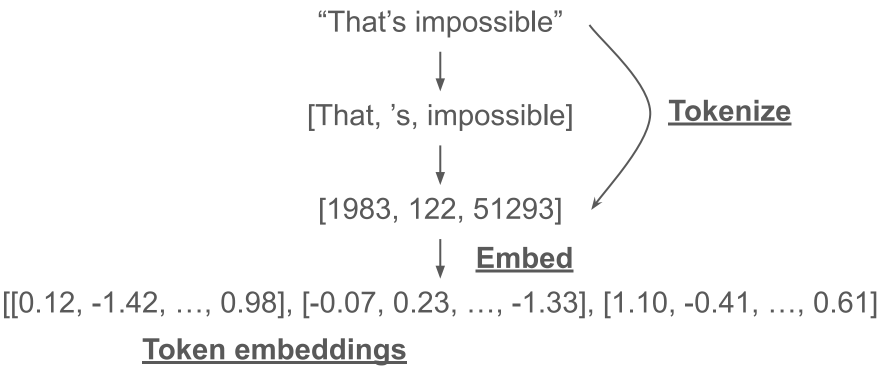
"그건 불가능해"



"That's impossible"

Attention is All You Need (Google, 2017)

- **Autoregressive** text generation
  - One forward pass produces one output token
  - Keep appending output token to input
  - Stop when the special EOS token is produced



“That’s impossible”

[That, ’s, impossible]

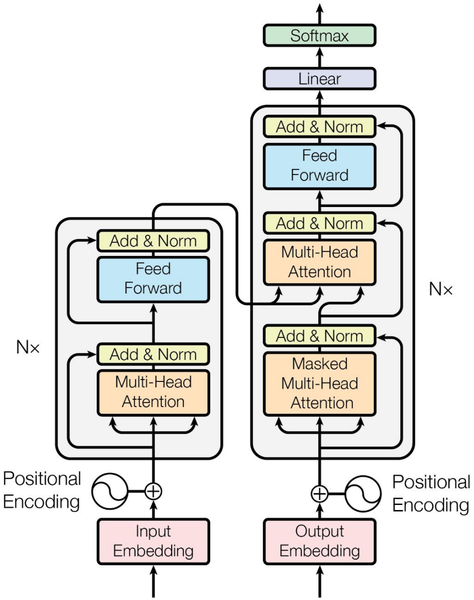
[1983, 122, 51293]

Tokenize

Embed

[[0.12, -1.42, …, 0.98], [-0.07, 0.23, …, -1.33], [1.10, -0.41, …, 0.61]]

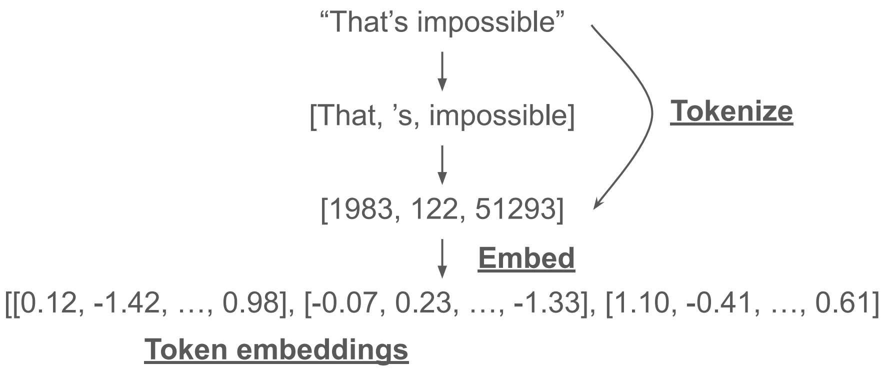
Token embeddings



[That, ’s, impossible]   [BOS]

Attention is All You Need (Google, 2017)

- **Autoregressive** text generation
    - One forward pass produces one output token
    - Keep appending output token to input
    - Stop when the special EOS token is produced



"That's impossible"

[That, 's, impossible]

[1983, 122, 51293]

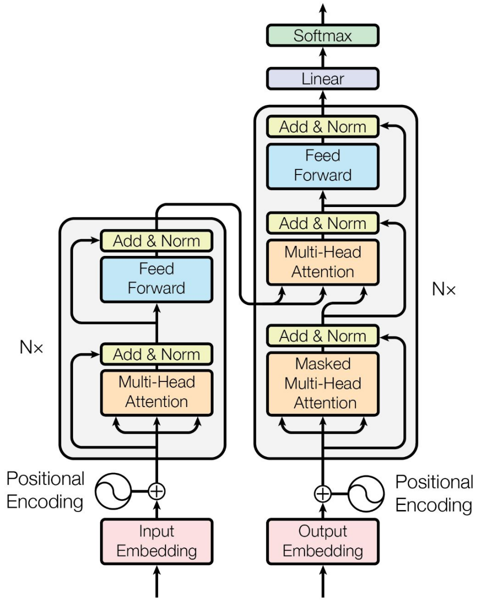
Tokenize

Embed

[[0.12, -1.42, …, 0.98], [-0.07, 0.23, …, -1.33], [1.10, -0.41, …, 0.61]]

Token embeddings

[ユ]



[That, 's, impossible] [BOS]

Attention is All You Need (Google, 2017)

- **Autoregressive** text generation
  - One forward pass produces one output token
  - Keep appending output token to input
  - Stop when the special EOS token is produced

“That’s impossible”

↓

[That, ’s, impossible]

↓

[1983, 122, 51293]

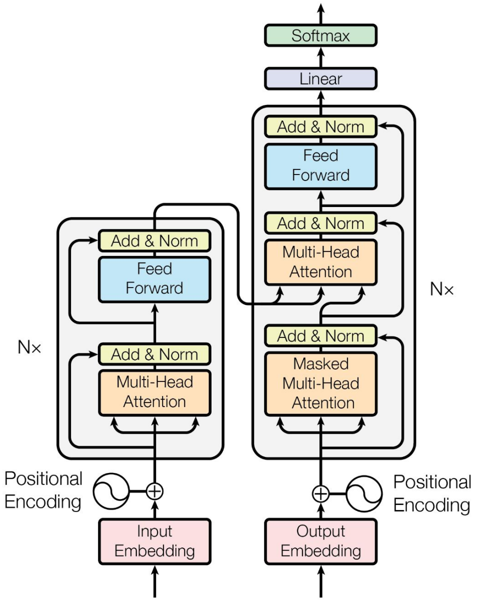
Tokenize

↓ **Embed**

[[0.12, -1.42, …, 0.98], [-0.07, 0.23, …, -1.33], [1.10, -0.41, …, 0.61]]

Token embeddings

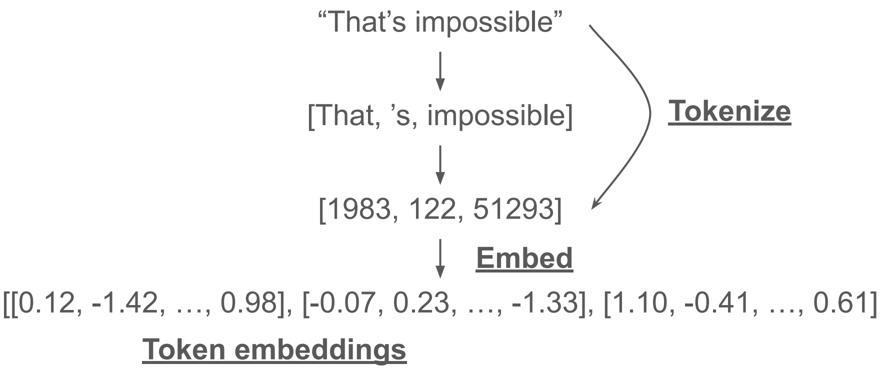
[그]



[That, ’s, impossible] [BOS, 그]

Attention is All You Need (Google, 2017)

- **Autoregressive** text generation
  - One forward pass produces one output token
  - Keep appending output token to input
  - Stop when the special EOS token is produced



"That's impossible"

[That, 's, impossible]

[1983, 122, 51293]

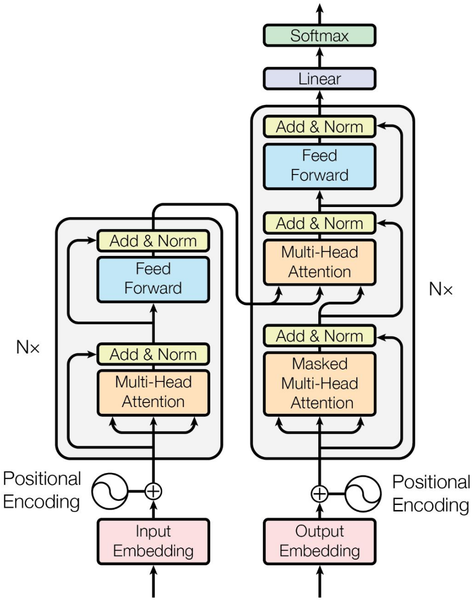
Tokenize

Embed

[[0.12, -1.42, …, 0.98], [-0.07, 0.23, …, -1.33], [1.10, -0.41, …, 0.61]]

Token embeddings

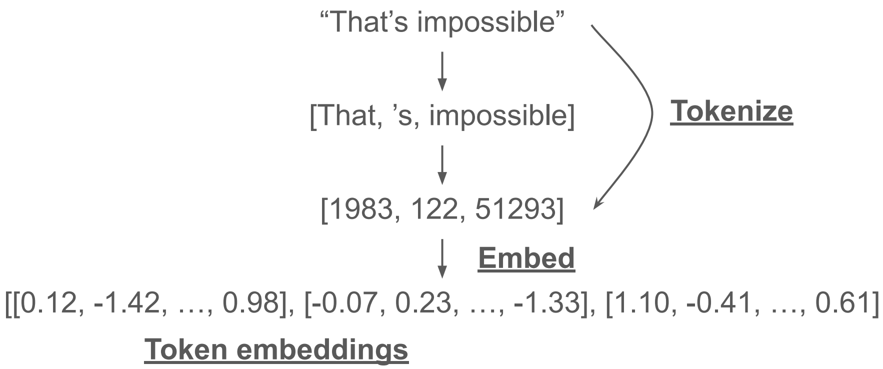
[그, 건]



[That, 's, impossible]   [BOS, 그]

Attention is All You Need (Google, 2017)

- **Autoregressive** text generation
  - One forward pass produces one output token
  - Keep appending output token to input
  - Stop when the special EOS token is produced



"That's impossible"

[That, 's, impossible]

[1983, 122, 51293]

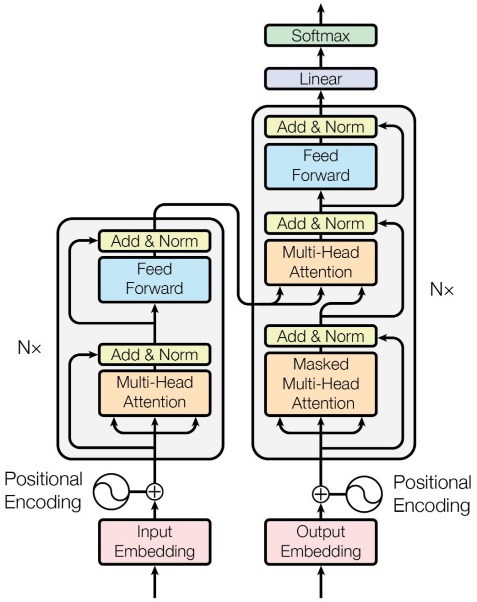
Tokenize

Embed

[[0.12, -1.42, …, 0.98], [-0.07, 0.23, …, -1.33], [1.10, -0.41, …, 0.61]]

Token embeddings

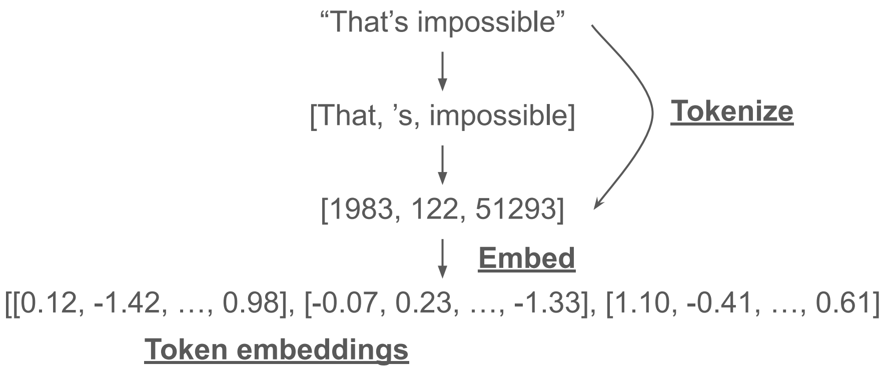
[그, 건]



[That, 's, impossible] [BOS, 그, 건]

Attention is All You Need (Google, 2017)

- **Autoregressive** text generation
  - One forward pass produces one output token
  - Keep appending output token to input
  - Stop when the special EOS token is produced



“That’s impossible”

[That, ’s, impossible]

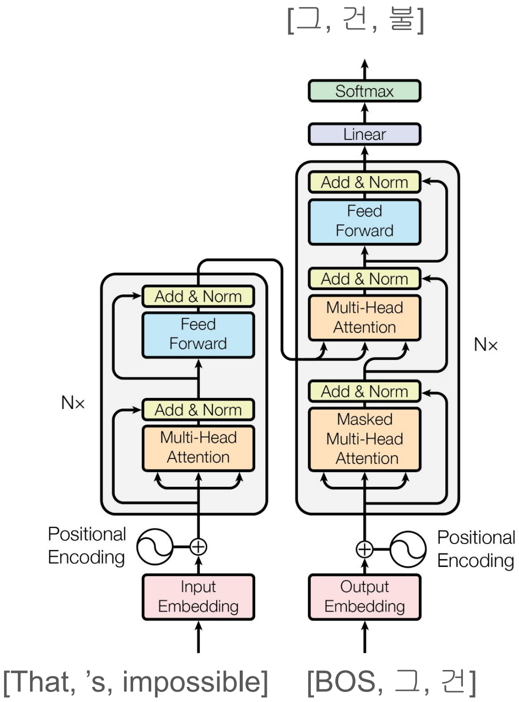
[1983, 122, 51293]

Tokenize

Embed

[[0.12, -1.42, …, 0.98], [-0.07, 0.23, …, -1.33], [1.10, -0.41, …, 0.61]]

Token embeddings



[그, 건, 불]

[That, ’s, impossible] [BOS, 그, 건]

Attention is All You Need (Google, 2017)

- **Autoregressive** text generation
  - One forward pass produces one output token
  - Keep appending output token to input
  - Stop when the special EOS token is produced

"That's impossible"

↓

[That, 's, impossible]

↓

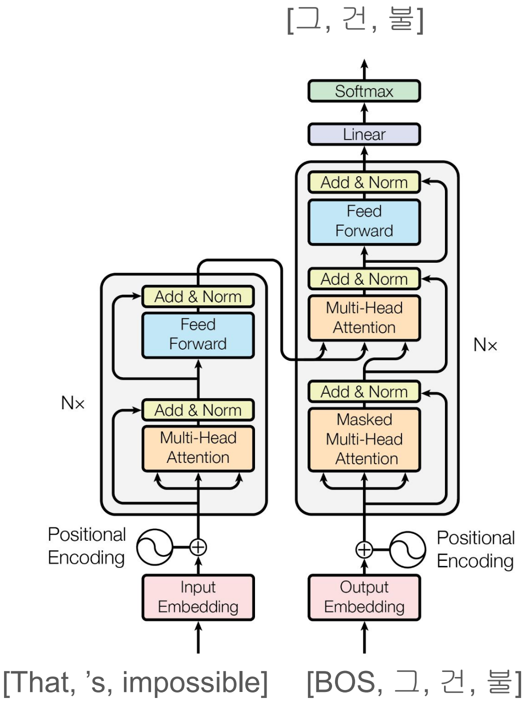
[1983, 122, 51293]

Tokenize

↓ **Embed**

[[0.12, -1.42, …, 0.98], [-0.07, 0.23, …, -1.33], [1.10, -0.41, …, 0.61]]

Token embeddings

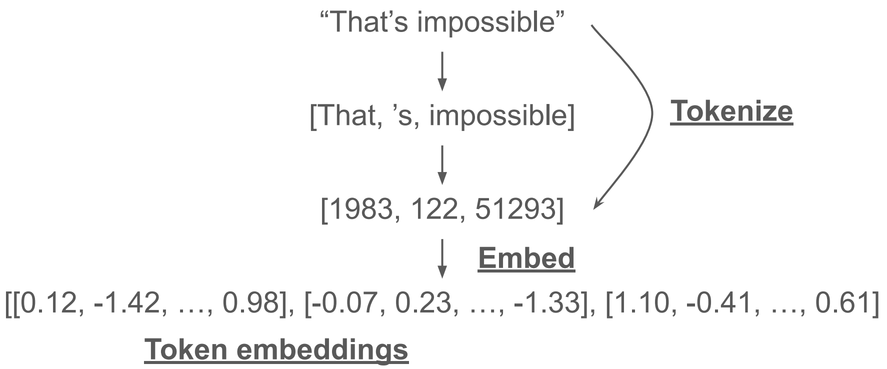


[그, 건, 불]

[That, 's, impossible] [BOS, 그, 건, 불]

Attention is All You Need (Google, 2017)

- **Autoregressive** text generation
  - One forward pass produces one output token
  - Keep appending output token to input
  - Stop when the special EOS token is produced



“That’s impossible”

[That, ’s, impossible]

[1983, 122, 51293]

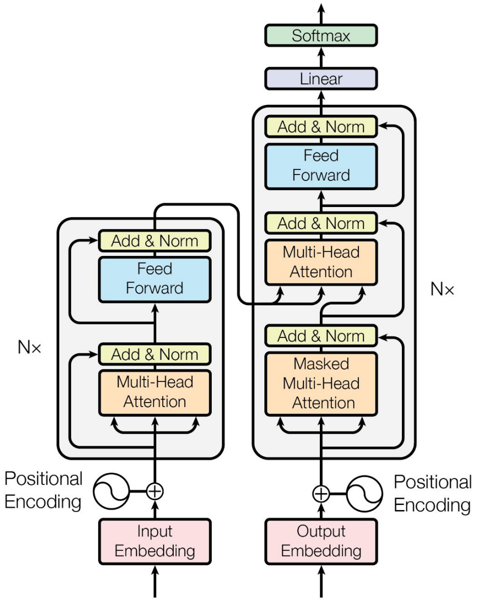
Tokenize

Embed

[[0.12, -1.42, …, 0.98], [-0.07, 0.23, …, -1.33], [1.10, -0.41, …, 0.61]]

Token embeddings

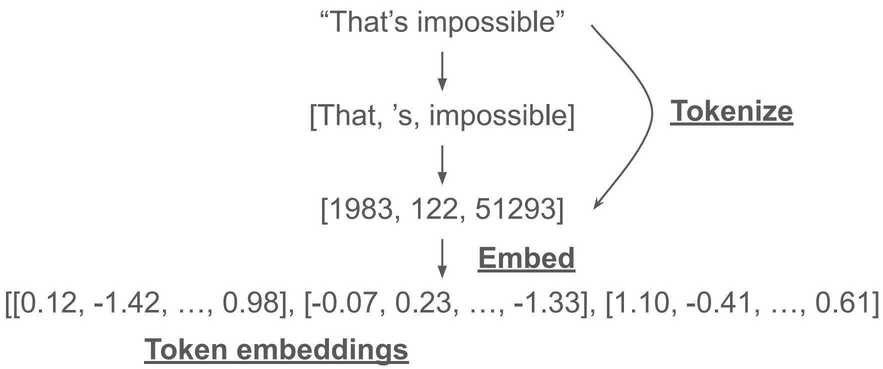
[그, 건, 불, 가능]



[That, ’s, impossible]   [BOS, 그, 건, 불]

Attention is All You Need (Google, 2017)

- **Autoregressive** text generation
  - One forward pass produces one output token
  - Keep appending output token to input
  - Stop when the special EOS token is produced



“That’s impossible”

[That, ’s, impossible]

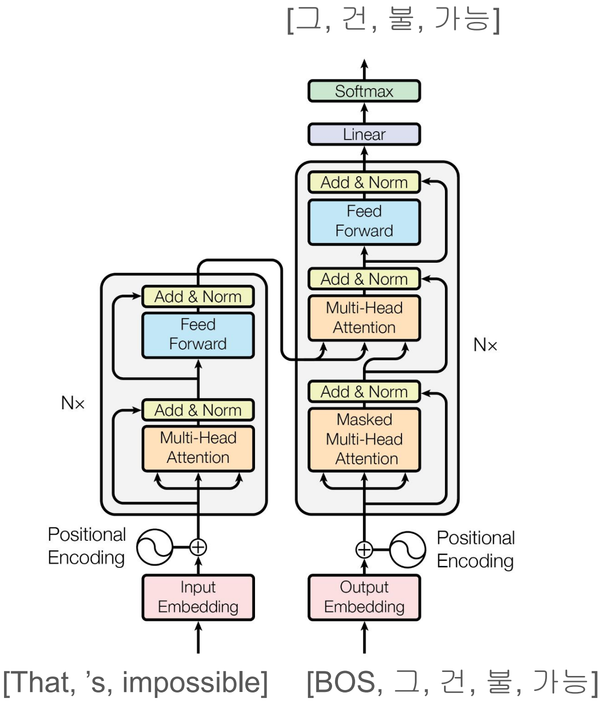
[1983, 122, 51293]

Tokenize

Embed

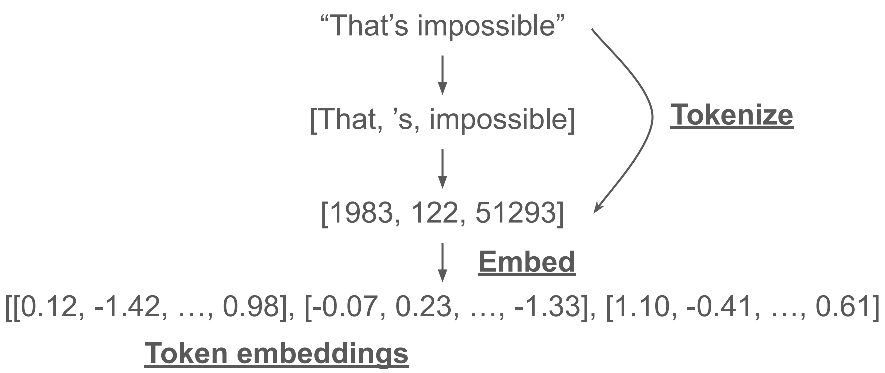
[[0.12, -1.42, …, 0.98], [-0.07, 0.23, …, -1.33], [1.10, -0.41, …, 0.61]]

Token embeddings



Attention is All You Need (Google, 2017)

- **Autoregressive** text generation
  - One forward pass produces one output token
  - Keep appending output token to input
  - Stop when the special EOS token is produced



"That's impossible"

[That, 's, impossible]

[1983, 122, 51293]

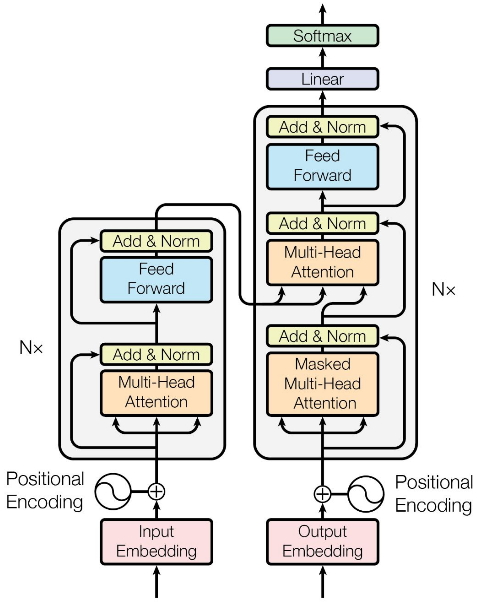
Tokenize

Embed

[[0.12, -1.42, …, 0.98], [-0.07, 0.23, …, -1.33], [1.10, -0.41, …, 0.61]]

Token embeddings

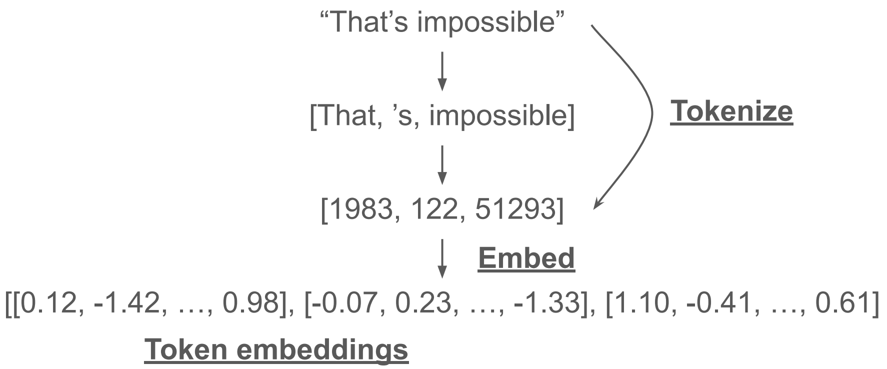
[그, 건, 불, 가능, 해]



[That, 's, impossible]   [BOS, 그, 건, 불, 가능]

Attention is All You Need (Google, 2017)

- **Autoregressive** text generation
  - One forward pass produces one output token
  - Keep appending output token to input
  - Stop when the special EOS token is produced



“That’s impossible”

[That, ’s, impossible]

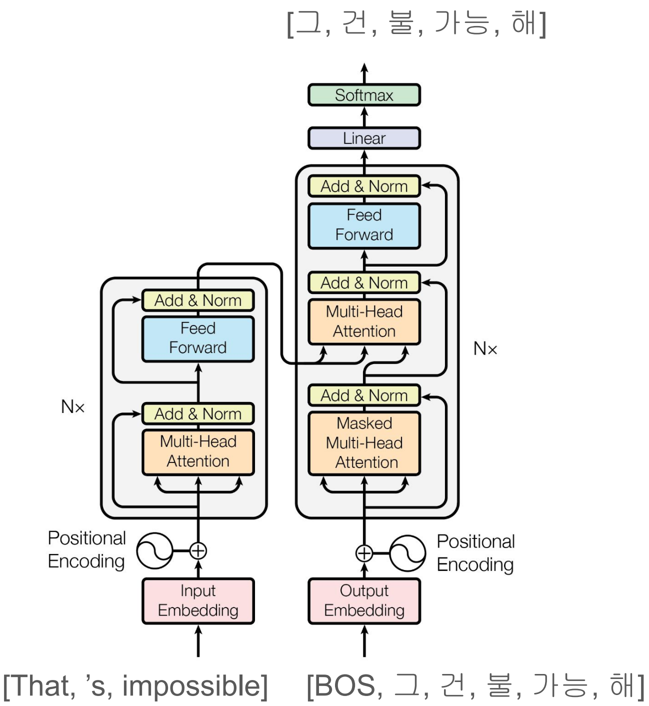
[1983, 122, 51293]

Tokenize

Embed

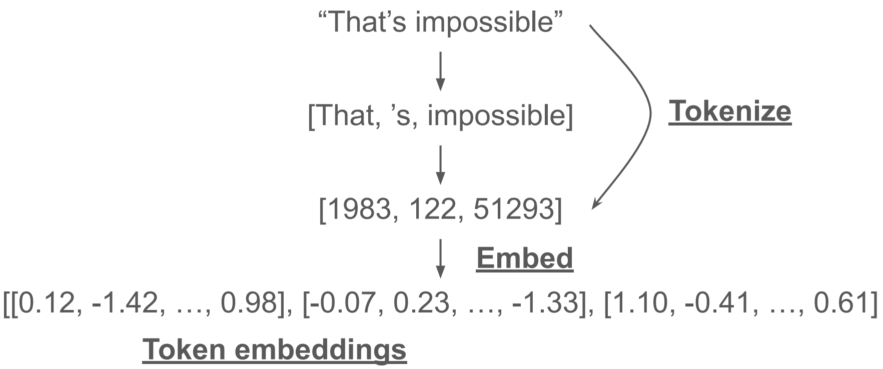
[[0.12, -1.42, …, 0.98], [-0.07, 0.23, …, -1.33], [1.10, -0.41, …, 0.61]]

Token embeddings



Attention is All You Need (Google, 2017)

- **Autoregressive** text generation
  - One forward pass produces one output token
  - Keep appending output token to input
  - Stop when the special EOS token is produced



"That's impossible"

[That, 's, impossible]

[1983, 122, 51293]

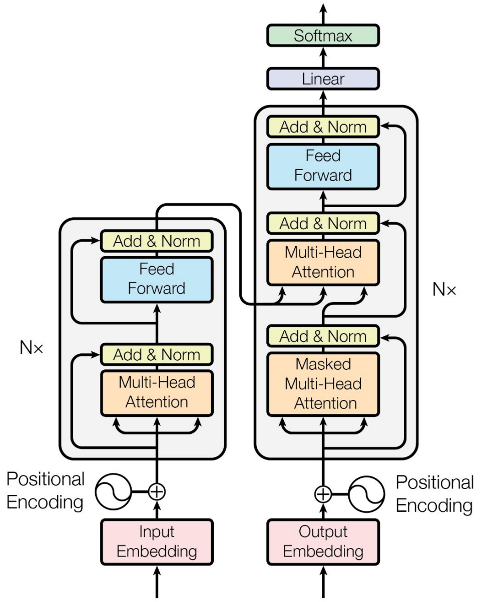
Tokenize

Embed

[[0.12, -1.42, …, 0.98], [-0.07, 0.23, …, -1.33], [1.10, -0.41, …, 0.61]]

Token embeddings

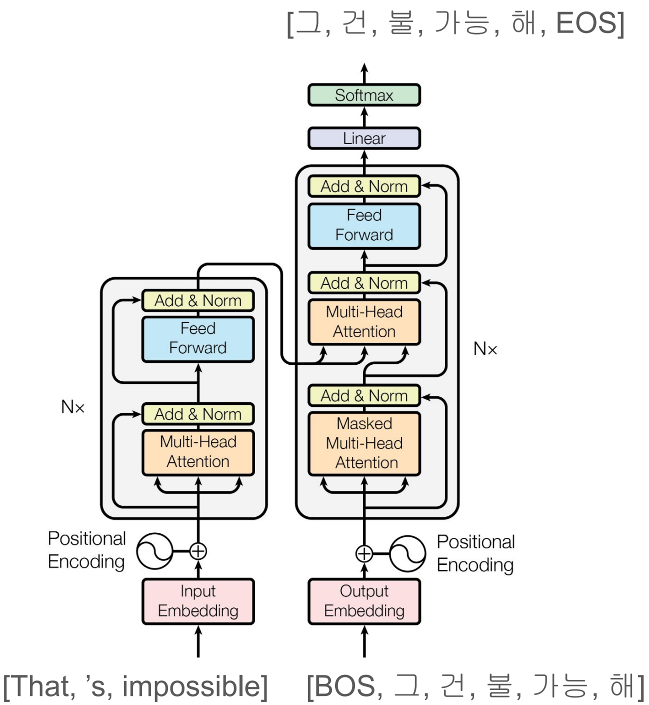
[그, 건, 불, 가능, 해, EOS]



[That, 's, impossible] [BOS, 그, 건, 불, 가능, 해]

Attention is All You Need (Google, 2017)

- The **<u>Attention</u>** mechanism
  - Details deferred to [The Illustrated Transformer]()
  - Importance-weighted sum over token embeddings
  - Two types
    - **<u>Causal attention (decoder)</u>**
      - Token *i* aggregates over *previous* tokens
    - Bidirectional attention (encoder)
      - Token *i* aggregates over *every* token



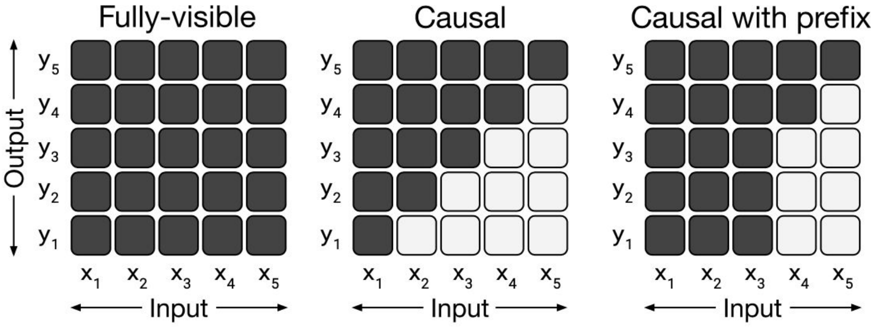
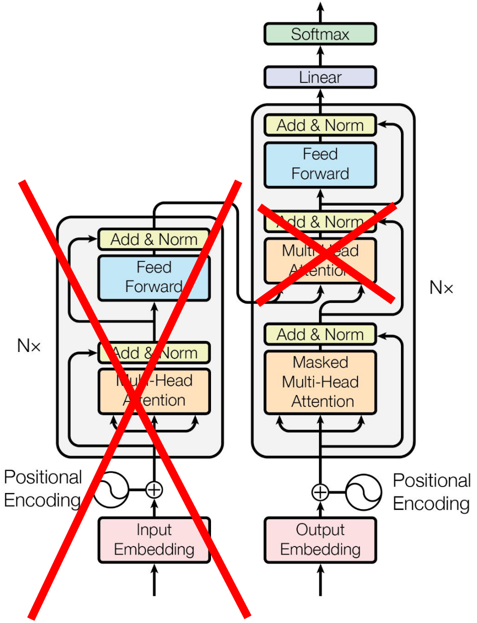


Image credit: Exploring the limits of transfer learning with a unified text-to-text transformer, 2020

GPT-1 (OpenAI, 2018)

- **Generative Pre-trained Transformer**
  - Improving language understanding with unsupervised learning (OpenAI blog post)
  - An **autoregressive text generation** model
  - 117M parameters, 5GB of text



Decoder-only Transformer

GPT (OpenAI, 2018)

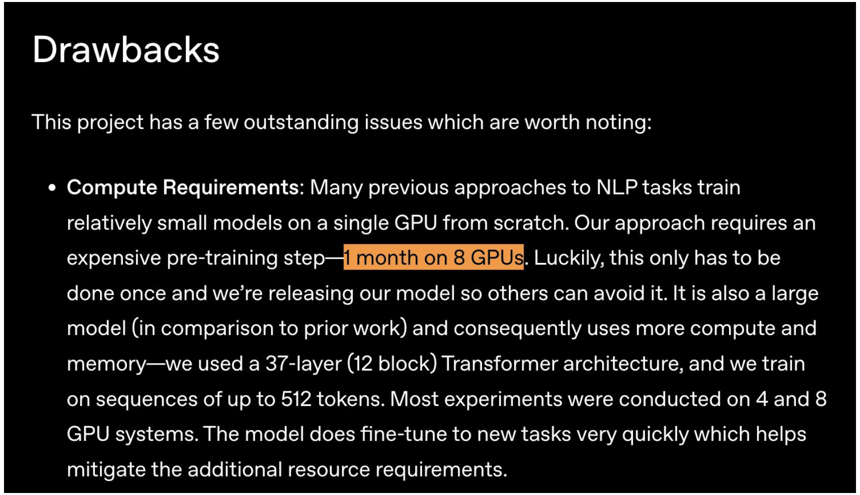
- A new paradigm
  - First, **pre-train** a large model on a large unlabeled corpus of text
  - Then, **fine-tune** the model on a *small* labeled task-specific dataset (e.g., translation)
  - The traditional way was to figure out how to build a *large* task dataset and train on it
  - Now, you can pre-train a large model once and reuse it for many downstream tasks

Next Token Prediction

- Attention is all you → ?
- Large language → ?
- University of California, San → ?
- This is the best steak I’ve ever had in my → ?

GPT (OpenAI, 2018)

- A new paradigm
  - First, **pre-train** a large model on a large unlabeled corpus of text
  - Then, **fine-tune** the model on a small labeled task-specific dataset (e.g., translation)
- Systems challenge?



Drawbacks

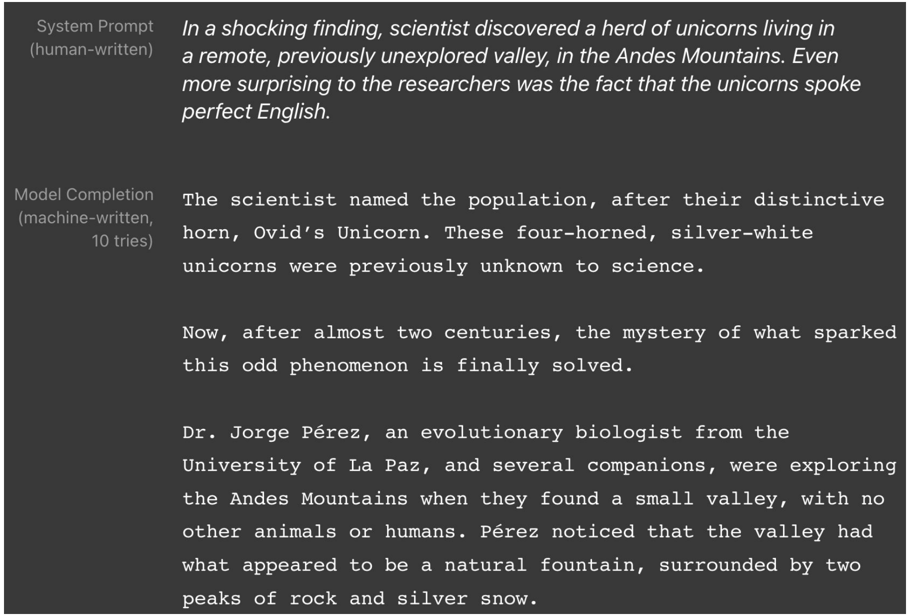
This project has a few outstanding issues which are worth noting:

- **Compute Requirements**: Many previous approaches to NLP tasks train relatively small models on a single GPU from scratch. Our approach requires an expensive pre-training step—1 month on 8 GPUs. Luckily, this only has to be done once and we're releasing our model so others can avoid it. It is also a large model (in comparison to prior work) and consequently uses more compute and memory—we used a 37-layer (12 block) Transformer architecture, and we train on sequences of up to 512 tokens. Most experiments were conducted on 4 and 8 GPU systems. The model does fine-tune to new tasks very quickly which helps mitigate the additional resource requirements.

5,760 GPU hours!

GPT-2 (OpenAI, 2019)

- Successor to GPT
  - Better language models and their implications (OpenAI blog)
  - 1.5B parameters (12.8x), 40GB of Internet text (8x)



System Prompt (human-written)

In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

Model Completion (machine-written, 10 tries)

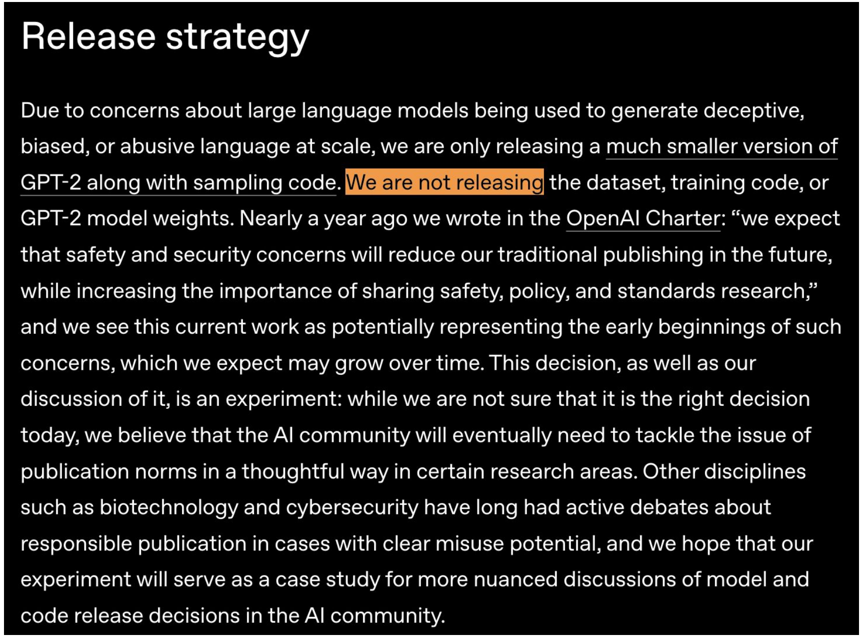
The scientist named the population, after their distinctive horn, Ovid’s Unicorn. These four-horned, silver-white unicorns were previously unknown to science.

Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.

Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

GPT-2 (OpenAI, 2019)

- Safety concerns and the closed model strategy



Release strategy

Due to concerns about large language models being used to generate deceptive, biased, or abusive language at scale, we are only releasing a much smaller version of GPT-2 along with sampling code. We are not releasing the dataset, training code, or GPT-2 model weights. Nearly a year ago we wrote in the OpenAI Charter: “we expect that safety and security concerns will reduce our traditional publishing in the future, while increasing the importance of sharing safety, policy, and standards research,” and we see this current work as potentially representing the early beginnings of such concerns, which we expect may grow over time. This decision, as well as our discussion of it, is an experiment: while we are not sure that it is the right decision today, we believe that the AI community will eventually need to tackle the issue of publication norms in a thoughtful way in certain research areas. Other disciplines such as biotechnology and cybersecurity have long had active debates about responsible publication in cases with clear misuse potential, and we hope that our experiment will serve as a case study for more nuanced discussions of model and code release decisions in the AI community.

Scaling Laws (OpenAI, Kaplan et al., Jan 2020)

- GPT-1 to GPT-2
  - Model parameters: 117M → 1.5B (12.8x)
  - Data: 5 GB → 40 GB (8x)
- Questions
  - Should we continue scaling the number of parameters and the dataset?
  - What’s the best ratio between scaling model parameters and the dataset?
    - Equivalently, given the same amount of target compute,
      what’s the (#params, data size) pair that leads to the best model quality?

Scaling Laws (OpenAI, Kaplan et al., Jan 2020)

- Questions
  - Should we continue scaling the number of parameters and the dataset?
  - What’s the best ratio between scaling model parameters and the dataset?
- “Let’s do a grid search and see what happens.”
  - Scaling Laws for Neural Language Models (OpenAI)

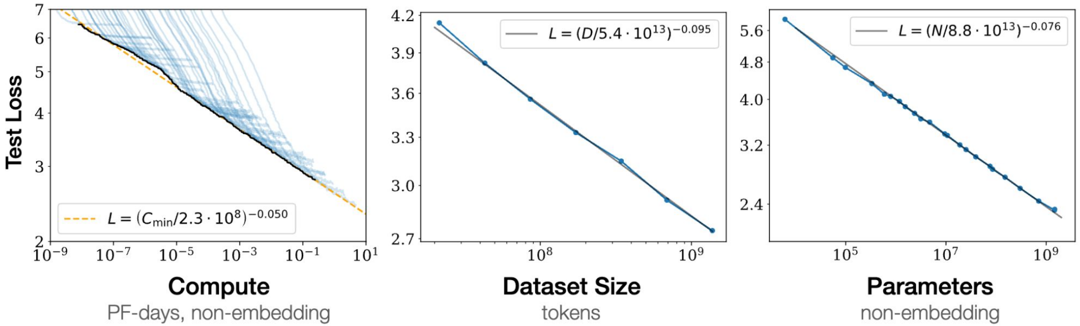


Figure 1 Language modeling performance improves smoothly as we increase the model size, datasetset size, and amount of compute[2] used for training. For optimal performance all three factors must be scaled up in tandem. Empirical performance has a power-law relationship with each individual factor when not bottlenecked by the other two.

Scaling Laws (OpenAI, Kaplan et al., Jan 2020)

- Questions *and Kaplan et al.'s answer*
  - ○ Should we continue scaling the number of parameters and the dataset? *Yeah, probably.*
  - ○ What's the best ratio between scaling model parameters and the dataset? *Roughly equally.*
- "Let's do a grid search and see what happens."
  - ○ Scaling Laws for Neural Language Models (OpenAI)

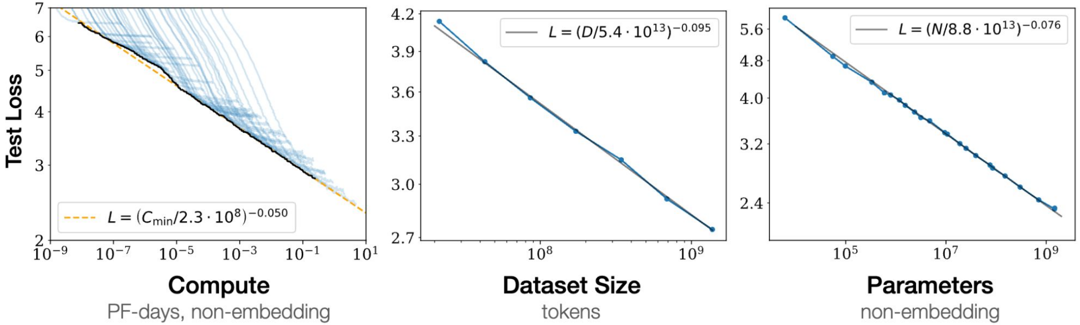


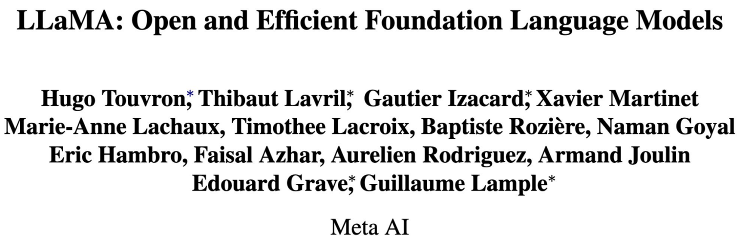
Figure 1 Language modeling performance improves smoothly as we increase the model size, datasetset size, and amount of compute[2] used for training. For optimal performance all three factors must be scaled up in tandem. Empirical performance has a power-law relationship with each individual factor when not bottlenecked by the other two.

Scaling Laws (OpenAI, Jan 2020)

- Two notable progressions after this
  - Chinchilla scaling law (Google DeepMind, Hoffman et al., 2022)
    - You should actually scale data more!

Scaling Laws (OpenAI, Jan 2020)

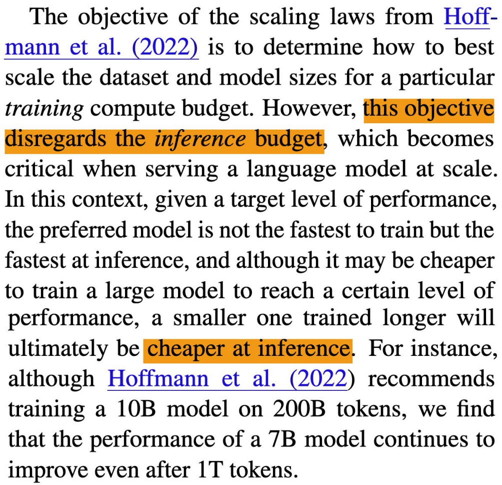
- Two notable progressions after this
  - Chinchilla scaling law (Google DeepMind, Hoffman et al., 2022)
    - You should actually scale data more!
  - Overtraining: Train models on more data than the Chinchilla-optimal point
    - Your training time compute is no longer *optimal* (in Chinchilla sense)
    - You end up with a smaller model with good model quality
    - But smaller models are cheaper to run inference on!



LLaMA: Open and Efficient Foundation Language Models

Hugo Touvron[*], Thibaut Lavril[*], Gautier Izacard[*], Xavier Martinet
Marie-Anne Lachaux, Timothee Lacroix, Baptiste Rozière, Naman Goyal
Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin
Edouard Grave[*], Guillaume Lample[*]

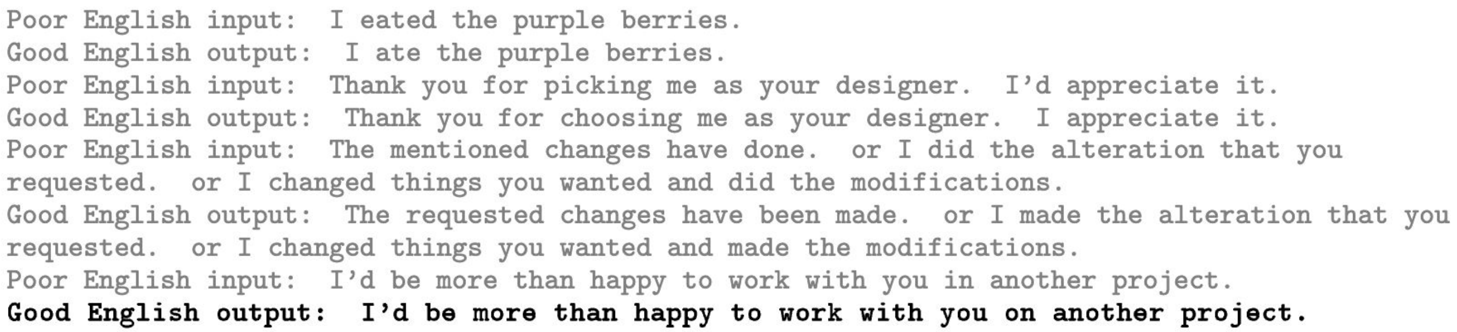
Meta AI



The objective of the scaling laws from Hoffmann et al. (2022) is to determine how to best scale the dataset and model sizes for a particular *training* compute budget. However, this objective disregards the *inference* budget, which becomes critical when serving a language model at scale. In this context, given a target level of performance, the preferred model is not the fastest to train but the fastest at inference, and although it may be cheaper to train a large model to reach a certain level of performance, a smaller one trained longer will ultimately be cheaper at inference. For instance, although Hoffmann et al. (2022) recommends training a 10B model on 200B tokens, we find that the performance of a 7B model continues to improve even after 1T tokens.

GPT-3 (OpenAI, May 2020)

- GPT-3
  - Language Models are Few-Shot Learners (OpenAI paper)
  - 175B parameters, 400B tokens
- A new paradigm
  - Previously: Pre-train on large corpus, and then fine-tune on task-specific dataset.
  - GPT-3: Don't even do the fine-tuning part. Instead, put a **few** examples in the prompt.



```
Poor English input:  I eated the purple berries.
Good English output:  I ate the purple berries.
Poor English input:  Thank you for picking me as your designer.  I'd appreciate it.
Good English output:  Thank you for choosing me as your designer.  I appreciate it.
Poor English input:  The mentioned changes have done.  or I did the alteration that you
requested.  or I changed things you wanted and did the modifications.
Good English output:  The requested changes have been made.  or I made the alteration that you
requested.  or I changed things you wanted and made the modifications.
Poor English input:  I'd be more than happy to work with you in another project.
Good English output:  I'd be more than happy to work with you on another project.
```

Few-shot examples

Actual prompt

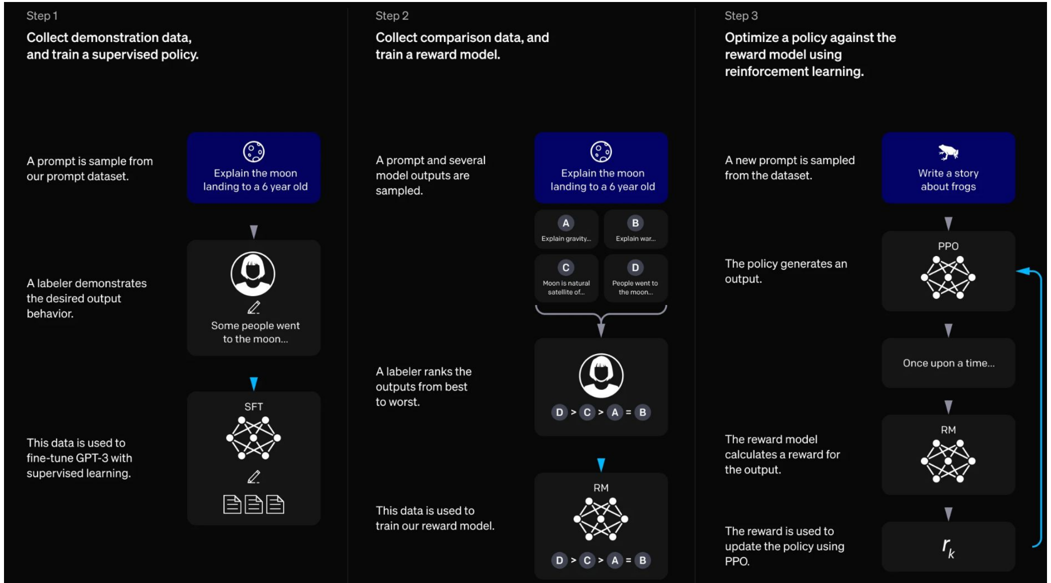
GPT's completion

RLHF (OpenAI, Jan 2022)

- Reinforcement Learning with Human Feedback (RHLF)
  - Aligning language models to follow instructions (OpenAI blog)
- Few-shot prompting works, but it has limitations
  - Hope: We want GPT to do text completion in a super friendly way
  - But nothing in the pre-training corpus really endorses the model to be friendly
  - We do need a step after pre-training to reinforce certain behaviors

RLHF (OpenAI, Jan 2022)

- Three steps
  - Among many GPT responses to the same prompt, a human indicates which one they prefer
  - Train a *reward model* that predicts human preference for a given response
  - Run RL on the LLM using the reward model (**PPO**; Proximal Policy Optimization)



Step 1

Collect demonstration data, and train a supervised policy.

A prompt is sample from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3 with supervised learning.

Step 2

Collect comparison data, and train a reward model.

A prompt and several model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.

Step 3

Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.

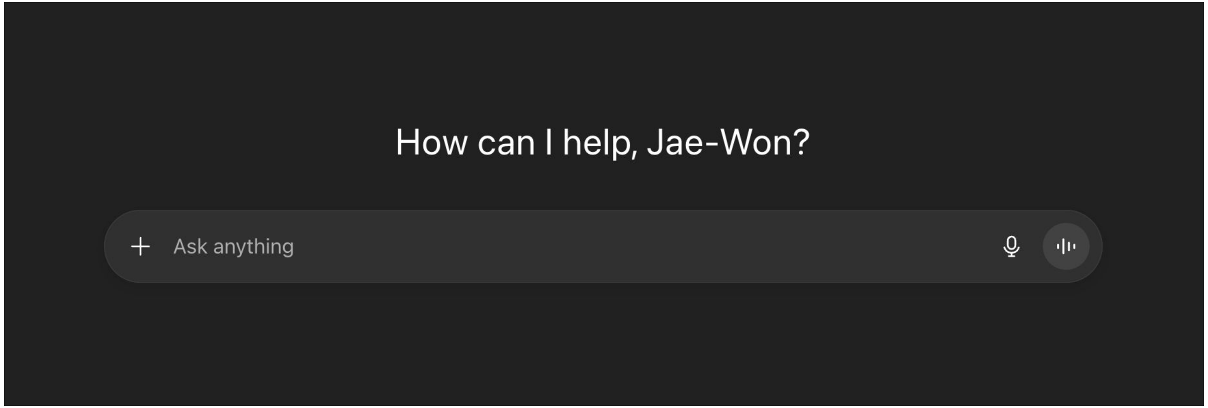
The policy generates an output.

The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.

Explain the moon landing to a 6 year old
Some people went to the moon...
SFT
Explain the moon landing to a 6 year old
A
Explain gravity...
B
Explain war...
C
Moon is natural satellite of...
D
People went to the moon...
D > C > A = B
RM
D > C > A = B
Write a story about frogs
PPO
Once upon a time...
RM
r_k

ChatGPT (OpenAI, Nov 2022)

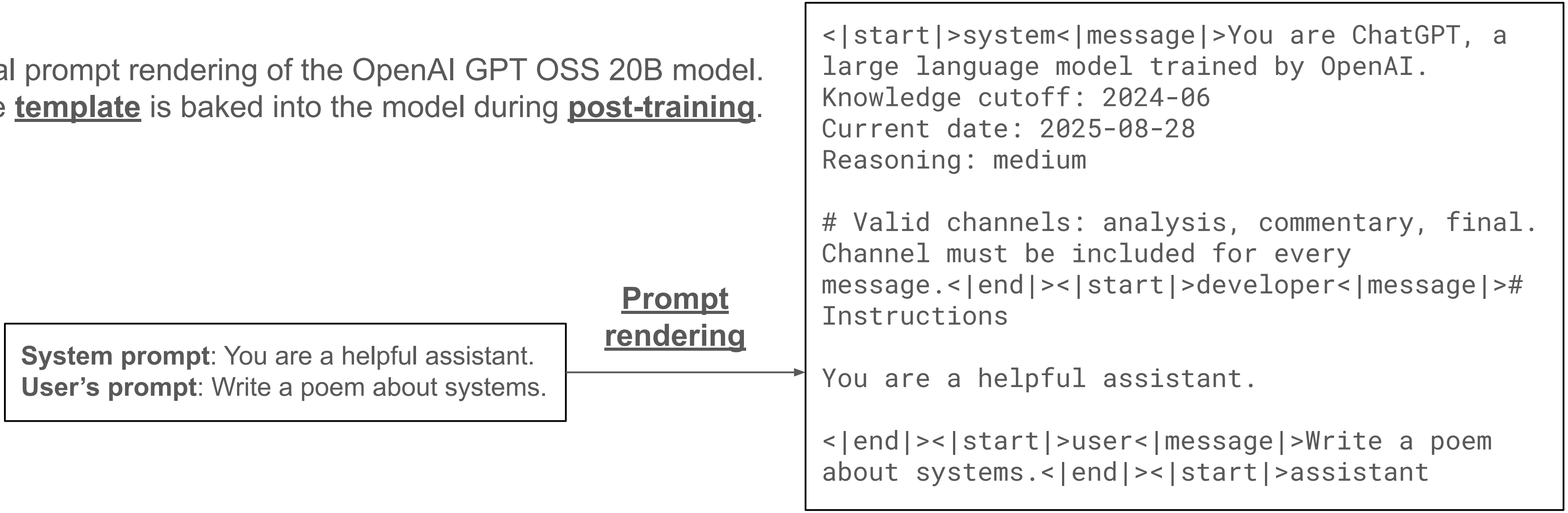


How can I help, Jae-Won?

ChatGPT (OpenAI, Nov 2022)

- Still a text generation model that predicts the next token!
- But the prompt is formatted in a particular way

Real prompt rendering of the OpenAI GPT OSS 20B model. The **template** is baked into the model during **post-training**.



System prompt: You are a helpful assistant.
User's prompt: Write a poem about systems.

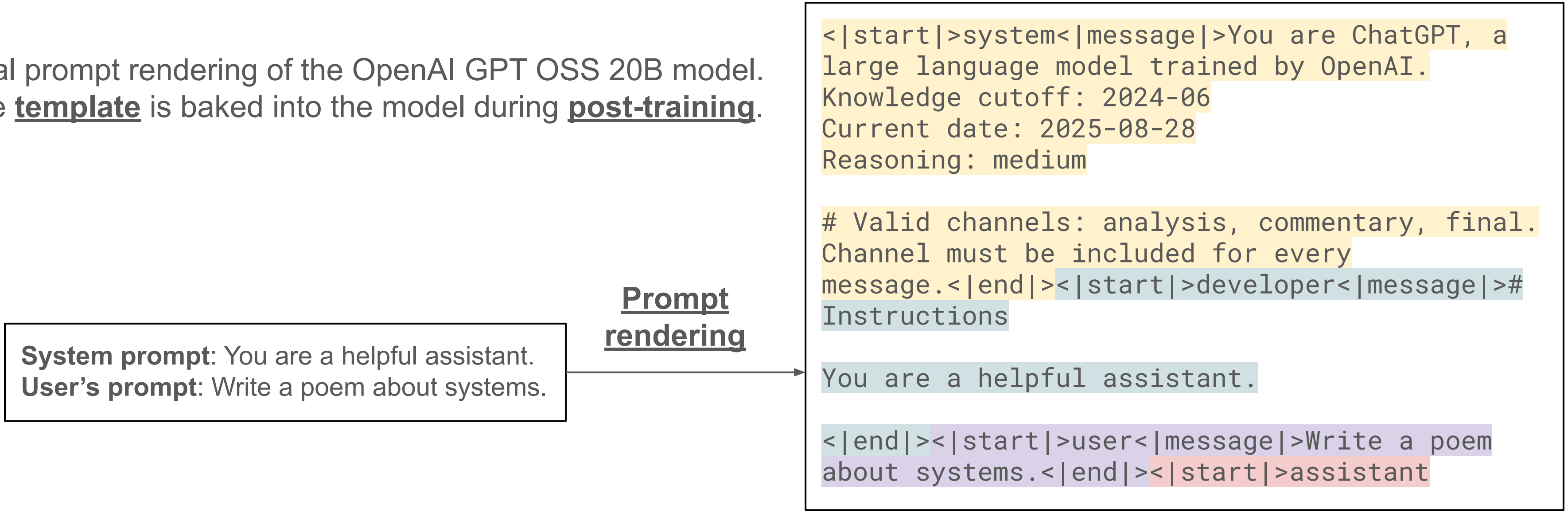
Prompt rendering

```
<|start|>system<|message|>You are ChatGPT, a
large language model trained by OpenAI.
Knowledge cutoff: 2024-06
Current date: 2025-08-28
Reasoning: medium

# Valid channels: analysis, commentary, final.
Channel must be included for every
message.<|end|><|start|>developer<|message|>#
Instructions

You are a helpful assistant.

<|end|><|start|>user<|message|>Write a poem
about systems.<|end|><|start|>assistant
```

ChatGPT (OpenAI, Nov 2022)

- Still a text generation model that predicts the next token!
- But the prompt is formatted in a particular way

Real prompt rendering of the OpenAI GPT OSS 20B model.
The **template** is baked into the model during **post-training**.



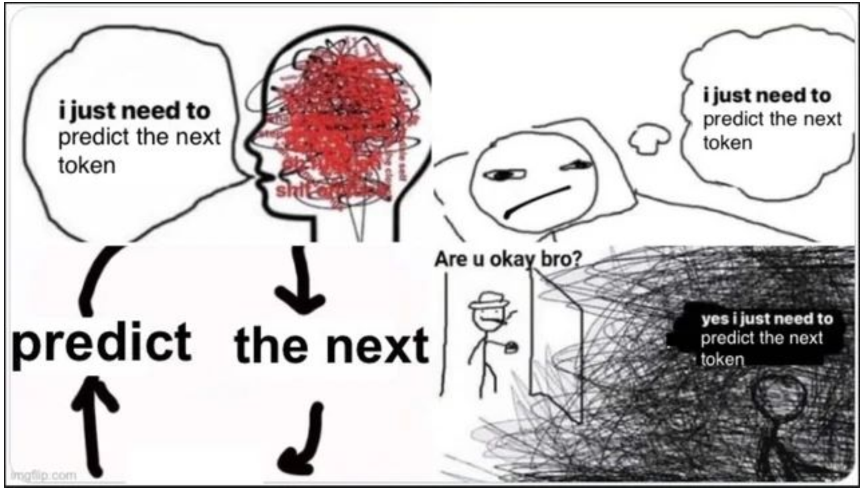
System prompt: You are a helpful assistant.
User's prompt: Write a poem about systems.

Prompt rendering

```
<|start|>system<|message|>You are ChatGPT, a
large language model trained by OpenAI.
Knowledge cutoff: 2024-06
Current date: 2025-08-28
Reasoning: medium

# Valid channels: analysis, commentary, final.
Channel must be included for every
message.<|end|><|start|>developer<|message|>#
Instructions

You are a helpful assistant.

<|end|><|start|>user<|message|>Write a poem
about systems.<|end|><|start|>assistant
```

ChatGPT (OpenAI, Nov 2022)

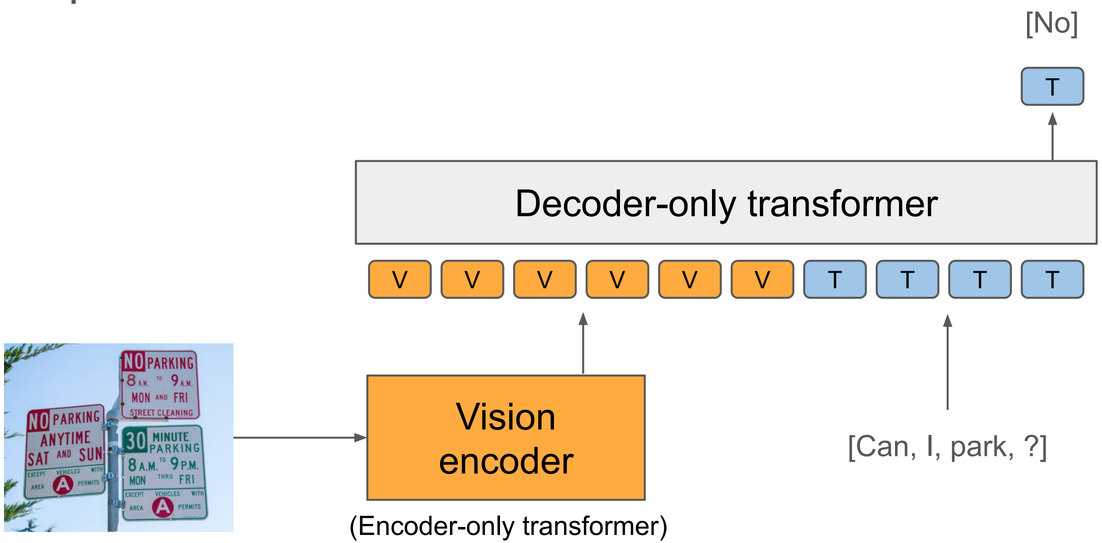
- Still a text generation model that predicts the next token!
- But the prompt is formatted in a particular way


i just need to predict the next token
predict
the next
i just need to predict the next token
Are u okay bro?
yes i just need to predict the next token
imgflip.com

```
<|start|>system<|message|>You are ChatGPT, a
large language model trained by OpenAI.
Knowledge cutoff: 2024-06
Current date: 2025-08-28
Reasoning: medium

# Valid channels: analysis, commentary, final.
Channel must be included for every
message.<|end|><|start|>developer<|message|>#
Instructions

You are a helpful assistant.

<|end|><|start|>user<|message|>Write a poem
about systems.<|end|><|start|>assistant
```

The multimodal era (2023 – now)

- Multimodal means non-text, e.g., image, video, audio
- How to input to LLMs is more or less fixed now



[No]

T

Decoder-only transformer

V V V V V V T T T T

Vision encoder

(Encoder-only transformer)

[Can, I, park, ?]

Recap

- Transformers
  - Training on next token prediction and doing autoregressive text generation.
  - It works pretty well!
- Scaling laws
  - More parameters and more data shall make your model even better.
  - Scale up more and more! We haven’t seen it significantly slow down yet.
- Which brought us …

Unprecedented scale

- Unprecedented scale of adoption
  - ChatGPT passed 1 million users in 5 days and has 800 million weekly active users now.
  - Why? Transformers worked pretty well.
- Unprecedented scale of ML workloads
  - Llama 4 series pre-training took 7.4 million GPU hours. OpenAI GPT OSS 2.1 million.
  - One pre-training job can use 100k GPUs, and there are plans to scale further.
  - We don’t know how many GPUs are serving ChatGPT, but it’s probably a fair amount.
  - Why? Scaling laws fueled the fad, and it has not betrayed us yet.

Unprecedented scale

With great scale

comes great systems challenges

Pre-training

- Much compute and memory requirements
- Much GPU
  - Like, 100k or more for a single training run
  - We are already in the era of multi-datacenter training
  - How are we going to split one model training into 100k+ GPUs?
- Much communication
  - The 100k GPUs need to talk to each other
  - If not optimized well, expensive GPUs are going to spend all their lives doing communication
- Much power and energy
  - An H100 GPU draws anywhere between 100 to 700 W during training, and we have 100k of it
  - Building power plants take years but we need that power now
- Much failure
  - Say a GPU fails at any moment with probability 0.01%
  - The probability of no GPUs failing is $(1 - 0.0001)^{100000} = 0.00004538$

Post-training

- All the problems of pre-training
- Unprecedented adoption also brought about so many requirements
    - Be friendly
    - Be helpful
    - Don’t hallucinate
    - Do reasoning to solve challenging problems (but also don’t take forever)
    - Understand multimodal (image, video, audio) inputs
    - Generate multimodal content, too
- After pre-training, there are typically **tens** of post-training phases with
    - Supervised fine-tuning (SFT)
    - Reinforcement learning (RL)
        - Different environments and rewards
        - Not all rewards are trivial to define

Inference serving

- 1% gain amounts to a lot if you're serving ChatGPT
  - Every drop of efficiency gain is precious!
- LLM text generation has a very distinct computation characteristic
  - Generating the first output token: computation ops > memory ops (**compute-bound**)
  - All other output tokens: computation ops < memory ops (**memory-bound**)
  - Memory management is key!
- Many use cases and application requirements
  - Conversational
  - Multimodal input and output
  - Agents
- ⇒ More and more complex and heterogeneous model architectures

Infrastructure

- Keeping a datacenter with O(10,000) GPUs in it healthy and optimized is hard
- Power management
  - Getting power is the first grand challenge
  - Power management at datacenter-scale is challenging (e.g., mitigating fluctuations)
- Datacenter networking
  - Large amounts of bursty communication during training
  - One straggler bottlenecks the whole training job
- Hardware heterogeneity
  - Especially accelerators (GPUs, TPUs, MTIAs, etc.)
  - (1) You can’t get enough of one type, and (2) relying on a single type introduces supply risks
- Multi-datacenter training makes all of the above even harder

Advanced Scalable Systems

- ML advances → Scalability challenges for systems
- We manage to build systems that scale → Accelerates ML advances
- It’s a great time to be alive and building!