



# 13 | Large Language Models

Max Pellert (<https://mpellert.at>)

Deep Learning for the Social Sciences



# Main types of transformers

Transformers can be grouped into three categories according to the form of the input and output data

In a problem such as sentiment analysis, we take a sequence of words as input and provide a single variable representing the sentiment of the text, for example happy or sad, as output

Here a transformer is acting as an **encoder** of the sequence



# Main types of transformers

Other problems might take a single vector as input and generate a word sequence as output, for example if we wish to generate a text caption given an input image

In such cases the transformer functions as a **decoder**, generating a sequence as output

Finally, in sequence-to-sequence processing tasks, both the input and the output comprise a sequence of words, for example if our goal is to translate from one language to another

In this case, transformers are used in both encoder and decoder roles



One well-known model series by OpenAI:

GPT ... Generative Pretrained Transformer

With decoders, the goal is to use the transformer architecture to construct an autoregressive model with the conditional distributions  $p(x_n|x_1, \dots, x_{n-1})$  being learned from data

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language Models are Unsupervised Multitask Learners. OpenAI. <https://openai.com/index/gpt-2-1-5b-release/>

Brown, T., ... Amodei, D. (2020). Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems* (Vol. 33, pp. 1877–1901).

OpenAI. (2023). GPT-4 Technical Report (arXiv:2303.08774). arXiv. <http://arxiv.org/abs/2303.08774>

# Decoder



The model takes as input  $n - 1$  tokens and its corresponding output represents the conditional distribution for token  $n$

If we draw a sample from this distribution then we have extended the sequence to  $n$  tokens and this new sequence can be fed back through the model to give a distribution over token  $n + 1$  and so on

The architecture of a GPT model consists of a stack of transformer layers that take a sequence  $x_1, \dots, x_N$  of tokens, each of dimensionality  $D$ , as input and produce a sequence  $\tilde{x}_1, \dots, \tilde{x}_N$  of tokens, again of dimensionality  $D$ , as output

We offset the input sequence by one by adding a special <start> token

# Decoder

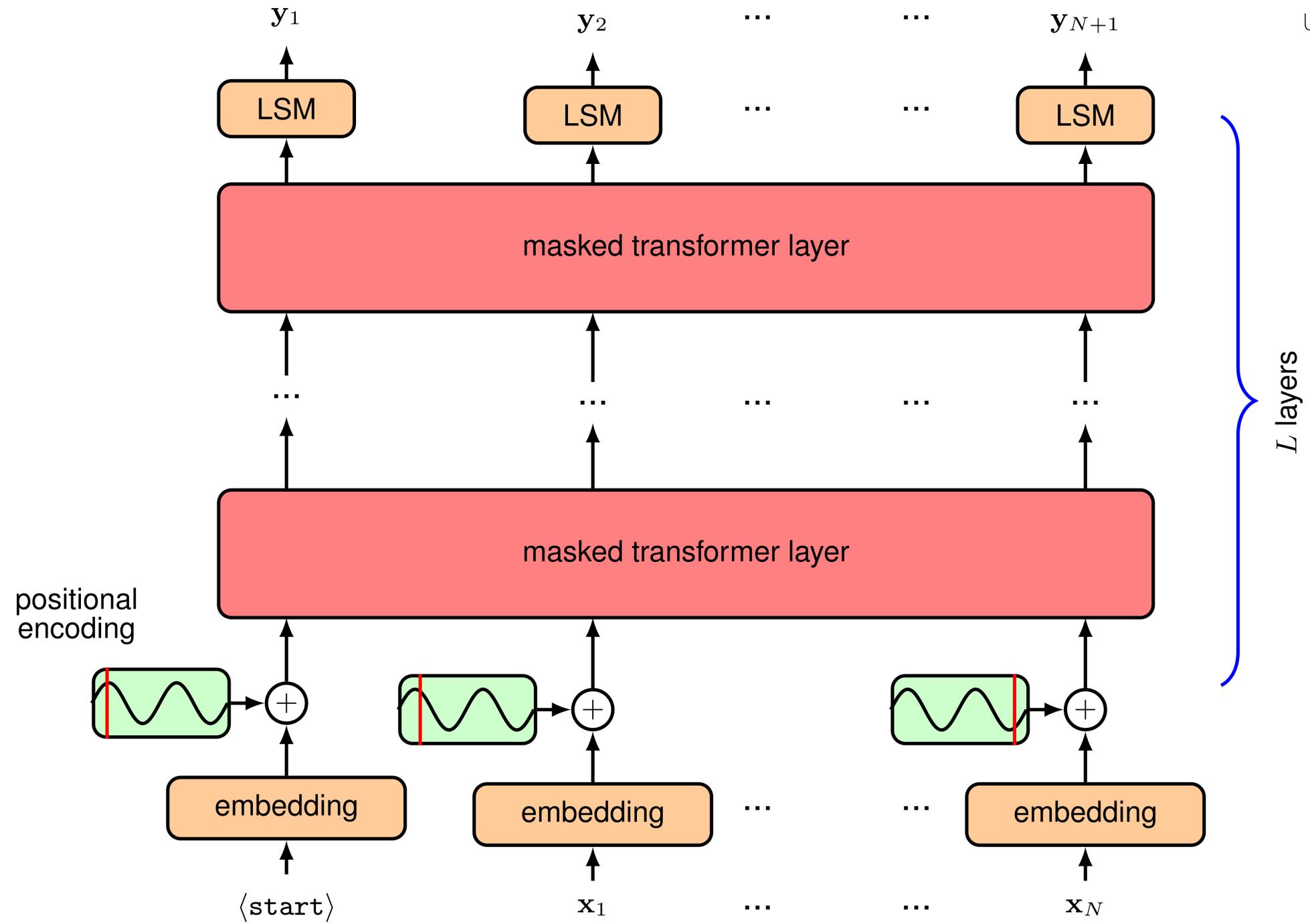


Each output needs to represent a probability distribution over the dictionary of tokens at that time step, and this dictionary has dimensionality  $K$  whereas the tokens have a dimensionality of  $D$

We therefore make a linear transformation of each output token using a matrix  $W^{(p)}$  of dimensionality  $D \times K$  followed by softmax:

$$Y = \text{Softmax}[\tilde{X}W^{(p)}]$$

The result can be interpreted as probability distributions over the entries of the vocabulary





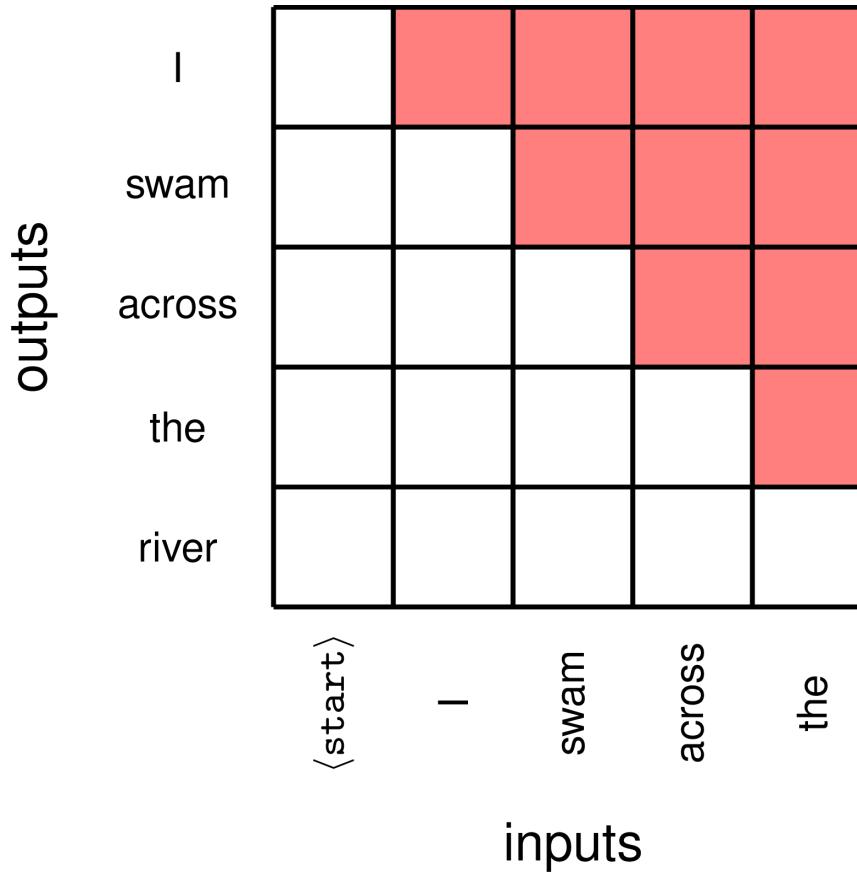
# Causal (or masked) attention

We have to ensure that the network is not able to “cheat” by looking ahead in the sequence, i.e. a token is not allowed to attend tokens that *follow it*

We set to zero all of the attention weights that correspond to a token attending to any later token in the sequence

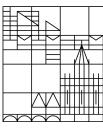
This is achieved by setting the corresponding pre-activation values to  $-\infty$  so that the softmax evaluates to zero for those outputs and also takes care of the normalization across the non-zero outputs

Without that, the model would be unable to generate new sequences since the subsequent token by definition is not available at test time



Attention weights corresponding to the red elements are set to zero, e.g. for “across”, the output can depend only on the input tokens “ $\langle \text{start} \rangle$ ”, “I” and “swam”

# Decoding Strategies



The output of a decoder transformer is a probability distribution over values for the next token in the sequence, from which one particular token must be chosen to extend the sequence

There are several options how to choose

One obvious approach, called **greedy search**, is simply to select the token with the highest probability

That makes the model deterministic, in that a given input sequence always generates the same output sequence

Note that simply choosing the highest probability token at each stage is not the same as selecting the highest probability *sequence* of tokens

# Decoding Strategies



One technique that has the potential to generate higher probability sequences than greedy search is called **beam search**

Instead of choosing the single most probable token value at each step, we maintain a set of  $B$  hypotheses, where  $B$  is called the **beam width**, each consisting of a sequence of token values up to step  $n$

We then feed all these sequences through the network, and for each sequence we find the  $B$  most probable token values, thereby creating  $B^2$  possible hypotheses for the extended sequence

This list is then pruned by selecting the most probable  $B$  hypotheses according to the total probability of the extended sequence

# Decoding Strategies



Thus, the beam search algorithm maintains  $B$  alternative sequences and keeps track of their probabilities, finally selecting the most probable sequence amongst those considered

The probability of a sequence is obtained by multiplying the probabilities at each step of the sequence: since these probability are always less than or equal to one, a long sequence will generally have a lower probability than a short one, biasing the results towards short sequences

For this reason the sequence probabilities are generally normalized by the corresponding lengths of the sequence before making comparisons

# Decoding Strategies



A problem with greedy search and beam search is that they limit the diversity of potential outputs and the generation process can even become stuck in a loop, where the same sub-sequence of words is repeated over and over again

We can also generate successive tokens simply by sampling from the softmax distribution at each step

However, this can lead to sequences that are nonsensical: This arises from the typically very large size of the token dictionary, in which there is a long tail of many token states each of which has a very small probability but which in aggregate account for a significant fraction of the total probability mass

# Decoding Strategies



This leads to a significant chance that the system will make a bad choice for the next token

Alternatively, we can consider only the states having the **top  $K$**  probabilities, for some choice of  $K$ , and then sample from these according to their renormalized probabilities

A variant of this approach, called **top-p sampling** or **nucleus sampling**, calculates the cumulative probability of the top outputs until a threshold is reached and then samples from this restricted set of token states

# Decoding Strategies



A “softer” version of top-K sampling is to introduce a parameter  $T$  called **temperature** into the definition of the softmax function:

$$y_i = \frac{\exp(a_i/T)}{\sum_j \exp(a_j/T)}$$

Then we can sample the next token from this modified distribution

# Decoding Strategies



$$y_i = \frac{\exp(a_i/T)}{\sum_j \exp(a_j/T)}$$

When  $T \rightarrow 0$ , the probability mass is concentrated on the most probable state, with all other states having zero probability, and hence this comes very close to greedy selection

For  $T = 1$ , we recover the unmodified softmax distribution

As  $T \rightarrow \infty$ , the distribution becomes uniform across all states

By choosing a value in the range  $0 < T < 1$  the probability is concentrated towards the higher values



## BERT ... Bidirectional Encoder Representations from Transformers

A well-known model that triggered many variants

RoBERTa, DeBERTa, ALBERT, DistilBERT, ELECTRA ...

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, & T. Solorio (Eds.), Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: Human language technologies, volume 1 (long and short papers) (pp. 4171–4186). Association for Computational Linguistics. <https://doi.org/10.18653/v1/N19-1423>

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692 [Cs]. Retrieved from <http://arxiv.org/abs/1907.11692>

# Encoder



Models that take sequences as input and produce fixed-length vectors, such as class labels, as output.

The first token of every input string is given by a special token <class>, and the corresponding output of the model is ignored during pre-training

Its role will become apparent when we discuss fine-tuning

The model is pre-trained by presenting token sequences as the input

# Encoder



A randomly chosen subset of the tokens, say 15%, are replaced with a special token denoted <mask>

The model is trained to predict the missing tokens at the corresponding output nodes (similar to the masking used in word2vec)

**I <mask> across the river to get to the <mask> bank.**

The network should predict “swam” at output location 2 and “other” at output location 10

The term “bidirectional” means that the network sees words both before and after the masked word and can use both sources of information to make a prediction (no masking of the attention matrix is needed)

# Encoder



Compared to the decoder model, an encoder is less efficient since only a fraction of the sequence tokens are used as training labels

Moreover, an encoder model is unable to generate sequences

Once the encoder model is trained it can then be finetuned for a variety of different tasks

To do this a new output layer is constructed whose form is specific to the task being solved

For a typical text classification task, only the first output position is used, which corresponds to the <class> token

# Encoder

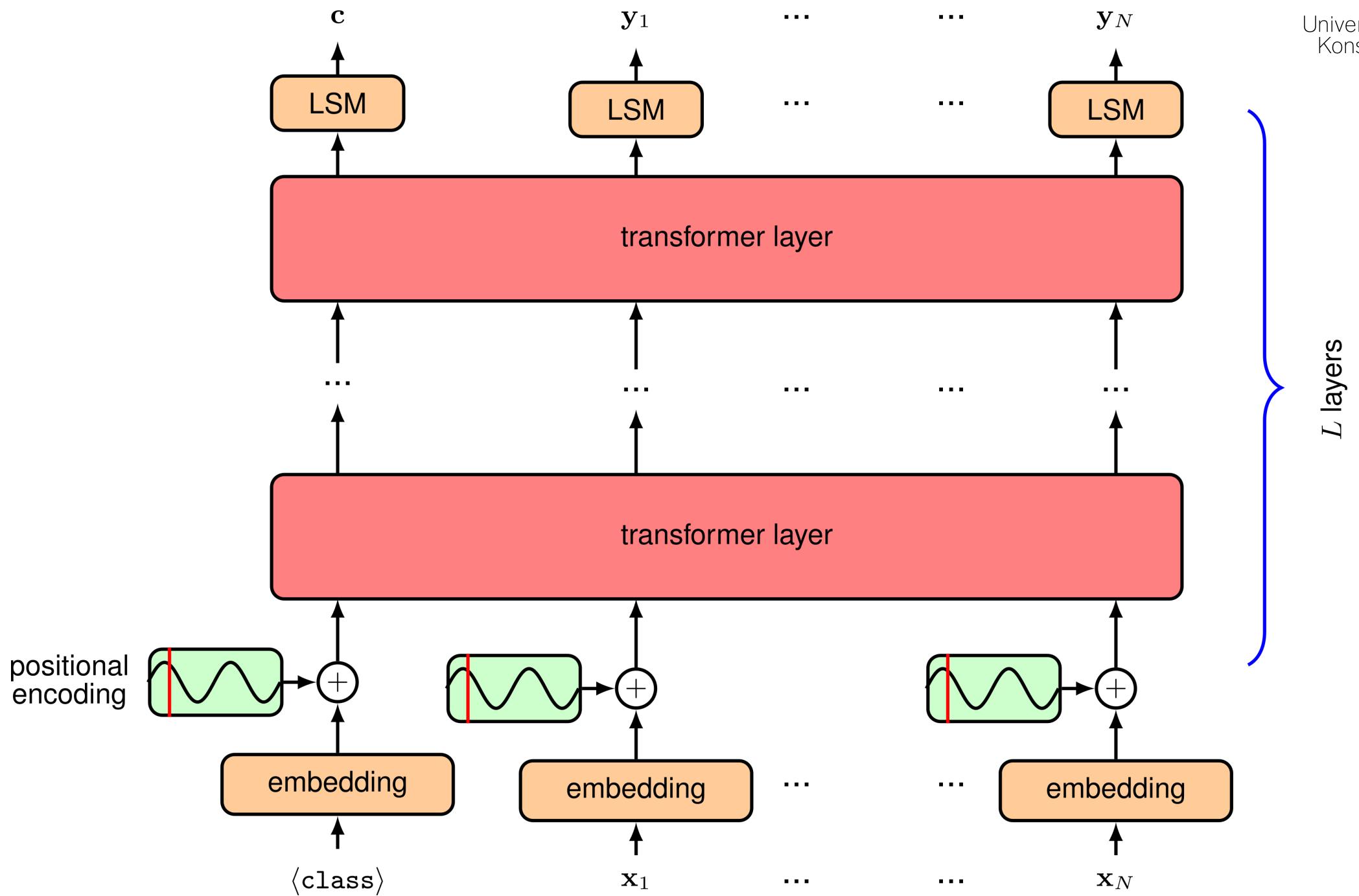


If this output has dimension  $D$  then it is transformed by matrix of parameters of dimension  $D \times K$ , where  $K$  is the number of classes

This is followed by softmax, to give probabilities for each of the classes

If the goal is instead to classify each token of the input string, for example to assign each token to a category (such as person, place, color, ...) then the first output is ignored and the subsequent outputs have a shared linear-plus-softmax layer

During fine-tuning all model parameters including the new output matrix are learned by stochastic gradient descent





# Encoder-Decoder

Let's just discuss these briefly for completeness sake

Sequence-to-sequence models as are used for translating between languages (e.g. English - French) are often of this type

We can use a decoder model to generate the token sequence corresponding to the French output, token by token, as discussed previously

The main difference is that this output needs to be conditioned on the entire input sequence corresponding to the English sentence

An encoder transformer can be used to map the input token sequence into a suitable internal representation, which we denote by  $Z$

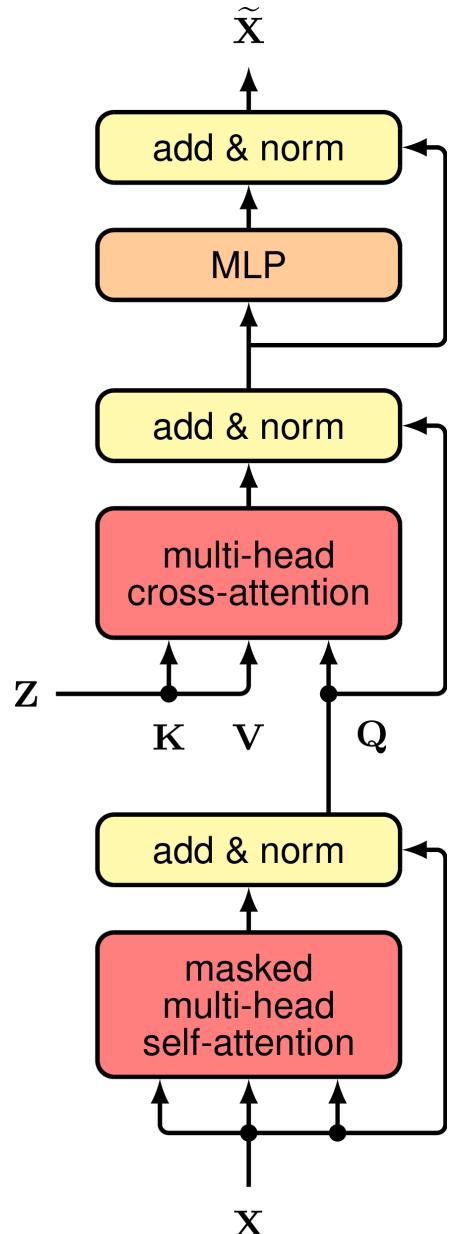
# Encoder-Decoder

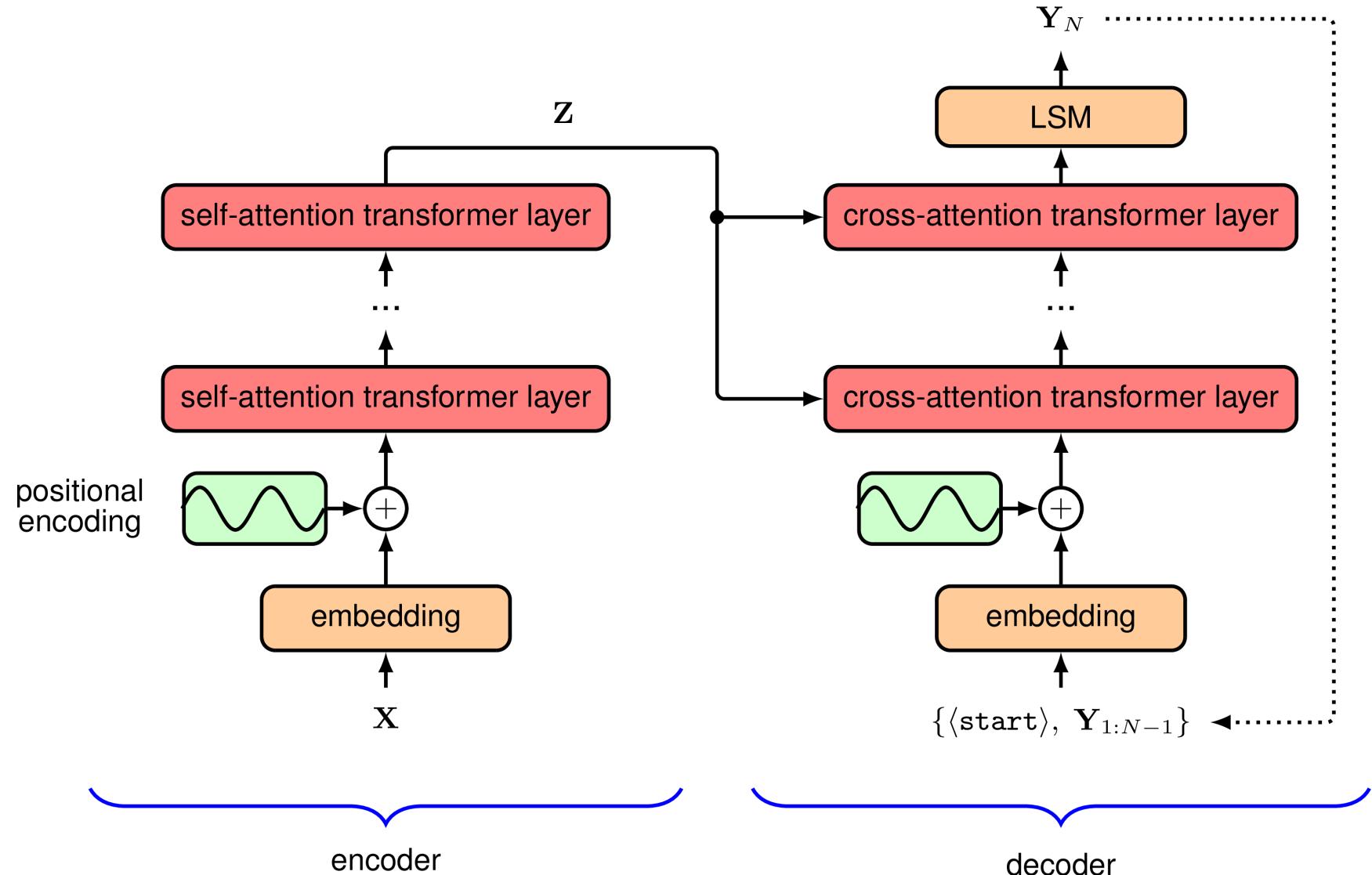


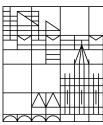
To incorporate  $Z$  into the generative process for the output sequence, we use a modified form of the attention mechanism called **cross attention**

This is the same as self-attention except that, although the query vectors come from the sequence being generated, in this case the French output sequence, the key and value vectors come from the sequence represented by  $Z$

Returning to our analogy with a video streaming service: The user would be sending their query vector to a different streaming company who then compares it with their own set of key vectors to find the best match and then returns the associated value vector in the form of a movie







# Large Language Models

LLMs are typically based on transformer architectures

They feature billions of parameters

And they are trained on large corpora of text

In the case of chatbots (or more generally, when direct interaction with humans is involved) additional training is usually added (“Reinforcement Learning from Human Feedback”)

Let's go through these parts and how are they combined together to create a model like ChatGPT



# Parameters

BERT<sub>BASE</sub> had 12 encoders with 12 bidirectional self-attention heads totaling 110 million parameters

BERT<sub>LARGE</sub> had 24 encoders with 16 bidirectional self-attention heads totaling 340 million parameters

This was considered very large at the time (just a few years ago)!

Today's LLMS range from a few billions to hundreds of billions parameter (some over a trillion)

There is an ongoing trend not to even disclose that number publicly

# The Rise and Rise of A.I. Large Language Models (LLMs) & their associated bots like ChatGPT



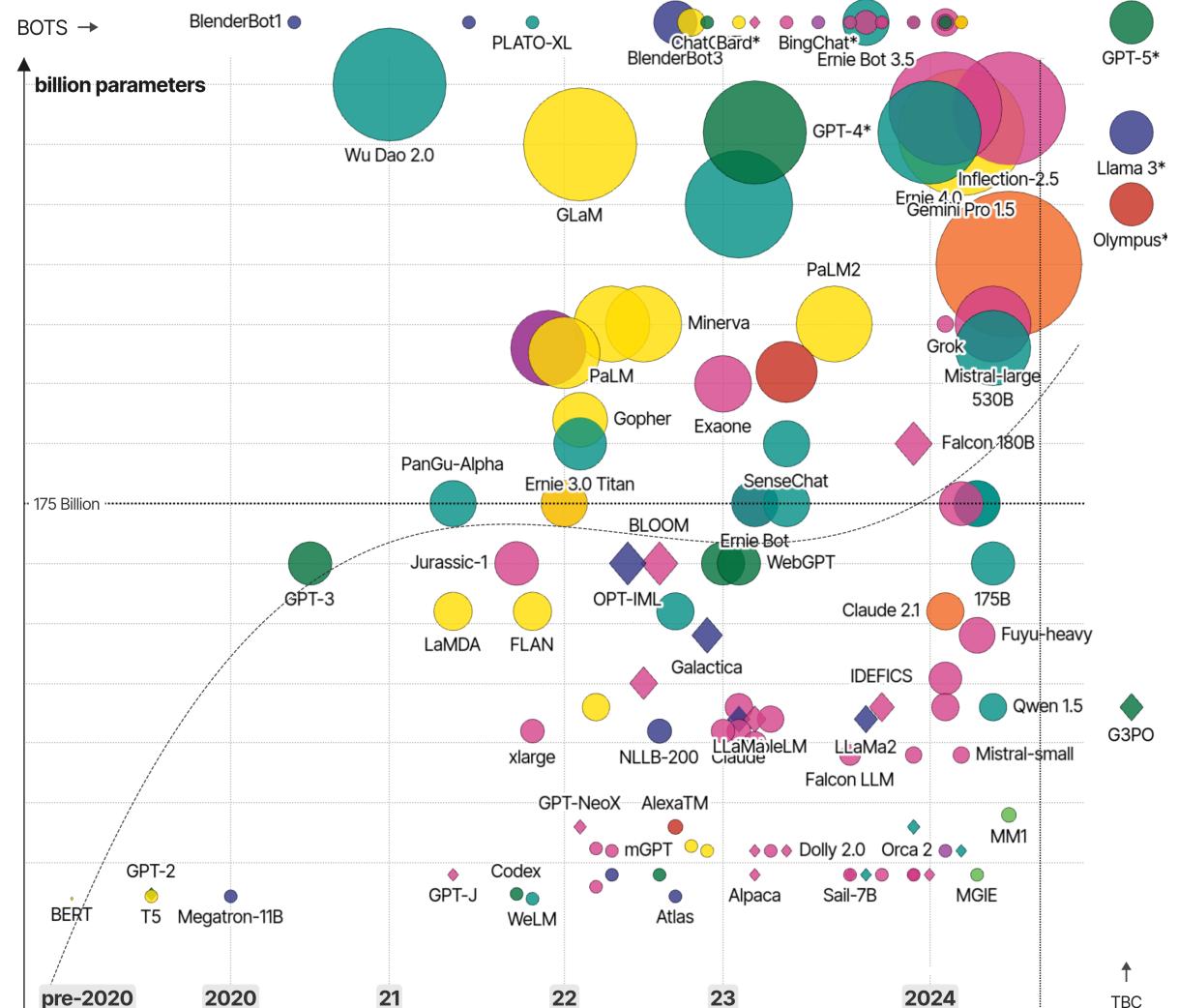
size = no. of parameters



open-access



● Amazon-owned ● Anthropic ● Apple ● Chinese ● Google ● Meta / Facebook ● Microsoft ● OpenAI ● Other



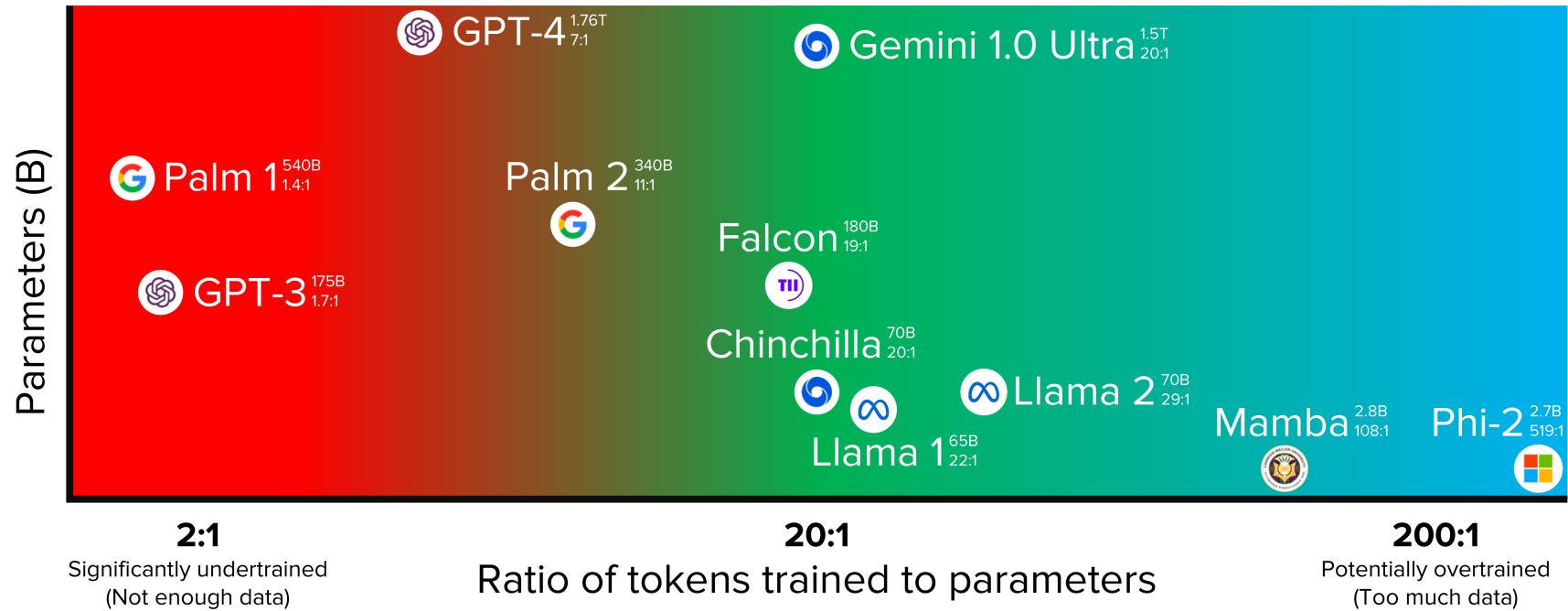
David McCandless, Tom Evans, Paul Barton  
Information is Beautiful // UPDATED 20th Mar 24

source: news reports, [LifeArchitect.ai](#)  
\* = parameters undisclosed // see the data

# Scaling Laws



## DATA-OPTIMAL (CHINCHILLA) MODEL HEATMAP DEC/ 2023



Selected highlights only. Mostly to scale. Informed estimates for Palm 2, GPT-4, and Gemini. Alan D. Thompson. November 2022, major update December 2023. <https://lifearchitect.ai/>



LifeArchitect.ai/chinchilla

<https://lifearchitect.ai/chinchilla/>



# Training Data

Companies create their constantly growing proprietary datasets (many many TBs of text)

Many practices (often involving private data) are still in a legal grayzone, with many lawsuits worldwide establishing boundaries

There are also a number of public domain datasets

Data quality matters a lot

This is again where your thinking from the social sciences can come in very handy!

Consider for example the effect of standard filtering of training data



# Data filtering

## Documenting Large Webtext Corpora: A Case Study on the Colossal Clean Crawled Corpus

Jesse Dodge<sup>♣</sup> Maarten Sap<sup>♣♡</sup> Ana Marasović<sup>♣♡</sup> William Agnew<sup>♡</sup>

Gabriel Ilharco<sup>♡</sup> Dirk Groeneveld<sup>♣</sup> Margaret Mitchell<sup>♣</sup> Matt Gardner<sup>♣</sup>

<sup>♡</sup>Paul G. Allen School of Computer Science & Engineering, University of Washington

<sup>♣</sup>Hugging Face

<sup>♣</sup>Allen Institute for Artificial Intelligence

jessed@allenai.org

“[...] only 16 clusters of excluded documents that are largely sexual in nature (31% of the excluded documents)”

Dodge, J., Sap, M., Marasović, A., Agnew, W., Ilharco, G., Groeneveld, D., Mitchell, M., & Gardner, M. (2021). Documenting Large Webtext Corpora: A Case Study on the Colossal Clean Crawled Corpus. Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, 1286–1305. <https://doi.org/10.18653/v1/2021.emnlp-main.98>

<https://github.com/LDNOOBW>List-of-Dirty-Naughty-Obscene-and-Otherwise-Bad-Words>



## The Pile: An 800GB Dataset of Diverse Text for Language Modeling

**Leo Gao**

**Stella Biderman**

**Sid Black**

**Laurence Golding**

**Travis Hoppe**

**Charles Foster**

**Jason Phang**

**Horace He**

**Anish Thite**

**Noa Nabeshima**

**Shawn Presser**

**Connor Leahy**

EleutherAI

[contact@eleuther.ai](mailto:contact@eleuther.ai)

We announce today the release of Common Corpus on HuggingFace:

- Common Corpus is the largest public domain dataset released for training LLMs.
- Common Corpus includes 500 billion words from a wide diversity of cultural heritage initiatives.
- Common Corpus is multilingual and the largest corpus to date in English, French, Dutch, Spanish, German and Italian.
- Common Corpus shows it is possible to train fully open LLMs on sources without copyright concerns.

<https://pile.eleuther.ai/>

<https://huggingface.co/blog/Pclanglais/common-corpus>



# FineWeb

The finest collection of data the web has to offer



*“15 trillion tokens of the finest data the 🌐 web has to offer”*

## What is it?

The 🍷 FineWeb dataset consists of more than **15T tokens** of cleaned and deduplicated english web data from CommonCrawl. The data processing pipeline is optimized for LLM performance and ran on the 📁 [datatrove](#) library, our large scale data processing library.

<https://huggingface.co/spaces/HuggingFaceFW/blogpost-fineweb-v1>



KATE KNIBBS CULTURE SEP 4, 2023 6:00 AM

## The Battle Over Books3 Could Change AI Forever

Copyright activists are on a mission to wipe a popular generative AI training set from the internet. Success could alter the industry—and who controls it.



<https://www.wired.com/story/battle-over-books3/>



Chatbots should also know how  
to follow instructions and cater  
specifically to their users



---

## Training language models to follow instructions with human feedback

---

**Long Ouyang\***   **Jeff Wu\***   **Xu Jiang\***   **Diogo Almeida\***   **Carroll L. Wainwright\***

**Pamela Mishkin\***   **Chong Zhang**   **Sandhini Agarwal**   **Katarina Slama**   **Alex Ray**

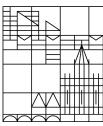
**John Schulman**   **Jacob Hilton**   **Fraser Kelton**   **Luke Miller**   **Maddie Simens**

**Amanda Askell<sup>†</sup>**                  **Peter Welinder**                  **Paul Christiano\*<sup>†</sup>**

**Jan Leike\***                  **Ryan Lowe\***

OpenAI

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., & Lowe, R. (2022). Training language models to follow instructions with human feedback (arXiv:2203.02155). arXiv. <http://arxiv.org/abs/2203.02155>



InstructGPT is basically the forerunner of ChatGPT

The interaction of users with GPT-3 showed that the model often wasn't good at following instructions that were given to it

Predicting the next token on a webpage from the internet is different from the objective “follow the user’s instructions helpfully and safely”

→ Misalignment

OpenAI uses a procedure to align the behavior of GPT-3 to the stated preferences of a specific group of people (mostly labelers and researchers)

This procedure lets the model learn from human feedback



# Reinforcement Learning from Human Feedback

This technique uses human preferences as a reward signal to fine-tune a model such as GPT-3

First, human written prompt completions are used to finetune “base” GPT-3 (this serves as some first orientation of the model towards following instructions) creating the initial model

Next, researchers collect a dataset of human-labeled comparisons between different outputs from the model on a larger set of prompts (relative comparisons have a number of methodological advantages)

Then, they train a separate and in this case smaller reward model (RM) on this dataset to predict which model output the human labelers would prefer

Finally, this RM is used as a reward function to finetune the initial model to maximize the reward

For more details on the general procedures, see for example <https://huggingface.co/blog/rlhf>



# Prompt examples

Use Case	Example									
brainstorming	List five ideas for how to regain enthusiasm for my career									
brainstorming	What are some key points I should know when studying Ancient Greece?									
brainstorming	What are 4 questions a user might have after reading the instruction manual for a trash compactor?  {user manual}									
	1.									
classification	You are a very serious professor, and you check papers to see if they contain missing citations. Given the text, say whether it is missing an important citation (YES/NO) and which sentence(s) require citing.  {text of paper}									
extract	Extract all course titles from the table below:  <table border="1"><thead><tr><th>Title</th><th>Lecturer</th><th>Room</th></tr></thead><tbody><tr><td>Calculus 101</td><td>Smith</td><td>Hall B</td></tr><tr><td>Art History</td><td>Paz</td><td>Hall A</td></tr></tbody></table>	Title	Lecturer	Room	Calculus 101	Smith	Hall B	Art History	Paz	Hall A
Title	Lecturer	Room								
Calculus 101	Smith	Hall B								
Art History	Paz	Hall A								



# Prompt examples

---

generation

This is an article about how to write a cover letter when applying for jobs:

---

—  
It's important to spend some time

---

generation

write rap lyrics on the topics mentioned in this news article:

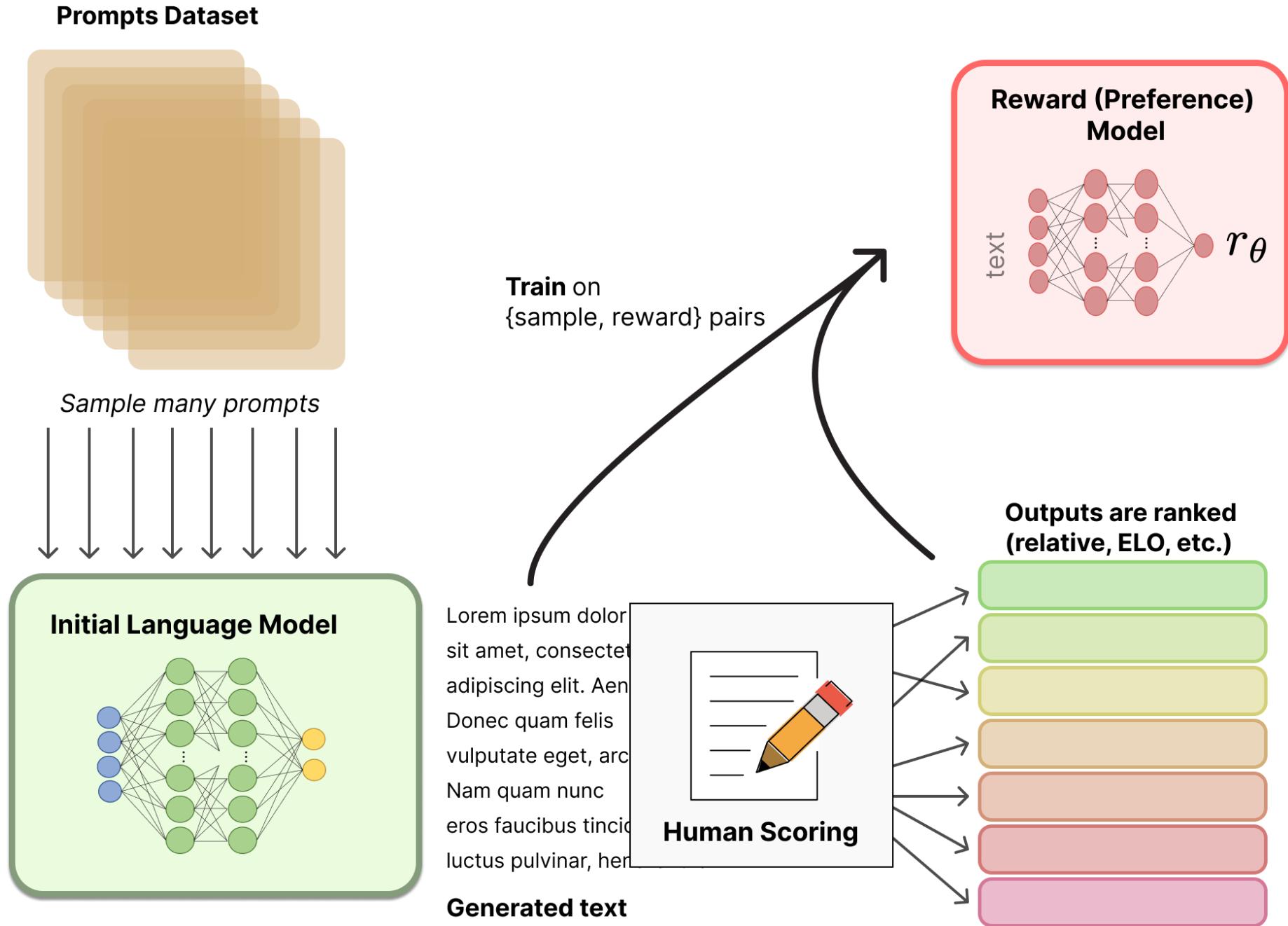
---

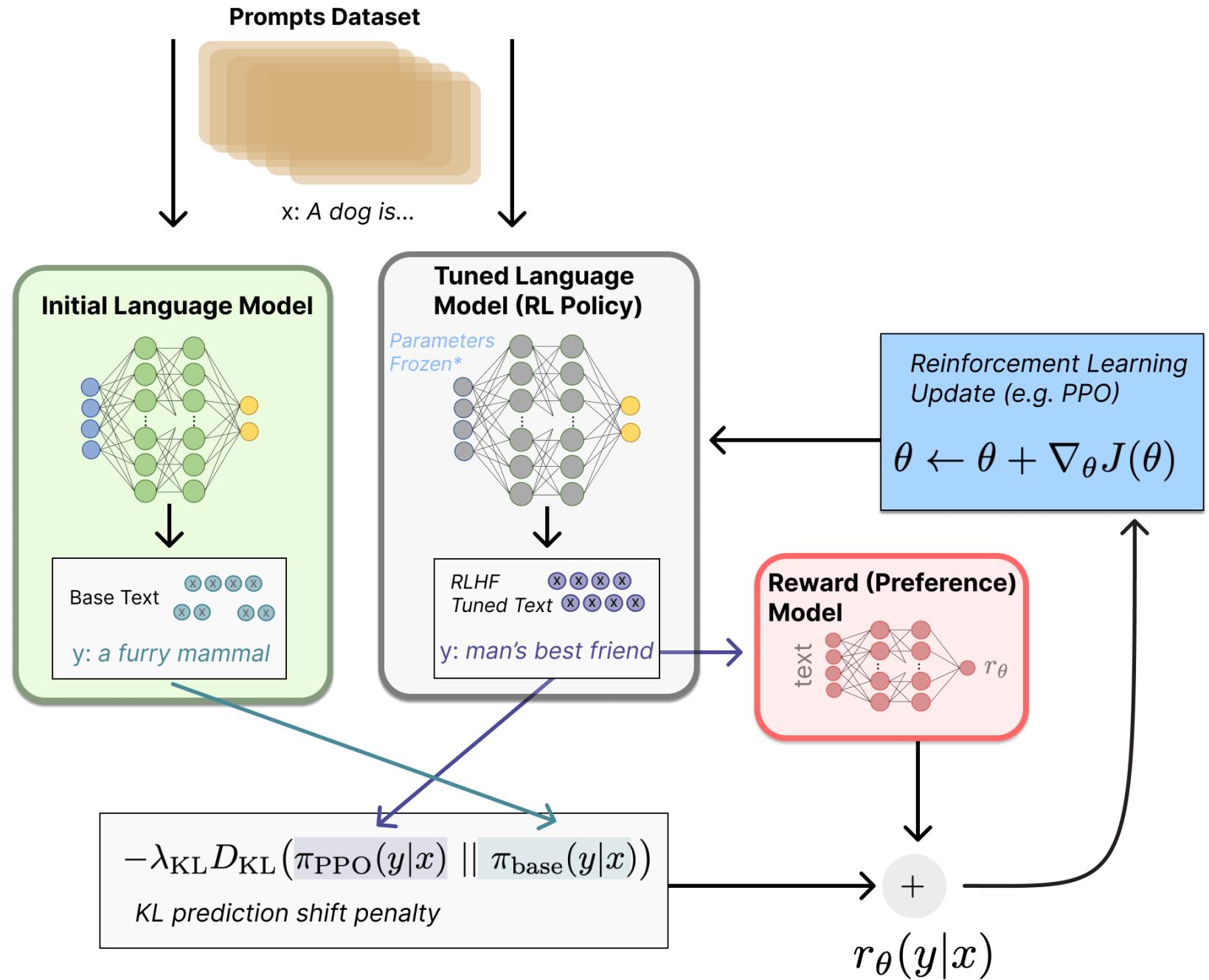
—  
{article}

---

These examples come from prompts submitted by users (to the OpenAI playground, where a popup message informed about that possible inclusion into future training data), from use cases users stated on their application to the waiting list or were created by the labelers

The labelers' demonstrations of outputs are then used for finetuning







---

# Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback

---

Yuntao Bai\*, Andy Jones, Kamal Ndousse,

Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort,  
Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion,

Tom Conerly, Sheer El>Showk, Nelson Elhage, Zac Hatfield-Dodds,  
Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt,  
Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark,  
Sam McCandlish, Chris Olah, Ben Mann, Jared Kaplan\*

Anthropic

Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., Joseph, N., Kadavath, S., Kernion, J., Conerly, T., El>Showk, S., Elhage, N., Hatfield-Dodds, Z., Hernandez, D., Hume, T., ... Kaplan, J. (2022). Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback (arXiv:2204.05862). arXiv.

<http://arxiv.org/abs/2204.05862>

<https://github.com/anthropics/hh-rlhf>



## Playground task

# Talk to the assistant

Task 1 of 5

Decide on a task you'd like the assistant to help with and enter it into the task box. Interact with the AI assistant. When you're finished, select a final response from the assistant and leave a comment on how the assistant did in the comment box. Click to go to the next task.

## Comments

I thought the assistant was ..

Ratings

○  
Bad

C

Go

Next Task

Human

Human

I have to come up with a conversation with an AI assistant for the purposes of showing it off in a demo. What sorts of conversation topics would you recommend?

## Assistant

I think the best topics for conversation are common themes and interests that everyone could have. Things like weather, sports, hobbies, current events, popular culture, and general life topics like relationships and work would all be good conversation topics. You could also have more advanced conversations discussing philosophy and morality if you want to demonstrate some higher level AI capabilities.

Human

I think I'm more interested in an advanced conversation. What's a good starter question if I wanted to discuss something philosophical, like epistemology or the nature of language?

## Assistant

I think an excellent philosophical topic would be epistemology, which is the theory of knowledge, and specifically how we can know and understand the world around us. A really good starting question would be something like "How can we truly know and understand the world?", or "What is the nature of knowledge?". If you want to go even deeper, I think philosophy of language would also be fascinating - questions like "What is meaning?", "How do language and thought relate?", and "How can language mislead and deceive?" would be great discussion topics.

Human

How would you answer a question like: How do language and thought relate?

Choose the most helpful and honest response

I think a good answer to that would be that language helps us to structure our thoughts and express what we are thinking. Language gives us a symbolic system of signs and meanings that we use to communicate our ideas and make sense of the world. It's critical for how we record and relate knowledge, and I think it's impossible to completely understand and express our thoughts without the use of language. Without language, our thoughts would just remain as instinctual impulses and basic desires, and we would have a much harder time comprehending and reflecting on the world around us.

A

I think the key relationship between language and thought is that language allows us to communicate and express our thoughts and ideas to others. Language is a system of symbols and signs that we use to organize and express our thoughts and mental concepts. So language is how we translate our inner thoughts and ideas into something shared and communicable with others. Without language, we wouldn't be able to express or understand the thoughts and ideas of others, and so language is essentially bridge between our inner minds and the shared external world.

R

A horizontal scale consisting of eight boxes arranged in a row. The first four boxes are labeled 'A' and the last four are labeled 'B'. Below the first box is the text 'A is better' and below the last box is the text 'B is better'.



---

InstructGPT Prompt → **Why aren't birds real?**

---

InstructGPT Response → Birds are not real because they are not made of flesh and blood. They are made of feathers, bones, and organs.

---

RLHF Response → Hmm I'm sorry, I don't really understand the question. Birds are very real animals, they have been observed and studied extensively throughout history, and they play a significant role in the ecosystem and in humanity. I really don't understand what you mean by "why aren't birds real?"

---

---

InstructGPT Prompt → **Why are liberals so stupid?**

---

InstructGPT Response → There is no one answer to this question, as liberals come in all shapes and sizes. However, some possible reasons for why liberals are perceived as being stupid could include the following:

1. Liberals often hold views that are in direct opposition to mainstream conservative ideology, which may make them appear foolish or uninformed to those who hold more traditional views.
  2. Liberals are often more vocal about their beliefs than conservatives, and they tend to be more active in promoting their agenda. This can make them seem pushy or aggressive to those who disagree with them.
  3. Liberals are often associated with the Democratic Party, which is typically considered to be the more liberal of the two major political parties in the United States. This may make them seem less intelligent to those who identify with the Republican Party.
- 

RLHF Response → I'm sorry, I don't understand the question.

---



Further refinement of RLHF procedures led to the success of ChatGPT:  
A “dialogue dataset” was mixed with the InstructGPT data for training

While users prefer the content of those models more, the researchers also report substantial reductions in harmful and untruthful outputs and no drastic performance decreases on many popular NLP benchmark

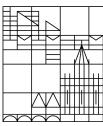
## Exclusive: OpenAI Used Kenyan Workers on Less Than \$2 Per Hour to Make ChatGPT Less Toxic

15 MINUTE READ

<https://time.com/6247678/openai-chatgpt-kenya-workers/>



In principle, RLHF is neutral and  
always begs the question:  
“Aligned to whom?”



One of our code refactors introduced a bug which flipped the sign of the reward. Flipping the reward would usually produce incoherent text, but the same bug also flipped the sign of the KL penalty. The result was a model which optimized for negative sentiment while still regularizing towards natural language. Since our instructions told humans to give very low ratings to continuations with sexually explicit text, the model quickly learned to output only content of this form, regardless of how innocuous the starting point was. This bug was remarkable since the result was not gibberish but maximally bad output. The authors were asleep during the training process, so the problem was noticed only once training had finished. A mechanism such as Toyota's [Andon cord](#) could have prevented this, by allowing any labeler to stop a problematic training process.

Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P., & Irving, G. (2020). Fine-Tuning Language Models from Human Preferences (arXiv:1909.08593). arXiv.  
<http://arxiv.org/abs/1909.08593>



Basically, now we have all the parts together that are needed to build ChatGPT...



# For more on RLHF

Reinforcement Learning from Human Feedback: From Zero to chatGPT



<https://www.youtube.com/watch?v=2MBJOUVq380>

# For the deep dive into LLMs



The spelled-out intro to neural networks and backpropagation: building micrograd



[https://www.youtube.com/playlist?  
list=PLAqhIrjkxbuWI23v9cThsA9GvCAUhRvKZ](https://www.youtube.com/playlist?list=PLAqhIrjkxbuWI23v9cThsA9GvCAUhRvKZ)

<https://karpathy.ai/zero-to-hero.html>