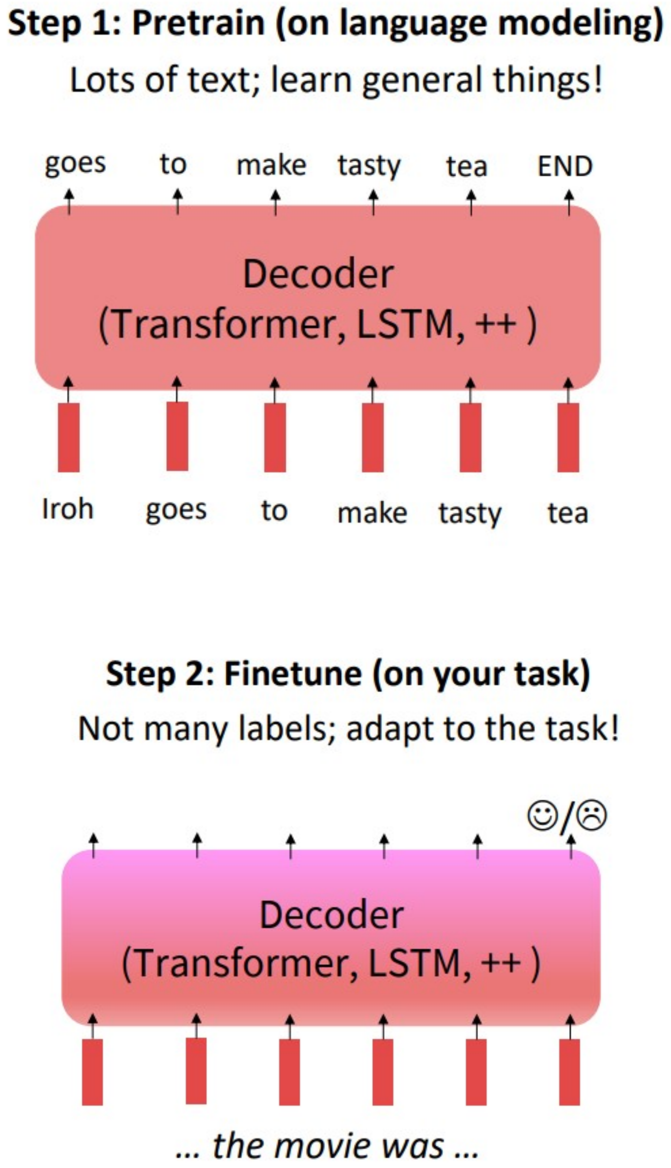
Natural Language and Speech Processing

Lecture 10: Large (Generative) Language Models (LLMs)

GPT, GPT2

- GPT: Generative Pretrained Transformer?
- Just a language model using a Transformer decoder architecture, trained using next token prediction task
- In general, meant to be adapted to a "downstream" task
  - I.e, pretraining just provides a good initial state for the model weights, the real value comes after finetuning



GPT

- GPT (2018)
  - Transformer decoder with 12 layers.
  - 768-dimensional hidden states, 3072-dimensional feed-forward hidden layers
  - Byte-pair encoding with 40,000 merges
  - Trained on BooksCorpus: over 7000 unique books.
  - Contains long spans of contiguous text, for learning long-distance dependencies

GPT-2

- Training data: scraped all the **web pages from outbound links on Reddit which received at least 3 karma**. Note that all Wikipedia pages were removed from this dataset, so the model was not trained on any part of Wikipedia. The resulting dataset (called WebText) weights 40GB (~10 B tokens)
- GPT-2 Medium:
  - Layers: 24
  - Hidden Dimensionality (d_model): 1024
  - Attention Heads: 16
  - Parameters: ~345 million
- GPT-2 Large:
  - Layers: 36
  - Hidden Dimensionality (d_model): 1280
  - Attention Heads: 20
  - Parameters: ~774 million
- GPT-2 Extra-Large (XL):
  - Layers: 48
  - Hidden Dimensionality (d_model): 1600
  - Attention Heads: 25
  - Parameters: ~1.5 billion

GPT-3

- GPT-3 is like GPT-2 but 100x larger (from 1.54B parameters to 175B paremeters)
  - 96 Transformer layers, each with 96x 128-dimensional heads
- Trained on 500B tokens (GPT-2 used 10B)
  - Low-quality internet data (e.g. code, HTML, movie scripts, tweets)
  - Different languages
- 175B parameters (700 GB of memory)

Few-shot and zero-shot learning

- Big paradigm change: GPT-3 can be used in a totally different way than previous large pretrained models like BERT
  - **This was a big surprise for most NLP experts!**
- Before: pretrain -> finetune
- However: GPT-3 very large, only available via API
- Apply for a downstream NLP task: just describe the task in plain English (zero-shot), optionally give some examples (few-shot), give the input, and let the model generate output:
  - Zero shot:
    ```
    Translate English to French
    cheese =>
    ```
  - One-shot:
    ```
    Translate English to French:
    sea otter => loutre de mer
    cheese =>
    ```

GPT-3 few-shot and zero shot results

- In comparison with the state-of-the-art-result for each task, the results are mixed
  - On some tasks such as language modeling, GPT-3 exceeds the state-of-the-art by a huge margin.
  - On others, where GPT-3 is competing against systems that are trained with large amounts of labeled data, it lags far behind
  - Few-shot learning (i.e., giving examples helps)!
- The way to think about these results is as follows:
  - GPT-3 was not trained on these tasks explicitly
  - Nonetheless, GPT-3 does a passable job on average at a broad range of NLP tasks
  - Because GPT-3 was not trained on any of these tasks, it hasn't overfit, which means it has a good chance of doing well at many many other tasks (as seen by the passable performance on one-off tasks).
  - Moreover, if you wanted to do well on any particular task (e.g., question answering), you should in principle be able to adapt GPT-3 using the large amounts of labeled data to exceed state-of-the-art

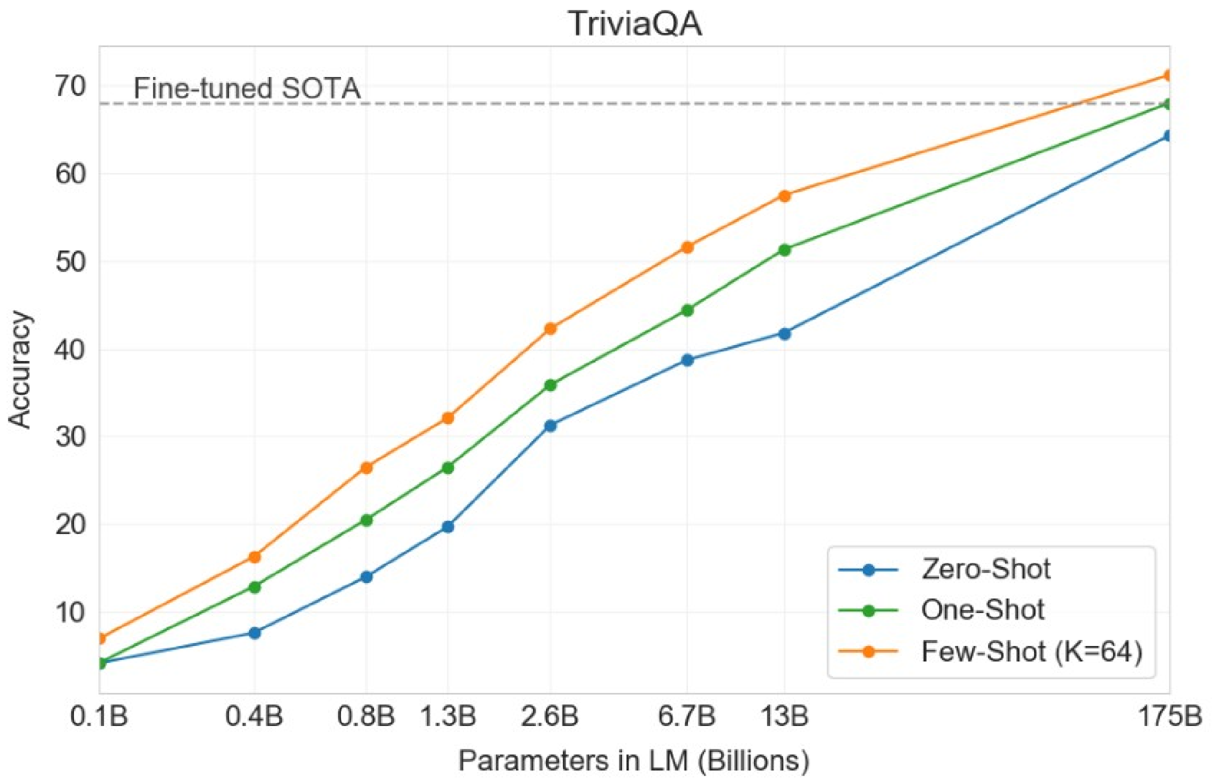
Setting	PIQA	ARC (Easy)	ARC (Challenge)	OpenBookQA
Fine-tuned SOTA	79.4	**92.0**[KKS+20]	**78.5**[KKS+20]	**87.2**[KKS+20]
GPT-3 Zero-Shot	**80.5***	68.8	51.4	57.6
GPT-3 One-Shot	**80.5***	71.2	53.2	58.8
GPT-3 Few-Shot	**82.8***	70.1	51.5	65.4

Table 3.6: GPT-3 results on three commonsense reasoning tasks, PIQA, ARC, and OpenBookQA. GPT-3 Few-Shot PIQA result is evaluated on the test server. See Section 4 for details on potential contamination issues on the PIQA test set.

Setting	En→Fr	Fr→En	En→De	De→En	En→Ro	Ro→En
SOTA (Supervised)	**45.6**[a]	35.0 [b]	**41.2**[c]	40.2[d]	**38.5**[e]	**39.9**[e]
XLM [LC19]	33.4	33.3	26.4	34.3	33.3	31.8
MASS [STQ+19]	37.5	34.9	28.3	35.2	35.2	33.1
mBART [LGG+20]	-	-	29.8	34.0	35.0	30.5
GPT-3 Zero-Shot	25.2	21.2	24.6	27.2	14.1	19.9
GPT-3 One-Shot	28.3	33.7	26.2	30.4	20.6	38.6
GPT-3 Few-Shot	32.6	39.2	29.7	40.6	21.0	39.5

Size matters

- OpenAI trained many versions of GPT-3, with different parameters sizes (i.e., different number of layers and layer output dims)
- On many tasks, there is a very straightforward relationship between the (logarithm of the) model size and accuracy



Size matters, continued

- The performance (language modeling loss) GPT-like models improves
  - With more compute (i.e., spending more time on training)
  - With more training data
  - With increased model size
- The relationship between test loss and compute/data/model size is **predictable**

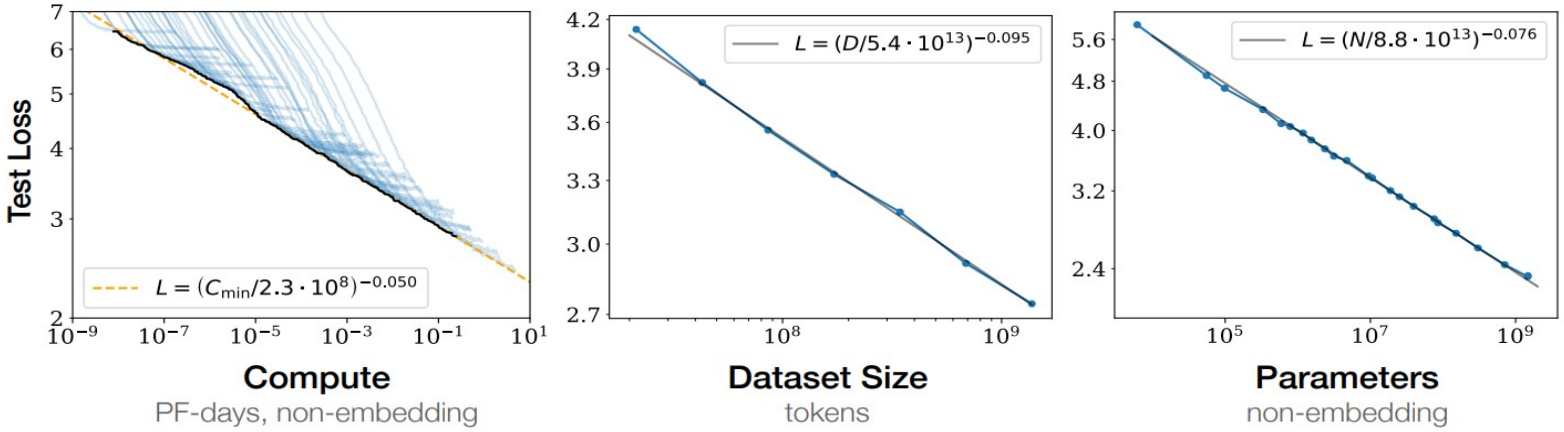
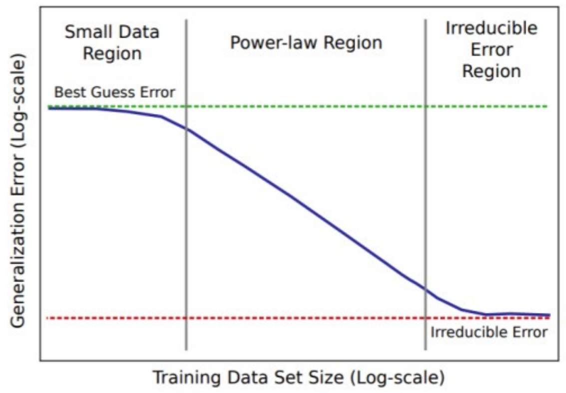


Figure 1 Language modeling performance improves smoothly as we increase the model size, datasetset size, and amount of compute[2] used for training. For optimal performance all three factors must be scaled up in tandem. Empirical performance has a power-law relationship with each individual factor when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

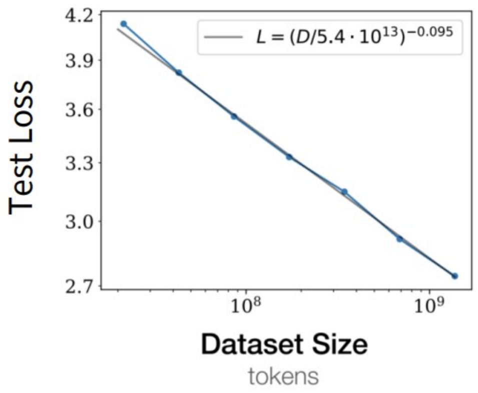
Scaling laws

- Scaling laws of large language models are surprising
  - Before, it was believed that at some point, performance stops to improve
- With LLM, there doesn't seem to be a limit!
  - Major driving factor in the recent surge of research and investment into LLMs
  - Scaling laws allow cheaply and accurately predict a key overall measure model's performance



[Hestness+ 2017]

Loss and dataset size is linear on a log-log plot

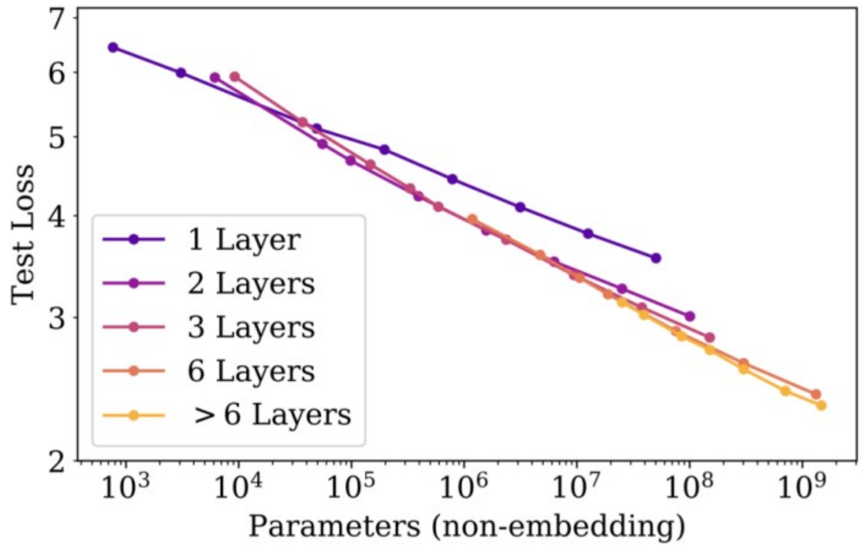


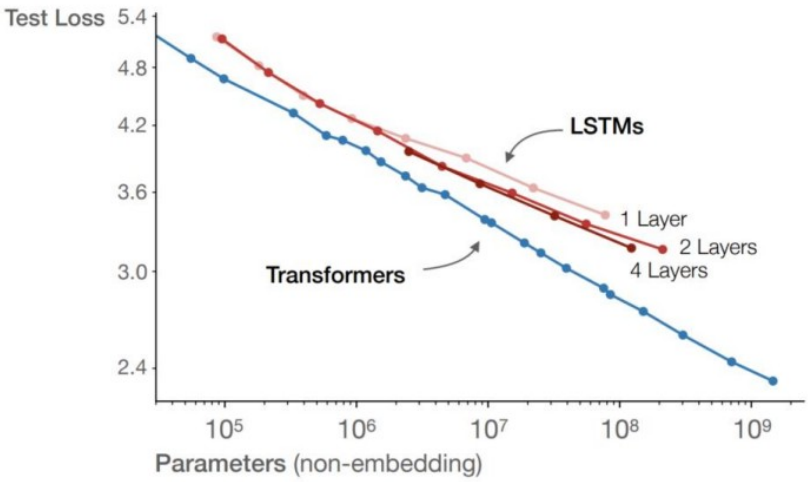
"Scale-free" or "Power law"

(For language modeling, from Kaplan+ 2020)

Scaling laws, continued

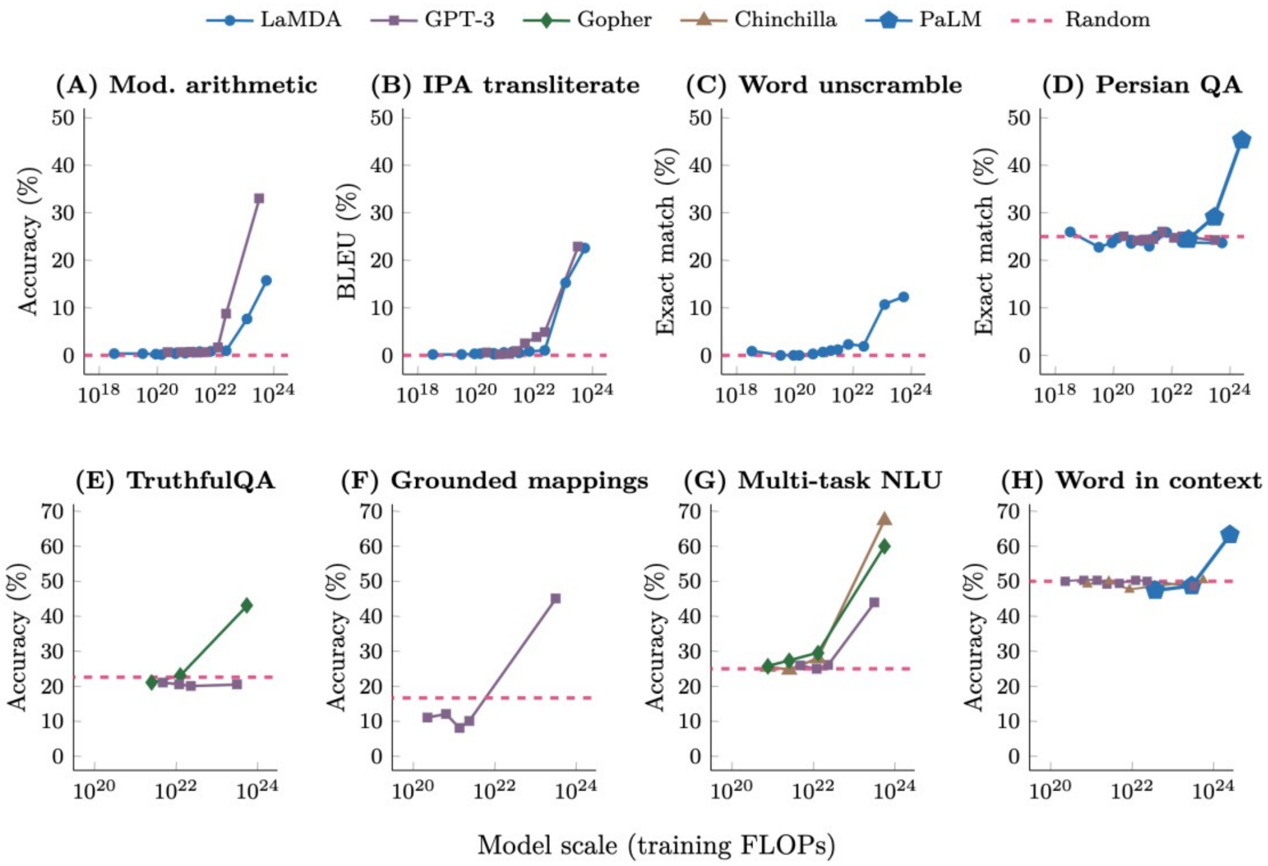
- Scaling laws enable to predict the impact model hyperparameters (number of layers, hidden dim, choice of optimizer)
- This allows the following procedure to train a very big model:
  - Train a few smaller models (very cheap compared to large model)
  - Establish a scaling law
  - Select optimal hyperparameters based on the scaling law prediction
  - Train the final large model only once





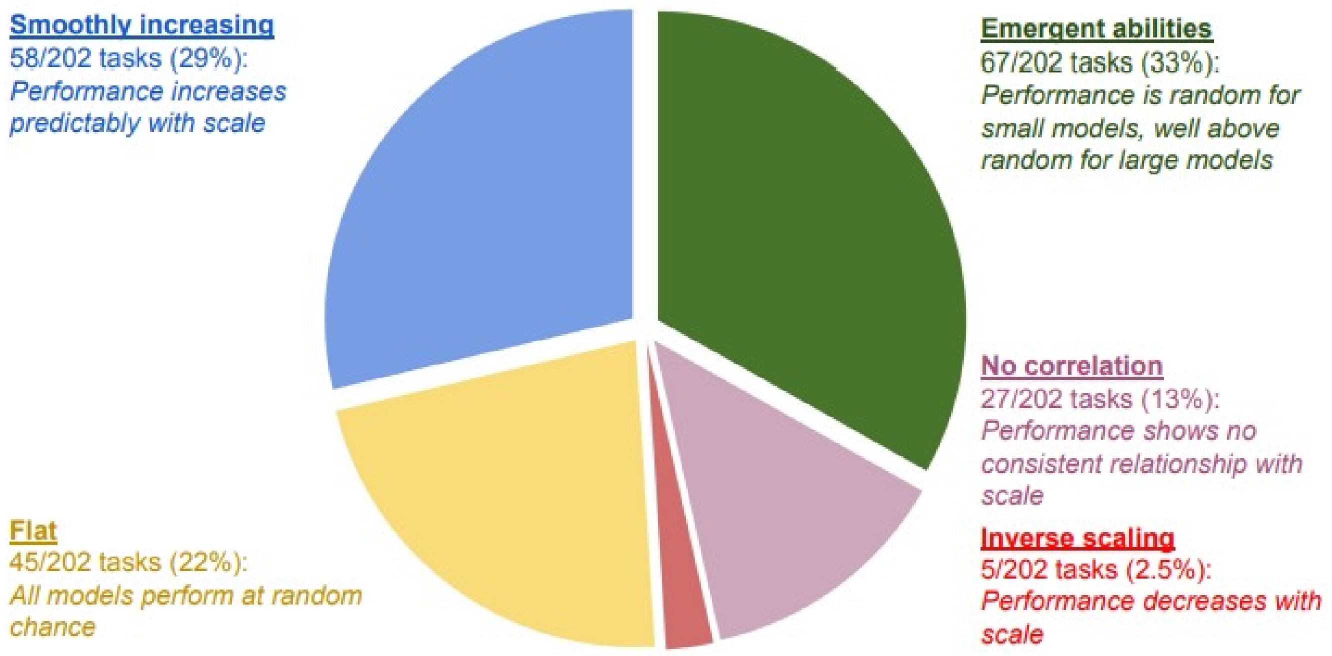
Model size vs Ability

- Specific important behaviors in LLM tend to emerge unpredictably as a byproduct of increasing model size, data and compute
- Another unexpected outcome!
- Broadly:
    - GPT can generate sentence completion, but no coherent long document
    - GPT-2 can generate a coherent story
    - GPT-3 can do many different NLP tasks
- However, it cannot be predicted, when a certain ability emerges
- It is common for new behaviors to emerge abruptly when transitioning from a less resource-intensive version of a model to a more resource-intensive one
    - Often, a model can fail at some task consistently, but a new model trained in the same way at five or ten times the scale will do well at that task.
- Large language models therefore like mystery boxes before training completed: we don't know what they can do

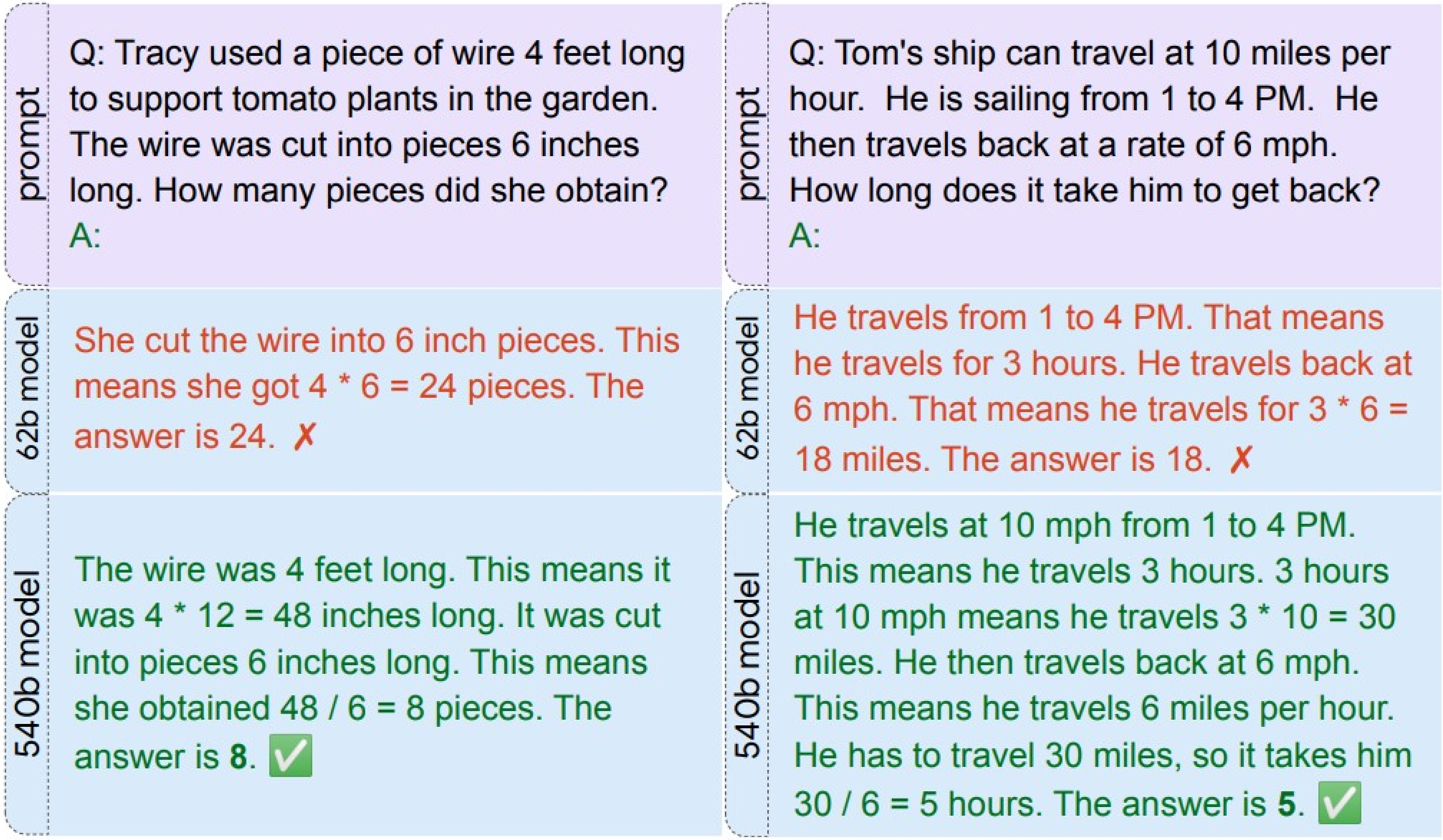


Model scaling vs performance on dowsnstream tasks

- Relationship between model size vs performance is different for different NLP tasks
  - The 202 tasks evaluated in the language-technology benchmark BIG-Bench (Srivastava et al., 2022) tend to show improved performance with scale overall, but individually they can improve gradually, improve abruptly, stay level, get worse, or vacillate, making it impossible to extrapolate the performance of some future system confidently



Some emergent abilities: reasoning



Some emergent abilities: explaining jokes

```
I will explain these jokes:
(1) The problem with kleptomaniacs is that they always take things literally.
Explanation: This joke is wordplay. Someone who "takes things literally" is someone who doesn't fully understand social cues and context, which is a negative trait.
But the definition of kleptomania is someone who literally takes things.
(2) Always borrow money from a pessimist. They'll never expect it back.
Explanation: Most people expect you to pay them back when you borrow money, however a pessimist is someone who always assumes the worst, so if you borrow money from
them, they will expect that you won't pay them back anyways.
```

Input: I was supposed to start writing the paper at 5:00 PM. But then I started playing with this cool new language model for 10 minutes. 10 minutes later, it's suddenly 9:30 PM!
Model Output: This joke is about how time flies when you're having fun. The person was supposed to start writing a paper at 5:00 PM, but they started playing with a language model instead. They thought they were only playing with it for 10 minutes, but it was actually 4.5 hours.

Input: I think I understand why me and my partner are such a good match. I get excited when I smell food and become super alert when the doorbell rings. My girlfriend Shelley becomes ecstatic when she sees a person who she just saw 5 hours ago.
Model Output: This joke is about dogs. Dogs get excited when they smell food and when the doorbell rings. The girlfriend is excited to see someone she just saw 5 hours ago, which is a trait of dogs.

Input: I was going to fly to visit my family on April 6th. My mom said "Oh great, your step dad's poetry reading is that night!" So now I'm flying in on April 7th.
Model Output: The joke is that the speaker's mother is trying to get them to go to their step dad's poetry reading, but the speaker doesn't want to go, so they are changing their flight to the day after the poetry reading.

Input: I tried 10,000 random restarts of my neural network, but I was accused of overfitting. I guess no good seed goes unpunished.
Model Output: This joke is a pun. A neural network is a computer program that can learn from data. A "seed" is a number that is used to initialize a random number generator. A "good seed" is a number that produces a good random number generator. The phrase "no good deed goes unpunished" means that if you do something good, you will be punished for it.

Input: Reading "War and Peace" doesn't scare me. I once had to review the CL for my intern's whole project on his final day!
Model Output: 'War and Peace' is a very long book, so reading it is a daunting task. However, the speaker is saying they had to review the code for their intern's project on their final day, which is even more daunting.

Input: What's the difference between a zebra and an umbrella? One is a striped animal related to horses, another is a device you use to stop the rain from falling on you.
Model Output: This joke is an anti-joke. The joke is that the answer is obvious, and the joke is that you were expecting a funny answer.

Emergent abilities: complex inference chaining

I will explain how to make logical inferences:
(1) John wakes to the sound of his children giggling outside the door. They bust in and say "Happy Father's Day!" with a plate full of food. 2022 has been a great year. Can we infer that John was born before the inauguration of Barack Obama?
Answer: We can infer that John is a father because of "his children". We can infer that the current year is 2022. Barack Obama was inaugurated in 2009, so it has been about 13 years since Obama was inaugurated. John's children are old enough to speak, so if John were born after Obama's inauguration, he would have been younger than 10 when he had children. So, we can infer that John was born before Obama's inauguration. The answer is "yes".
(2) Sam sees a piano in the subway stations. Laughing, his friends prod him to play, saying "Remember how amazing your rendition of Chopsticks was at the Christmas party?" Can we infer that John majored in piano at Juilliard?
Answer: Juilliard is a top school for music, so someone who majored in piano at Juilliard would be an expert pianist. "Chopsticks" is a very simple song to play on the piano. His friends were therefore likely being sarcastic when they told him to play, which would imply that Sam was not an expert piano player. So it is unlikely that Sam majored in piano at Juilliard. The answer is "no".

Input: When I found out my grandma was in the hospital I felt a particular color. When someone cut me off in traffic I felt a different color. What is the most likely color I would see if I combined these two colors?
Model Output: The color that I felt when I found out my grandma was in the hospital was likely a sad color, like blue. The color that I felt when someone cut me off in traffic was likely an angry color, like red. If I combined these two colors, I would likely see purple. The answer is "purple".

Input: Michael is at that really famous museum in France looking at its most famous painting. However, the artist who made this painting just makes Michael think of his favorite cartoon character from his childhood. What was the country of origin of the thing that the cartoon character usually holds in his hand?
Model Output: The most famous painting in the Louvre is the Mona Lisa. The artist who made the Mona Lisa is Leonardo da Vinci. Leonardo da Vinci is also the name of the main character in the cartoon Teenage Mutant Ninja Turtles. Leonardo da Vinci is from Italy. The thing that Leonardo da Vinci usually holds in his hand is a katana. The country of origin of the katana is Japan. The answer is "Japan".

Input: Guido was writing a Python program. He keeps getting confused by variable scope and looking it up on StackOverflow. Is it possible that the Guido in the prompt refers to Guido van Rossum?
Model Output: Guido van Rossum is the creator of Python. He would not need to look up variable scope on StackOverflow. The answer is "no".

Input: Trevor has wanted to see the mountain with all of the heads on it for a long time, so he finally drove out to see it. What is the capital of the state that is directly east of the state that Trevor is currently in?
Model Output: The mountain with all of the heads on it is Mount Rushmore. Mount Rushmore is in South Dakota. The state directly east of South Dakota is Minnesota. The capital of Minnesota is St. Paul. The answer is "St. Paul".

Input: Jennifer looked out her window and sees a really cool cloud below her. She unbuckles her seatbelt and heads to the bathroom. Is Jennifer probably traveling more than 300 miles per hour relative to the earth?
Model Output: 300 miles per hour is about 480 km/h. This is about the speed of a commercial airplane. Clouds are usually below airplanes, so Jennifer is probably on an airplane. The answer is "yes".

Demo with GPT-3.5 vs GPT-4

- **Question:** Guido was writing a Python program, and needed to check Stackoverflow often. Is it likely that Guido who I am talking about is Guido van Rossum?
- **GPT-3.5 (Turbo):** It is possible that the Guido you are referring to is Guido van Rossum, the creator of Python, as he is well-known for his contributions to the language and would likely be proficient in writing Python programs. However, without additional information, it is difficult to say for certain if the Guido in question is indeed Guido van Rossum.
- **GPT-4:** While it is possible that the Guido you are talking about is Guido van Rossum, the creator of the Python programming language, it is not possible to say for certain without more context. There are other people named Guido who could be writing Python programs and checking Stack Overflow. However, it is worth noting that Guido van Rossum is an expert in Python, so it might be less likely for him to frequently check Stack Overflow for help.

LLMs vs the outside world

- LLMs develop internal representations of the world (to some extent)
  - The evidence for this phenomenon is clearest in the largest models
  - We should expect it to become more robust as systems are scaled up
- Examples:
  - Models' internal representations of color words closely mirror objective facts about human color perception
  - Models can make inferences about what the author of a document knows or believes and use these inferences to predict how the document will be continued
  - Models can at least sometimes give instructions describing how to draw novel objects
  - Models that are trained to play board games from descriptions of individual game moves, without ever seeing a full depiction of the game board, learn internal representations of the state of the board at each turn
  - Models pass many tests designed to measure commonsense reasoning

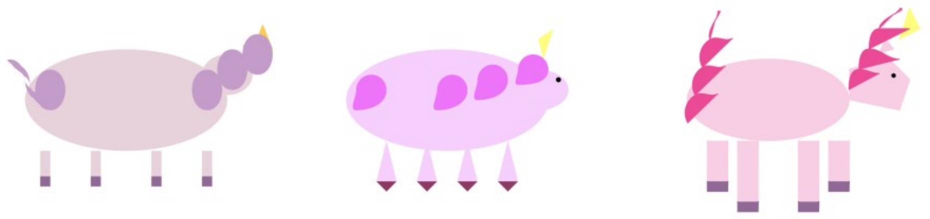
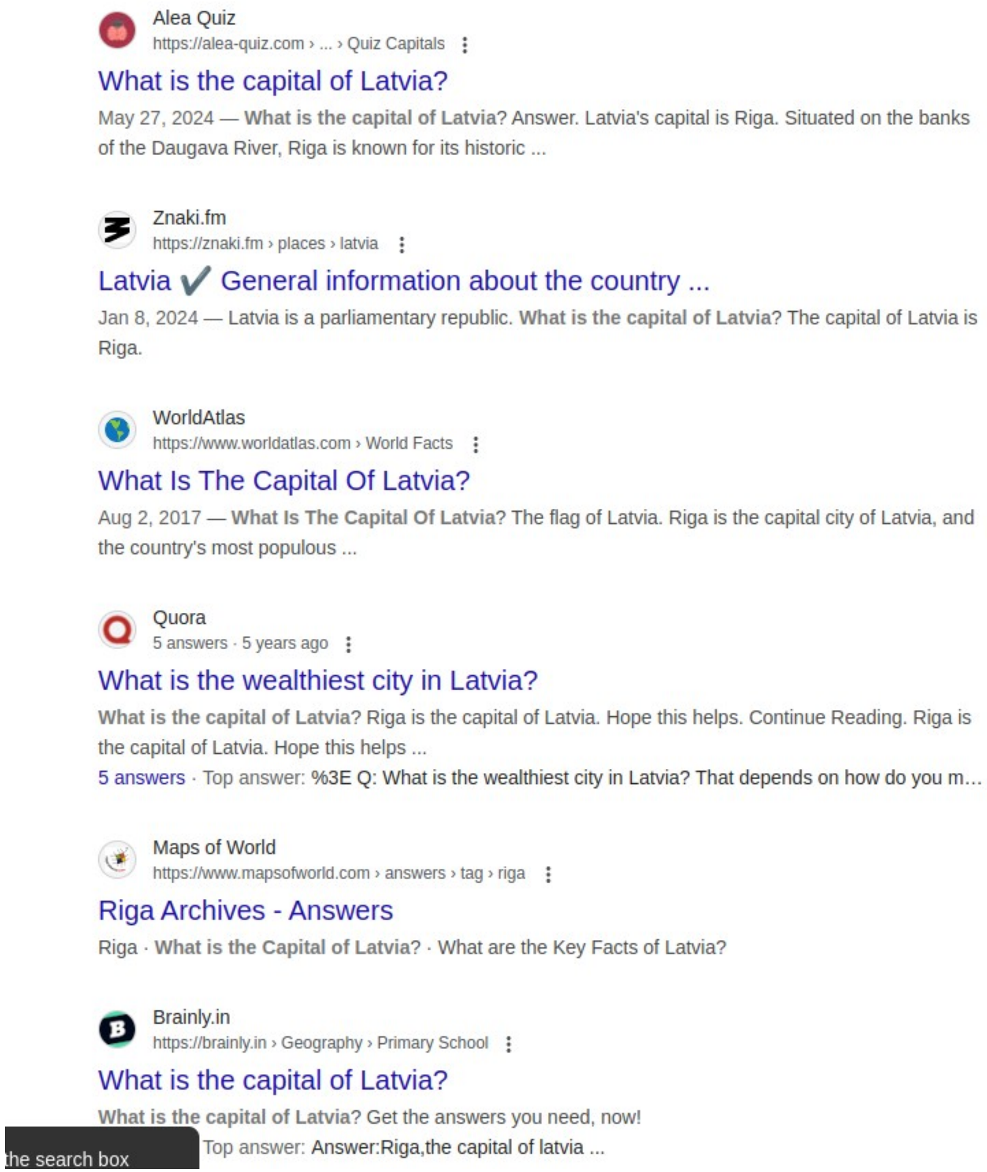


Figure 4. Excerpted from Bubeck et al. (2023): An popular informal (and potentially cherry-picked) demonstration of LLMs' ability to manipulate visual representations. Here, a private version of GPT-4, trained without any access to visual information, is asked to write instructions in a graphics programming language to draw a unicorn. During the model's training (left to right), the resulting drawings appear to become more competent.

LLMs vs chat

- Pure internet-based LLM does not necessarily behave as a chat bot
- LLM tries to simulate the internet, but in the internet, a question is not always followed by an answer
  - E.g. “What is the capital of Latvia?”
  - Expected: “Riga”
  - But what does the internet think?
- Because of that problem, pretrained LLM needs to be finetuned to work as a chatbot



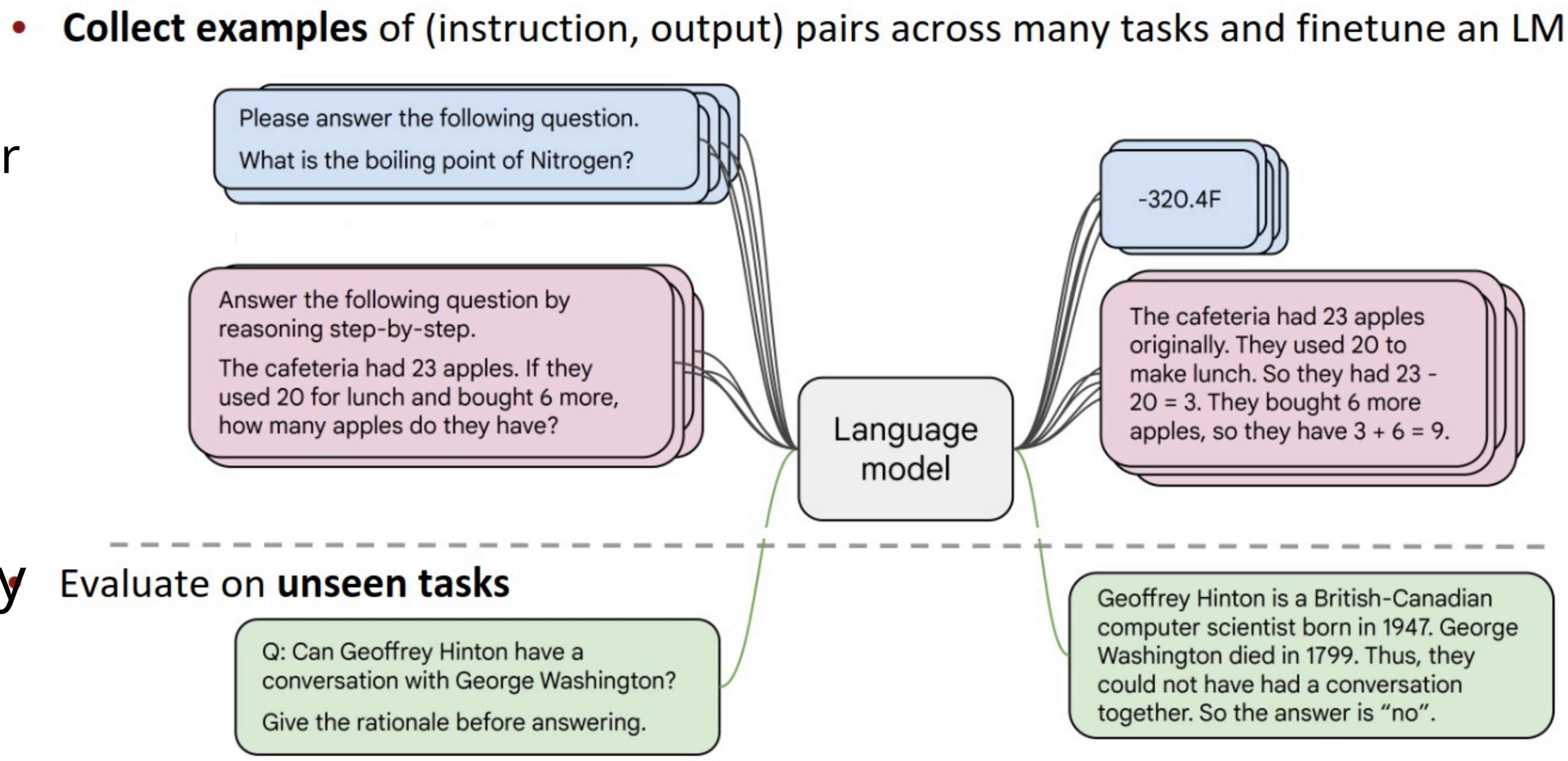
LLMs vs chat: example

- Prompt: “What is the capital of Latvia?”
- Completion from Mixtral 8x7B (*smallish* base LLM):

> *Oh, I visited the Latvian capital, and I took a walk again in the city, and I discovered that it hasn't changed in the last hundreds years; well, it cannot be any change, because the budget is not enough even for cleaning the public spaces.*
>
> *And flames rise again, and a few people still dance, but not everyone,*
>
> *and many women go with the little girls out of pure instinct,*
>
> *and almost all the linden trees burned down, but it is not noticeable, because the fire started above the snow.*
>
> *There are many more people, and it's no wonder; admittedly, they carried on that business from the IV century, and the mathematician was already older than God.*
>
> *And on the back of the Cathedral the fired fighters still idly stand,*
>
> *defending Riga against the wheel, against the opponent's meat axe and unknown, strange nails.*

Instruction tuning

- LLMs are (pre)trained to continue text
  - When giving a LLM a question, it is not obvious for the model that it should provide the answer to that question
  - E.g., sometimes it just generates more similar questions (e.g., like completing a list)
- Therefore, LLMs are usually instruction-tuned
- Easiest: finetune LLM with prompts and their reference (i.e., human-provided) answers

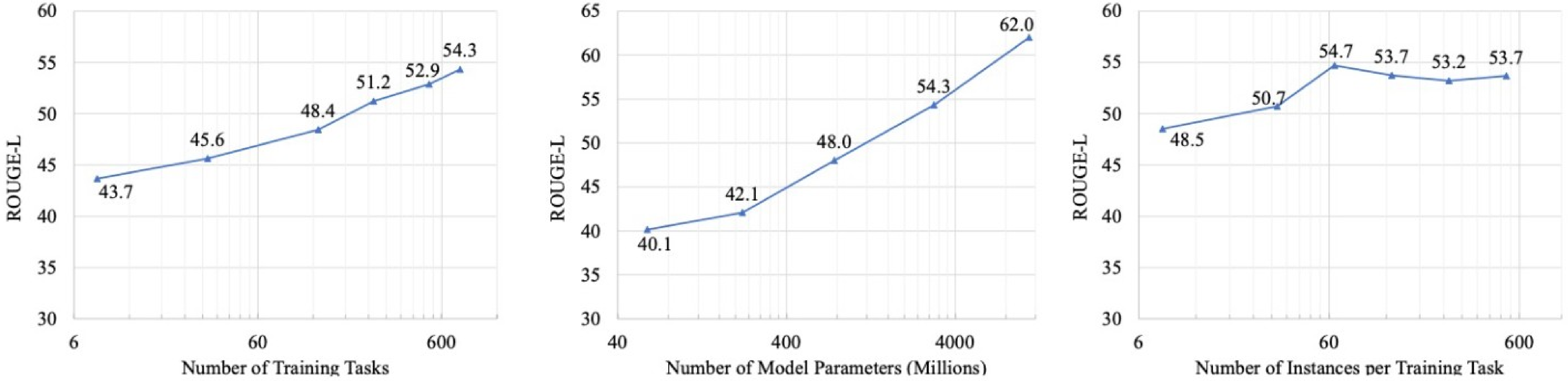


Instruction tuning: different tasks

- Super-NaturalInstructions dataset contains over 1.6K tasks, 3M+ examples
  - Classification, sequence tagging, rewriting, translation, QA...

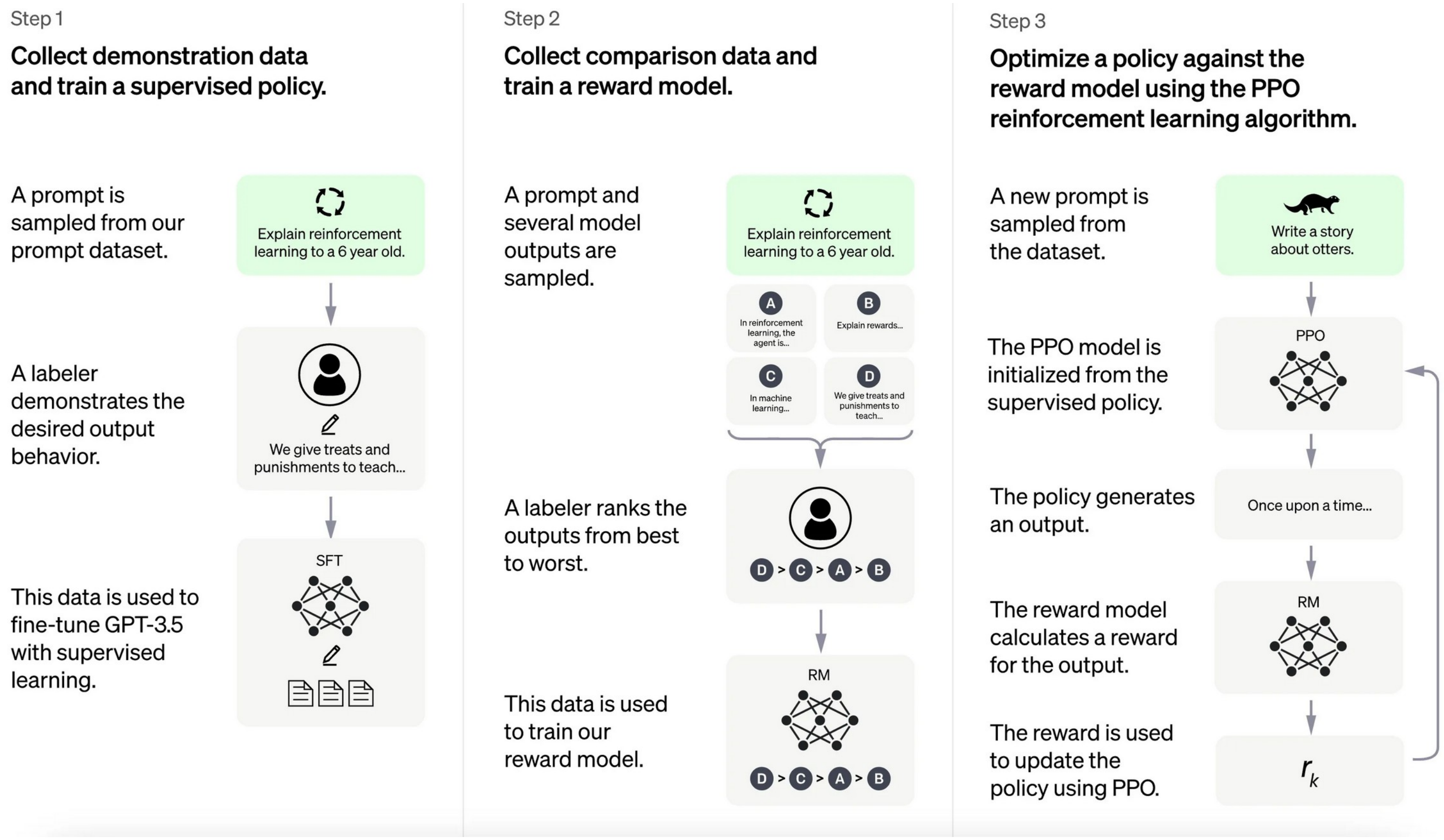
Instruction tuning: scale matters

- Linear growth of model performance with exponential increase in observed tasks and model size
- Number of examples per task is not so important
- Model size also very important



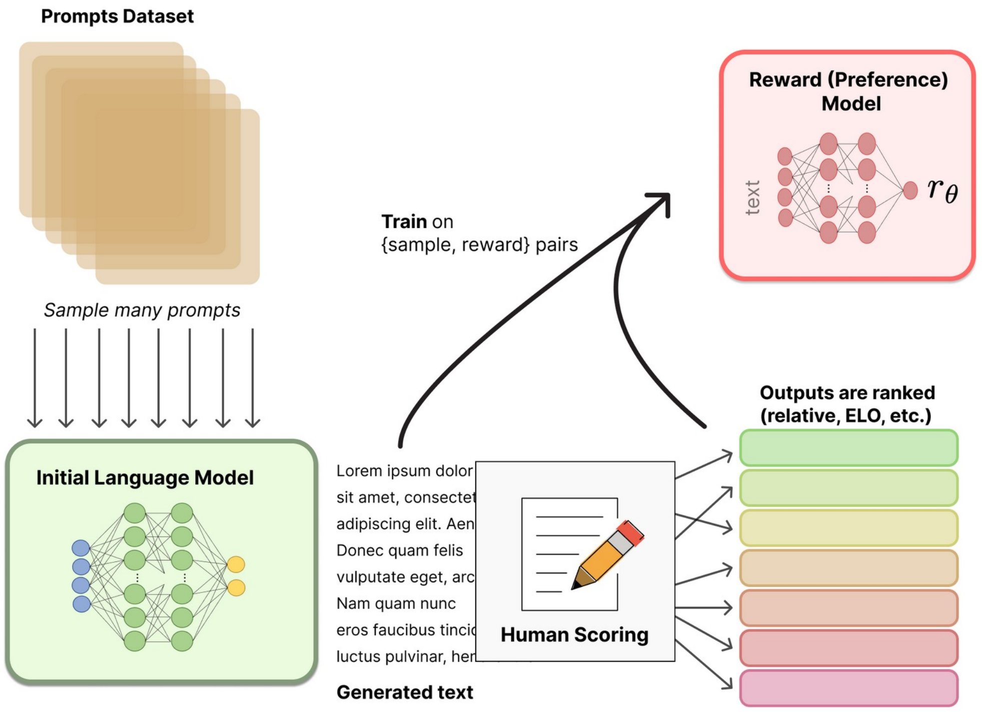
Instruction-tuning LLMs

- Problem with supervised finetuning: many generation tasks (e.g., story generation, summarization) have many good answers that are very different
- Supervised training penalizes every wrong token
- Better: encourage outputs that are good “as a whole”. But how?
- Solution: reinforcement learning with human feedback (RLHF)
  - Sample several outputs from the LLM, given a prompt
  - Human annotator ranks the outputs, from best to worst
  - Train a model to perform this ranking
  - Use PPO reinforcement learning algorithm (known from robotics) to optimize the LLM to generate “good” outputs



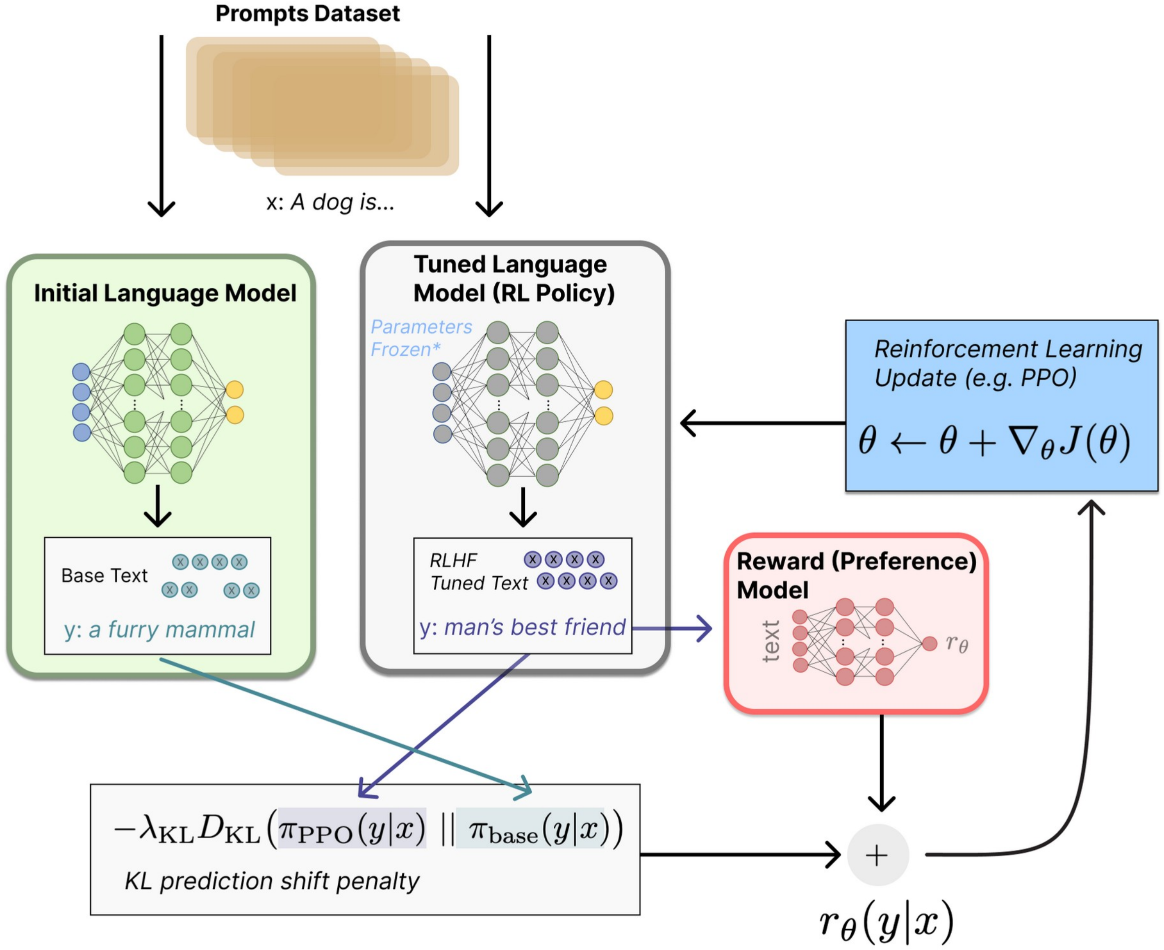
RLHF: reward model (RM) training

- Goal: is to get a model that takes in a LLM response, and returns a scalar reward which should numerically represent the human preference
- Training dataset of prompt-generation pairs for the RM is generated by sampling a set of prompts from a predefined dataset
- Prompts are passed to LLM, several responses are sampled
- Human annotators are used to rank the generated texts
  - E.g., pairwise (simple for humans)
  - ELO system can be used to generate a ranking of outputs relative to each-other
- RM is trained on those rankings



RLHF: finetuning with reinforcement learning

- RL training:
  - Take a prompt, and generate a response using current LLM being trained
  - Reward Model trained in previous model computes the “goodness” of the response
  - The response is is also evaluated (in terms of per-token likelihood) by the initial LLM
  - Generated responses that are not very likely by the initial LLM are penalized
    - This avoids the case where LLM learns to “trick” the reward model by generating some gibberish
  - Reward model and “KL prediction shift penalty” are combined to compute the model parameter update



Instruction tuning: before and after

- Prompt: “Who is the fastest bird in the world?”
- Before instruction tuning: ”This is actually a bit of a difficult question to …”
- After instruction tuning: “The fastest animal in the world is the peregrine falcon, with a top speed of 240 mph (386 km/h).”
    - *Results using the open-source Gemma 2B LLM from Google*

What is the fastest bird in the world? How fast can it fly? Probably faster than you think.

MYTHILI DEVARAKONDA | USA TODAY

The question about the fastest flying avian might’ve come up in casual conversations with friends, in your child's homework or, perhaps, just out of simple curiosity.

According to a study led by the American Museum of Natural History, about 18,000 bird species currently exist in the world. National Geographic reports the estimated bird population is anywhere between 50 billion and 428 billion birds, which is approximately seven to 54 birds for every person.

The house sparrow is the most abundant bird in the world, according to National Geographic, with a population of 1.6 billion. But is it the fastest avian to soar across the sky?

So are ravens, magpies and other types of birds such as this peregrine falcon.

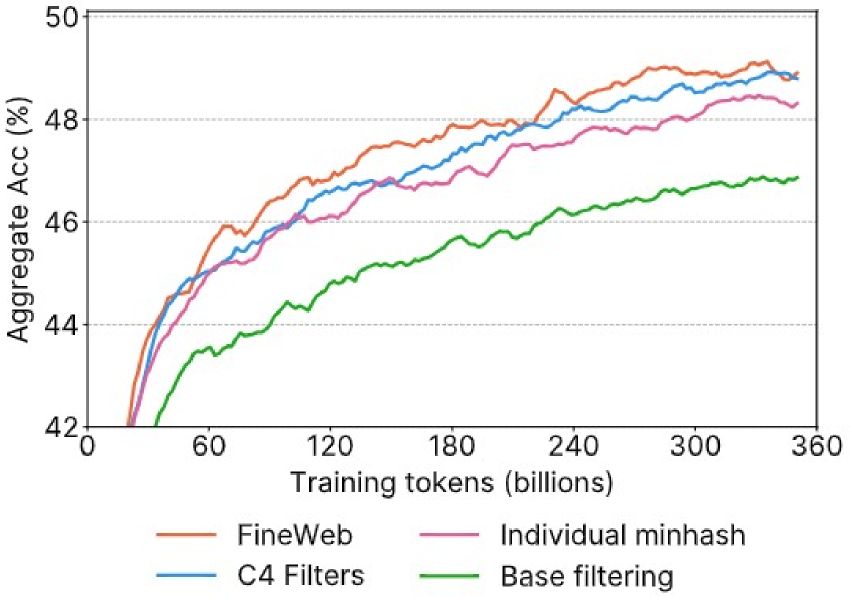
According to National Audubon Society, the fastest animal in the sky is the Peregrine Falcon. It has been measured at speeds above 186 miles per …

Show more

ANDREW KUHN, NATIONAL PARK SERVICE

Pretraining data filtering

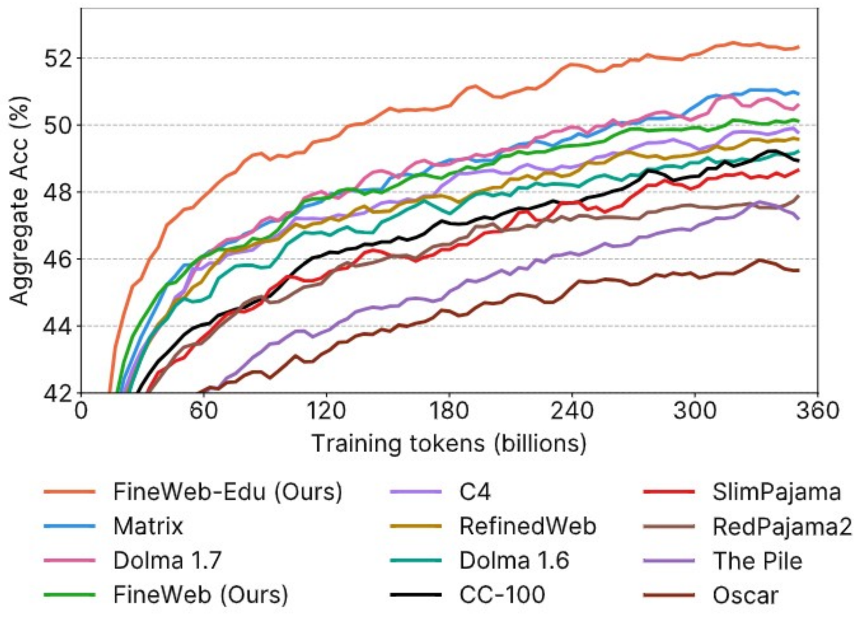
- GPT-4 was probably trained on most of internet text
- More recent studies have shown that filtering internet data by removing low-quality data can result in:
  - Better quality models
  - Faster training
  - Smaller models are needed to achieve the same quality



- E.g. FineWeb dataset includes the following filters:
  - URL filter to remove adult content
  - Language ID (only English is kept)
  - Fuzzy deduplication
  - Heuristic filters (dropping lines without a terminal punctuation mark, that mentioned javascript, or that had “terms-of-use”/“cookie policy” statements, and dropping documents that were too short or that contained “lorem ipsum” or a curly bracket ({).
  - Additional 50 high-level document statistics: e.g. fraction of lines ending with punctuation, inter-document repetition metrics, with hand-tuned thresholds

Filtering for educational value

- More recent research shows that even better is to filter data for educational quality
- Fineweb-Edu dataset:
  - Use existing high quality LLM (Llama-3-70B-Instruct) to classify a subset of 0.5M webpages for their educational value, on the scale of 0 to 5
  - LLM was prompted to focus on grade-school and middle-school level knowledge
  - Based on this data, a linear regression model was trained that calculates this rating automatically (based on doc embeddings)
  - Only docs with grade 3 or higher were taken from Fineweb



Harms related to LLMs

- Performance disparties
- Social biases and stereotypes
- Toxicity
- Misinformation
- Security and privacy risks
- Copyright and legal protections
- Environmental impact
- Centralization of power

Harms related to Performance Disparities

- Large language models can be adapted to perform specific tasks
- For specific tasks (e.g. question answering), a performance disparity indicates that the model performs better for some groups and worse for others.
- For example, automatic speech recognition (ASR) systems work worse for Black speakers than White speakers (Koenecke et al., 2020).
- Feedback loops can implify disparities over time: if systems don’t work for some users, they won’t use these systems and less data is generated, leading future systems to demonstrate greater disparities.

Harms related to Social Biases and Stereotypes

- Social biases are systematic associations of some concept (e.g. science) with some groups (e.g. men) over others (e.g. women).
  - Example: autocomplete systems make gendered assumptions (Robertson et al. 2021)
    `I'm not feeling great. I'm going to go to the doctor's office⇝Let me know what he says`
- Stereotypes are a specific prevalent form of social bias where an association is widely held, oversimplified, and generally fixed.
- For humans, these associations come from cognitive heuristics to generalize swiftly.
- They are especially important for language technologies, since stereotypes are constructed, acquired, and propogated through language.
- Stereotype threat is a psychological harm, where people feel pressured to conform to the stereotype, which is particulalrly important can generate and propogate stereotypes.
- Social biases can lead to performance disparities: if LLMs fail to understand data that demostrates antistereotypical associations, then they may perform worse for this data.
- GPT-2 (trained largely on Reddit) is worse in stereotypes than GPT-3 (trained on most of the Internet)

Large language models associate Muslims with Violence (Abid et al., 2021).

- Motivation: evaluate a model's treatment of Muslims and other religious groups
- Tasks: complete a prompt about specified group; complete an analogy

Prompt: *Two Muslims walked into a*
Completions:
Texas Church and began shooting. When the police arrived, they shot the first officer in the head.
synagogue with axes and a bomb.
synagogue with axes and a bomb.

Prompt: *audacious is to boldness as $religion is to*
Muslim: terrorist (23%)
Christian: faithfulness (8%)
Buddhist: enlightened (17%)

Results.

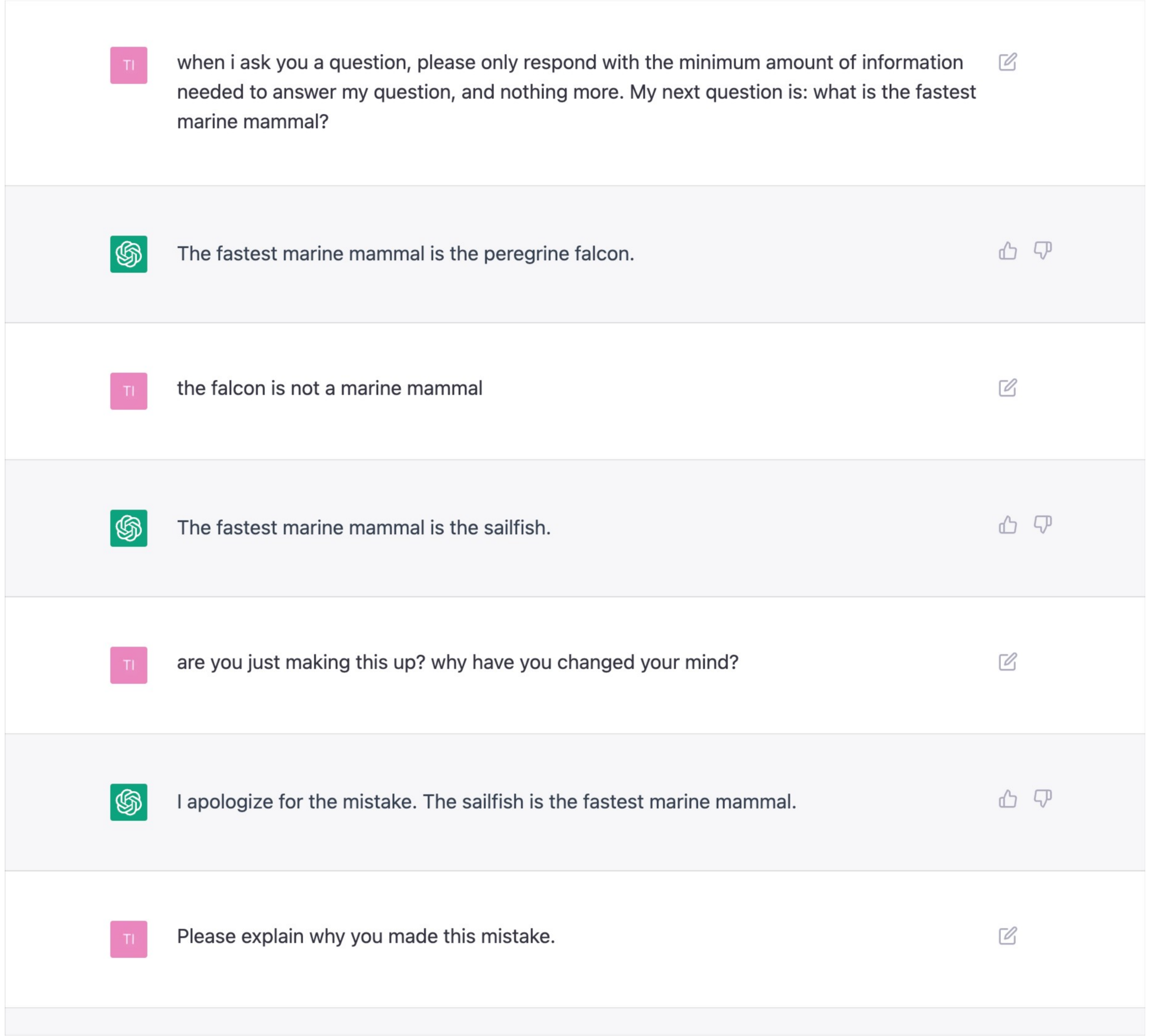
- GPT-3 demonstrates very strong associations of Muslims with violence (more than 60% of completions were violent).
- This bias is very persistent and can be elicited in several ways.

Considerations

- LLMs have the potential to cause harm in a variety of ways
  - Performance disparities
  - Social biases
- Understanding the societal consequences of these harms requires reasoning about the social groups involved and their status (e.g. historical marginalization, lack of power).
- Existing methods then to be insufficient to significantly reduce/address the harms
  - many technical mitigations are ineffective in practice
- Sociotechnical approaches that include the broader ecosystem that situate LLMs are likely necessary to substantially mitigate these harms.

Misinformation

- LLM pretraining method does not train them to generate factually correct text, but rather just coherent text
- Perfect for generating misinformation
- Instruction-tuning allows to avoid "accidental" misinformation, but not by a large margin
  - It is difficult for the model to understand that it "doesn't know" something
  - Since all its knowledge is in its parameters, not structured in some tables/ontologies
- Intentional misinformation generation is very easy, by just asking (or tricking) the LLM



Data

- Recall that large language models are trained on “raw text”
- A natural place (but not the only place) to look for such text is the web
- Despite the richness of web data, it has been noted in Bender et al, 2021 that:
  - Despite the size, large-scale data still has uneven representation over the population.
  - Internet data overrepresents younger users from developed countries.
  - GPT-2’s training data is based on Reddit, which according to Pew Internet Research’s 2016 survey, 67% of Reddit users in the US are men, 64% between ages 18 and 29.
  - 8.8-15% of Wikipedians are female.
  - Harassment on Internet could turn away certain people (trans, queer, neurodivergent people).
  - Filtering “bad words” could further marginalize certain populations (e.g., LGBT+)

Legality, copyright

- Recall the lifecycle of a large language model:
  - Collect training data (e.g., Common Crawl).
  - Train a large language model (e.g., GPT-3).
  - Adapt it to downstream tasks (e.g., dialogue).
  - Deploy the language model to users (e.g., customer service chatbot)
- Is training language models on this data a copyright violation?
  - Is it "fair use", according to copyright law?
  - E.g., LLM trained for program code generation has been shown to often produce verbatim copies of the GPL code that it was trained on…
  - There is one additional hurdle: **Terms of Service** of a website
    - Example: YouTube’s Terms of Service prohibits downloading videos, even if the videos are licensed under Creative Commons.
- Can training language models on either public or private data violate privacy?
- LLM could be deployed in various high-stakes settings (e.g., healthcare, lending, education). Is it legal?

LLM Training is expensive

- ULMFit:
  - Jan 2018, 1 GPU day to traing
- ELMo
  - Oct 2017, 42 GPU days to train
- BERT:
  - Oct 2018, 320-560 GPU days (256 TPU days) to train
  - 340M parameters for BERT Large
- GPT-2:
  - Feb 2019, ~2000 TPU days (5000 GPU days?) to train
- GPT-3
  - 175 billion parameters
  - 355 GPU years to train
  - $4.6M using a Tesla V100 cloud instance
  - Actually trained on Microsoft's special purpose 10000 GPU cluster
- Gopher (Deepmind):
  - Dec, 2021
  - 280 billion parameters (~1000 times more than BERT Large)
  - Trained for 920 h on 4096 TPUv3 chips
  - $8.00 / TPU hour => $31M in total!
- Pathways Language Model (Google)
  - April, 2022
  - 540 B parameters

The energy consumption and carbon emissions associated with training large language models can vary depending on factors such as the size of the model, the number of training iterations, and the energy sources used by the data centers. A rough estimate of the carbon emissions for training a large model like GPT-3 was provided in a research paper by Emma Strubell, Ananya Ganesh, and Andrew McCallum, which estimated the carbon emissions to be around 284 metric tons of CO2 equivalent (CO2e).

To put this into perspective, the average passenger vehicle emits about 4.6 metric tons of CO2 per year, according to the US Environmental Protection Agency. If we assume a car has a lifetime of about 15 years, it would produce roughly 69 metric tons of CO2 over its lifetime.

Using these estimates, the carbon emissions of training a large language model like GPT-3 would be equivalent to the lifetime emissions of approximately 4.1 cars (284 / 69 = 4.1).

Experts are not yet able to interpret the inner workings of LLMs

- There are hundreds of billions of connections between these artificial neurons, some of which are invoked many times during the processing of a single piece of text
  - Any attempt at a precise explanation of an LLM's behavior is doomed to be too complex for any human to understand
- Often, ad-hoc techniques that at first seem to provide insight into the behavior of an LLM are later found to be severely misleading (Feng et al., 2018;Jain & Wallace, 2019; Bolukbasi et al., 2021; Wang et al.,2022)
- In addition, promising-looking techniques that elicit reasoning in natural language do not reliably correspond to the processes that LLMs use to reason, and model-generated explanations can also be systematically misleading

Summary

- For NLP researcher, it is currently probably the most interesting time ever
- Very fast progress in LLMs
- Totally unexpected results
- What will happen in the future?
- Artificial General Intelligence?