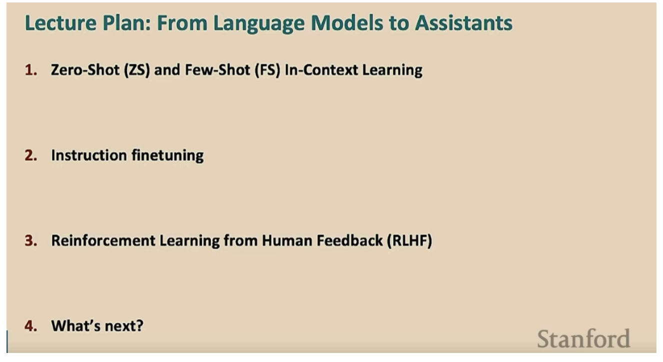
Notes

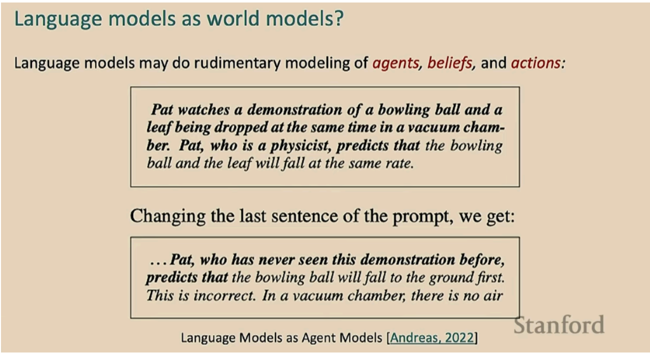
03 June 2024 08:28

Lecture Outline:



Recap and Motivation:

- Model and Data scales are getting bigger by each passing day.
- Pre-training helps these models to learn in-context (I.e. without explicit input + gradient step for a specific task)
- LLM as World models: These LLMs seem to have gathered basic knowledge of the world through the text they have seen. E.g. an LLM with basic info and a prompt predicts behavior of a ball in vacuum.



- Other examples of LLM capabilities – Coding, Drug discovery...
- LLMs as multi-task agents?

Zero shot and Few shot in-context learning:

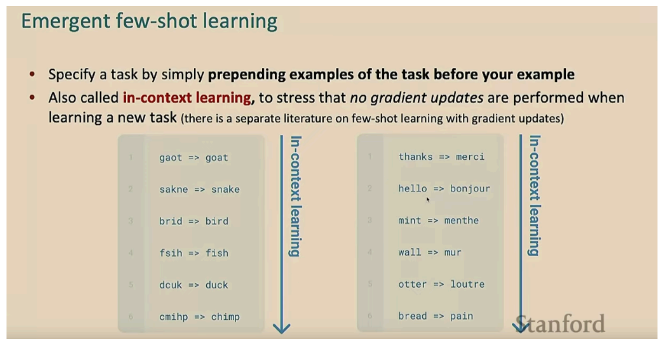
Year	Model size	Data size	scores
2018	117M	4GB books	
2019	1.5B	40GB internet	
2020	175B	600GB	

Emergent abilities of GPTs:

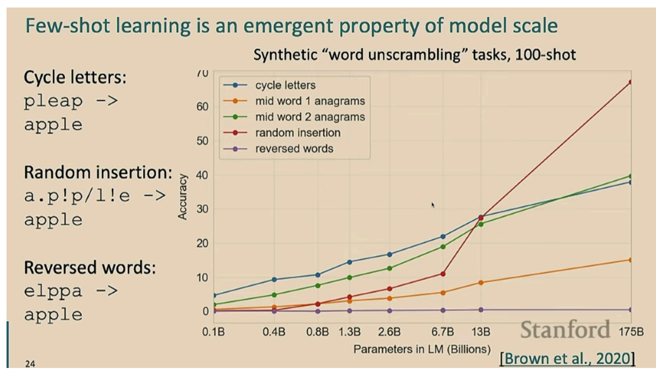
- GPT2 Paper - Language models are unsupervised multi-task learners. Radford et al., 2019

- GPT-2 showed zero-shot learning for many tasks e.g.
    - Language modeling
    - Translation
    - Sentiment analysis
    - Text summarization

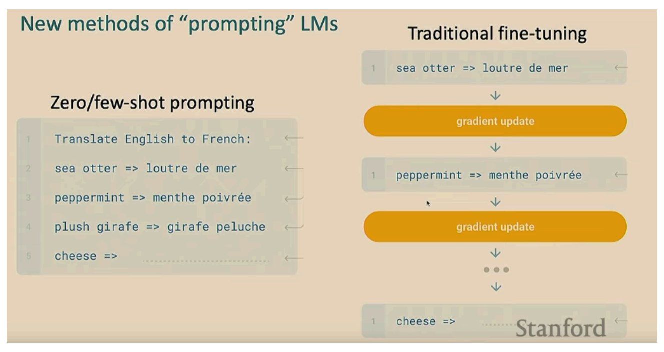
- GPT3 – showed strong few shot/in-context learning.
- GPT3 Paper – Language models are few shot learners. Brown et al., 2020



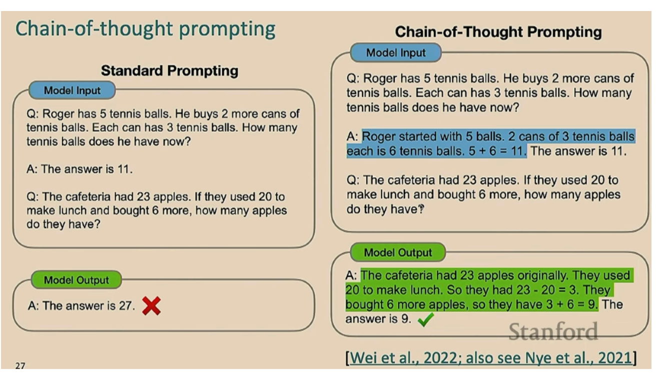
- Few-shot learning is "unlocked" with higher model scale...

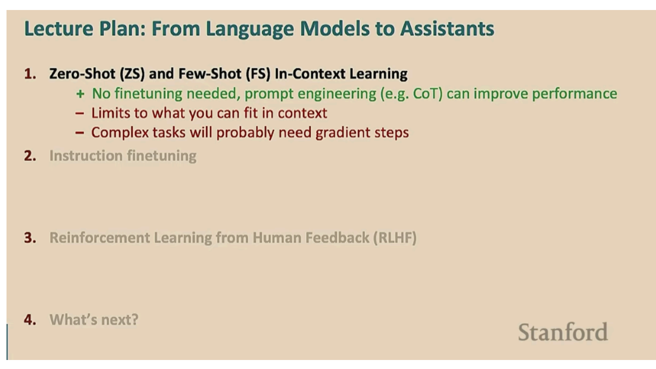


- Fine tuning vs few-shot prompting example of English to French translation:
    - In few-shot scenario – model is frozen and some examples of how to do this task are given in prompt..



- Limitations of few-shot/prompting:
    - Some tasks are too difficult, e.g. math
        - Chain of thought prompting can help:

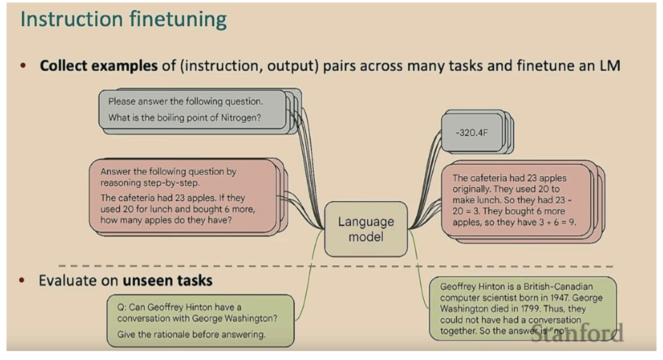




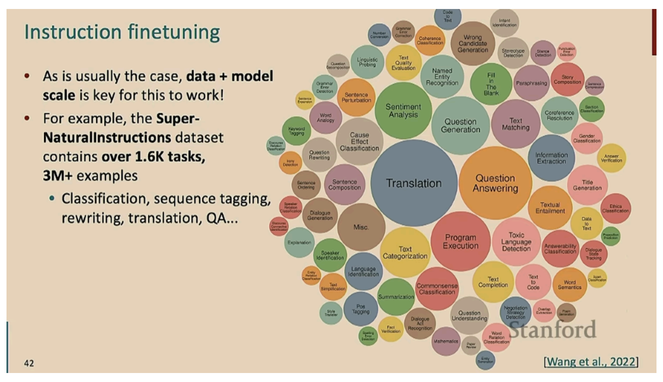
Instruction Finetuning:

Why finetuning?

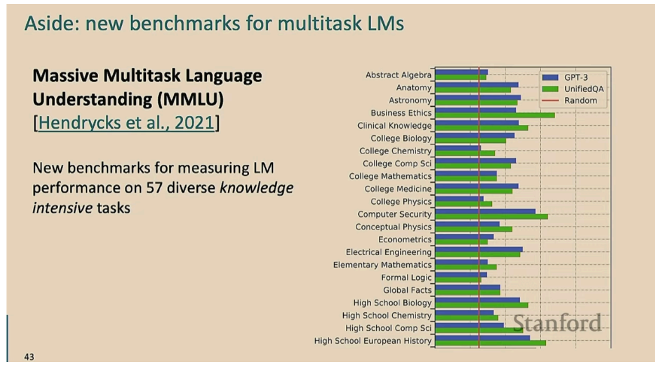
- LLMs are pre-trained to predict the next word. They are not specifically made for assisting the user in a specific context. They are not aligned with user intent.
- With fine tuning, we can teach the model to deal in specific context that will be encountered in real world.



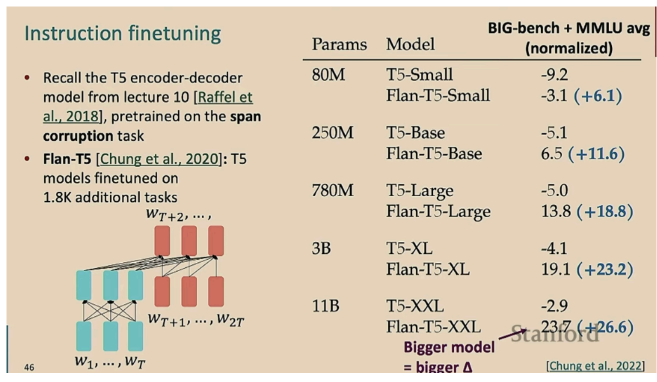
- e.g. dataset for finetuning -- super natural instructions



- How to evaluate the fine-tuned models??
  - e.g. Massive Multi Task Language Understanding (MMLU) Benchmark

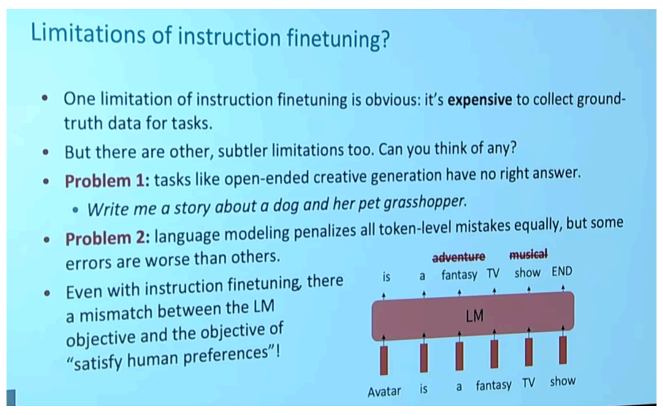


- Does instruction finetuning works?
  - It does. Key: bigger the model, bigger the benefit of fine tuning



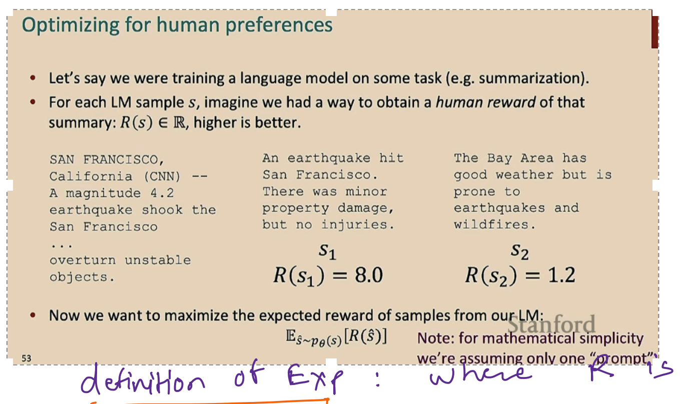
Params	Model	BIG-bench + MMLU avg (normalized)
80M	T5-Small	-9.2
	Flan-T5-Small	-3.1 (+6.1)
250M	T5-Base	-5.1
	Flan-T5-Base	6.5 (+11.6)
780M	T5-Large	-5.0
	Flan-T5-Large	13.8 (+18.8)
3B	T5-XL	-4.1
	Flan-T5-XL	19.1 (+23.2)
11B	T5-XXL	-2.9
	Flan-T5-XXL	23.7 (+26.6)

- Limitations of fine-tuning:



Reinforcement Learning from Human Feedback (RLHF):

- How do you accommodate user's preference without fine tuning?



definition of Exp : where R is reward function.

- Policy Gradient/REINFORCE -- the reinforcement learning algorithm behind RLHF

$$\nabla_\theta E_{\hat{s} \sim P_\theta(s)} [R(\hat{s})] = \nabla_\theta \boxed{\sum_s} R(s) \cdot \boxed{P_\theta(s)}$$

due to linearity of the gradient, we can

$$\nabla_\theta E_{\hat{s} \sim P_\theta(s)} [R(\hat{s})] = \sum_s R(s) \nabla_\theta P_\theta(s) \rightarrow \text{I}$$

using log derivative trick, i.e

$$\nabla_\theta \log P_\theta(s) = \frac{1}{P_\theta(s)} \nabla_\theta P_\theta(s) \Rightarrow \nabla_\theta P_\theta(s) = \nabla_\theta \log P_\theta(s) \, P_\theta(s).$$

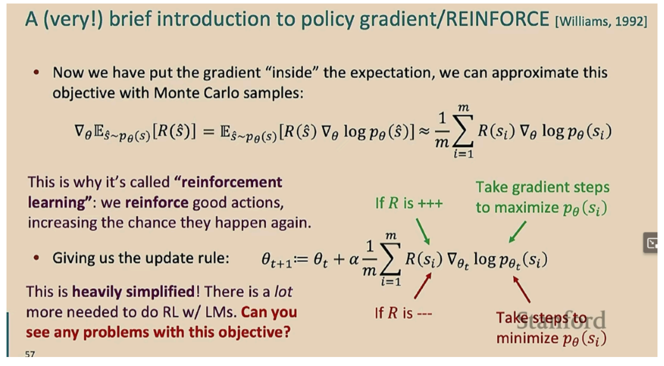
Plug back $\nabla_\theta P_\theta(s)$ in I

$$\sum_s R(s) \nabla_\theta P_\theta(s) = \boxed{\sum_s P_\theta(s)} \underline{R(s) \nabla_\theta \log P_\theta(s)}$$

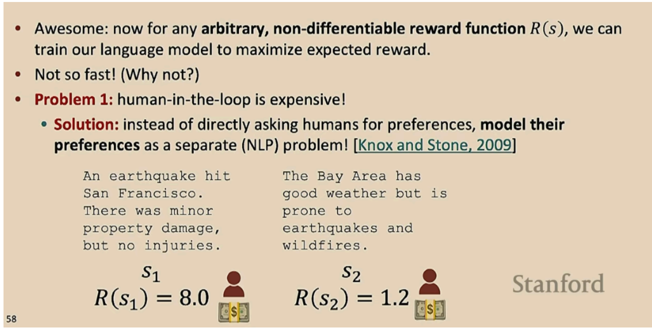
This is expectation of

$$\sum_s R(s) \nabla_\theta P_\theta(s) = E_{\hat{s} \sim P_\theta(s)} [R(\hat{s}) \nabla_\theta \log P_\theta(s)]$$

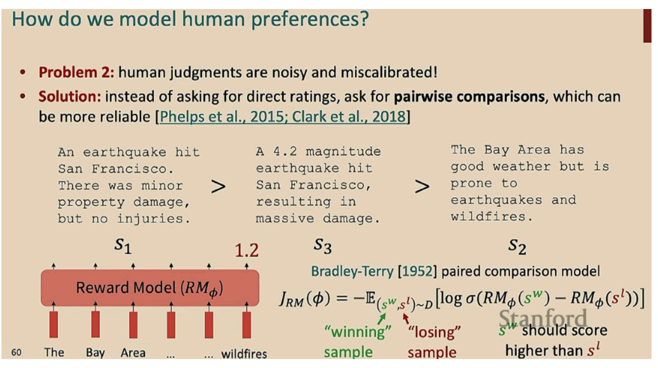
- In summary:



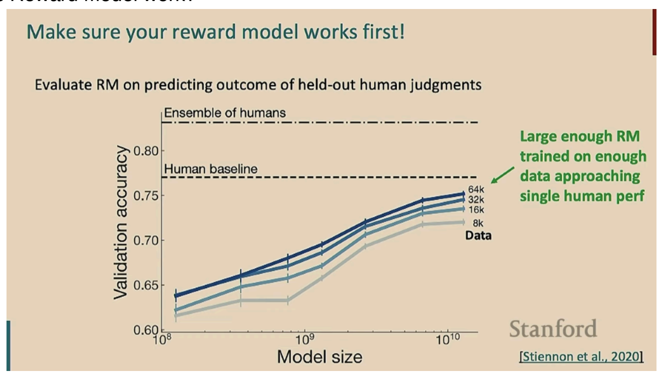
- Problems with RLHF:
    - Human feedback is expensive --- so why not train a model to predict reward?



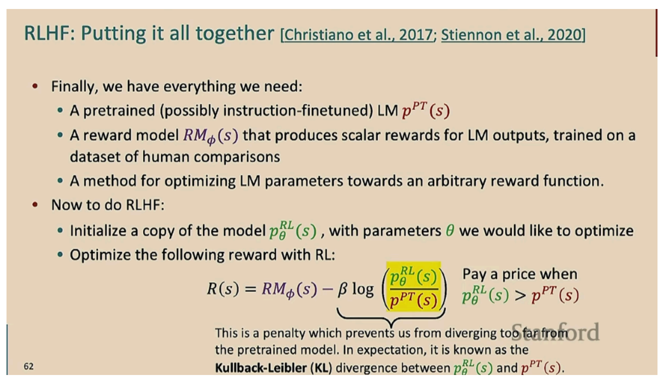
- Human judgement is not consistent --- use relative scores



- Does Reward Model work?



- RLHF Summary:



What's Next?