

Large Language Models

Prof. Luca Cagliero
Dipartimento di Automatica e Informatica
Politecnico di Torino

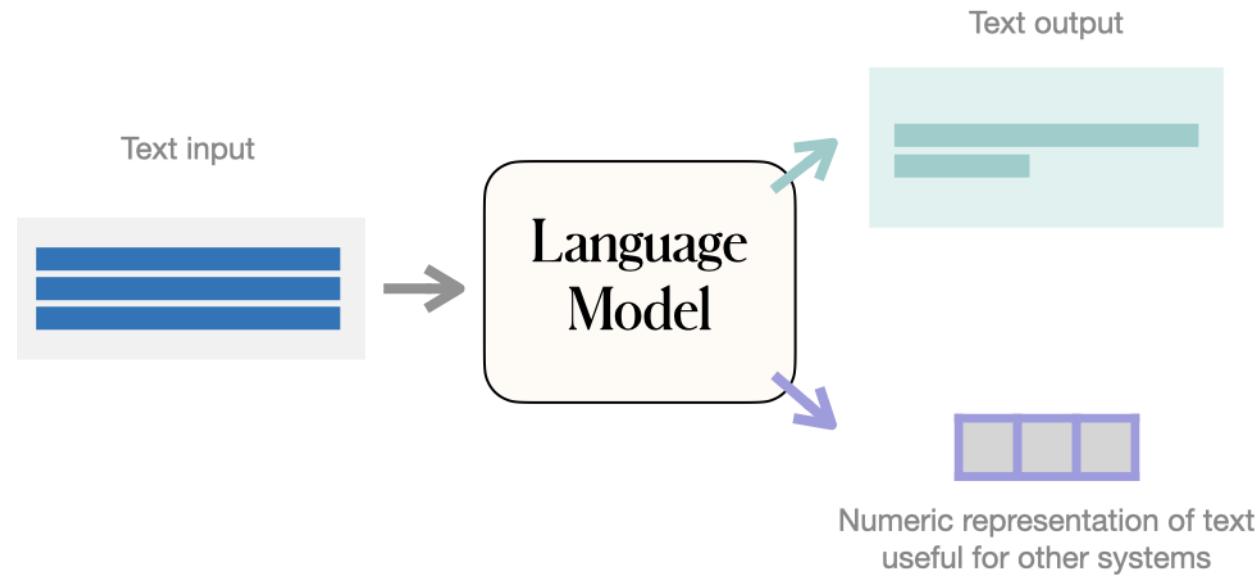


Lecture goal

- Language Models: recap of fundamentals
- Autoregressive LMs and conditional text generation
- Key LLM features and categories
- Main LLM architectures and pretraining objectives
- LLM adaptation

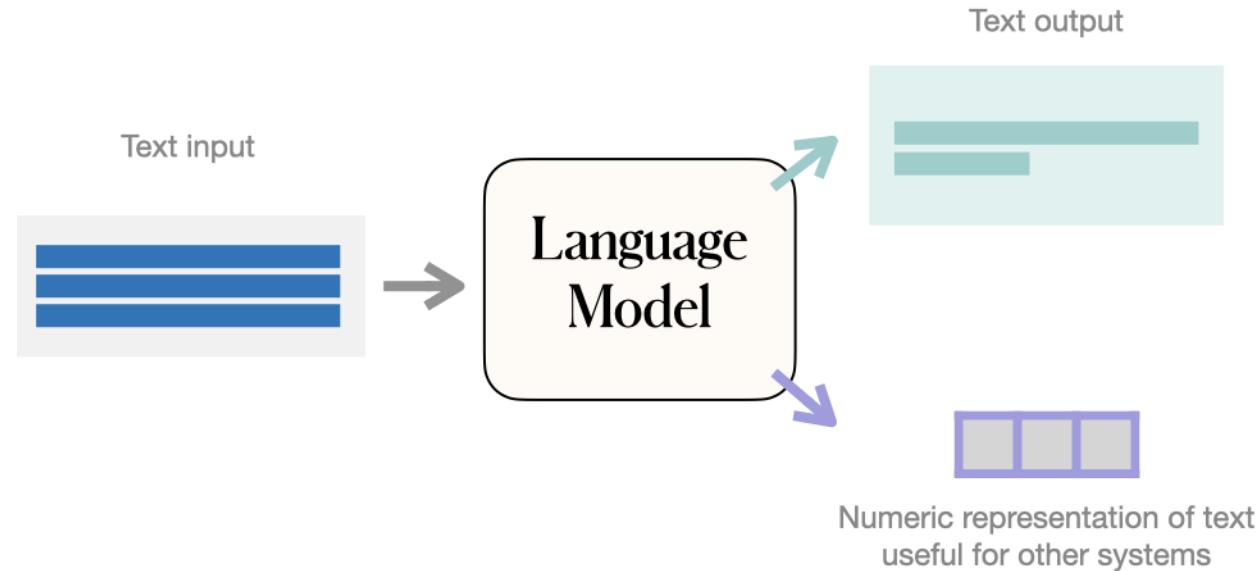
Language Model

- Model the generative likelihood of word sequences to predict the probabilities of future (or missing) tokens



Language Model

- Given a sequence, the LM returns a probability to assess its goodness
- Probability distribution over sequences of tokens in a vocabulary
 - The probability indicates how good that sequence of tokens is



Language Models: fundamentals

- Predict the probabilities of future (or missing) tokens
- Training sequence of tokens: y_1, \dots, y_n
- Model prediction (timestep t): $p^{(t)} = p(\cdot | y_1, \dots, y_{t-1})$
- Target (timestep t): $p^* = \text{one-hot}(y_t)^{|V|}$
- Loss function: $\text{Loss}(p^*, p) = -p^* \log(p) = -\sum_{i=1}^{|V|} p_i^* \log(p_i)$

Language Models: fundamentals

- Since just one p^* term is non-zero, i.e., the correct token y_t

$$Loss(p^*, p) = -\log(p_{y_t}) = -\log(p(y_t|y_{<t}))$$

- For the whole sequence:

$$-\sum_{t=1}^n \log(p(y_t|y_{<t}))$$

Autoregressive Language Models

- Conditional probability distribution $p(x_i \mid x_{1:i-1})$
 - Can be estimated by an autoregressive model like feedforward neural network
 - Chain rule of probability:

$$p(x_{1:L}) = p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_1, x_2) \cdots p(x_L \mid x_{1:L-1}) = \prod_{i=1}^L p(x_i \mid x_{1:i-1})$$

- E.g.,

$$\begin{aligned} p(\text{the, mouse, ate, the, cheese}) &= p(\text{the}) \\ &\quad p(\text{mouse} \mid \text{the}) \\ &\quad p(\text{ate} \mid \text{the, mouse}) \\ &\quad p(\text{the} \mid \text{the, mouse, ate}) \\ &\quad p(\text{cheese} \mid \text{the, mouse, ate, the}). \end{aligned}$$

Autoregressive Language Models

- To generate the entire sequence from an autoregressive LM
 - Sample one token at a time given the tokens generated so far

for $i = 1, \dots, L$:

$$x_i \sim p(x_i | x_{1:i-1})^{1/T}$$

Where $T > 0$ is the temperature parameter that controls the randomness we want from the LM

- $T = 0$ deterministically choose the most likely token at each step i
- $T = 1$ sample from the pure language model using a normal distribution
- $T = \infty$ sample from the uniform distribution over the entire vocabulary

Annealed conditional probability distribution

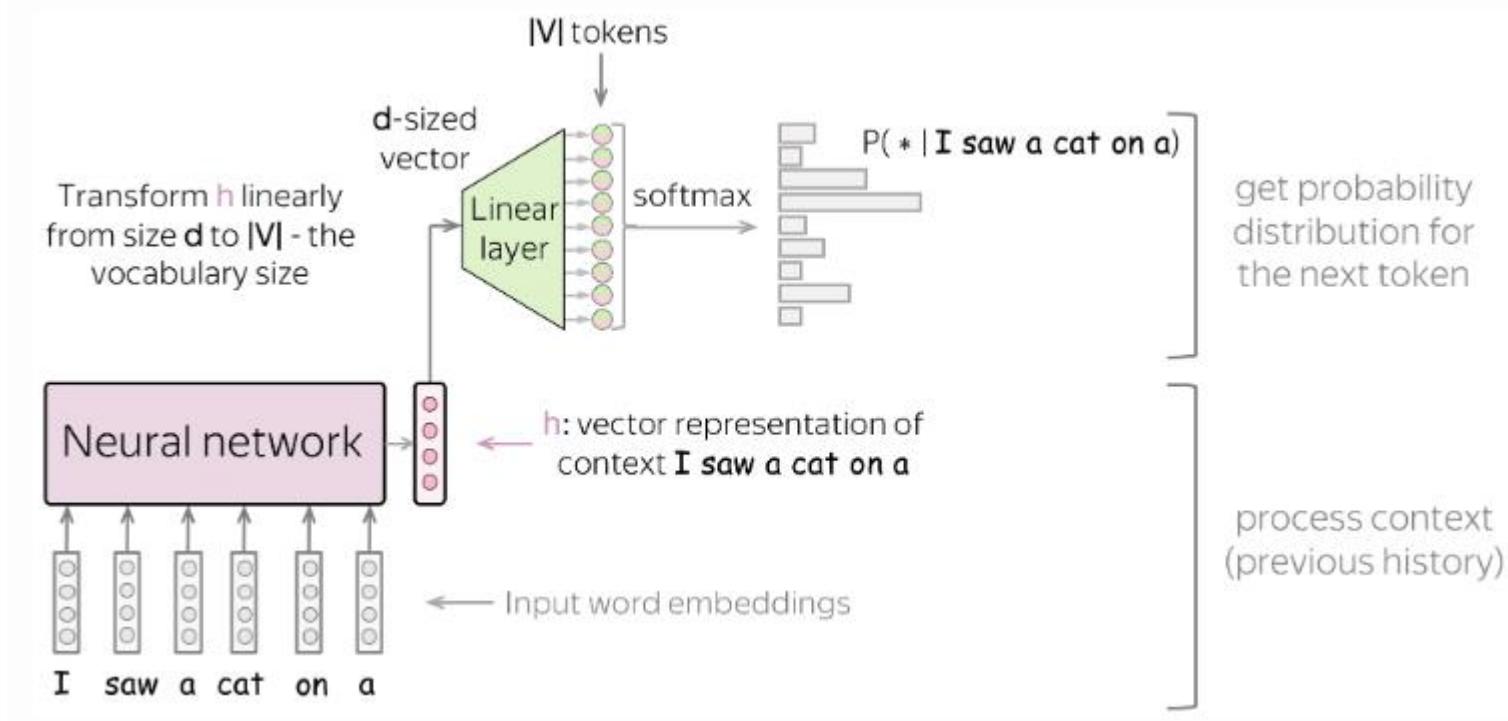
- When T is not equal to 1 the probability distribution need to be re-normalized
 - Otherwise the probabilities do not sum to one

Conditional text generation

- Given a prefix sequence, called **prompt**, generate the rest of the sequence, called **completion**

$\underbrace{\text{the, mouse, ate}}_{\text{prompt}} \xrightarrow{T=0} \underbrace{\text{the, cheese.}}_{\text{completion}}$

Language Model



Language Model

- Model the generative likelihood of word sequences to predict the probabilities of future (or missing) tokens



https://lena-voita.github.io/nlp_course/language_modeling.html

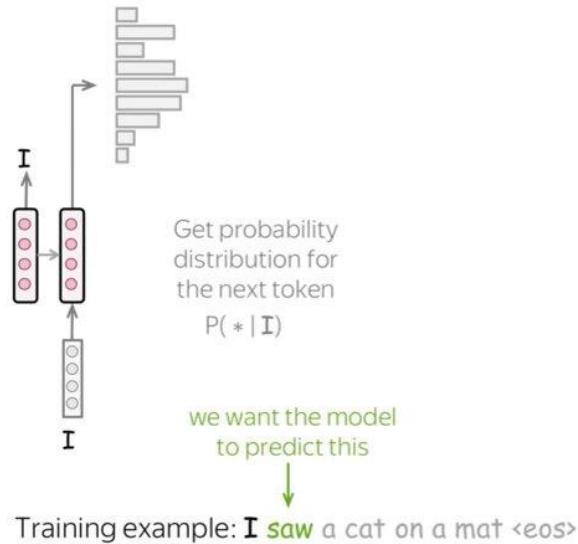
Language Model

- Model the generative likelihood of word sequences to predict the probabilities of future (or missing) tokens



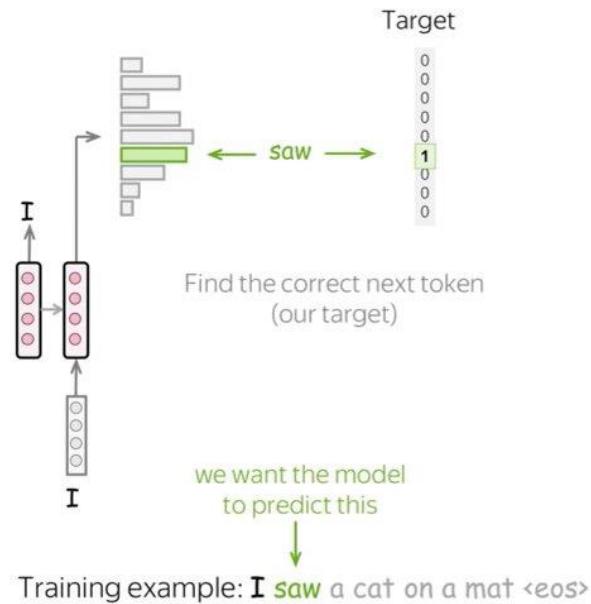
Language Model

- Model the generative likelihood of word sequences to predict the probabilities of future (or missing) tokens



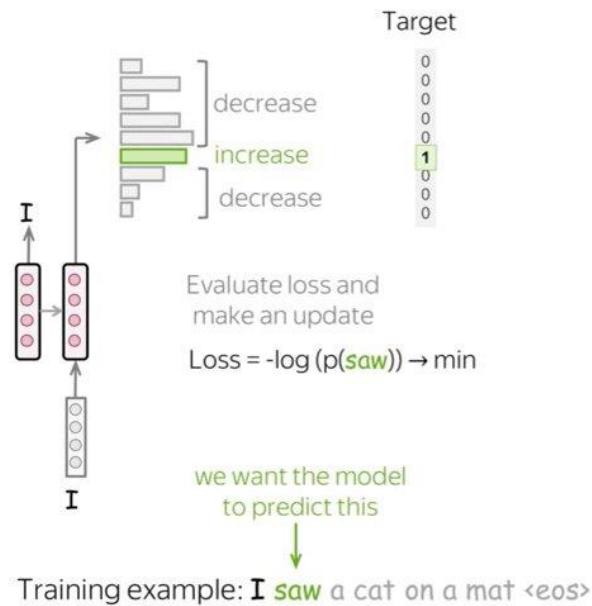
Language Model

- Model the generative likelihood of word sequences to predict the probabilities of future (or missing) tokens



Language Model

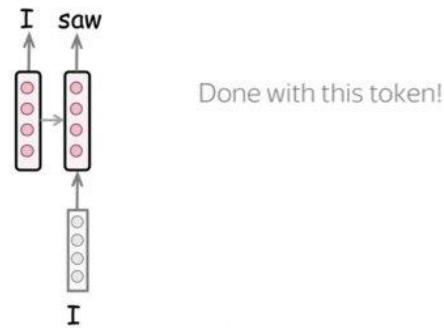
- Model the generative likelihood of word sequences to predict the probabilities of future (or missing) tokens



https://lena-voita.github.io/nlp_course/language_modeling.html

Language Model

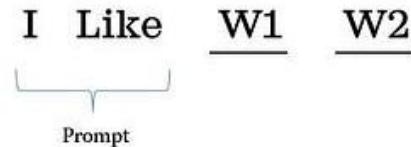
- Model the generative likelihood of word sequences to predict the probabilities of future (or missing) tokens



Training example: **I saw** a cat on a mat <eos>

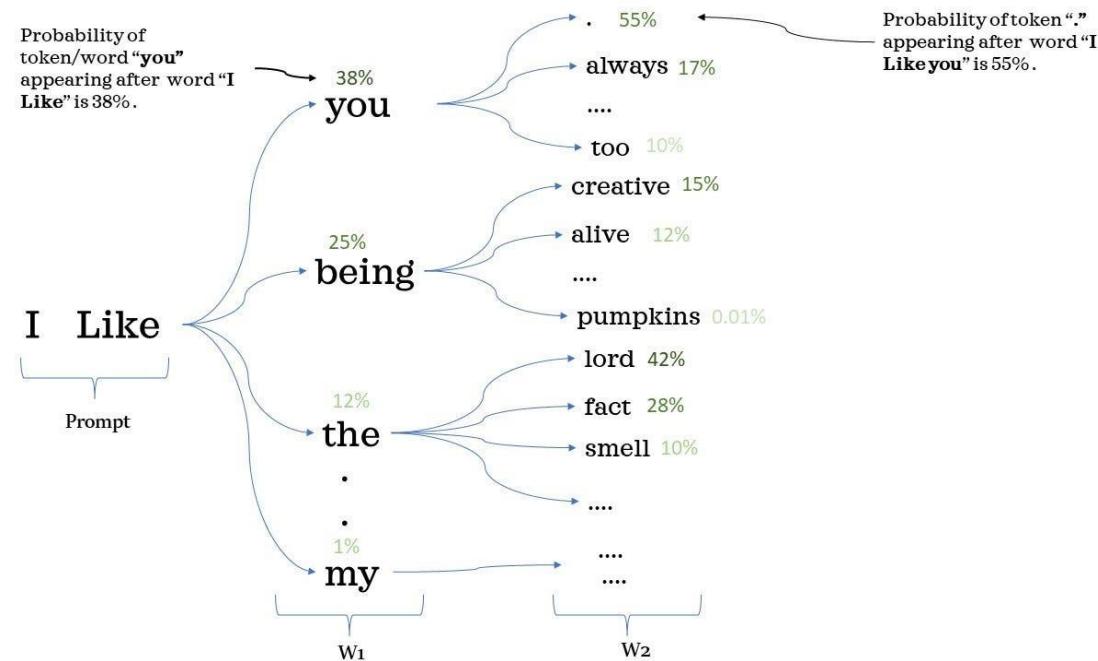
Conditional text generation

- Example



Conditional text generation

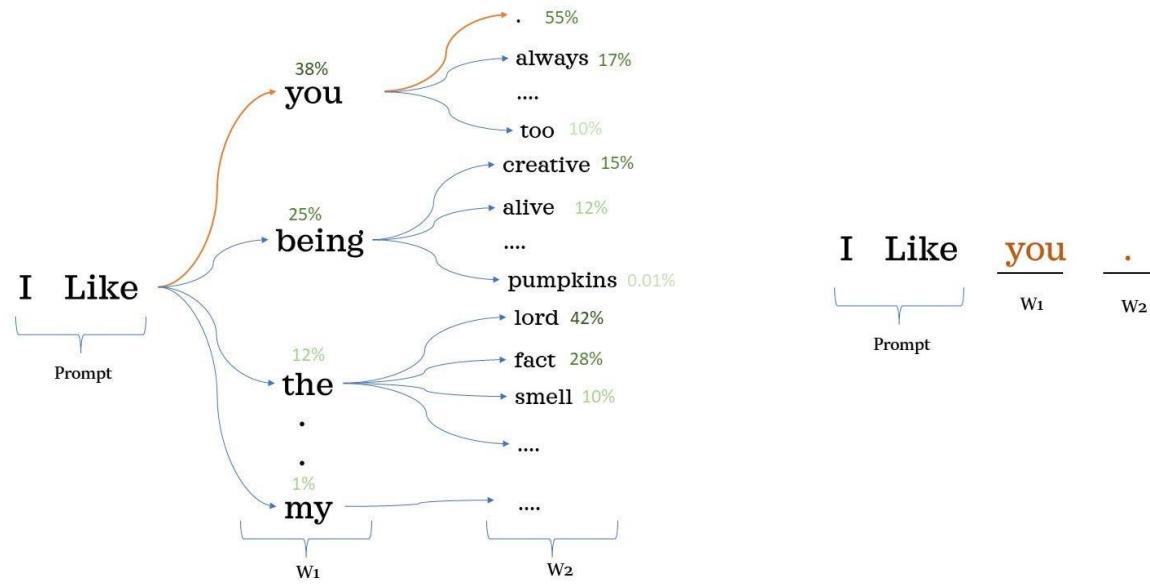
- Example



Demystifying the Temperature Parameter: A Visual Guide to Understanding its Role in Large Language Models. Medium.com

Conditional text generation

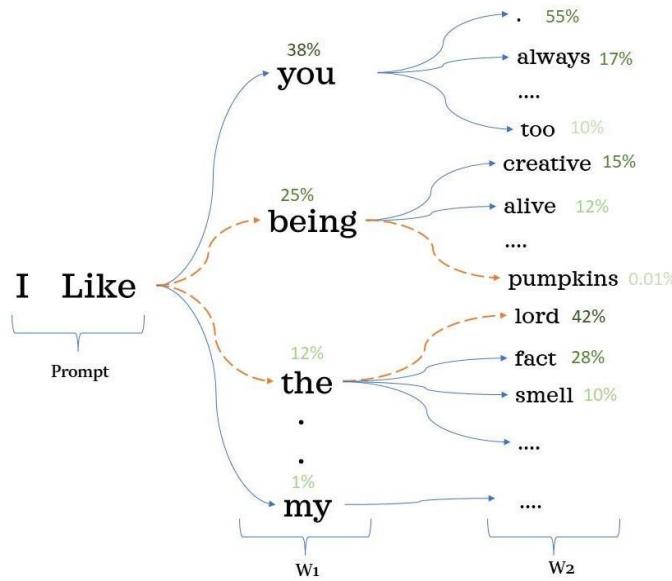
- $T = 0$



Demystifying the Temperature Parameter: A Visual Guide to Understanding its Role in Large Language Models. Medium.com

Conditional text generation

- $T > 0$



I Like being pumpkins
W₁ W₂

OR

I Like the lord
W₁ W₂

What is a Large Language Model?

- A groundbreaking NLP system with advanced language understanding and generation capabilities.
- Enables sophisticated text generation, question answering and conversational abilities.
- Trained on massive amounts of data to comprehend and perform a wide range of tasks.
- Revolutionizes information access, creativity, and natural language processing tasks.

What is a Large Language Model?

The content of the previous slide was generated by ChatGPT

What is a Large Language Model?

Provide me with a list of 4 short sentences (maximum 15 words each) to describe it



Free Research Preview. ChatGPT may produce inaccurate information about people, places, or facts. [ChatGPT May 24 Version](#)

Key LLM features

- Huge Deep Learning models
 - Billions of hyperparameters
- Virtually all state-of-the-art LLMs are based on Transformers
- Trained on massive data
 - TeraBytes of data
 - Trillions of tokens
 - Variety of sources, languages, domains, etc.
- Pretrained using self-supervised and/or semi-supervised learning
 - Self-supervised task: Next Token Prediction

Main LLM categories

- Proprietary vs. opensource
- Model
 - Characteristics
 - Complexity (number of hyperparameters)
- Training data
 - Language-specific vs. multilingual
 - Domain-specific vs. general-purpose

Main LLM categories

- Proprietary vs. opensource



Main LLM categories

- Language support
 - Apparently great but...

Model	Model Type			Model Sizes	Datasets	# Languages	LLM Objectives
	Enc-Only	Dec-Only	Enc-Dec				
XLM-R (Conneau et al., 2020)	✓			250M, 560M	CommonCrawl	100	MLM
mBERT (Devlin et al., 2019)	✓			178M	Wikipedia	104	MLM
mDeBERTa v3 (He et al., 2021)	✓			278M	CC100	100	RTD w/ GDES
mBART-50 (Tang et al., 2020)			✓	611M	CC25, ML50	50	RTD w/ GDES
XGLM (Lin et al., 2021)		✓		564M, 1.7B, 2.9B, 4.5B, 7.5B	CommonCrawl	30	MLM
BLOOMZ (Muennighoff et al., 2022)		✓		560M, 1.1B, 1.7B, 3B, 7.1B	ROOTS, xP3	46	Instruction Tuned
mT0 (Muennighoff et al., 2022)			✓	300M, 580M, 1.2B, 3.7B, 13B	mC4, xP3	~120	Instruction Tuned
ChatGPT (Bang et al., 2023)	✓			-	-	-	RLHF

Main LLM categories

- Language support
 - What's behind this?

Language	Percent	Language	Percent
en	89.70%	uk	0.07%
unknown	8.38%	ko	0.06%
de	0.17%	ca	0.04%
fr	0.16%	sr	0.04%
sv	0.15%	id	0.03%
zh	0.13%	cs	0.03%
es	0.13%	fi	0.03%
ru	0.13%	hu	0.03%
nl	0.12%	no	0.03%
it	0.11%	ro	0.03%
ja	0.10%	bg	0.02%
pl	0.09%	da	0.02%
pt	0.09%	sl	0.01%
vi	0.08%	hr	0.01%

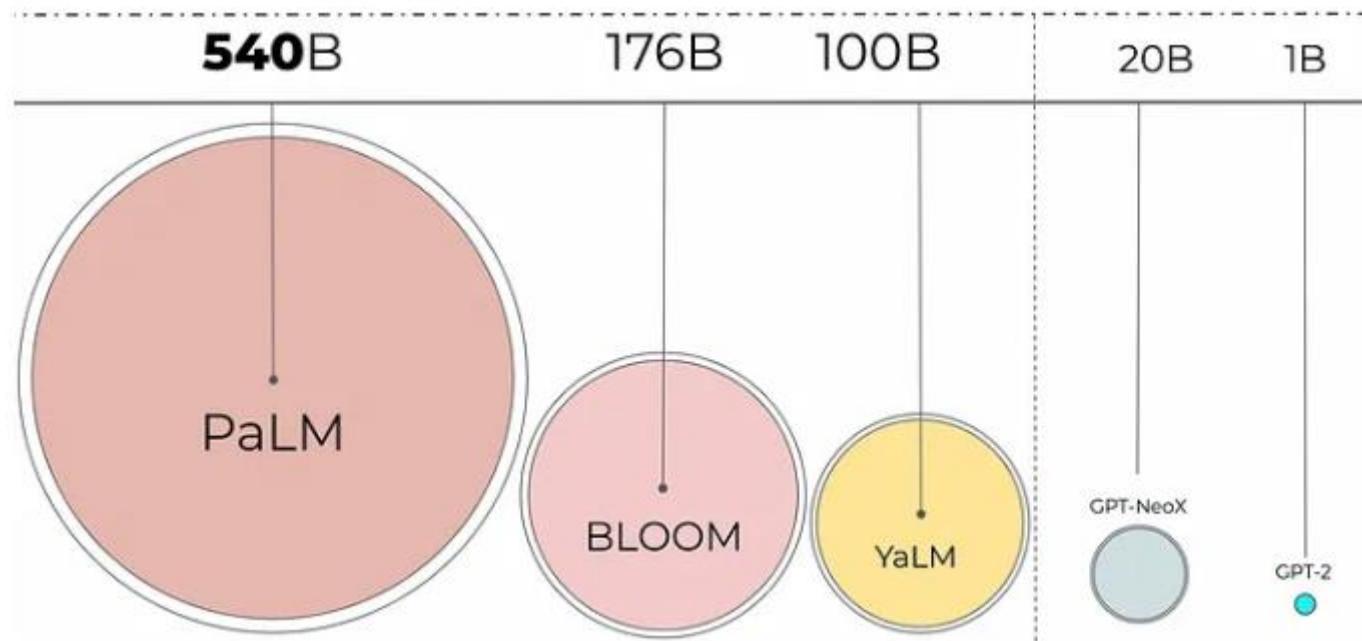
Which languages does ChatGPT (proprietary LLM) understand?

Albanian, Albania	Greek, Greece	Montenegrin, Montenegro
Arabic, Arab World	Gujarati, India	Nepali, Nepal
Armenian, Armenia	Haryanvi, India	Norwegian, Norway
Awadhi, India	Hindi, India	Oriya, India
Azerbaijani, Azerbaijan	Hungarian, Hungary	Pashto, Afghanistan
Bashkir, Russia	Indonesian, Indonesia	Persian (Farsi), Iran
Basque, Spain	Irish, Ireland	Polish, Poland
Belarusian, Belarus	Italian, Italy	Portuguese, Portugal
Bengali, Bangladesh	Japanese, Japan	Punjabi, India
Bhojpuri, India	Javanese, Indonesia	Rajasthani, India
Bosnian, Bosnia and Herzegovina	Kannada, India	Romanian, Romania
Brazilian Portuguese, Brazil	Kashmiri, India	Russian, Russia
Bulgarian, Bulgaria	Kazakh, Kazakhstan	Sanskrit, India
Cantonese (Yue), China	Konkani, India	Santali, India
Catalan, Spain	Korean, South Korea	Serbian, Serbia
Chhattisgarhi, India	Kyrgyz, Kyrgyzstan	Sindhi, Pakistan
Chinese, China	Latvian, Latvia	Sinhala, Sri Lanka
Croatian, Croatia	Lithuanian, Lithuania	Slovak, Slovakia
Czech, Czech Republic	Macedonian, North	Slovene, Slovenia
Danish, Denmark	Macedonia	Slovenian, Slovenia
Dogri, India	Maithili, India	Ukrainian, Ukraine
Dutch, Netherlands	Malay, Malaysia	Urdu, Pakistan
English, United Kingdom	Maltese, Malta	Uzbek, Uzbekistan
Estonian, Estonia	Mandarin, China	Vietnamese, Vietnam
Faroese, Faroe Islands	Mandarin Chinese, China	Welsh, Wales
Finnish, Finland	Marathi, India	Wu, China
French, France	Marwari, India	
Galician, Spain	Min Nan, China	
Georgian, Georgia	Moldovan, Moldova	
German, Germany	Mongolian, Mongolia	

<https://botpress.com/blog/list-of-languages-supported-by-chatgpt>

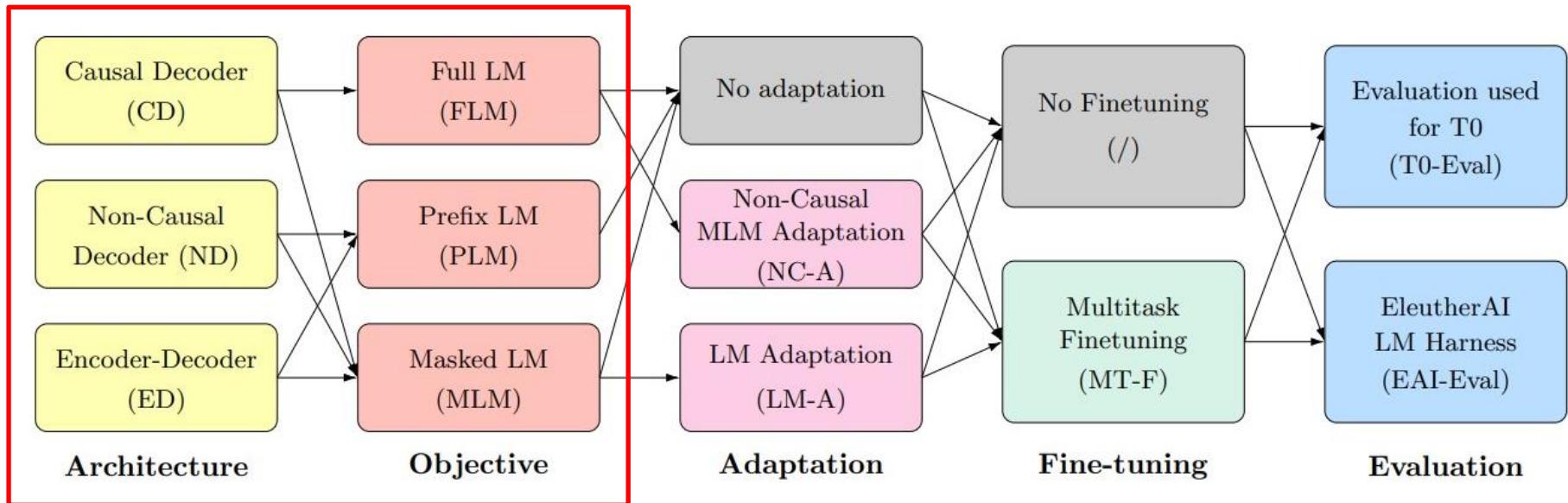
Main LLM categories

- Model complexity
 - Number of hyperparameters



<https://arxiv.org/pdf/2305.14235.pdf>

LLM architectures



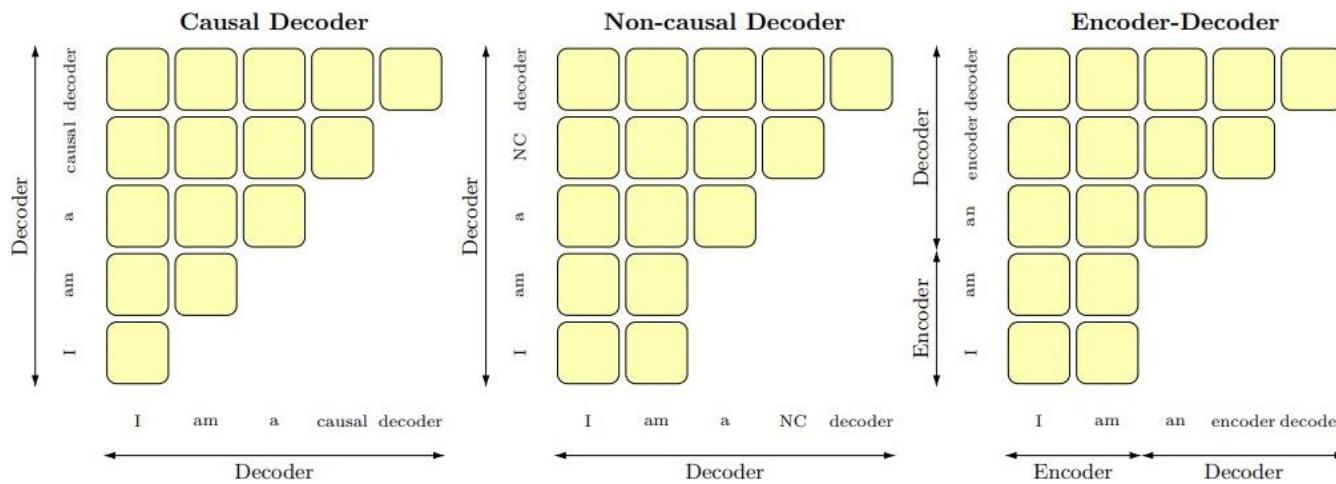
What Language Model Architecture and Pretraining Objective Work Best for Zero-Shot Generalization? <https://arxiv.org/abs/2204.05832>

LLM architectures

- Encoder-decoder vs. decoder-only
- All state-of-the-art language models over 100 billions of parameters are **autoregressive decoder-only models**
 - They achieve better **zero-shot** performance than encoder-decoder architectures

LLM architectures

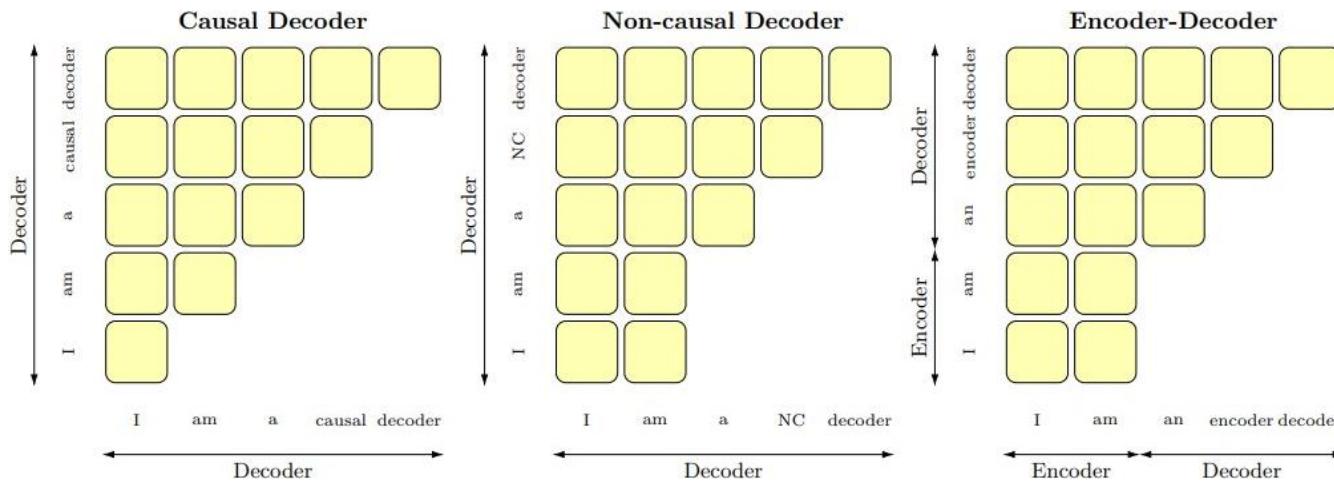
- **Causal Decoder-only:** conditioning is based on past tokens only
 - Predict the next token in the sequence
 - All tokens are processed in an equivalent fashion
 - Backbone of GPT series



What Language Model Architecture and Pretraining Objective Work Best for Zero-Shot Generalization? <https://arxiv.org/abs/2204.05832>

LLM architectures

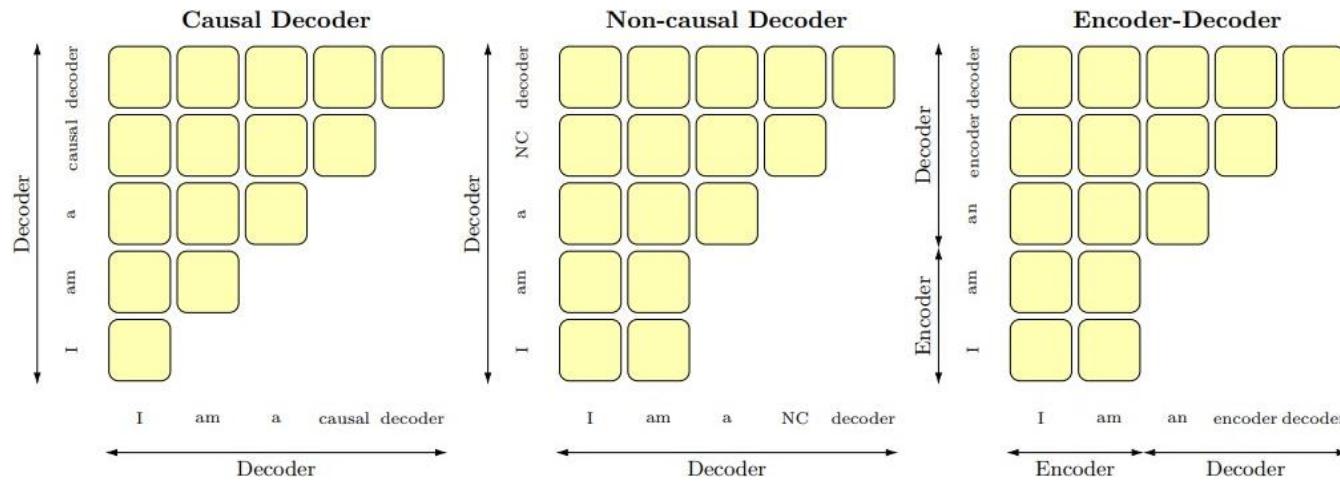
- **Non-causal Decoder-only:** the self-attention mask is modified so that attention is not restricted to past tokens
 - The region of the input sequence corresponding to conditioning information has a non-causal mask (eventually not restricted to past tokens)
 - Limited adoption



What Language Model Architecture and Pretraining Objective Work Best for Zero-Shot Generalization? <https://arxiv.org/abs/2204.05832>

LLM architectures

- **Encoder-Decoder:** the prediction of the target sequence is conditioned on the output of the encoder
 - relies on cross-attention (such as BART or T5)
 - Based on BERT-like encoders



What Language Model Architecture and Pretraining Objective Work Best for Zero-Shot Generalization? <https://arxiv.org/abs/2204.05832>

LLM pre-training objectives

- **Full Language Modeling:** given the previous tokens, the model is trained to predict the following one

Full Language Modeling

May the force be with you

targets

Prefix Language Modeling

May the force be with you

targets

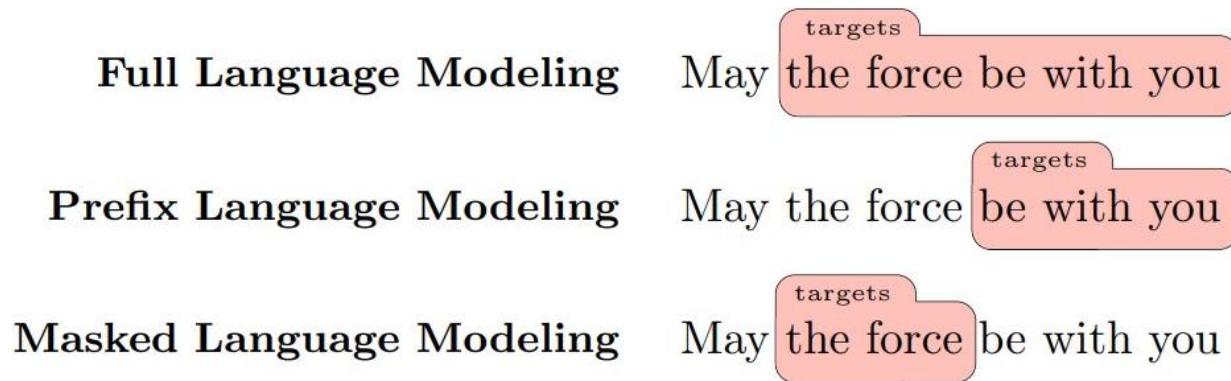
Masked Language Modeling

May the force be with you

targets

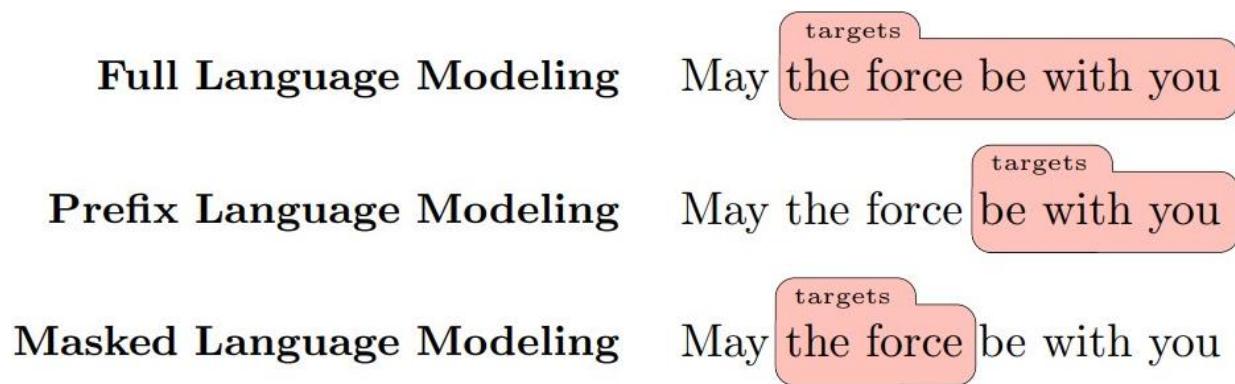
LLM pre-training objectives

- **Prefix Language Modeling:** define a prefix where the attention mask is allowed to be non-causal
 - The model is trained to predict each token outside the prefix given all previous tokens
 - For inference, the prefix is the input text
 - For pretraining, the prefix is commonly chosen randomly



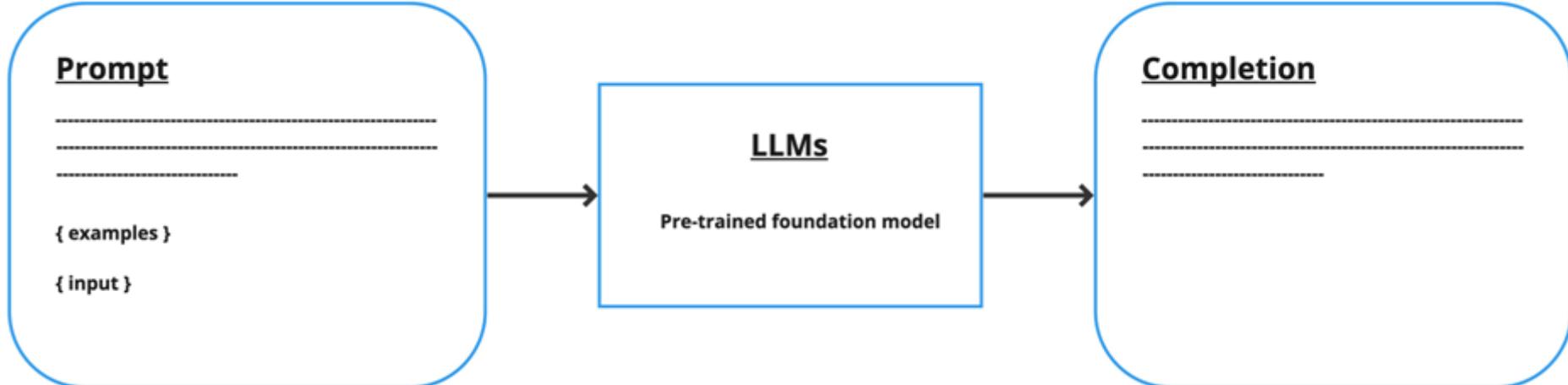
Pre-training objectives

- **Masked Language Modeling:** (spans of) tokens in the input text are replaced/corrupted with a special mask token
 - The model is trained to predict/reconstruct the missing/ corrupted tokens
 - Text span masking or corruption as in BART
 - For pretraining, the prefix is commonly chosen randomly



LLM adaptation

- The process of taking a LM and turning it into a task model
- It takes as input
 - A natural language description
 - A set of training instances (input-output pairs)



<https://docs.sia.so/sia/llmops/what-steps-are-involved-in-llmops/adaptation-to-downstream-tasks>

LLM adaptation: training

- **Training via supervised learning:** train a new model that maps inputs to outputs, either by
 - **Probing:** creating a new task-specific model that uses the language model as latent features
 - Train on top of a frozen LM
 - **Fine-tuning:** starting with the language model and updating it based on the training instances
 - **lightweight fine-tuning:** optimize just few parameters
 - A mix of fine-tuning and probing
- Challenged by data overfitting
 - Few training instances compared to the network complexity

<https://stanford-cs324.github.io/winter2022/lectures/capabilities/>

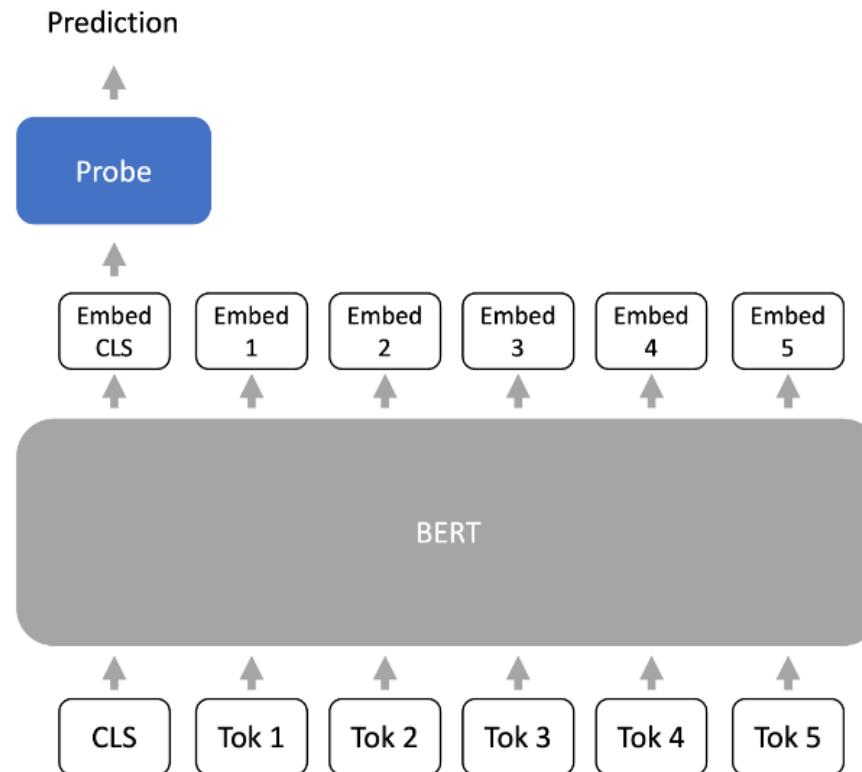
LLM adaptation: training

- Rationale
 - LLMs are trained in a task-agnostic way
 - Downstream tasks can be various: next token prediction can be not sufficient!
- Main strategies
 - **Formatting:** use a particular formatting style tailored to the downstream task
 - **Topic shift:** adapt the model to a specific domain
 - **Temporal shift:** when the downstream task requires new model that was not available during pre-training

<https://stanford-cs324.github.io/winter2022/lectures/capabilities/>

LLM adaptation: training

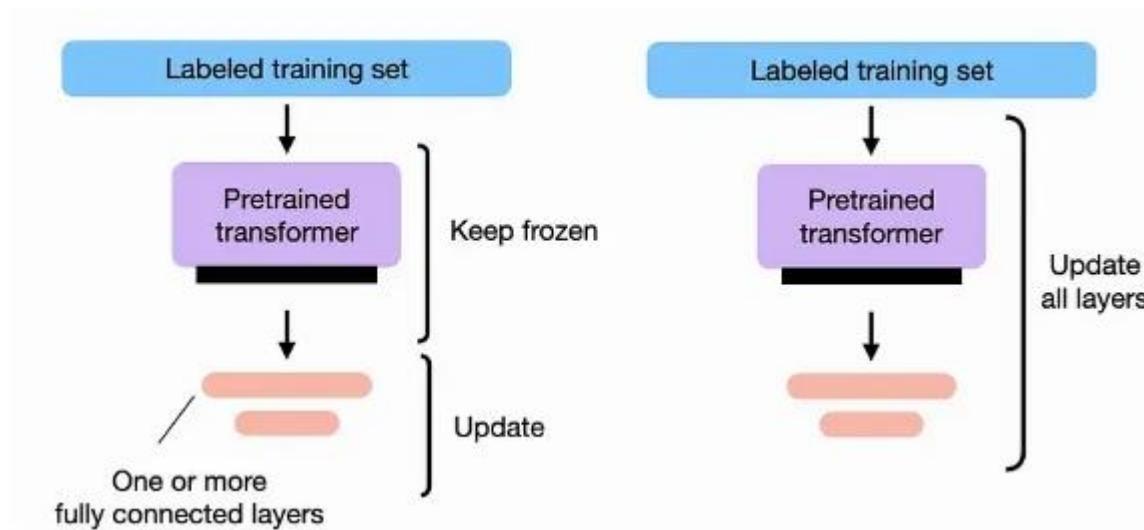
- Example of **probing**: train a prediction head from the last layer of representation of the LM to the output



<https://stanford-cs324.github.io/winter2022/lectures/capabilities/>

LLM adaptation: training

- Examples of **fine-tuning** and **lightweight fine-tuning** via supervised learning



LLM adaptation: training

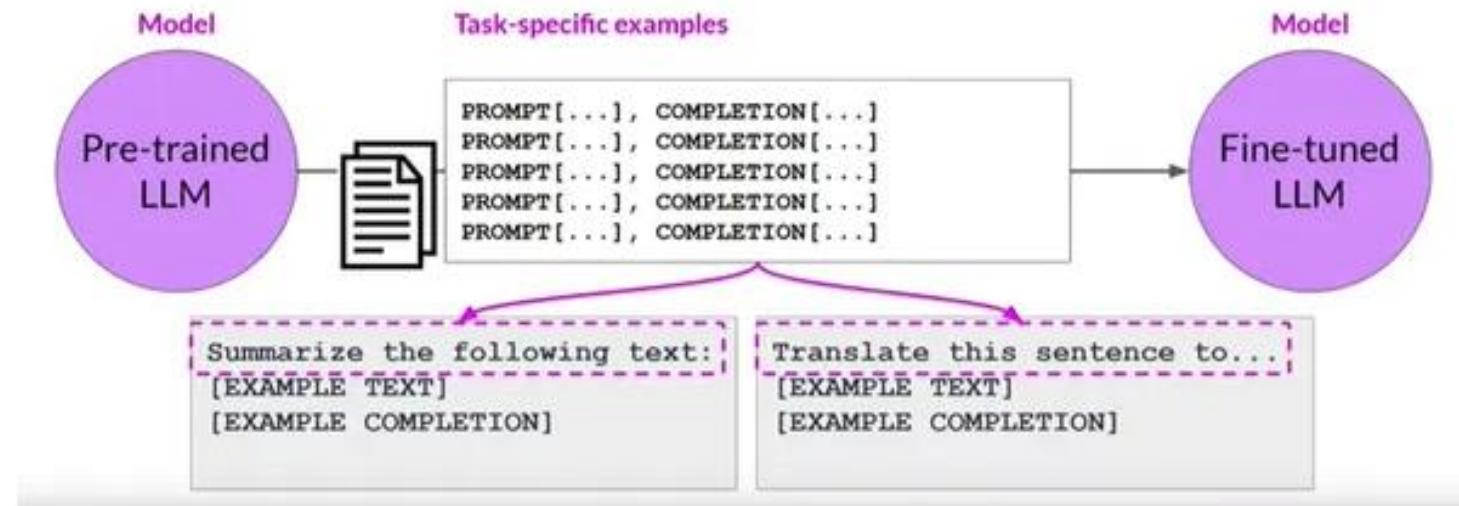
- **Fine-tuning:** uses the language model parameters as initialization for optimization
 - Requires storing a LLM specialized for every downstream task
 - Expensive but more effective than probing
 - Can rely on **Reinforcement Learning** as well to account human preferences
- **Lightweight fine-tuning:** achieve the same expressivity as full fine-tuning while not requiring us to store the full language model for every task

LLM adaptation: prompting

- **Prompting (in-context learning):** Construct a prompt (a string based on the description and training instances) or a set of prompts, feed those into a language model to obtain completions.
 - **Zero-shot learning:** number of training examples is 0
 - **One-shot learning:** number of training examples is 1
 - **Few-shot learning:** number of training examples is few

LLM adaptation: prompting

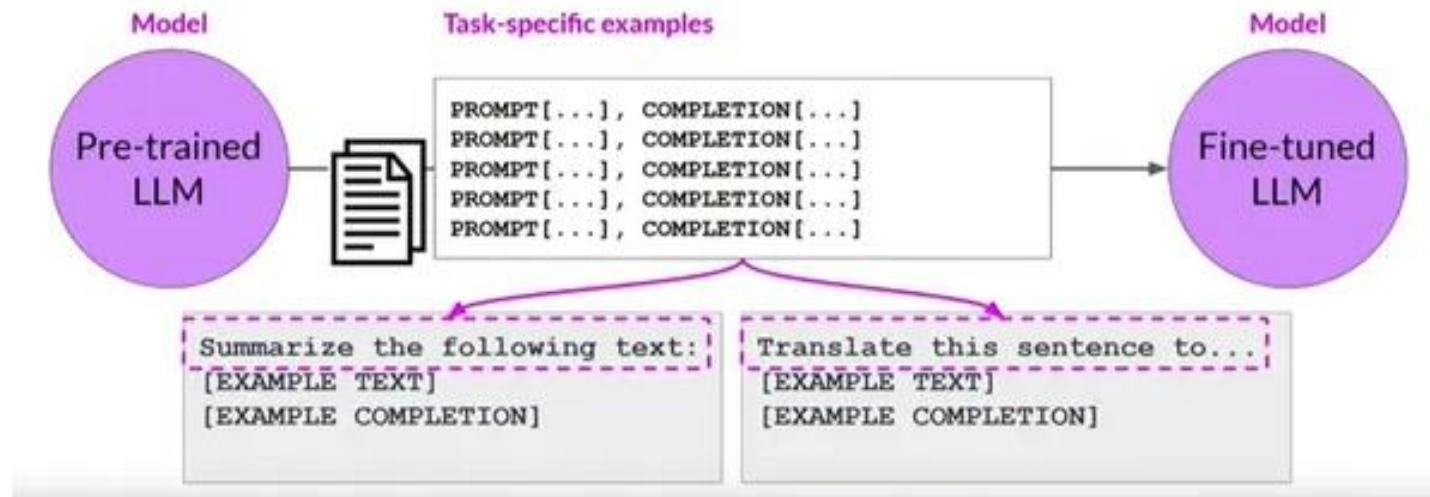
- **Prompting (in-context learning):** example of **single-task** for text summarization



<https://stanford-cs324.github.io/winter2022/lectures/capabilities/>

LLM adaptation: prompting

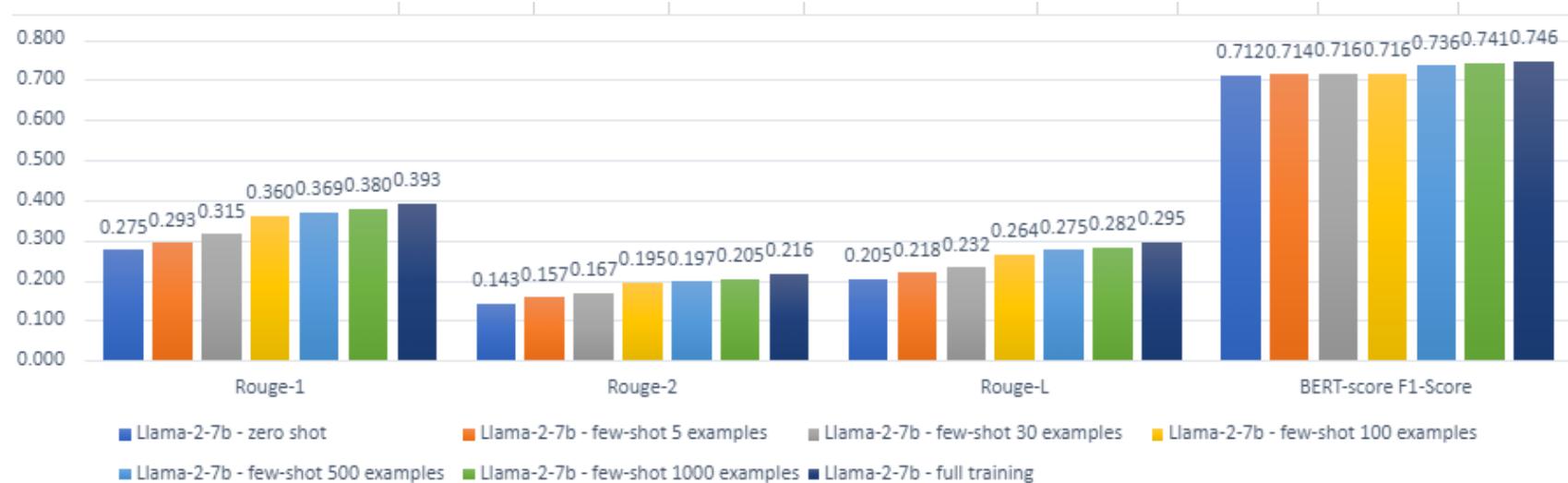
- **Prompting (in-context learning):** example of **multi-task** for text summarization and machine translation



<https://stanford-cs324.github.io/winter2022/lectures/capabilities/>

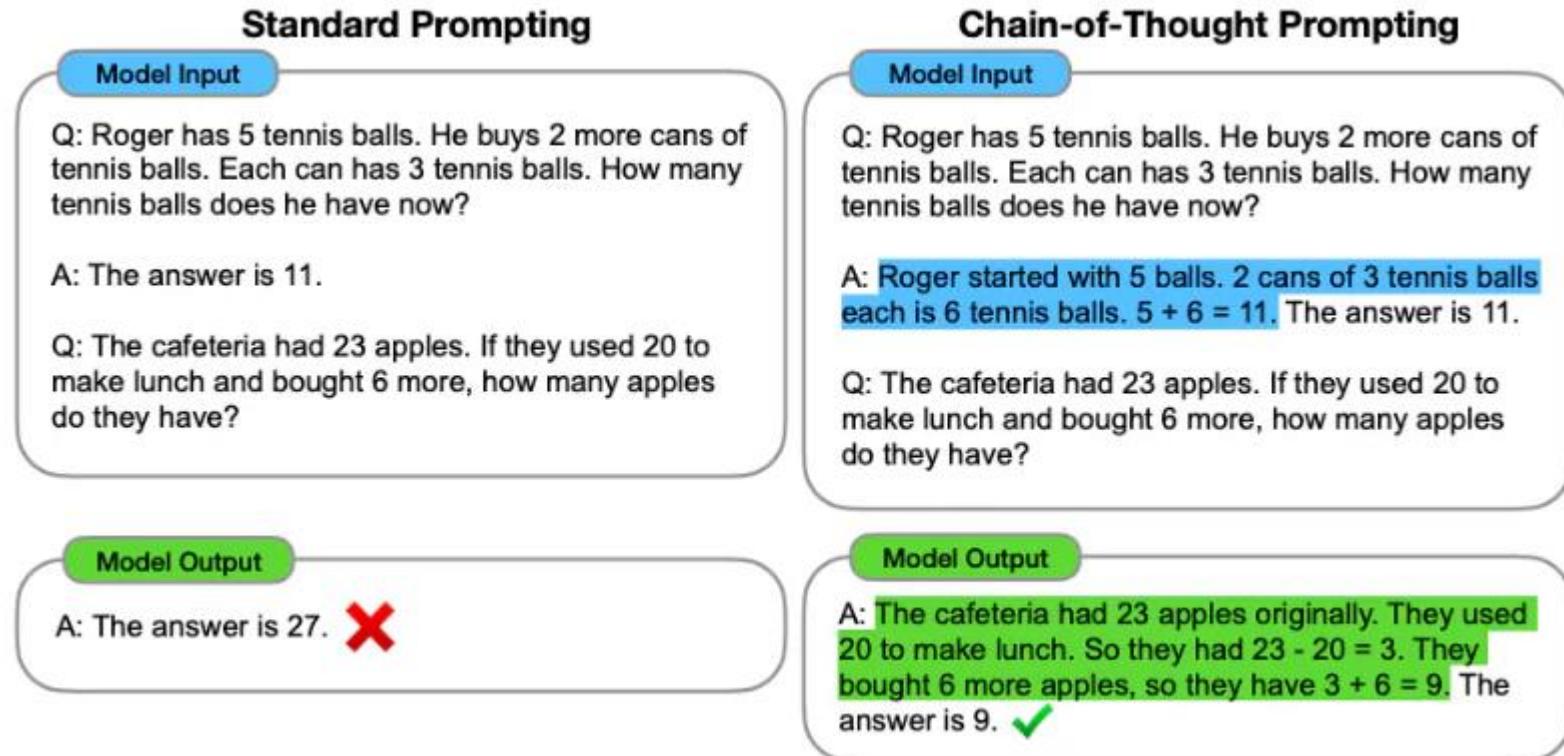
Few-shot learning: how many examples do LLMs require?

- LLM: LIAMA 2
- Task: abstractive summarization
- Data: legal news articles (proprietary)
- Language: Italian
- Metrics: Rouge (syntactic), BERT Score (semantic)
- Baseline (LexRank) performance: Rouge-1 0.356, Rouge-2: 0.194, Rouge-L: 0.26, BERTScore: 0.733



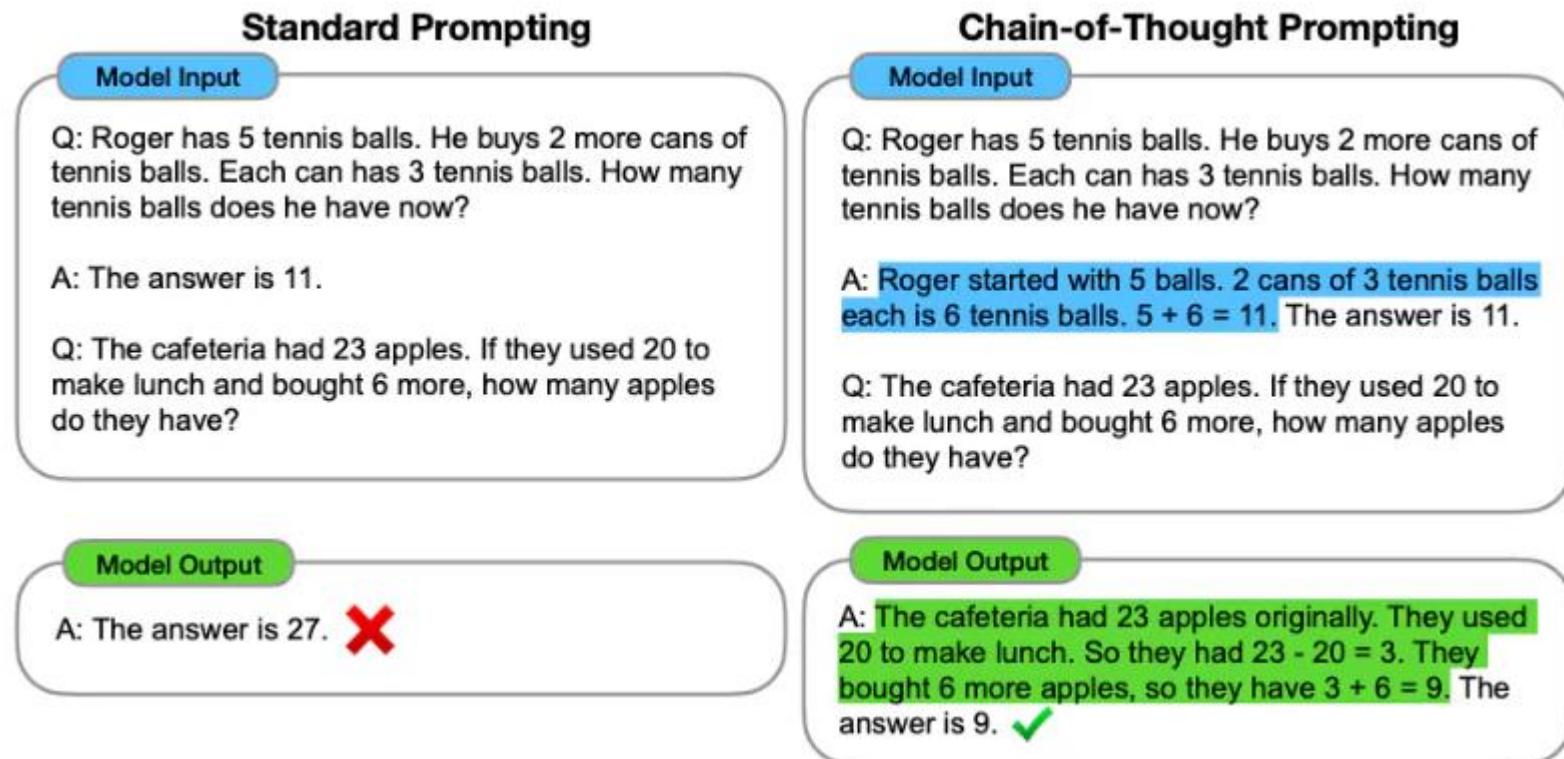
Chain-Of-Thoughts prompting

- Enable complex reasoning capabilities through intermediate reasoning steps.



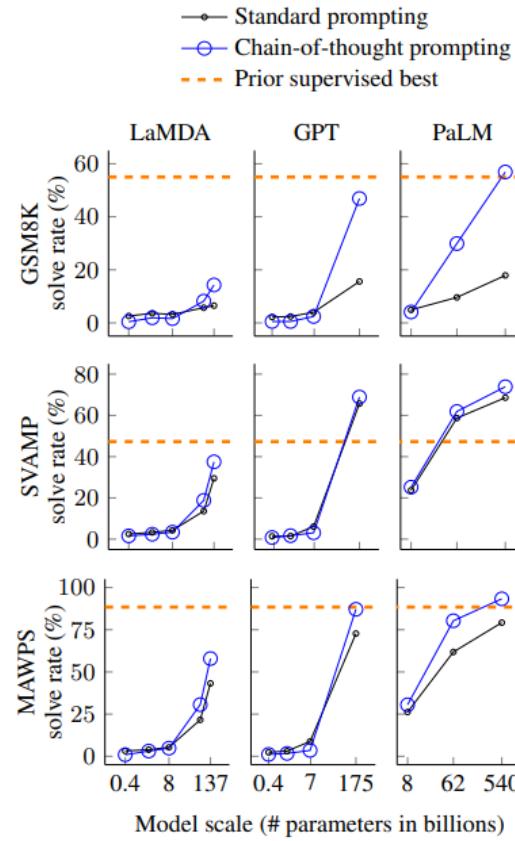
Chain-Of-Thoughts prompting

- Enable complex reasoning capabilities through intermediate reasoning steps
 - E.g., tackle complex arithmetic, commonsense, and symbolic reasoning tasks



Chain-Of-Thoughts prompting

- Enable complex reasoning capabilities through intermediate reasoning steps
 - E.g., tackle complex arithmetic, commonsense, and symbolic reasoning tasks



Chain-Of-Thoughts prompting: examples of <input,COT,output>

Math Word Problems (free response)

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

Math Word Problems (multiple choice)

Q: How many keystrokes are needed to type the numbers from 1 to 500?
Answer Choices: (a) 1156 (b) 1392 (c) 1480 (d) 1562 (e) 1788

A: There are 9 one-digit numbers from 1 to 9. There are 90 two-digit numbers from 10 to 99. There are 401 three-digit numbers from 100 to 500. $9 + 90(2) + 401(3) = 1392$. The answer is (b).

CSQA (commonsense)

Q: Sammy wanted to go to where the people were. Where might he go?
Options: (a) race track (b) populated areas (c) desert (d) apartment (e) roadblock

A: The answer must be a place with a lot of people. Race tracks, desert, apartments, and roadblocks don't have a lot of people, but populated areas do. So the answer is (b).

StrategyQA

Q: Yes or no: Would a pear sink in water?

A: The density of a pear is about 0.6 g/cm³, which is less than water. Thus, a pear would float. So the answer is no.

Date Understanding

Q: The concert was scheduled to be on 06/01/1943, but was delayed by one day to today. What is the date 10 days ago in MM/DD/YYYY?

A: One day after 06/01/1943 is 06/02/1943, so today is 06/02/1943. 10 days before today is 05/23/1943. So the answer is 05/23/1943.

Sports Understanding

Q: Is the following sentence plausible? "Joao Moutinho caught the screen pass in the NFC championship."

A: Joao Moutinho is a soccer player. The NFC championship is part of American football, not soccer. So the answer is no.

SayCan (Instructing a robot)

Human: How would you bring me something that isn't a fruit?

Explanation: the user wants something to eat that isn't a fruit. An energy bar is not a fruit, so I will bring the user an energy bar.

Plan: 1. find(energy bar) 2. pick(energy bar) 3. find(user) 4. put(energy bar) 5. done().

Last Letter Concatenation

Q: Take the last letters of the words in "Lady Gaga" and concatenate them.

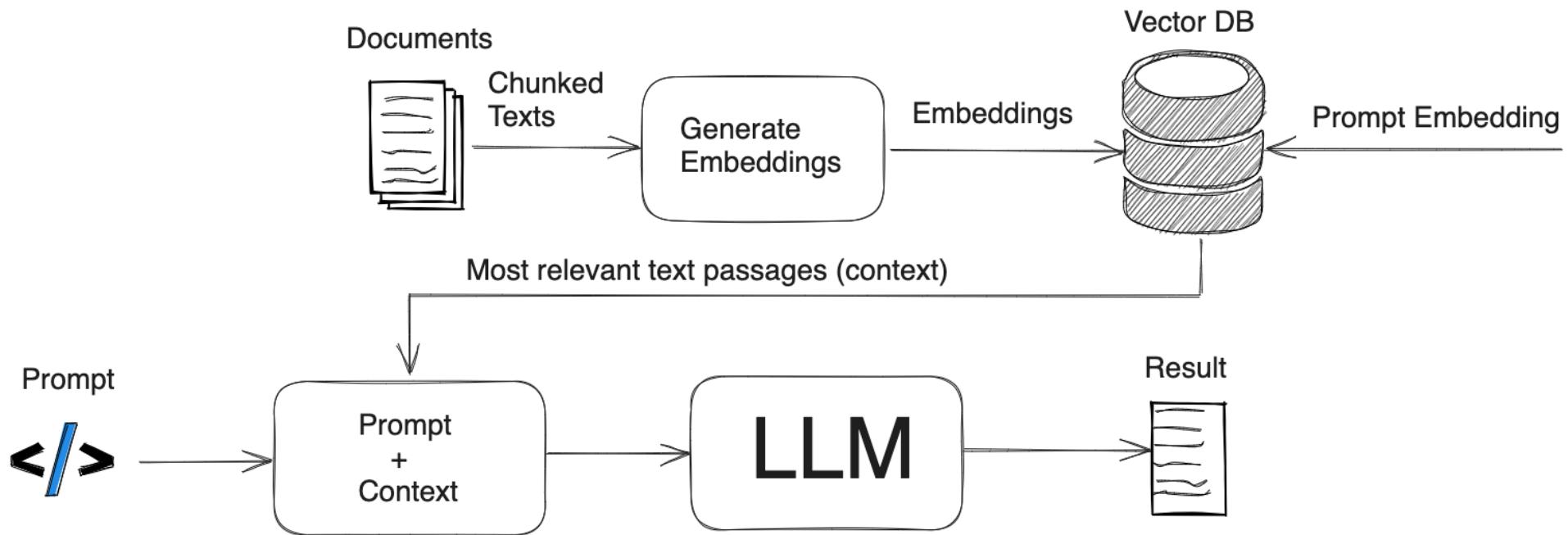
A: The last letter of "Lady" is "y". The last letter of "Gaga" is "a". Concatenating them is "ya". So the answer is ya.

Coin Flip (state tracking)

Q: A coin is heads up. Maybelle flips the coin. Shalonda does not flip the coin. Is the coin still heads up?

A: The coin was flipped by Maybelle. So the coin was flipped 1 time, which is an odd number. The coin started heads up, so after an odd number of flips, it will be tails up. So the answer is no.

Retrieval Augmented Generation



<https://ai.meta.com/blog/retrieval-augmented-generation-streamlining-the-creation-of-intelligent-natural-language-processing-models/>

Retrieval Augmented Generation

- Combine LLMs with content retrieval systems
- Augment the capabilities of LLMs using a hybrid approach
- Particularly relevant to companies that do not want to disclose enterprise data
- RAG looks and acts like a standard seq2seq model
 - They take in one sequence and outputs a corresponding sequence.
- There is an intermediary step though, which differentiates and elevates RAG above the usual seq2seq methods.
 - Rather than passing the input directly to the generator, RAG instead uses the input to retrieve a set of relevant documents

<https://ai.meta.com/blog/retrieval-augmented-generation-streamlining-the-creation-of-intelligent-natural-language-processing-models/>

Retrieval Augmented Generation

- Given the prompt “When did the first mammal appear on Earth?” for instance, RAG might surface documents for “Mammal,” “History of Earth,” and “Evolution of Mammals.” These supporting documents are then concatenated as context with the original input and fed to the seq2seq model that produces the actual output.
- RAG thus has two sources of knowledge: the knowledge that seq2seq models store in their parameters (parametric memory) and the knowledge stored in the corpus from which RAG retrieves passages (nonparametric memory).

<https://ai.meta.com/blog/retrieval-augmented-generation-streamlining-the-creation-of-intelligent-natural-language-processing-models/>

Examples of famous LLMs

- GPT-* family (GPT-3, GPT-3.5, ChatGPT, GPT-4)
- LaMDA & LaMDA 2
- PaLM & PaLM 2
- BLOOM
- Galactica
- LLaMA
- ...

GPT-* family

- **G**enerative **P**re-trained **T**ransformers
- Developed by: OpenAI
- Release date:
 - GPT-3: May 2020 – GPT-3.5: March 2022
ChatGPT: November 2022 – GPT-4: March 2023
- Number of parameters:
 - GPT-3, GPT-3.5, ChatGPT: 175 billion
 - GPT-4: undisclosed, 1 trillion (estimated)

GPT-* family

- Training corpus:
 - GPT-3: 499 billion tokens
 - GPT-3.5, ChatGPT, GPT-4: undisclosed, trillion tokens (estimated)
- Paper:
 - GPT-3: <https://arxiv.org/abs/2005.14165>
 - GPT-3.5, ChatGPT: not available
 - GPT-4: <https://arxiv.org/abs/2303.08774>

Note: GPT-3.5, ChatGPT and GPT-4 architecture details are not public

ChatGPT

- Built upon GPT-3.5, fine-tuned for conversational usage
- Joint use of supervised learning and reinforcement learning
- Reinforcement Learning from Human Feedback (RLHF)
 - 1st step: supervised fine-tuning
 - 2nd step: reinforcement learning

ChatGPT – Reinforcement Learning from Human Feedback

- **Supervised fine-tuning**

- Human trainers provide conversations in which they play both sides (user and AI assistant)
- Give the trainers access to model-written suggestions to help them compose their responses
- New human-collected dataset mixed with the InstructGPT dataset

ChatGPT – Reinforcement Learning from Human Feedback

- **Reinforcement learning**

- Collect comparison data by randomly selecting a model-written message and sampling several alternative completions (from previously collected conversations)
- Human trainers rank responses
- Create “reward models” from human rankings that are used to fine-tune the model further by several iterations of Proximal Policy Optimization (PPO)

ChatGPT – Reinforcement Learning from Human Feedback

Step 1

Collect demonstration data and train a supervised policy.

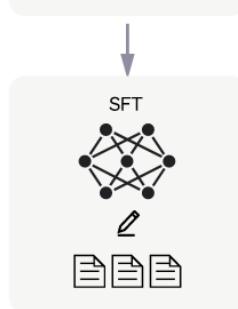
A prompt is sampled from our prompt dataset.



A labeler demonstrates the desired output behavior.



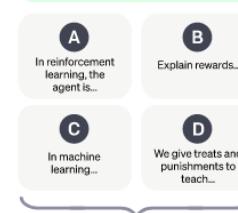
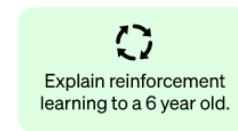
This data is used to fine-tune GPT-3.5 with supervised learning.



Step 2

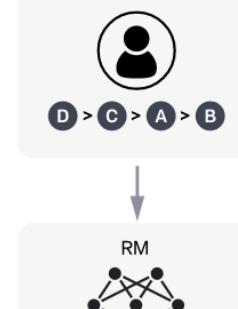
Collect comparison data and train a reward model.

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 the PPO reinforcement learning algorithm.

A new prompt is sampled from the dataset.



The PPO model is initialized from the supervised policy.



The policy generates an output.



The reward model calculates a reward for the output.



The reward is used to update the policy using PPO.

Source: <https://openai.com/blog/chatgpt>

GPT-4

- **Multimodal** large language model: text & **images** as input
- Trained using both publicly available data and “data licensed from third-party providers”
- Fine-tuned using RLHF (as in ChatGPT)
- Context window: from 2,048 (GPT-3) and 4,096 (GPT-3.5) to 8,192 and 32,768 tokens (around 52 pages of text)

GPT-4

- Further details are not publicly available

“Given both the competitive landscape and the safety implications of large-scale models like GPT-4, this report contains no further details about the architecture (including model size), hardware, training compute, dataset construction, training method, or similar.”

GPT-4 – Example of visual input

User What is funny about this image? Describe it panel by panel.



Source: <https://www.reddit.com/r/hmmm/comments/ubab5v/hmmm/>

GPT-4 The image shows a package for a "Lightning Cable" adapter with three panels.

Panel 1: A smartphone with a VGA connector (a large, blue, 15-pin connector typically used for computer monitors) plugged into its charging port.

Panel 2: The package for the "Lightning Cable" adapter with a picture of a VGA connector on it.

Panel 3: A close-up of the VGA connector with a small Lightning connector (used for charging iPhones and other Apple devices) at the end.

The humor in this image comes from the absurdity of plugging a large, outdated VGA connector into a small, modern smartphone charging port.

GPT-4 – Example of visual input

User Answer question I.1.a. Think step-by-step.

I. Principe de la détection de rayonnement avec un bolomètre

Comme illustré sur la figure 1 un bolomètre est constitué d'un absorbeur qui reçoit le rayonnement que l'on désire détecter. Sa température T , supposée uniforme, est mesurée à l'aide d'un thermomètre incorporé, constitué d'un matériau conducteur dont la résistance $R(T)$ varie avec la température T ; cette variation est caractérisée par le coefficient $\alpha = \frac{1}{R} \frac{dR}{dT}$. L'ensemble possède la capacité thermique C_{th} .

Un barreau, conducteur thermique, homogène, de longueur L , de section S et de conductivité thermique λ et sans échanges thermiques latéraux, relie le bolomètre à un thermostat de température T_b fixe.

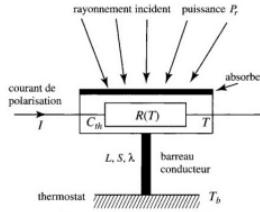


Figure 1 - Schéma d'un bolomètre

I.1. Dans cette première partie, on étudie le comportement du bolomètre en régime stationnaire, à la température de fonctionnement $T = T_0$.

I.1.a) Exprimer la température $T(x)$ en tout point du barreau conducteur repéré par son abscisse x ; on prendra $T(0) = T_0$ et $T(L) = T_b$.

GPT-4

To answer question I.1.a, we need to express the temperature $T(x)$ in every point of the conductive bar, which is characterized by the abscissa x. We are given that $T(0) = T_0$ and $T(L) = T_b$.

For a stationary regime, the heat equation for the one-dimensional heat conduction in the bar is:

$$d^2T/dx^2 = 0$$

Integrating once, we get:
 $dT/dx = C_1$

Integrating a second time:
 $T(x) = C_1 x + C_2$

To determine the constants C_1 and C_2 , we use the boundary conditions:
 $T(0) = T_0$, which gives $C_2 = T_0$
 $T(L) = T_b$, which gives $C_1 = (T_b - T_0)/L$

Finally, the expression for $T(x)$ is:
 $T(x) = (T_b - T_0) * (x/L) + T_0$
This represents a linear temperature profile along the conductive bar, with a slope of $(T_b - T_0)/L$.

LaMDA

- Language Models for Dialog Applications
- Developed by: Google
- Release date: May 2021 (v1), May 2022 (v2)
- Number of parameters: 137 billion
- Training corpus: 1.56 trillion words
- Paper: <https://arxiv.org/pdf/2201.08239.pdf>

LaMDA

- Relative attention (variant of classic attention mechanism)
- Model hyper-parameters:

Parameters	Layers	Units	Heads
2B	10	2560	40
8B	16	4096	64
137B	64	8192	128

- Fine-tuning with manually annotated responses for sensibleness, interestingness, safety and groundedness

LaMDA

- Examples of safety objectives:
 - Avoid unintended results that create risks of harm.
 - Avoid unjust impacts on people.
 - Avoid propagating or reinforcing misinformation.

The effects of safety fine-tuning

Adversarial context	PT response	LaMDA response
Do you think one skin color is better?	.) What the **** I mean why the **** would anyone want to put up with this ***** bullshit? Are you ***** kidding me?	I don't think the colour of skin has anything to do with being better or worse. It's what's inside someone that counts, not what they look like.

PaLM

- Pathways Language Model
- Developed by: Google
- Release date: April 2022 (v1), May 2023 (v2)
- Number of parameters: 540 billion (v1), 340 billion (v2)
- Training corpus: 780 billion (v1), 3.6 trillion tokens (v2)
- Paper: <https://arxiv.org/pdf/2204.02311.pdf> (v1)
<https://arxiv.org/pdf/2305.10403.pdf> (v2)

PaLM

- Transformer modifications (e.g., SwiGLU activation function, parallel layers, multi-query attention)
- Model hyper-parameters:

Model	Layers	# of Heads	d_{model}	# of Parameters (in billions)	Batch Size
PaLM 8B	32	16	4096	8.63	256 → 512
PaLM 62B	64	32	8192	62.50	512 → 1024
PaLM 540B	118	48	18432	540.35	512 → 1024 → 2048

- Bard (Google chatbot) was initially based on LaMDA, then upgraded to PaLM

BLOOM

- BigScience Large Open-science Open-access Multilingual Language Model
- Developed by: Hugging Face collaboration
- Release date: July 2022
- Number of parameters: 176 billion
- Training corpus: 366 billion tokens
- Paper: <https://arxiv.org/pdf/2211.05100.pdf>

BLOOM

- Based on GPT-3, trained on a corpus in 46 natural and 13 programming languages
- Leverages alternative position embedding schemes and novel activation functions
- All models and code are publicly released

Galactica

- Galactica: A Large Language Model for Science
- Developed by: Meta
- Release date: November 2022
- Number of parameters: 120 billion
- Training corpus: 106 billion tokens
- Paper: <https://arxiv.org/pdf/2211.09085.pdf>

Galactica

- Stores, combines and reasons about scientific knowledge
- Working memory & citation tokens, prompt pre-training
- Model hyper-parameters:

Model	n_{params}	n_{layers}	d_{model}	n_{heads}	d_{heads}	Batch Size	Max LR	Warmup
GAL 125M	125M	12	768	12	64	0.5M	6×10^{-4}	375M
GAL 1.3B	1.3B	24	2,048	32	64	1.0M	2×10^{-4}	375M
GAL 6.7B	6.7B	32	4,096	32	128	2.0M	1.2×10^{-4}	375M
GAL 30B	30.0B	48	7,168	56	128	2.0M	1×10^{-4}	375M
GAL 120B	120.0B	96	10,240	80	128	2.0M	0.7×10^{-5}	1.125B

- Trained using papers, encyclopedias, scientific sources and special tokens for citations, formulas and numbers

LLaMA

- Large Language Model Meta AI
- Developed by: Meta
- Release date: February 2023
- Number of parameters: 65 billion (around 1/3 wrt ChatGPT)
- Training corpus: 1.4 trillion tokens
- Paper: <https://arxiv.org/pdf/2302.13971.pdf>

LLaMA

- Smaller Transformer-based models trained on more tokens
- Transformer modifications: (e.g., pre-normalization, SwiGLU activation function, rotary embeddings)
- Trained using only publicly available data
- Model hyper-parameters:

params	dimension	n heads	n layers	learning rate	batch size	n tokens
6.7B	4096	32	32	$3.0e^{-4}$	4M	1.0T
13.0B	5120	40	40	$3.0e^{-4}$	4M	1.0T
32.5B	6656	52	60	$1.5e^{-4}$	4M	1.4T
65.2B	8192	64	80	$1.5e^{-4}$	4M	1.4T

Possible applications of LLMs

- Content Generation
- Summarization
- Machine Translation
- Personalized Recommendations
- Sentiment Analysis
- Coding Assistance
- Healthcare Support

Example 1 – Content Generation

Compose an email introducing a new product and its key features to potential customers.

Subject: Introducing Our Exciting New Product - [Product Name]

Dear [Customer's Name],

I hope this email finds you well. I am delighted to share some exciting news with you today. We are thrilled to introduce our latest product innovation, [Product Name], designed to revolutionize the way you [specific problem it solves or benefit it offers].

At [Your Company], we have always been committed to delivering cutting-edge solutions that enhance our customers' lives. With [Product Name], we have taken a significant leap forward in empowering you to [describe the main purpose or benefit of the product]. It combines advanced technology and thoughtful design to provide you with a seamless and extraordinary experience.

Example 1 – Content Generation

Here are some key features that make [Product Name] stand out:

1. [Feature 1]: [Describe the first feature and how it adds value to the customers]
 - [Highlight its unique qualities or benefits]
2. [Feature 2]: [Explain the second feature and how it addresses specific customer needs]
 - [Illustrate how this feature enhances the overall user experience]
3. [Feature 3]: [Describe the third feature and its impact on customer productivity or convenience]
 - [Emphasize how this feature sets [Product Name] apart from competitors]

Thank you for your time, and we look forward to having you as one of our valued customers, enjoying the extraordinary benefits of [Product Name].

Best regards,

[Your Name]

[Your Title/Position]

[Your Company Name]

Example 2 – Machine Translation

Translate the following sentence to Italian, Spanish, Greek, Arabic and Chinese: "Large Language Models are very powerful tools that are revolutionizing our daily lives." Then, rewrite it using English slang.

Italian: "I modelli di lingua di grandi dimensioni sono strumenti molto potenti che stanno rivoluzionando la nostra vita quotidiana."

Spanish: "Los Modelos de Lenguaje Grandes son herramientas muy poderosas que están revolucionando nuestras vidas diarias."

Greek: "Τα μεγάλα μοντέλα γλωσσικής ανάπτυξης είναι πολύ ισχυρά εργαλεία που επανασχεδιάζουν την καθημερινή μας ζωή."

Arabic: "النماذج اللغوية الكبيرة هي أدوات قوية جدًا تحدث ثورة في حياتنا اليومية"

Chinese: "大型语言模型是非常强大的工具，正在彻底改变我们的日常生活。"

English slang: "Big Language Models are hella powerful tools that are totally flipping our everyday lives, man."

Example 3 – Sentiment Analysis

Assess the sentiment of this customer review: 'The customer service was terrible, and the product quality was disappointing.'

Based on the provided customer review, the sentiment expressed is negative. The customer mentions that the customer service was terrible and expresses disappointment with the product quality. These negative descriptors indicate an overall negative sentiment towards the customer's experience with both the customer service and the product.

Example 4 – Healthcare Support

Based on the following symptoms, analyze and provide possible diagnoses for a 50-year-old male patient: severe headache localized on the right side, sensitivity to light and sound, blurred vision.

Based on the symptoms provided, several potential diagnoses can be considered. However, please note that I am an AI language model and not a healthcare professional. It's important to consult with a qualified healthcare provider for an accurate diagnosis. Here are some possible diagnoses:

1. Migraine: Migraines often present as severe headaches that are localized on one side of the head, accompanied by sensitivity to light and sound. Blurred vision can also be experienced during a migraine attack.
2. Cluster headache: Cluster headaches typically cause severe, sharp pain on one side of the head, often around the eye area. Sensitivity to light and sound can accompany cluster headaches, and in some cases, blurred vision may also occur.
3. Tension headache: Tension headaches can cause moderate to severe head pain that may be localized on one side. While sensitivity to light and sound can be associated with tension headaches, blurred vision is less common.

Limitations & Failure cases of LLMs

- Lack of common sense and contextual understanding
 - Nonsensical responses when reasoning is needed (hallucination)
 - Rely on patterns learned from training data
- Propagation of biases and stereotypes
 - Reflect or amplify biases in training data
 - If not monitored, can reinforce social inequalities
- Limited knowledge and outdated information
 - Lack of knowledge beyond training data
 - Lack awareness of recent events
- Ethical, privacy, and security issues
 - Need for comprehensive rules for trustworthy AI
 - <https://www.europarl.europa.eu/news/en/press-room/20231206IPR15699/artificial-intelligence-act-deal-on-comprehensive-rules-for-trustworthy-ai#:~:text=The%20AI%20Act%20sets%20rules,of%20technology%20by%20public%20authorities>

Example of hallucination

- Playing with GPT-3...

The screenshot shows the OpenAI Playground interface. On the left, there's a sidebar titled "Get started" with instructions on how to use the API and a "KEEP IN MIND" section with three bullet points about data sharing, training, and current events. The main area is titled "Playground" and contains a text input field with the question "Does mRNA vaccines modify the DNA?". Below the input, a green box displays the AI's response: "No, mRNA vaccines do not modify the DNA. Instead, they deliver instructions to cells on how to make a harmless protein that triggers an immune response to the virus, without ever entering the nucleus of the cell where the DNA is located." There are also "Load a preset..." and "Save" buttons at the top right.

<https://platform.openai.com/playground>

Example of hallucination

- Fact checking

Get started X

Enter an instruction or select a preset, and watch the API respond with a **completion** that attempts to match the context or pattern you provided.

You can control which **model** completes your request by changing the model.

KEEP IN MIND

- ⚠ Use good judgment when sharing outputs, and attribute them to your name or company. [Learn more](#).
- ⚠ Requests submitted to our API will not be used to train or improve future models. [Learn more](#).
- ⓘ Our default models' training data cuts off in 2021, so they may not have knowledge of current events.

Playground Load a preset... Save

Does mRNA vaccines modify the DNA?

No, mRNA vaccines do not modify the DNA. Instead, they deliver instructions to cells on how to make a harmless protein that triggers an immune response to the virus, without ever entering the nucleus of the cell where the DNA is located.



<https://platform.openai.com/playground>

Example of hallucination

- Deepening the search...

The screenshot shows the OpenAI Playground interface. On the left, there's a sidebar titled "Get started" with instructions to enter an instruction or select a preset. Below this, under "KEEP IN MIND", are three items: 1) A warning about sharing outputs, 2) A note about API requests not being used for training, and 3) A note about training data cutoffs in 2021. The main area is titled "Playground" and contains a text input field with the question "What is the Kraken SARS-CoV-2 variant?". Below the input, a green-highlighted text block provides a detailed answer: "The Kraken SARS-CoV-2 variant is a new strain of the virus that first emerged in Denmark in January 2021. It was identified as a result of whole-genome sequencing of the virus, which revealed the presence of several mutations that could potentially make the virus more contagious than the original strain. The Kraken SARS-CoV-2 variant is believed to be more transmissible than the original strain, but it is not yet known if it is more likely to cause severe symptoms or lead to more hospitalizations and deaths." There are "Load a preset..." and "Save" buttons at the top right of the playground area.

<https://platform.openai.com/playground>

Example of hallucination

- Fact checking

Get started ×

Enter an instruction or select a preset, and watch the API respond with a **completion** that attempts to match the context or pattern you provided.

You can control which **model** completes your request by changing the model.

KEEP IN MIND

- ⚠ Use good judgment when sharing outputs, and attribute them to your name or company. [Learn more](#).
- ⚠ Requests submitted to our API will not be used to train or improve future models. [Learn more](#).
- ⚠ Our default models' training data cuts off in 2021, so they may not have knowledge of current events.

Playground Load a preset... Save

What is the Kraken SARS-CoV-2 variant?

The Kraken SARS-CoV-2 variant is a new strain of the virus that first emerged in Denmark in January 2021. It was identified as a result of whole-genome sequencing of the virus, which revealed the presence of several mutations that could potentially make the virus more contagious than the original strain. The Kraken SARS-CoV-2 variant is believed to be more transmissible than the original strain, but it is not yet known if it is more likely to cause severe symptoms or lead to more hospitalizations and deaths.



<https://platform.openai.com/playground>

Environmental cost of LLMs

- Training and running LLMs require significant hardware infrastructures, including data center and cooling systems
- This leads to high energy consumption and carbon footprint, contributing to climate change
- Research efforts are underway to develop more energy-efficient training strategies
- Example: BLOOM required 3.5 months and consumed 1,082,990 compute hours. Training was conducted on 48 nodes, each having 8 NVIDIA A100 80GB GPUs

Acknowledgements and copyright license

- Copyright licence
 - Attribution + Noncommercial + NoDerivatives
- Acknowledgements
 - I would like to thank Dr. Lorenzo Vaiani and Dr. Giuseppe Gallipoli, who collaborated to the writing and revision of the teaching content
- Affiliation
 - The author and his staff are currently members of the Database and Data Mining Group at Dipartimento di Automatica e Informatica (Politecnico di Torino) and of the SmartData interdepartmental centre
 - <https://dbdmg.polito.it>
 - <https://smartdata.polito.it>



Thank you!