Introduction to Deep Learning

Lecture 20
Large Language Models

Roshan Sharma

Some slides borrowed from Danqi Chen, Chenyan Xiong and Graham Neubig – thanks!

11-785, Spring 2024

Agenda

- Emergent Abilities and Scaling Effects
- What are LLMs?
- Modern LLM Architecture
- LLM Training Procedure
- LLM Inference – Prompting, In-Context Learning and Chain of Thought
- Evaluating LLMs
- Multimodal LLMs

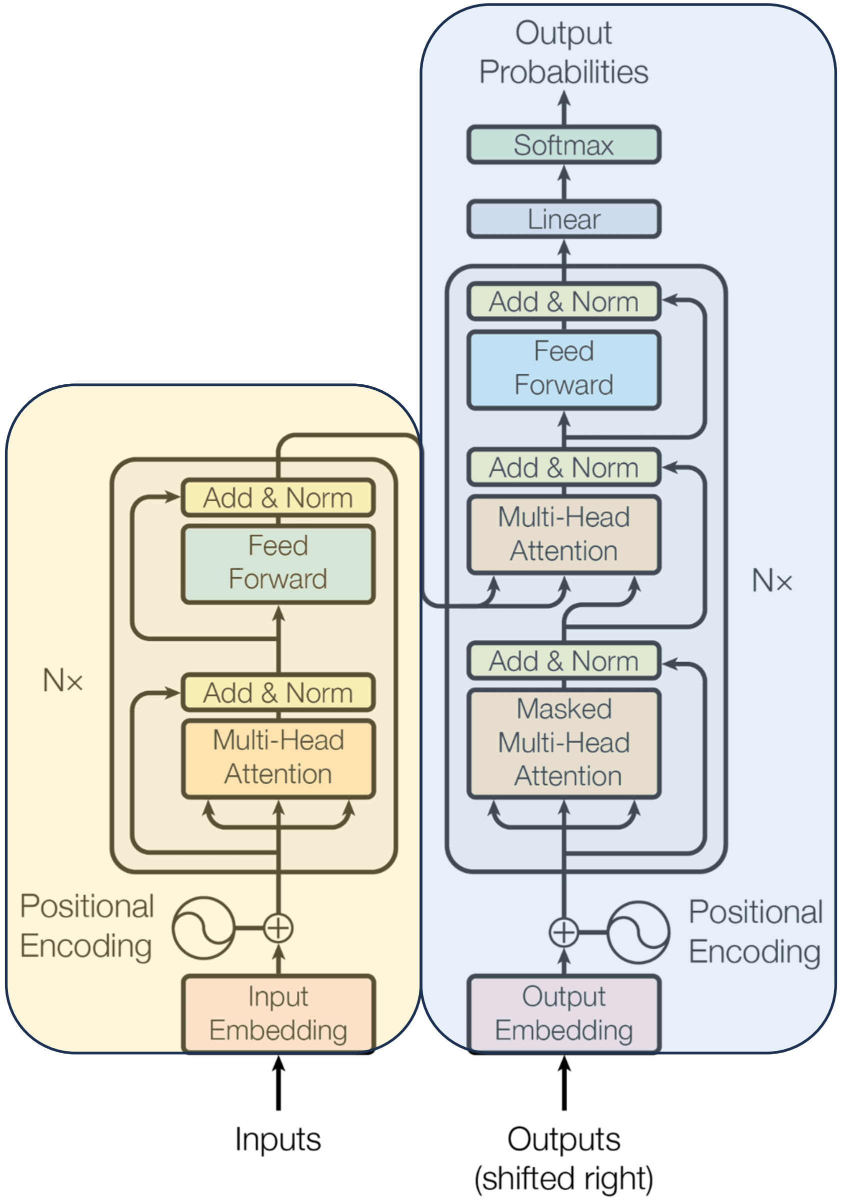
Review: Language Models as Generalists

- Language models can be used to not just perform a single task, but multiple tasks by learning to predict the next token or sentence

Review: The LLM Era – Paradigm Shift in Machine Learning

BERT
Oct 2018

Representation



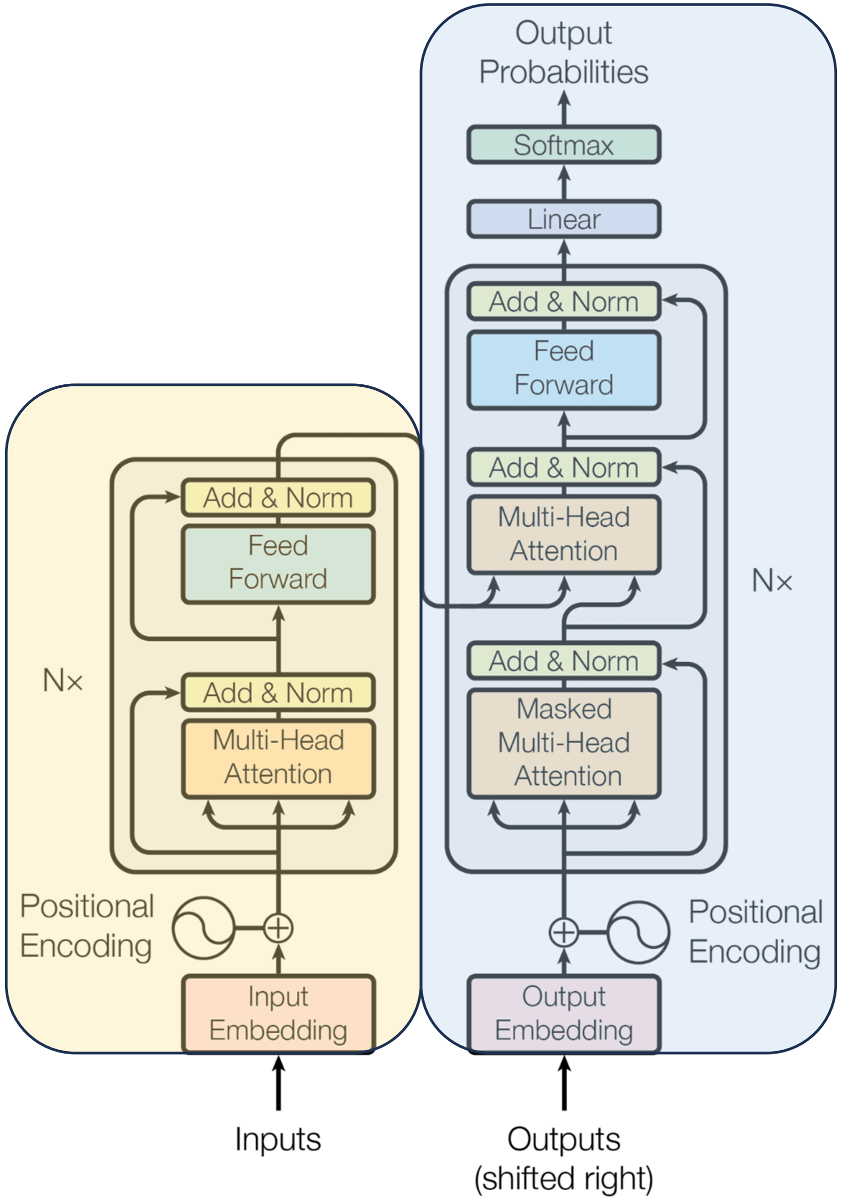
GPT
Jun 2018

Generation

Review: The LLM Era – Paradigm Shift in Machine Learning

BERT
Oct 2018

Representation



GPT
Jun 2018

Generation

GPT 2 – Generalizing to Unseen Tasks

- LMs can be used for different tasks by pre-training a “base” model and then fine-tuning for the task(s) of interest

- Practical Issues:
  - Too many copies of the model
  - Need for large-scale labeled data for fine-tuning
  - Can do only specific task

GPT 2 – Generalizing to Unseen Tasks

- LMs can be used for different tasks by pre-training a “base” model and then fine-tuning for the task(s) of interest

- Practical Issues:
  - Too many copies of the model
  - Need for large-scale labeled data for fine-tuning

- Multi-task Training?
  - Data remains a challenge
  - Humans don’t need such large volumes of data to learn – can we do better?

- Train a model that can perform NLP tasks in a zero-shot manner

GPT 2 – Task Specifications

- Primary shift comes from modeling assumptions from single-task to general model

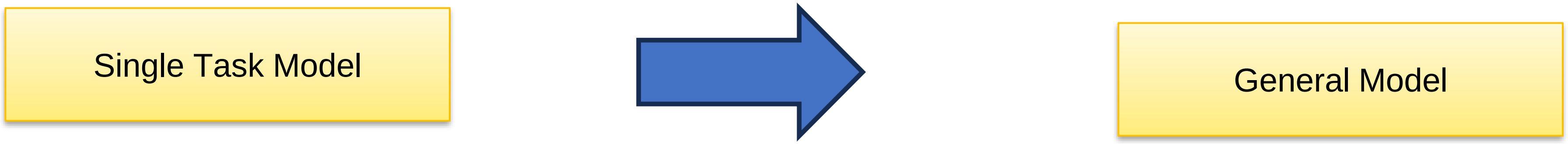
Single Task Model → General Model

P(output | input)

P(output | input,**task**)

GPT 2 – Task Specifications

- Primary shift comes from modeling assumptions from single-task to general model



P(output | input)

P(output | input,**task**)

- Task descriptions may be provided as text – for example, translate this French text to English

GPT 2 – what makes such an LM work ?

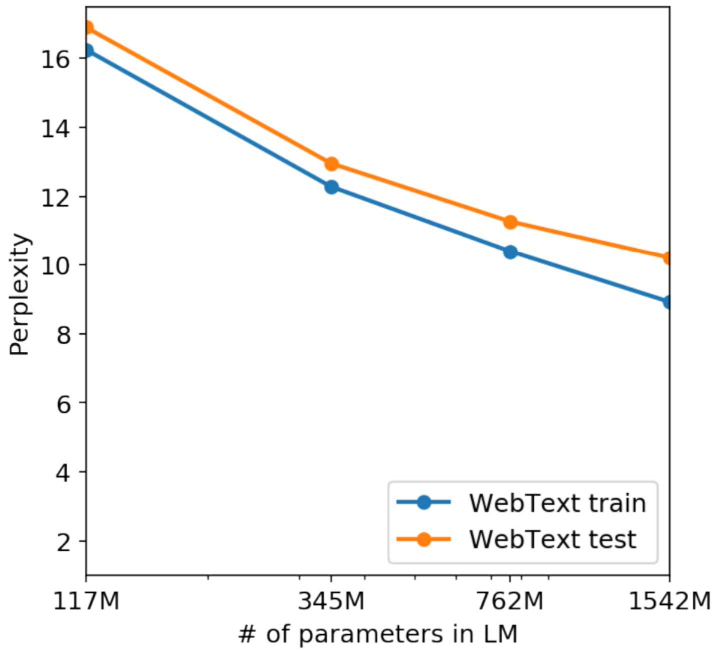
- Diverse training data
  - Model can do many disparate tasks with no training at all!

- Scaling model capacity and data

	LAMBADA (PPL)	LAMBADA (ACC)	CBT-CN (ACC)	CBT-NE (ACC)	WikiText2 (PPL)	PTB (PPL)	enwik8 (BPB)	text8 (BPC)	WikiText103 (PPL)	1BW (PPL)
SOTA	99.8	59.23	85.7	82.3	39.14	46.54	0.99	1.08	18.3	**21.8**
117M	**35.13**	45.99	**87.65**	**83.4**	**29.41**	65.85	1.16	1.17	37.50	75.20
345M	**15.60**	55.48	**92.35**	**87.1**	**22.76**	47.33	1.01	**1.06**	26.37	55.72
762M	**10.87**	**60.12**	**93.45**	**88.0**	**19.93**	**40.31**	**0.97**	**1.02**	22.05	44.575
1542M	**8.63**	**63.24**	**93.30**	**89.05**	**18.34**	**35.76**	**0.93**	**0.98**	**17.48**	42.16

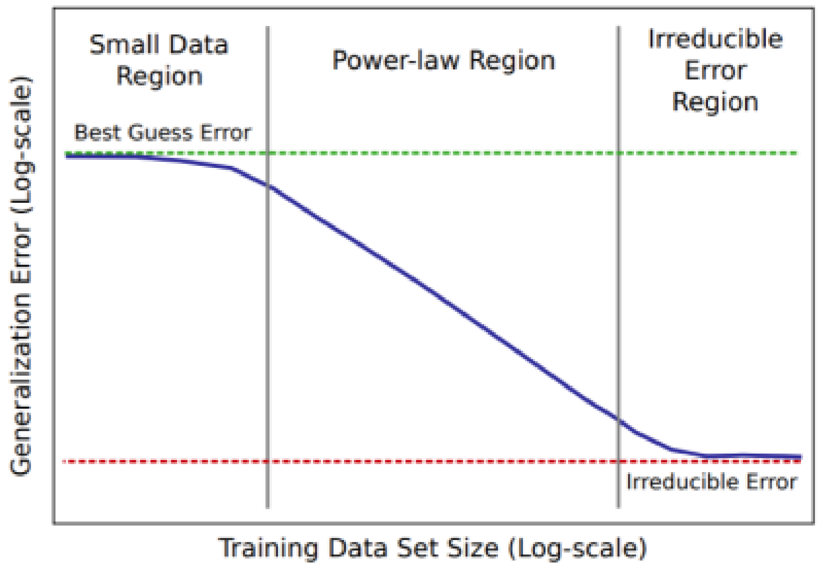
Scaling in GPT-2

- Scaling improves the perplexity of the LM and improves performance



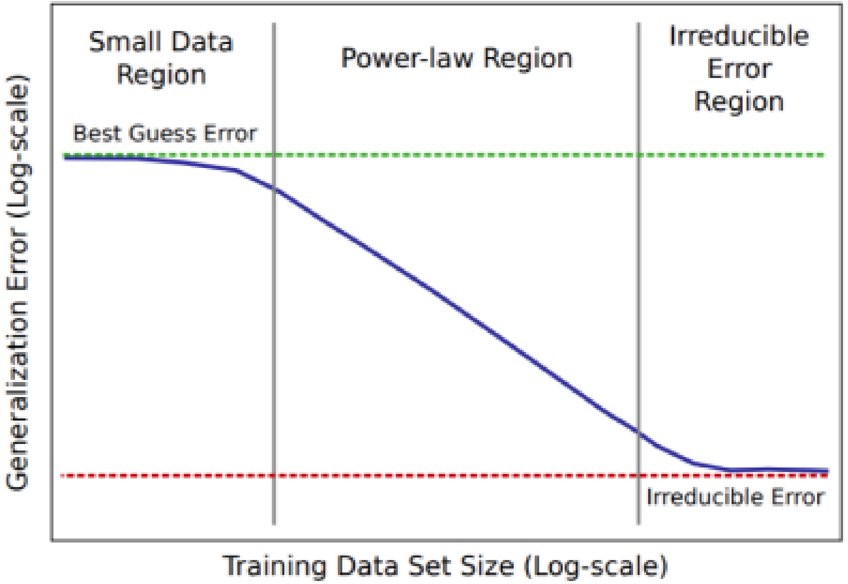
Why is this interesting? Look at data scaling

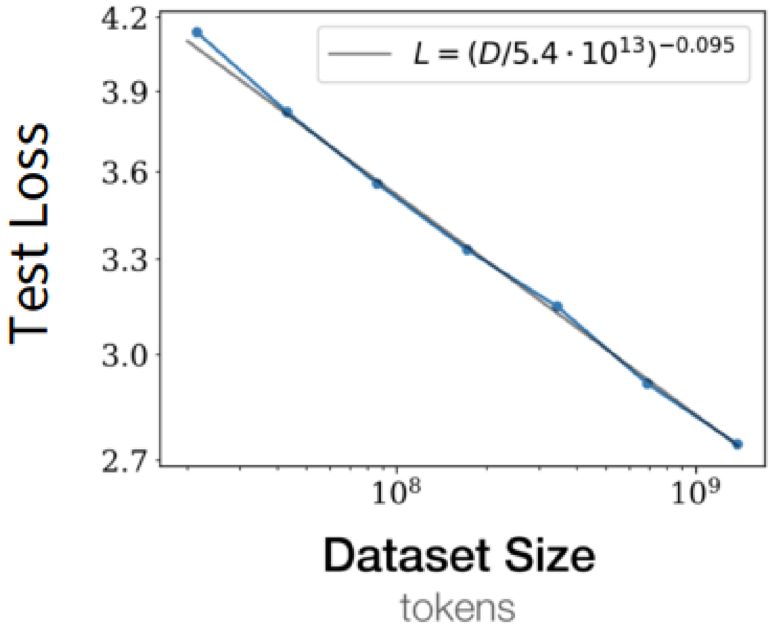
- We know that typical scaling effects look like this when we increase the amount of training data



Why is this interesting? Look at data scaling

- Loss and dataset size is linear on a log-log plot
- This is “power-law scaling”





Scaling - (Kaplan,2020)

- Can we understand scaling by positing scaling laws ?

- With scaling laws, we can make decisions on architecture, data, hyperparameters by training smaller models

- Open AI Study : **Scaling Laws for Neural Language Models** (Kaplan et al. 2020)

Scaling - (Kaplan,2020)

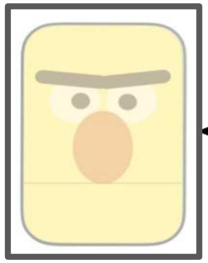
- Open AI Study : **Scaling Laws for Neural Language Models** (Kaplan et al. 2020)

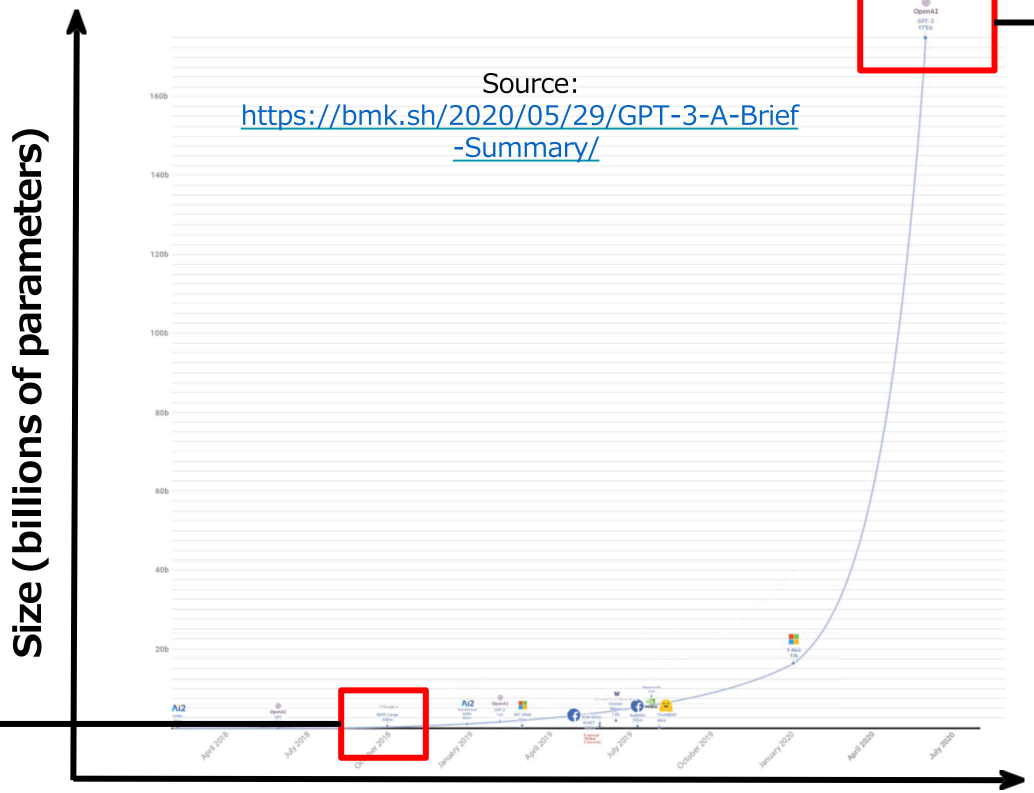
- Key Findings:
  - Performance depends strongly on scale, and weakly on the model shape
  - Larger models are more sample-efficient
  - Smooth power laws ($y = ax^k$) b/w empirical performance & N - parameters, D - dataset size, C - compute

Scaling Effects

- The effect of some hyperparameters on big LMs can be predicted before training – optimizer (Adam v/s SGD), model depth, LSTM v/s Transformer
- Idea:
  - Train a few smaller models
  - Establish a scaling law (e.g. ADAM vs SGD scaling law)
  - Select optimal hyper param based on the scaling law prediction

Model Scaling: GPT-3





175b params!
GPT-2 was 1.5b

Emergent Abilities with GPT-3 – Wei et. al 2022

- Emergent abilities:
  - not present in smaller models but is present in larger models
  - Do LLMs like GPT3 have these ?

- Findings:
  - GPT-3 trained on text can do arithmetic problems like addition and subtraction
  - Different abilities “emerge” at different scales

Emergent Abilities with GPT-3 – Wei et. al 2022

- Emergent abilities:
  - not present in smaller models but is present in larger models
  - Do LLMs like GPT3 have these ?

- Findings:
  - GPT-3 trained on text can do arithmetic problems like addition and subtraction
  - Different abilities “emerge” at different scales

Emergent Abilities with GPT-3 – Wei et. al 2022

- Emergent abilities:
  - not present in smaller models but is present in larger models
  - Do LLMs like GPT3 have these ?

- Findings:
  - GPT-3 trained on text can do arithmetic problems like addition and subtraction
  - Different abilities "emerge" at different scales
  - Model scale is not the only contributor to emergence – for 14 BIG-Bench tasks, LaMDA 137B and GPT-3 175B models perform at near-random, but PaLM 62B achieves above-random performance
  - Problems LLMs can't solve today may be emergent for future LLMs

Large Language Models

- Language models that have many parameters (over 1B) and can perform multiple tasks through prompting

- Eg. GPT, Llama2, Gemini, PaLM, Mistral, Mixtral etc.

LLM Realization - Architecture

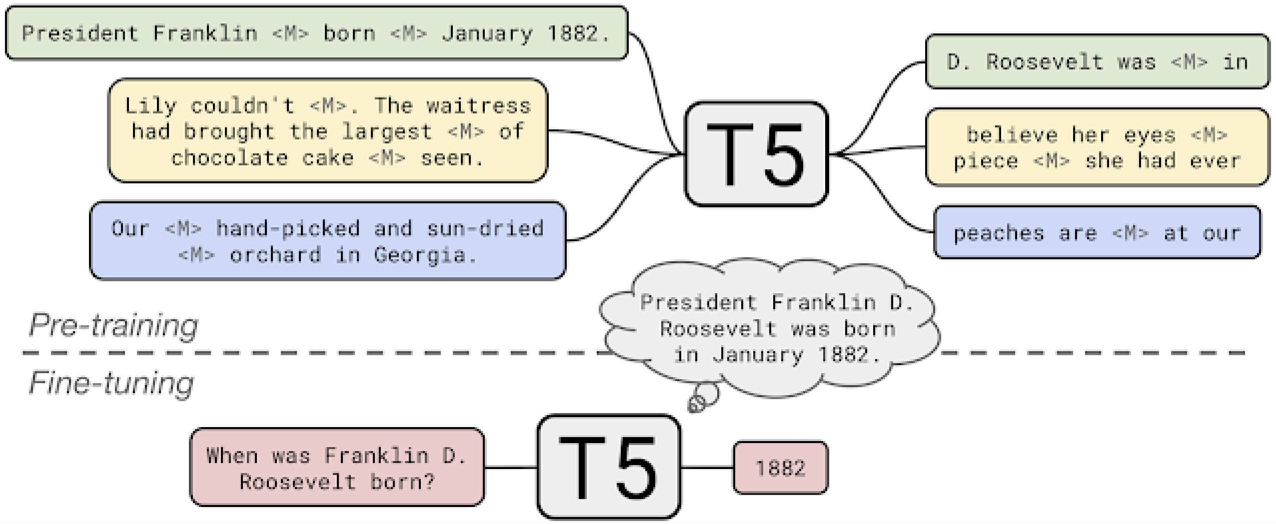
- Encoder-only (BERT)
  - Pre-training : Masked Language Modeling (MLM)
  - Great for classification tasks, but hard to do generation

- Decoder-only (GPT)
  - Pre-training: Auto-regressive Language Modeling
  - Stable training, faster convergence
  - Better generalization after pre-training

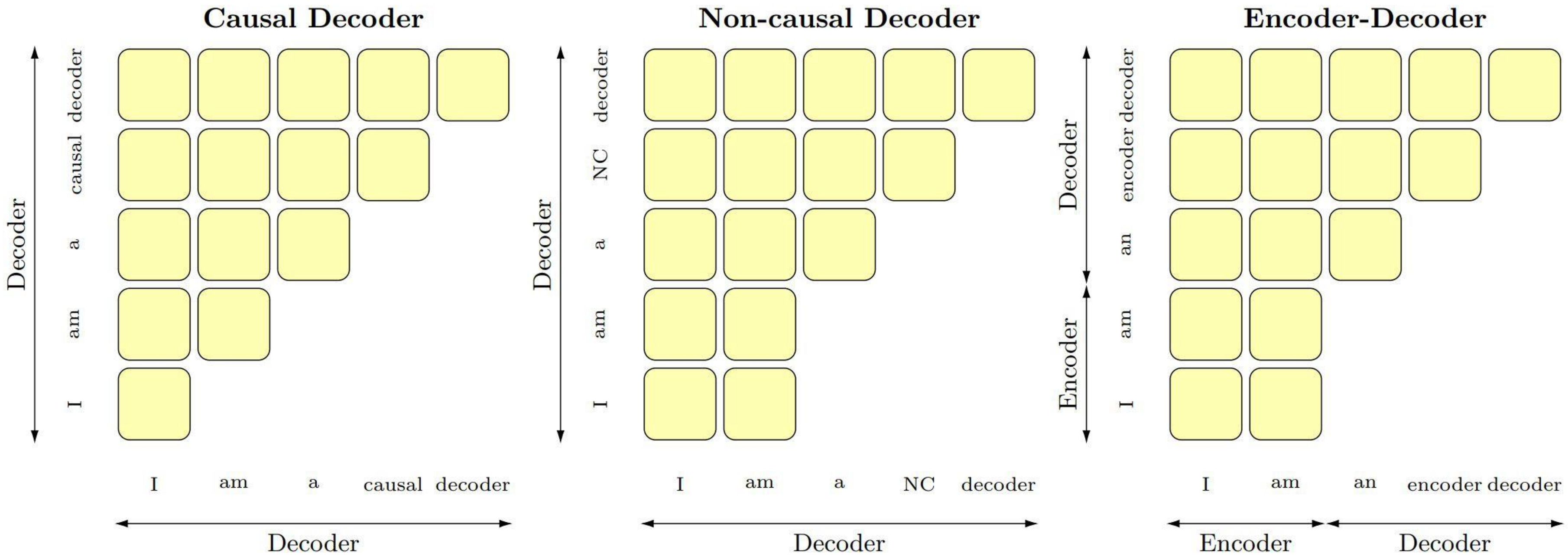
- Encoder-decoder (T0/T5)
  - Pre-training : Masked Span Prediction
  - Good for tasks like MT, summarization

[T5/ T0]: Masked Span Prediction

- Masked span prediction involves:
  - Mask continuous set of tokens (span) in input
  - Predict this masked span from the decoder



Attention patterns (Wang et. al)



- Causal decoder -- each token attends to the previous tokens only.
- In both non-causal decoder and encoder-decoder, attention is allowed to be bidirectional on any conditioning information.
- For the encoder-decoder, that conditioning is fed into the encoder part of the model.

Empirical Observations (Wang et. al)

- Decoder-only models outperform encoder-decoder models using similar configuration

	EAI-EVAL	T0-EVAL
Causal decoder	**44.2**	**42.4**
Non-causal decoder	43.5	41.8
Encoder-decoder	39.9	41.7
Random baseline	32.9	41.7

Llama 2 Architecture (Ouyang et. al.)

- Decoder-only model
- Changes in transformer module:
    - Norm after sublayer -> Norm before sublayer
    - LayerNorm -> RMSNorm for stability
    - Activation: ReLU -> SwiGLU(x) = Swish(xW)xV = xWSigmoid(AxW)xV
    - Position Embedding: Absolute/Relative -> RoPE (Rotary PE)
    - Long contexts : Multi-head attention -> Grouped-query attention

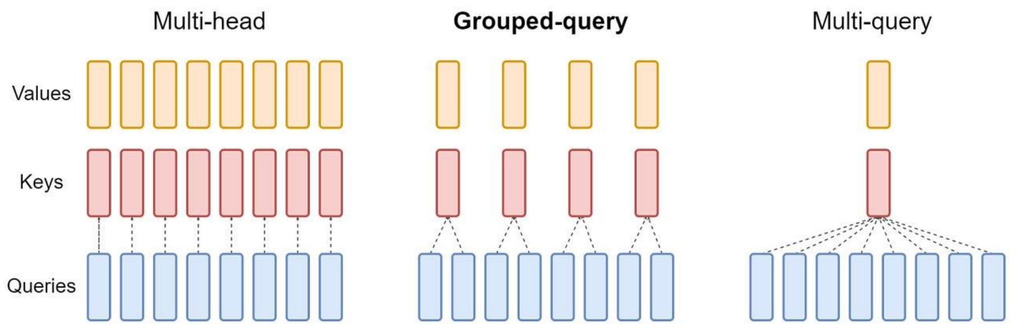


Figure 2: Overview of grouped-query method. Multi-head attention has H query, key, and value heads. Multi-query attention shares single key and value heads across all query heads. Grouped-query attention instead shares single key and value heads for each *group* of query heads, interpolating between multi-head and multi-query attention.

Poll 1

Which of the following is true about emergent abilities?

A. A language model with fewer parameters than 175B cannot have any emergent abilities
B. They are found in large models but not in small models
C. Summarization is likely an emergent ability in a model pre-trained on a summarization corpus
D. Emergent abilities arise only because of scaling.

Poll 1

Which of the following is true about emergent abilities?

A. A language model with fewer parameters than 175B cannot have any emergent abilities
B. They are found in large models but not in small models
C. Summarization is likely an emergent ability in a model pre-trained on a summarization corpus
D. Emergent abilities arise only because of scaling.

Training of Decoder-only LLMs – Llama 2

1. Auto-regressive Pre-training - Train to predict the next token on very large-scale corpora (~3 trillion tokens)

Training of Decoder-only LLMs – Llama 2

1. Auto-regressive Pre-training - Train to predict the next token on very large scale corpora (~3 trillion tokens)
2. Instruction Fine-tuning/ Supervised Fine-tuning (SFT) - Fine-tune the pre-trained model with pairs of (instruction+input,output) with large dataset and then with small high-quality dataset

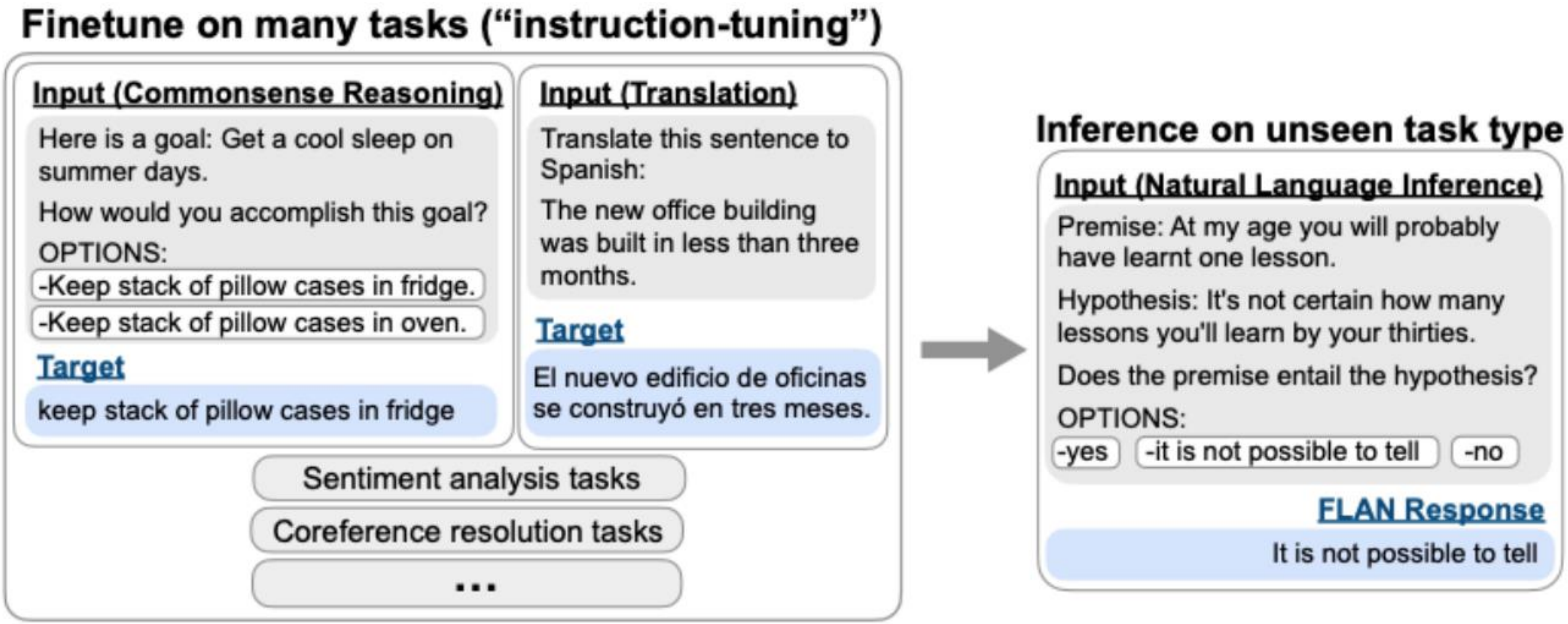
Instruction fine-tuning provides as a prefix a natural language description of the task along with the input.

- E.g. Translate into French this sentence: my name is -> je m'appelle

Supervised Fine-tuning versus Pre-training

- Objective function
  - Loss computed only for target tokens in SFT, all tokens are targets in pre-training
- Input and Target
  - Instruction + input as input with the target in SFT and only input as input with shifted input as target
- Purpose
  - Pre-training makes good generalist auto-completes but good SFT builds models that can do many unseen tasks
  - SFT can also guide nature of outputs in terms of safety and helpfulness

Instruction Tuning (Wei et. al. 2021)



Unsafe Outputs – Alignment Problem

- LLMs may produce
  - Harmful text – unparliamentary language, bias and discrimination
  - Text that can cause direct harm – allowing easy access to dangerous information
- Therefore, LLMs should be trained to produce outputs that align with human preferences and values
- Modern LLMs do so by using SFT and by using human preference directly in model training

Training of Decoder-only LLMs – Llama 2

1. Auto-regressive Pre-training - Train to predict the next token on very large scale corpora (~3 trillion tokens)
2. Instruction Fine-tuning/ Supervised Fine-tuning (SFT) - Fine-tune the pre-trained model with pairs of (instruction+input,output) with large dataset and then with small high-quality dataset
3. Safety / RLHF - Design a reward model based on human feedback and use policy gradient methods with the trained reward model to update LLM parameters so that outputs align with human values

RLHF

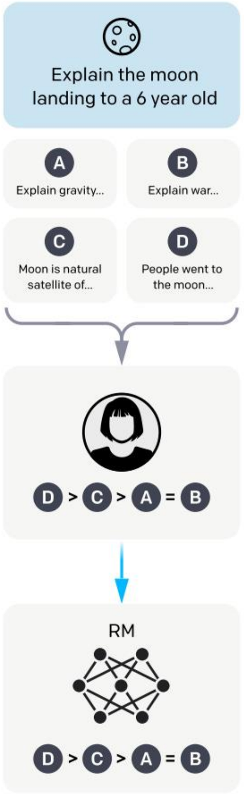
Step 2

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

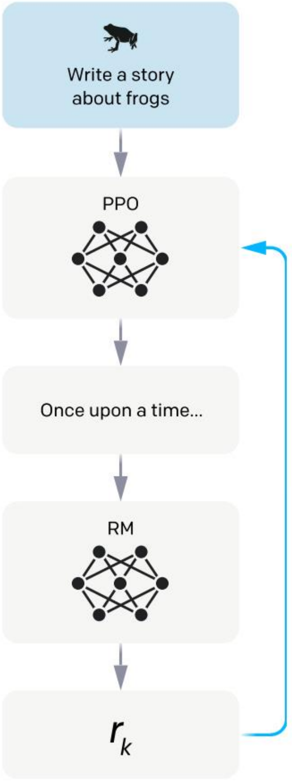
Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.

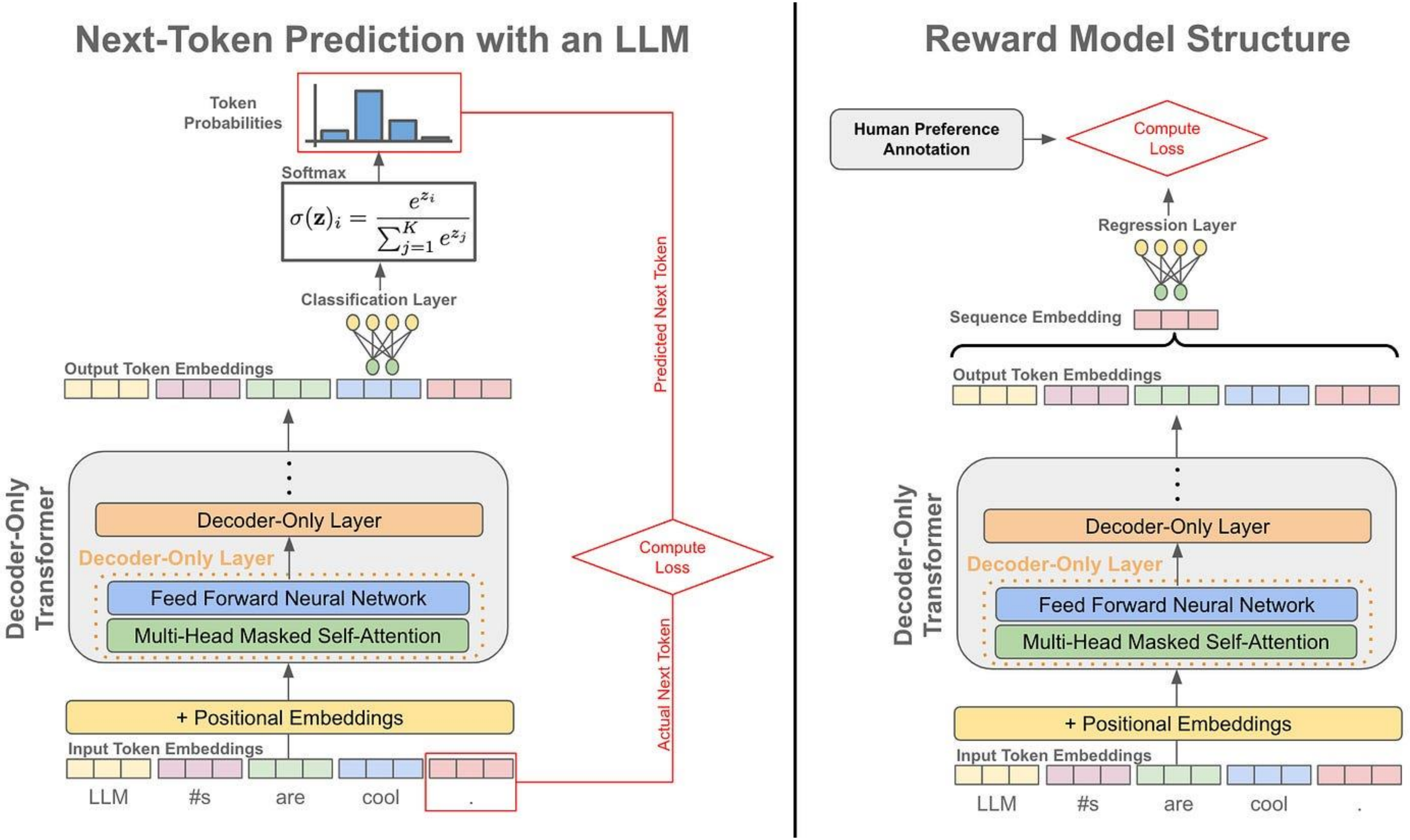
The policy generates an output.

The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.



Model Fine-tuning for RLHF



Note on LLM Safety and Harmfulness

- Does doing RLHF and safety tuning mean LLMs will never produce harmful outputs ?

Note on LLM Safety and Harmfulness

- Does doing RLHF and safety tuning mean LLMs will never produce harmful outputs?
- No! The list of harmful outputs is not exhaustive and very large
- What are the other concerns?
    - Adversarial Robustness – adversaries can force the LLM to produce harmful outputs by attacking the model
- In our experience, Claude produces harmful outputs the least when compared to models like ChatGPT and Llama

Poll 2

Which of the following is a feature of Llama 2?

A. Swishy activations
B. Relativistic positional embeddings
C. Multi-query attention
D. Grouped-query attention

Poll 2

Which of the following is a feature of Llama 2?

A. Swishy activations
B. Relativistic positional embeddings
C. Multi-query attention
D. Grouped-query attention

LLM Inference: Prompting

- Prompts
  - Tell the model what to do in natural language
  - For example, generate a textual summary of this paragraph:
  - Can be as short or long as required

- Prompt Engineering
  - The task of identifying the correct prompt needed to perform a task
  - General rule of thumb be as specific and descriptive as possible
  - Can be manual or automatic (prefix-tuning, paraphrasing etc.)

ChatGPT Prompt example

```
messages=[
  {
    "role": “system",
    "content": "You are an assistant that translates corporate jargon into plain English.”
  },
  {
    “role”: “system”,
    “name":"example_user",
    "content": "New synergies will help drive top-line growth.”
  },
  {
    "role": “system",
    "name": “example_assistant",
    "content": "Things working well together will increase revenue.”
  },
  ...,
  {
    "role": “user",
    "content": "This late pivot means we don't have time to boil the ocean for the client deliverable.”
  },
]
```

In-context learning/ Few-shot prompting (Brown,21)

- Provide a few examples along with the instruction

Instruction	`Please classify movie reviews as 'positive' or ‘negative’.`

Examples	`Input: I really don’t like this movie.`
	`Output: negative`
	
	`Input: This movie is great!`
	`Output: positive`

Chain of thought prompting (Wei, 2021)

- Get the model to work through the steps of the problem

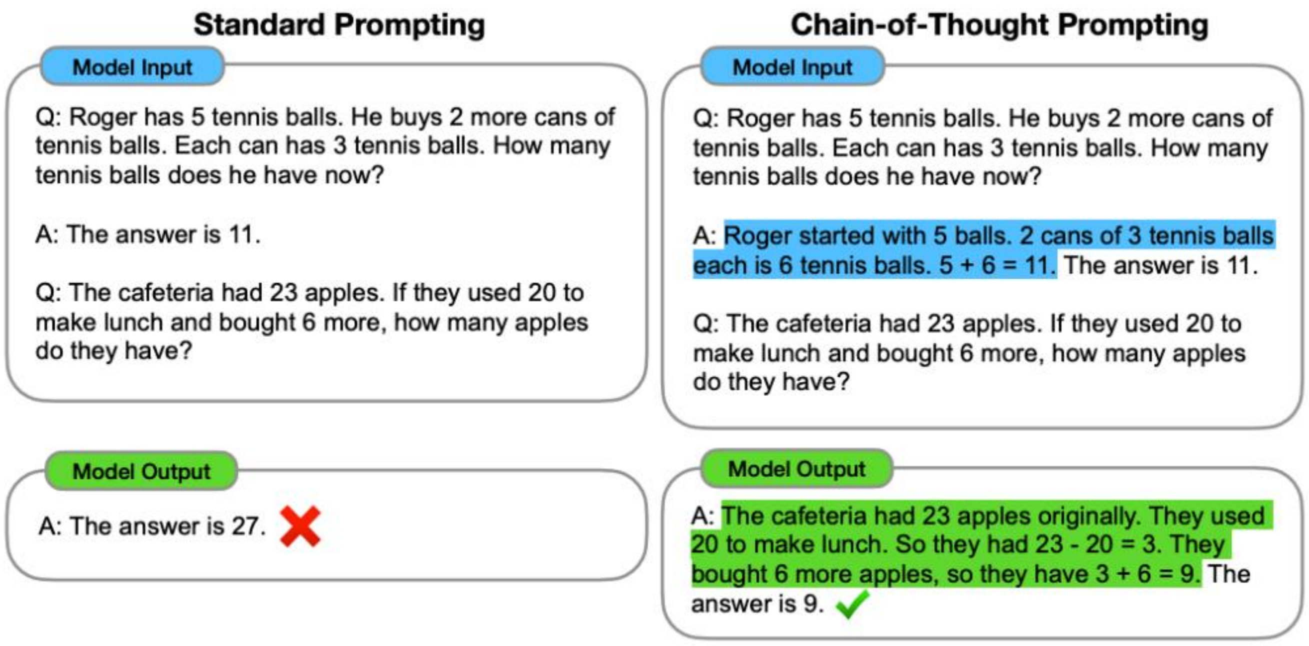
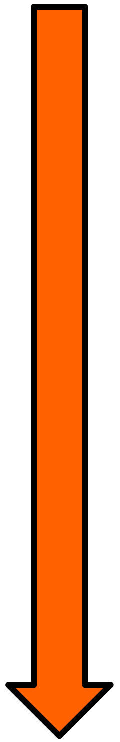


Figure 1: Chain-of-thought prompting enables large language models to tackle complex arithmetic, commonsense, and symbolic reasoning tasks. Chain-of-thought reasoning processes are highlighted.

What to Pick?

Stronger
task-specific
performance

1. Full Fine-tuning (FT)
    a. +Strongest performance
    b. - Need curated and labeled dataset for each new task (typically 1k-100k+ ex.)
    c. - Poor generalization, spurious feature exploitation
2. Few-shot (FS)
    a. +Much less task-specific data needed
    b. +No spurious feature exploitation
    c. - Challenging
3. One-shot (1S)
    a. +"Most natural," e.g. giving humans instructions
    b. - Challenging
4. Zero-shot (0S)
    a. +Most convenient
    b. - Challenging, can be ambiguous



More convenient,
general, less data

Note on Parameter Efficient Fine-tuning

- When we don’t have large enough data for SFT
    - Freeze the LM and keep some parameters trainable (which?)
    - Add an external adapter module to adapt model parameters to the task
    - Perform Low-rank Adaptation (LoRA)

Poll 3

Which of the following describes in-context learning?

A. Providing detailed instructions during RLHF
B. Providing examples within LLM prompts
C. Asking the LLM to show its work
D. Zero-shot prompting

Poll 3

Which of the following describes in-context learning?

A. Providing detailed instructions during RLHF
B. Providing examples within LLM prompts
C. Asking the LLM to show its work
D. Zero-shot prompting

Evaluating LLMs

- Evaluation is challenging
  - Evaluate on as many datasets and tasks as possible

Benchmark (shots)	GPT-3.5	GPT-4	PaLM	PaLM-2-L	LLAMA 2
MMLU (5-shot)	70.0	**86.4**	69.3	78.3	68.9
TriviaQA (1-shot)	–	–	81.4	**86.1**	85.0
Natural Questions (1-shot)	–	–	29.3	**37.5**	33.0
GSM8K (8-shot)	57.1	**92.0**	56.5	80.7	56.8
HumanEval (0-shot)	48.1	**67.0**	26.2	–	29.9
BIG-Bench Hard (3-shot)	–	–	52.3	**65.7**	51.2

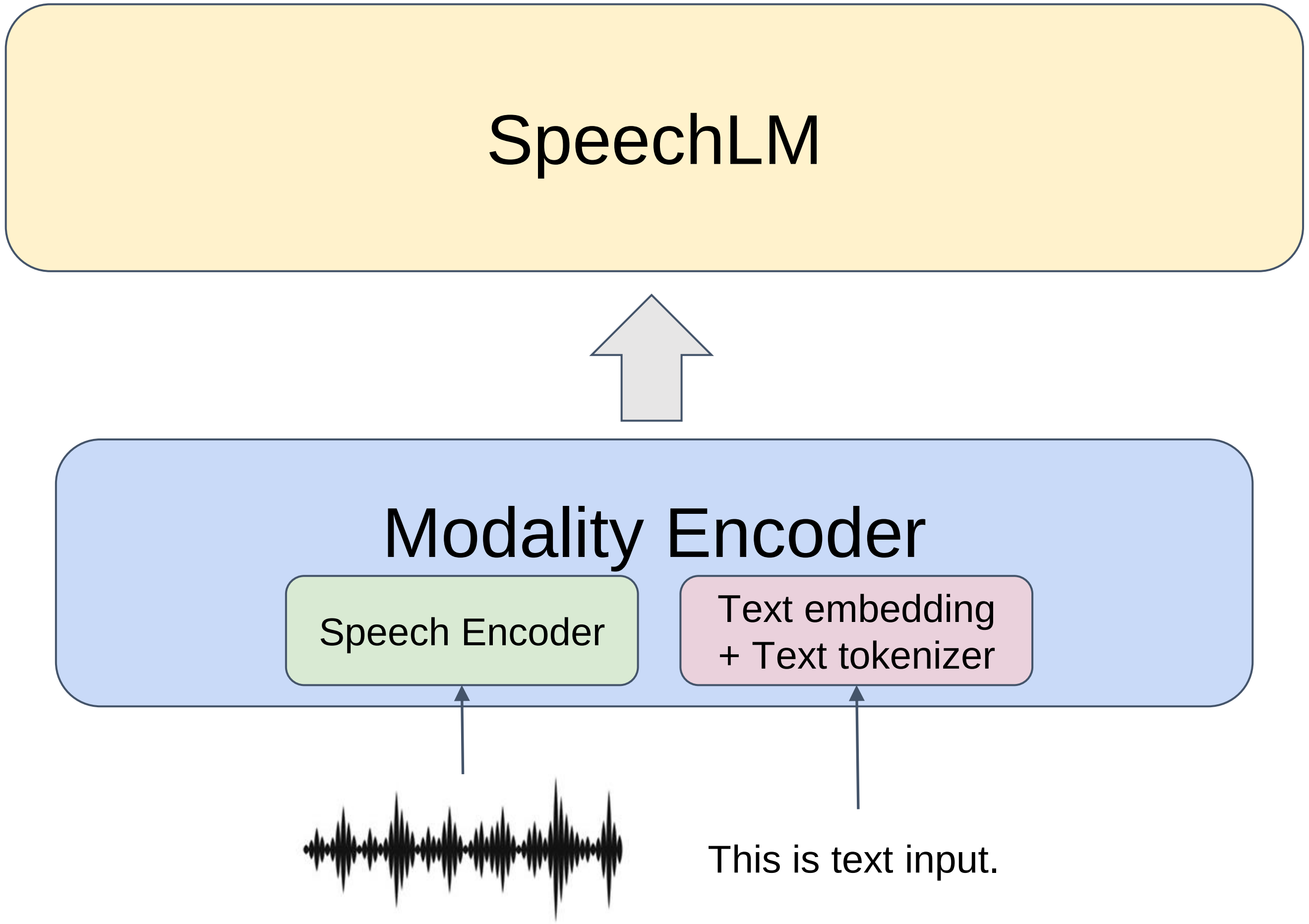
Table 4: Comparison to closed-source models on academic benchmarks. Results for GPT-3.5 and GPT-4 are from OpenAI (2023). Results for the PaLM model are from Chowdhery et al. (2022). Results for the PaLM-2-L are from Anil et al. (2023).

Multimodal LLMs

- Text is only part of the picture
  - We want LLMs that can understand the world by seeing and listening as well
  - Models should be able to do cross-modal reasoning and learning
- Multimodality can be introduced
  - From pre-training: Gemini
  - From instruction-tuning: AudioGPT, Flamingo

Modelling data using continuous representations

- Using continuous speech representations
  - Pros
    - Rich information
    - Good performance
  - Cons
    - Computationally heavy
    - Storage heavy

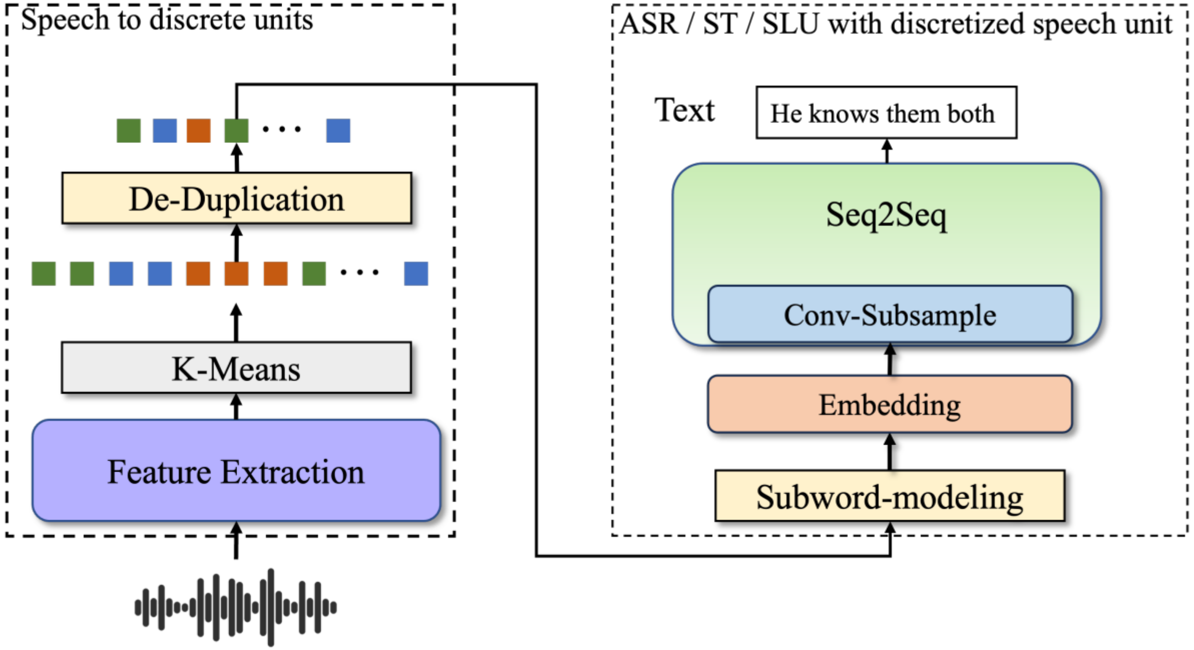


Modeling data using Discrete Units

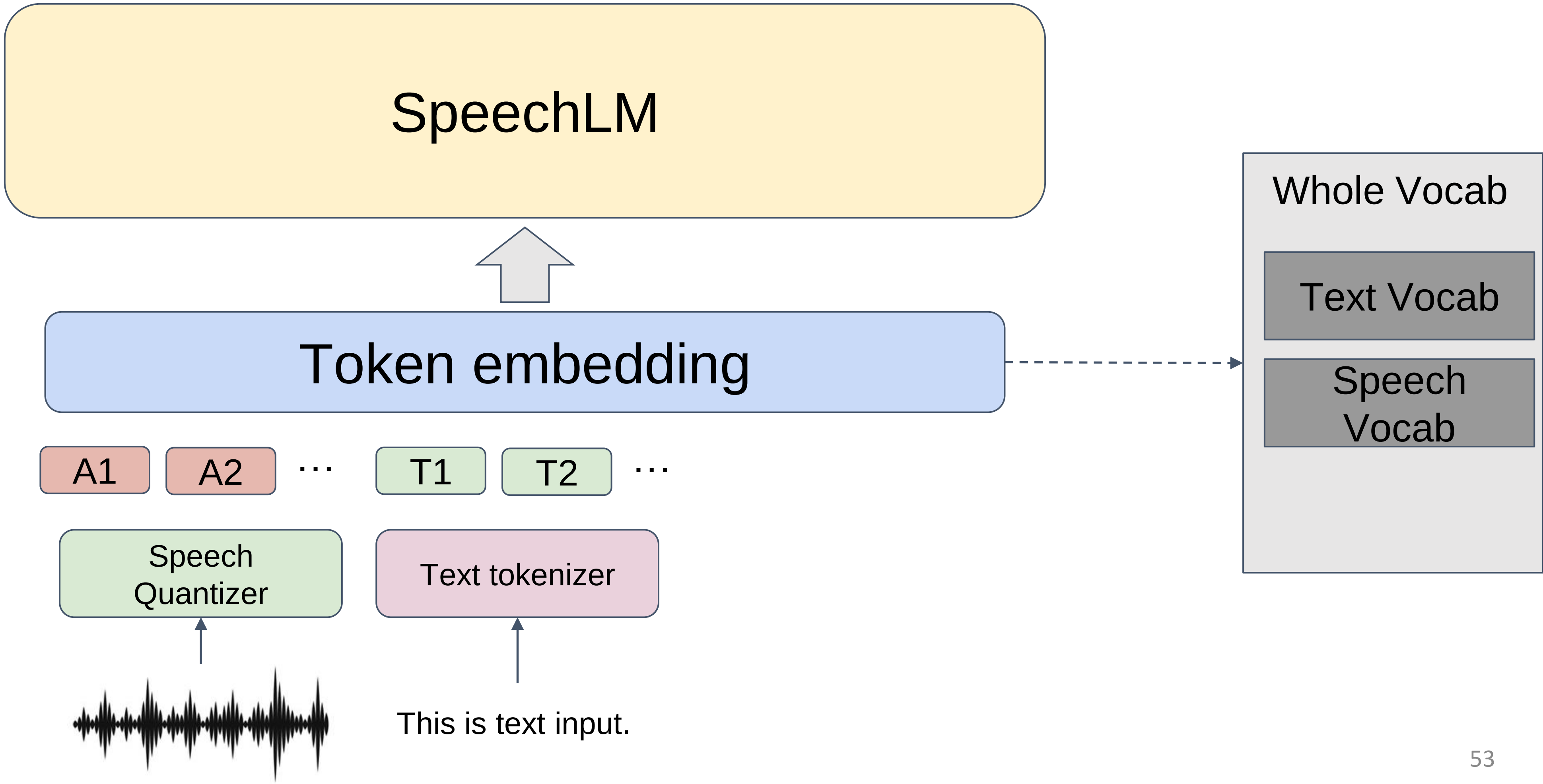
- Recently discrete units shows promising performance and benefit

Chang, Xuankai, et al. "Exploring Speech Recognition, Translation, and Understanding with Discrete Speech Units: A Comparative Study." *arXiv preprint arXiv:2309.15800* (2023).

  - Storage
    - Audio features (HuBERT): 1024 dim * 32 bit (float)
    - Discrete unit (1000 / 2000-cluster): 12 bit
  - Sequence length (> 50% reducti
    - De-duplication
    - Subword Modeling
  - Performance is okay
    - >fbank, ~<SSL feature
  - We used semantic features fron
    - ASR / ST / SLU



Modeling data using Discrete Unit



Multimodal LLMs – Representing Images

- Continuous embeddings
  - concatenated with the embeddings of text inputs to LLMs
  - Pre-trained independently
  - Ex: CLIP

- Discrete representations
  - Extracted from self-supervised audio models like VQ-VAEs

Open Challenges - LLMs

- New Capabilities
  - Multimodal
  - Multi-lingual
  - More Complex Tasks
- Performance
  - Reduce Hallucinations
  - Improve Alignment with Human Preference
  - Increase Context Length Efficiently
  - Improve Data, Training Strategy, and Model Architecture
- Efficiency
  - Computational cost, time, and money
  - Compute architecture – GPU/ TPU/ HPU

Open Challenges - LLMs

- Safety
  - Reduce Harm
  - Improve Adversarial Robustness
  - Privacy Concerns
- Interpretability
  - Why do LLMs do what they do?

Summary

- LLMs are large-scale models that possess astounding abilities
- Scaling both data and model capacity is important for performance and leads to the emergence of new abilities
- Decoder-only architectures are popular for convergence and performance
- LLMs are trained using pre-training, SFT, RLHF
- LLMs are evaluated using prompting/ strategies like ICL and CoT
- Multimodal LLMs can process audio, text, images and more.

Thank you!