COLUMBIA UNIVERSITY

EECS 6895 Adv. Big Data and AI

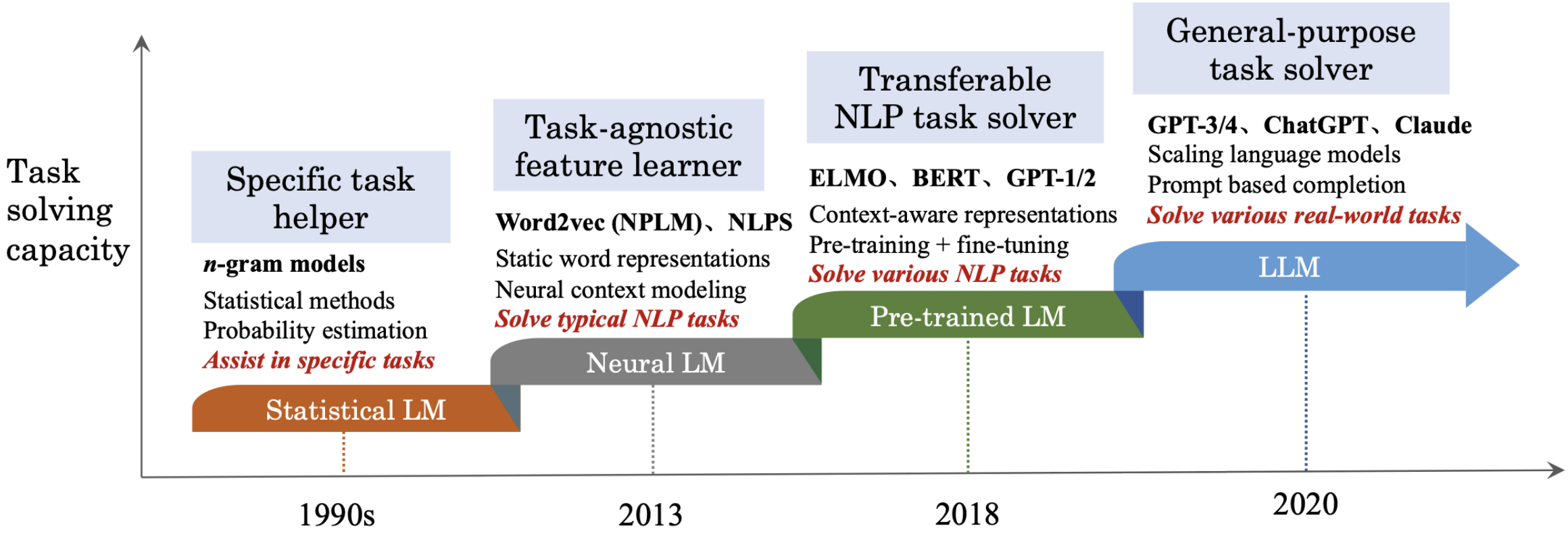
Lecture 3: LLM Step-by-Step

Prof. Ching-Yung Lin

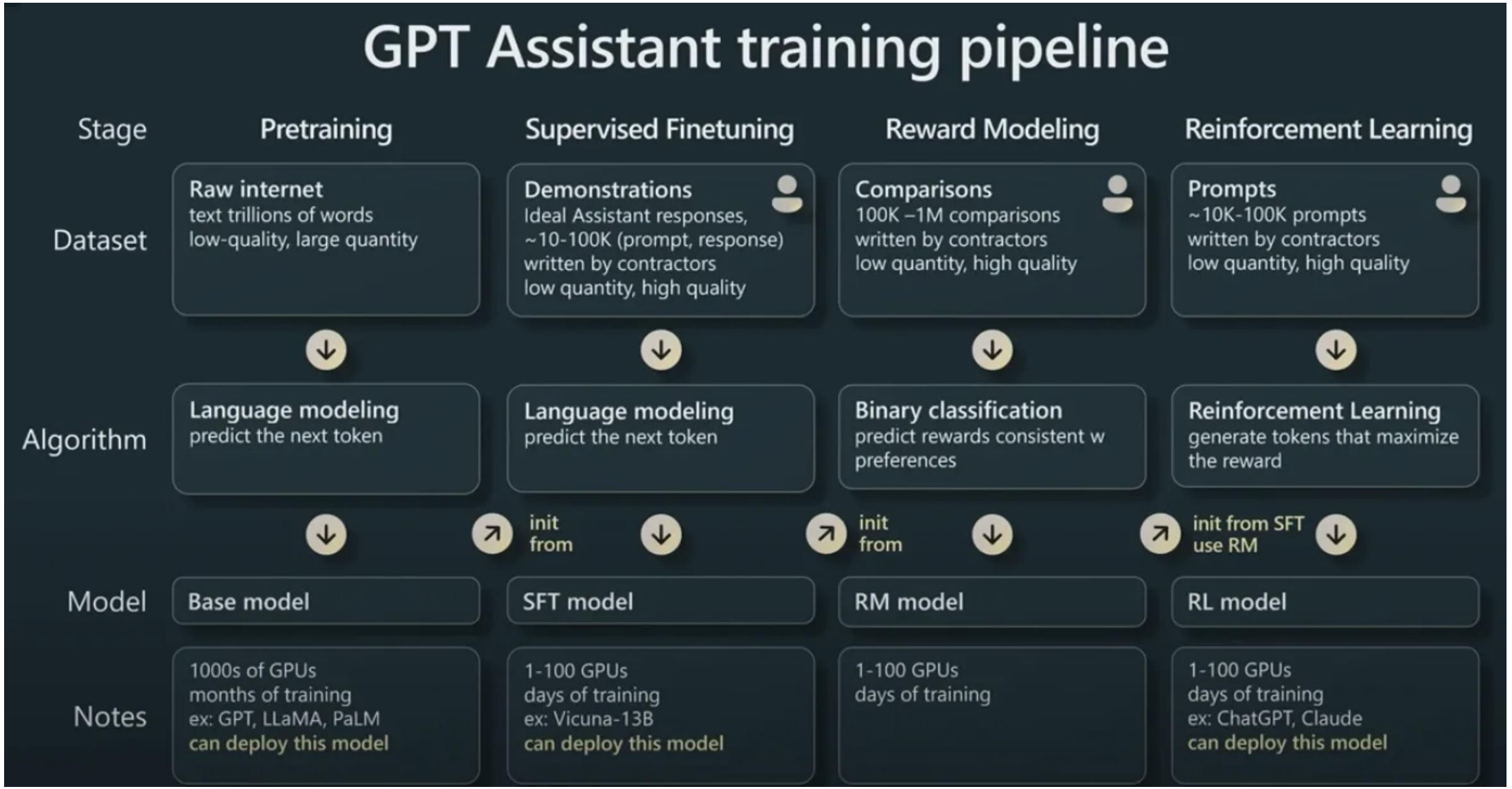
Columbia University

February 4th, 2025

Task Solving Capabilities



Building LLMs for AI Assistant



https://medium.com/@tonytong.ai/andrej-karpathys-keynote-at-microsoft-build-2023-8b45a2bbf22e

Pretraining

- Building up Base Model

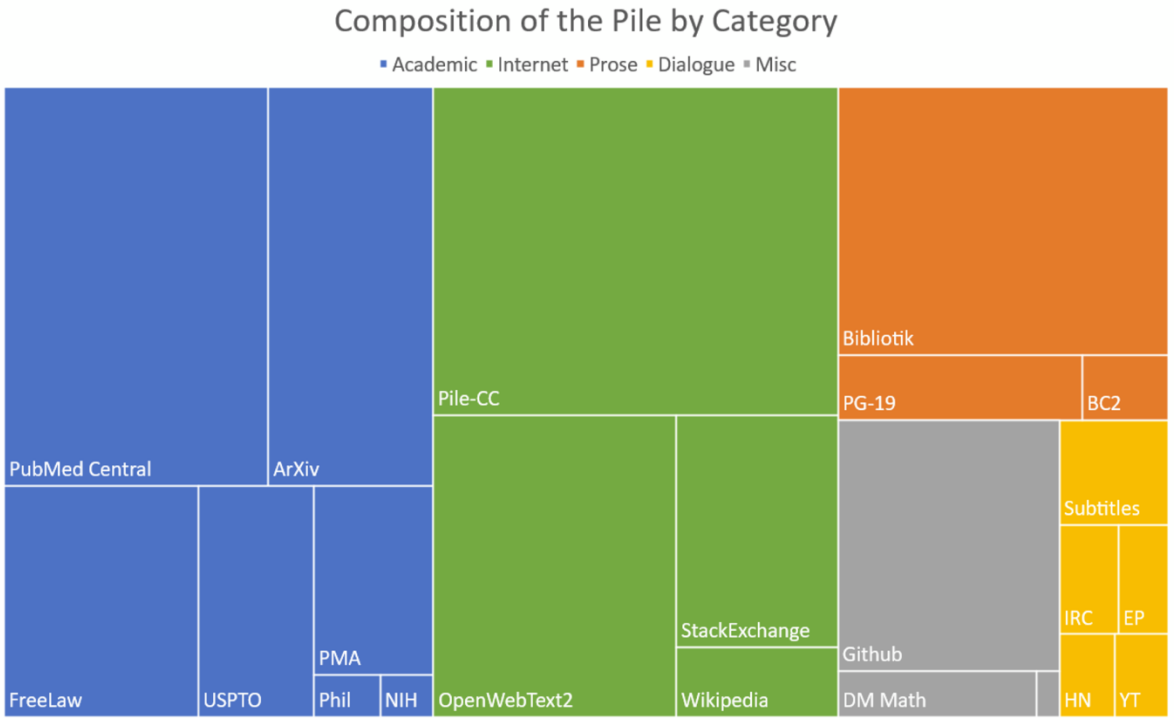
- Make AI to learn “World Knowledge”
  - Factual Knowledge
  - Common Sense

- Large-scale datasets were gathered from sources like CommonCrawl, Wikipedia, GitHub, and others.

- Text is tokenized into sequences of integers, which serve as the input for the transformer model.

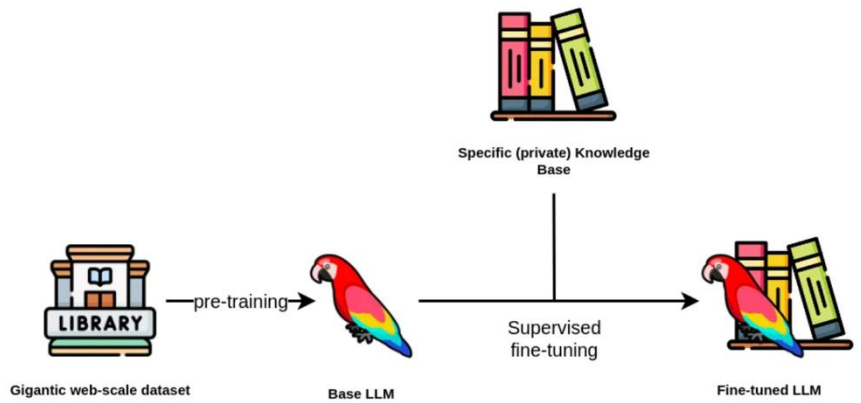
- GPT-3 Training
  - 17.5B model using 1000 GPUs
    - Nvidia A100 80GB GPU for 2 months

- BLOOM Model
  - 17.5B using 384 GPUs for 3.5 months

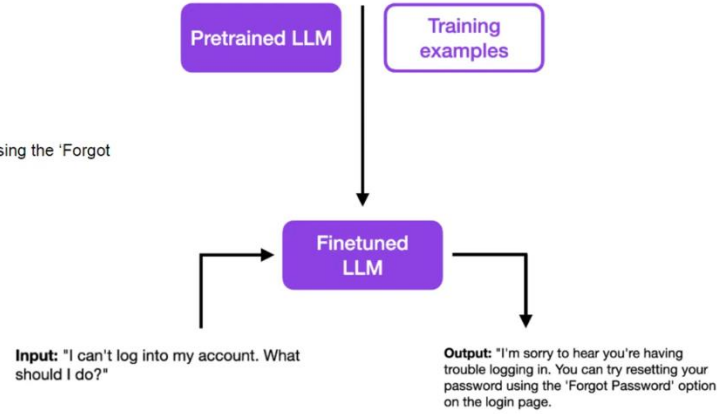


Supervised Fine-Tuning (SFT)

- Small, high-quality datasets were collected, consisting of prompt and ideal response pairs created by human.
- Language modelling is still used.
- Capable of Q&A, translation, and writing.
- The final model retains the general information from its original knowledge base but has an in-depth understanding of the particular domain.
- Common SFT techniques:
  - Full Fine-Tuning
  - Parameter-Efficient Fine-Tuning (PEFT)
  - Instruction Fine-Tuning
- Fine-Tuning vs. Retrieval Augmented Generation (RAG)



Supervised Finetuning on LLMs. Source: Neo4j



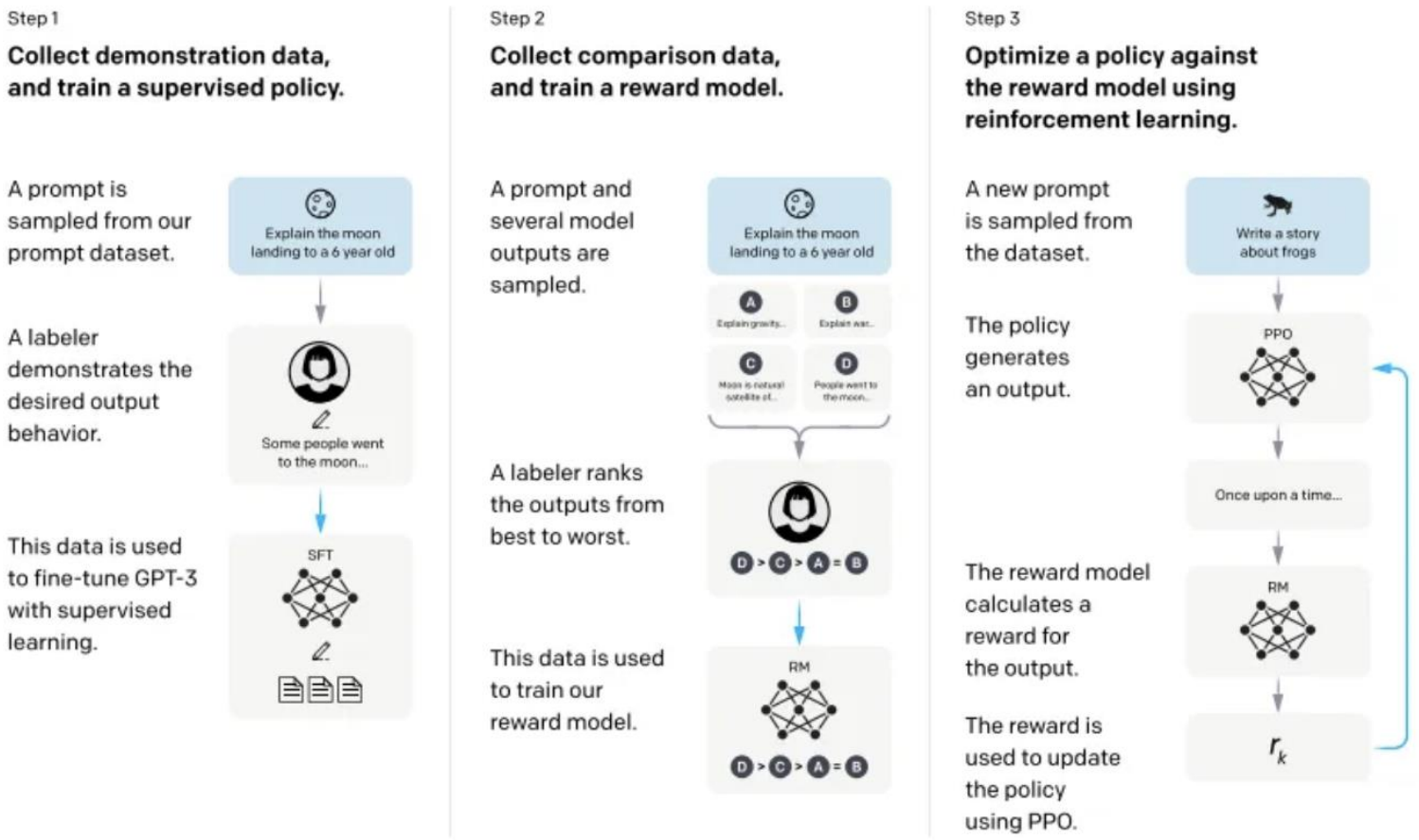
https://medium.com/mantisnlp/supervised-fine-tuning-customizing-llms-a2c1edbf22c3

Outcome of Supervised Fine-Tuning (SFT)

- Essential Instruction and Context Reasoning Capability.
- Achieved some level of QA, reading, translation, and coding.
- Needs 10s of GPUs using several days.
- Many ChatGPT-like models are like this.
- Some evaluations achieved 90% of ChatGPT capabilities.
- Can be used as a basic assistant.

Reward Modeling

- Comparisons of multiple completions from the SFT model, ranked by human.
- The transformer model is trained to predict rewards for each completion based on the ground truth from the human rankings.
- Reward models allow scoring the quality of any arbitrary completion for a given prompt.

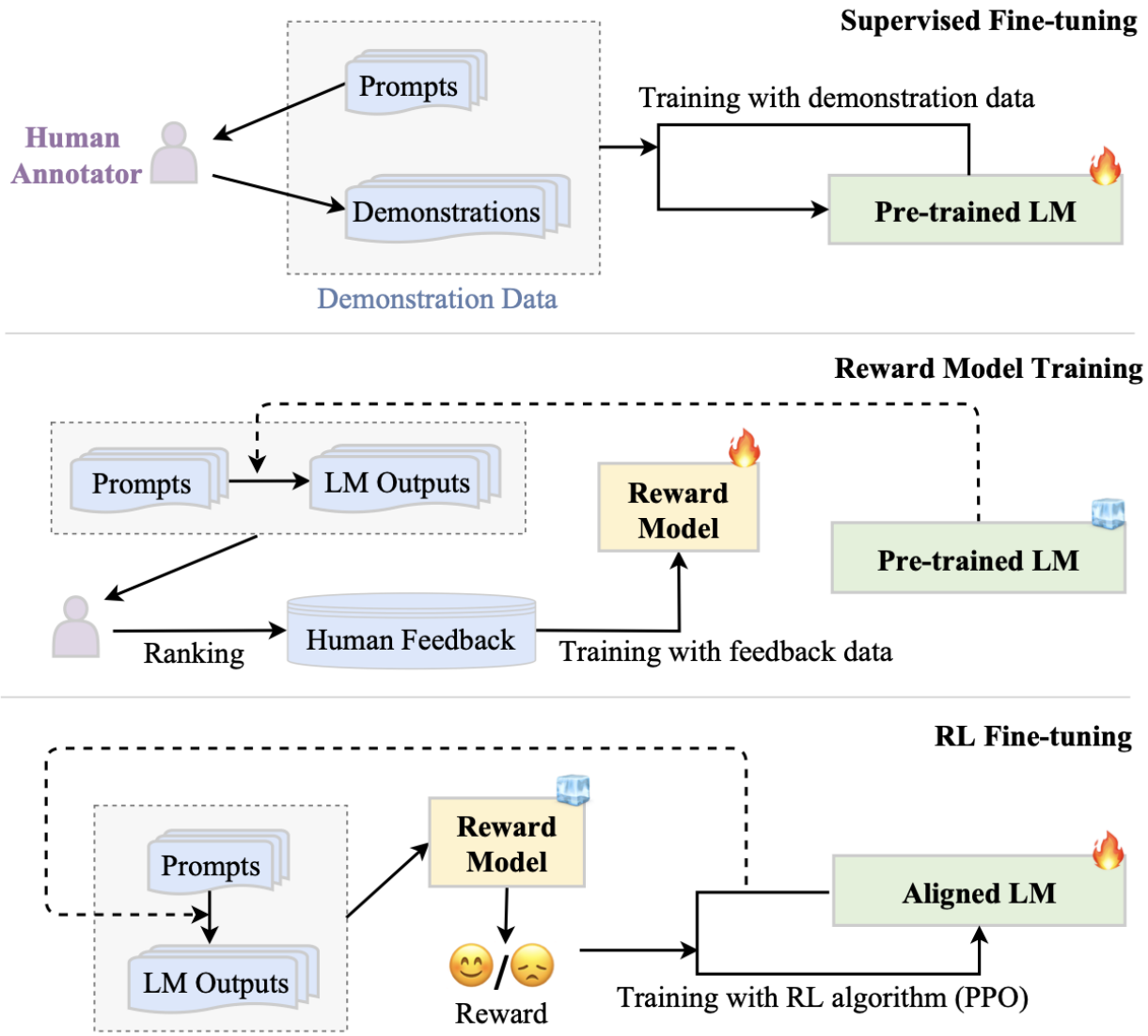


A diagram illustrating the three steps of training

https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf

Reinforcement Learning from Human Feedback

- Reinforcement Learning is performed with respect to the fixed reward model.
- Completions generated by the model are weighted by the rewards indicated by the reward model.
- Model learns to generate completions that score high according to the reward model.



Base Models vs Assistant Models

Base Models:

- Pretrained on large-scale datasets.
- Document completion focus.
- Less Task-Specific

Assistant Models:

- Fine-tuned for specific applications.
- Task-oriented.
- Better performance for tasks.

Chain of Thoughts

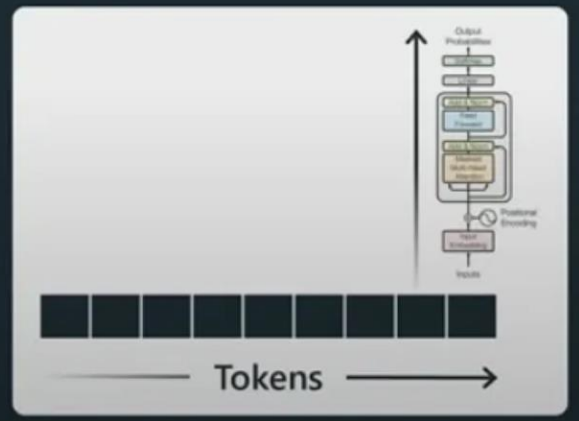
Chain of thought

"Models need tokens to think"

Break up tasks into multiple steps/stages

Prompt them to have internal monologue

Spread out reasoning over more tokens



(b) Few-shot-CoT

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?
A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?
A:

(Output) The juggler can juggle 16 balls. Half of the balls are golf balls. So there are 16 / 2 = 8 golf balls. Half of the golf balls are blue. So there are 8 / 2 = 4 blue golf balls. ***The answer is 4.*** ✓

(d) Zero-shot-CoT (Ours)

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?
A: ***Let's think step by step.***

(Output) There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls. ✓

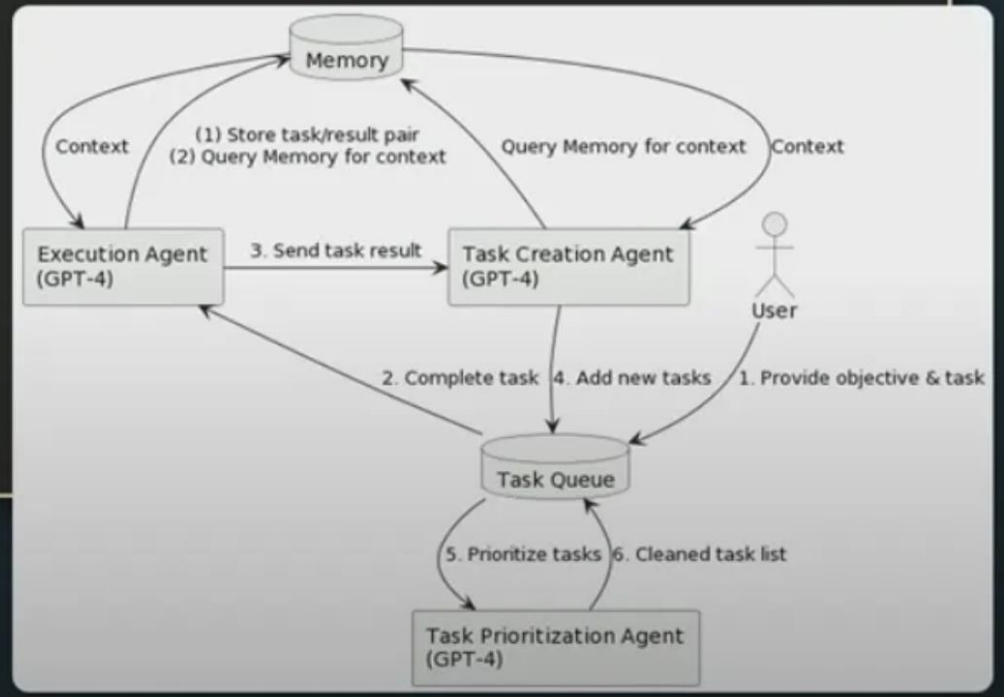
[Chain-of-Thought Prompting Elicits Reasoning in Large Language Models, Wei et al. 2022]
[Large Language Models are Zero-Shot Reasoners, Kojima et al. 2022]

Agents

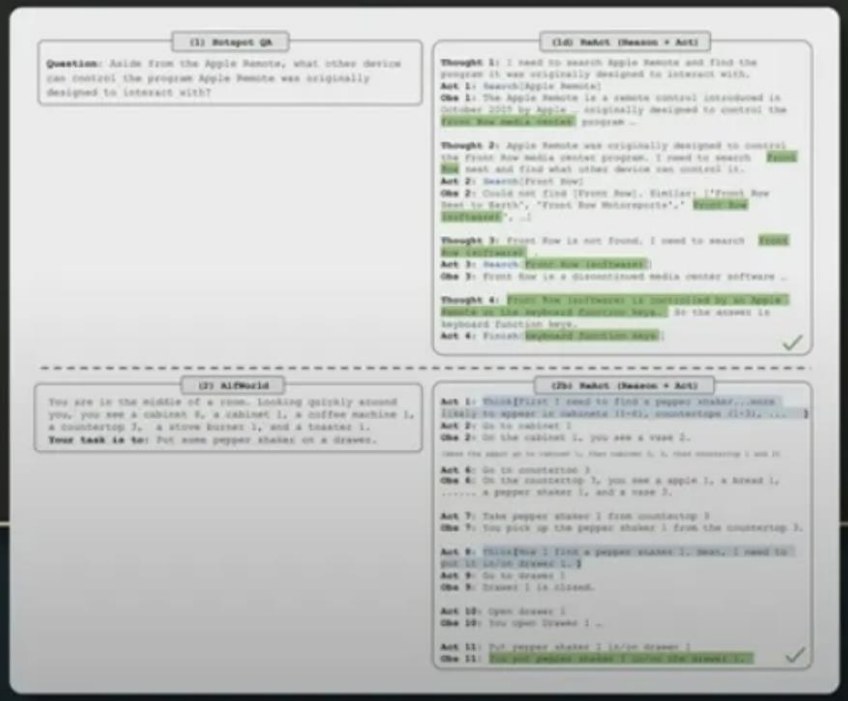
Chains / Agents

Think less "one-turn" Q&A, and more chains, pipelines, state machines, agents.

[AutoGPT]



[ReAct: Synergizing Reasoning and Acting in Language Models, Yao et al. 2022]



Agents

- Breaking down tasks into smaller steps
- Linking Prompts together.
- Chains:
  - Breaking down a complex task into a series of smaller, more manageable steps.
- Designing prompts and interactions that help the model understand its role in solving a particular problem.
- Involves asking the model to think step-by-step
- Pipelines:
  - The sequential process of using multiple prompts or tools to solve a problem.
  - In a pipeline, the output of one prompt or tool is used as the input for the next.
  - This can involve using external tools like calculators or code interpreters alongside the LLM.
- State Machines:
  - Using a structured approach to manage the flow of information and control behavior of the LLM.
  - Create a more controlled and predictable interaction with the LLM.


To be discussed

Basic Concept of LLM

- Transformer
  - Embedded Layer
  - Attention Layer
  - Feedback
  - Residue Linkage
  - Encoder and Decoder Structure

- Pre-trained GPT
  - No-supervision Pre-training
  - Supervised Fine Tuning (SFT)
  - Supervised Fine Tunning based on HuggingFace

- Large Language Model Structure
  - Llama Language Model
  - Attention Optimization

LLM Pretraining Datasets

- Data Sources
  - General Data
  - Specialized Data
- Data Processing
  - Quality Filtering
  - Fault Detection and Removal
  - Privacy
  - Tokenization
- Impact of Data
  - Data Size
  - Data Quality
  - Data Diversity
- Open-Source Datasets
  - The Pile
  - ROOTS
  - RefinedWeb
  - SlimPajama

Distributed Training in LLM

- Overview of Distributed Training

- Strategies of parallel distributed training
  - Parallel Data
  - Parallel Model
  - Hybrid Training
  - Optimization of Memory

- Cluster Structure of Distributed Training
  - Hardware components of high performance computing
  - Structure of Parameter Tuning
  - De-centralized Architecture

- Training Examples
  - Basic concepts
  - Llama Distributed Training Examples

Supervised Tuning in LLM

- Prompt Learning and Concept Learning
  - Prompt Learning
  - Concept Learning
- High-Performance Model Tuning
  - LoRA
  - Variation of LoRA
- Model Context Window Expansion
  - Exploration of Position Coding
  - Exploration
- Instruction Data Construction
  - Manual Construction Instructions
  - Automatic Construction Instructions
  - Open-Source Instruction Datasets
- Chat SFT Examples

Reinforcement Learning in LLM

- Human-based Reinforcement Learning
  - Difference between Reinforcement Learning and Supervised Learning
  - Process of Human-based Reinforcement Learning

- Award Model
  - Data Collection
  - Model Training
  - Open-source training data

- Optimization of Near-End Strategies
  - Strategy Gradients
  - Optimization Estimations
  - Optimal Algorithms of Near-End Strategies

- MOSS-RLHF Examples
  - Award Model Training
  - PPO Fine Tuning

LLM Applications

- Reasoning
  - Prompt of Reasoning Chain
  - Prompt from Few to More
- Framework of LLM Applications
  - LangChain Core Module
  - Q&A Knowledge System Implementation
- Smart Agents
  - Smart Agents Infrastructure
  - Smart Agent Applications
- Multi-Modal Models
  - Model Architecture
  - Data Graining and Training Strategy
  - Multi-Modal Capability Examples
- LLM Reasoning Optimization
  - FastServe Architecture
  - vLLM Reasoning Architecture

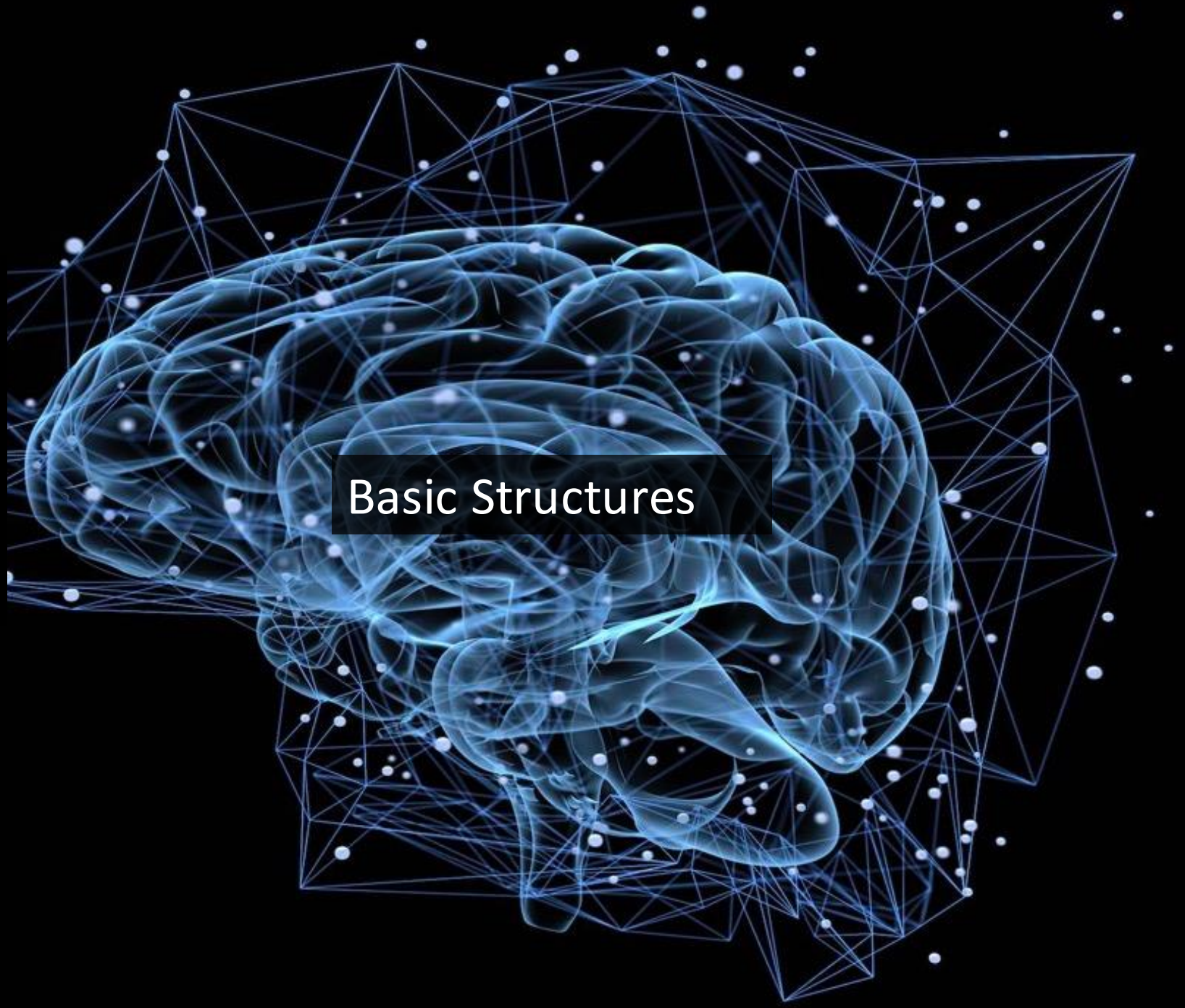
LLM Evaluations

- Model Evaluations

- LLM Evaluation Systems
  - Knowledge and Capability
  - Ethics and Safety
  - Vertical Area Evaluation

- Methodologies for LLM Evaluation
  - Evaluation Metrics
  - Evaluation Methodologies

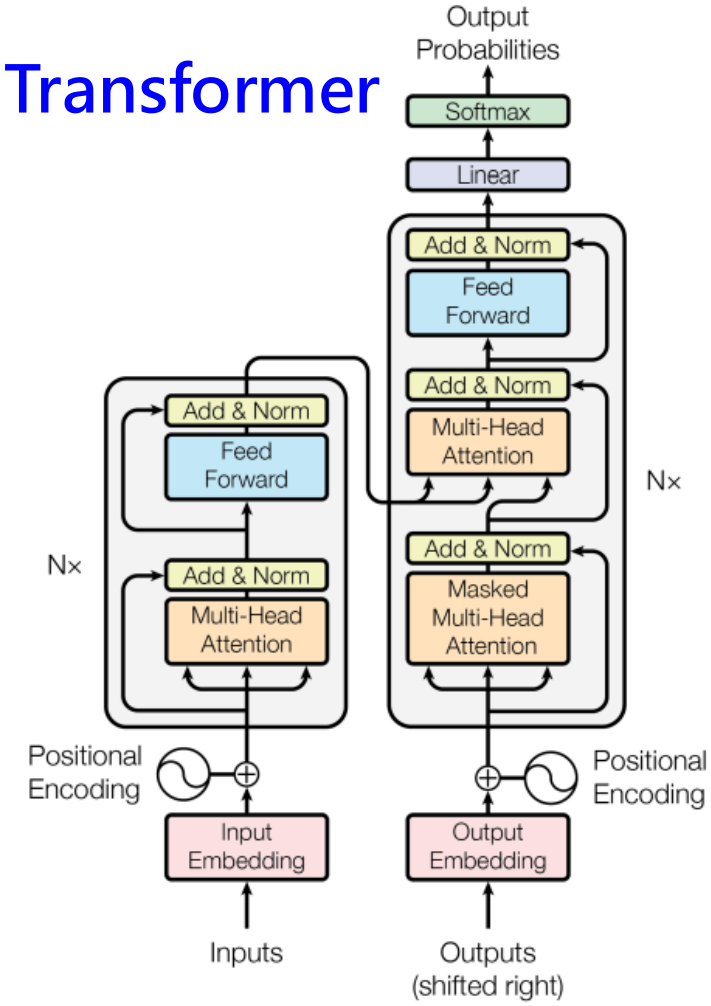
- LLM Evaluation Experiments
  - Basic Model Evaluation
  - SFT Model and RL Model Evaluation


Basic Structures

Transformer

Encoder

Transformer



Decoder

Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).

Input Embedding

- Convert each word to its corresponding vector
- Position Encoding:
  - Encode the position of each work in the sequence to a vector

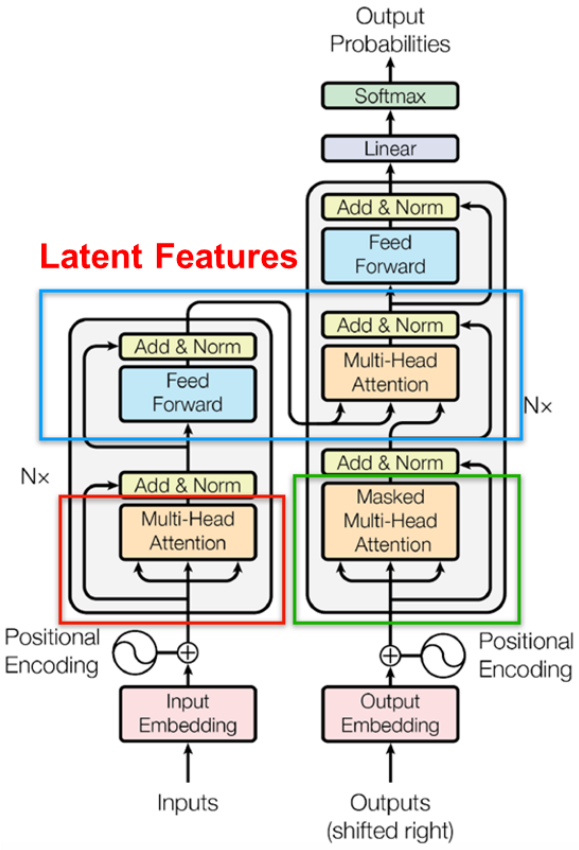
Self-Attention

- Build up relationships between source language and target language
- Multi-Head Attention

Transformer to GPT

Transformer

Input -> **Encoder** -> Latent Feature + Masked Output -> **Decoder** -> Output



An [redacted] a day keeps the doctor away

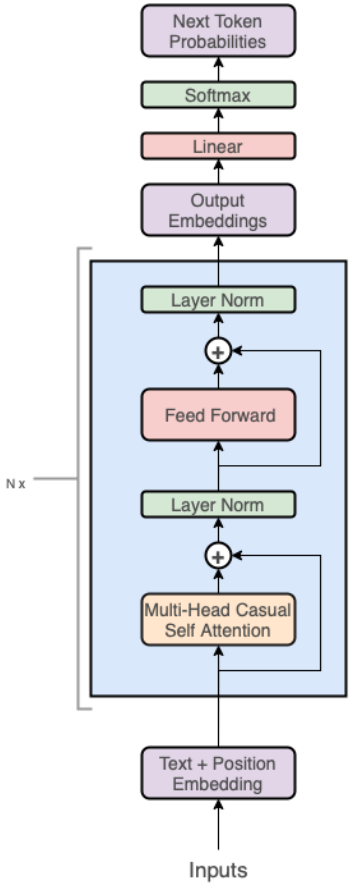
apple 95%
banana 5%

An apple a day keeps the doctor away

Masked Language Learning

GPT

Input -> **Decoder(with Casual mask)** -> shift Output



An

apple 99%
almond 1%

An apple

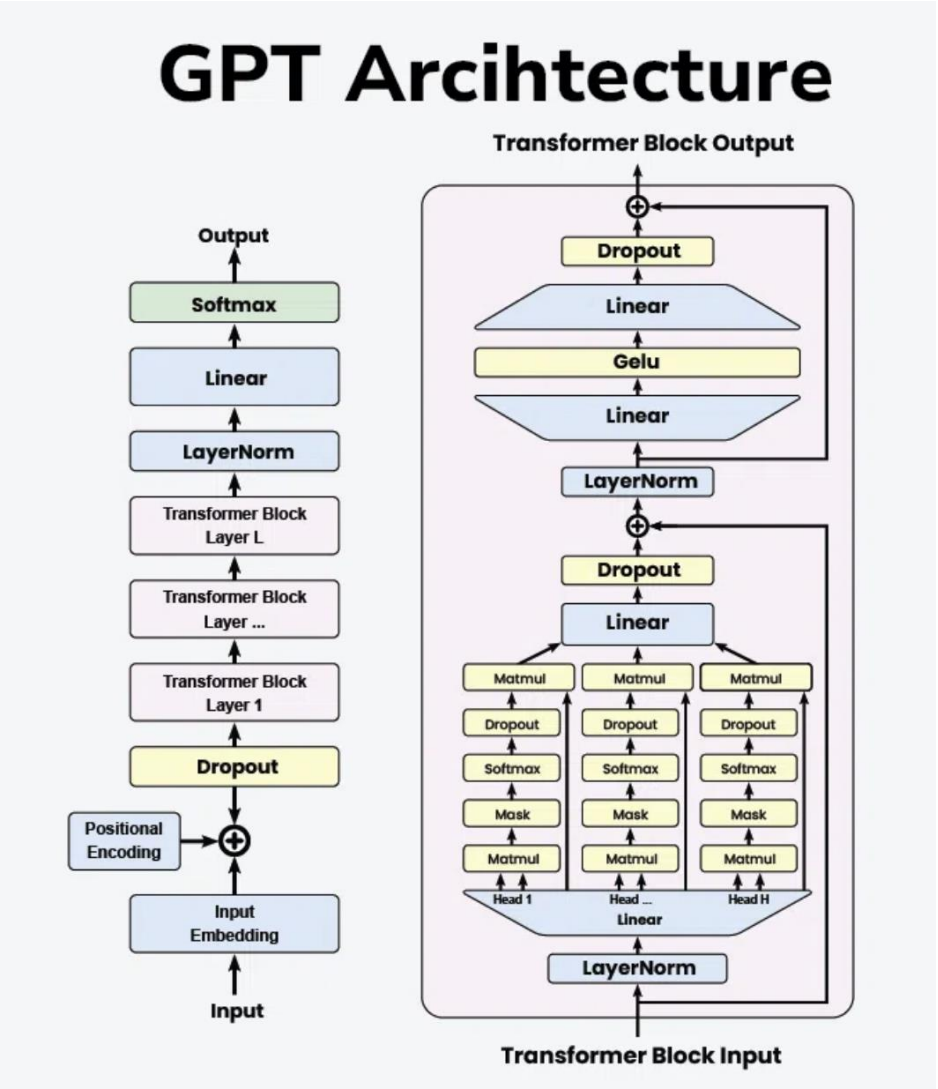
a 99%
watch 1%

An apple a

Autoregressive Learning

Key elements of GPT

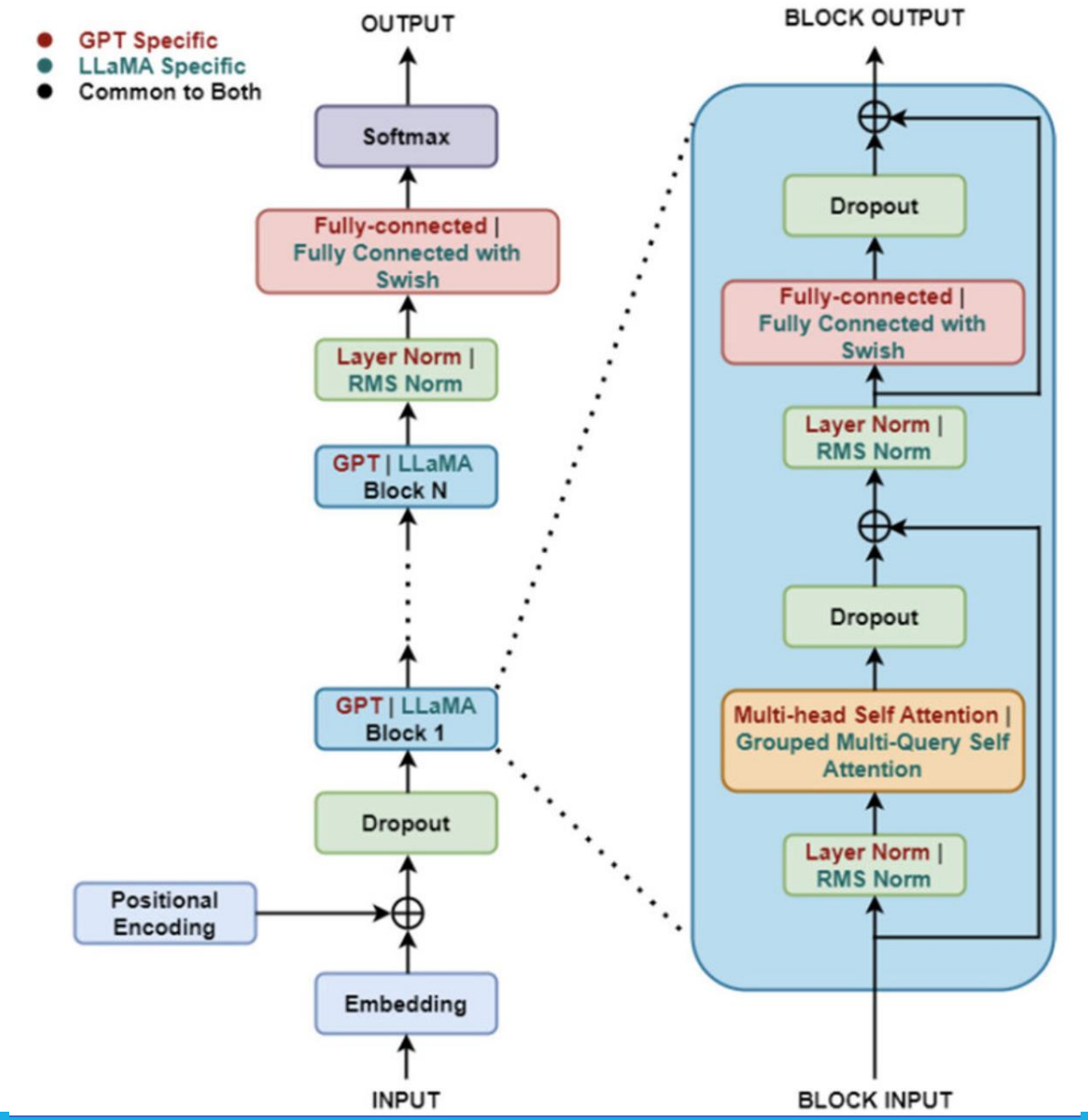
- Input Embedding
- Position Encoding
- Dropout Layer
- Transformer Blocks
- Layer Stack
- Final Layers



- **Unsupervised Pre-Training**
- **Downstream Task Fine-tuning**

https://www.geeksforgeeks.org/introduction-to-generative-pre-trained-transformer-gpt/

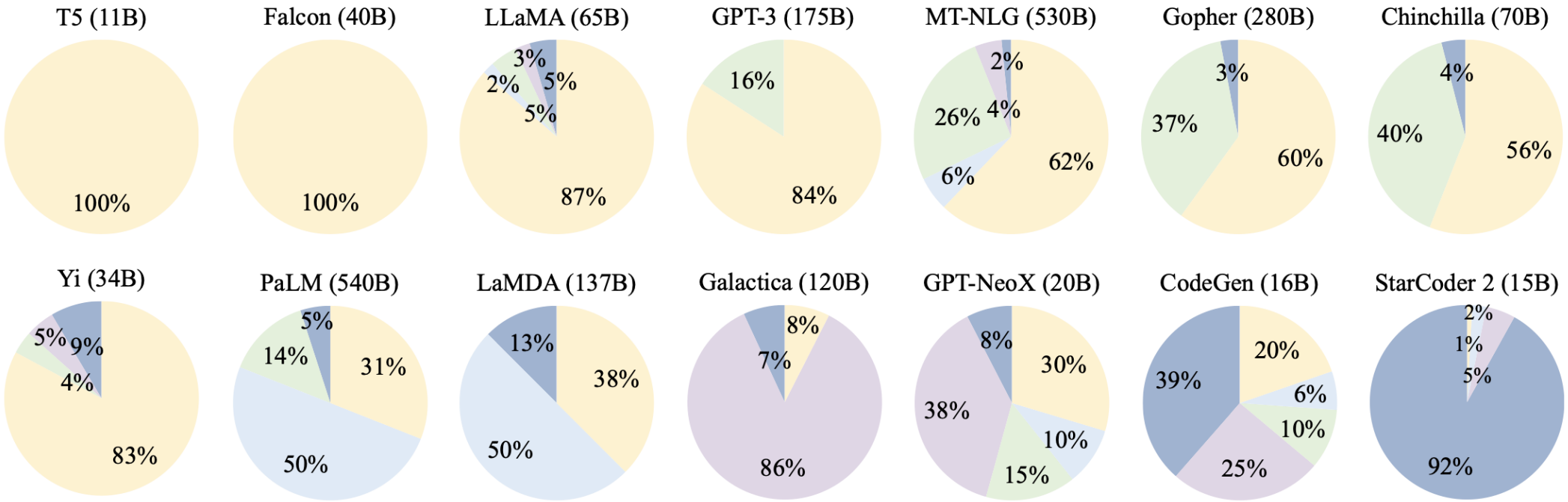
Llama Model Structure



https://www.researchgate.net/figure/Basic-architecture-of-GPT-and-LLaMA-models-with-differences_fig1_381112819


Data Processing

Data Ratios



Webpages	C4 (800G, 2019), OpenWebText (38G, 2023), Wikipedia (21G, 2023)
Conversation Data	the Pile - StackExchange (41G, 2020)
Books & News	BookCorpus (5G, 2015), Gutenberg (-, 2021), CC-Stories-R (31G, 2019), CC-NEWES (78G, 2019), REALNEWs (120G, 2019)
Scientific Data	the Pile - ArXiv (72G, 2020), the Pile - PubMed Abstracts (25G, 2020)
Code	BigQuery (-, 2023), the Pile - GitHub (61G, 2020)

https://arxiv.org/pdf/2303.18223

Classic Data Processing Procedure

Raw Corpus

Filtering & Selection

- Language Filtering
- Metric Filtering
- Statistic Filtering
- Keyword Filtering

Alice is writing a paper about LLMs. ~~#$^&~~ Alice is writing a paper about LLMs.

De-duplication

- Sentence-level
- Document-level
- Set-level

Alice is writing a paper about LLMs. ~~Alice is writing a paper about LLMs.~~

Privacy Reduction

- Detect Personality Identifiable Information (PII)
- Remove PII

`Replace('Alice')` is writing a paper about LLMs.

Tokenization

- Reuse Existing Tokenizer
- SentencePiece
- Byte-level BPE

`Encode('[Somebody] is writing a paper about LLMs.')`

Ready to pre-train!

32, 145, 66, 79, 12, 56, ...

https://arxiv.org/pdf/2303.18223

General Data

- Webpages: ClueWeb09, ClueWeb12, SogouT-16, CommonCrawl, etc.
- Conversation Text: PushShift.io Reddit, Ubuntu Dialogue Corpus, Douban Conversation Corpus, Chromium Conversations Corpus, etc.
- Book: Books3 and BookCorpus2 in the Pile dataset.

https://www.geeksforgeeks.org/introduction-to-generative-pre-trained-transformer-gpt/

Specialized Data

- Multilingual Text:
  - BLOOM pretraining dataset includes 46 languages
  - PaLM's pretraining dataset includes 122 languages
- Scientific Text:
  - Papers, wiki, etc.
  - For instance:
    - Functions to be represented by LaTex.
    - Chemical Structures are represented by SMILES (Simplified Molecular Input Line Entry System).
- Code:
  - Stack Exchange for QA
  - GitHub QA

Data Quality Filtering

- Quality of training data has a big impact to the results.

- Classification based filtering.
  - Adopted by GPT-3, PaLM, Glam, etc.
  - Using high quality data to train.

- Heuristic-based filtering:
  - Language-based filtering
  - Metrics-based filtering
  - Statistic-based filtering
  - Keyword-based filtering

- Text Quality Evaluation:
  - Previously used by search engine, social media, recommendation, advertising, etc.

- Text Duplicate Detection

Personally Identifiable Information (PII) Removal

- Rule –based approaches
- Detect names, address, telephone, etc.
- Using Transformer model.
- Using Translations.
- Reference BigScience ROOTS Corpus

Major Datasets

- The Pile (see Lecture 2)

- ROOTS (Responsible Open-Science Open-collaboration Text Sources): 46 languages and 13 coding languages.

- RefinedWeb:
  - Repetition Removal
  - Document-wise Filtering
  - Line-wise Correction
  - Fuzzy Deduplication
  - Exact Deduplication
  - URL Deduplication

- SlimPajama:
  - NFC normalization
  - Filtering short documents
  - Deduplication
  - MinHash Generation
  - Duplicated Graph Construction & Search for Connected Components