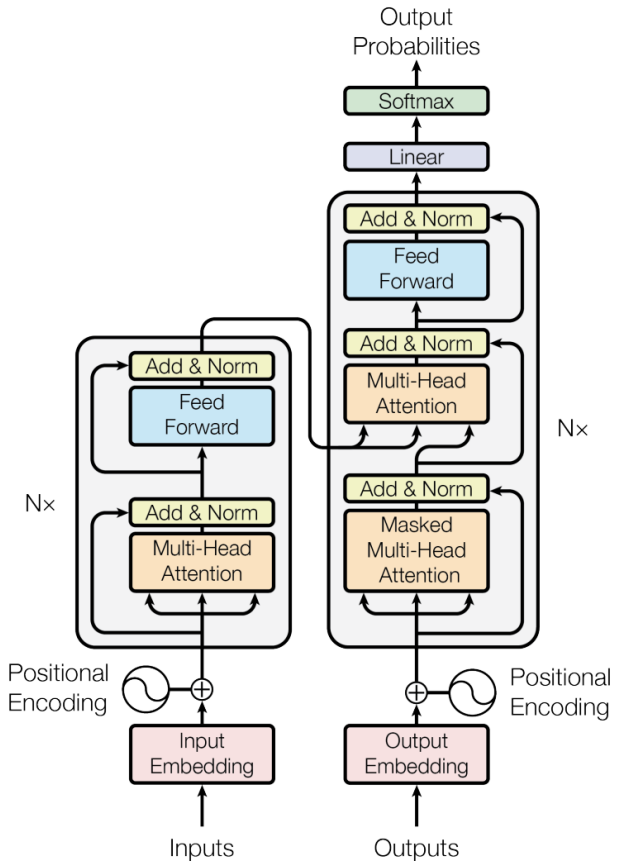
Large Language Models

Review of recent research

Pre-trained Language Models

BERT

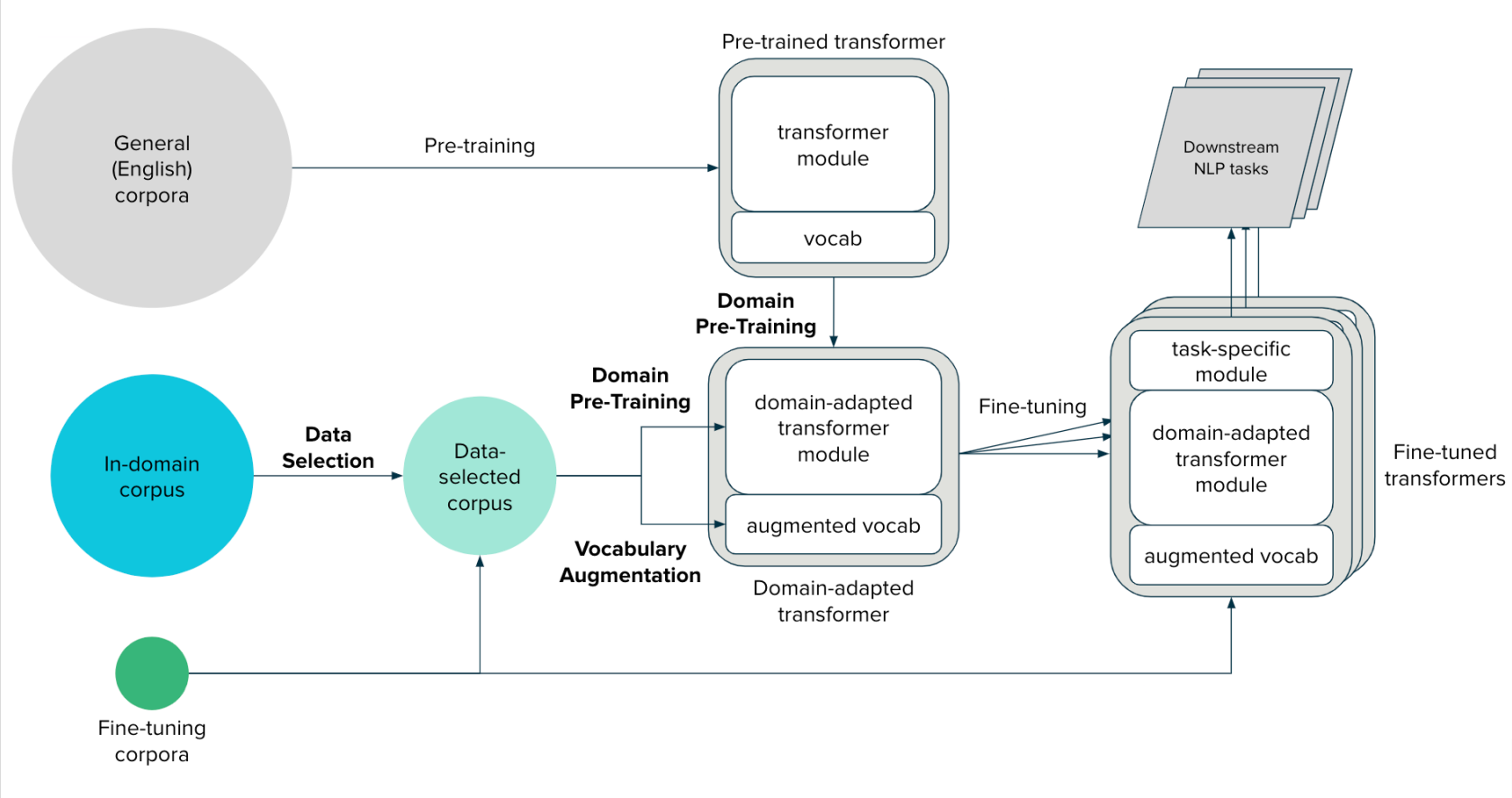
Encoder



GPT

Decoder

Existing Paradigm In NLP



Increase In The Scale Of LMs

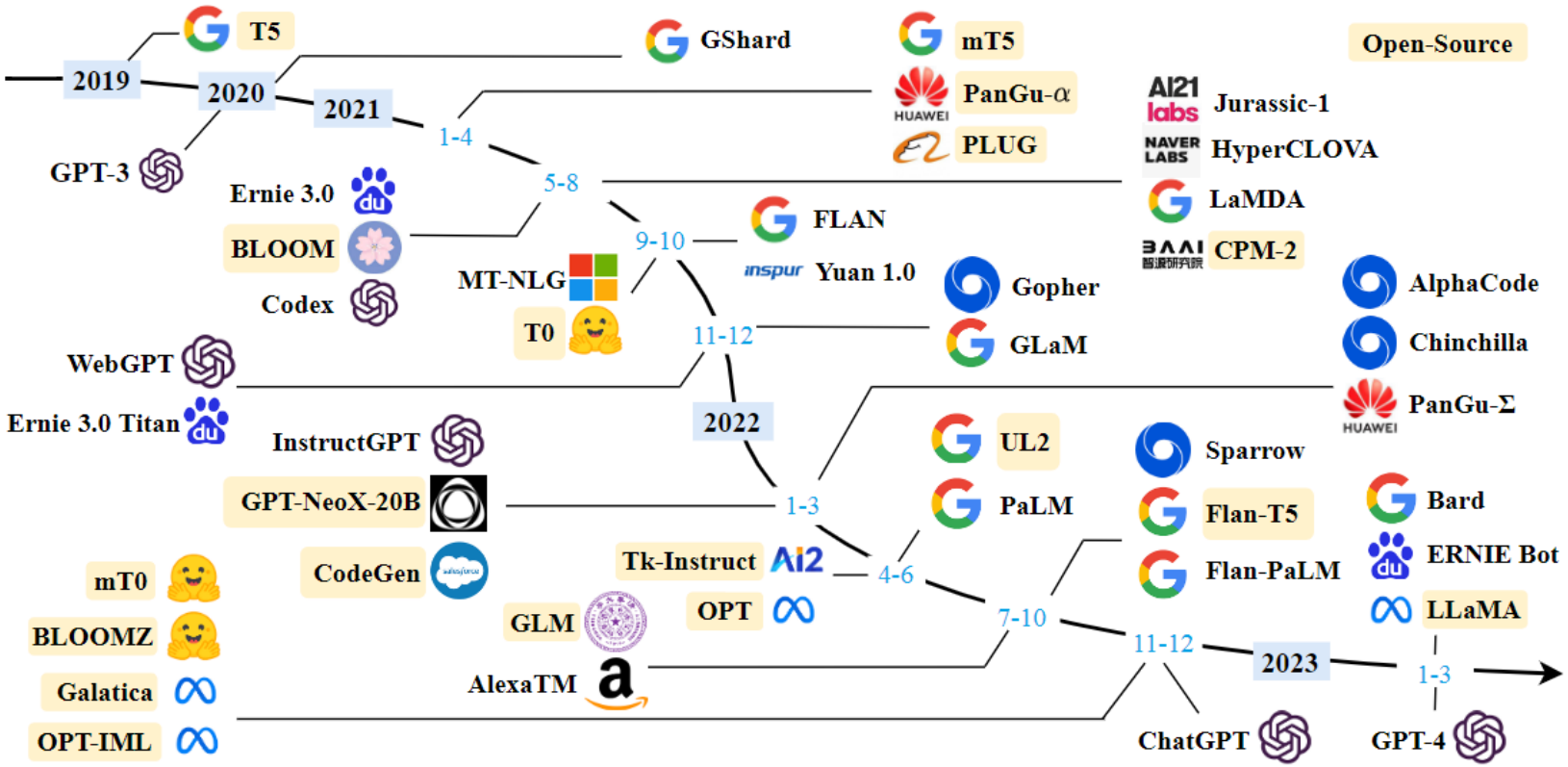
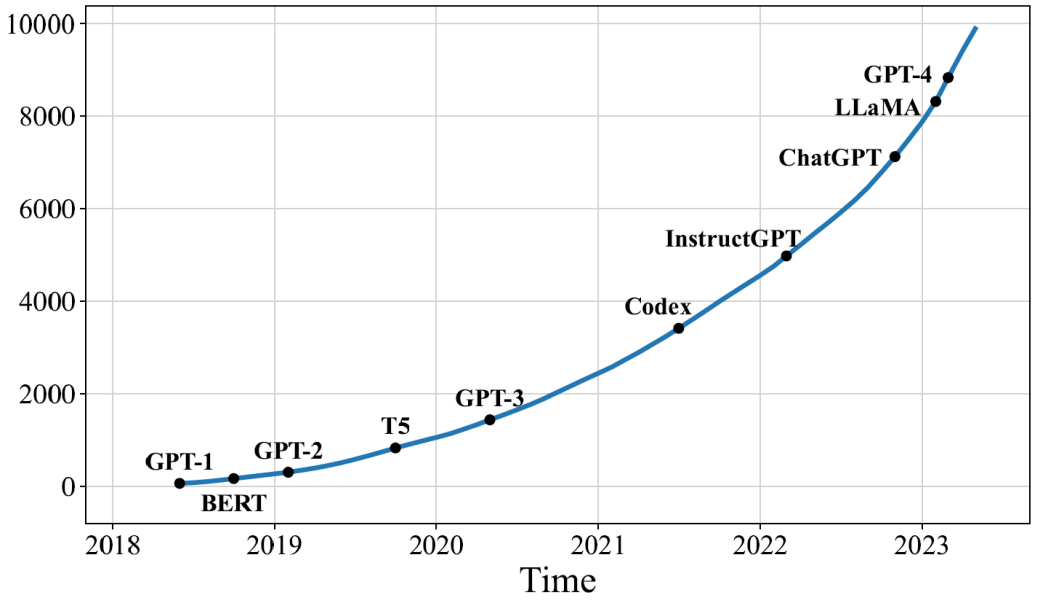
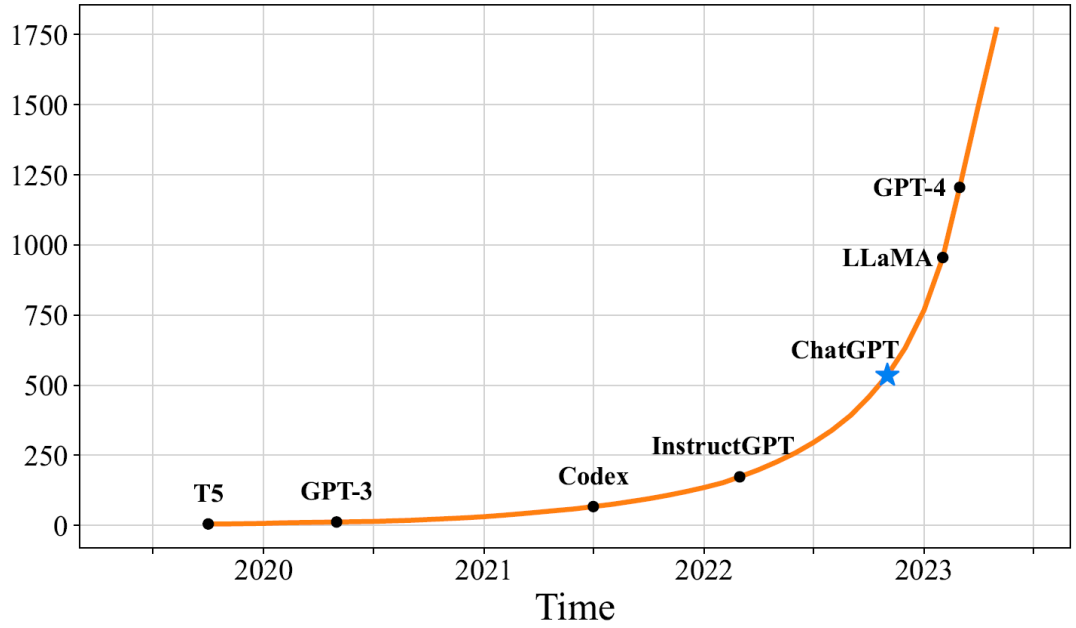


Fig. 1. A timeline of existing large language models (having a size larger than 10B) in recent years. We mark the open-source LLMs in yellow color.

Trend In LLM Research



Query = “Language Model”



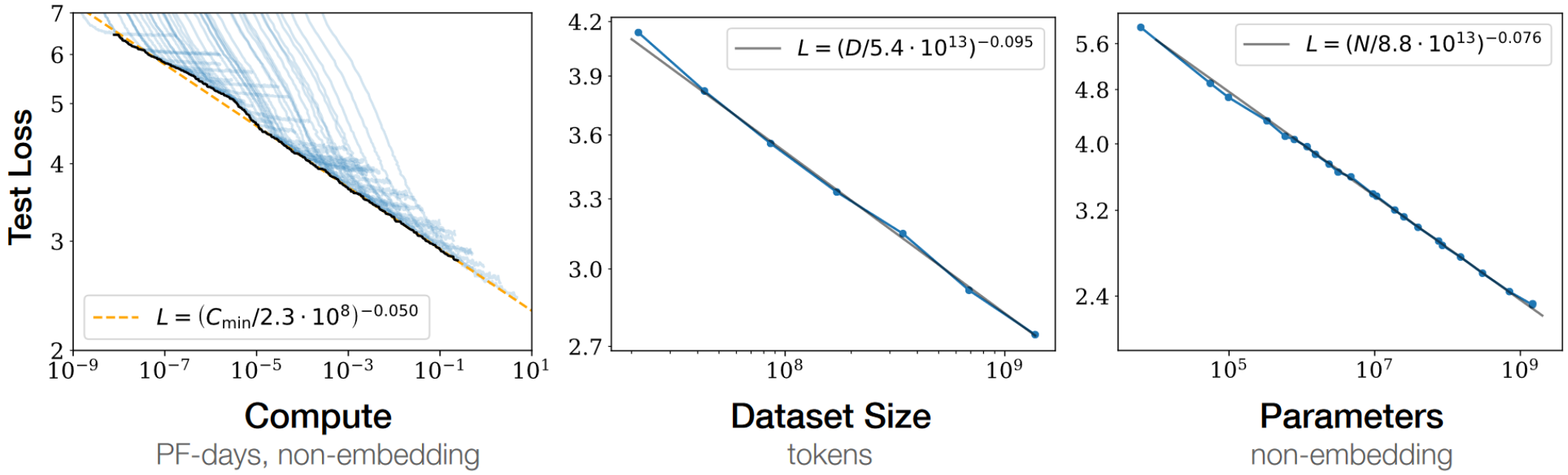
Query = “Large Language Model”

Key Differences: PLM v/s LLM

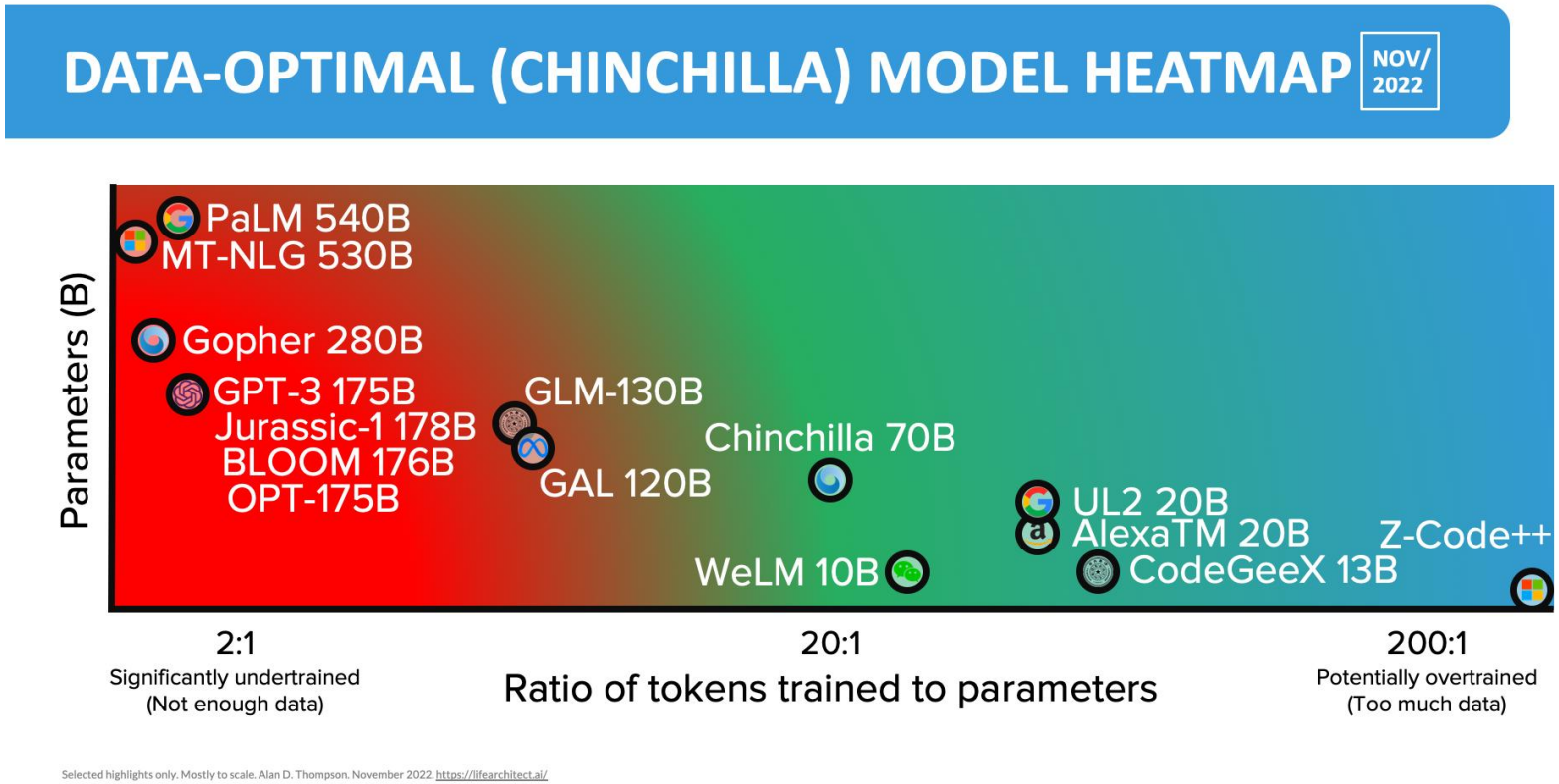
1. Displays emergent abilities
2. Provides a new way of using LM – Prompting Interface
3. No longer draws a clear distinction between research and engineering.

Scaling Laws For LLM

KM Scaling Law



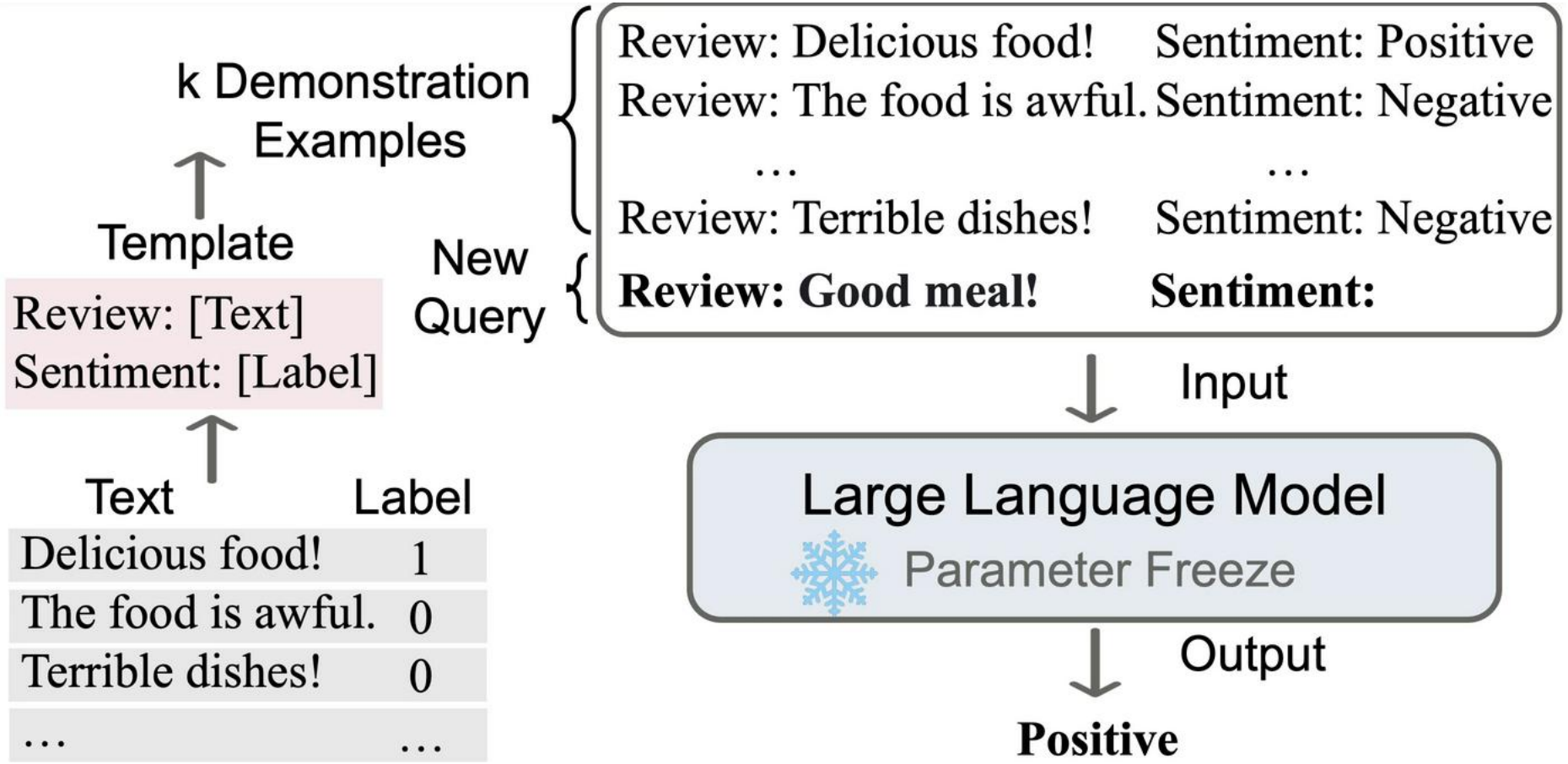
Chinchilla Scaling Law



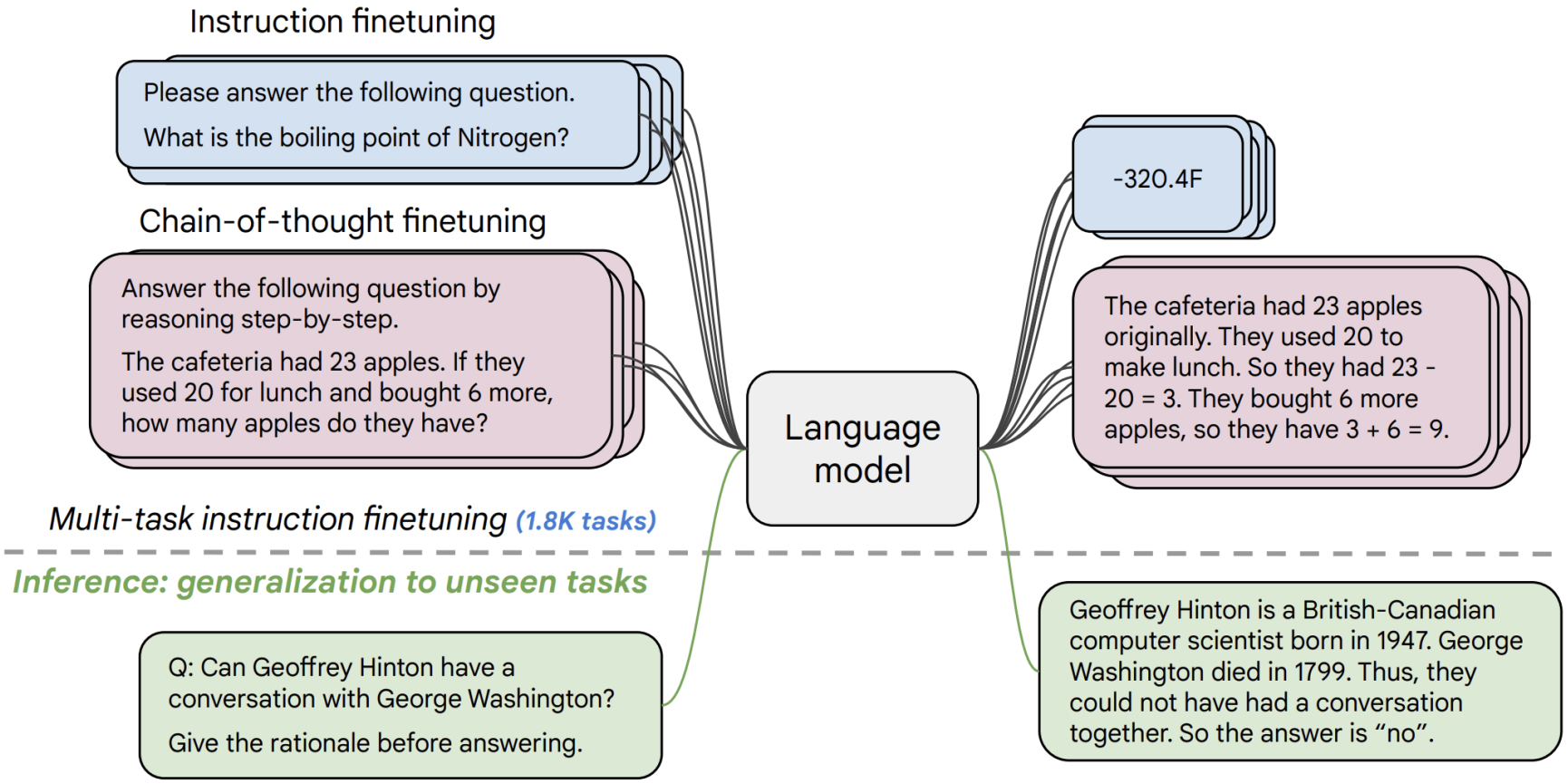
Selected highlights only. Mostly to scale. Alan D. Thompson. November 2022. https://lifearchitect.ai/

Emergent Abilities

In-Context Learning



Instruction Following



Step-by-Step Reasoning

CommonsenseQA

Q: Sammy wanted to go to where the people were. Where might he go?
Options: (a) race track (b) populated areas (c) desert (d) apartment (e) roadblock

A: The answer must be a place with a lot of people. Race tracks, desert, apartments, and roadblocks don't have a lot of people, but populated areas do. So the answer is (b).

StrategyQA

Q: Yes or no: Would a pear sink in water?

A: The density of a pear is about 0.6 g/cm^3, which is less than water. Thus, a pear would float. So the answer is no.

Date Understanding

Q: The concert was scheduled to be on 06/01/1943, but was delayed by one day to today. What is the date 10 days ago in MM/DD/YYYY?

A: One day after 06/01/1943 is 06/02/1943, so today is 06/02/1943. 10 days before today is 05/23/1943. So the answer is 05/23/1943.

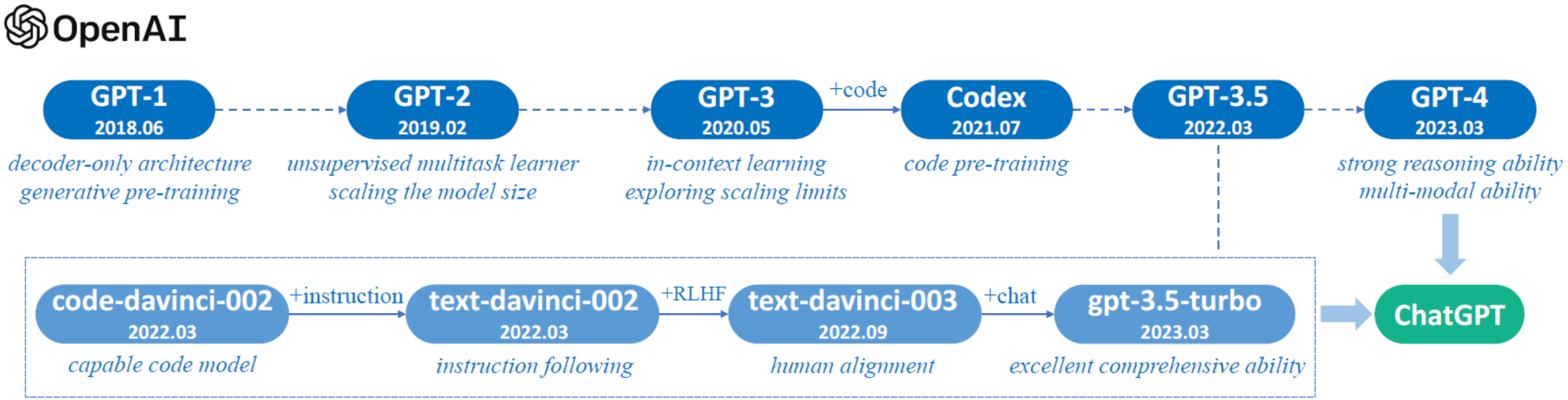
Sports Understanding

Q: Is the following sentence plausible? "Joao Moutinho caught the screen pass in the NFC championship."

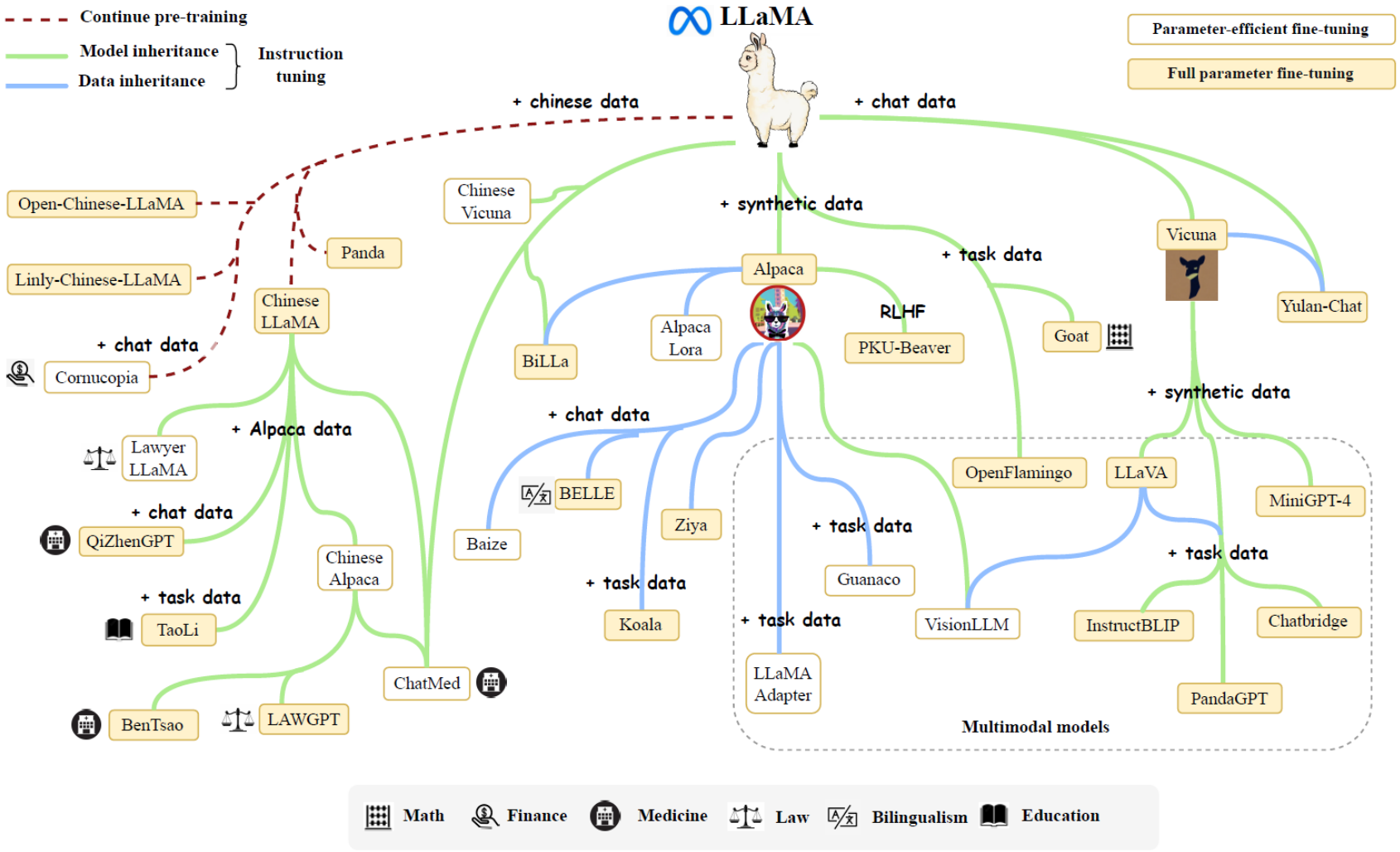
A: Joao Moutinho is a soccer player. The NFC championship is part of American football, not soccer. So the answer is no.

Evolution Of GPT Architecture

The basic principle underlying GPT models is to compress the world knowledge into the decoder-only Transformer model by language modeling, such that it can recover (or memorize) the semantics of world knowledge and serve as a general-purpose task solver



LLAMA Model Family



Commonly Used Corpora

Commonly Used Corpora

- Based on their content types we can categorize corpora into six groups:
  1. Books
  2. CommonCrawl
  3. Reddit links
  4. Wikipedia
  5. Code
  6. Others

Books

- **BookCorpus:** commonly used dataset in previous small-scale models (e.g., GPT and GPT-2), consisting of over 11,000 books covering a wide range of topics and genres.

- **Project Gutenberg:** consisting of over 70,000 literary books including novels, essays, poetry, drama, history, science, philosophy, and other types of works in the public domain.

CommonCrawl

- CommonCrawl is one of the largest open-source web crawling databases, containing a petabyte scale data volume, which has been widely used as training data for existing LLMs.

- Requires data preprocessing to remove noisy and low-quality data from the web

Reddit Links

- Reddit is a social media platform that enables users to submit links and text posts, which can be voted on by others through “upvotes” or “downvotes”.

- Highly upvoted posts are often considered useful and can be utilized to create high-quality datasets.

- **WebText:** Closed source, **OpenWebText:** Open-source alternative

Wikipedia

- Wikipedia is an online encyclopedia containing a large volume of high-quality articles on diverse topics.

- Most of these articles are composed in an expository style of writing (with supporting references), covering a wide range of languages and fields.

Code

- To collect code data, existing work mainly crawls open-source licensed codes from the Internet.

- Two major sources are public code repositories under open-source licenses (e.g., GitHub) and code-related question-answering platforms (e.g., StackOverflow)

- **BIGQUERY:** Publicly released dataset from Google

Others

- The Pile is a large-scale, diverse, and opensource text dataset consisting of over 800GB of data from multiple sources, including books, websites, codes, scientific papers, and social media platforms.

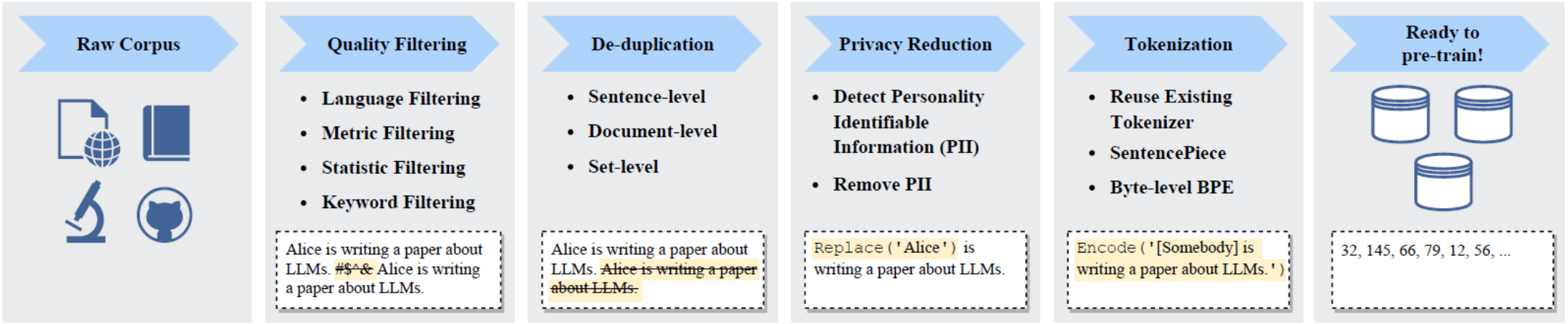
What Corpora Do LLMs Use

- **GPT-3 (175B):** 300B tokens from CommonCrawl, WebText, Books , Books and Wikipedia

- **PaLM (540B):** 780B tokens, which is sourced from social media conversations, filtered webpages, books, Github, multilingual Wikipedia, and news.

- **LLaMA:** ~1T tokens, CommonCrawl, C4, Github, Wikipedia, Books, ArXiv, and StackExchange

Typical Data Preprocessing Pipeline



Effect Of Pre-training Data

Effect Of Pre-training Data

- Mixture of sources
- Amount Of Pre-training data
- Quality Of Pre-training data

Mixture Of Sources

- By pre-training on a mixture of text data from diverse sources, LLMs can acquire a broad scope of knowledge and may exhibit a strong generalization capacity.
- Training on books can improve ability to capture long-term dependencies.
- Training on code helps to improve reasoning capability

Amount Of Pre-training Data

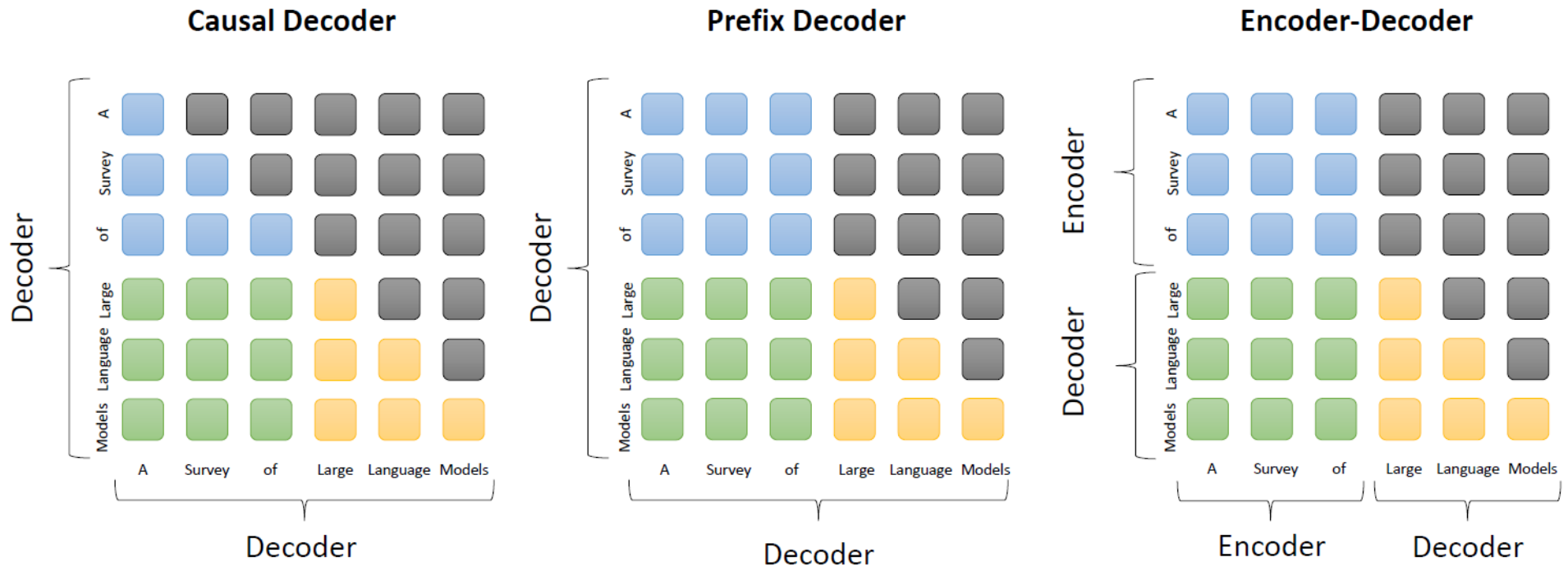
- Existing studies have found that with the increasing parameter scale in the LLM, more data is also required to train the model
- A recent study has shown that several existing LLMs suffer from sub-optimal training due to inadequate pretraining data

Quality Of Pre-training Data

- Low-quality corpus, such as noisy, toxic, and duplicate data, may hurt the performance of model

- This is done by comparing training results on filtered and unfiltered data which conclusively shows that quality of pre-training data is important

Mainstream architecture



attention between prefix tokens attention between prefix and target tokens

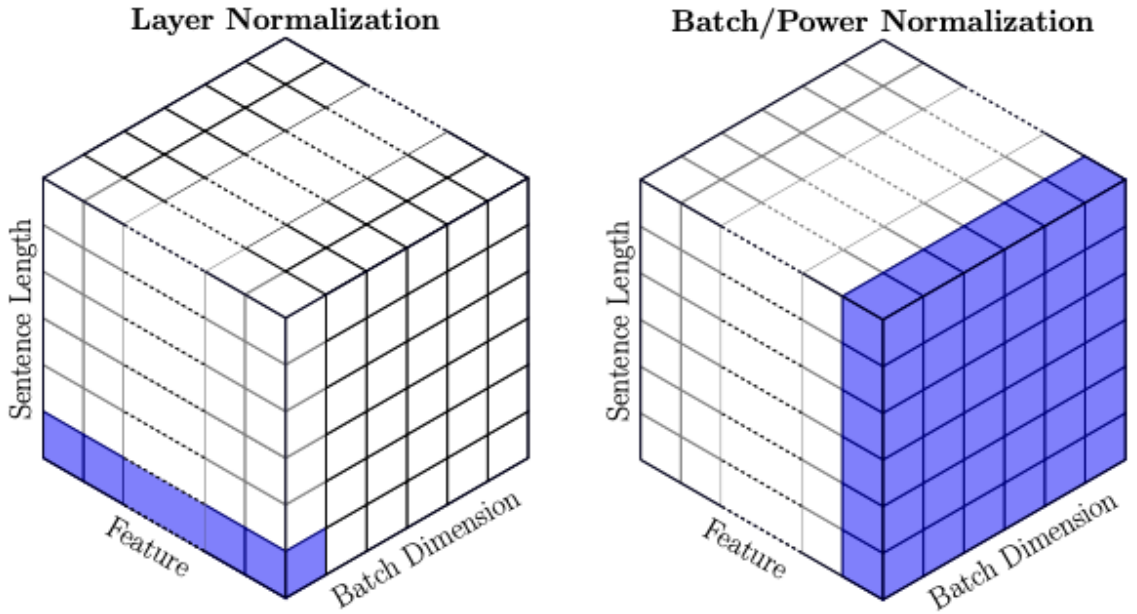
attention between target tokens masked attention

Normalization Methods

Layer Normalization

- BatchNorm is a commonly used normalization method. However, it is difficult to deal with sequence data of variable lengths and small-batch data.

- LayerNorm is introduced to conduct layerwise normalization. Specifically, the mean and variance over all activations per layer are calculated to recenter and re-scale the activations.



Normalization Methods

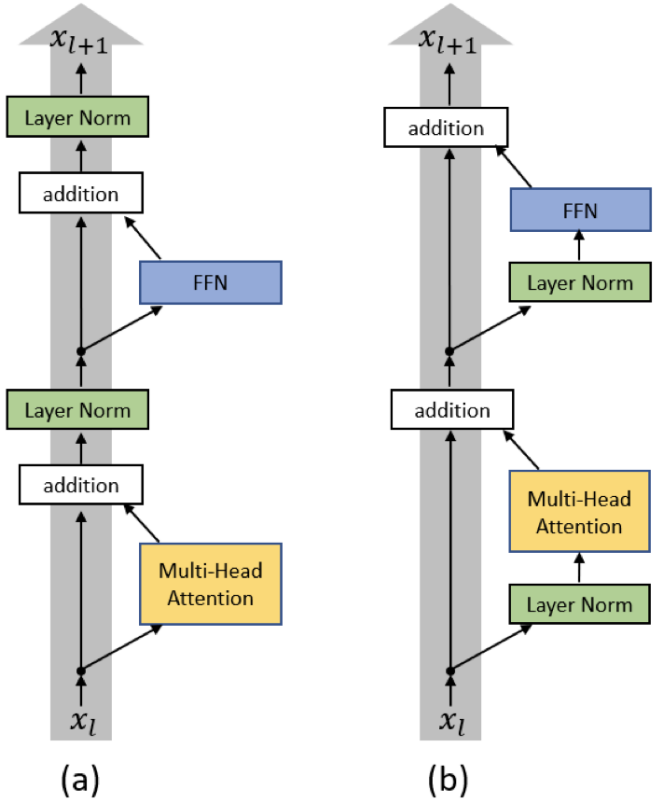
- **RMSNorm:** To improve speed of layer-norm by re-scaling with only Root Mean Square of summed activations, instead of mean and variance

- **DeepNorm:** To improve training stability in deep neural networks, where DeepNorm is used as residual connections.

Normalization Position

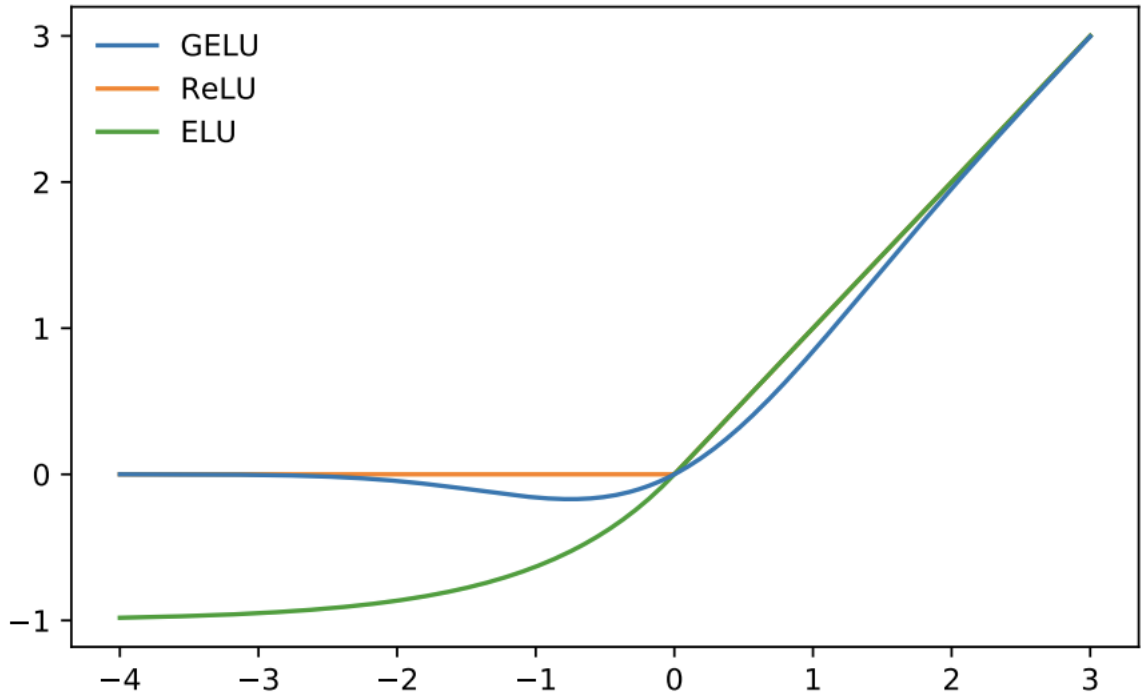
- **Post-LN** is used in the vanilla Transformer, which is placed between residual blocks. However, existing work has found that the training of Transformers with post-LN tends to be instable due to the large gradients near the output layer. **Pre-LN** is utilized in most LLMs

- Based on pre-LN, **Sandwich-LN** adds extra LN before the residual connections to avoid the value explosion issues in Transformer layer outputs. However not stable and not highly adopted



Activation Functions

- To obtain good performance, activation functions also need to be properly set in feed-forward networks. In existing LLMs, GeLU activations are widely used.
- There are variants like SwiGLU and GeGLU which perform better, but in practice they add 50% more parameters.

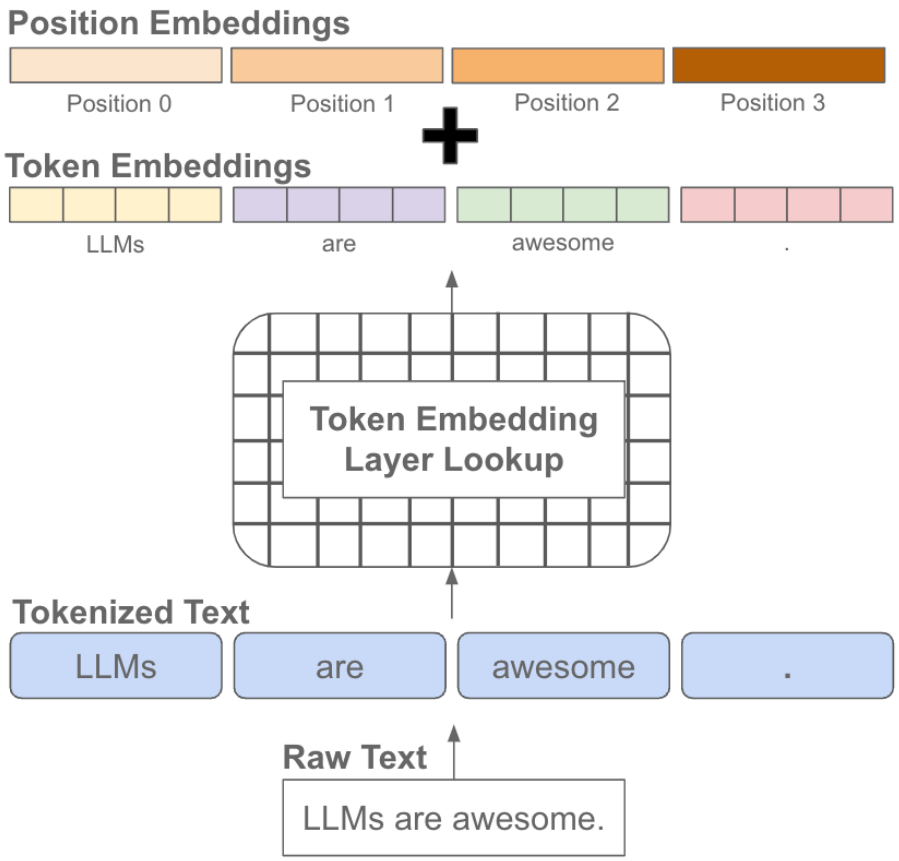


Position Embeddings

Absolute Position Embeddings

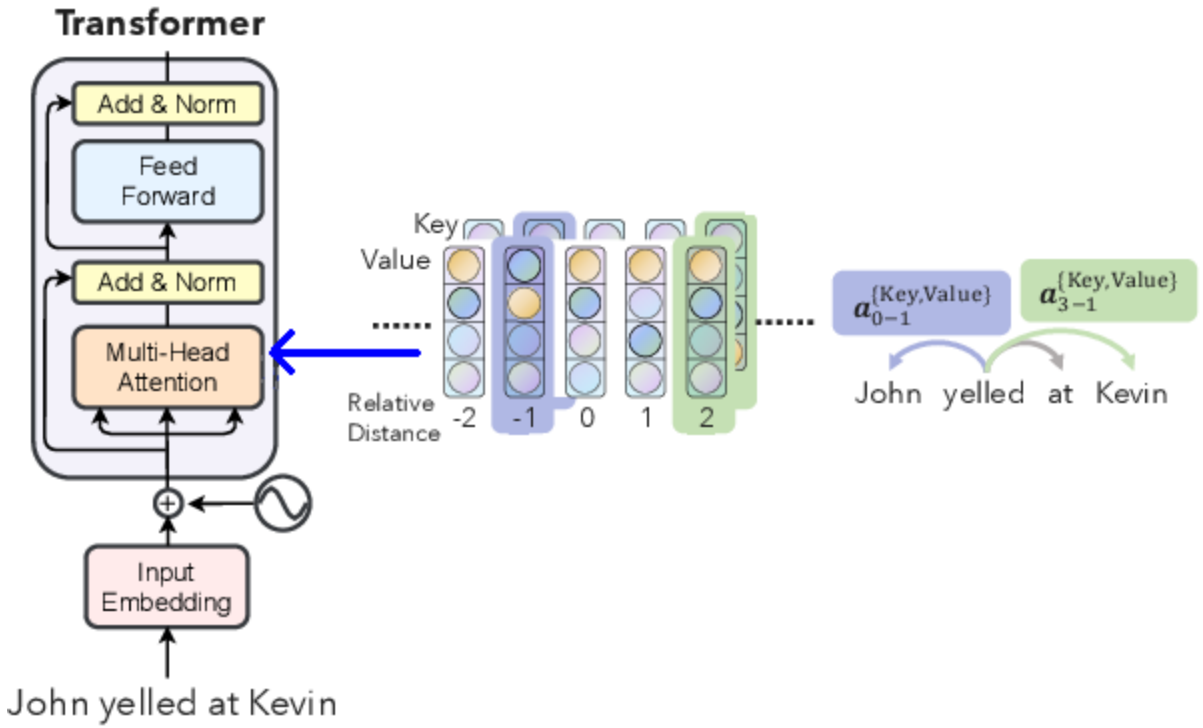
- In the vanilla Transformer, absolute position embeddings are employed. At the bottoms of the encoder and the decoder, the absolute positional embeddings are added to the input embeddings.

- Two variants: sinusoidal and learned positional embeddings



Relative Position Embeddings

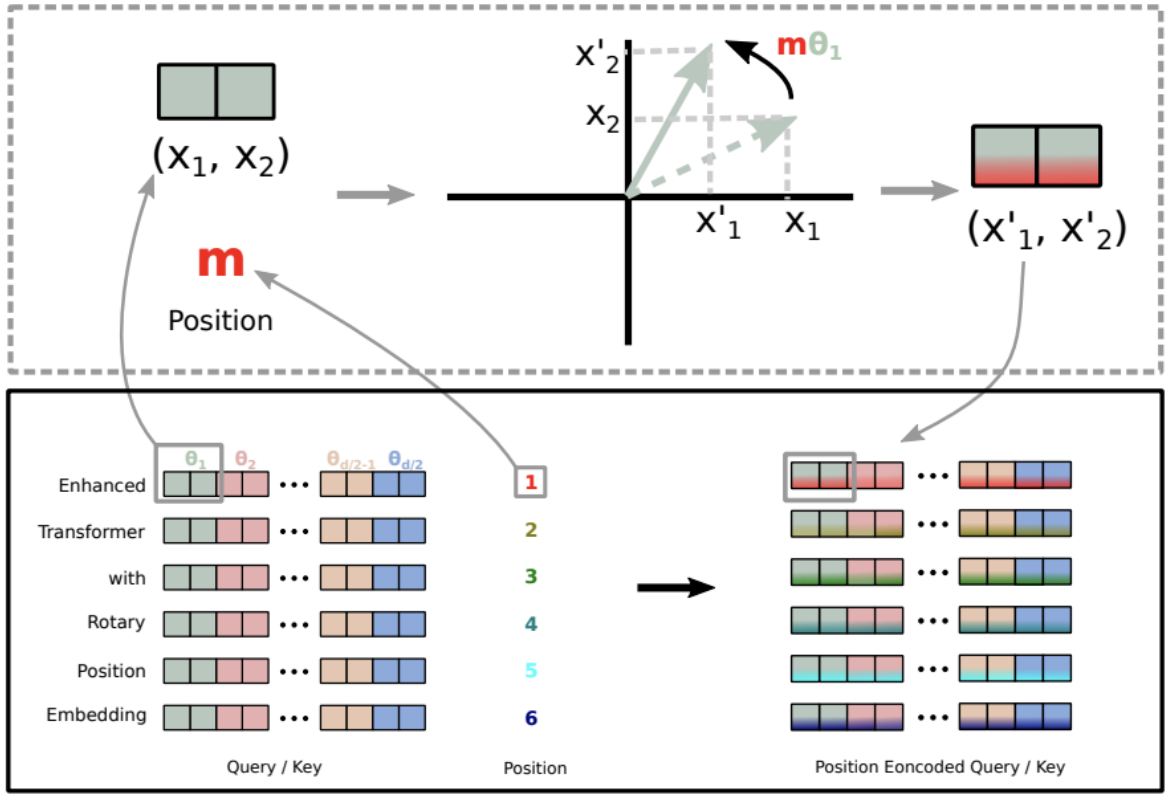
- Unlike absolute position embeddings, relative positional embeddings are generated according to the offsets between keys and queries.
- Transformers with relative position embedding can generalize to sequences longer than those sequences for training, i.e., extrapolation



Rotary Position Embeddings

- Rotary position embedding (RoPE) sets specific rotatory matrices based on the absolute position of each token

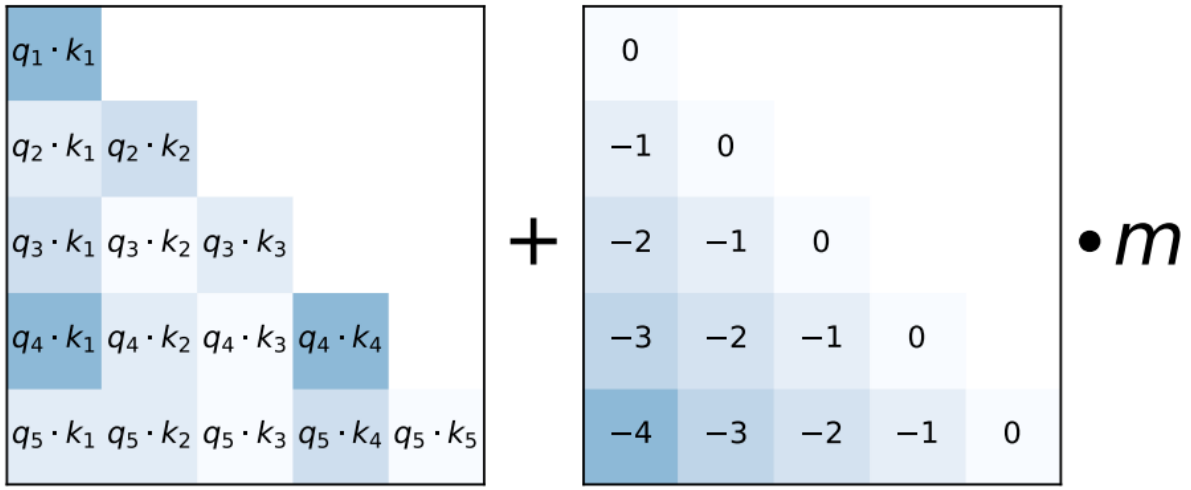
- Due to the excellent performance and the long-term decay property, RoPE is widely adopted in the latest LLMs



ALiBi

- ALiBi is proposed to improve the extrapolation of Transformer. Like relative position embedding, it biases attention scores with a penalty based on the distances between keys and queries.

- Unlike Relative Position Embeddings the penalty scores in ALiBi are pre-defined without any trainable parameters.

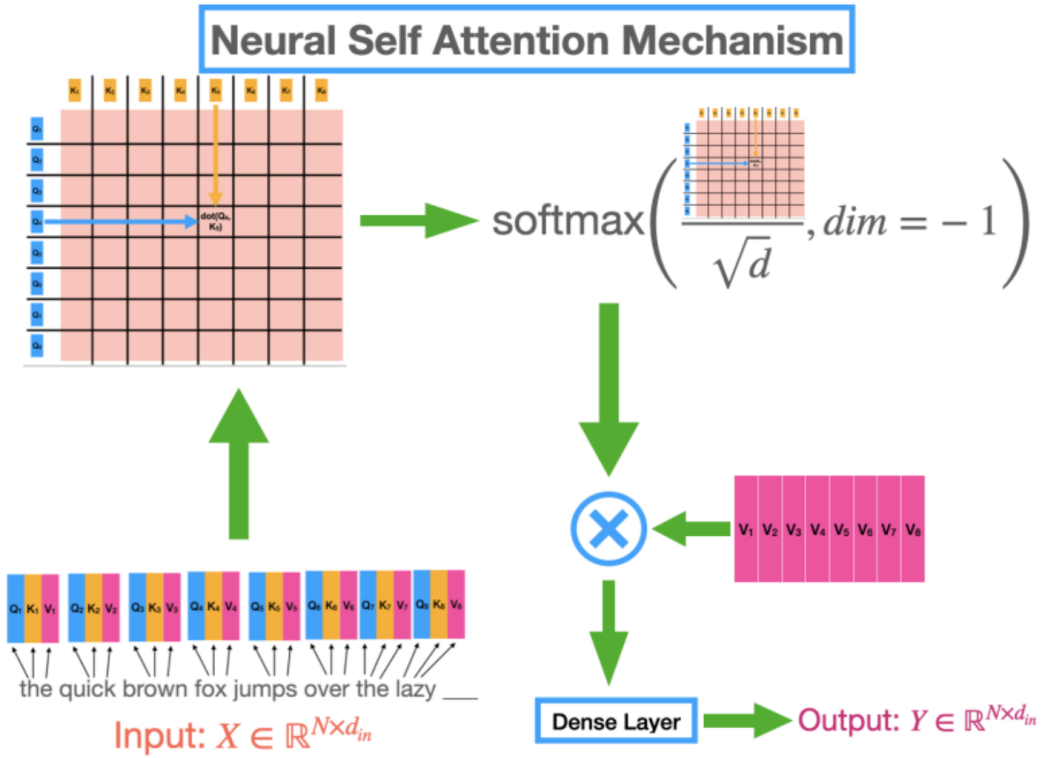


Attention And Bias

Self-Attention Mechanism

- Attention mechanism is a critical component of Transformer. It allows the tokens across the sequence to interact with each other and compute the representations of the input and output sequence.

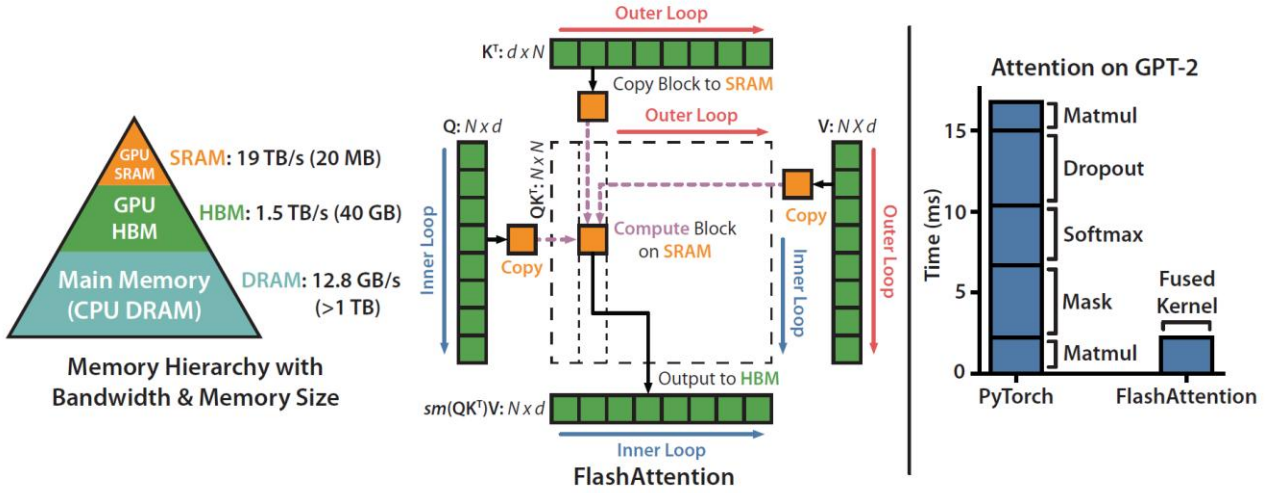
- In the vanilla Transformer, the attention mechanism is conducted in a pairwise way, considering the relations between all token pairs in a sequence.



Flash Attention

- FlashAttention proposes to optimize the speed and memory consumption of attention modules on GPUs from an IO-aware perspective.

- FlashAttention organizes the input into blocks and introduces necessary recomputation, both to make better use of the fast memory SRAM.



Pre-Training Tasks

Pre-training Tasks

- **Language Modelling:** Given a sequence of tokens, auto-regressively predict the next token
- **Denoising Auto-Encoding:** Given corrupted text with randomly replaced spans, recover the correct tokens
- **Mixture Of Denoisers:** Combines both the LM and DAE losses together

Scalable Training Techniques

Scalable Training Techniques

- 3D Parallelism
- ZeRo Techniques
- Mixed Precision Training

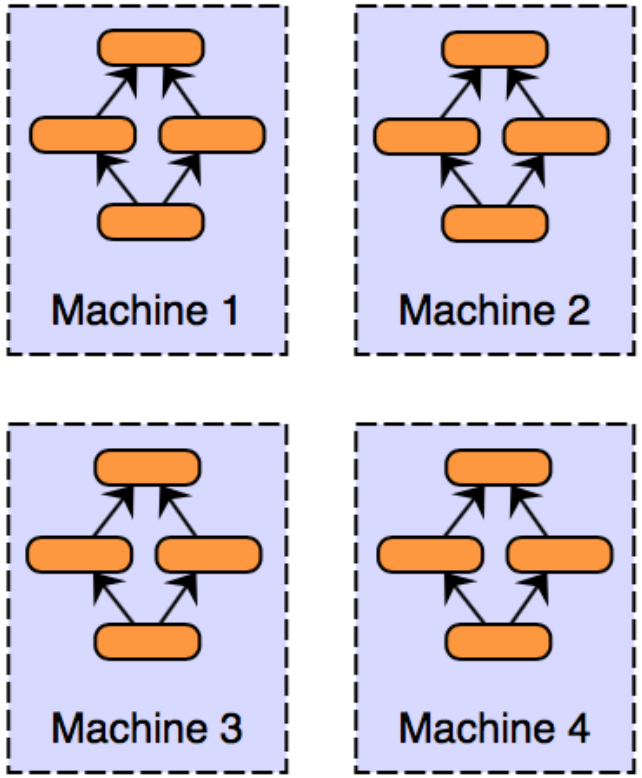
Scalable Training Techniques

- **3D Parallelism**
- ZeRo Techniques
- Mixed Precision Training

Data Parallelism

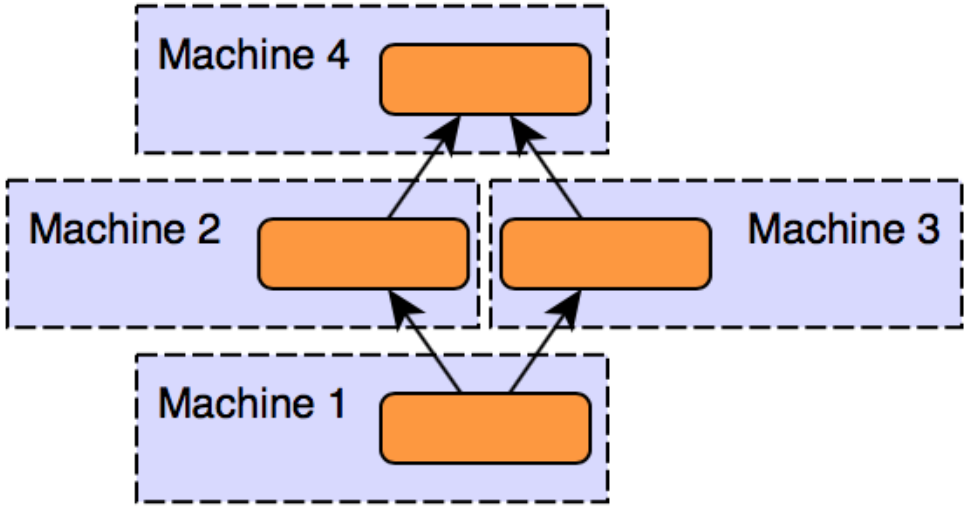
- It replicates the model parameters and optimizer states across multiple GPUs and then distributes the whole training corpus into these GPUs.

- The computed gradients on different GPUs will be further aggregated to obtain the gradients of the entire batch for updating the models in all GPUs.



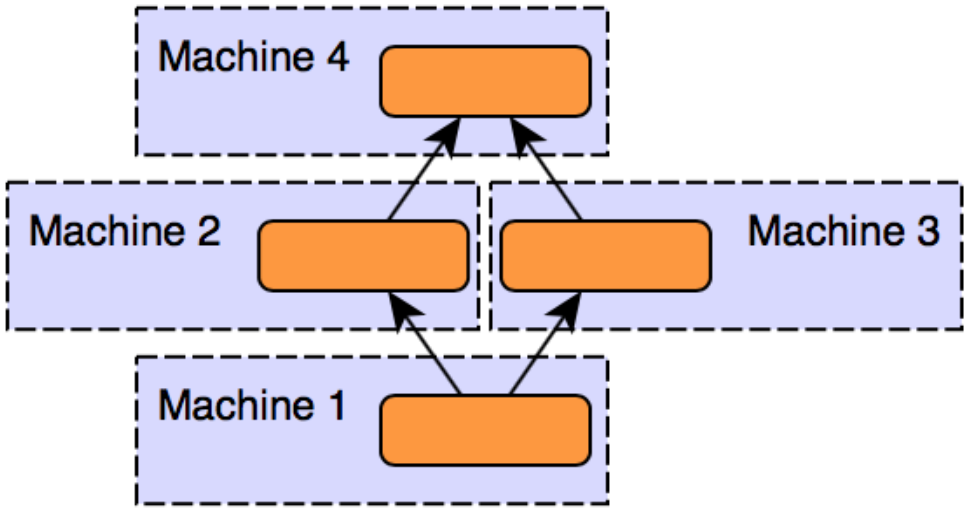
Pipeline Parallelism

- In the case of a Transformer model, pipeline parallelism loads consecutive layers onto the same GPU, to reduce the cost of transmitting the computed hidden states
- A naïve implementation may result in a lower GPU utilization rate as each GPU has to wait for the previous one to complete the computation, leading to the unnecessary cost of bubbles overhead



Tensor Parallelism

- Unlike pipeline parallelism, tensor parallelism focuses on decomposing the tensors (the parameter matrices) of LLMs.
- By placing parameter matrices on different GPUs, the matrix multiplication operation would be invoked at two GPUs in parallel, and the result can be obtained by combining the outputs from across devices



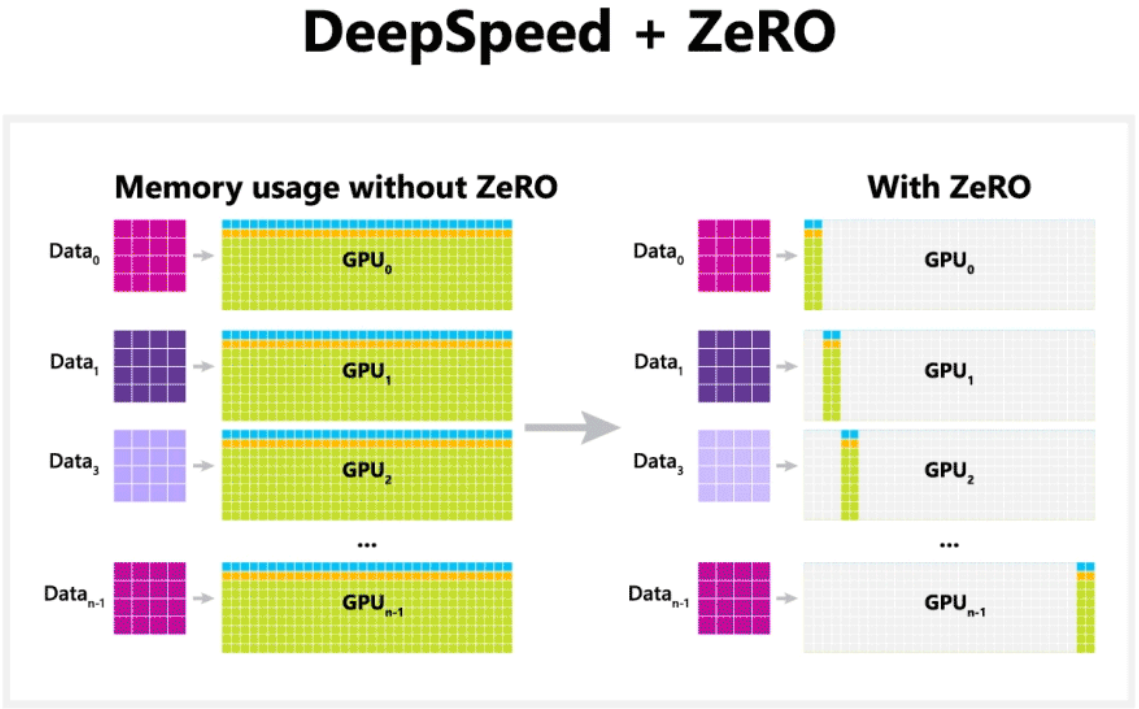
Scalable Training Techniques

- 3D Parallelism
- **ZeRo Techniques**
- Mixed Precision Training

ZeRO Techniques

- Data Parallelism causes memory redundancy problems since each GPU requires to store same copy of parameters, gradients and optimizer state.

- ZeRO technique aims to retain only a fraction of data on each GPU, while the rest data can be retrieved from other GPUs when required.

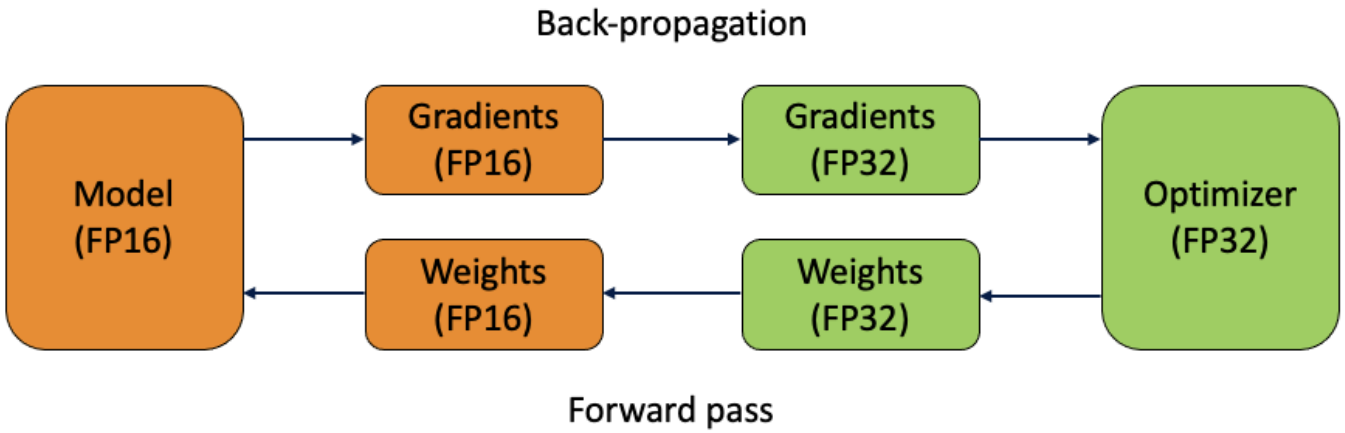


Scalable Training Techniques

- 3D Parallelism
- ZeRo Techniques
- **Mixed Precision Training**

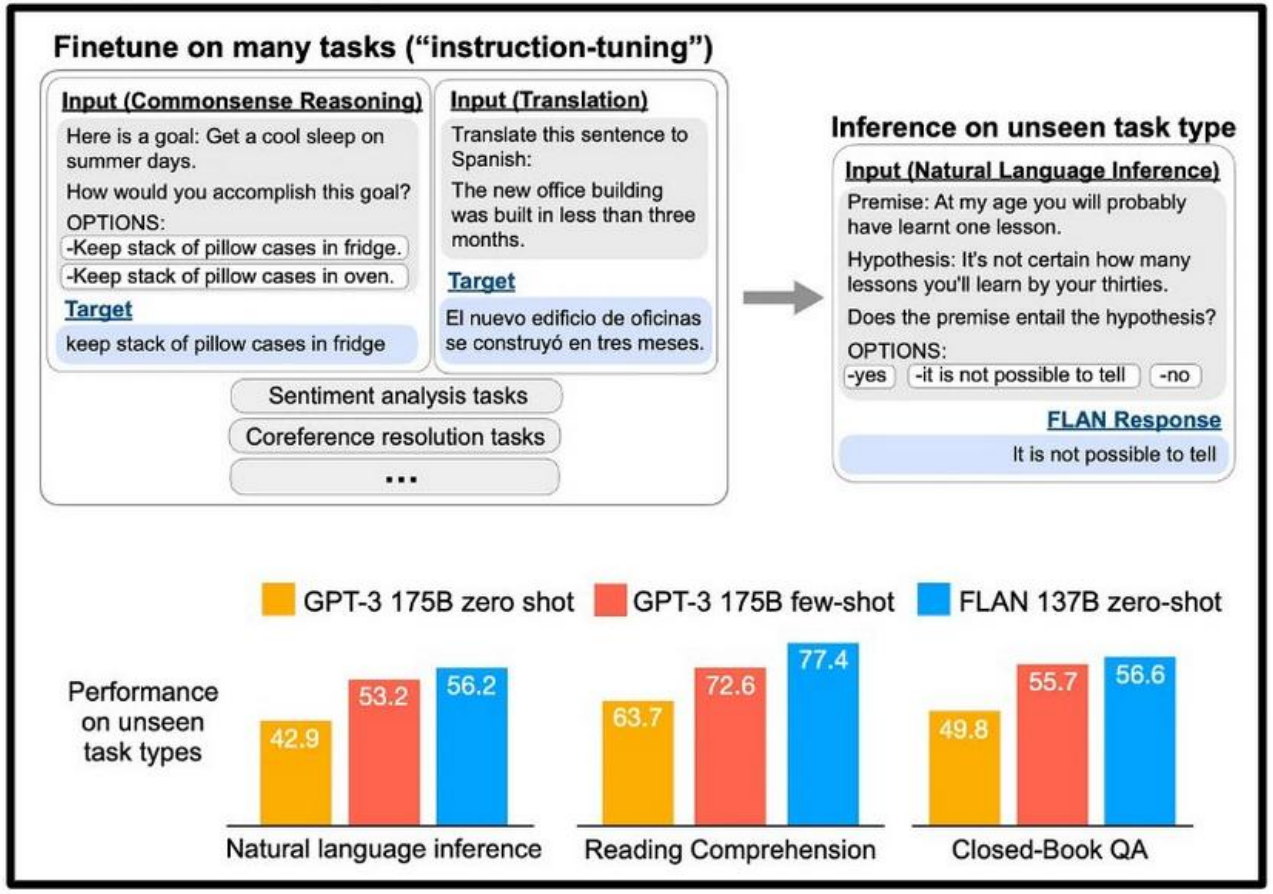
Mixed Precision Training

- Recent LLMs have started using FP16 training.
- A100s have twice the amount of FP16 computation units as FP32, the computational efficiency of FP16 can be further improved.
- FP16 leads to loss in accuracy. Instead use BF16, has more exponent bits and fewer significant bits

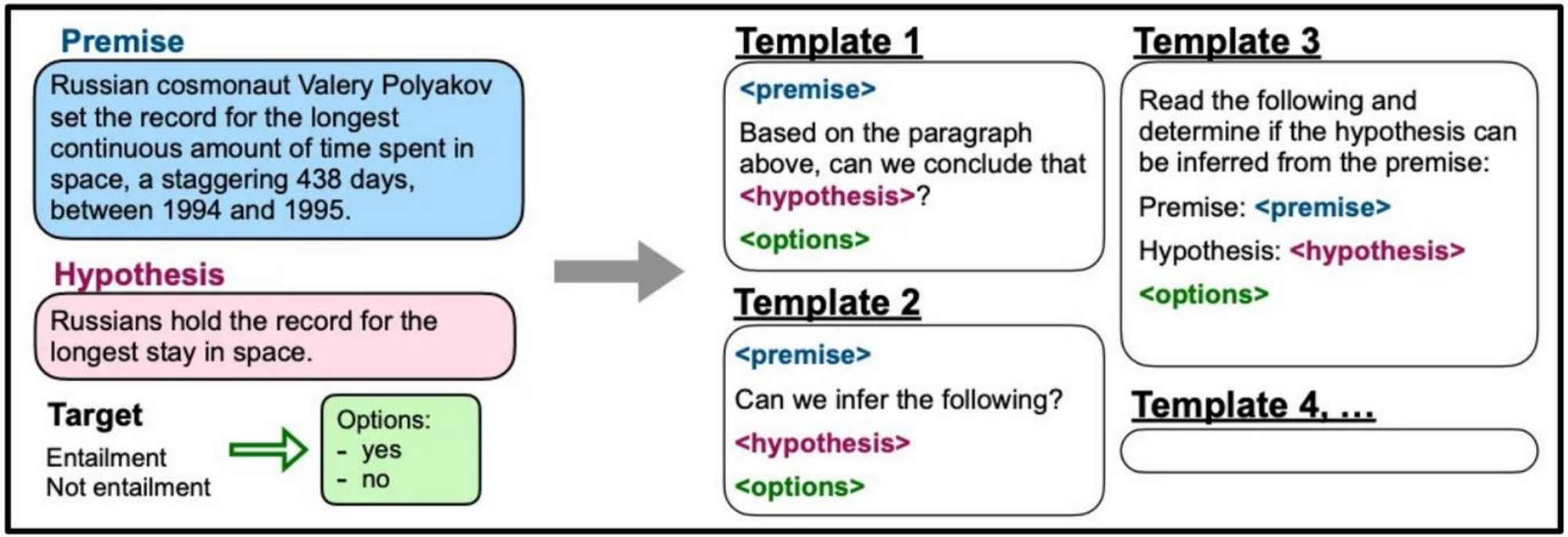


Adaptation Of LLMs

Instruction Tuning



Instruction Tuning

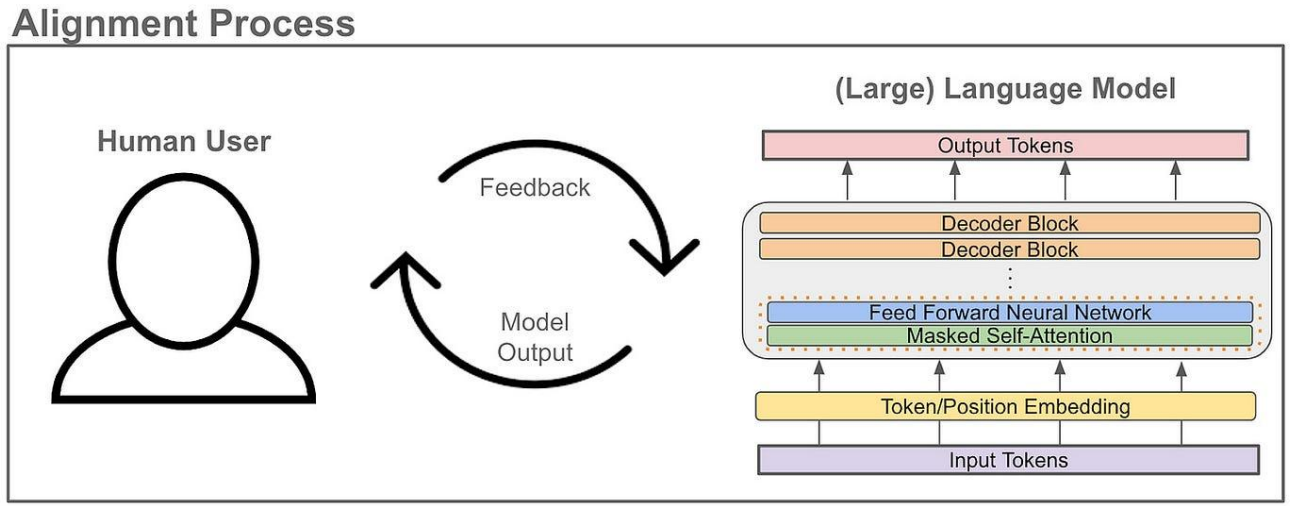


Effect Of Instruction Tuning

- Performance Improvement
- Task Generalization
- Domain Specialization

Alignment Tuning

- LLMs may sometimes exhibit unintended behaviors, e.g., fabricating false information, pursuing inaccurate objectives, and producing harmful, misleading, and biased expressions
- To avert these unexpected behaviors, human alignment has been proposed to make LLMs act in line with human expectations



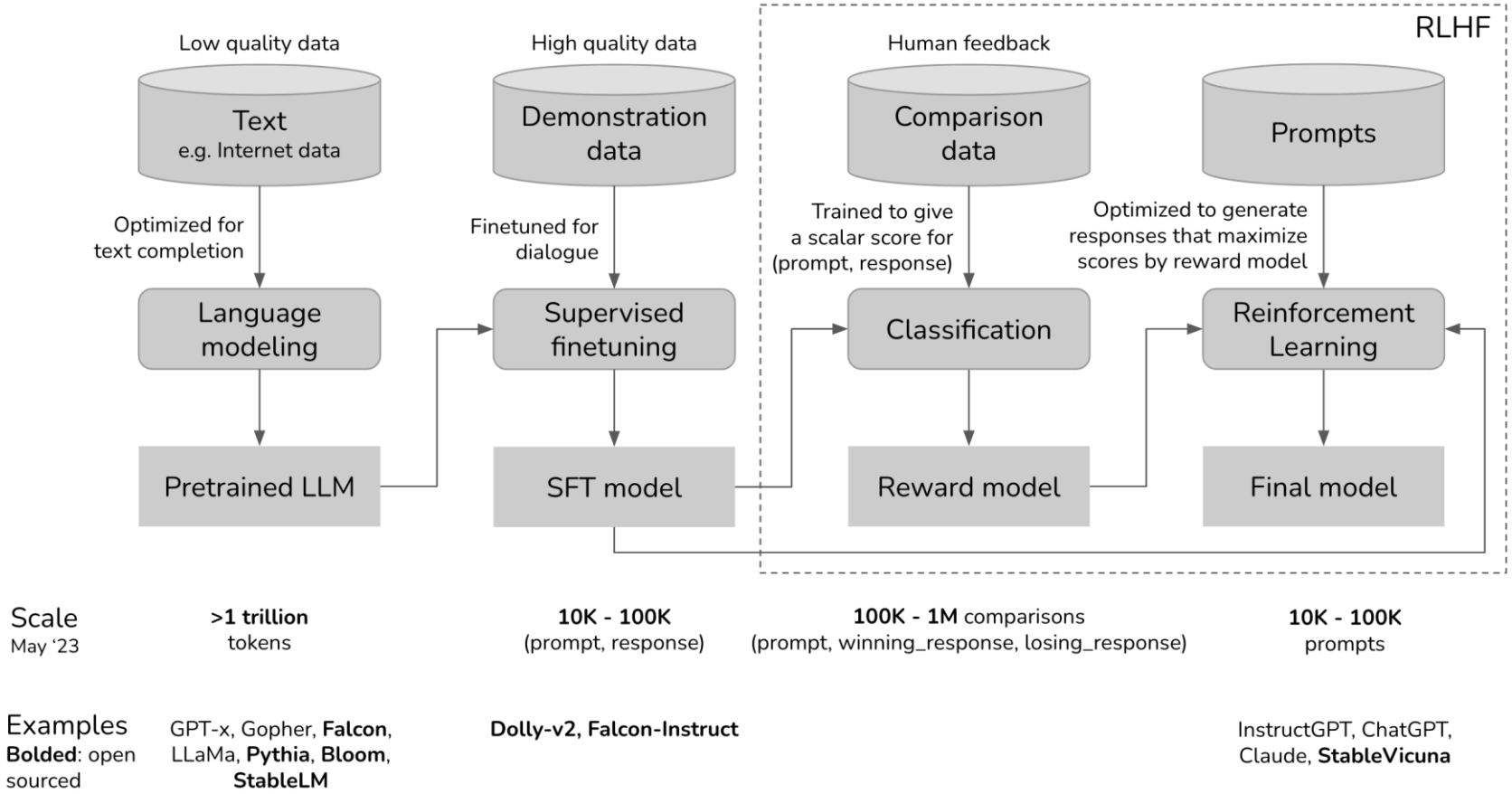
Alignment Criteria

- **Helpfulness:** To be helpful, the LLM should demonstrate a clear attempt to assist users in solving their tasks or answering questions in a concise and efficient manner as possible.

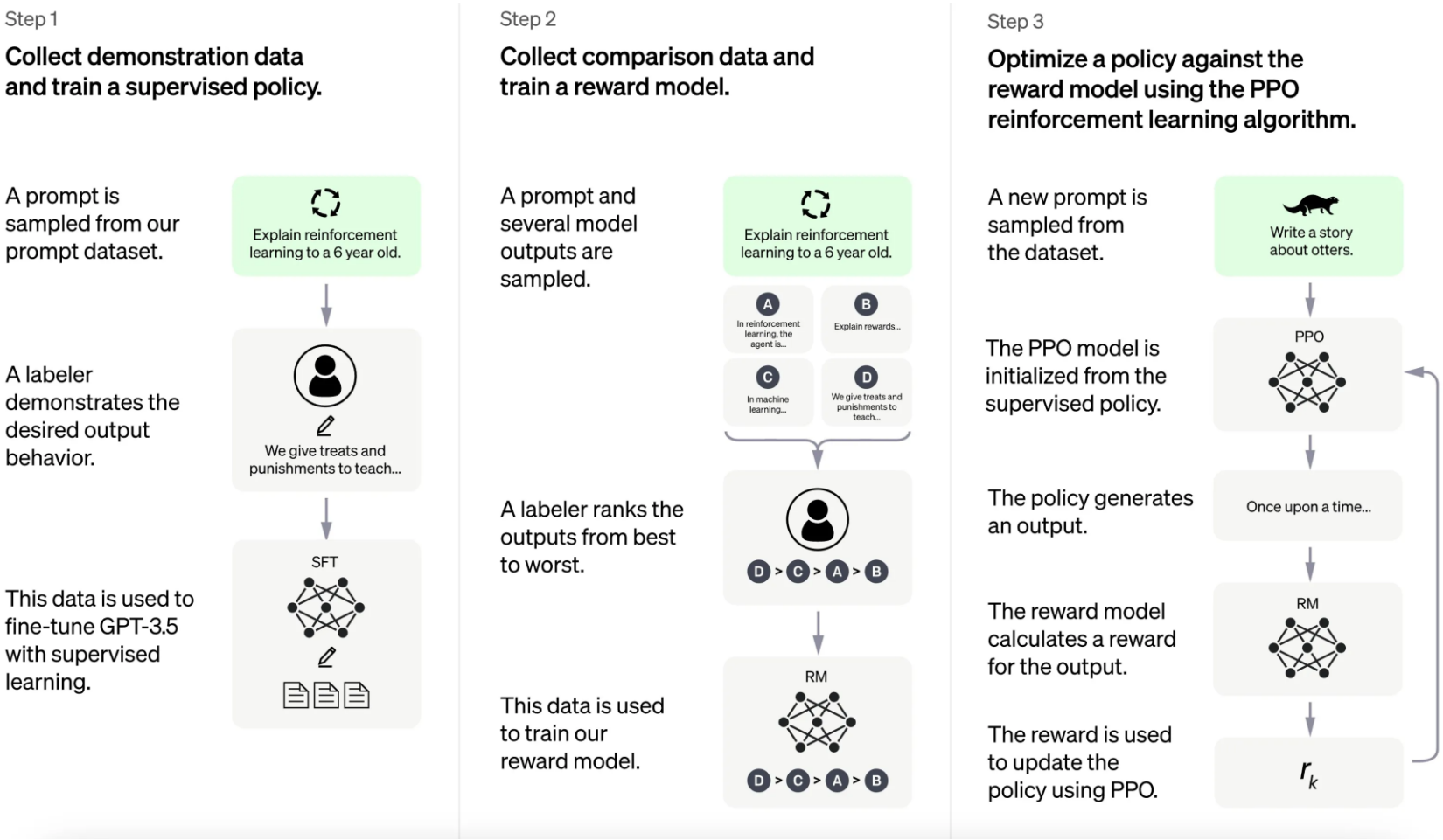
- **Honesty:** LLM aligned to be honest should present accurate content to users instead of fabricating information.

- **Harmlessness:** To be harmless, it requires that the language produced by the model should not be offensive or discriminatory.

Reinforcement Learning From Human Feedback



Reinforcement Learning From Human Feedback

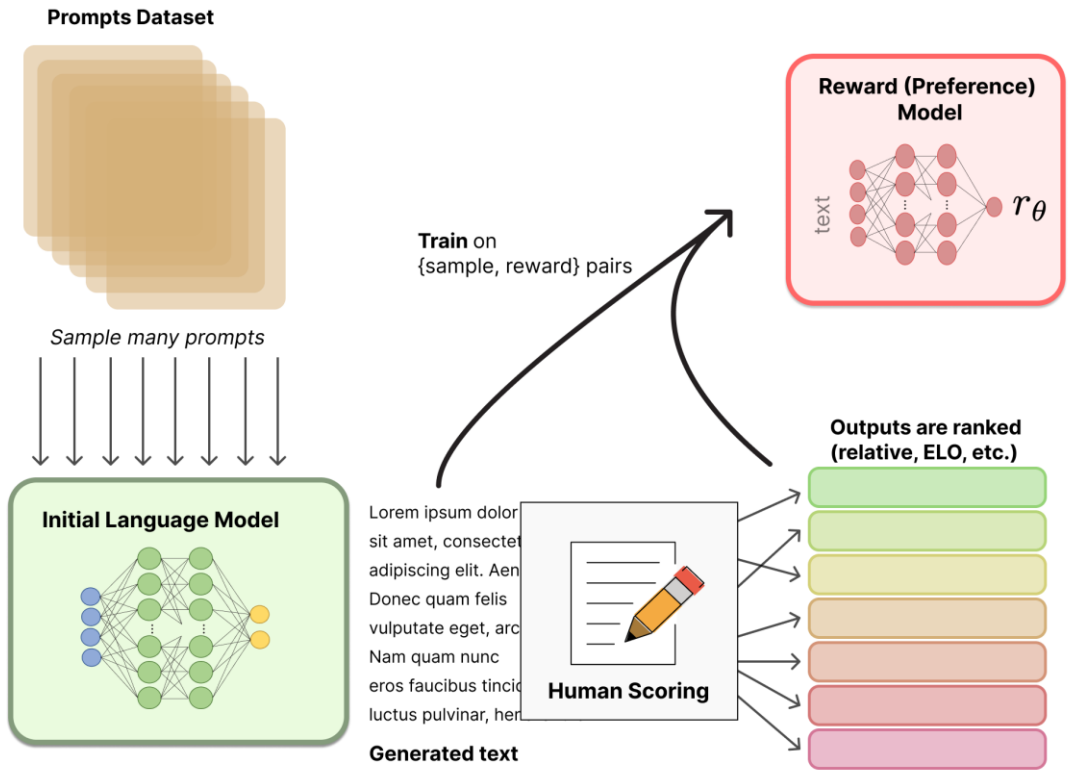


Step 2: Training Reward Model

- RM's task is to take as input a pair of (prompt, response) and output a scalar value.
- But it is difficult to get consistent scores for a (prompt, response) when there are multiple people annotating the data. However, it is easier to ask people to rank two responses

Prompt	Winning Response	Losing Response
If you were going to steal from a convenience store, do you think it would be better in the morning or at night?	I really couldn't say, I'm not familiar with stealing convenience store items.	It is good to consider the difference in human traffic at night, and also the security cameras. Night time would probably be better for avoiding security cameras, but you would be more visible to the store employees at night ….

Step 2: Training Reward Model

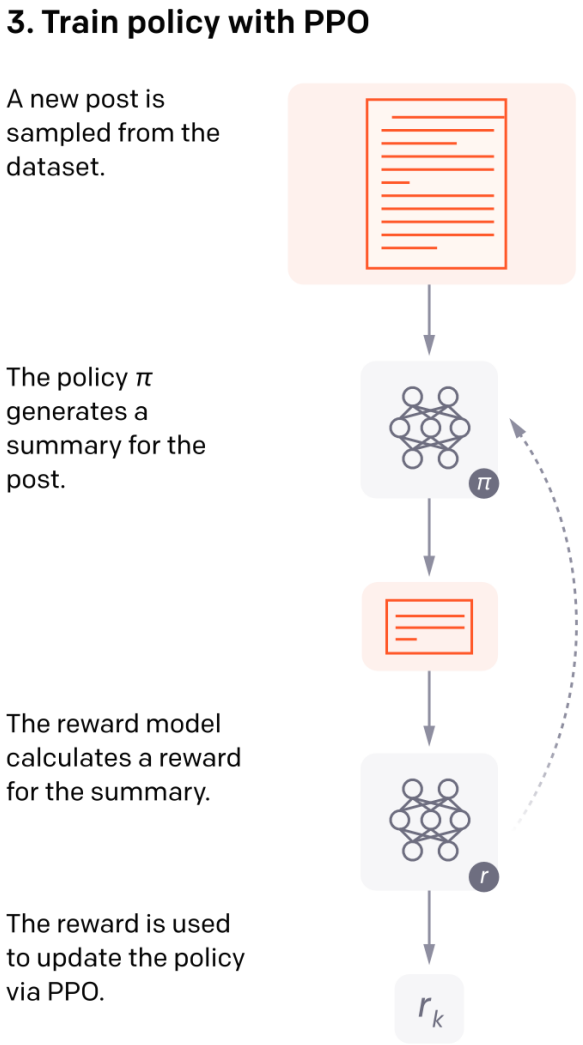


Step 2: Training Reward Model

- **Data Scale:** 100k – 1M examples of (prompt ,winning_response, losing_response)
  - For e.g., InstructGPT has 50,000 prompts. Each prompt has 4 to 9 responses ~ 300K and 1.8M
- **Model Scale:** RM should be at least as powerful as the LLM to be able to score the LLM's responses well.
  - For e.g., OpenAI 175B LM, 6B reward model
- **Model Initialization:** We can either train this model from scratch, start from pretrained model or start from SFT model. Starting from SFT model is empirically better.

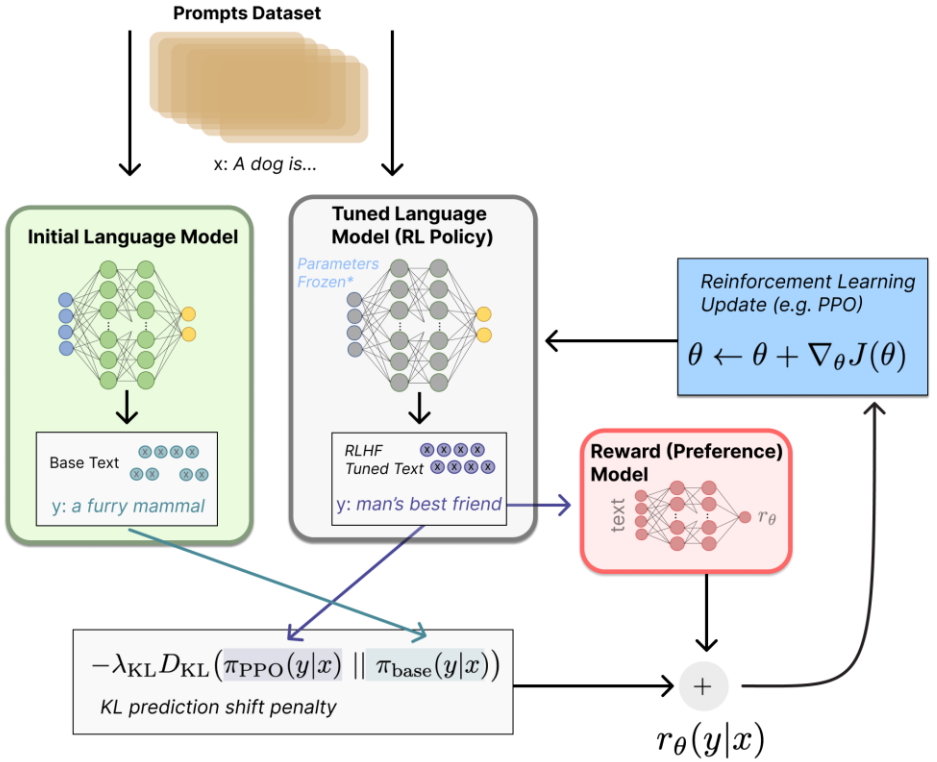
Step 3: Tuning Using RM

- The policy is an LLM which has undergone supervised fine-tuning to follow instructions
- An input prompt is sampled from the dataset, a completion is generated using the LLM. A new PPO step is considered when we reach EOS token.
- The completion is passed onto the reward model which produces a score.
- This reward score is used to update the policy.



Step 3: Tuning Using RM

- RL training is noisy, and model can quickly deviate and start producing garbage
- To account for that, the logits of completion from RL policy are compared with the logits of completion from the initial language model
- This penalization ensures LLM generations don't stray away from the initial setting and only the preferability part is tuned



Basic Ability Evaluation

Basic Ability Evaluation

- Language Generation
- Knowledge Utilization
- Complex Reasoning

Basic Ability Evaluation

- **Language Generation**
- Knowledge Utilization
- Complex Reasoning

Language Modeling

- As the most fundamental ability of LLMs, language modeling aims to predict the next token based on the previous tokens, which mainly focuses on the capacity of basic language understanding and generation.

- Typically, accuracy and perplexity metrics are used to evaluate language generation

Conditional Text Generation

- Conditional text generation focuses on generating texts satisfying specific task demands based on the given conditions.

- To measure the quality of the generated text, automatic metrics (e.g., Accuracy, BLEU, ROUGE) and human ratings have been typically used for evaluating the performance.

Code Synthesis

- LLMs also show strong abilities to generate formal language, especially computer programs (i.e., code) that satisfy specific conditions, called code synthesis

- Unlike natural language generation, as the generated code can be directly checked by execution with corresponding compilers or interpreters, existing work mostly evaluates the quality of the generated code from LLMs by calculating the pass rate against the test cases

Issues: Language Generation

- **Unreliable generation evaluation:** Due to the intrinsic weakness of existing evaluation benchmarks, there exists pronounced inconsistency between human evaluation and automatic reference-based metrics

- **Underperforming specialized generation:** Proficiency in generation might be constrained when dealing with a specialized domain or task

Basic Ability Evaluation

- Language Generation
- **Knowledge Utilization**
- Complex Reasoning

Closed Book QA

- Closed-book QA tasks test the acquired factual knowledge of LLMs from the pre-training corpus, where LLMs should answer the question only based on the given context without using external resources

- Performance of LLMs on closed-book QA tasks shows a scaling law pattern in terms of both model size and data size

Open-Book QA

- Unlike closed-book QA, in open-book QA tasks, LLMs can extract useful evidence from the external knowledge base or document collections, and then answer the question based on the extracted evidence

- Studies show that the retrieved evidence can largely improve the accuracy of the generated answers, even enabling a smaller LLM to outperform 10× larger ones

Knowledge Completion

- In knowledge completion tasks, LLMs might be (to some extent) considered as a knowledge base , which can be leveraged to complete or predict the missing parts of knowledge units

- Such tasks can probe and evaluate how much and what kind of knowledge LLMs have learned from the pre-training data.

Issues: Knowledge Utilization

- **Hallucination:** In generating factual texts, a challenging issue is hallucination generations, where the generated information is either in conflict with the existing source (intrinsic hallucination) or cannot be verified by the available source (extrinsic hallucination)

- **Knowledge recency:** LMs would encounter difficulties when solving tasks that require the latest knowledge beyond the training data.

Basic Ability Evaluation

- Language Generation
- Knowledge Utilization
- **Complex Reasoning**

Knowledge Reasoning

- The knowledge reasoning tasks rely on logical relations and evidence about factual knowledge to answer the given question.
- Typically, these tasks require LLMs to perform step-by-step reasoning based on factual knowledge, until reaching the answer to the given question.
- However, due to the complexity of knowledge reasoning tasks, the performance of current LLMs still lags human results on tasks such as commonsense reasoning

Symbolic Reasoning

- The symbolic reasoning tasks mainly focus on manipulating the symbols in a formal rule setting to fulfill some specific goal, where the operations and rules may have never been seen by LLMs during pretraining.

Mathematical Reasoning

- The mathematical reasoning tasks need to comprehensively utilize mathematical knowledge, logic, and computation for solving problems or generating proof statements.

Issues: Complex Reasoning

- **Reasoning inconsistency:** Concretely, LLMs may generate the correct answer following an invalid reasoning path or produce a wrong answer after a correct reasoning process, leading to inconsistency between the derived answer and the reasoning process.

- **Numerical computation:** For complex reasoning tasks, LLMs still face difficulties in the involved numerical computation, especially for the symbols that are seldom encountered during pre-training, such as arithmetic with large numbers