fun-ai-talk

A Primer on Large Language Models (LLM)

github.com/hululuzhu/llm-primer

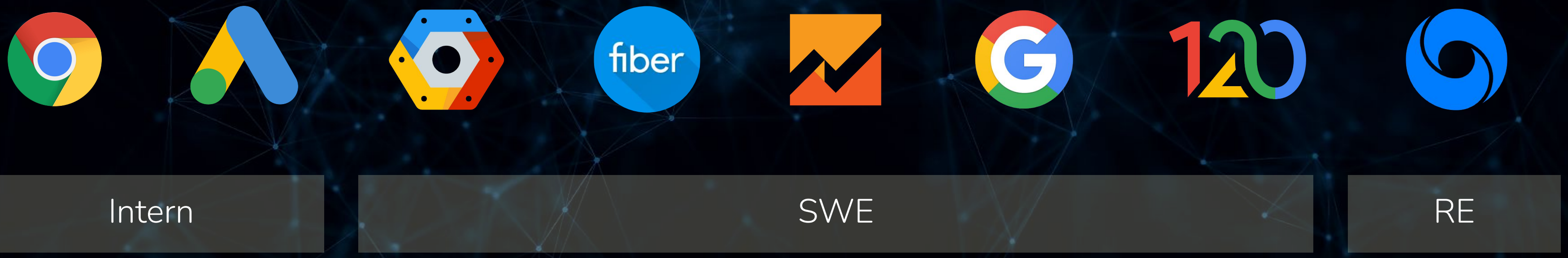
Last update: Jan 2023
hululu.zhu@gmail.com

Disclaimer

- All content in this deck is based on public papers, shared codes/models, blog articles, social media discussions, and demos

- All opinions in this slide deck are of my personal own (*hululu.zhu@gmail.com*), and not those of DeepMind®, Google®, or Alphabet®

Me: 10+ yoe, various roles @ Alphabet®



Intern

SWE

RE

Agenda - LLM Primer

- Intro: Building blocks & capabilities [10 mins]
- Core: Models, players, concepts, toolings & applications [40 mins]
- Break [3 mins]
- Bonus: Deep dive into ChatGPT [20 mins]
- Q&A

- *No/Little coverage*
  - *Multilingual, Multimodal, Bias, Ethics, Safety, Serving, Carbon Emission, AGI*

Part 1/3: LLM Intro

Language Models (LM) and Large Language Models (LLM)

LM for understanding (*e.g. BERT*)

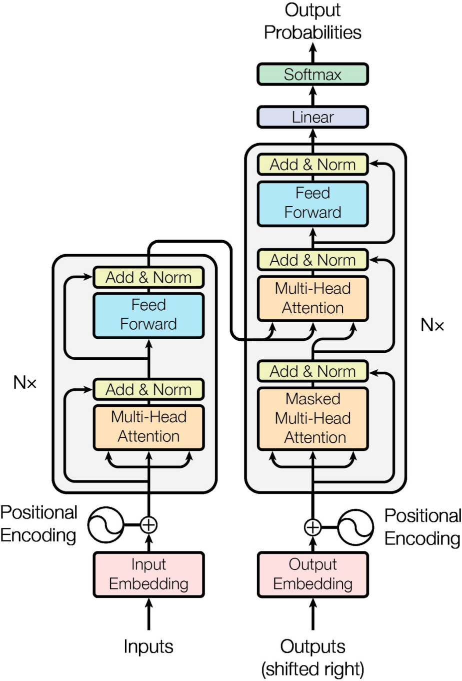
- Text in
- Embedding (numeric representation of understanding) out
    - The Embedding can be connected to other output heads for tasks like classification or regression

LM for generation (*e.g. GPT or T5*)

- Text in
- Text out

* In most cases, **LLM** refers to **huge** (e.g. >1B params) Deep Learning LM for **generation**

LLM building blocks: Deep Learning and Transformer



LLM is on top of Deep Learning

Transformer as dominating architecture for NLP since 2018

- Multi-head attention
- Encoder-Decoder
- Embedding layers
- Positional encoding
- Cross-Attention in decoder layers
- Output Softmax

Note: Tokenization (e.g. wordpiece, sentencePiece, BPE) is needed (outside Transformer) to convert text to token ids

Note: Sometimes we call it XFormer since there are many variations to the original Transformer

Attention Is All You Need (Transformer paper) by Google | The Annotated Transformer by Harvard NLP | A Survey of Transformers by Fudan

LLM Intro: How are LLMs trained [initially]?

- Fill the blanks (aka masks) for “Masked Language Models”
  - **Ground Truth:** “Paris is a beautiful city”
    - **X:** “Paris is a **[MASK]** city”
    - **Y:** “beautiful”
    - **Model:** “good”
    - **Optimize:** “good”👎 “beautiful”👍

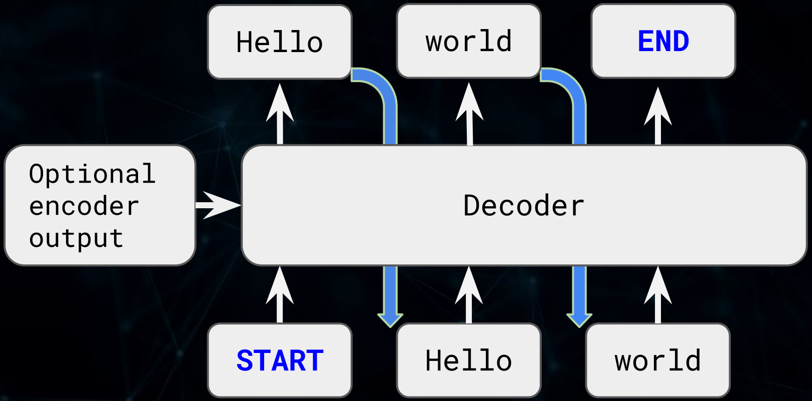
- Predict the remaining text given prompt (on the left), for “Generative Language Models”
  - **Ground Truth:** “Paris is a **|** beautiful city”
    - **X:** “Paris is a”
    - **Y:** “beautiful”
    - **Model:** “good”
    - **Optimize:** “good”👎 “beautiful”👍
    - **X:** “Paris is a beautiful”
    - **Y:** “city”
    - **Model:** “place”
    - **Optimize:** “place”👎 “city”👍

- The “Self-supervised” Learning Paradigm
  - It is supervised (given x, predict y)
  - It does NOT require expensive human labels (more precisely, this statement is only true for pre-training)

Decoding/Generating Algorithms in Generative LLMs

Decode **token** by **token**, left to right. A new output token is appended as next token's decoder input

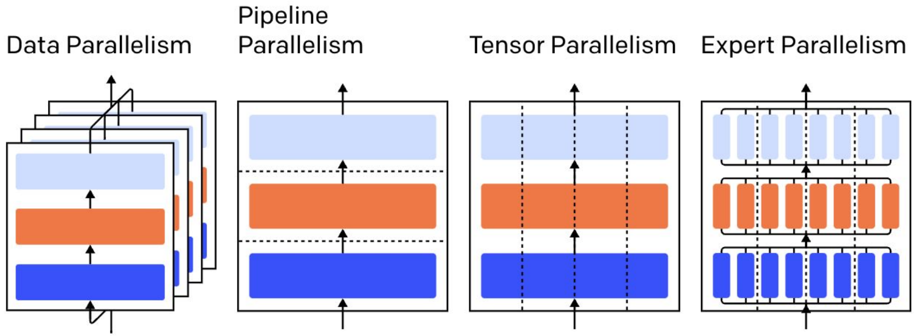
- Beam Search
  - Maintain a max size of searching "beams (paths)" to get best overall best beam
- Sampling
  - Sampling based on probabilities
- Greedy
  - Select the argmax(prob) token at every position
- Top-k, Top-p and more



https://huggingface.co/blog/how-to-generate

How to train LLM (in parallel)?

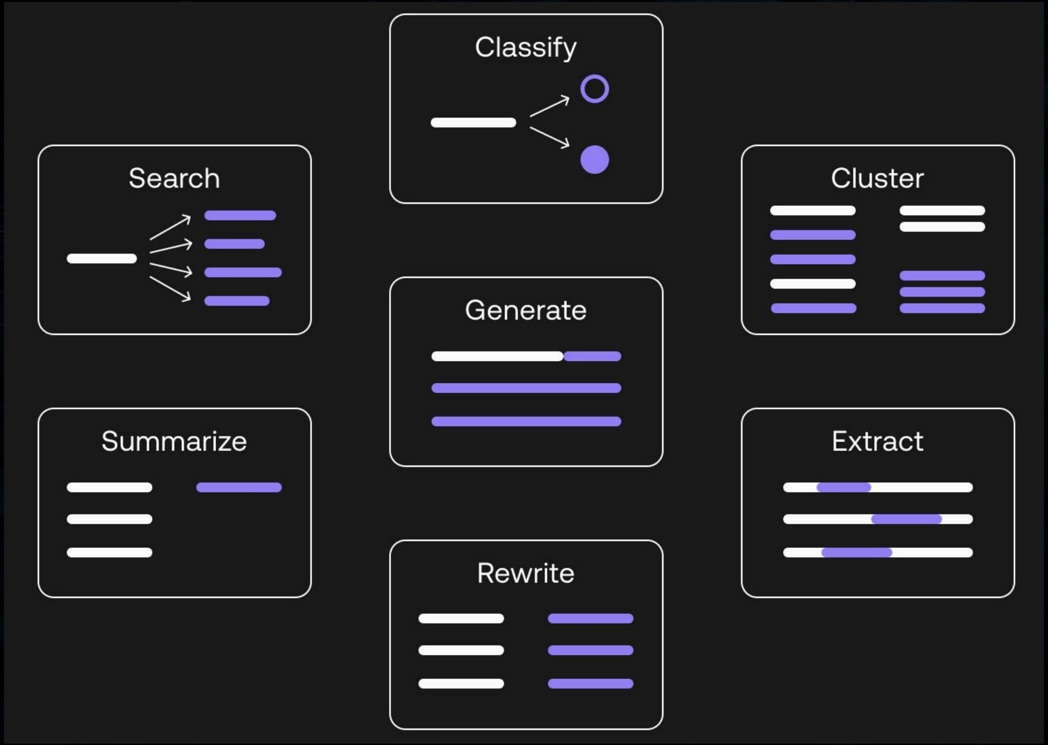
- Data Parallelism
  - different subsets of the batch on different GPU/TPUs
- Pipeline parallelism
  - different layers of the model on different TPU/GPUs
- Tensor Parallelism
  - Break up tensor operation (e.g. matrix multiplication) to different TPU/GPUs
- Mixture of Experts (sparse)
  - Gated layer to only activate factions (one of few of all the experts) of the model



An illustration of various parallelism strategies on a three-layer model. Each color refers to one layer and dashed lines separate different GPUs.

https://openai.com/blog/techniques-for-training-large-neural-networks/

LLM capabilities: High-level tasks that LLMs can do



Large Language Models and Where to Use Them @ Cohere.ai

LLM capabilities: Some advanced tasks

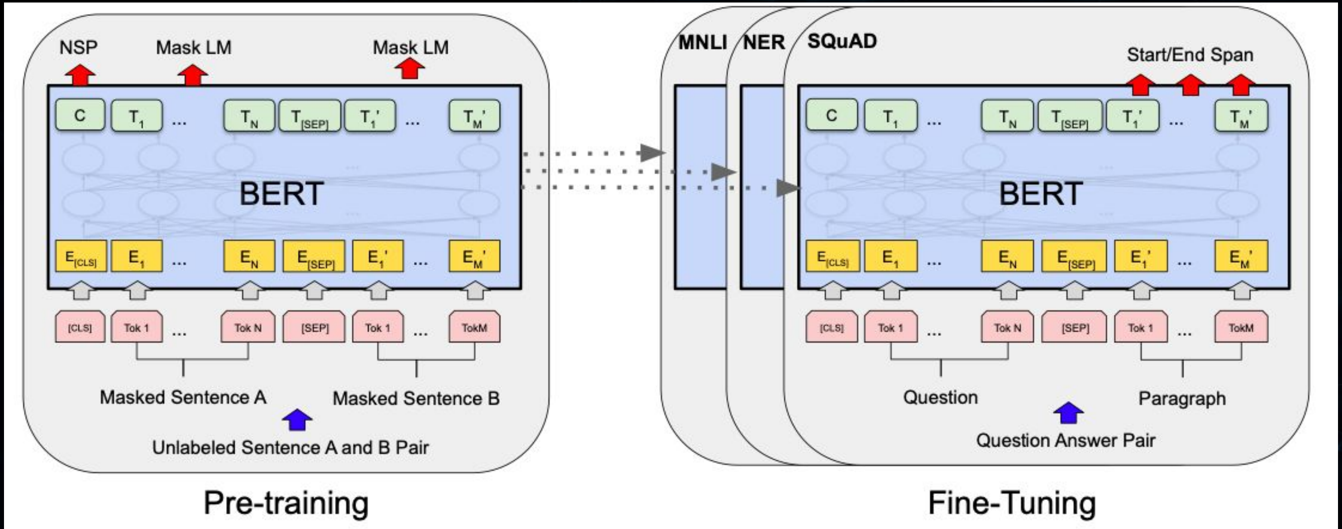
- Write code, GitHub's AI Coding Assistant Copilot Launches - Voicebot.ai
- Writing a journal paper, Researcher Tells AI to Write a Paper About Itself, Then Submits It to Academic Journal
- “Sentient” Dialog conversation, Google Sidelines Engineer Who Claims Its AI (Google LaMDA) Is Sentient - The New York Times
- Quantitative reasoning, Google AI Blog: Minerva: Solving Quantitative Reasoning Problems with Language Models
- Explaining a joke, Google's Massive New Language Model Can Explain Jokes

LLM capabilities: Even more challenging tasks

- Write competitive code, DeepMind's AlphaCode AI writes code at a competitive level | TechCrunch
- Write better code with reinforcement learning, Salesforce's CodeRL Achieves SOTA Code Generation Results With Strong Zero-Shot Transfer Capabilities | Synced
- Solve college level Math/Physics/Chemistry/Economics problems, see Google AI Introduces Minerva: A Natural Language Processing (NLP) Model That Solves Mathematical Questions
- Solve Math Olympiad Problems, OpenAI: Solving (Some) Formal Math Olympiad Problems
- GPT-F by OpenAI, automated prover and proof assistant for the Metamath formalization language

Part 2/3: LLM Core

LLM example: BERT (encoder-only LLM)



Pretraining:

- Masked language training
- Next sentence prediction (NSP)

Fine-tuning:

- Connect to BERT output and work for many tasks

[1810.04805] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

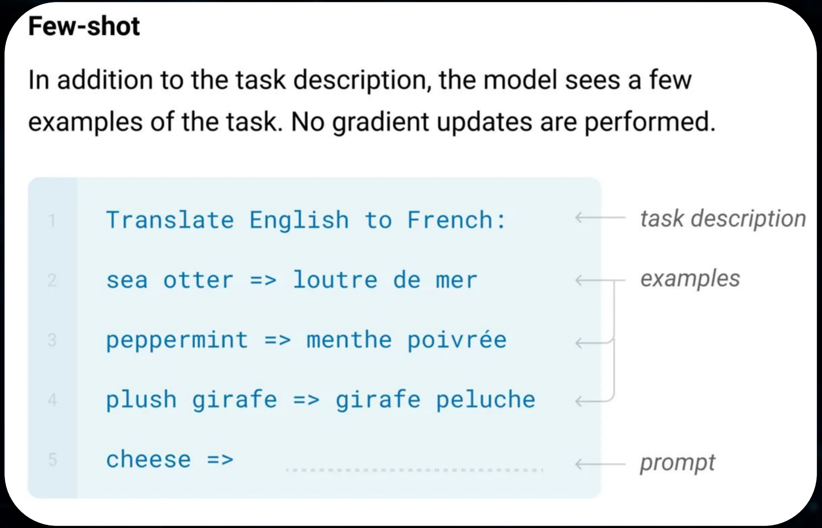
LLM example: GPT (decoder-only LLM)

GPT often refers to a family of models (GPT, GPT2, GPT3…)

First influential decoder-only models

GPT creates the “Few/Zero shot Prompt”

OpenAI started to “un-share” models since GPT2



Few-shot

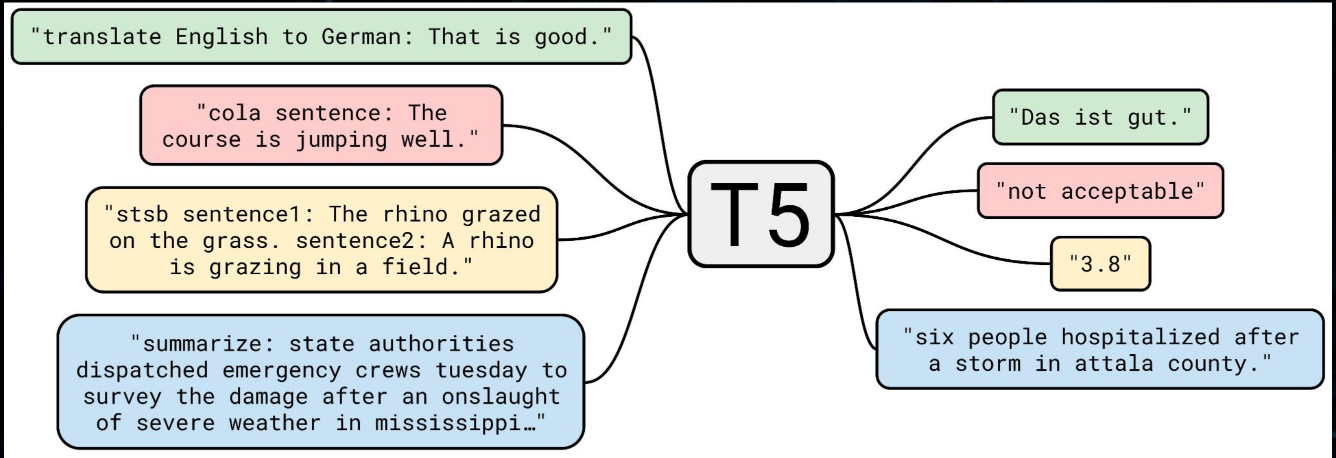
In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

```
Translate English to French:
sea otter => loutre de mer
peppermint => menthe poivrée
plush girafe => girafe peluche
cheese =>
```

task description
examples
prompt

[Improving Language Understanding by Generative Pre-Training](), [OpenAI says its text-generating algorithm GPT-2 is too dangerous to release](), [What is GPT-3? Everything your business needs to know about OpenAI’s breakthrough AI language program | ZDNET]()

LLM example: T5 (encoder-decoder LLM)


"translate English to German: That is good."
"cola sentence: The course is jumping well."
"stsb sentence1: The rhino grazed on the grass. sentence2: A rhino is grazing in a field."
"summarize: state authorities dispatched emergency crews tuesday to survey the damage after an onslaught of severe weather in mississippi…"
T5
"Das ist gut."
"not acceptable"
"3.8"
"six people hospitalized after a storm in attala county."

T5: unified framework that converts all text-based language problems into a **text-to-text** format

- T5 works well on a variety of tasks out-of-the-box with “prompts”

[1910.10683] Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

LLM Example: GLM 130B (public for research, Chinese+English)

Hardware	GPU Memory	Quantization	Weight Offload
8 * A100	40 GB	No	No
8 * V100	32 GB	No	Yes (BMInf)
8 * V100	32 GB	INT8	No
8 * RTX 3090	24 GB	INT8	No
4 * RTX 3090	24 GB	INT4	No
8 * RTX 2080 Ti	11 GB	INT4	No

👍

GLM-130B: An Open Bilingual Pre-Trained Model

LLM Players: OpenAI and selected work

- GPT-1 2018
- GPT-2, 2019
  - OpenAI: Too dangerous to share, How OpenAI Sold its Soul for $1 Billion
- GPT-3, 2020
  - 175B parameters! 100x larger
- Codex (powers github copilot), 2021
  - Text+Code pretrain
- GPT-3.5, Q4 2021
  - Instruction finetune
- InstructGPT, Q1 2022
  - RLHF (Reinforcement Learning Human Feedback)
- ChatGPT, Dec 2022
  - Product launch

LLM Players: Google/DeepMind and selected work

- BERT, 2018
  - Completely changed the NLP research and industry
- T5, 2020
  - Consolidate all NLP task to text-to-text
- FLAN, 2021
  - Instruction Fine-Tuning (probably inspires GPT3.5)
- LaMDA, 2021
  - LaMDA and the Sentient AI Trap | WIRED
- Chinchilla, 2022
  - "Most LLMs are under-trained!"
- PaLM, 2022
  - 540B params, 3x GPT3
- Sparrow,2022
  - Reinforcement-learning LLM, only paper, no public product

LLM Players: Facebook (aka Meta) and selected models

- RoBERTa, 2019
  - A more popular version of enhanced BERT for the industry
- BART, 2020
  - Pretraining sequence-to-sequence models
- OPT-175B, 2022
  - “Democratizing access to large-scale language models”
- BlenderBot3, 2022
  - Probably largest chatbot-specific LM
- Galactica (research purpose LLM), 2022
  - Taken down after 3 days after many harsh criticism like this

LLM Players: Other multinational companies

- Megatron-Turing by NVidia and Microsoft
  - 530B params on 2240 NVIDIA A100 GPUs
- CodeT5 and CodeRL by Salesforce
  - [probably] the most popular coding-assist base models
- *[Most likely missing many great work from other organizations, sorry]*

LLM Players: Large Chinese companies

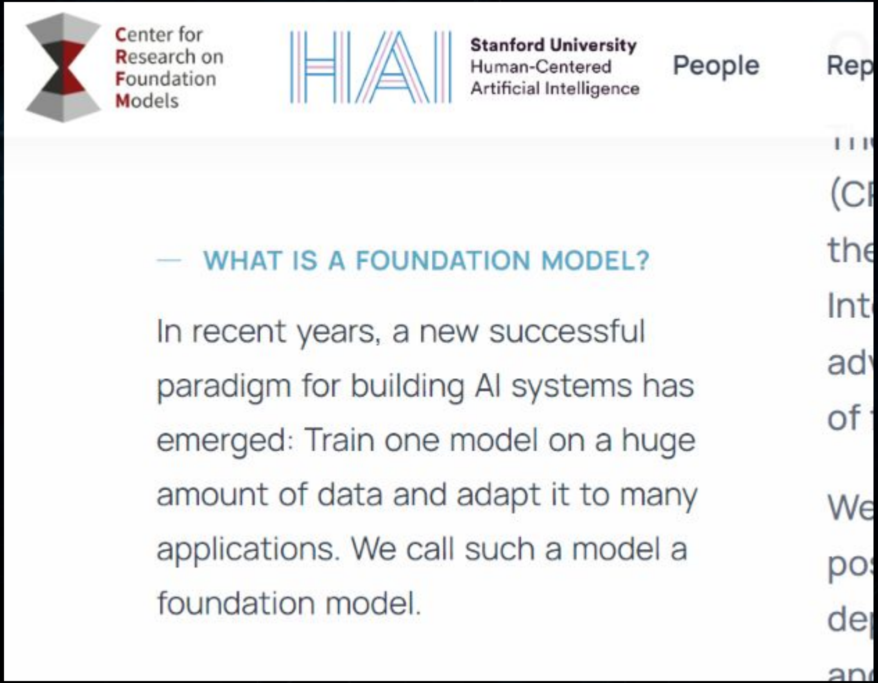
- Pangu-alpha by Huawei, 2021
  - 200B params, *[I read from articles mentioning it was believed to be under-trained, lack reference]*
- Wudao 2.0 by BAAI, 2021
  - A ***sparse*** *(thus less powerful IMO)* multimodal model with 1.75 Trillion params
- Ernie 3.0 Titan by Baidu, 2021
  - 260B params, on top of PaddlePaddle (Baidu Deep learning framework), most likely the best Chinese LLM
- M6 by Alibaba & Tsinghua, 2021
  - 100B, Later 2021 a sparse version with 1 trillion+ params

LLM Players: Selected Institutes, Groups & Startups

- Allen Institute for AI (AI2)
- Tsinghua University (GLM 130B 2022 public)
- BigScience research workshop (bloom, 176B, 2022 public)
- Eleuther AI (GPT-neox 20B public, 2021)
- Anthropic (founders wrote the Transformer paper, their RLHF LLM paper, 2022)
- Zhuiyi Technology (Su Jianlin and RoFormer 2021)

LLM Concepts: Foundational Models

- Brought up by Researchers @ Stanford HAI
- LLM is one of the foundational models



WHAT IS A FOUNDATION MODEL?

In recent years, a new successful paradigm for building AI systems has emerged: Train one model on a huge amount of data and adapt it to many applications. We call such a model a foundation model.

LLM Concepts: Benchmarks

- GLUE: A Multi-Task Benchmark for NLP, 2018
  - Leaderboard: https://gluebenchmark.com/leaderboard/
- SuperGLUE, 2019
- BIG-bench, 2020
- GSM8K (math), 2021
- And more…

LLM Concepts: Pretraining, Finetuning & Prompt-Tuning

- Pretraining
  - Self-supervised training with Masked Language Prediction or Next Token Prediction objectives
- Finetuning
  - Take a pretraining model into a downstream use case
  - The parameters of pretraining model will often change
- Prompt-Tuning
  - The pretrained generative model will keep unchanged
  - The prompt (input to the model) will be tuned/engineered

* The whole concepts here also applied to vision domains recently, e.g. MAE work

LLM Concepts: Scaling Laws for LLM

OpenAI, 2020

- If you have 10x more budget, ~5x model size, ~2x data size

DeepMind, 2022

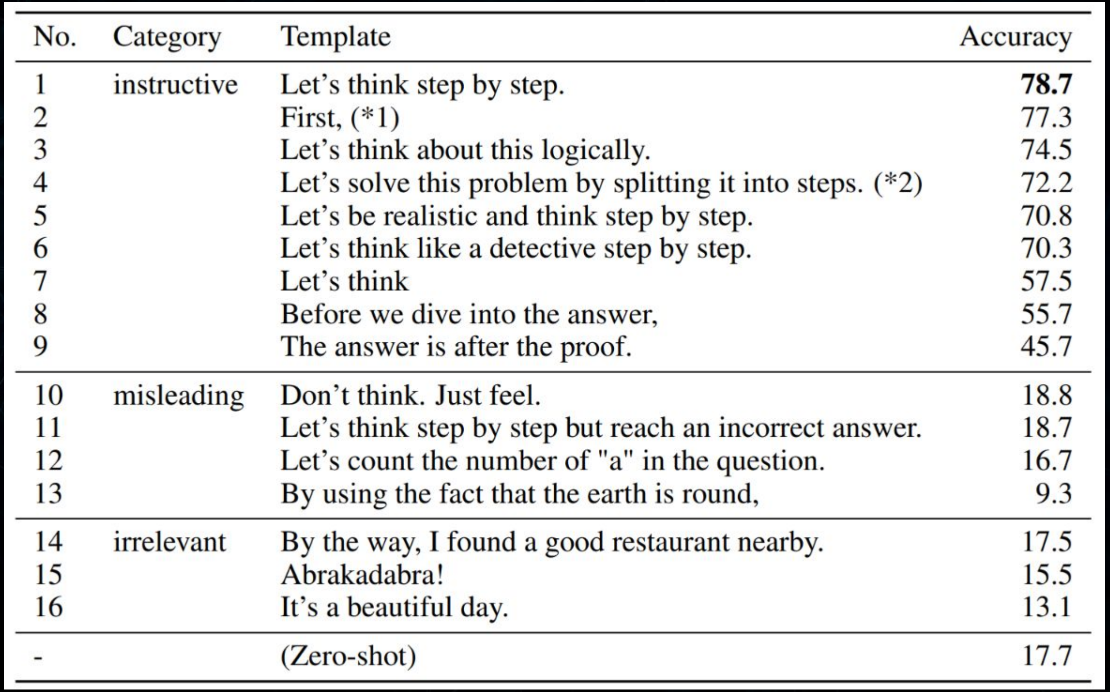
- If you have 10x more budget, ~3x model size, ~3x data size
- “Most LLMs are under-trained” *[because oversized model, while lacking training data]*
- But, *karpathy: I can't exactly reproduce Chinchilla paper results*

LLM Concepts: Prompt Engineering

"***Let's think step by step***" to increase accuracy from 17.7% to 78.7!

- See GPT3 paper for more details

More prompt engineering guide, see openai-cookbook/techniques_to_improve_reliability

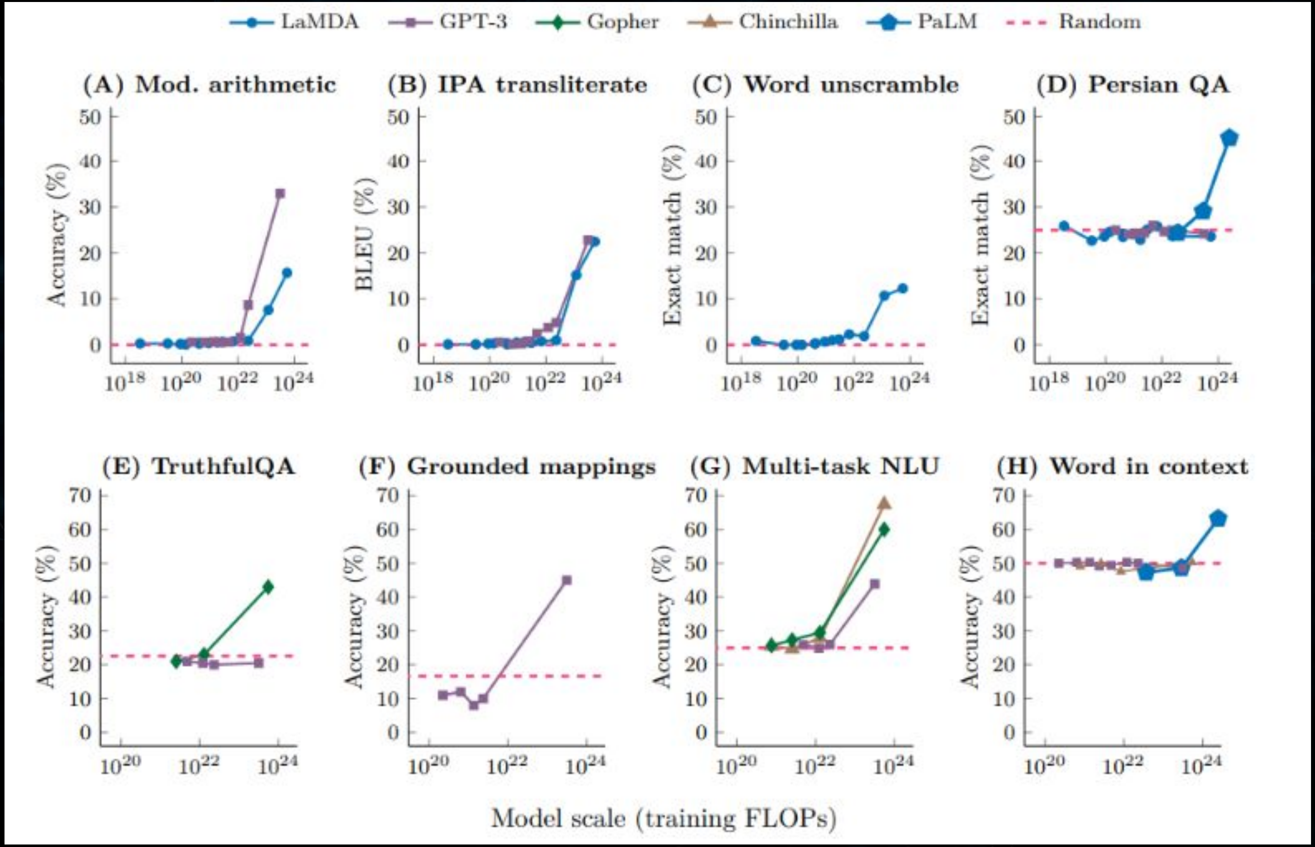


No.	Category	Template	Accuracy
1	instructive	Let's think step by step.	**78.7**
2		First, (*1)	77.3
3		Let's think about this logically.	74.5
4		Let's solve this problem by splitting it into steps. (*2)	72.2
5		Let's be realistic and think step by step.	70.8
6		Let's think like a detective step by step.	70.3
7		Let's think	57.5
8		Before we dive into the answer,	55.7
9		The answer is after the proof.	45.7
10	misleading	Don't think. Just feel.	18.8
11		Let's think step by step but reach an incorrect answer.	18.7
12		Let's count the number of "a" in the question.	16.7
13		By using the fact that the earth is round,	9.3
14	irrelevant	By the way, I found a good restaurant nearby.	17.5
15		Abrakadabra!	15.5
16		It's a beautiful day.	13.1
-		(Zero-shot)	17.7

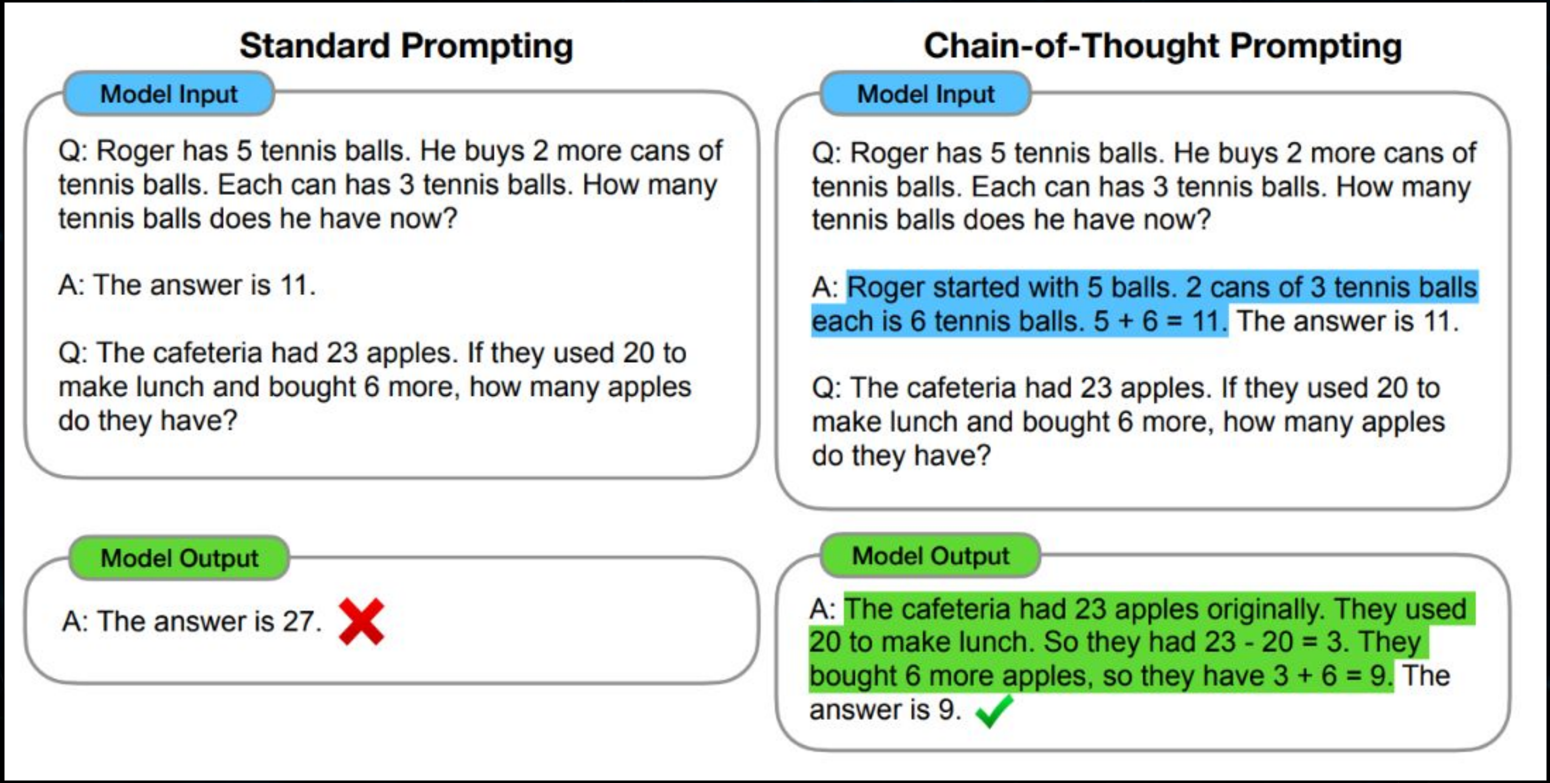
LLM Concepts: “Emerging abilities”

See paper for more details

- Note X axis unit is floating-point operations per second (FLOPS)



LLM Concepts: Chain of Thoughts (paper)



Standard Prompting

Model Input

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: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. ✗

Chain-of-Thought Prompting

Model Input

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.

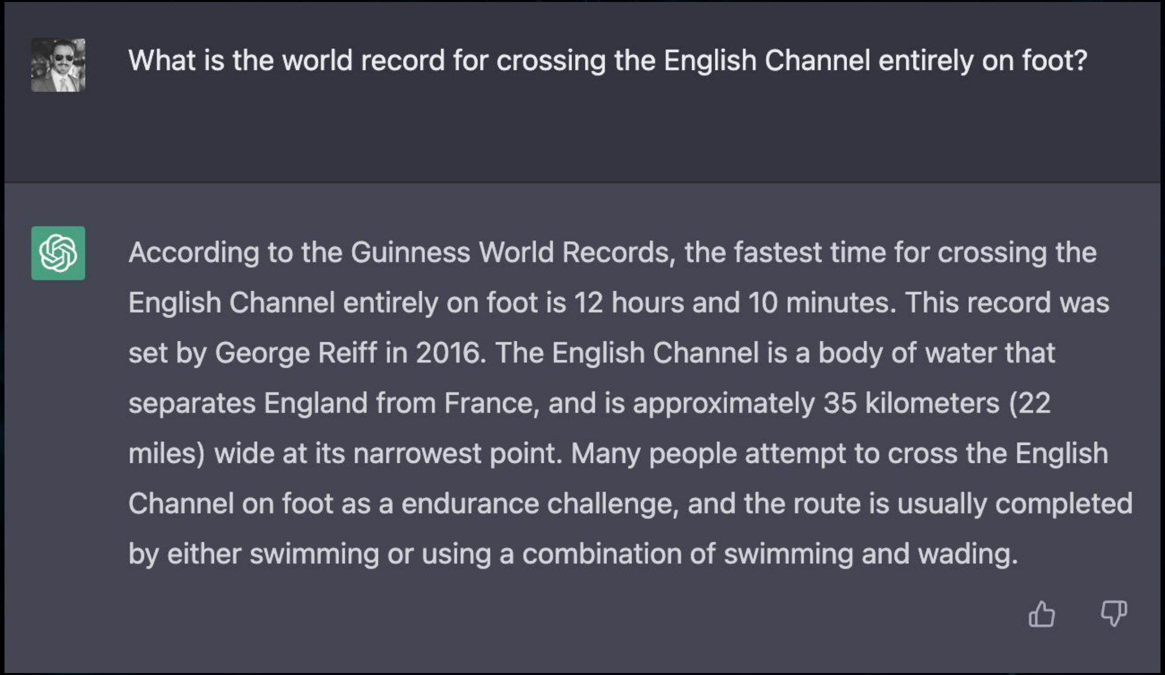
Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had 23 - 20 = 3. They bought 6 more apples, so they have 3 + 6 = 9. The answer is 9. ✓

LLM Concepts: Hallucination (part of safety concerns)

"What is the world record for crossing the English Channel entirely on foot?" from Twitter



What is the world record for crossing the English Channel entirely on foot?

According to the Guinness World Records, the fastest time for crossing the English Channel entirely on foot is 12 hours and 10 minutes. This record was set by George Reiff in 2016. The English Channel is a body of water that separates England from France, and is approximately 35 kilometers (22 miles) wide at its narrowest point. Many people attempt to cross the English Channel on foot as a endurance challenge, and the route is usually completed by either swimming or using a combination of swimming and wading.

LLM Concepts: Retrieval-system powered LLM

- RETRO by DeepMind, 2021
- LaMDA 2022 paper by Google, 2022
- Atlas by Facebook, 2022

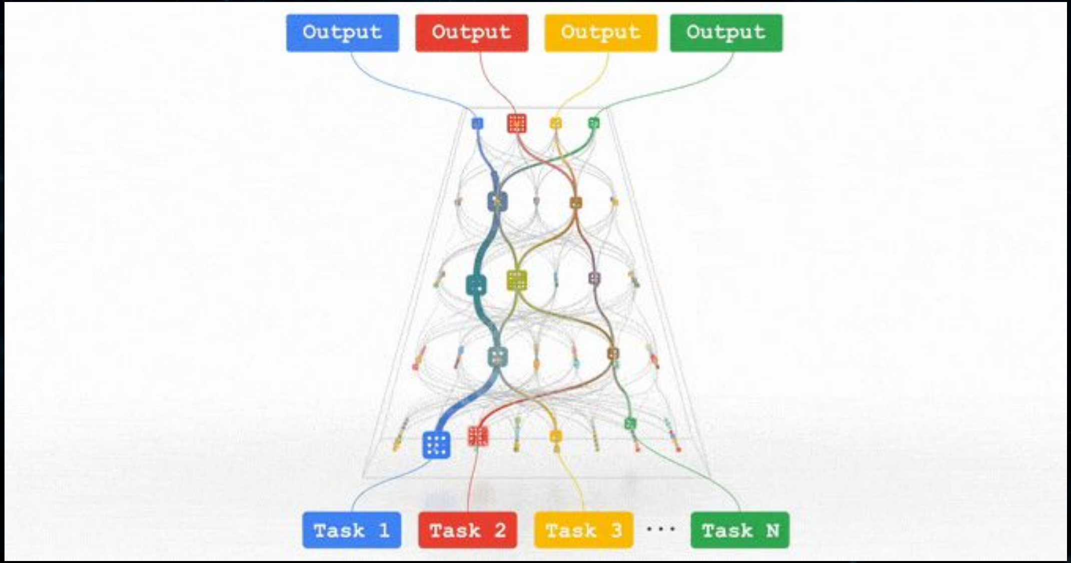
** personally, I think LLM+Retrieval System is the good way to balance LLM capabilities and data freshness to meet business purposes in the short term*

LLM Concepts: RLHF for LLM

- RL (reinforcement learning) was something popular back to 2016 when AlphaGo is made the news
- OpenAI has some RLHF research back in 2017
- InstructGPT (paper published 2022) is believed to let GPT3 be more powerful
- Other players
  - Anthropic
  - Google
  - DeepMind
  - More

LLM Concepts: MoE LLM

- MoE concept by Google, 2017
- Jeff Dean Pathways, 2021
  - “Today's models are dense and inefficient. Pathways will make them sparse and efficient.” 🤔



LLM Tooling: HuggingFace

- Best LLM tools and model hub, period [my favorite!]
- Easy
  - To load models
  - To tokenize
  - To start out of the box with Pipelines
  - To tune with examples
  - To publish and deploy
- My pet projects
  - Chinese poem model https://huggingface.co/hululuzhu/chinese-poem-t5-mengzi-finetune
  - Solidity code model https://huggingface.co/hululuzhu/solidity-t5

** Do you know the super popular Stable Diffusion model is published and hosted at HuggingFace?*

LLM Tooling: TF Hub, PyTorch-NLP & PaddleNLP

TF Hub by Google (and community)

PyTorch-NLP by Meta (and community)

PaddleNLP by Baidu (and community)

Alibaba recently started ModelScope

** No one is ever close to HuggingFace as of Jan 2023, in my opinion*

LLM Tooling: Transformers, Colossal-AI, Ray & NanoGPT

Transformers Library (github 77k stars) by Huggingface

- Best of the best

Colossal-AI by Prof Yang You (who developed LAMB optimizer)

- Pretty promising open-source distributed AI training infra

Ray by anyscale

- Believed to be used to train ChatGPT

NanoGPT

- A tiny but cool library by Andrej Karpathy (I am his big fan!)

LLM Tooling: Other [more fine-grained] toolings

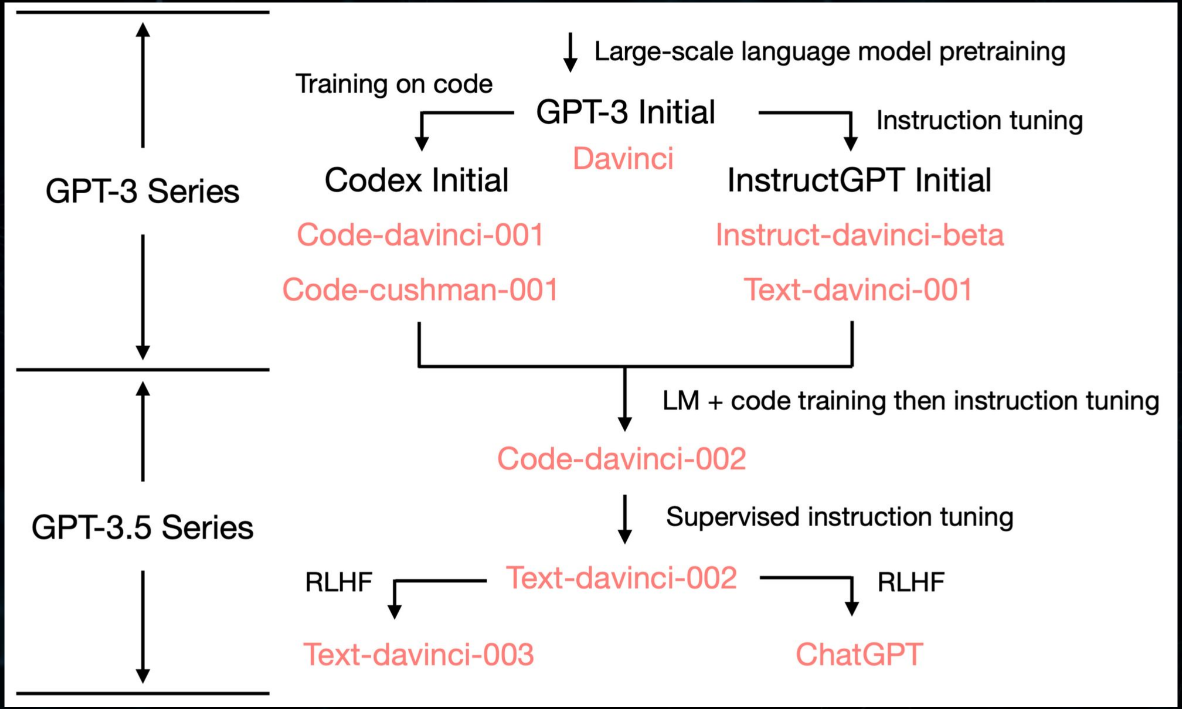
- TensorFlow
- PyTorch
- PaddlePaddle
- Keras
- PyTorch Lightning
- Jax/Haiku/Flax/Trax/T5X

LLM Applications:

- Search/Ranking, Recommendation
- Chatbot
- Spam detect
- Censorship
- Code assist
- And more

Part 3/3: ChatGPT

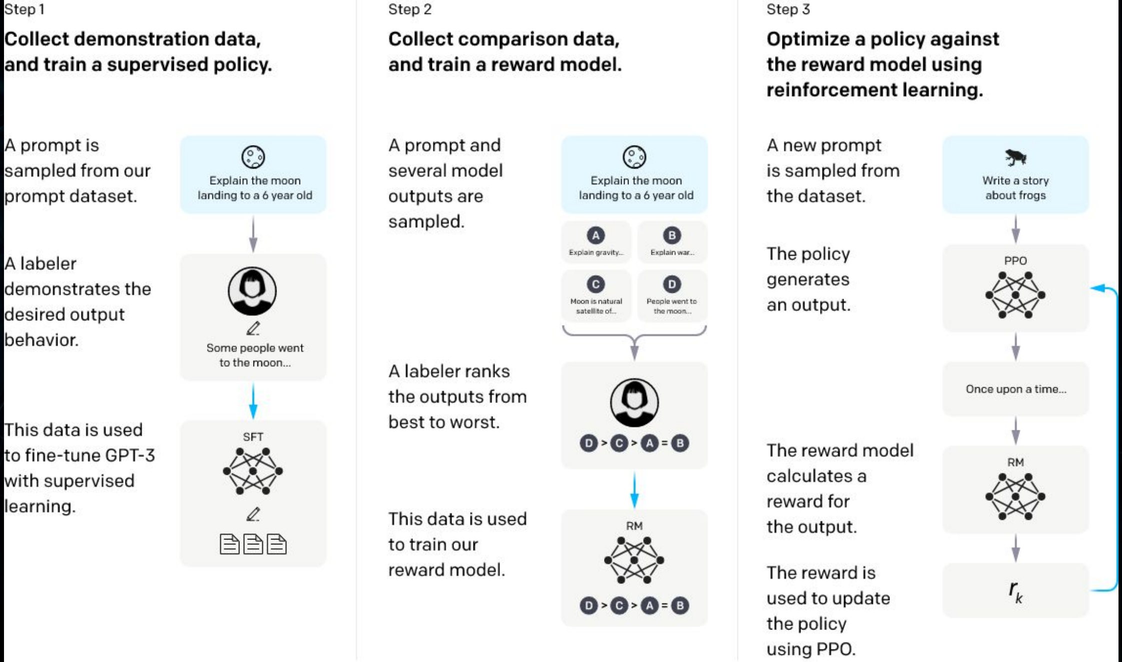
ChatGPT’s Model Evolving



Check out this great article by Yao Fu, yao.fu@ed.ac.uk, pic above is from this article too

ChatGPT Research overview

OpenAI chatGPT blog only mentioned “using the same methods as InstructGPT”, so assume InstructGPT is the research behind



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.

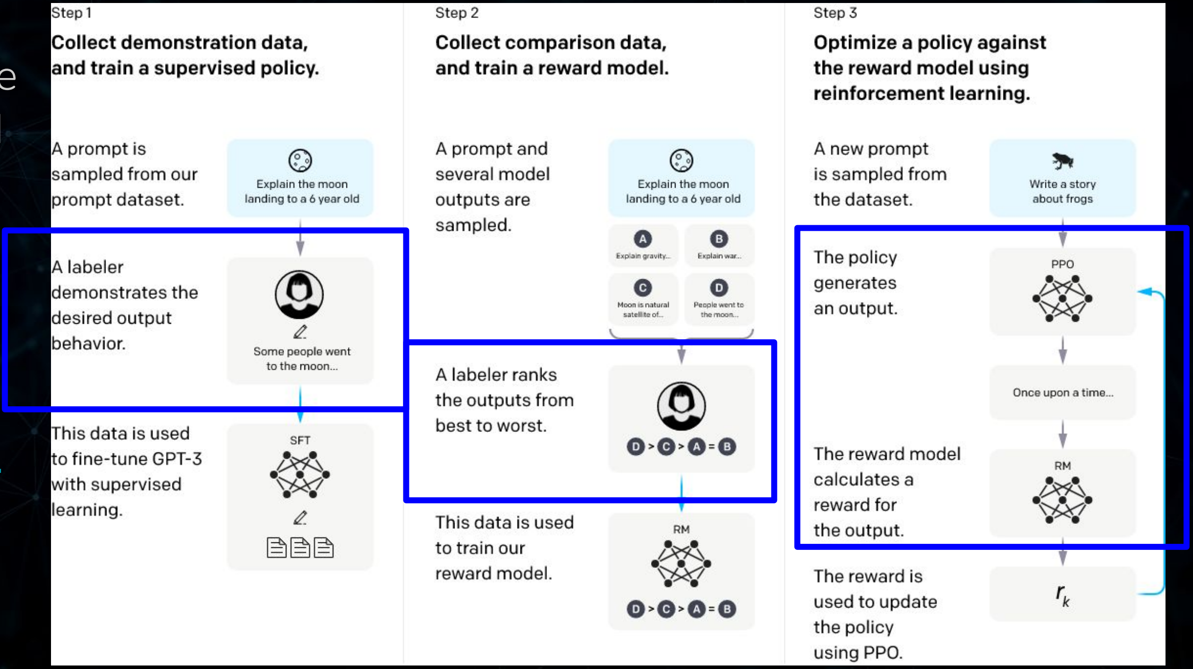
The reward is used to update the policy using PPO.

Explain the moon landing to a 6 year old
Some people went to the moon...
SFT
Explain the moon landing to a 6 year old
A
Explain gravity...
B
Explain war...
C
Moon is natural satellite of...
D
People went to the moon...
D > C > A = B
RM
D > C > A = B
Write a story about frogs
PPO
Once upon a time...
RM
r_k

Check out this nice video (Chat GPT (可能)是怎麼煉成的- GPT 社會化的過程) by Dr. Hung-yi Lee if you understand Chinese

Possible next steps for ChatGPT? (by Jonathan Hui)

- Step 1: Instead of pure human labeler, we may use reward model to pick good data for “semi-supervised” learning
- Step 2: Ranking stage can also be “semi-supervised”
- Step 3: Besides RL based self-play, we can use GAN (generative adversarial network)



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.

Explain the moon landing to a 6 year old
Some people went to the moon...
SFT
Explain the moon landing to a 6 year old
A
Explain gravity...
B
Explain war...
C
Moon is natural satellite of...
D
People went to the moon...
D > C > A = B
RM
D > C > A = B
Write a story about frogs
PPO
Once upon a time...
RM
r_k

Jonathan Hui: how-much-do-i-like-chatgpt

ChatGPT Engineering

Frontend

- One-page application
- Social login supported
- Perceived low latency (as if someone is typing)
- Markdown support (tables, pics, ascii art, code besides text)

Backend

- Hard to know how many model replicas
  - Assume half million people use GPT 3x per day, so average QPS = 1.5M/86400 ~ 20qps
- Recently (from 01/10), ChatGPT is often too crowded to reject use

[Very very rough] $$ estimate to train/serve (1-month) a ChatGPT?

People: 25 (OPT 20, GPT3 ~30, LaMDA ~40, GLM ~20)
- Assume $25k per month, avg ~6 months, thus 3.75 million USD

Pretraining Computing
- Let’s say 2M (references from 0.5M to 4.6M)

Labeling (people write high quality instructions)
- Assume 100 (20-200 people based on Anthropic and InstructGPT and my irresponsible guess)
- Each paid 2k USD per month * 3 mon, so roughly 0.6M USD

Finetuning & RLHF training
- Hard to predict, but finetune is <10% than pretrain, while RL varies from 100% to 10x in my opinion
- So assume same cost as pretrain, 2M

Serving
- Assume half million users issued 3 request per day, and OpenAI CEO: “average is probably single-digits cents per chat”, so if we serve for 1 month
- 2-9 cents per request * 1.5 million request per day * 30 = 0.9 million to 4 million USD, assume 2M

* My [probably bad] guess: 3.75 + 2 + 0.6 + 2 + 2 = 10.35 million USD!

Assume enough resources ($$), what are the technical challenges?

1. **Difficult** to train GPT-3 codex version basic model using a large amount of text and code on a GPU cluster
2. Using high-quality expert Q&A data as a demonstration, fine-tuning it into a GPT-SFT.
3. **Difficult** to open the model and understand which representative questions are asked in which scenarios by the users.
4. **Difficult** to collect Q&A scenarios, allowing the model to generate various answers for human sorting, and using this to learn the reward model (input Q&A, output predicted reward value).
5. **Very difficult** to use language models and reward models to improve ability through reinforcement learning."
6. Difficult to wrap steps three to five to iterate multiple times, expecting to understand more questions, have a more accurate reward model, and a stronger model. The process also needs to introduce better evaluation mechanisms.
7. Using Moderation API to determine if the user's question is harmful.
8. ChatGPT system is released, questions are first reviewed and then the language model outputs the answer

ChatGPT fun facts and discussions

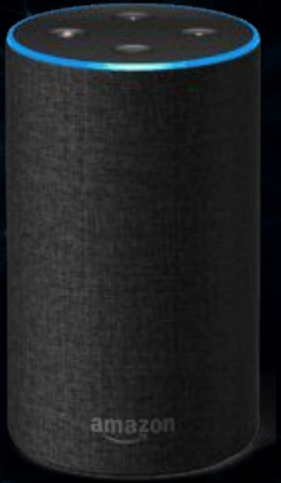
- Role-playing prompts
  - “Act as a […], blabla”
- Moderation API
  - A separate API/model as compared to ChatGPT model
- Coding help and analysis
- Context size
  - 8192 tokens?
- Integration with Microsoft
  - Bing
  - Office
- More to be added

ChatGPT Challenges

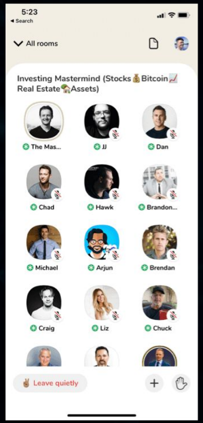
- Safety
- Bias
- Banned by schools (others said should NOT ban), stackoverflow, and…
- Potentially help to cheat for online interviews
- Hallucination
  - Gary Marcus: "How come GPT can seem so brilliant one minute and so breathtakingly dumb the next?"
  - Non-existing source citation
- Copyright (who ultimately owns content)
- Cost
  - Paid version came out recently
- And more

ChatGPT may become a next [...]?

iPhone?



Alexa?



Investing Mastermind (Stocks💰Bitcoin📈 Real Estate🏡Assets)

Clubhouse?

Thank you!

A Primer on Large Language Models (LLM)

github.com/hululuzhu/llm-primer

Last update: Jan 2023
hululu.zhu@gmail.com