

LARGE LANGUAGE MODELS

Continual Learning

WHO AM I?



➤ Bachelor & Master Computer Science (2016-2022)



➤ PhD at the Artificial Intelligence & Machine Learning Lab (since 2022)



➤ Research Scientist at the German Research Center for AI (DFKI) (since 2023)



➤ Visiting Research Scientist at Adobe (2023 & 2024)



➤ Co-founder of OcciGlot Research Initiative for European Language Models (since 2024)



AGENDA

1 Introduction

- Context
- Natural Language Processing
- Language Modelling
- Components of LLMs

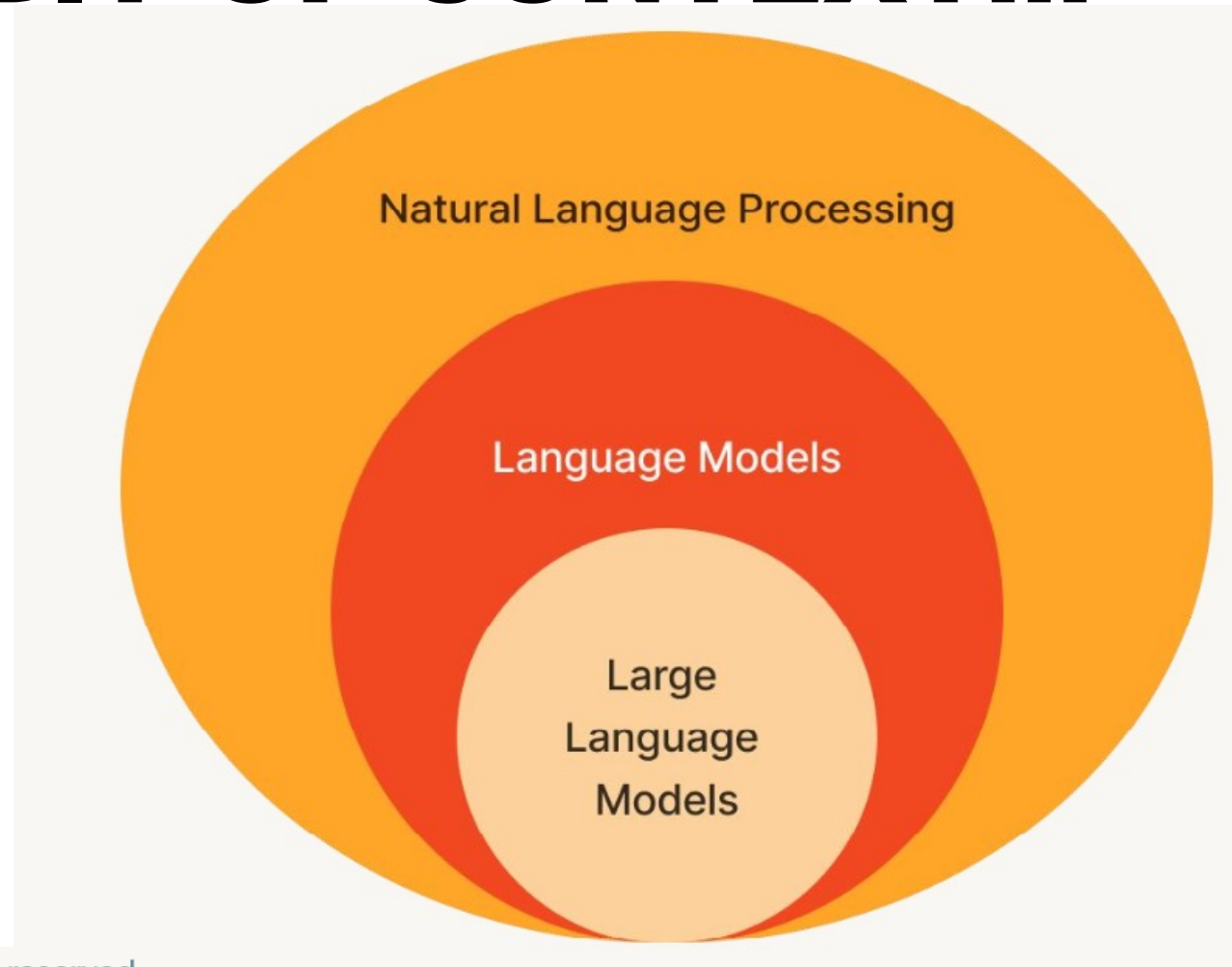
2 Building LLMs

- Stages of Training
- Pre-Training
- Instruction Tuning
- Preference Tuning

3 Applications & Challenges

- Domain Adaptation
- Unique Challenges

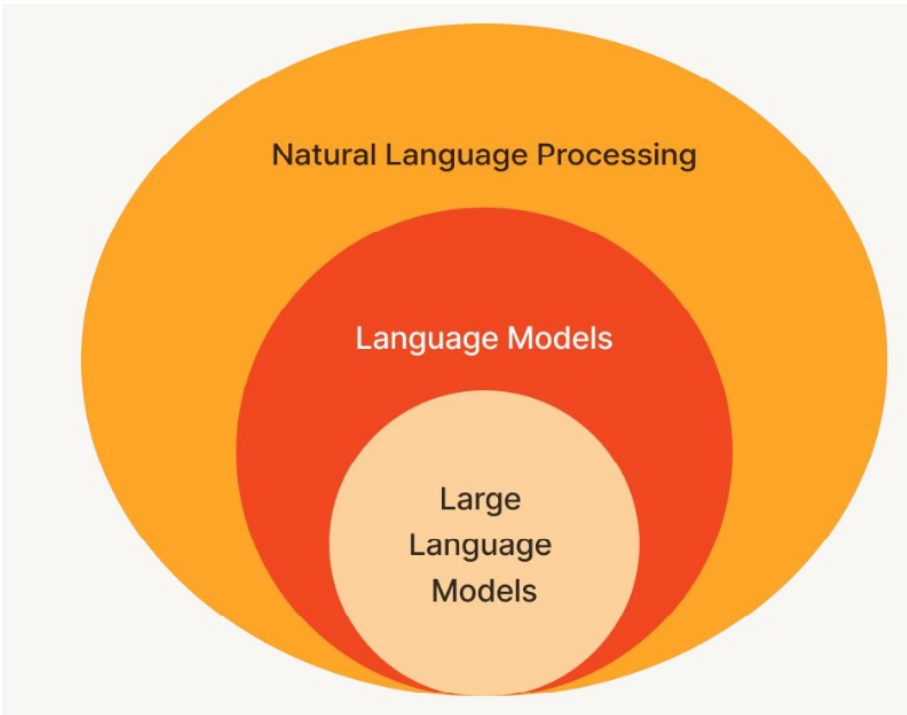
LITTLE BIT OF CONTEXT...



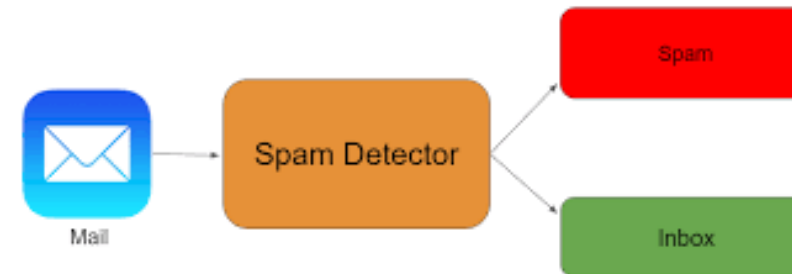
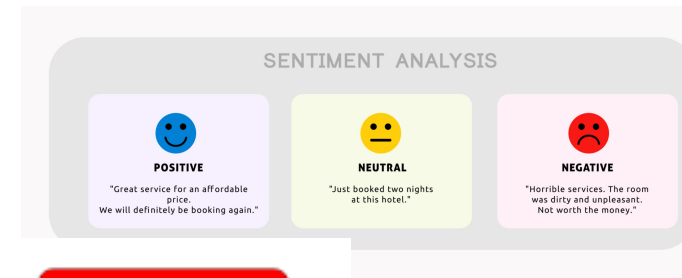
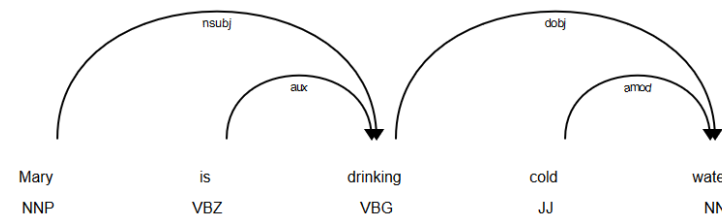
NATURAL LANGUAGE PROCESSING (NLP)

- Over 50 years old field
- Originated from linguistics

“giving computers the ability to support and manipulate human language”



©2023 Databricks Inc. — All rights reserved



Figures from:
<https://bhatnagar91.medium.com/analyzing-youtube-fans-feelings-uncovering-the-power-of-sentiment-analysis-5d840909ac58>
<https://towardsdatascience.com/spam-detection-in-emails-de0398ea3b48>
<https://suttipong-kull.medium.com/how-to-extract-subject-verb-and-object-by-nlp-4149323a7d7d>

LANGUAGE MODELING

- Probabilistic of sequences of “words”

“dsfh hjaiorpghh fdhjol adhjj auezoiyh”



Low probability

“language modelling is fun”



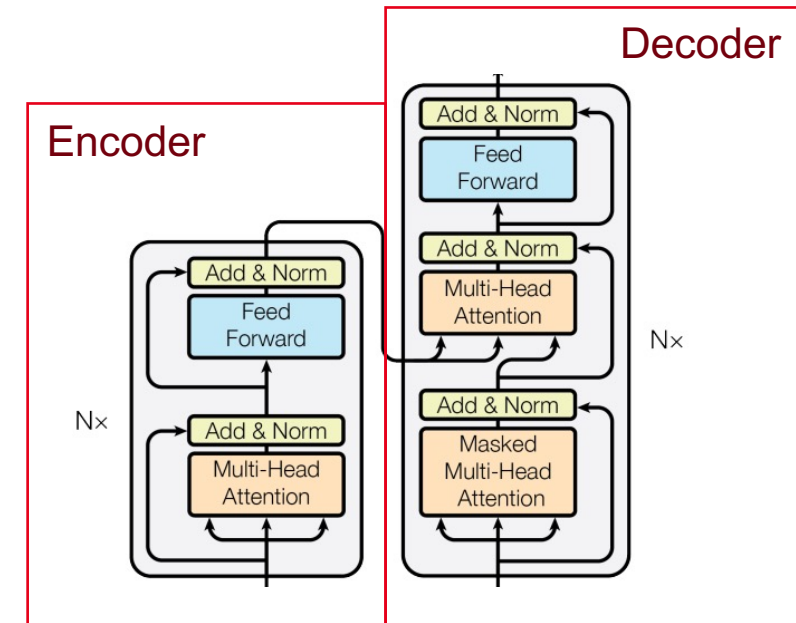
High probability

- **Assumption:** *The probability of the next word in a sequence only depends on the previous ones*

$$\mathcal{P}(w_n | w_{n-1} \dots w_1) = \prod \mathcal{P}(w_i | w_{i-1})$$

LARGE LANGUAGE MODELS

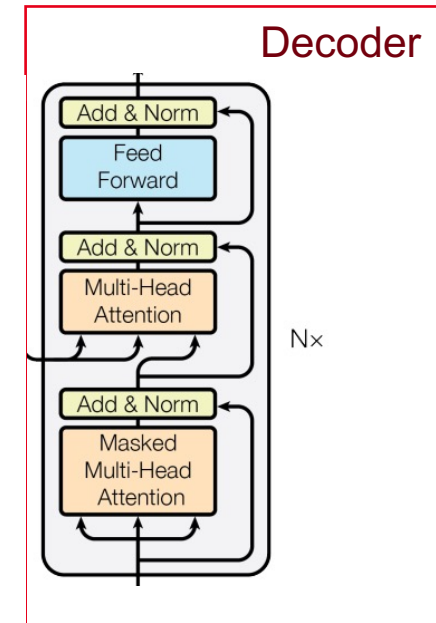
- (Large) neural networks that model text probabilistics
 - BERT (2018) up to **340M** parameters & **2.5B** words
 - GPT-4 (2023) probably 16x111B MoE = **1.7T** parameters
 - Llama-3 (2024) **15T** words



Vaswani et al. "Attention is all you need." *NeurIPS* (2017).

How do we need to setup our neural network to perform language modelling on text?

$$\mathcal{P}(w_n | w_{n-1} \dots w_1) = \prod \mathcal{P}(w_i | w_{i-1})$$



Vaswani et al. "Attention is all you need." *NeurIPS* (2017).

LLMS IN A NUTSHELL

➤ Build a large vocabulary of (sub-)words (30k-250k)

➤ Tokenize input text into sequence of vocabulary IDs

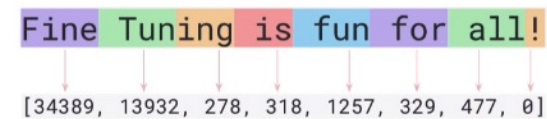
"language modelling is fun" → ['<s>', '_language', '_mod', 'elling', '_is', '_fun'] → [1, 3842, 968, 3572, 349, 746]

➤ Each token is embedded to a learned representation

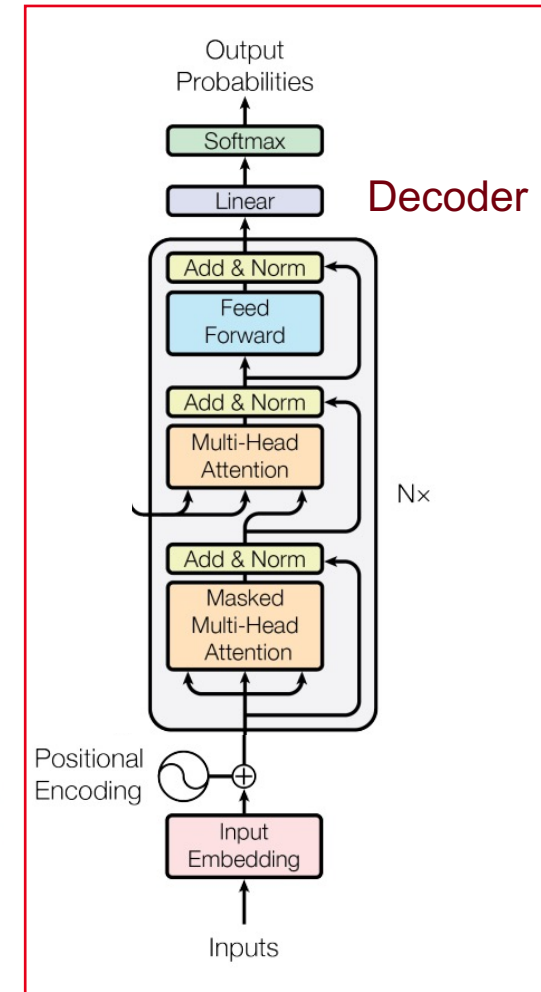
➤ Make forward pass through the model

➤ Sample next word from LM layer → classification over entire vocabulary

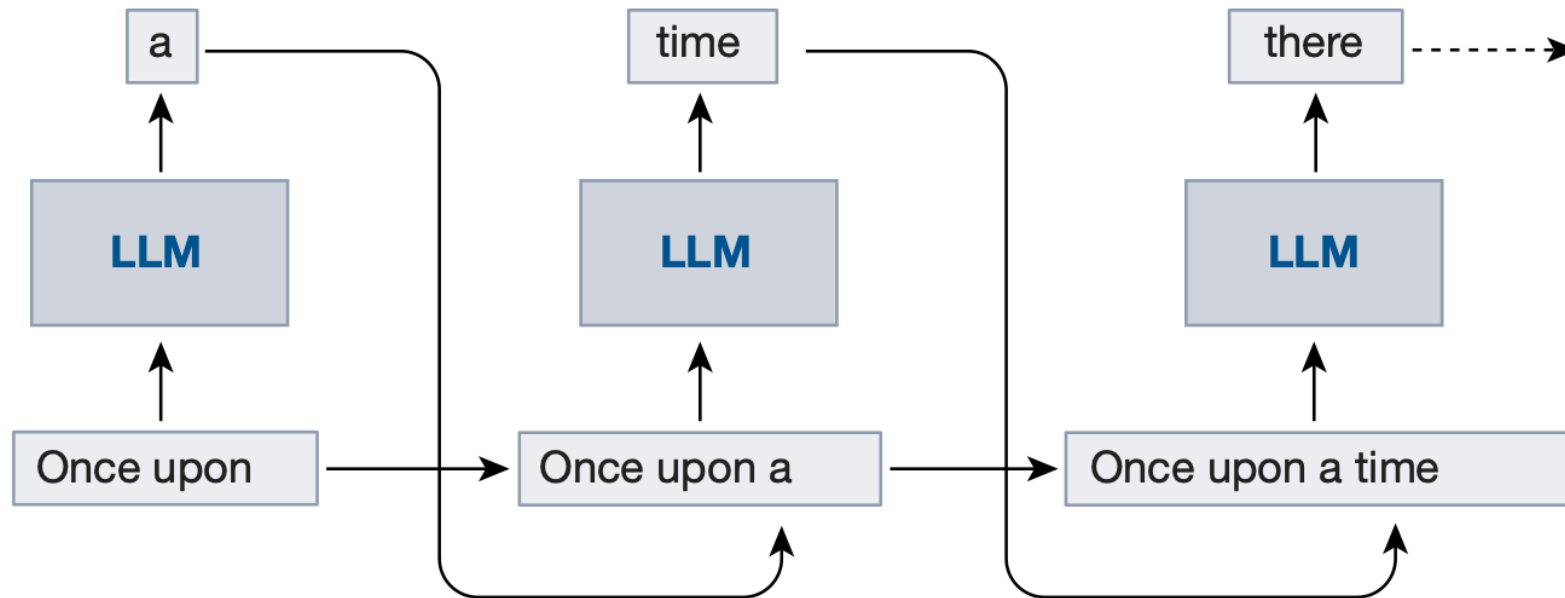
What do we need to also generate new text?



<https://teetracker.medium.com/llm-fine-tuning-step-tokenizing-caebb280cfc2>



AUTOREGRESSIVE SAMPLING



Shanahan, Murray, Kyle McDonell, and Laria Reynolds. "Role play with large language models." *Nature* (2023)

TRAINING LLMS

➤ LLMs for chat applications are usually trained in 3 stages

1 Pre-Training

- Build capable base-model
- General purpose
- Application-specific models are built on top

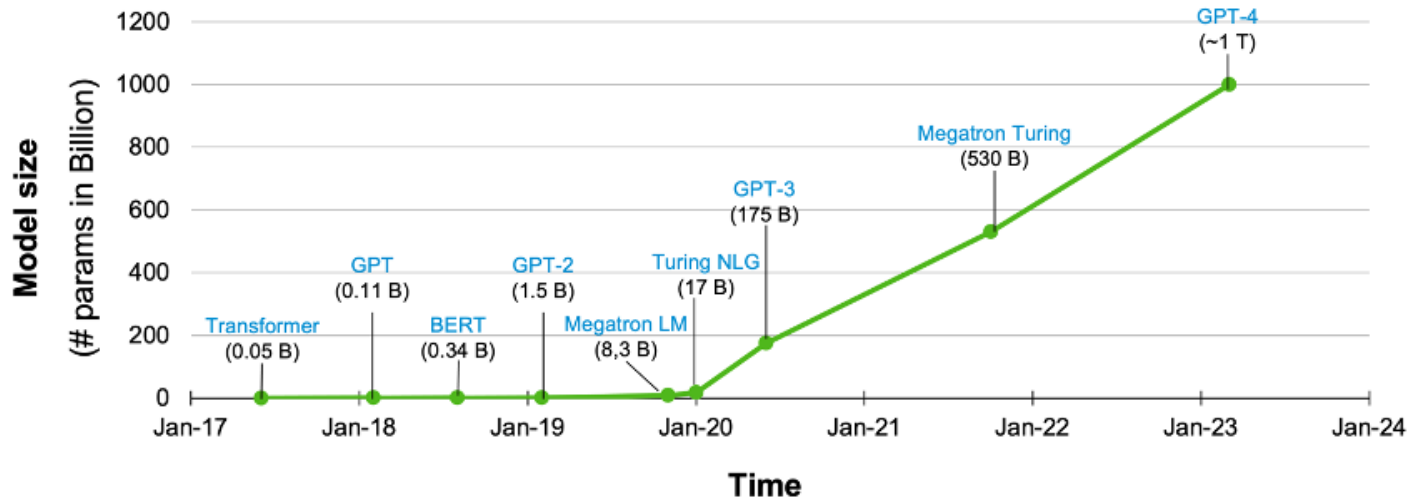
2 Instruction Tuning

- Train capability to follow instructions
- For example for a chat model

3 Preference Tuning

- *Align* model to some preferred „*behavior*“

PRE-TRAINING

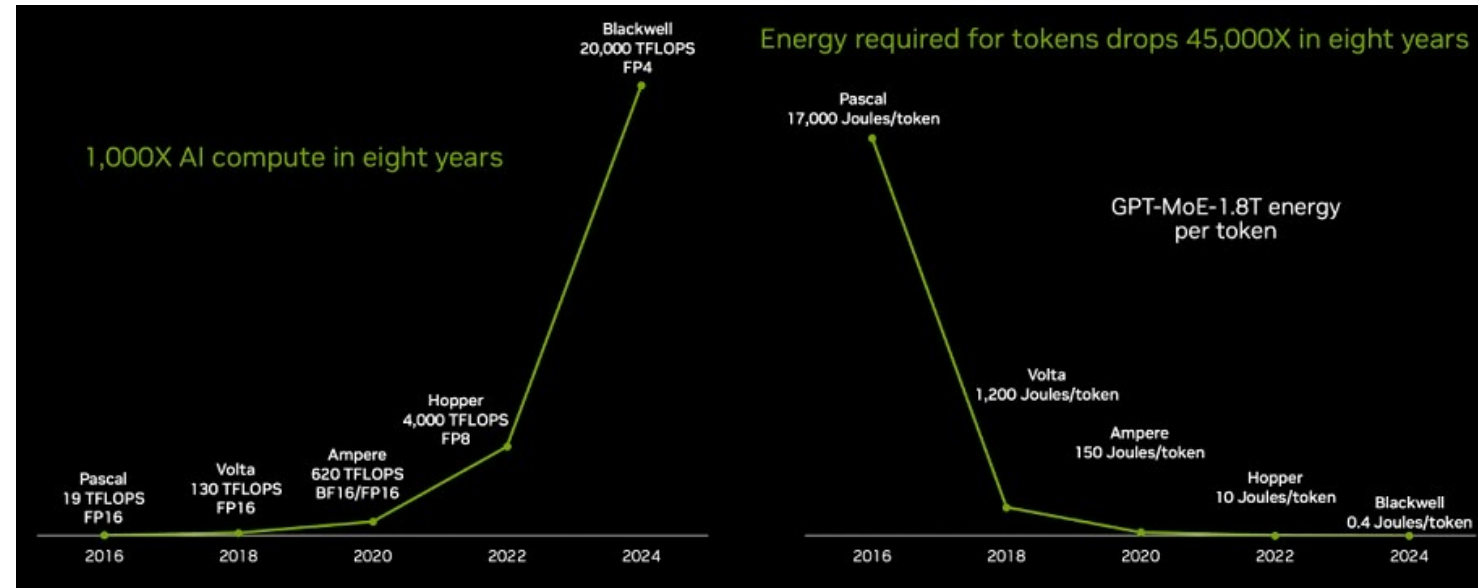


- LLM sizes have grown immensely
- Significant jump in last 3-4 years
- Growing computational requirements

<https://medium.com/@gladabhi/optimize-cost-to-host-llm-with-sagemaker-async-endpoints-1a6755e458c5>

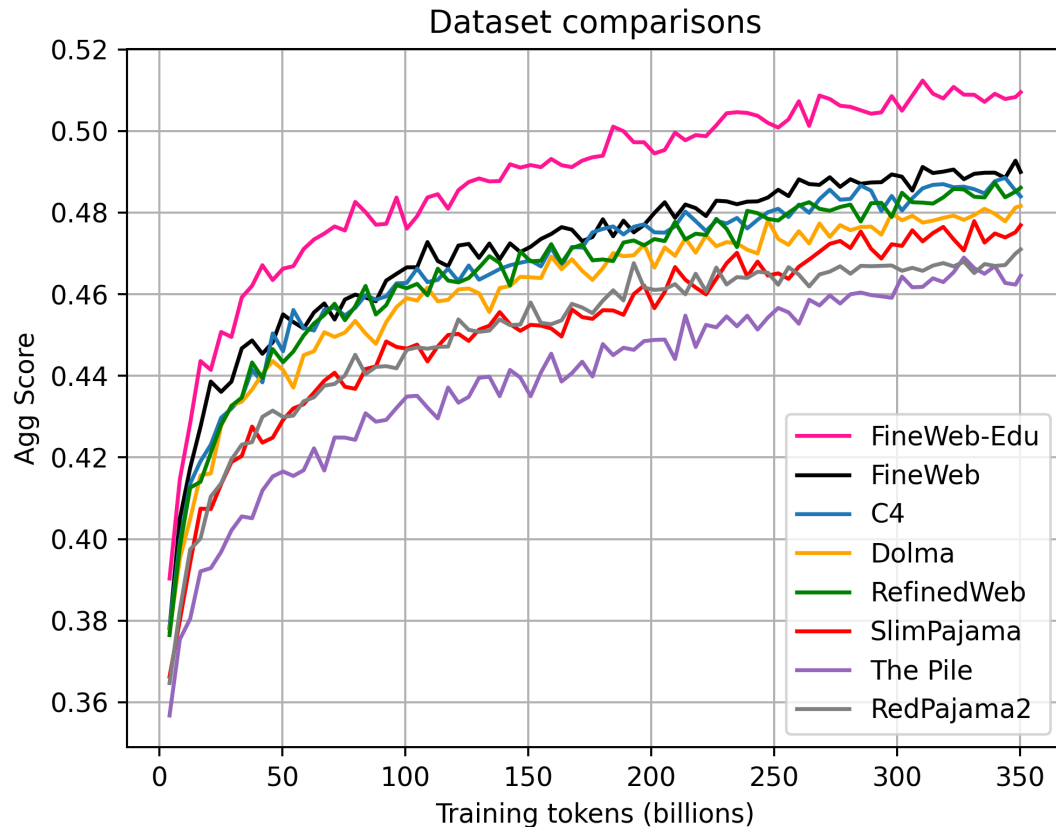
PRE-TRAINING

- Enabled through rapid hardware improvements
- Hardware performance/throughput
- Similar results at lower precision
- Energy optimization



NVIDIA Computex 2024 Keynote

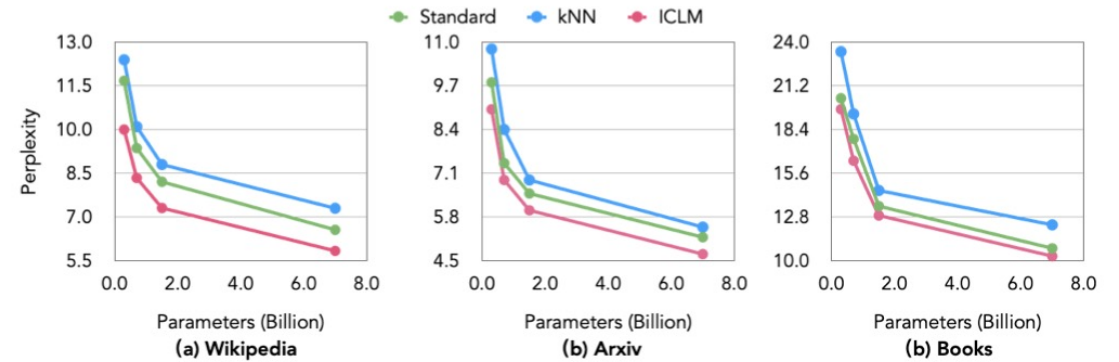
DATA IS EVERYTHING TODO: SPLIT UP



- Limited architectural changes in recent years
- Quality of data far more important
- Large amounts of data on the web
- Filtering & Curation is key

DATA IS EVERYTHING TODO: SPLIT UP

- Design choices go beyond what data
- How should data be presented during training?
- Packing, Context Grouping, “*Curriculum*”
can have significant influences



Shi, Weijia, et al. "In-Context Pretraining: Language Modeling Beyond Document Boundaries." *arXiv:2310.10638* (2023).

Task	Naive Packing	Fewer Truncations Packing	Percentage Increase
truthfulqa_mc	0.452648	0.467687	3.32%
arc_challenge	0.517918	0.528157	1.98%
truthful_qa_de	0.485529	0.492979	1.53%
arc_challenge_de	0.480375	0.493174	2.66%
hellaswag	0.776041	0.773352	-0.35%
hellaswag_de	0.655248	0.653356	-0.29%
MMLU	0.573719	0.579802	1.06%
MMLU-DE	0.504509	0.503863	-0.13%

<https://occlot.eu/posts/llama-3-german-8b/>



In contrast to OLMo 1.0, we trained OLMo 1.7 with a two-stage curriculum:

- In the first stage, we train the model from scratch on the Dolma 1.7 dataset. We set a cosine learning rate schedule with a warmup of 2500 steps, a peak learning rate of $3e-4$, and a cosine decay to $3e-5$ after 3T tokens. We cut off this stage after 2T tokens, when the learning rate is still high.
- At this point we switch to the second stage, in which we train on a curated subset of Dolma 1.7 for another 50B tokens, while linearly decaying the learning rate to 0. We curate this high-quality subset by (1) using all available Wikipedia, OpenWebMath and Flan data, (2) removing Dolma CC, CC News, and Megawika, and (3) rebalancing remaining sources to achieve approximately equal proportions of each. See exact token counts and relative proportions of this second stage mix below.

*How does this definition of “curriculum learning”
align with last weeks lecture?*

Olmo 1.7 blogpost

<https://blog.allenai.org/olmo-1-7-7b-a-24-point-improvement-on-mmlu-92b43f7d269d>

CURRICULA IN LLM PRE-TRAINING

- Very informal/flexible definition
- Any change to data-mixture or training setup
- Usually hard → easy or
noisy data → clean/high-quality data



In contrast to OLMo 1.0, we trained OLMo 1.7 with a two-stage curriculum:

- In the first stage, we train the model from scratch on the Dolma 1.7 dataset. We set a cosine learning rate schedule with a warmup of 2500 steps, a peak learning rate of $3e-4$, and a cosine decay to $3e-5$ after 3T tokens. We cut off this stage after 2T tokens, when the learning rate is still high.
- At this point we switch to the second stage, in which we train on a curated subset of Dolma 1.7 for another 50B tokens, while linearly decaying the learning rate to 0. We curate this high-quality subset by (1) using all available Wikipedia, OpenWebMath and Flan data, (2) removing Dolma CC, CC News, and Megawika, and (3) rebalancing remaining sources to achieve approximately equal proportions of each. See exact token counts and relative proportions of this second stage mix below.

Olmo 1.7 blogpost

<https://blog.allenai.org/olmo-1-7-7b-a-24-point-improvement-on-mmlu-92b43f7d269d>



In contrast to OLMo 1.0, we trained OLMo 1.7 with a two-stage curriculum:

- In the first stage, we train the model from scratch on the Dolma 1.7 dataset. We set a cosine learning rate schedule with a warmup of 2500 steps, a peak learning rate of $3e-4$, and a cosine decay to $3e-5$ after 3T tokens. We cut off this stage after 2T tokens, when the learning rate is still high.
- At this point we switch to the second stage, in which we train on a curated subset of Dolma 1.7 for another 50B tokens, while linearly decaying the learning rate to 0. We curate this high-quality subset by (1) using all available Wikipedia, OpenWebMath and Flan data, (2) removing Dolma CC, CC News, and Megawika, and (3) rebalancing remaining sources to achieve approximately equal proportions of each. See exact token counts and relative proportions of this second stage mix below.

What is the scoring and pacing function?

CURRICULA IN LLM PRE-TRAINING

- Very informal/flexible definition
- Any change to data-mixture or training setup
- Usually hard → easy or
noisy data → clean/high-quality data
- Setup often chosen based on best-practices
(manual decision of pacing)
- Scoring based on some quality heuristic



In contrast to OLMo 1.0, we trained OLMo 1.7 with a two-stage curriculum:

- In the first stage, we train the model from scratch on the Dolma 1.7 dataset. We set a cosine learning rate schedule with a warmup of 2500 steps, a peak learning rate of $3e-4$, and a cosine decay to $3e-5$ after 3T tokens. We cut off this stage after 2T tokens, when the learning rate is still high.
- At this point we switch to the second stage, in which we train on a curated subset of Dolma 1.7 for another 50B tokens, while linearly decaying the learning rate to 0. We curate this high-quality subset by (1) using all available Wikipedia, OpenWebMath and Flan data, (2) removing Dolma CC, CC News, and Megawika, and (3) rebalancing remaining sources to achieve approximately equal proportions of each. See exact token counts and relative proportions of this second stage mix below.

Olmo 1.7 blogpost

<https://blog.allenai.org/olmo-1-7-7b-a-24-point-improvement-on-mmlu-92b43f7d269d>

INSTRUCTION TUNING

- Goal: Capabilities beyond completing sentences

Input: *"Teach me how to bake bread"* → Output: *"in an oven."*

- Additional training on structured conversations
- General instruction following
- Elicit in-context learning (few-shot prompting)

Chatbot

Teach me how to make bread

Baking bread can be a rewarding and enjoyable experience, especially when using fresh ingredients. To bake your own bread, you'll need to know a few basic steps and have access to a few kitchen tools. Here's how you can make bread:

1. Gather your ingredients: You'll need 4 cups of all-purpose flour, 2 teaspoons of salt, 2 teaspoons of yeast, and 2 tablespoons of sugar, if desired. Additionally, you'll need 1 1/2 cups of warm water, and a half cup of vegetable oil, and a tablespoon of butter. For extra flavor, you can also add herbs or spices to your bread mixture.
2. Prepare your dough: In a large mixing bowl, combine 3 1/2 cups of flour, the yeast, salt, sugar (if using), and the half cup of vegetable oil. Then, add 1 cup of warm water and mix until a dough forms. If the dough is too sticky, gradually add more flour. If it's too dry, add a splash of water.
3. Knead the dough: Turn the dough out onto a well-floured surface and knead it for about 10 minutes. The dough should be smooth and elastic. If it's still sticky, add more flour. If it feels dry, add a splash of water.
4. Allow the dough to rise: Clean the mixing bowl, and lightly grease it with the tablespoon of butter. Return the kneaded dough to the bowl and cover it with a damp kitchen towel. Let it rise in a warm place for about an hour or until it has doubled in size.
5. Shape your loaves: Once the dough has risen, punch it down and turn it out onto a floured surface. Divide the dough into two equal parts and shape each into a loaf shape. You can use a loaf pan or shape it by hand into a round or oval shape.
6. Bake the loaves: Place the loaves on a greased or parchment-lined baking sheet. Cover them with a kitchen towel and let them rise for an additional 30 minutes.
7. Preheat your oven: Preheat the oven to 375°F (190°C). Bake the loaves for 20-25 minutes, or until the loaves are golden brown and sound hollow when tapped.
8. Cool your loaves: Allow the loaves to cool completely before slicing and serving.

Remember, baking bread takes time and patience. Don't be discouraged if your first attempt doesn't turn out perfectly. With practice, you'll get the hang of it and can experiment with different flavors and shapes.

Retry Undo Clear

Type a message... Submit

INSTRUCTION TUNING

`<|begin_of_text|><|start_header_id|>system<|end_header_id|>`

You are a helpful, smart, kind, and efficient AI assistant. You always fulfill the user's requests to the best of your ability.<|eot_id|>

`<|start_header_id|>human<|end_header_id|>`

Teach me how to bake bread<|eot_id|>

`<|start_header_id|>assistant<|end_header_id|>`

Baking bread can be a rewarding and enjoyable experience, especially when using fresh ingredients. To bake your own bread, you'll need to know a few basic steps and have access to a few kitchen tools.

Here's how you can make bread:

1. Gather your ingredients: [...]

2. Prepare your dough: [...]

3. Knead the dough: [...]

[...]

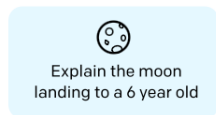
With practice, you'll get the hang of it and can experiment with different flavors and shapes<|eot_id|>

PREFERENCE TUNING

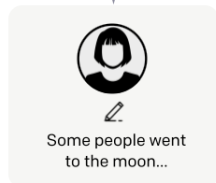
Step 1

Collect demonstration data, and train a supervised policy.

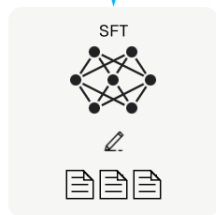
A prompt is sampled from our prompt dataset.



A labeler demonstrates the desired output behavior.



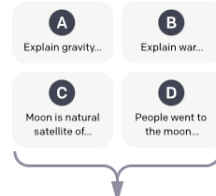
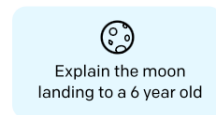
This data is used to fine-tune GPT-3 with supervised learning.



Step 2

Collect comparison data, and train a reward model.

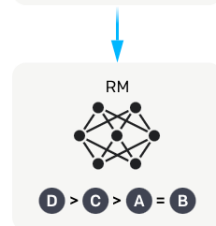
A prompt and several model outputs are sampled.



A labeler ranks the outputs from best to worst.



This data is used to train our reward model.



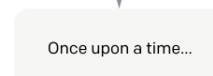
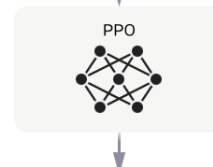
Step 3

Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.



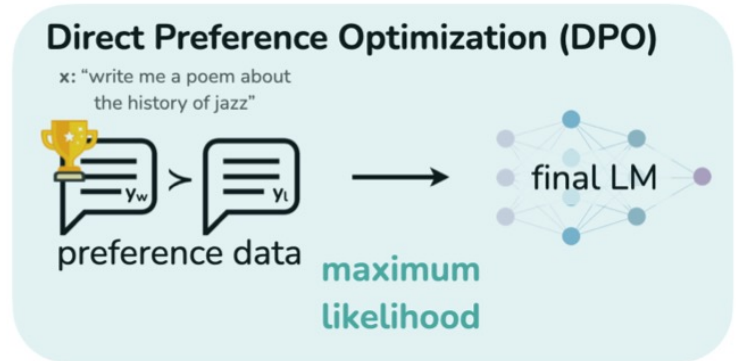
The policy generates an output.



The reward model calculates a reward for the output.



The reward is used to update the policy using PPO.



Rafailov, Rafael, et al. "Direct preference optimization: Your language model is secretly a reward model." *NeurIPS* (2023)

PREFERENCE TUNING

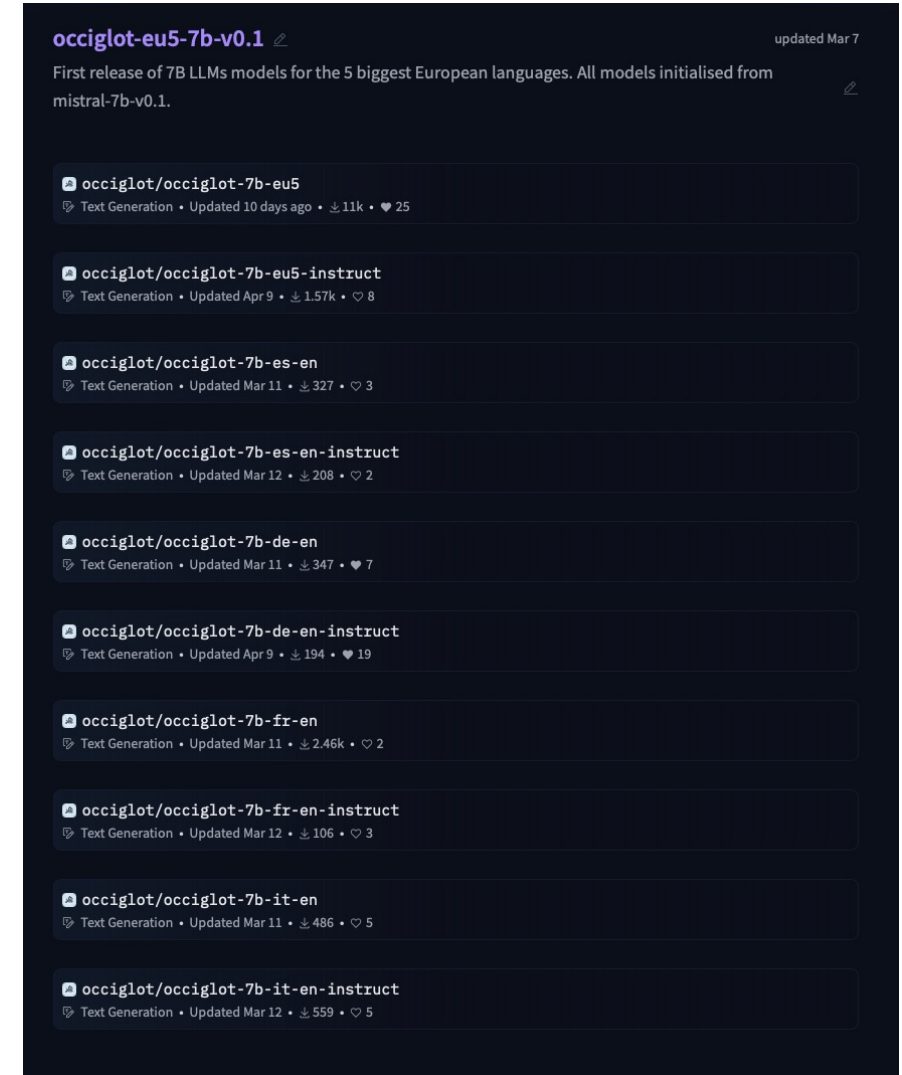
How does preference tuning relate to domain adaptation & transfer learning?

PREFERENCE TUNING

- Goal: Instill (human) preference on outputs for same input
- No changes to the underlying task
- Still adaptation of the target domain
- We change the distribution of model outputs/predictions

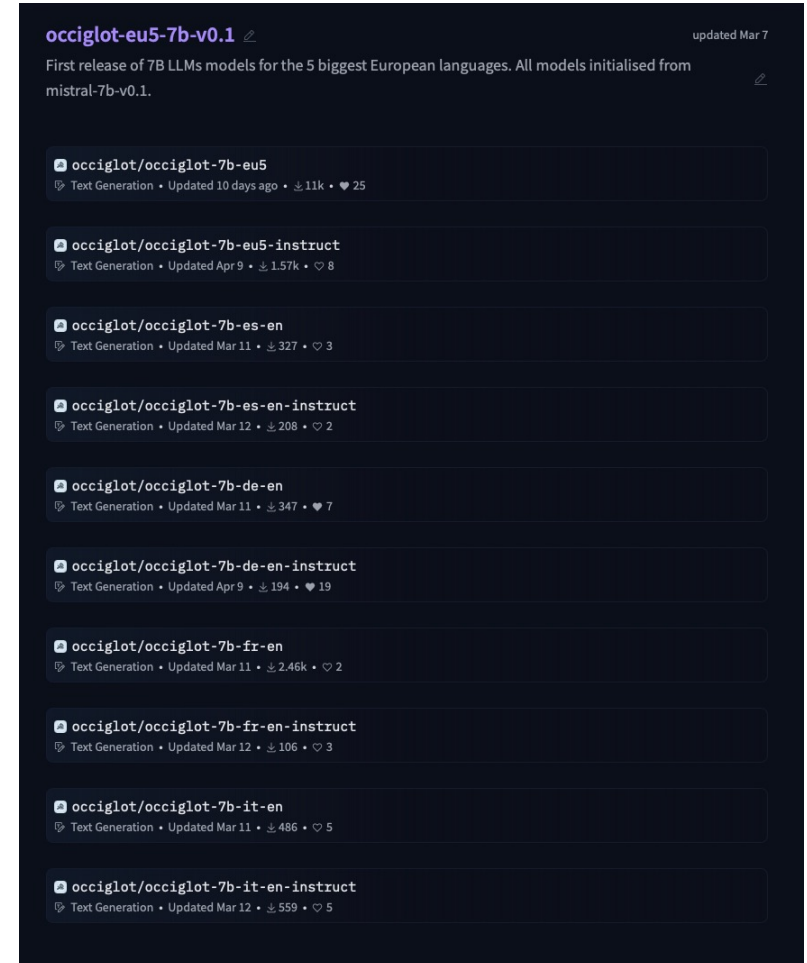
DOMAIN ADAPTATION

- Again: Terminology is blurry between different fields
- LLM domain adaption focuses on use cases (domain)
- E.g. Adapt LLM for a different language



CONTINUAL PRE-TRAINING

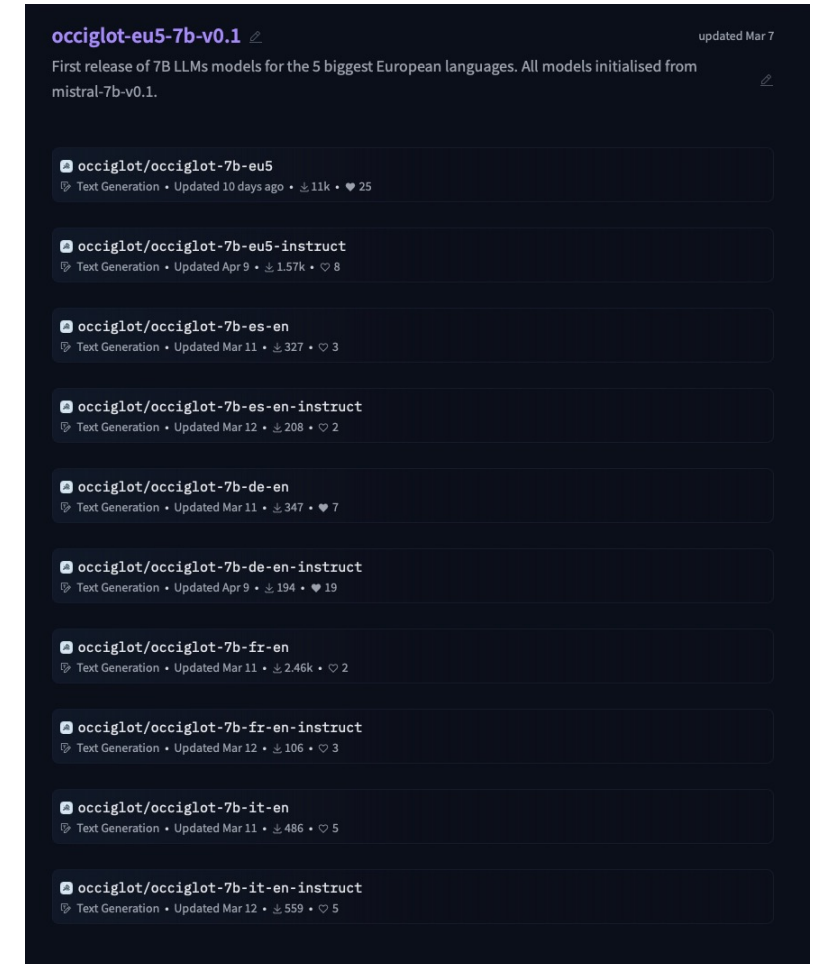
- Take pre-trained Language Model
- Continue pre-training stage on new data/other language



Why would we want to use a pre-trained model instead of training from scratch?

CONTINUAL PRE-TRAINING

- Take pre-trained Language Model
- Continue pre-training stage on new data/other language
- LLM training is costly & hard
- Language Modelling can transfer between languages
- Maybe we want to retain performance on original language
- Often not enough mono-lingual data for training from scratch



CHALLENGES W/ CLOSED MODELS

- The best models come from large companies
- Very sparse/limited information on data & training

What challenges does that entail for continual pre-training



<https://www.analyticsvidhya.com/blog/2024/04/meta-llama-a-breakthrough-in-open-ai-models/>



<https://de.wikipedia.org/>

CHALLENGES W/ CLOSED MODELS

- The best models come from large companies
- Very sparse/limited information on data & training
- Memory buffers for replay are hard to construct
- What data may have been seen already?
- Careful rewarming/choice of learning rate



<https://www.analyticsvidhya.com/blog/2024/04/meta-llama-a-breakthrough-in-open-ai-models/>



<https://de.wikipedia.org/>

CHALLENGES

- Large-scale model training is very different than previous examples in this lecture

*What are some distinct challenges/differences
arising for LLM training?*

CHALLENGES

- Large-scale model training is very different than previous examples in this lecture
- A lot of machine learning engineering required
- Usually only trained for one epoch
- You may only be able to effort one training run at full scale
 - No extensive hyper-parameter ablations
 - Instead try to establish scaling laws from smaller models
- Scale of the data may lead to unintended inclusion of problematic material

EVALUATION

- How to evaluate a (pre-trained) LLM
- Naïve idea: Based on the Language Modeling objective

$$\mathcal{P}(w_n | w_{n-1} \dots w_1) = \prod \mathcal{P}(w_i | w_{i-1})$$

- Model the (un-)certainty of the LLM for a given sequence (entropy) → inverse

$$\frac{1}{\prod \mathcal{P}(w_i | w_{i-1})}$$

- Normalize by word length (geometric average) & write in log-scale for numeric stability

$$\exp\left(\frac{1}{n} \sum \log(\mathcal{P}(w_i | w_{i-1}))\right)$$

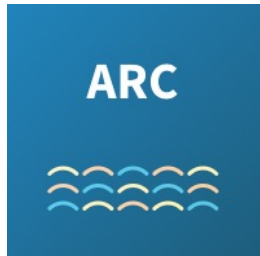
EVALUATION

➤ Problems with Perplexity

- How to you chose representative test set?
- Usefulness of comparison between different approaches might be limited
- Does not capture desired probabilities of chat bots very well

EVALUATION

AI2 Reasoning Challenge (ARC)



Which property of a mineral can be determined just by looking at it?

- (A) luster
- (B) mass
- (C) weight
- (D) hardness

Clark, Peter, et al. "Think you have solved question answering? try arc, the ai2 reasoning challenge." *arXiv:1803.05457* (2018)

Massive Multitask Language Understanding (MMLU)

Microeconomics

- One of the reasons that the government discourages and regulates monopolies is that
- (A) producer surplus is lost and consumer surplus is gained. ✗
 - (B) monopoly prices ensure productive efficiency but cost society allocative efficiency. ✗
 - (C) monopoly firms do not engage in significant research and development. ✗
 - (D) consumer surplus is lost with higher prices and lower levels of output. ✓

Conceptual Physics

- When you drop a ball from rest it accelerates downward at 9.8 m/s². If you instead throw it downward assuming no air resistance its acceleration immediately after leaving your hand is
- (A) 9.8 m/s² ✓
 - (B) more than 9.8 m/s² ✗
 - (C) less than 9.8 m/s² ✗
 - (D) Cannot say unless the speed of throw is given. ✗

06.06.24 Hendrycks, Dan, et al. "Measuring massive multitask language understanding." ICLR (2021).

HellaSwag

Category: Shaving (ActivityNet; In-domain)

A bearded man is seen speaking to the camera and making several faces. the man

- a) then switches off and shows himself via the washer and dryer rolling down a towel and scrubbing the floor. (0.0%)
- b) then rubs and wipes down an individual's face and leads into another man playing another person's flute. (0.0%)
- c) is then seen eating food on a ladder while still speaking. (0.0%)
- d) then holds up a razor and begins shaving his face. (100.0%)**

Zellers, Rowan, et al. "Hellaswag: Can a machine really finish your sentence?." ACL (2019)

GMS8K

Problem: Beth bakes 4, 2 dozen batches of cookies in a week. If these cookies are shared amongst 16 people equally, how many cookies does each person consume?

Solution: Beth bakes 4 2 dozen batches of cookies for a total of $4 \cdot 2 = \ll 4 \cdot 2 = 8 \gg$ 8 dozen cookies

There are 12 cookies in a dozen and she makes 8 dozen cookies for a total of $12 \cdot 8 = \ll 12 \cdot 8 = 96 \gg$ 96 cookies

She splits the 96 cookies equally amongst 16 people so they each eat $96 / 16 = \ll 96 / 16 = 6 \gg$ 6 cookies

Final Answer: 6

Cobbe, Karl, et al. "Training verifiers to solve math word problems." *arXiv:2110.14168* (2021).

EVALUATION

1111

A

Ma

Microeconomics

Conceptual Physics

Open LLM Leaderboard

LLM Benchmark | Metrics through time | About | FAQ | Submit

Search: Separate multiple queries with |

Select Columns to Display: Average, ARC, HellaSwag, MMLU, TruthfulQA, Winogrande, GSM8K, Type, Architecture, Precision, Merged, Hub License, #Params (B), Hub, Model sha

Model types: base merges and moerges, fine-tuned on domain-specific datasets, chat models (RLHF, DPO, IFT, ...), continuously pretrained, pretrained

Precision: bfloat16, float16, 4bit, 8bit, GPTQ

Select the number of parameters (B): 7, 10

Hide models: Private or deleted, Contains a merge/moerge, MoE, Flagged

T	Model	Average	ARC	HellaSwag	MMLU	TruthfulQA	Winogrande	GSM8K
📌	daavidkim205/Rhea-72b-v0.5	81.22	79.78	91.15	77.95	74.5	87.85	76.12
🗨️	MTSAIR/MultiVerse_70B	81	78.67	89.77	78.22	75.18	87.53	76.65
📌	MTSAIR/MultiVerse_70B	80.98	78.58	89.74	78.27	75.09	87.37	76.8
📌	abacusai/Smaug-72B-v0.1	80.48	76.02	89.27	77.15	76.67	85.08	78.7
📌	ibivibiv/alpaca-dragon-72b-v1	79.3	73.89	88.16	77.4	72.69	86.03	77.63
🗨️	mistralai/Mixtral-8x22B-Instruct-v0.1	79.15	72.7	89.08	77.77	68.14	85.16	82.03
🗨️	MazyarPanahi/Llama-3-70B-Instruct-DPO-v0.2	78.96	72.53	86.22	80.41	63.57	82.79	88.25
🗨️	MazyarPanahi/Llama-3-70B-Instruct-DPO-v0.4	78.89	72.61	86.03	80.5	63.26	83.58	87.34
🗨️	MazyarPanahi/Llama-3-70B-Instruct-DPO-v0.3	78.74	72.35	86	80.47	63.45	82.95	87.19
🗨️	mmnga/Llama-3-70B-japanese-suzume-vector-v0.1	78.6	72.35	85.81	80.28	62.93	82.79	87.41
🗨️	moxeh/MoMo-72B-LoRA-1.8.7-DPO	78.55	70.82	85.96	77.13	74.71	84.06	78.62
🗨️	tenyx/Llama3-TenxChat-70B	78.4	72.1	86.21	80.04	62.85	82.95	86.28

cookies
son

tal of
/16 =

EVALUATION

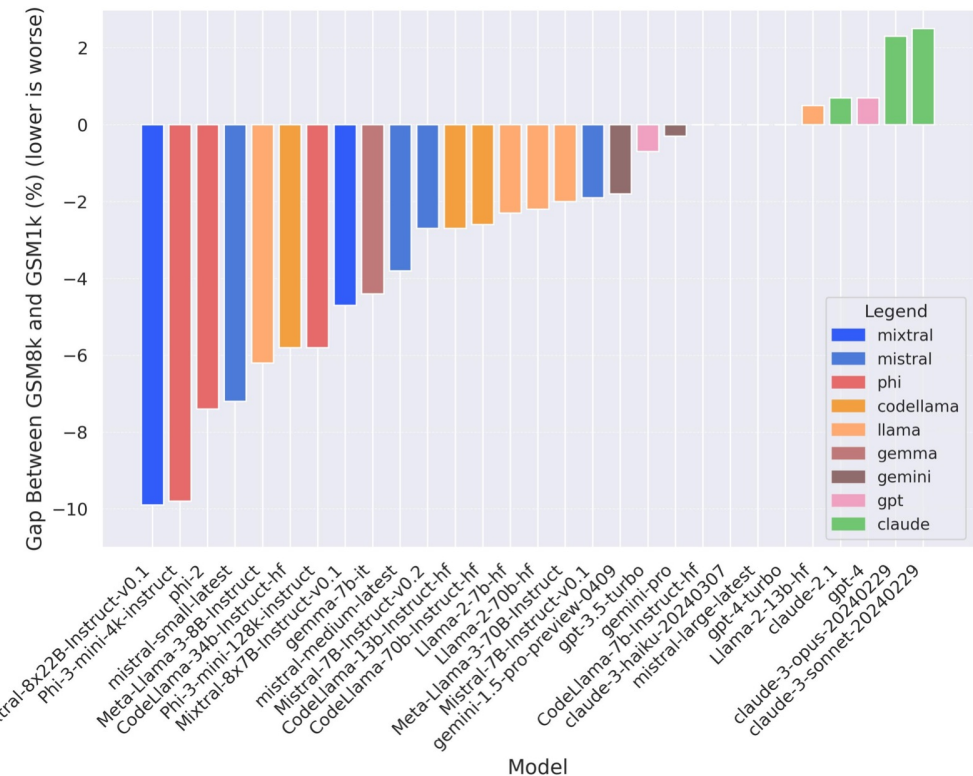
What are potential problems of these open benchmarks?

EVALUATION

- Leaderboards can be gamed
- Developers might overfit to a certain benchmark
- Webcrawling leads to (unintentional) training data contamination

1075A	The King's Race	implementation, math	🚩 ⭐	800	x7779	1802A	Likes	greedy, implementation	🚩 ⭐	800	x10715
1065A	Vasya and Chocolate	implementation, math	🚩 ⭐	800	x14032	1800A	Is It a Cat?	implementation, strings	🚩 ⭐	800	x19580
1064A	Make a triangle!	brute force, geometry, math	🚩 ⭐	800	x19739	1799A	Recent Actions	data structures, greedy, implementation, math	🚩 ⭐	800	x9233
1061A	Coins	greedy, implementation, math	🚩 ⭐	800	x19476	1796A	Typical Interview Problem	brute force, implementation, strings	🚩 ⭐	800	x15004
1060A	Phone Numbers	brute force	🚩 ⭐	800	x12567	1795A	Two Towers	brute force, implementation, strings	🚩 ⭐	800	x19658
1056A	Determine Line	implementation	🚩 ⭐	800	x6028	1794A	Prefix and Suffix Array	strings	🚩 ⭐	800	x12561
1054A	Elevator or Stairs?	implementation	🚩 ⭐	800	x8520	1793A	Yet Another Promotion	greedy, math	🚩 ⭐	800	x15111
1047A	Little C Loves 3 I	math	🚩 ⭐	800	x17785	1792A	GamingForces	greedy, sortings	🚩 ⭐	800	x22860
1043A	Elections	implementation, math	🚩 ⭐	800	x10356	1791C	Prepend and Append	implementation, two pointers	🚩 ⭐	800	x28936
1041A	Heist	greedy, implementation, sortings	🚩 ⭐	800	x22026	1791B	Following Directions	geometry, implementation	🚩 ⭐	800	x31133

<https://x.com/CHHillee/status/1635790330854526981>



Zhang, Hugh, et al. "A careful examination of large language model performance on grade school arithmetic." *arXiv:2405.00332* (2024).

CONTRIBUTE @ OCCIGLOT



manuel@occiglot.org



QUESTIONS?

