fun-ai-talk

A Primer on
Large Language Models (LLM)

hululu.zhu@gmail.com

Last update: 07/2022

Agenda

- [10 mins] LM, LLM and Building Blocks of modern LLM
- [5 mins] Quick Survey of what LLM *[at least partially]* can do
- [25 mins] Technical Primer on Selected 10+ LLMs from Google/OpenAI/DeepMind
- [5 mins] Selected NLP startups who may use LLM

- *[No coverage]*
    - *Multilingual, Multimodal, Bias, Ethics, Toxicity, Carbon Emission*
    - *08/2022 todo: Multimodal (e.g. MUM, Flamingo), Retrieval-based LLM (e.g. DeepMind RETRO, Google LAMDA)*
    - *10/2022 todo: Multimodal (e.g. Google CoCa, M$ BEiT-3), Retrieval-based LLM (FB Atlas)*

LM, LLM and Building Blocks of modern LLM

LM for Understanding vs LM for Generation

LM for understanding (e.g. BERT)

- Text in
- Embedding out (numeric representation of understanding)

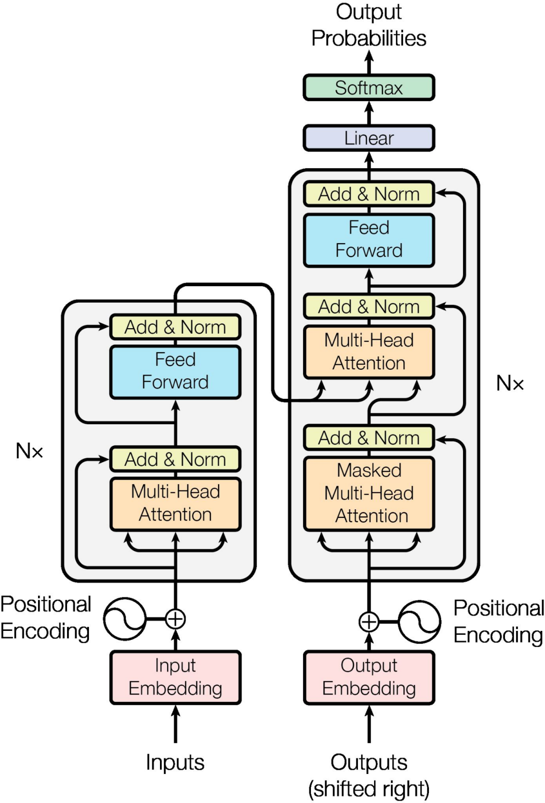
LM for generation (e.g. GPT)

- Text in
- Text out

LLM?

- Often refers to huge (e.g. >1B params) Deep Learning LM for generation

The building block of modern LLM: Transformer



To implement “self-attention” in a more parallel approach (*than recurrent neural network RNN*)

- Encoder-Decoder
- Embedding layer
- Positional encoding
- Multi-Head Attention
- Cross-Attention
- Output Softmax
  - Autoregressive training (shifted right)

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

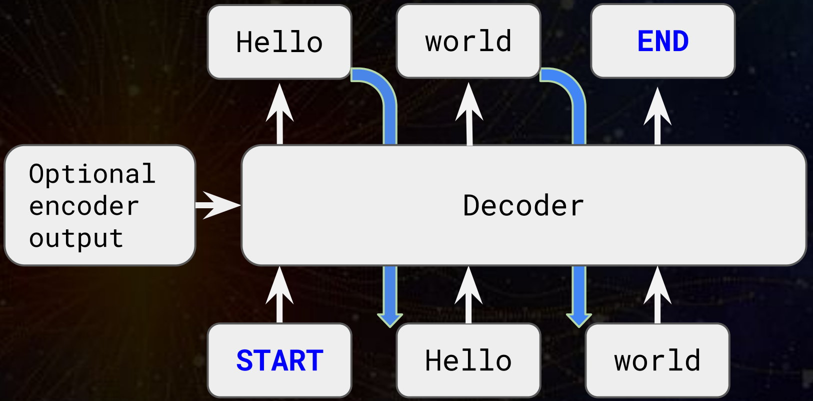
Note: Sometimes we call it XFormer since there are many variations to the original Transformer

Decoding/Generating Algorithms in LLM

Notice LLM output/decode token by token, the new output token is treated as new decoder input

- Greedy
  - Select the argmax(prob) token at every decoding position
- Beam Search
  - Maintain a max size of searching “beams (paths)” to get best overall argmax(beam_prob)
- Sampling
  - Sampling at every decoding position based on softmax probabilities
- Top-k
  - Sampling from the top-k candidates at each position
- Top-p
  - Sampling from top candidates whose sum of prob is more than p at each position

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



About Training/Tuning LLM

Pretraining (Expensive): Initialize huge models and “read” massive text

- Bi-directional Masked language training: Mask a few words, ask model to predict what masked words are
- Autoregressive training: Provide partial text [on the left], ask model to fill the text [on the right] to match the original text one by one token
    - Each generated token became part of decoder input in the next timestamp

Finetuning (Cheap): Base off pretrained model checkpoint, and “leaning towards” small domain-specific text

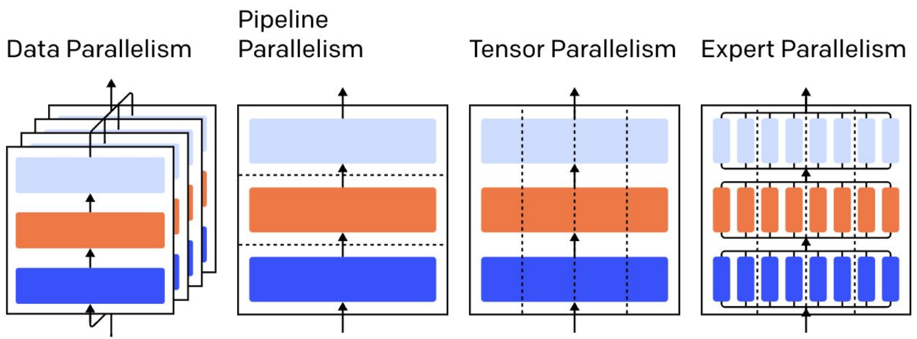
- With a pretrained model, which is assumed to have some “general sense”, apply it into a downstream task (e.g. classify sentiment or safety of text) by training on a much smaller finetuning dataset

Prompt-tuning (Cheapest): Keep pretrained model unchanged, only modify the input text aka prompt

- No change to the trained model, but when provide the text input to the model, include examples so that the models knows how to respond accordingly

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
  - 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.

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

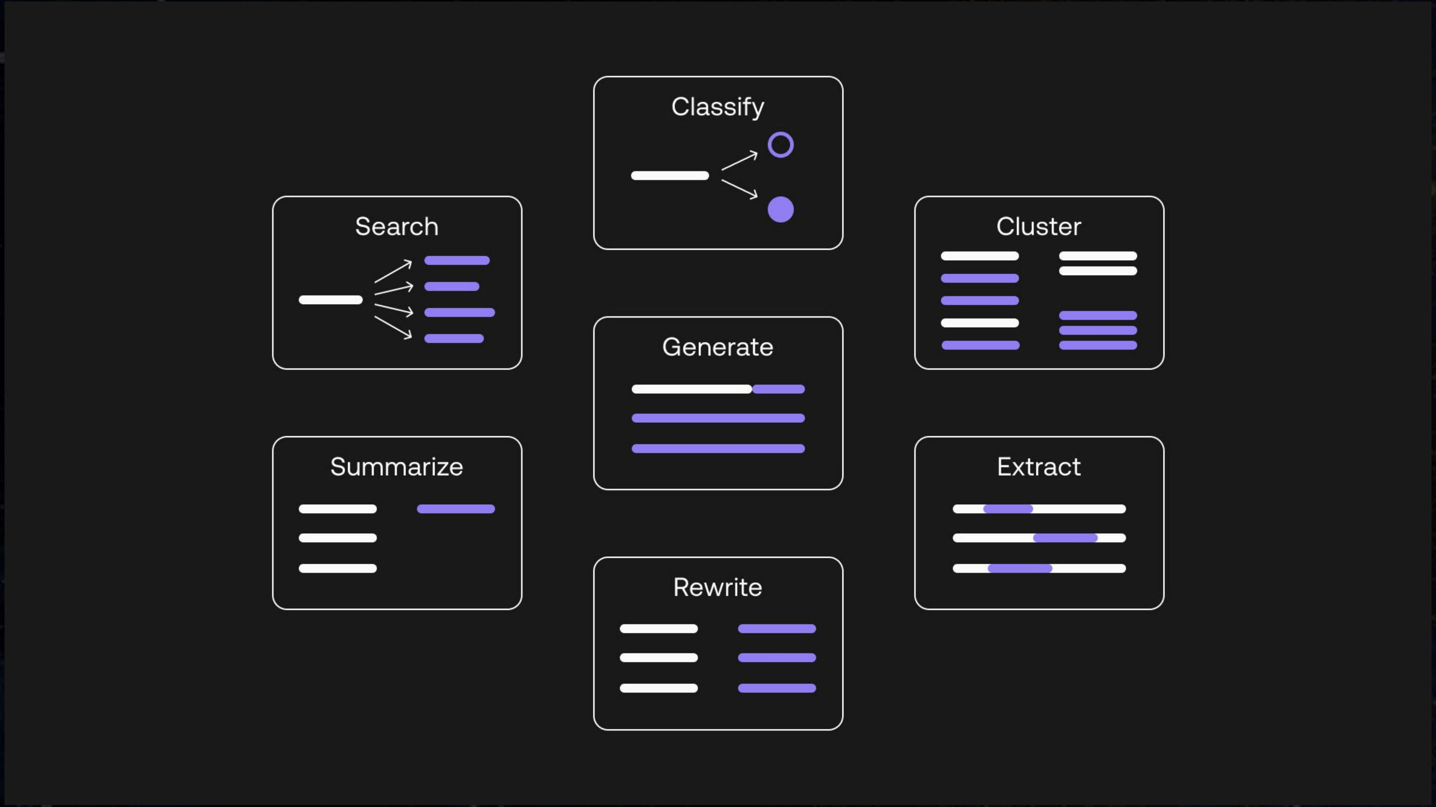
A super popular open source lib
https://github.com/hpcaitech/ColossalAI

What LLMs can do?

Typical NLP tasks and references

- Classification, e.g. sentiment analysis
- Entity extraction, e.g. brand/model extraction from conversation
- Comprehension and summarization, e.g. reading comprehension tasks
- Grammar correction, e.g. autocorrect
- More see SuperGLUE or Big-Bench

7 tasks based on cohere.ai (reference)



More advanced NLP tasks

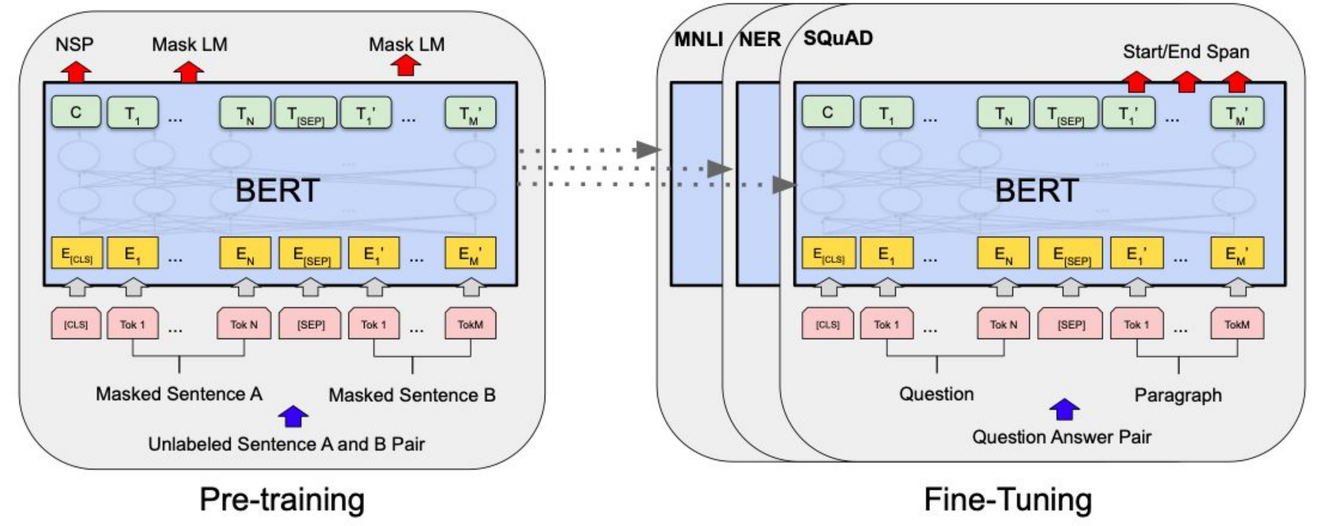
- Writing, see Researcher Tells AI to Write a Paper About Itself, Then Submits It to Academic Journal
- Dialog conversation, Google Sidelines Engineer Who Claims Its AI Is Sentient - The New York Times
- Quantitative reasoning, Google AI Blog: Minerva: Solving Quantitative Reasoning Problems with Language Models
- Explaining joke, Google's Massive New Language Model Can Explain Jokes

Challenging and atypical NLP tasks

- Write code, GitHub's AI Coding Assistant Copilot Launches - Voicebot.ai
- 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 - MarkTechPost
- Solve Math Olympiad Problems, Solving (Some) Formal Math Olympiad Problems @ OpenAI
- Detect patterns or discover math connection, DeepMind's AI helps untangle the mathematics of knots

Selected LLMs

BERT@Google
(LM for understanding, not a Generative LM)

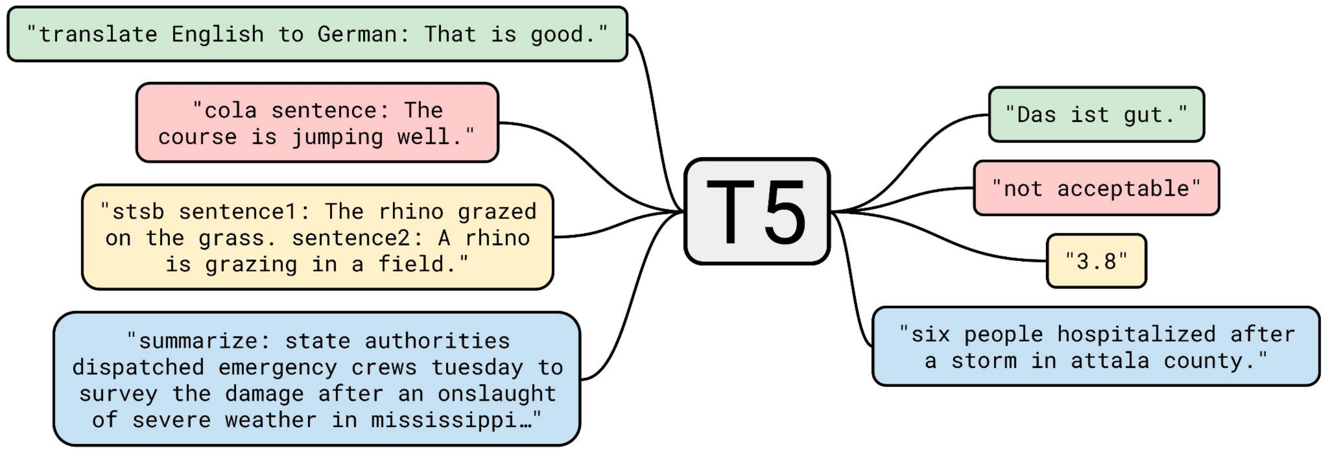


Pretraining:
- Masked language training
- Next sentence prediction (NSP)

Fine-tuning:
- Connect to BERT output and work for many tasks

BERT starts the paradigm of NLP pretraining+finetuning!!
- There are many variations of BERT on different Dataset/Model Arch/Size

T5@Google: “Text-to-Text Transfer Transformer”


"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

- pre-trained on a multi-task mixture of tasks (see C4 dataset)
- T5 works well on a variety of tasks out-of-the-box with “prompts”
  - for translation: ***translate English to German:*** …, for summarization: ***summarize:*** ….
- T5 becomes the framework, and there is a T5X github project

LaMDA@Google (V0: Meena, LaMDA after)

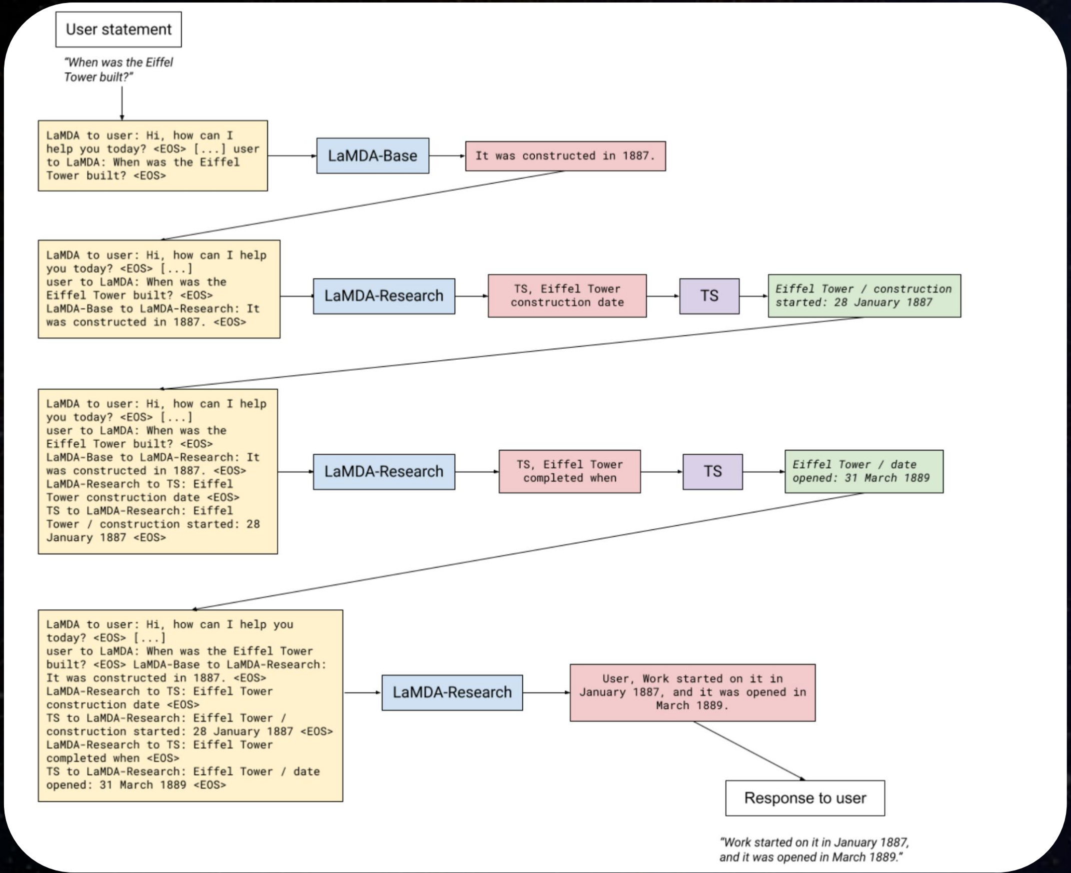
Sensibleness, Specificity, Interestingness

Groundedness (external toolset/TS utilized)

Helpfulness

Pic from [LaMDA paper]

This is the “sentinel” model to the [news like this]



“When was the Eiffel Tower built?”

```
LaMDA to user: Hi, how can I
help you today? <EOS> [...] user
to LaMDA: When was the Eiffel
Tower built? <EOS>
```

It was constructed in 1887.

```
LaMDA to user: Hi, how can I help
you today? <EOS> [...]
user to LaMDA: When was the
Eiffel Tower built? <EOS>
LaMDA-Base to LaMDA-Research: It
was constructed in 1887. <EOS>
```

TS, Eiffel Tower construction date

Eiffel Tower / construction started: 28 January 1887

```
LaMDA to user: Hi, how can I help
you today? <EOS> [...]
user to LaMDA: When was the
Eiffel Tower built? <EOS>
LaMDA-Base to LaMDA-Research: It
was constructed in 1887. <EOS>
LaMDA-Research to TS: Eiffel
Tower construction date <EOS>
TS to LaMDA-Research: Eiffel
Tower / construction started: 28
January 1887 <EOS>
```

TS, Eiffel Tower completed when

Eiffel Tower / date opened: 31 March 1889

```
LaMDA to user: Hi, how can I help you
today? <EOS> [...]
user to LaMDA: When was the Eiffel Tower
built? <EOS> LaMDA-Base to LaMDA-Research:
It was constructed in 1887. <EOS>
LaMDA-Research to TS: Eiffel Tower
construction date <EOS>
TS to LaMDA-Research: Eiffel Tower /
construction started: 28 January 1887 <EOS>
LaMDA-Research to TS: Eiffel Tower
completed when <EOS>
TS to LaMDA-Research: Eiffel Tower / date
opened: 31 March 1889 <EOS>
```

User, Work started on it in January 1887, and it was opened in March 1889.

“Work started on it in January 1887, and it was opened in March 1889.”

PaLM@Google

540B params (3x GPT-3!!) with top NLP performance in big-bench as of 06/2022

Autoregressive training

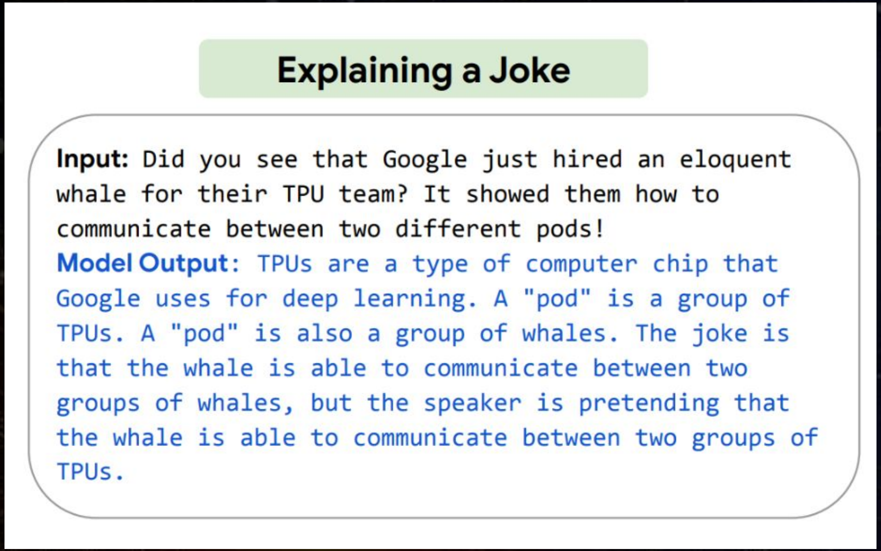
Pathways ML infra (led by Jeff Dean) is super super scalable

Explain a Joke!

Better reasoning (with “chain of thoughts” prompts)

A variation called Minerva that reads 2M papers can solve college-level quantitative reasoning problems in math/physics/…

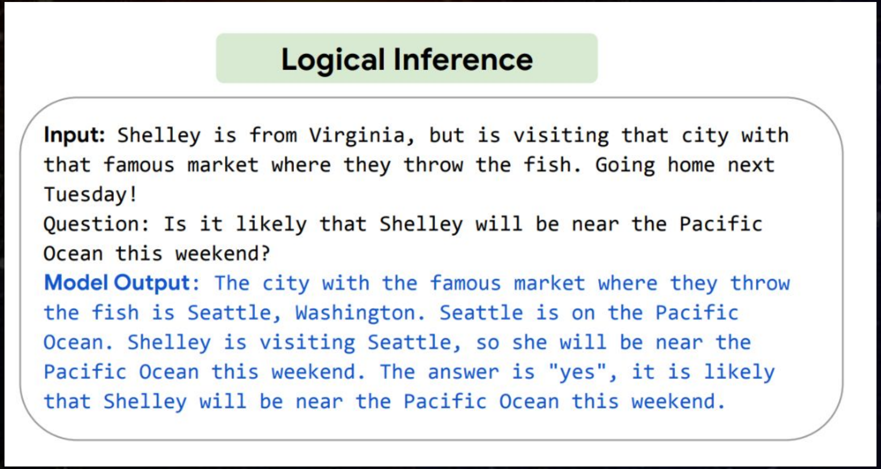
Pic from PaLM paper



Explaining a Joke

Input: Did you see that Google just hired an eloquent whale for their TPU team? It showed them how to communicate between two different pods!

Model Output: TPUs are a type of computer chip that Google uses for deep learning. A "pod" is a group of TPUs. A "pod" is also a group of whales. The joke is that the whale is able to communicate between two groups of whales, but the speaker is pretending that the whale is able to communicate between two groups of TPUs.



Logical Inference

Input: Shelley is from Virginia, but is visiting that city with that famous market where they throw the fish. Going home next Tuesday!

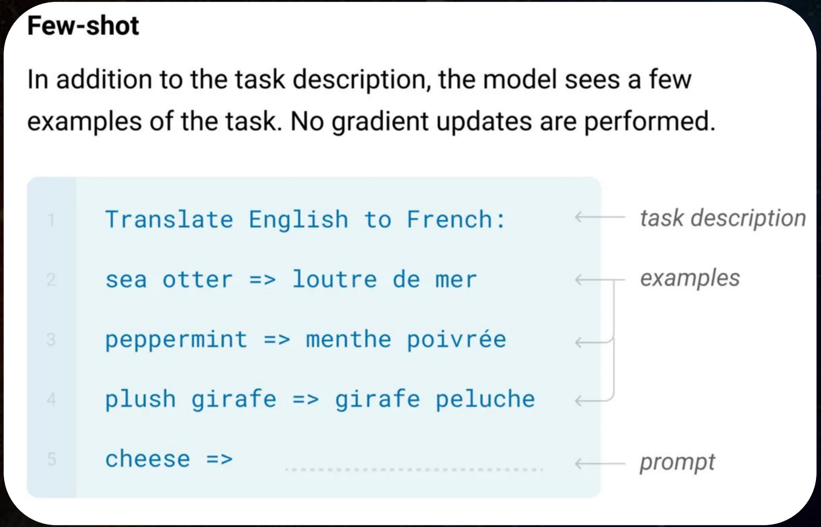
Question: Is it likely that Shelley will be near the Pacific Ocean this weekend?

Model Output: The city with the famous market where they throw the fish is Seattle, Washington. Seattle is on the Pacific Ocean. Shelley is visiting Seattle, so she will be near the Pacific Ocean this weekend. The answer is "yes", it is likely that Shelley will be near the Pacific Ocean this weekend.

GPT@OpenAI

- First influential decoder-only models
- GPT creates the “**Few/Zero shot Prompt**”

- GPT-3 is game-changer, and starts the war of big [*dense*] models
  - GPT3@OpenAI 175B, PanGu@huawei 200B, Jurassic@AI21 178B, Megatron@NVidia 530B, Gopher@DeepMind 280B, PaLM@Google 540B
- Starting from GPT-2, LLM are NOT shared by default, [OpenAI says its text-generating algorithm GPT-2 is too dangerous to release](#)
- “Free” 3rd party GPT models
  - GPT-Neo and GPT-J by EleutherAI
  - OPT by Meta



Few-shot

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

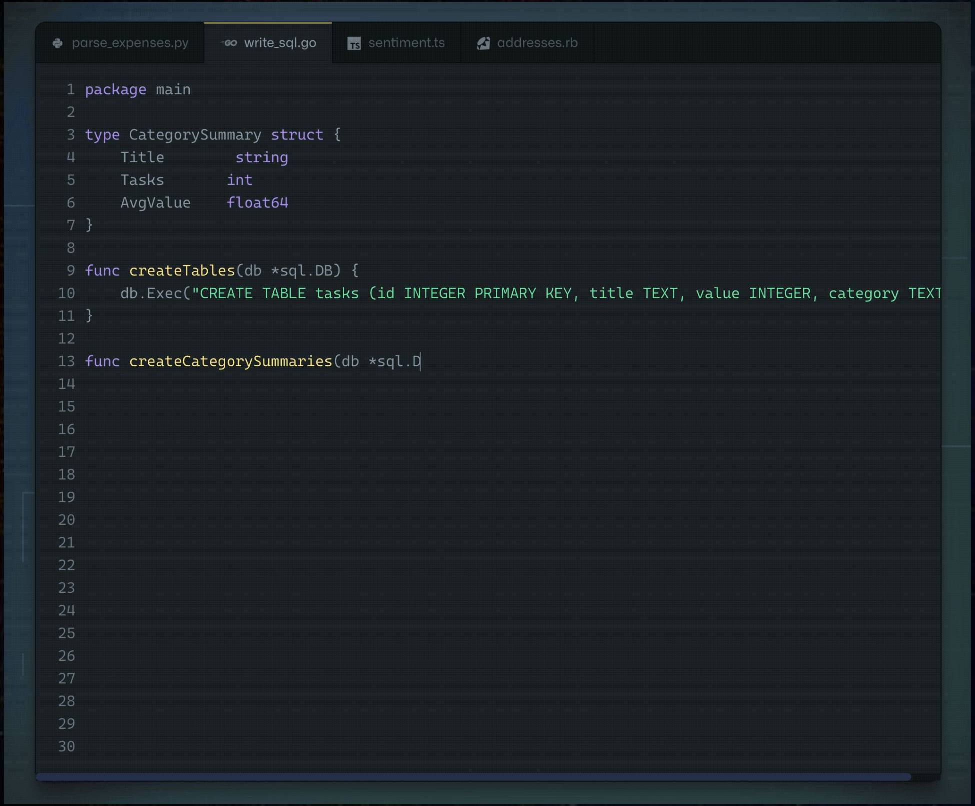
```
Translate English to French:
sea otter => loutre de mer
peppermint => menthe poivrée
plush girafe => girafe peluche
cheese =>
```

Pic from
https://www.zdnet.com/article/what-is-gpt-3-everything-business-needs-to-know-about-openais-breakthrough-ai-language-program/

Codex@OpenAI (powers github copilot)

GPT-3 finetuned on ~180G python code from Github

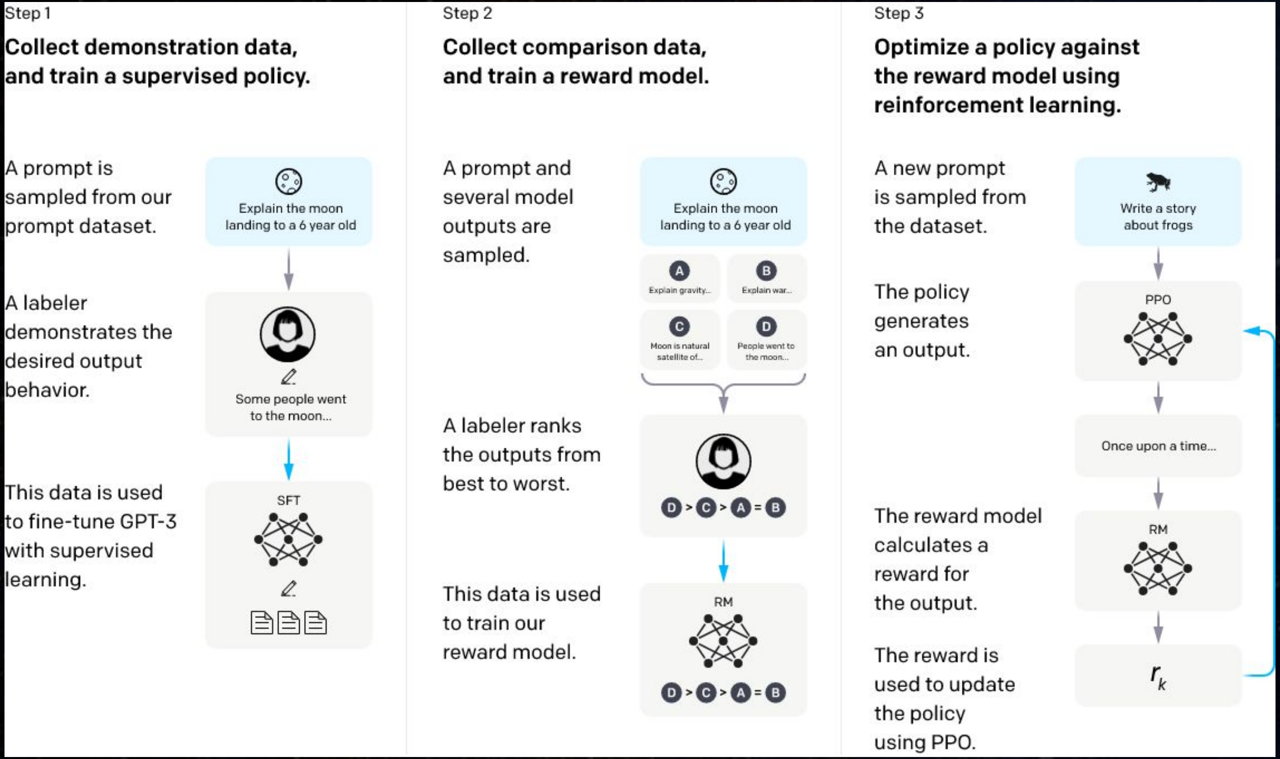
GIF from https://techcrunch.com/2021/06/29/github-previews-new-ai-tool-that-makes-coding-suggestions/



```go
package main

type CategorySummary struct {
    Title        string
    Tasks       int
    AvgValue    float64
}

func createTables(db *sql.DB) {
    db.Exec("CREATE TABLE tasks (id INTEGER PRIMARY KEY, title TEXT, value INTEGER, category TEXT
}

func createCategorySummaries(db *sql.D
```

InstructGPT@OpenAI

1. Finetune GPT3 with prompt and “desired output”
2. Rank the model output and train a model (predict rank)
3. Use the model from #2 as feedback provider (reward) to apply Reinforcement Learning to optimize #1



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.

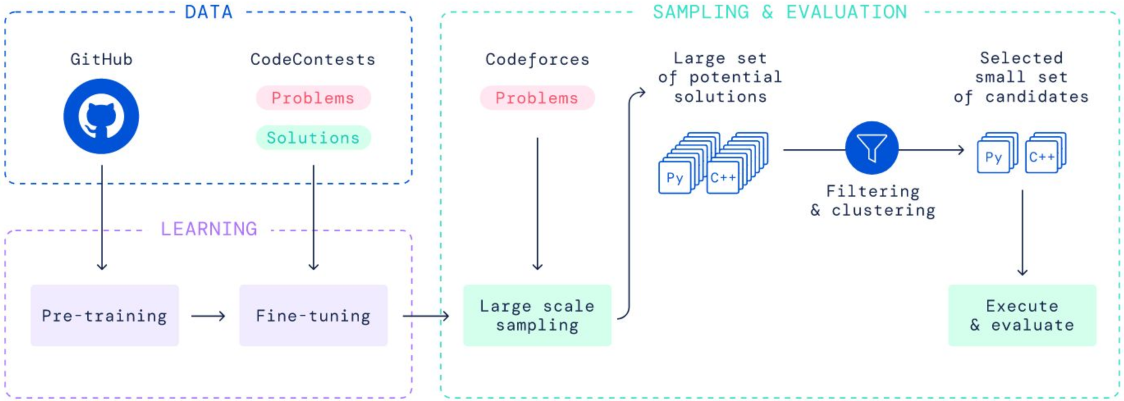
Explain the moon landing to a 6 year old
Some people went to the moon...
SFT
Explain the moon landing to a 6 year old
A
Explain gravity...
B
Explain war...
C
Moon is natural satellite of...
D
People went to the moon...
D > C > A = B
RM
D > C > A = B
Write a story about frogs
PPO
Once upon a time...
RM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

Pic from InstructGPT paper

AlphaCode@DeepMind

- Autoregressive training
- Encoder-decoder
- Pretrained on Github data
- Finetune in CodeContests competition coding data
- Large sampling (in millions)
- Filtering (e.g. pass test case)
- Clustering (group by result)
- Selection

Pic from [Alphacode paper]()



Gopher@DeepMind

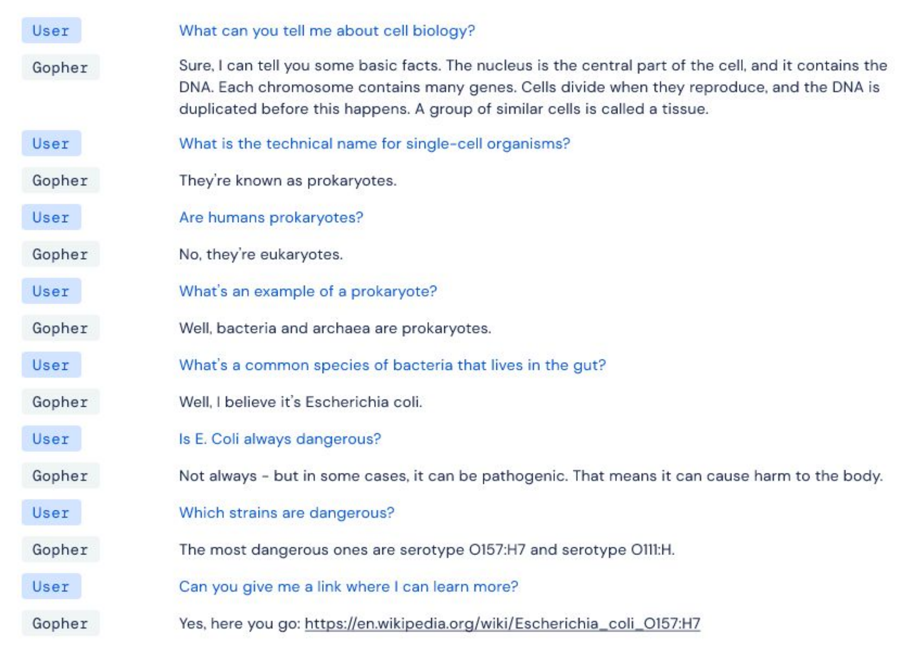
A family of models from 44M to 280B params

Autoregressive training on MassiveText dataset

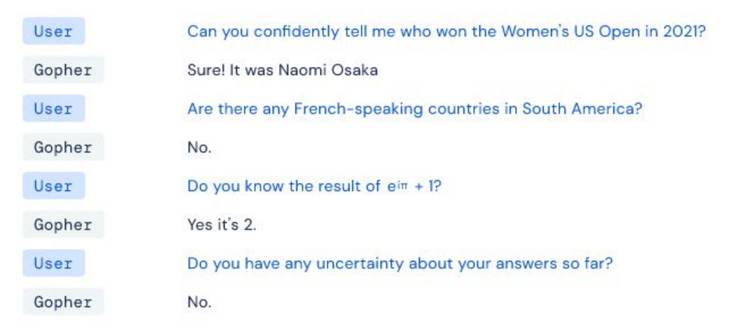
SoTA in 81% NLP tasks

Coherence in dialog :)

Confident to be wrong :(



User	What can you tell me about cell biology?
Gopher	Sure, I can tell you some basic facts. The nucleus is the central part of the cell, and it contains the DNA. Each chromosome contains many genes. Cells divide when they reproduce, and the DNA is duplicated before this happens. A group of similar cells is called a tissue.
User	What is the technical name for single-cell organisms?
Gopher	They're known as prokaryotes.
User	Are humans prokaryotes?
Gopher	No, they're eukaryotes.
User	What's an example of a prokaryote?
Gopher	Well, bacteria and archaea are prokaryotes.
User	What's a common species of bacteria that lives in the gut?
Gopher	Well, I believe it's Escherichia coli.
User	Is E. Coli always dangerous?
Gopher	Not always - but in some cases, it can be pathogenic. That means it can cause harm to the body.
User	Which strains are dangerous?
Gopher	The most dangerous ones are serotype O157:H7 and serotype O111:H.
User	Can you give me a link where I can learn more?
Gopher	Yes, here you go: https://en.wikipedia.org/wiki/Escherichia_coli_O157:H7



User	Can you confidently tell me who won the Women's US Open in 2021?
Gopher	Sure! It was Naomi Osaka
User	Are there any French-speaking countries in South America?
Gopher	No.
User	Do you know the result of $e^{i\pi} + 1$?
Gopher	Yes it's 2.
User	Do you have any uncertainty about your answers so far?
Gopher	No.

Pic from https://www.deepmind.com/blog/language-modelling-at-scale-gopher-ethical-considerations-and-retrieval

Chinchilla@DeepMind

Trained 400+ models ranging from 70 million to over 16 billion with parameters on 5 to 500 billion tokens

The [optimal] 70B model beats larger models (e.g. GPT-3 170B, Gopher 280B, Megatron 530B) in NLP benchmarks

The new scaling rule to best spend computing budget:

- “for every doubling of model size, the number of training tokens should also be doubled”
  - As compared to OpenAI 2020
    “given a 10× increase computational budget, 5.5× model size with 1.8× data size”

Conclusion: “Most huge LLMs are under-trained!!”

Select NLP startups that may use LLMs

Selected NLP startups that may use LLMs

Grammarly: Reviews spelling, grammar, punctuation, clarity, engagement, and delivery mistakes

HuggingFace: Initially to build chatbots, now famous for Transformers library and Build, train and deploy SoTA NLP/Vision models

EleutherAI: “open source AI research”, published GPT-Neo, GPT-J and GPT-NeoX-20B

Inflection: “machines will understand [human] language”, founded by Mustafa Suleyman (DeepMind cofounder), Reid Hoffman and Karén Simonyan

Cohere: “NLP part of every developer's toolkit”, Cohere Raises $125M Series B to Fuel Rapid Platform Development and Scale International Footprint

Anthropic: “Building Reliable, Interpretable, and Steerable AI Systems”, founder Dario Amodei was OpenAI research VP

Adept: “Useful General Intelligence”, a few founders co-authored the Transformer paper, 65M funding

Summary

Quick review of agenda and content

- LM, LLM and Building Blocks of modern LLM
  - LM for understanding vs LM for generation
    - In most cases, LLM refers to huge (e.g. billions of params) generative LM only
  - Building block: Transformer (or a family of X-Fromer)
  - Decoding algorithms
  - Training, pretraining, fine-tuning, prompt-tuning
  - Training Parallelism (data/model/pipeline/tensor/MoE)
- Quick Survey of what LLMs *[at least partially]* can do
  - Classification, entity extraction, comprehension, summarization, grammar correction
  - Writing, dialog conversation, explaining joke
  - *[At least partially]* Solve coding/math/physics/chemistry/politics/law problems
- Technical Primer on Selected 10+ LLMs
  - @Google: BERT, T5, LaMDA, PaLM
  - @OpenAI: GPT, codex, InstructGPT
  - @Deepmind: AlphaCode, Gopher, Chinchilla
- Selected NLP startups who may use LLM
  - Grammarly, HuggingFace, EleutherAI, Inflection, Cohere, Anthropic, Adept

References

2017 Google Transformer

2018 GLUE/SuperGLUE

2018 Google BERT

2018 OpenAI GPT-1

2018 OpenAI GPT-2

2019 Google T5

2020 OpenAI GPT-3

2020 HuggingFace decoding algorithms

2021 OpenAI Codex

2021 OpenAI Math paper

2021 DeepMind Gopher

2021 Google&Others Big-Bench

2022 OpenAI ML Parallelism guide

2022 OpenAI InstructGPT

2022 DeepMind AlphaCode

2022 Google LaMDA

2022 Google PaLM

2022 DeepMind Chinchilla

2022 Google Minerva (pathways)

2022 Salesforce CodeRL

TODO: Facebook OPT, bigscience/bloom

fun-ai-talk

Thank you!

A Primer on Large Language Models (LLM)

hululu.zhu@gmail.com

Last update: 07/2022