

Generative LLMs

Gaurav Gada

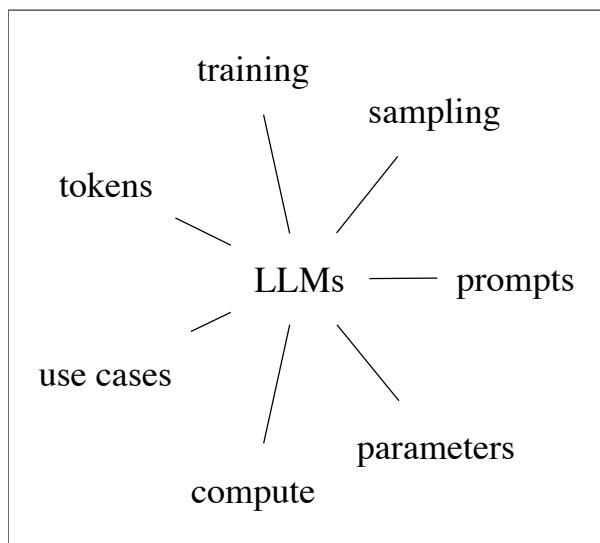
Created: 2024-01-24 Wed 00:52

LLMs

Note: The subject of this talk is **Generative LLMs**, i.e. ChatGPT etc.; whenever we mention LLM, we mean **Generative LLMs**. BERT, DistilBERT are also "LLMs";

LLM aspects that we'll talk about:

- What are LLMs?
- Recent Breakthroughs
- How do they work?
- How are they trained?
- Shifts in NLP
- LLM Challenges



What is a "LLM"

- LLM: Large Language Model
 - Models with a large number of trained parameters
 - GPT-4 expected to have 100 trillion parameters.
- Deep Neural Nets
- Trained on large quantities of unlabeled data
 - ChatGPT: Common Crawl, WebText2, Books1 & Books2 (public domain books), Wikipedia
 - 570 GB, 300 B words

LLM Terms 101

We will briefly discuss the common terms used to talk about LLMs: *tokens*, *vocabulary*, *training*, *parameters*, *encoder*, *decoder* and *prompt*

TOKENS

- Fundamental unit of text used to train model
- A word by itself can be a token
 - (-) Too large vocabulary to cover language
 - English Wiktionary: ~1.4 million definitions
 - (-) memory constraints
- Sub word tokens
 - I have a new GPU!
 - "i", "have", "a", "new", "gp", "##u", "!"

VOCABULARY

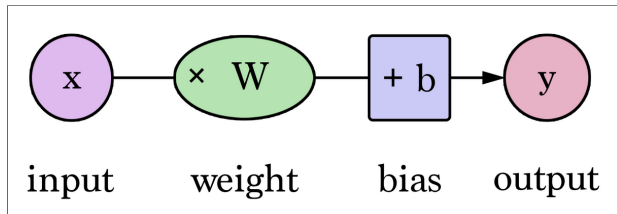
- The set of tokens in the training data that the model sees.
 - As mentioned, English Wiktionary: ~1.4 million definitions
- With sub-word tokenization
 - ~50K vocabulary for English
 - ~250K vocabulary for ~100 languages
 - (+) memory efficient

MODEL TRAINING

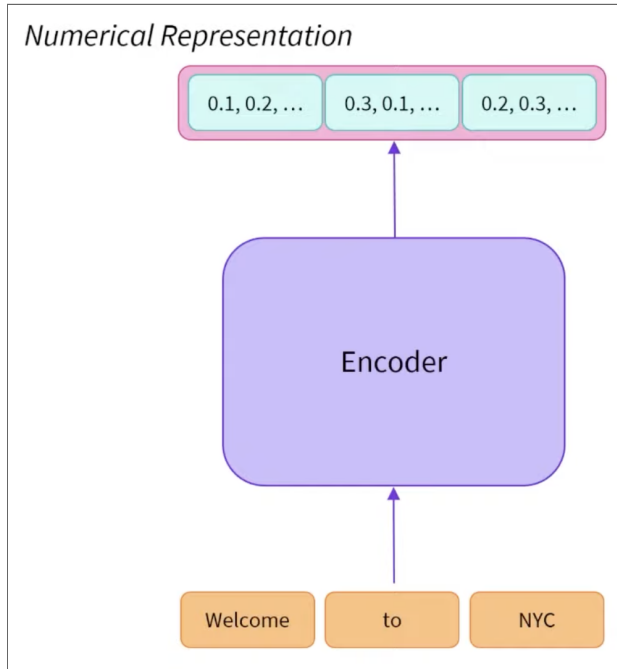
- The goal of a generative LLM is to predict the probability distribution over a vocabulary
- The *model loss* is the difference in the model prediction (probability distribution over vocabulary) and actual next word
 - Cross-entropy
 - $-np.\text{sum}(Y * np.\log(P) + (1 - Y) * np.\log(1 - P))$
- The *model training* process minimizes this loss over all training data

PARAMETERS

- Corresponding term to "weights" in standard neural network
- Random co-efficients that are *trained* to minimize loss
- They are the strength of connection between neurons
- And also the biases in a neuron



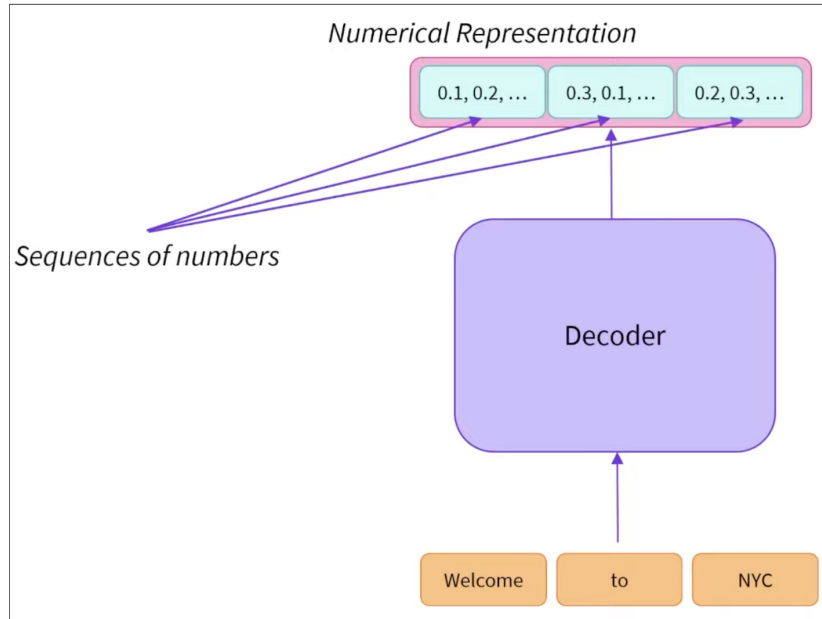
ENCODER



- Input: text, output: vector
- One vector sequence per word
- Takes left and right context into account (bidirectional)
- (+) extracting meaningful information
- (+) Masked Language Modeling (fill in the blanks)
- (+) sequence classification
- BERT is an encoder with 768 vector length

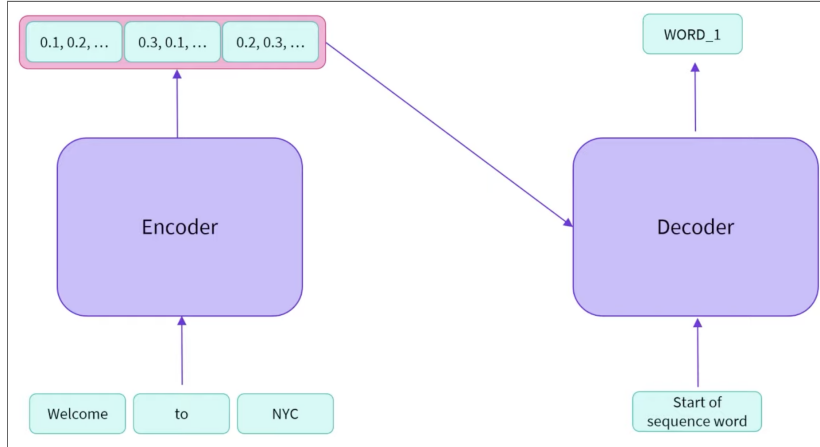
[HuggingFace. "Transformer models: Encoders." YouTube, 14 June 2021, www.youtube.com/watch?v=MUqNwgPjJvQ.](https://www.youtube.com/watch?v=MUqNwgPjJvQ)

DECODER



- Input: text, output: vector
 - "autoregressive" feeding
- One vector sequence per word
- Takes either left or right context (unidirectional)
 - Context window, e.g. 1,024 tokens for GPT-2
- (+) Generating sequences when stacked with a language modeling head
- (+) Causal tasks
- GPT-2, GPT-Neo are decoders

ENCODER-DECODER



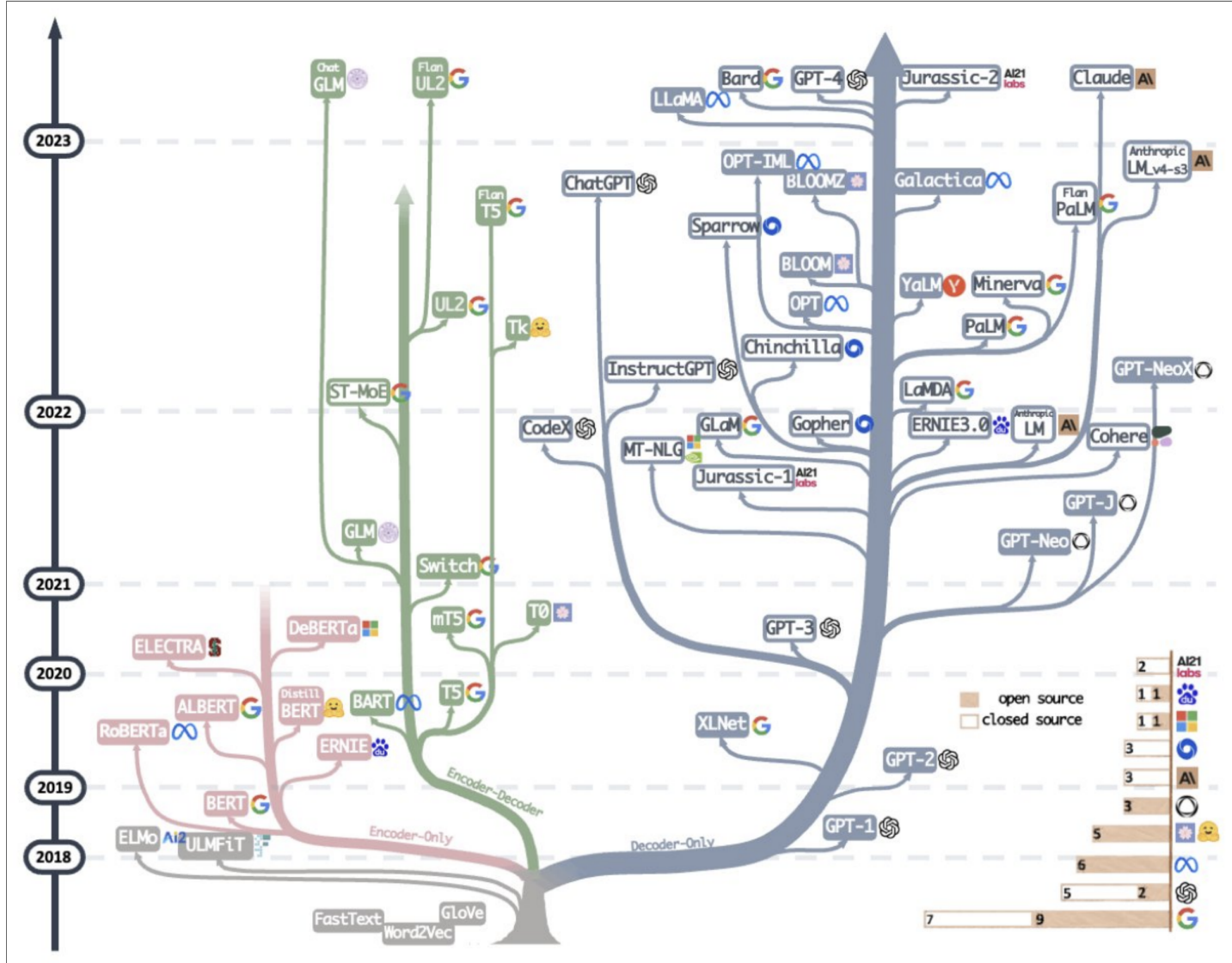
- Input sequence, output sequence
 - Vector output of encoder fed to decoder along with start sequence token
 - Output of decoder fed back to itself with output of encoder in the next time steps
- Encoder and decoder do not share weights
- Input and output sequence can be different lengths
- (+) translation
- (+) summarization
- (+) multi modal applications

[HuggingFace. "Transformer models: Encoder-Decoders." YouTube, 14 June 2021, www.youtube.com/watch?v=04KEb08xrE.](https://www.youtube.com/watch?v=04KEb08xrE)

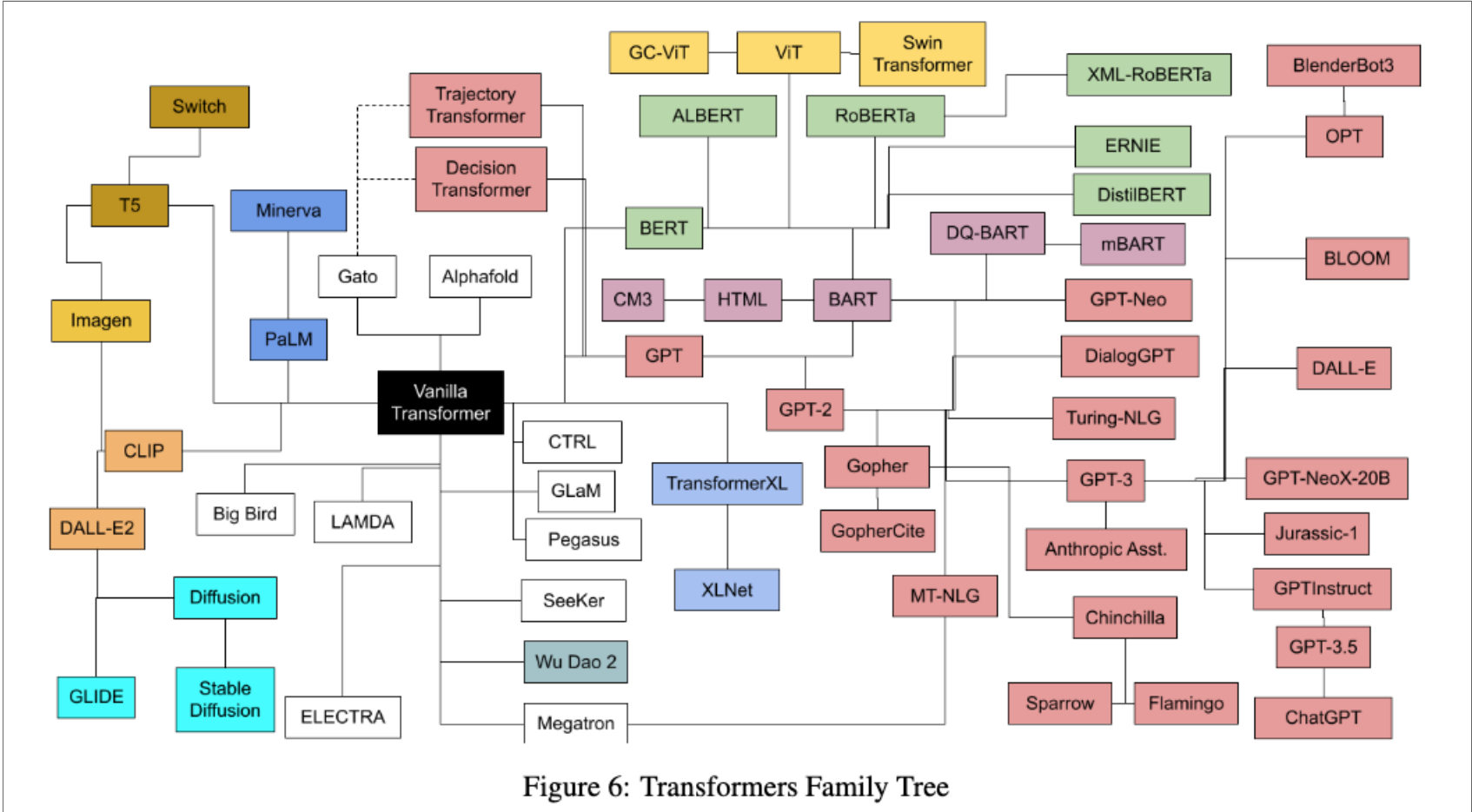
PROMPT

- Input text given to LLM to kick off generation
- Also called "prefix"
- We "prime" the model with a prompt

LLM FAMILY



Growth of the Transformer Family



Amatriain, Xavier. "Transformer models: an introduction and catalog." arXiv, 12 Feb. 2023, <https://doi.org/10.48550/arXiv.2302.07730>.

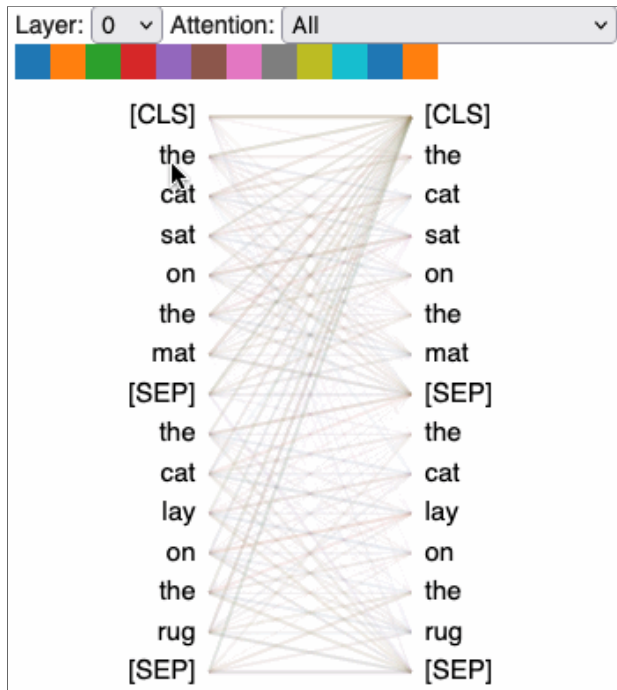
How does a LLM work

- At the core of Generative Text is the decoding process.
- Sampling techniques that help LLMs achieve human-like text quality

Attention at the core

Attention (2017) from the transformer architecture revolutionized text encoding.

- Multiple heads concept (heads can be thought of as channels), different heads pay attention to different parts of text/image/video. Each color intensity represents head value of self-attention in chart below.



BertViz - Visualize Attention in NLP Models

Gen LLM Families

- There are 2 families of generative LLMs (from LLM 101)
 - Decoder only: a.k.a *autoregressive generative model*
 - Encoder-Decoder: Pay attention to input as well as generated tokens; e.g. BART
- Given a sequence of words, it generates the next words.
- It predicts the probability distribution over the vocabulary given a sequence of tokens
 - Based on learned parameters

Recurrent Decoding

- A generative LLM takes an input and predicts the next word probability over the entire vocabulary
- It then picks a token from the vocabulary based on the the probability distribution of the output
- It appends the predicted token and feeds it back to input to the model in the next time step
 - "recurrent decoding"
 - "autoregressive decoding"
- The whole process repeats until it finds a stopping token (or when we ask it to stop)

Sampling The Output

Methods to sample from the predicted vocabulary probability distribution:

- Greedy: pick the token with highest probability always
- Beam Search: Explore multiple sequences in parallel
- Top-k: Randomly sample among top-k with tokens highest probability
- Top-p/Nucleus: Randomly sample among top tokens whose probabilities adds up to p

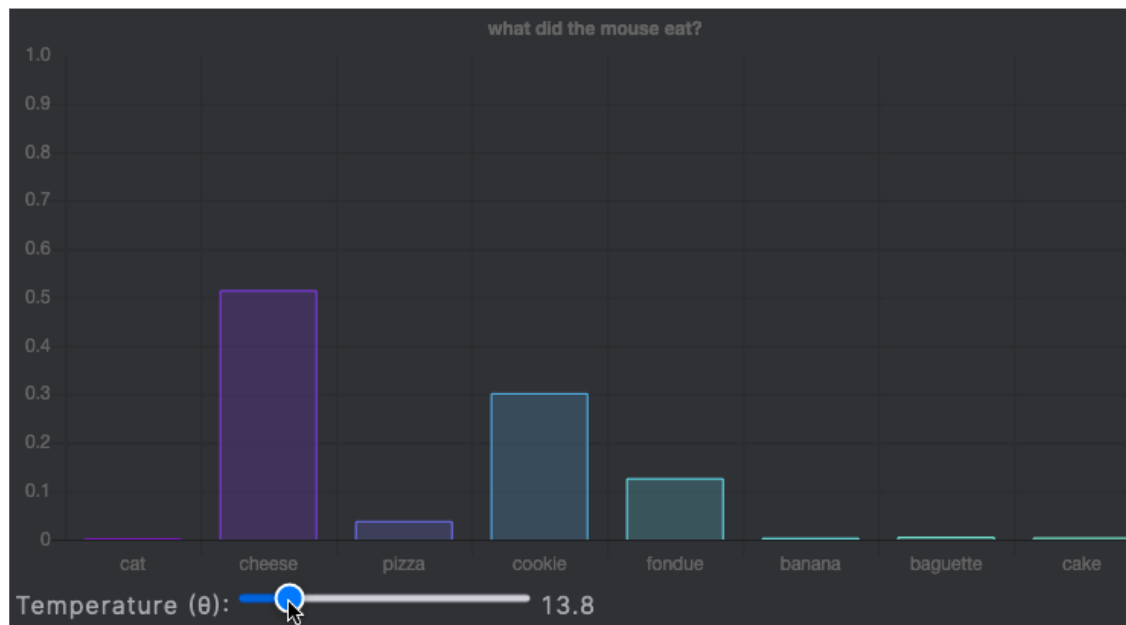
Each have a different impact on computation time and text quality. Greedy doesn't give the best text output, beam search is slightly better, top-p and top-k with temperature are more useful and human-like.

TEMPERATURE

Temperature parameter helps with shaping output probabilities.

$$p(x = V_l | x_{1:i-1}) = \frac{\exp(\frac{u_l}{t})}{\sum_{l'} \exp(\frac{u_{l'}}{t})}$$


1. Lower temperature: focus on top probabilities; more concise and factual responses
2. Higher temperature: dampening, bring in the tail, more creative responses; e.g. poem generation.



["What is Temperature in NLP? 🐭." Luke Salamone's Blog, 2 Apr. 2021, lukesalamone.github.io/posts/what-is-temperature.](https://lukesalamone.github.io/posts/what-is-temperature/)

Effects of sampling parameters


Input:



An unprecedented number of mostly young whales have become stranded on the West Australian coast since 2008.

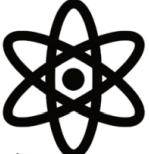
WebText

Beam Search:



The number of stranded whales has increased by more than 50 per cent in the past year, with the number of stranded whales on the West Australian coast increasing by more than 50 per cent in the past year. The number of whales stranded on the West Australian coast has increased by more than 50 per cent in the past year, with the number of stranded whales on the West Australian coast increasing by more than 50 per cent in the past year.

Beam Search, $b=16$



There has been an unprecedented number of calves caught in the nets of whaling stations that operate in WA. Pilot whales continue to migrate to feeding grounds to feed their calves. They are now vulnerable due to the decline of wild populations; they are restricted to one breeding site each year. Image copyright Yoon Bo Kim But, with sharp decline in wild populations the size of the Petrels are shrinking and dwindling population means there will only be room for a few new fowl.

Nucleus, $p=0.95$

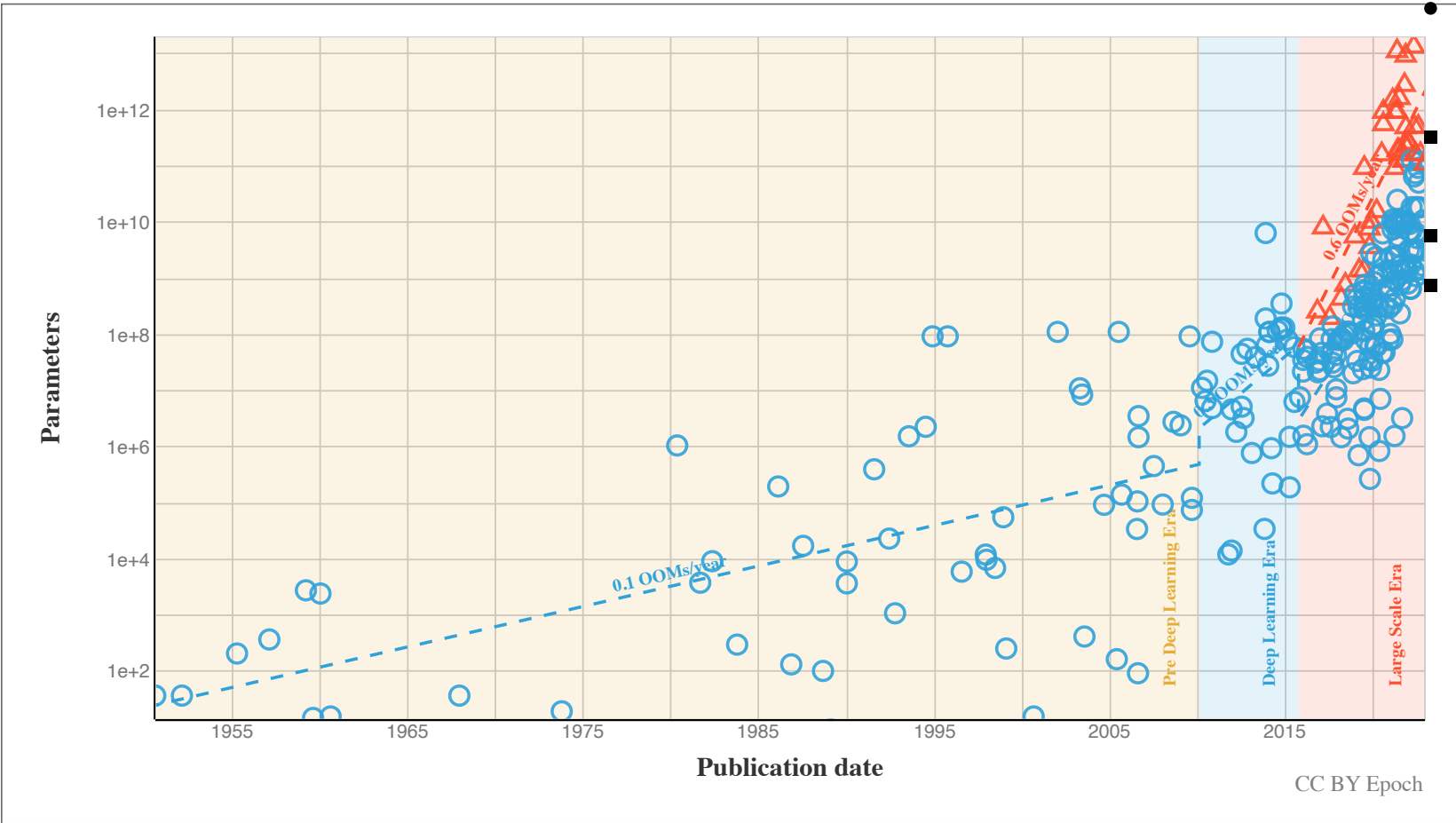
Holtzman, Ari, et al. "The Curious Case of Neural Text Degeneration." arXiv, 22 Apr. 2019, doi:10.48550/arXiv.1904.09751.

Recent Breakthroughs

- Parameters, compute and training data increase rapidly since 2015
- Attention! (2017), basis of BERT and other advances in NLP
 - Introduces the Transformer architecture
 - GPT: Generatively Pre-Trained **Transformer**
- Emergent properties emerge as parameters increase exponentially
 - An ability is emergent if it is not present in smaller models but is present in larger models. [1]
 - Given a few prompts that illustrate an unseen task, model responds correctly (few-shot setting)
- Reinforcement Learning joins forces with NLP, especially PPO (proximal policy optimization), a new breed of RL optimization.

[1] Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., ... Fedus, W. (2022). Emergent Abilities of Large Language Models. Transactions on Machine Learning Research. Retrieved from <https://openreview.net/forum?id=yzkSU5zdwD>

Model Parameter Sizes



● Since 2015:

- ▲ Large Scale
- All

■ Parameters increased 6x

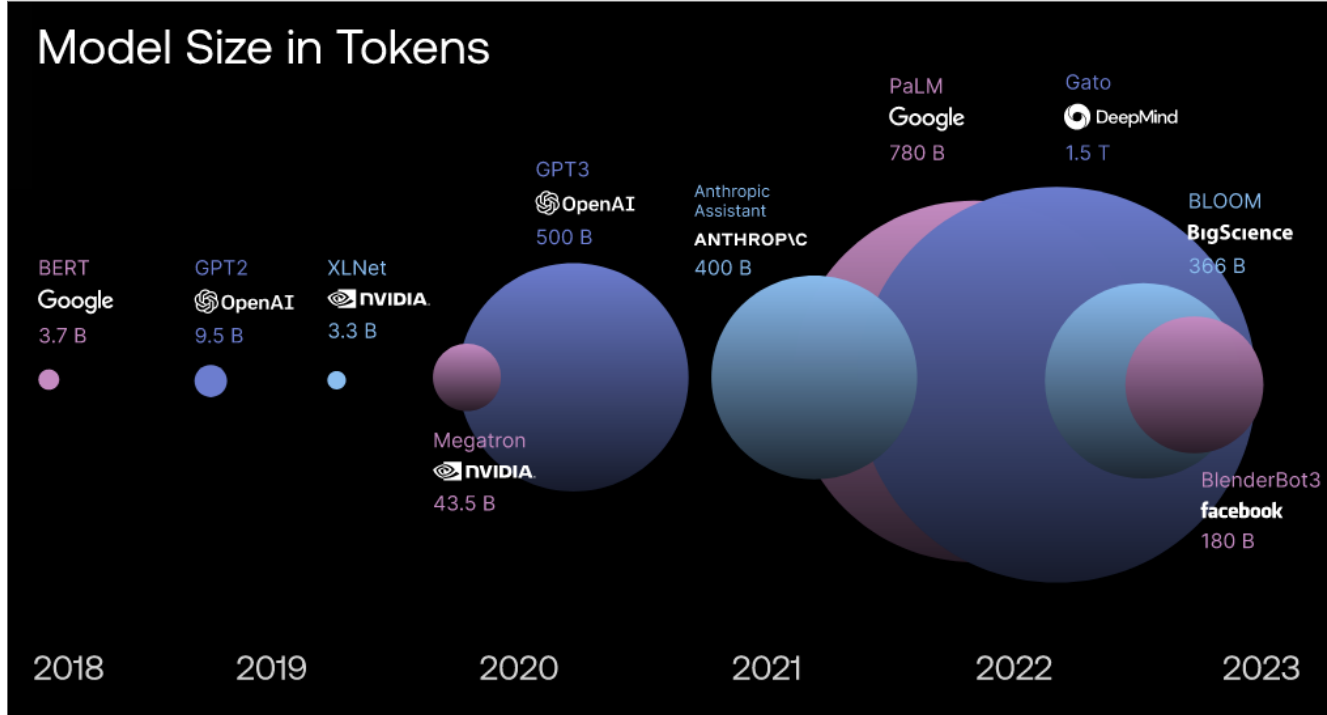
■ Training FLOPs 3.5x

■ Training Dataset Size 6x

Sevilla, Jaime, et al. "Compute Trends Across Three Eras of Machine Learning." arXiv, 11 Feb. 2022,

doi:10.48550/arXiv.2202.05924.; [chart url](#)

Training Data



Summary of tokens in training data and model parameters:

- BERT (2018) was 3.7B tokens and 240 million parameters.
- GPT-2 (2019) was 9.5B tokens and 1.5 billion parameters.
- GPT-3 (2020) has 499B tokens and 175 billion parameters.
- PaLM (2022) was 780B tokens and 540 billion parameters.

"AI Readiness Report 2023 | Scale AI." ScaleAI, 21 Apr. 2023, scale.com/ai-readiness-report.

Training LLMs

Training an LLM consists of the following steps.

- Data preparation: corpus collection, sampling, tokenization
- Generative Pre-training (i.e. training for scratch)
- Supervised Fine-tuning
- RL from human feedback

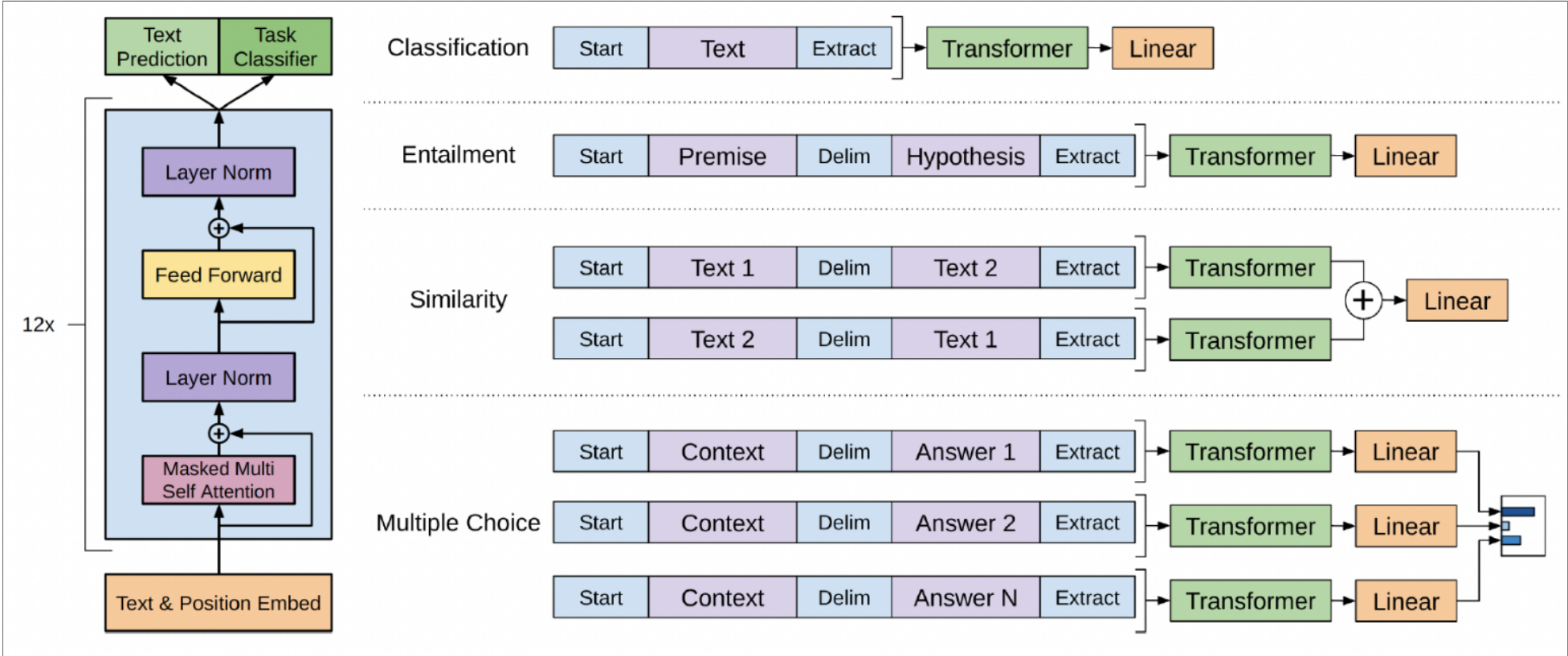
Refer to "W&B's LLM Whitepaper." 24 Apr. 2023, wandb.ai/site/llm-whitepaper. and **Building a GPT from scratch by Andrej Karpathy for more details.**

Generative Pre-training

- Model weights are randomly initialized
- They are trained from scratch on a unlabeled dataset
- Task is to predict the next token, given a sequence of tokens (upto context window)

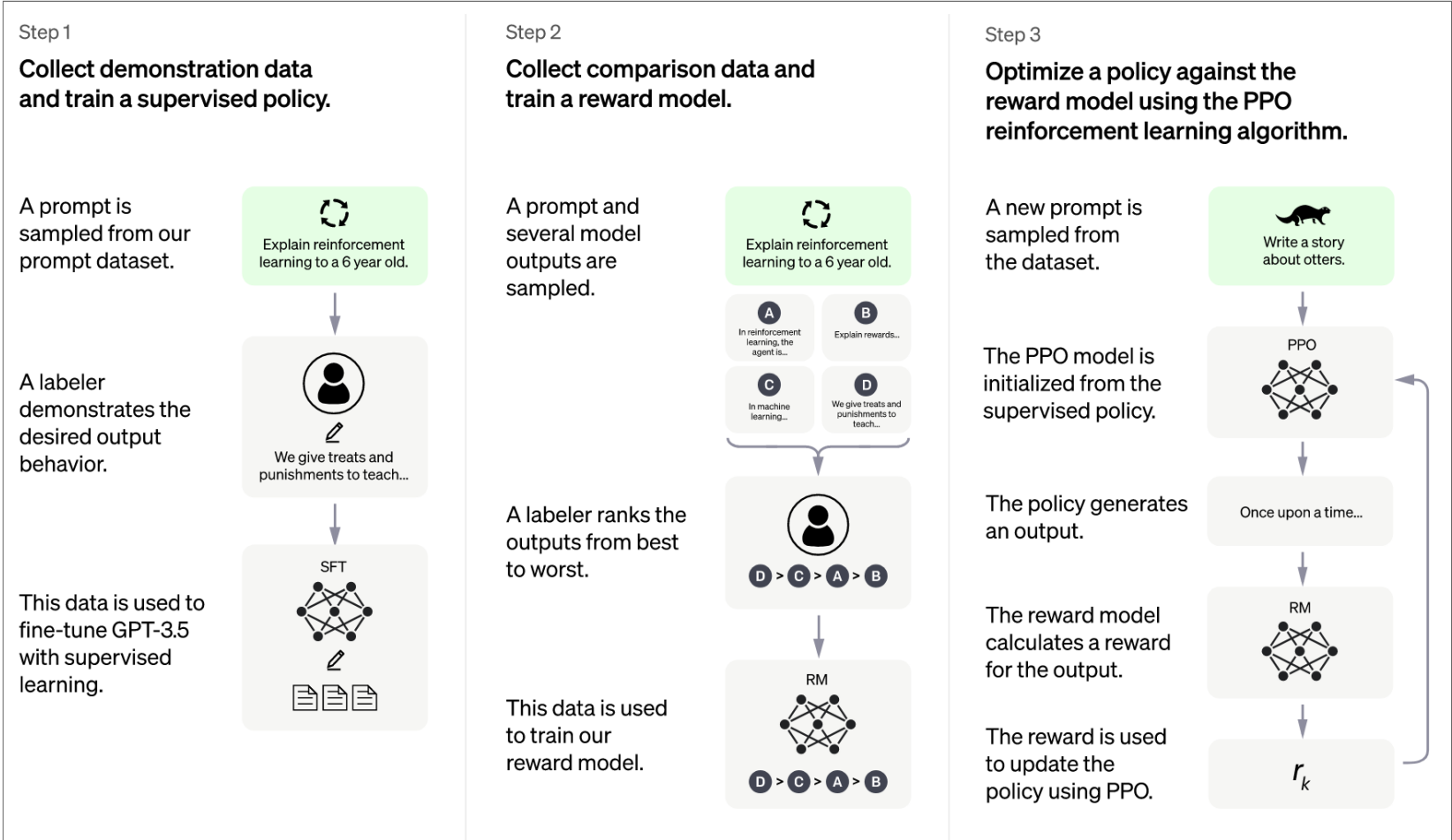
$$L_1(U) = \sum_i \log P(u_i | u_{i-k}, \dots, u_{i-1}; \theta)$$

Supervised Fine Tuning



- Weights from previous step are further fine-tuned on labeled data
- Data comprising of different tasks is fed to the model
- Humans annotators may also be used to refine the output of Generative Pre-trained model for further supervision

RL from human feedback



"Introducing ChatGPT." 24 Apr. 2023

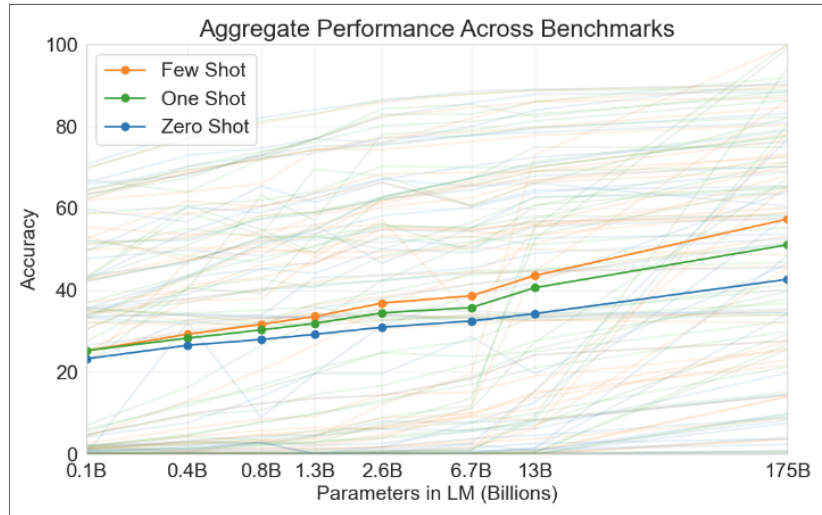
In-context learning

e.g. Classification with few-shot prompts:

```
Apple is a fruit  
Banana is a fruit  
Broccoli is a vegetable  
Carrot is a  
-----  
vegetable
```

- Giving examples in the prompts helps the model learn "in-context"
- A "shot" is a sample data point in the prompt
- You give the model the input and the expected output
- You can give n-number of shots but you are limited by the context window

In-context learning vs. Fine Tuning vs. Training



[1]

Training from scratch is a capital intensive process, but fine-tuning and in-context learning can help alleviate that and help us use LLMs in a cost-effective manner. A few strategies for optimal performance:

- First, try zero-shot.
- Then few-shot
- Then fine-tune, since due to context length limitations, we can't pass the entire dataset as a prompt
- Finally pre-train from scratch.

[1] **Brown, Tom B., et al. "Language Models are Few-Shot Learners." arXiv, 28 May. 2020, doi:10.48550/arXiv.2005.14165.**

What changes now in NLP

- Earlier you'd train a separate NLP model for each task, e.g. one for classification, one for summarization etc.
 - This had task specific guarantees of input and output formats
- Now the task is part of the prompt. The instruction of the task is itself in the input to the model!
 - a.k.a. Instruction Tuning; e.g. "summarize this for me", "apple: fruit; lettuce: vegetable; tomato:"

The Design of Prompts

| Standard Prompting | Chain-of-Thought Prompting |
|---|---|
| <p>Model Input</p> <p>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?</p> <p>A: The answer is 11.</p> <p>Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?</p> | <p>Model Input</p> <p>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?</p> <p>A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.</p> <p>Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?</p> |
| <p>Model Output</p> <p>A: The answer is 27. ❌</p> | <p>Model Output</p> <p>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. ✅</p> |

[2]

- Prompt Engineering: designing how to structure the text to get the best results from the model
- Zero shot CoT; e.g. Just adding "Let's think step by step" to prompt improves accuracy from 17.7 to 78.7%!

[1]

- Ever growing experimentative and trial-and-error field.

[1] Kojima, Takeshi, et al. "Large Language Models are Zero-Shot Reasoners." arXiv, 24 May. 2022, doi:10.48550/arXiv.2205.11916.

[2] Wei, Jason, et al. "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models." arXiv, 28 Jan. 2022, doi:10.48550/arXiv.2201.11903.

Challenges of LLMs

The major challenges are around cost, latency, quality (truthfulness of output), legal constraints (copyright)

Costs

- TFLOPs: tera Floating point operations per second
 - Tesla V100 (p3 EC2) = 125 TFLOPs
 - Average laptop = 20 GLOPs
- GPT-4 training: $2.1e+13$ TFLOPs
- LLaMA 65B trained on 2,048 GPUs with 80 GB RAM for 21 days
 - NVIDIA A100 \$3.93/hour = total approx. 4.05MM USD
- Optimisitc estimate of running ChatGPT, \$100K/day (via [@tomgoldsteincs](#); Prof. at UMD)

Latency

Latency increases as no. of output sequences increase. e.g. gpt-3.5-turbo latencies below:


| # tokens | p50 latency (sec) | p75 latency | p90 latency |
|--------------------------------------|-------------------|-------------|-------------|
| input: 51 tokens, output: 1 token | 0.58 | 0.63 | 0.75 |
| input: 232 tokens, output: 1 token | 0.53 | 0.58 | 0.64 |
| input: 228 tokens, output: 26 tokens | 1.43 | 1.49 | 1.62 |

"Building LLM applications for production." 17 Apr. 2023, huyenchip.com/2023/04/11/llm-engineering.html#finetuning_with_distillation.

Quality of output

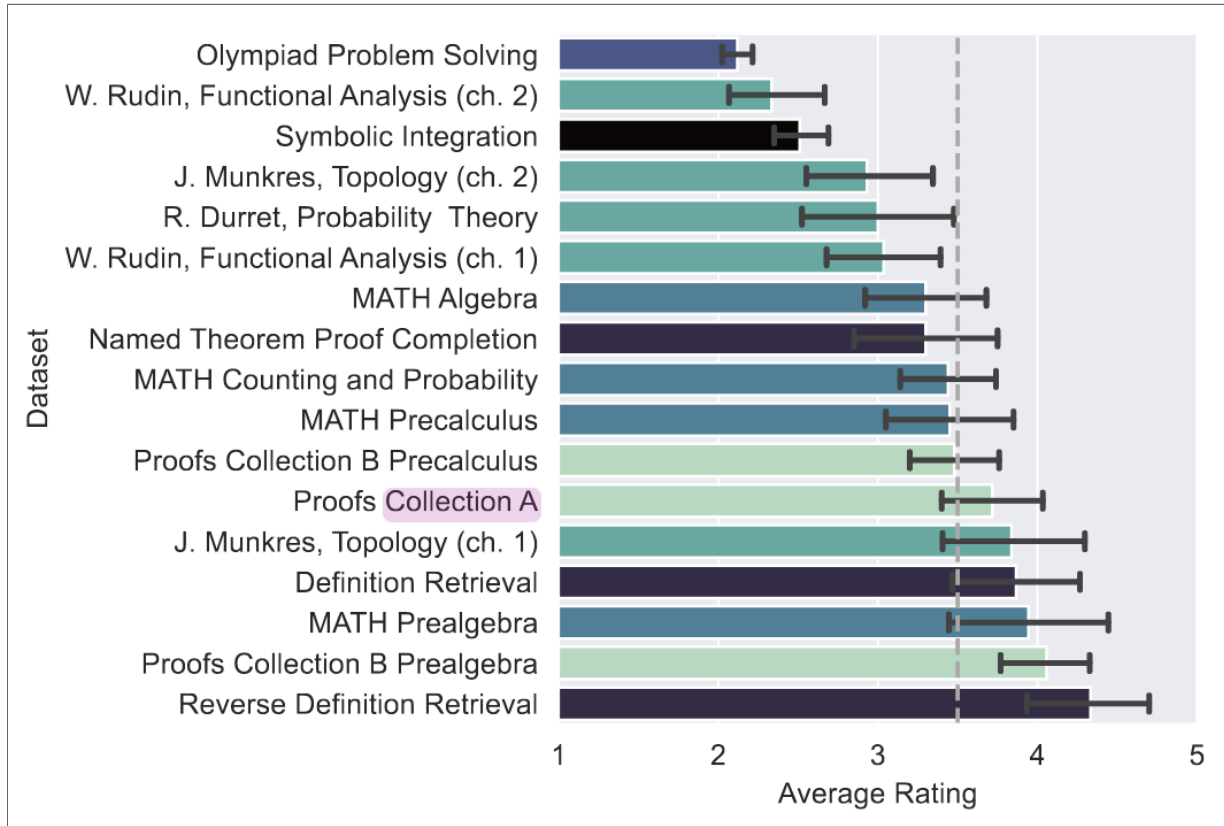
- Content generated can be totally wrong, useless, repetitive or harmful. a.k.a "AI hallucination"
 - 41% truthful on the TruthfulQA dataset
- LLMs are great at pretending to think!
- No guarantees of faithfulness to logic
- e.g. ChatGPT summarizing a non-existent NYT article

PA summarise this article <https://www.nytimes.com/2023/03/11/technology/chatgpt-prompts-to-avoid-content-filters.html>

 The article discusses how ChatGPT, an AI language model, is being used to circumvent content filters that are designed to detect and block certain types of content. By generating prompts that are less likely to trigger these filters, ChatGPT can be used to create content that might otherwise be censored. While this technology could be useful for those seeking to express themselves freely, it also raises concerns about the potential misuse of AI-generated content, as well as the difficulty of regulating such content in the future. The article concludes by noting that there is still much to learn about the impact of AI language models like ChatGPT on the way we communicate online.

"Aligning language models to follow instructions." 4 May. 2023, openai.com/research/instruction-following#fnref4.

Can LLMs win the Math Olympiad?



- Very good at retrieving definitions
- Not so much at solving Olympiad questions
- Math Word Problems (MWP) failure rate at 84% (without showing work)

Frieder, Simon, et al. "Mathematical Capabilities of ChatGPT." arXiv, 31 Jan. 2023, doi:10.48550/arXiv.2301.13867.

Shakarian, Paulo, et al. "An Independent Evaluation of ChatGPT on Mathematical Word Problems (MWP)." arXiv, 23 Feb. 2023, doi:10.48550/arXiv.2302.13814.

Legal

- Copyright
- Data rights enforcement

Use Cases

From the InstructGPT paper, this gives us an idea of use-cases handled by the ChatGPT API. Open QA is not restricted to a domain, which Closed QA is. (e.g. legal, medical docs)

Table 1: Distribution of use case categories from our API prompt dataset.

| Use-case | (%) |
|----------------|-------|
| Generation | 45.6% |
| Open QA | 12.4% |
| Brainstorming | 11.2% |
| Chat | 8.4% |
| Rewrite | 6.6% |
| Summarization | 4.2% |
| Classification | 3.5% |
| Other | 3.5% |
| Closed QA | 2.6% |
| Extract | 1.9% |

Ouyang, Long, et al. "Training language models to follow instructions with human feedback." arXiv, 4 Mar. 2022, doi:10.48550/arXiv.2203.02155.

Content Policy Automation

e.g. Use case: filter out dating advice.

Given a text, tell me if it contains advice on dating.
Answer as "yes" or "no"

Text: ""
Don't talk about religion on the first date.
""

Result: Yes .

Childproofing

Given a text, tell me if it is suitable for children. Answer as "yes" or "no"

Text:

""

Drugs are the best.

""

No.

More Examples

Note: These examples are generated using the [Vicuna 13B model chat interface](#)

Given a text, tell me if it is suitable for children. Answer as "yes" or "no"

Text:

```
""""  
I found a teddy bear and I killed it with a knife.  
""""
```

No.

Given a text, tell me if it is suitable for children. Answer as "yes" or "no"

Text:

```
""""  
I found a teddy bear and it was cute.  
""""
```

Yes.

Training data generation

Note: for illustrative purposes only

Generating diverse datasets similar to existing data for improving text models:

Give me more examples similar to these:

Text:

I hate Indians

Text:

I hate Americans

Text:

In practice

- Add human checks to ensure quality when using LLMs to collect training data.
- Highly efficient as compared to labeling all data to collect true positives in sparse categories
- There might be some overfitting problems due to the data being too similar

Future

- Open research topic
- There are several models fine-tuned to generate toxic content, erotica etc. [1]
- We can use them to improve detection of such content

[1] Crataco. "ai-guide." GitHub, 22 Apr. 2023, github.com/Crataco/ai-guide/blob/main/guide/original.md.

Resources

Thank you!

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., ... Fedus, W. (2022). Emergent Abilities of Large Language Models. Transactions on Machine Learning Research. Retrieved from

<https://openreview.net/forum?id=yzkSU5zdwD>

Sevilla, Jaime, et al. "Compute Trends Across Three Eras of Machine Learning." arXiv, 11 Feb. 2022, doi:10.48550/arXiv.2202.05924.; [chart url](#)

["AI Readiness Report 2023 | Scale AI." ScaleAI, 21 Apr. 2023, scale.com/ai-readiness-report.](#)

[BertViz - Visualize Attention in NLP Models](#)

[Schulman, John, et al. "Proximal Policy Optimization Algorithms." arXiv, 20 July 2017,](#)

[doi:10.48550/arXiv.1707.06347.](#)

[Holtzman, Ari, et al. "The Curious Case of Neural Text Degeneration." arXiv, 22 Apr. 2019,](#)

[doi:10.48550/arXiv.1904.09751.](#)

[Holtzman, Ari, et al. "The Curious Case of Neural Text Degeneration." arXiv, 22 Apr. 2019,](#)

[doi:10.48550/arXiv.1904.09751.](#)

[Kojima, Takeshi, et al. "Large Language Models are Zero-Shot Reasoners." arXiv, 24 May. 2022,](#)

[doi:10.48550/arXiv.2205.11916.](#)

[Wei, Jason, et al. "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models." arXiv, 28 Jan. 2022, doi:10.48550/arXiv.2201.11903.](#)

["Ask HN: Is prompt engineering just snake oil? | Hacker News." 22 Apr. 2023, news.ycombinator.com/item?id=35665168.](#)

[Saravia, E. \(12 2022\). Prompt Engineering Guide. <https://github.com/Dair-AI/Prompt-Engineering-Guide>.](#)

["Best practices for prompt engineering with OpenAI API | OpenAI Help Center." 22 Apr. 2023,](#)

[help.openai.com/en/articles/6654000-best-practices-for-prompt-engineering-with-openai-api.](#)

[Brown, Tom B., et al. "Language Models are Few-Shot Learners." arXiv, 28 May. 2020,](#)

[doi:10.48550/arXiv.2005.14165.](#)

Ouyang, Long, et al. "Training language models to follow instructions with human feedback." arXiv, 4 Mar. 2022, doi:10.48550/arXiv.2203.02155.

Crataco. "ai-guide." GitHub, 22 Apr. 2023, github.com/Crataco/ai-guide/blob/main/guide/original.md.

"Stanford CRFM." 13 Apr. 2023, crfm.stanford.edu/2023/03/13/alpaca.html.

Building a GPT from scratch by Andej Karpathy [[YouTube](#)]

Speaker notes