

Text Generation with Large Language Models (LLMs)

1121AITA08

MBA, IM, NTPU (M5265) (Fall 2023)

Tue 2, 3, 4 (9:10-12:00) (B3F17)



<https://meet.google.com/miy-fbif-max>



Min-Yuh Day, Ph.D,
Associate Professor

Institute of Information Management, National Taipei University

<https://web.ntpu.edu.tw/~myday>



Syllabus

Week Date Subject/Topics

- 1 2023/09/13 Introduction to Artificial Intelligence for Text Analytics
- 2 2023/09/20 Foundations of Text Analytics:
Natural Language Processing (NLP)
- 3 2023/09/27 Python for Natural Language Processing
- 4 2023/10/04 Natural Language Processing with Transformers
- 5 2023/10/11 Case Study on Artificial Intelligence for Text Analytics I
- 6 2023/10/18 Text Classification and Sentiment Analysis

Syllabus

Week Date Subject/Topics

7 2023/10/25 Multilingual Named Entity Recognition (NER)

8 2023/11/01 Midterm Project Report

9 2023/11/08 Text Similarity and Clustering

10 2023/11/15 Text Summarization and Topic Models

11 2023/11/22 Text Generation with Large Language Models (LLMs)

12 2023/11/29 Case Study on Artificial Intelligence for Text Analytics II

Syllabus

Week Date Subject/Topics

13 2023/12/06 Question Answering and Dialogue Systems

14 2023/12/13 Deep Learning, Generative AI, Transfer Learning,
Zero-Shot, and Few-Shot Learning for Text Analytics

15 2023/12/20 Final Project Report I

16 2023/12/27 Final Project Report II

17 2024/01/03 Self-learning

18 2024/01/10 Self-learning

Text Generation with Large Language Models (LLMs)

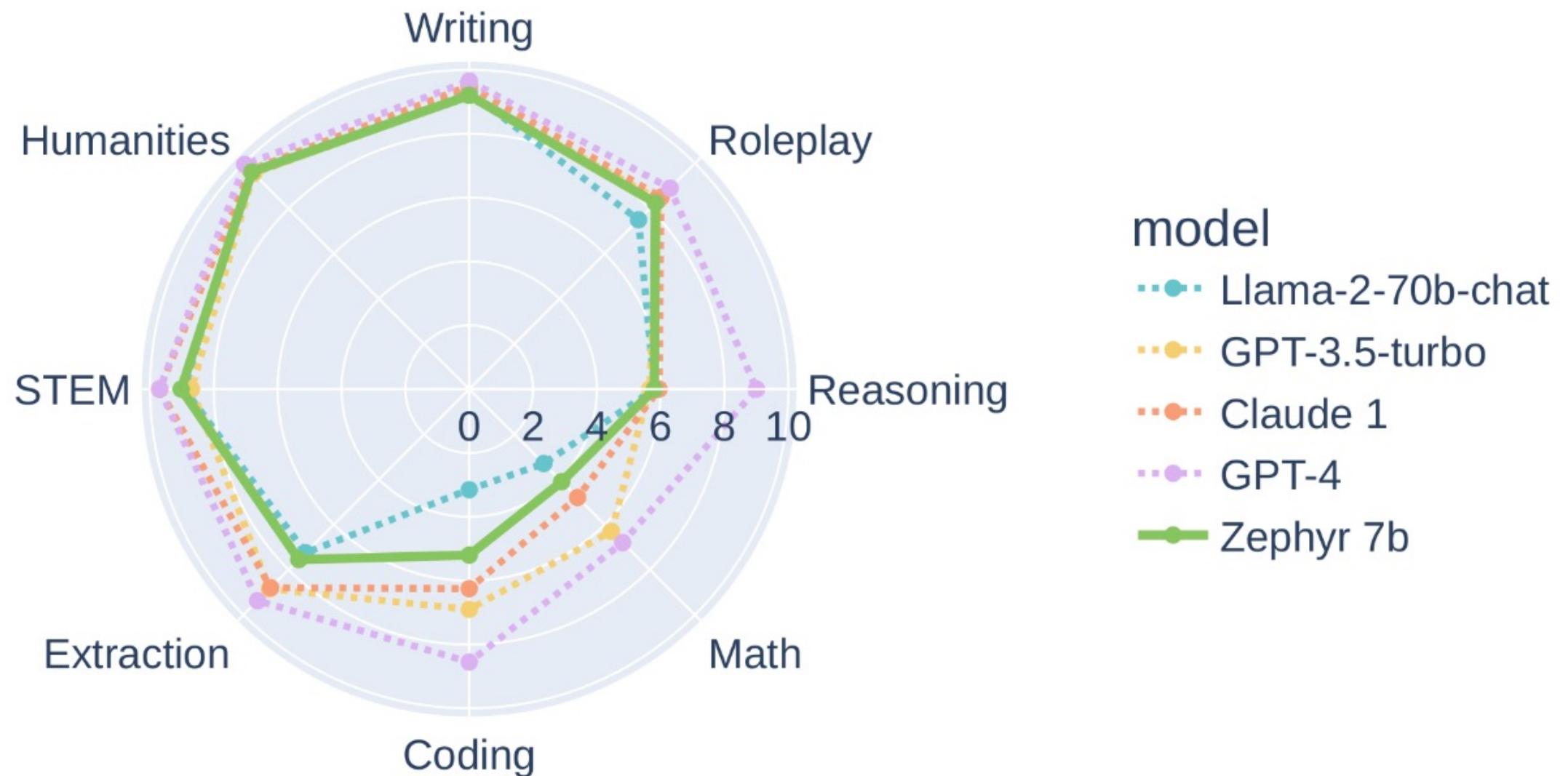
Outline

- **Text Generation**
- **Large Language Models (LLMs)**
- **Prompt Engineering**
- **Fine-tuning**
- **Retrieval Augmented Generation (RAG)**

Zephyr-7B- β , Llama2-Chat-70B, GPT-4

Model	Size	Alignment	MT-Bench (score)	AlpacaEval (win rate %)
StableLM-Tuned- α	7B	dSFT	2.75	-
MPT-Chat	7B	dSFT	5.42	-
Xwin-LMv0.1	7B	dPPO	6.19	87.83
Mistral-Instructv0.1	7B	-	6.84	-
Zephyr-7b- α	7B	dDPO	6.88	-
Zephyr-7b-β 	7B	dDPO	7.34	90.60
Falcon-Instruct	40B	dSFT	5.17	45.71
Guanaco	65B	SFT	6.41	71.80
Llama2-Chat	70B	RLHF	6.86	92.66
Vicuna v1.3	33B	dSFT	7.12	88.99
WizardLM v1.0	70B	dSFT	7.71	-
Xwin-LM v0.1	70B	dPPO	-	95.57
GPT-3.5-turbo	-	RLHF	7.94	89.37
Claude 2	-	RLHF	8.06	91.36
GPT-4	-	RLHF	8.99	95.28

Zephyr-7B- β , Llama2-Chat-70B, GPT-4



Zephyr: Direct Distillation of LM Alignment

distilled supervised fine-tuning (dSFT)

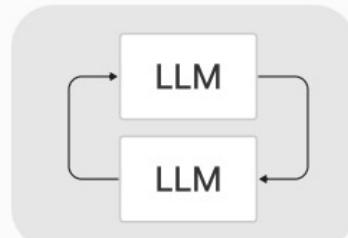
Step 1 - dSFT

Generate multi-turn AI dialogues

Prompt sampled from dataset of prompts.

Create a scenario for a game about space exploration

LLM simulates multi-turn user-assistant interactions.



Dialogues are used for supervised fine-tuning.

π_{dSFT}

AI Feedback (AIF)

Step 2 - AIF

Response generation and AI ranking

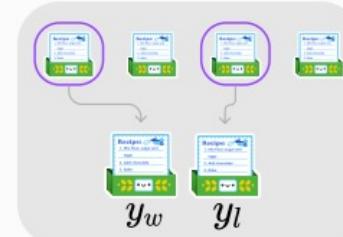
Prompt sampled from dataset of prompts.

Describe how to make chocolate brownies

4 different language models generate responses.



GPT-4 ranks the responses.



distilled Direct Preference Optimization (dDPO)

Step 3 - dDPO

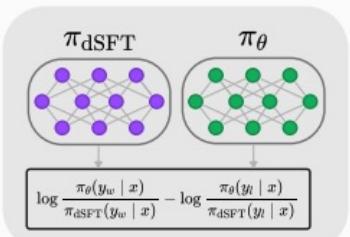
Distillation of AI preferences

Prompt sampled from dataset of prompts.

Describe how to make chocolate brownies



Best and another random response are selected.

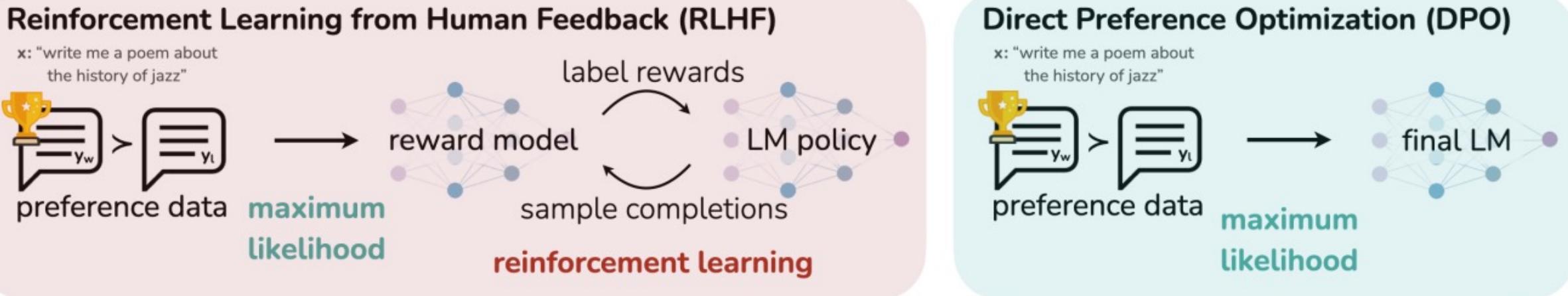


Zephyr: Direct Distillation of LM Alignment

The three steps of **Zephyr**:

- (1) large scale, self-instruct-style dataset construction
(UltraChat), followed by **distilled supervised fine-tuning (dSFT)**,
- (2) **AI Feedback (AIF)** collection
via an ensemble of chat model completions, followed by scoring by
GPT-4 (UltraFeedback) and
binarization into preferences, and
- (3) **distilled direct preference optimization (dDPO)** of the dSFT
model utilizing the feedback data.

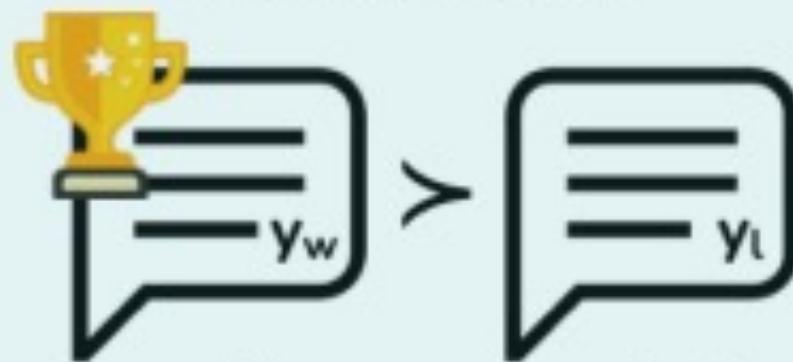
DPO optimizes for human preferences while avoiding reinforcement learning



Direct Preference Optimization (DPO)

Direct Preference Optimization (DPO)

x: "write me a poem about
the history of jazz"

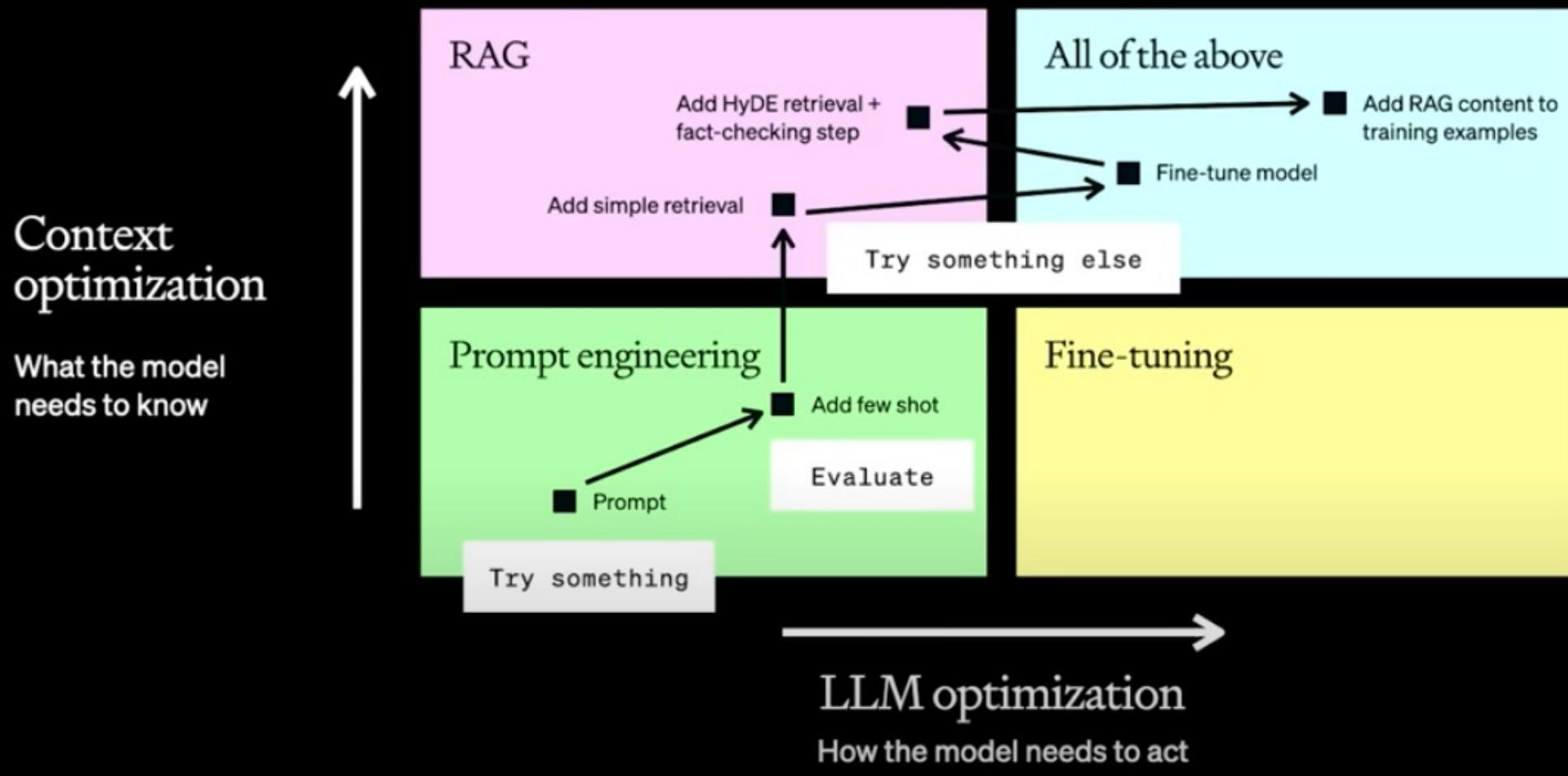


maximum
likelihood

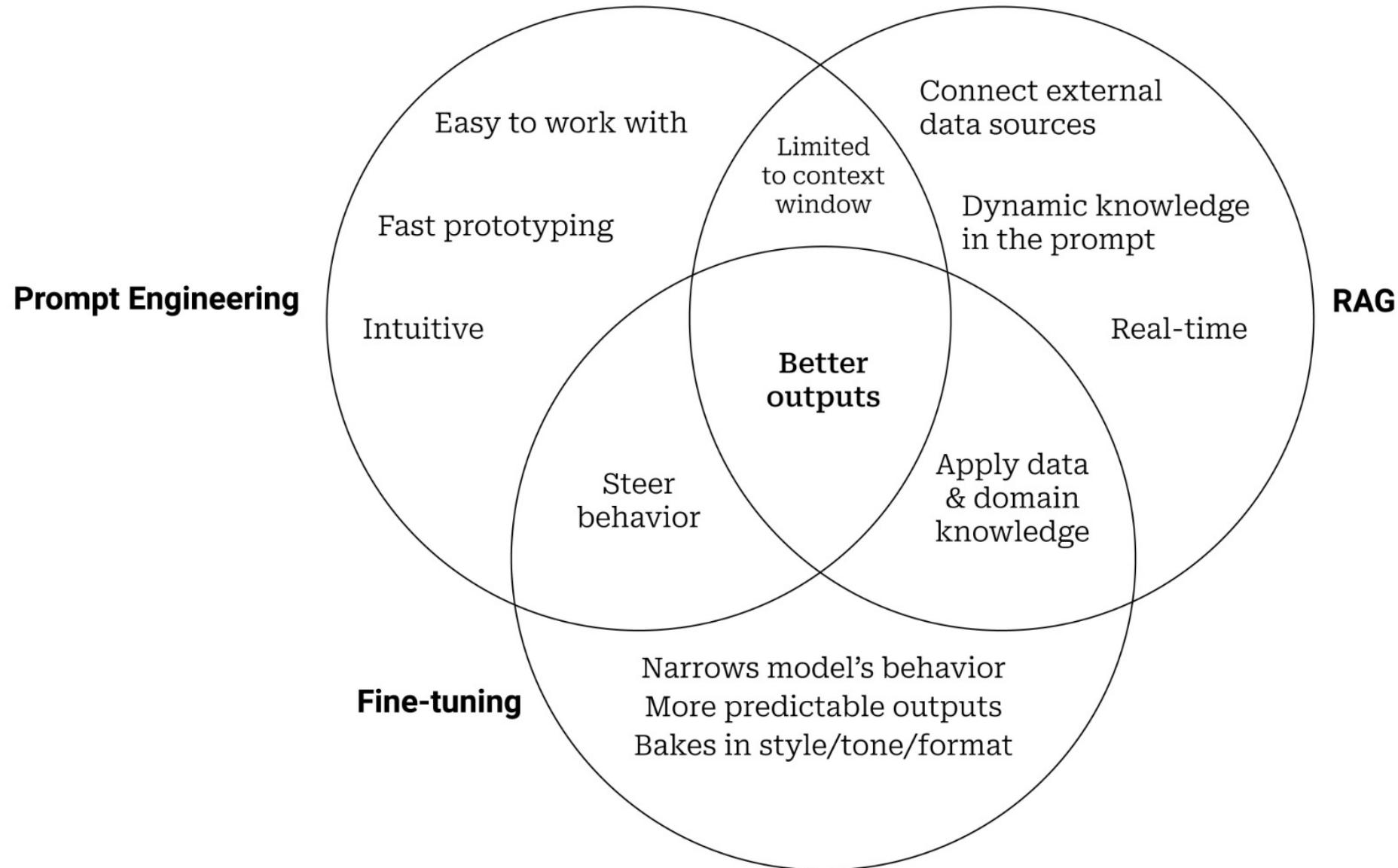


Maximizing LLM Performance

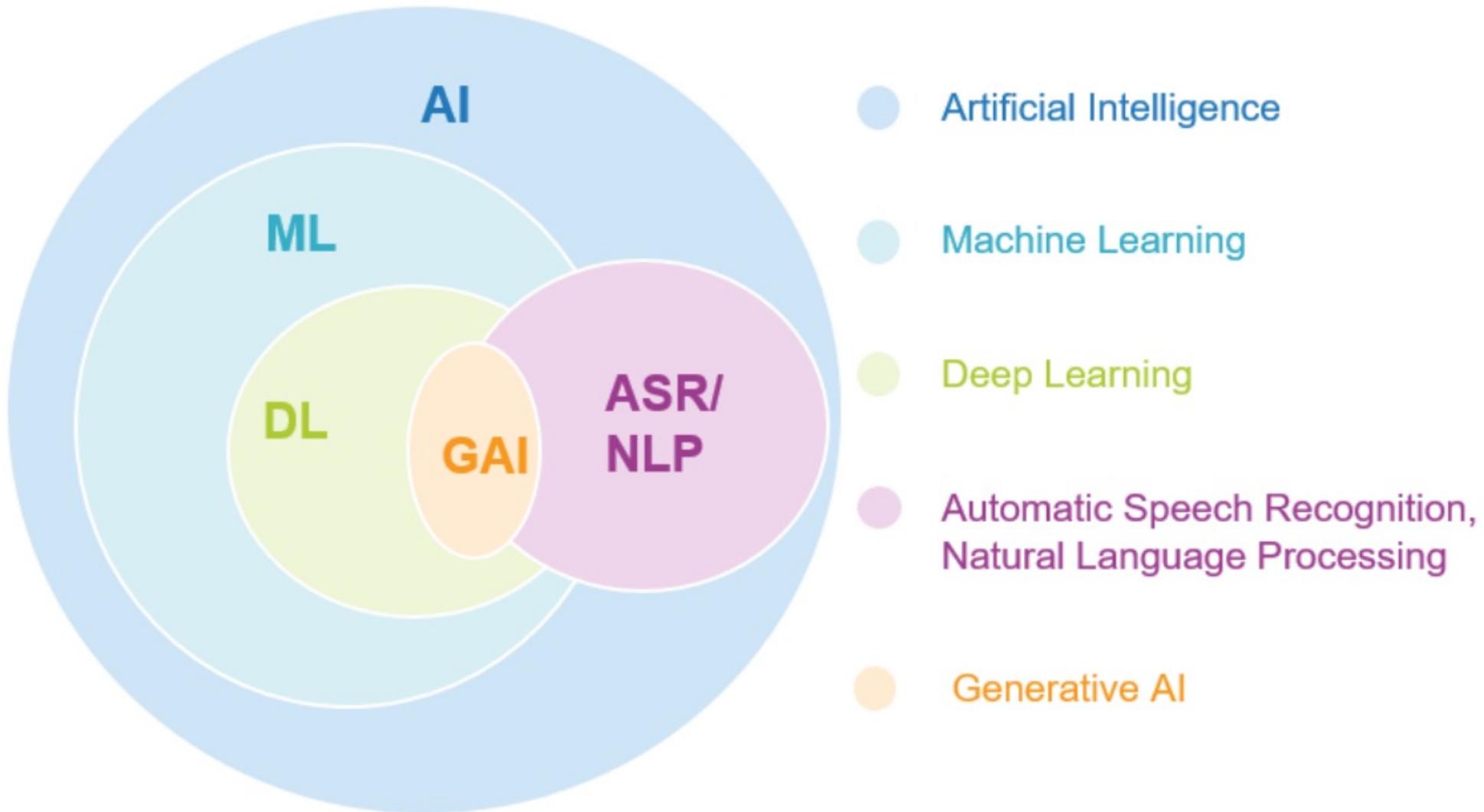
The optimization flow



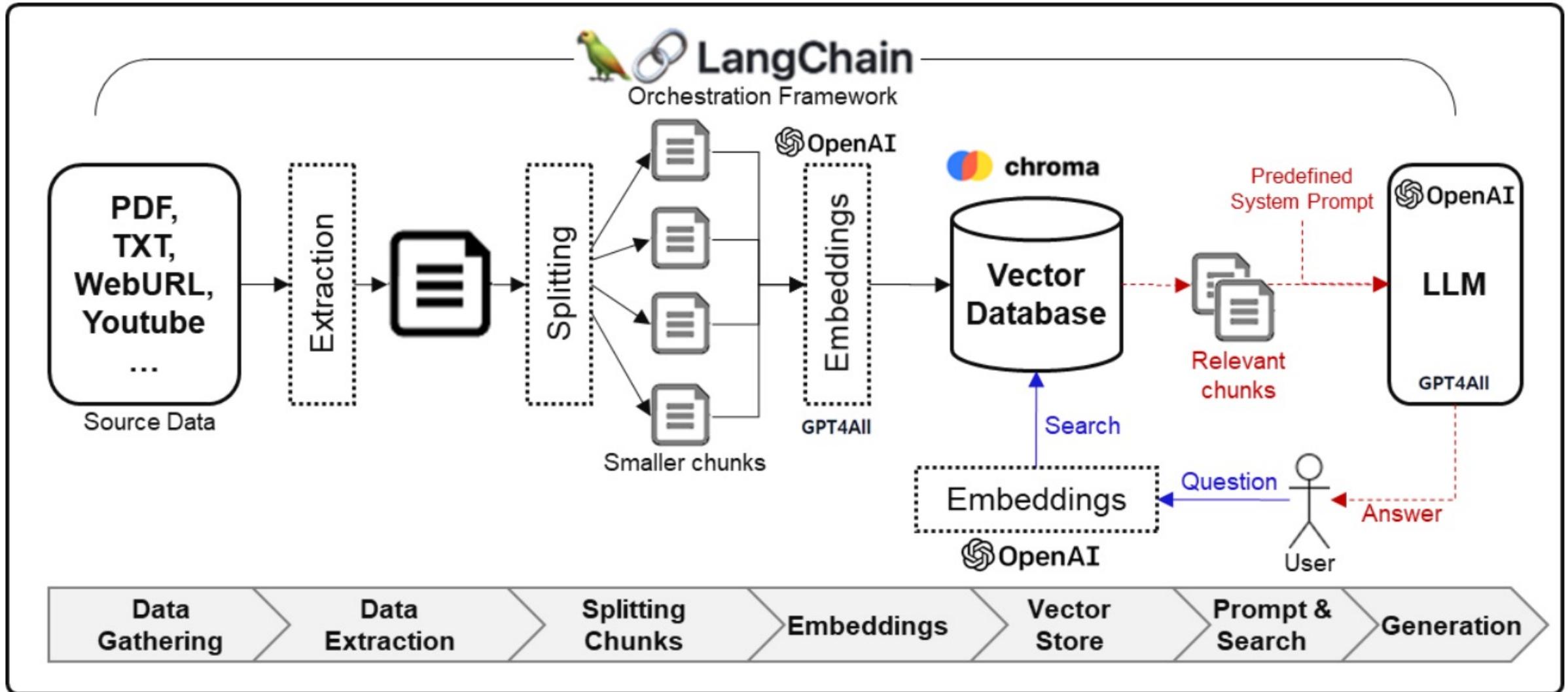
Prompt Engineering, Fine-tuning, and RAG



Generative AI

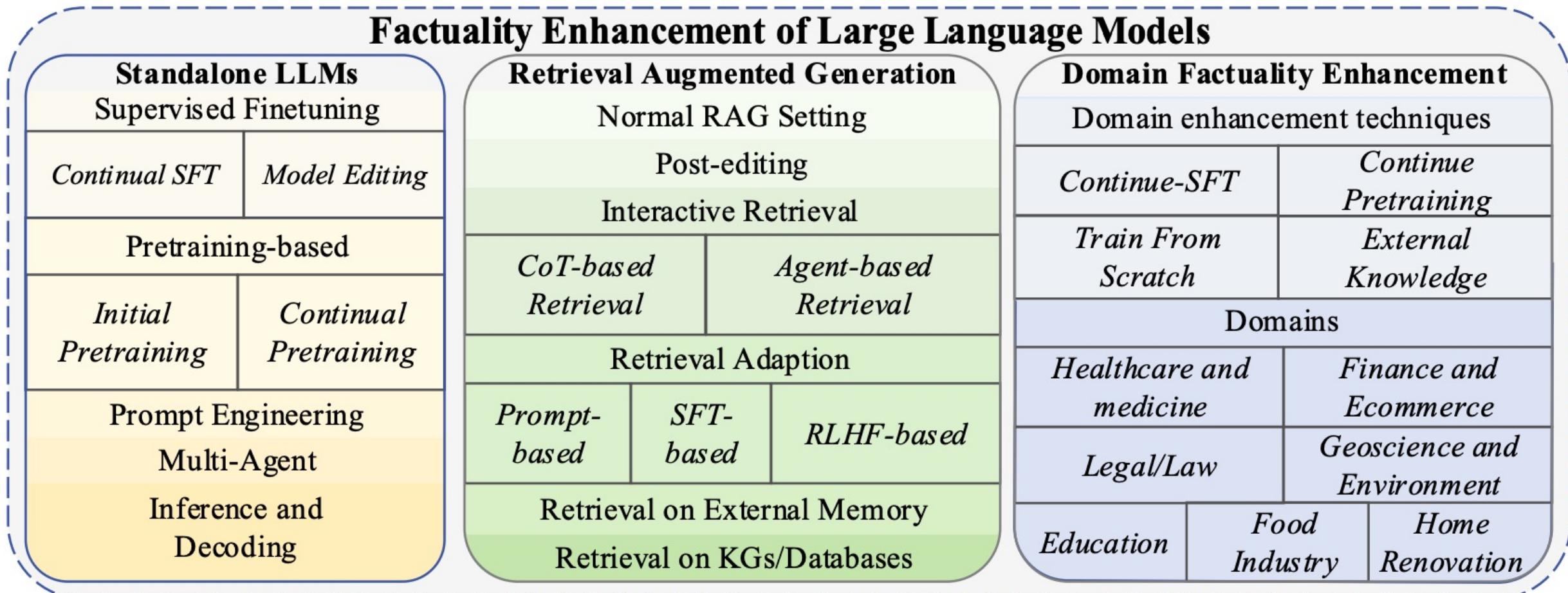


Framework for Implementing Generative AI Services using RAG Model



Factuality Enhancement of Large Language Models (LLMs)

Factuality Enhancement of Large Language Models



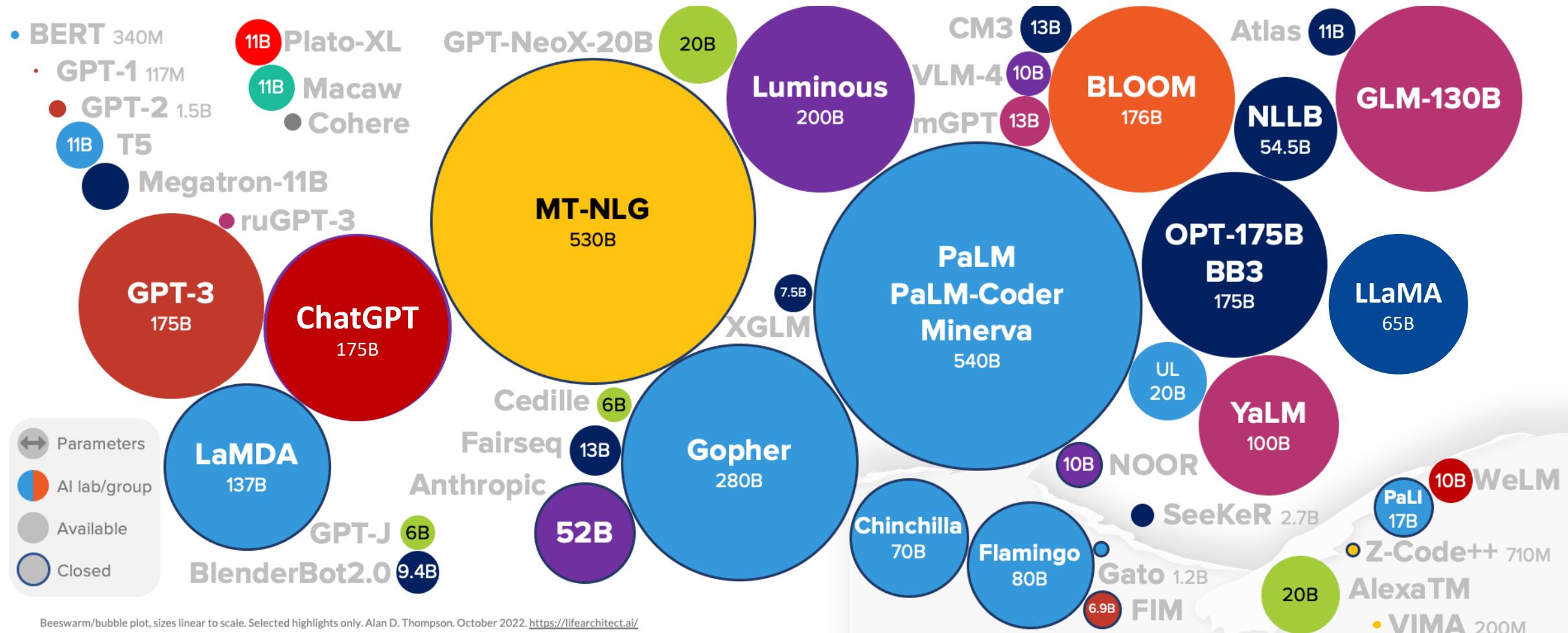
ChatGPT

Large Language Models (LLMs)

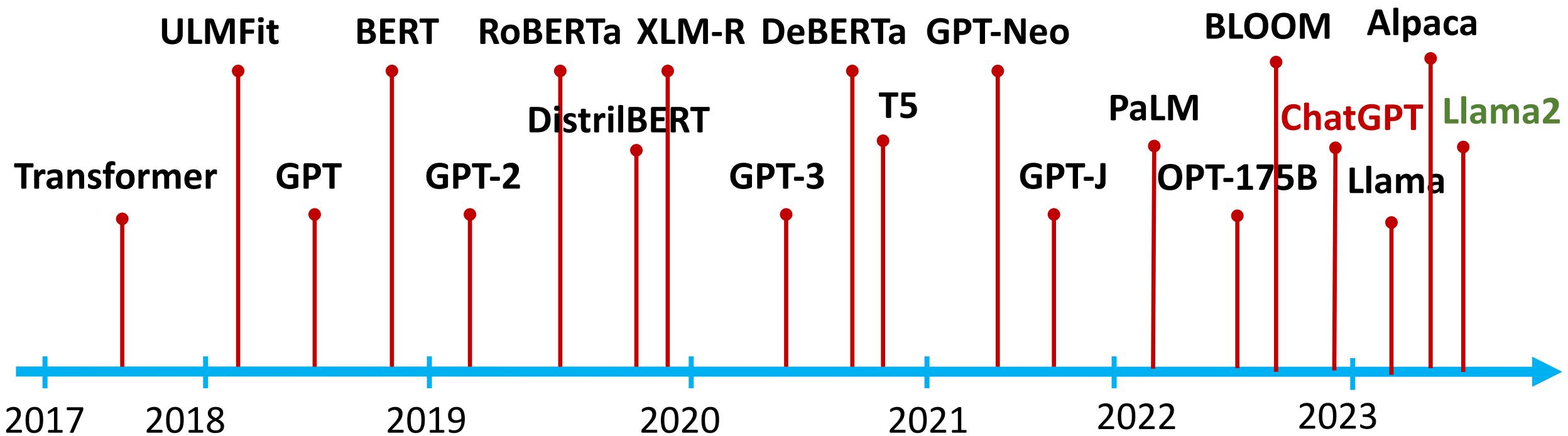
Foundation Models

Large Language Models (LLM)

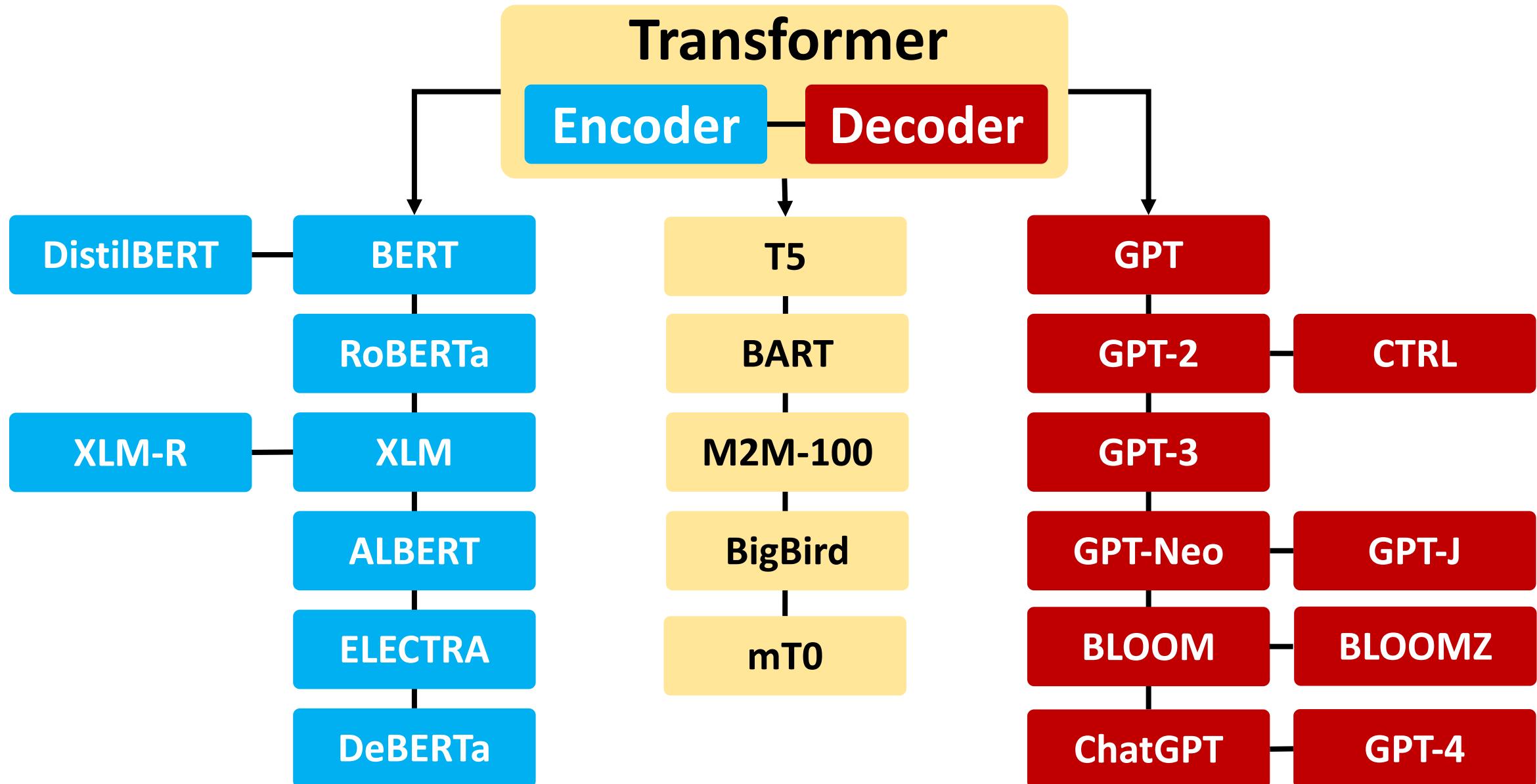
(GPT-3, ChatGPT, PaLM, BLOOM, OPT-175B, LLaMA)



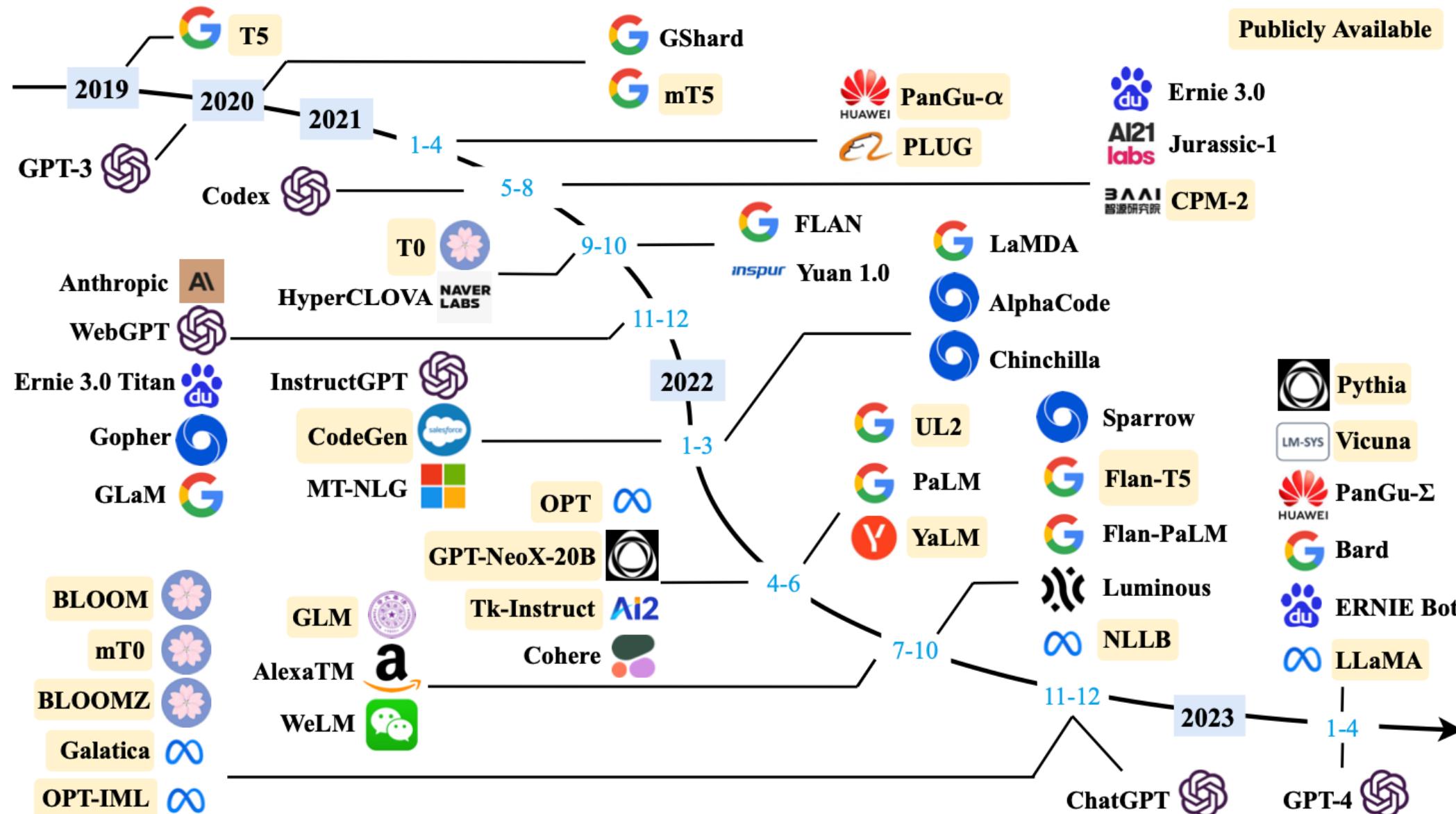
The Transformers Timeline



Transformer Models



Large Language Models (LLMs) (larger than 10B)



Large Language Models (LLMs) (larger than 10B)

Model	Release Time	Size (B)	Base Model	Adaptation IT	Adaptation RLHF	Pre-train Data Scale	Latest Data Timestamp	Hardware (GPUs / TPUs)	Training Time	Evaluation ICL	Evaluation CoT
Publicly Available	T5 [72]	Oct-2019	11	-	-	1T tokens	Apr-2019	1024 TPU v3	-	✓	-
	mT5 [73]	Oct-2020	13	-	-	1T tokens	-	-	-	✓	-
	PanGu- α [74]	Apr-2021	13*	-	-	1.1TB	-	2048 Ascend 910	-	✓	-
	CPM-2 [75]	Jun-2021	198	-	-	2.6TB	-	-	-	-	-
	T0 [28]	Oct-2021	11	T5	✓	-	-	512 TPU v3	27 h	✓	-
	CodeGen [76]	Mar-2022	16	-	-	577B tokens	-	-	-	✓	-
	GPT-NeoX-20B [77]	Apr-2022	20	-	-	825GB	-	96 40G A100	-	✓	-
	Tk-Instruct [78]	Apr-2022	11	T5	✓	-	-	256 TPU v3	4 h	✓	-
	UL2 [79]	May-2022	20	-	-	1T tokens	Apr-2019	512 TPU v4	-	✓	✓
	OPT [80]	May-2022	175	-	-	180B tokens	-	992 80G A100	-	✓	-
	NLLB [81]	Jul-2022	54.5	-	-	-	-	-	-	✓	-
	GLM [82]	Oct-2022	130	-	-	400B tokens	-	768 40G A100	60 d	✓	-
	Flan-T5 [83]	Oct-2022	11	T5	✓	-	-	-	-	✓	✓
	BLOOM [68]	Nov-2022	176	-	-	366B tokens	-	384 80G A100	105 d	✓	-
	mT0 [84]	Nov-2022	13	mT5	✓	-	-	-	-	✓	-
	Galactica [35]	Nov-2022	120	-	-	106B tokens	-	-	-	✓	✓
	BLOOMZ [84]	Nov-2022	176	BLOOM	✓	-	-	-	-	✓	-
	OPT-IML [85]	Dec-2022	175	OPT	✓	-	-	128 40G A100	-	✓	✓
	LLaMA [57]	Feb-2023	65	-	-	1.4T tokens	-	2048 80G A100	21 d	✓	-
	Pythia [86]	Apr-2023	12	-	-	300B tokens	-	256 40G A100	-	✓	-

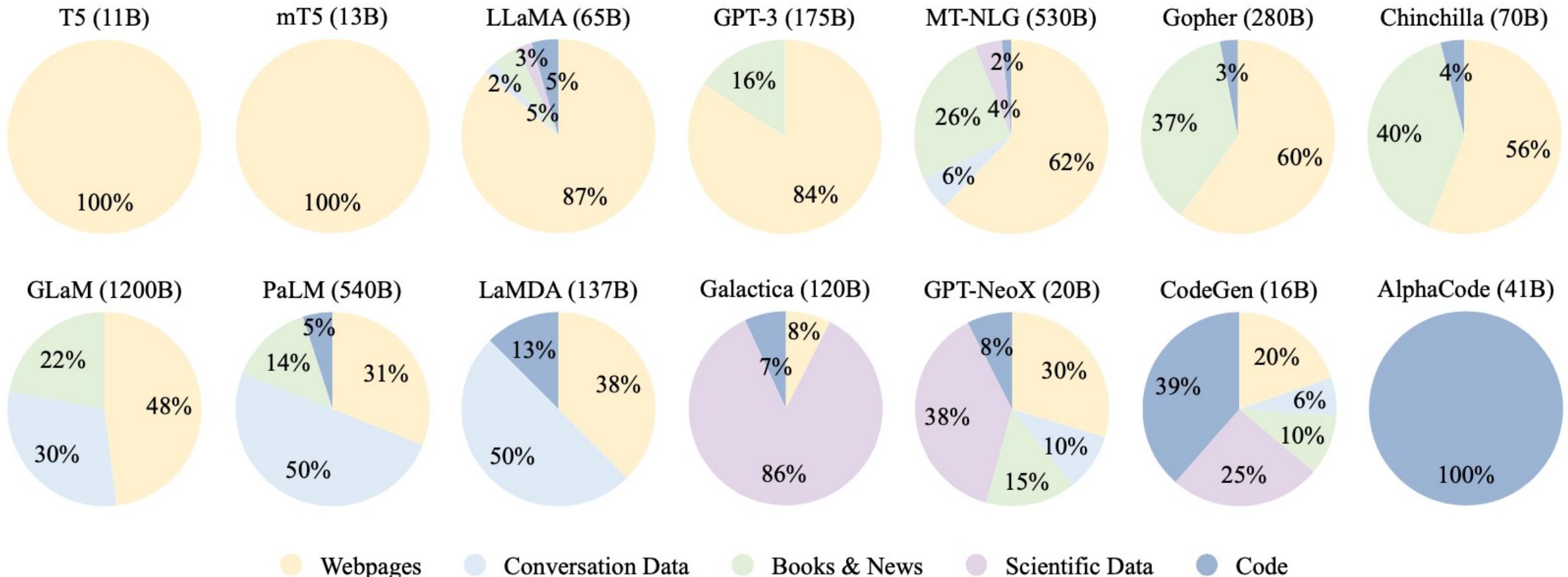
Large Language Models (LLMs) (larger than 10B)

Model	Release Time	Size (B)	Base Model	Adaptation IT	Adaptation RLHF	Pre-train Data Scale	Latest Data Timestamp	Hardware (GPUs / TPUs)	Training Time	Evaluation ICL	Evaluation CoT
GPT-3 [55]	May-2020	175	-	-	-	300B tokens	-	-	-	✓	-
GShard [87]	Jun-2020	600	-	-	-	1T tokens	-	2048 TPU v3	4 d	-	-
Codex [88]	Jul-2021	12	GPT-3	-	-	100B tokens	May-2020	-	-	✓	-
ERNIE 3.0 [89]	Jul-2021	10	-	-	-	375B tokens	-	384 V100	-	✓	-
Jurassic-1 [90]	Aug-2021	178	-	-	-	300B tokens	-	800 GPU	-	✓	-
HyperCLOVA [91]	Sep-2021	82	-	-	-	300B tokens	-	1024 A100	13.4 d	✓	-
FLAN [62]	Sep-2021	137	LaMDA	✓	-	-	-	128 TPU v3	60 h	✓	-
Yuan 1.0 [92]	Oct-2021	245	-	-	-	180B tokens	-	2128 GPU	-	✓	-
Anthropic [93]	Dec-2021	52	-	-	-	400B tokens	-	-	-	✓	-
WebGPT [71]	Dec-2021	175	GPT-3	-	✓	-	-	-	-	✓	-
Gopher [59]	Dec-2021	280	-	-	-	300B tokens	-	4096 TPU v3	920 h	✓	-
ERNIE 3.0 Titan [94]	Dec-2021	260	-	-	-	300B tokens	-	2048 V100	28 d	✓	-
GLaM [95]	Dec-2021	1200	-	-	-	280B tokens	-	1024 TPU v4	574 h	✓	-
LaMDA [96]	Jan-2022	137	-	-	-	2.81T tokens	-	1024 TPU v3	57.7 d	-	-
MT-NLG [97]	Jan-2022	530	-	-	-	270B tokens	-	4480 80G A100	-	✓	-
AlphaCode [98]	Feb-2022	41	-	-	-	967B tokens	Jul-2021	-	-	-	-
InstructGPT [61]	Mar-2022	175	GPT-3	✓	✓	-	-	-	-	✓	-
Chinchilla [34]	Mar-2022	70	-	-	-	1.4T tokens	-	-	-	✓	-
PaLM [56]	Apr-2022	540	-	-	-	780B tokens	-	6144 TPU v4	-	✓	✓
AlexaTM [99]	Aug-2022	20	-	-	-	1.3T tokens	-	128 A100	120 d	✓	✓
Sparrow [100]	Sep-2022	70	-	-	✓	-	-	64 TPU v3	-	✓	-
WeLM [101]	Sep-2022	10	-	-	-	300B tokens	-	128 A100 40G	24 d	✓	-
U-PaLM [102]	Oct-2022	540	PaLM	-	-	-	-	512 TPU v4	5 d	✓	✓
Flan-PaLM [83]	Oct-2022	540	PaLM	✓	-	-	-	512 TPU v4	37 h	✓	✓
Flan-U-PaLM [83]	Oct-2022	540	U-PaLM	✓	-	-	-	-	-	✓	✓
GPT-4 [46]	Mar-2023	-	-	✓	✓	-	-	-	-	✓	✓
PanGu- Σ [103]	Mar-2023	1085	PanGu- α	-	-	329B tokens	-	512 Ascend 910	100 d	✓	-

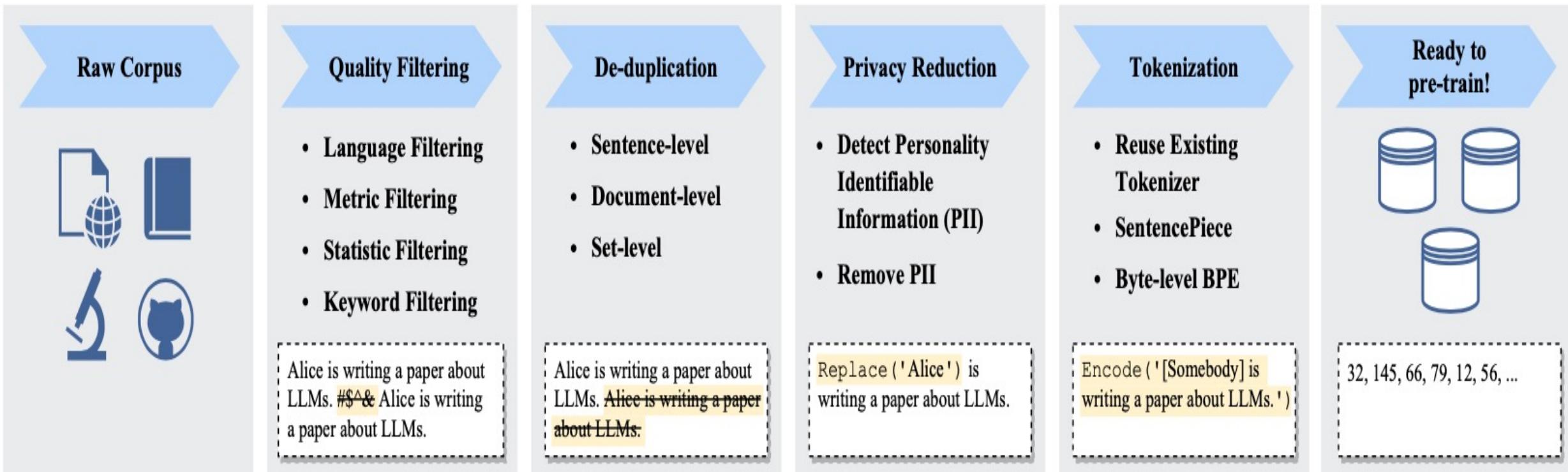
Statistics of Commonly-used Data Sources for LLMs

Corpora	Size	Source	Latest Update Time
BookCorpus [109]	5GB	Books	Dec-2015
Gutenberg [110]	-	Books	Dec-2021
C4 [72]	800GB	CommonCrawl	Apr-2019
CC-Stories-R [111]	31GB	CommonCrawl	Sep-2019
CC-NEWS [27]	78GB	CommonCrawl	Feb-2019
REALNEWS [112]	120GB	CommonCrawl	Apr-2019
OpenWebText [113]	38GB	Reddit links	Mar-2023
Pushift.io [114]	-	Reddit links	Mar-2023
Wikipedia [115]	-	Wikipedia	Mar-2023
BigQuery [116]	-	Codes	Mar-2023
the Pile [117]	800GB	Other	Dec-2020
ROOTS [118]	1.6TB	Other	Jun-2022

Ratios of various data sources in the pre-training data for existing LLMs



Typical Data Preprocessing Pipeline for Pre-training Large Language Models (LLMs)



LLMs with Public Configuration Details

Model	Category	Size	Normalization	PE	Activation	Bias	#L	#H	d_{model}	MCL
GPT3 [55]	Causal decoder	175B	Pre Layer Norm	Learned	GeLU	✓	96	96	12288	2048
PanGU- α [74]	Causal decoder	207B	Pre Layer Norm	Learned	GeLU	✓	64	128	16384	1024
OPT [80]	Causal decoder	175B	Pre Layer Norm	Learned	ReLU	✓	96	96	12288	2048
PaLM [56]	Causal decoder	540B	Pre Layer Norm	RoPE	SwiGLU	✗	118	48	18432	2048
BLOOM [68]	Causal decoder	176B	Pre Layer Norm	ALiBi	GeLU	✓	70	112	14336	2048
MT-NLG [97]	Causal decoder	530B	-	-	-	-	105	128	20480	2048
Gopher [59]	Causal decoder	280B	Pre RMS Norm	Relative	-	-	80	128	16384	2048
Chinchilla [34]	Causal decoder	70B	Pre RMS Norm	Relative	-	-	80	64	8192	-
Galactica [35]	Causal decoder	120B	Pre Layer Norm	Learned	GeLU	✗	96	80	10240	2048
LaMDA [96]	Causal decoder	137B	-	Relative	GeGLU	-	64	128	8192	-
Jurassic-1 [90]	Causal decoder	178B	Pre Layer Norm	Learned	GeLU	✓	76	96	13824	2048
LLaMA [57]	Causal decoder	65B	Pre RMS Norm	RoPE	SwiGLU	✓	80	64	8192	2048
GLM-130B [82]	Prefix decoder	130B	Post Deep Norm	RoPE	GeGLU	✓	70	96	12288	2048
T5 [72]	Encoder-decoder	11B	Pre RMS Norm	Relative	ReLU	✗	24	128	1024	512

Note: PE denotes position embedding, #L denotes the number of layers, #H denotes the number of attention heads, d_{model} denotes the size of hidden states, and MCL denotes the maximum context length during training.

Detailed Optimization Settings of LLMs

Model	Batch Size (#tokens)	Learning Rate	Warmup	Decay Method	Optimizer	Precision Type	Weight Decay	Grad Clip	Dropout
GPT3 (175B)	32K→3.2M	6×10^{-5}	yes	cosine decay to 10%	Adam	FP16	0.1	1.0	-
PanGu- α (200B)	-	2×10^{-5}	-	-	Adam	-	0.1	-	-
OPT (175B)	2M	1.2×10^{-4}	yes	manual decay	AdamW	FP16	0.1	-	0.1
PaLM (540B)	1M→4M	1×10^{-2}	no	inverse square root	Adafactor	BF16	lr^2	1.0	0.1
BLOOM (176B)	4M	6×10^{-5}	yes	cosine decay to 10%	Adam	BF16	0.1	1.0	0.0
MT-NLG (530B)	64 K→3.75M	5×10^{-5}	yes	cosine decay to 10%	Adam	BF16	0.1	1.0	-
Gopher (280B)	3M→6M	4×10^{-5}	yes	cosine decay to 10%	Adam	BF16	-	1.0	-
Chinchilla (70B)	1.5M→3M	1×10^{-4}	yes	cosine decay to 10%	AdamW	BF16	-	-	-
Galactica (120B)	2M	7×10^{-6}	yes	linear decay to 10%	AdamW	-	0.1	1.0	0.1
LaMDA (137B)	256K	-	-	-	-	BF16	-	-	-
Jurassic-1 (178B)	32 K→3.2M	6×10^{-5}	yes	-	-	-	-	-	-
LLaMA (65B)	4M	1.5×10^{-4}	yes	cosine decay to 10%	AdamW	-	0.1	1.0	-
GLM (130B)	0.4M→8.25M	8×10^{-5}	yes	cosine decay to 10%	AdamW	FP16	0.1	1.0	0.1
T5 (11B)	64K	1×10^{-2}	no	inverse square root	AdaFactor	-	-	-	0.1
ERNIE 3.0 Titan (260B)	-	1×10^{-4}	-	-	Adam	FP16	0.1	1.0	-
PanGu- Σ (1.085T)	0.5M	2×10^{-5}	yes	-	Adam	FP16	-	-	-

Generative AI

Text, Image, Video, Audio

Applications

Popular Generative AI

- OpenAI ChatGPT (GPT-3.5, GPT-4)
- OpenAI DALL·E 3
- Perplexity.ai
- Chat.LMSys.org
 - Llama 2, Mistral, Zephyr, Vicuna, MPT, Falcon
- ChatPDF
- Stable Diffusion
- Video: D-ID, Synthesia
- Audio: Speechify

OpenAI ChatGPT (GPT-4, GPT-3.5)

ChatGPT 4

- ❖ GPT-4
With DALL-E, browsing and analysis
Limit 40 messages / 3 hours
- ⚡ GPT-3.5
Great for everyday tasks
- 🔌 Plugins



I can I help you today?

Suggest fun activities
for a family of 4 to do indoors on a rainy day

Help me pick
a birthday gift for my mom who likes gardening

Design a database schema
for an online merch store

Give me ideas
for what to do with my kids' art

Message ChatGPT... 

ChatGPT can make mistakes. Consider checking important information.

<https://chat.openai.com/>

OpenAI ChatGPT (GPT-4) DALL·E 3

Vector graphic of a flowchart depicting the integration of generative AI in the education process, from content creation to virtual experiments, personalized learning, and innovative learning.



Perplexity.ai

perplexity

Anonymous Like ... ⚡ Share

Large Language Models for Data Science

Sources

What Is a Large Language Model and What Can... snowflake · 1	What are Large Language Models? NVIDIA nvidia · 2	Catch Up On Large Language Models by... towardsdatascien... · 3	   View 3 more
---	--	--	--

Answer

Large Language Models (LLMs) are advanced AI systems designed to understand human language intricacies and generate intelligent, creative responses to queries ¹. They are deep learning algorithms that can recognize, summarize, translate, predict, and generate content using very large datasets ². LLMs are trained on enormous data sets typically measured in petabytes, sourced from books, articles, websites, and other text-based sources ¹. They are dynamic tools that are continuously updated and refined as new data becomes available ¹.

In the field of data science, LLMs can potentially transform how teams source, manage, and analyze data ¹. They are used in numerous data science applications, and their ability to process and interpret vast amounts of text data has made them an indispensable part of modern data science workflows.

Ask follow-up... Copilot ?

social media posts or customer reviews, to determine whether the overall sentiment is

<https://www.perplexity.ai/>

Chat with Open Large Language Models:

<https://chat.lmsys.org/>

- [Llama 2](#): open foundation and fine-tuned chat models by Meta
- [Mistral](#): a large language model by Mistral AI team
- [WizardLM](#): an instruction-following LLM using evol-instruct by Microsoft
- [Zephyr](#): a chatbot fine-tuned from Mistral by Hugging Face
- [Code Llama](#): open foundation models for code by Meta
- [ChatGLM](#): an open bilingual dialogue language model by Tsinghua University
- [Qwen](#): a large language model by Alibaba Cloud
- [Vicuna](#): a chat assistant fine-tuned on user-shared conversations by LMSYS
- [Falcon](#): TII's flagship series of large language models

Chat with Open Large Language Models

chat.lmsys.org

✖ Chatbot Arena ✖ : Benchmarking LLMs in the Wild

[GPT-3.5](#): GPT-3.5 by OpenAI

[GPT-4](#): ChatGPT-4 by OpenAI

[Vicuna](#): a chat assistant fine-tuned on user-shared conversations by LMSYS

[OpenChat 3.5](#): OpenChat 3.5 is a versatile, open-source language model fine-tuned using C-RLFT

[Qwen](#): a large language model by Alibaba Cloud

[PaLM 2 Chat](#): PaLM 2 for Chat (chat-bison@001) by Google

[GPT-3.5-Turbo-1106](#): GPT-3.5-Turbo-1106 by OpenAI

[Claude](#): Claude 2 by Anthropic

[Llama 2](#): open foundation and fine-tuned chat models by Meta

[Mistral](#): a large language model by Mistral AI team

[Code Llama](#): open foundation models for code by Meta

[Falcon](#): TII's flagship series of large language models

[GPT-4-Turbo](#): GPT-4-Turbo by OpenAI

[Claude Instant](#): Claude Instant by Anthropic

[ChatGLM](#): an open bilingual dialogue language model by Tsinghua University

[Zephyr](#): a chatbot fine-tuned from Mistral by Hugging Face

[WizardLM](#): an instruction-following LLM using evol-instruct by Microsoft

Chat with Open Large Language Models: Chatbot Arena

Large Language Models for Data Science

Llama 2-70b-chat

llama-2-70b-chat

Model A

Large Language Models for Data Science

Large language models have gained significant attention in recent years for their ability to process and analyze natural language data. These models have been successfully applied in various fields such as natural language processing, text classification, sentiment analysis, and machine translation. In data science, large language models can be used for a variety of tasks, including data cleaning, data transformation, and data analysis.

In this article, we will explore the use of large language models for data science tasks and discuss their advantages and limitations. We will also provide examples of large language models that can be used for data science applications.

Advantages of Large Language Models for Data

Mistral-7b-instruct

mistral-7b-instruct

Model B

Large Language Models for Data Science

Large Language Models (LLMs) are becoming increasingly important in data science due to their ability to understand and generate natural language text. Here are some of the key ways that LLMs can be used in data science:

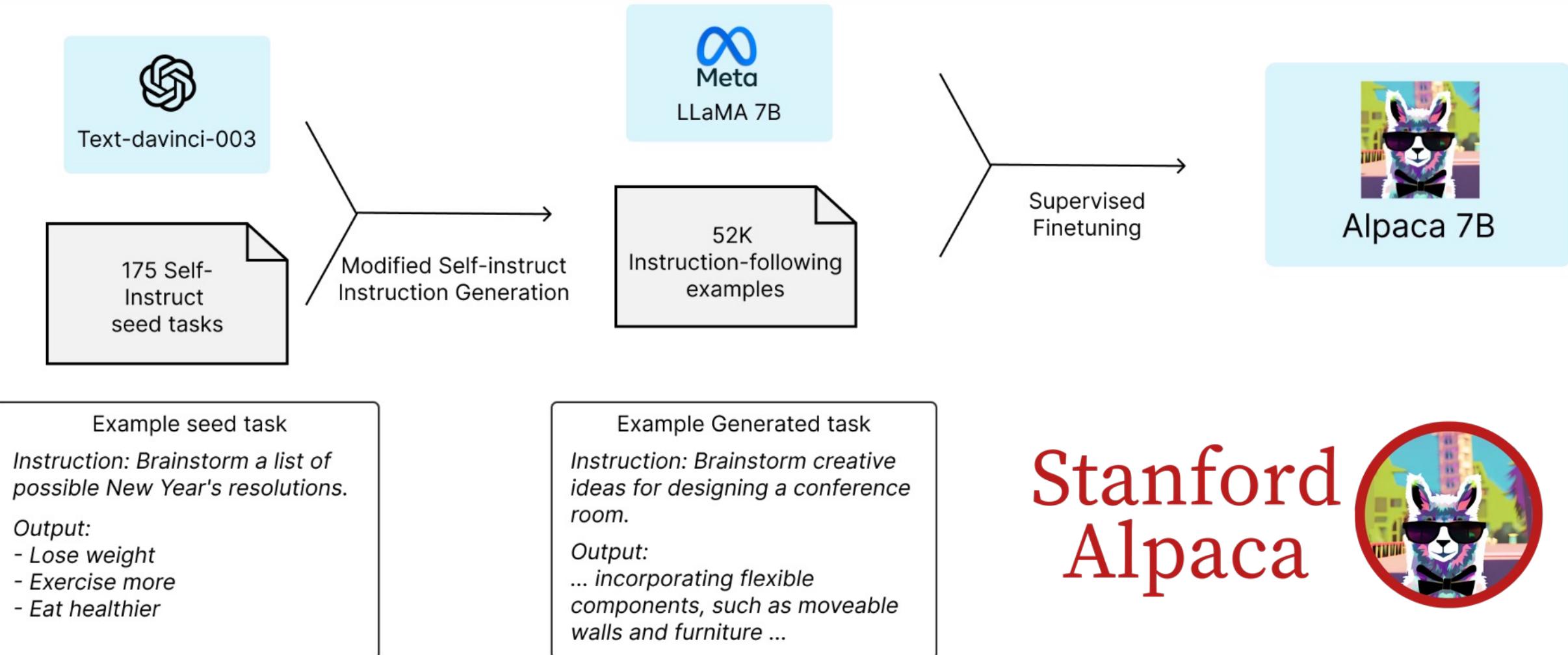
1. Text classification: LLMs can be trained on large datasets of labeled text and then used to classify new text into different categories. This can be useful in a variety of applications, such as sentiment analysis, spam filtering, and topic classification.
2. Named entity recognition: LLMs can also be used to identify and extract specific pieces of information from text, such as names of people, organizations, and locations. This can be useful in applications such as information extraction

Chatbot Arena Leaderboard

LLM Leaderboard

Model	⭐ Arena Elo rating	↗ MT-bench (score)	MMLU	License
GPT-4-Turbo	1210	9.32		Proprietary
GPT-4	1159	8.99	86.4	Proprietary
Claude-1	1146	7.9	77	Proprietary
Claude-2	1125	8.06	78.5	Proprietary
Claude-instant-1	1106	7.85	73.4	Proprietary
GPT-3.5-turbo	1103	7.94	70	Proprietary
WizardLM-70b-v1.0	1093	7.71	63.7	Llama 2 Community
Vicuna-33B	1090	7.12	59.2	Non-commercial
OpenChat-3.5	1070	7.81	64.3	Apache-2.0
Llama-2-70b-chat	1065	6.86	63	Llama 2 Community
WizardLM-13b-v1.2	1047	7.2	52.7	Llama 2 Community
zephyr-7b-beta	1042	7.34	61.4	MIT
MPT-30B-chat	1031	6.39	50.4	CC-BY-NC-SA-4.0
Vicuna-13B	1031	6.57	55.8	Llama 2 Community
QWen-Chat-14B	1030	6.96	66.5	Qianwen LICENSE

Stanford Alpaca: A Strong, Replicable Instruction-Following Model

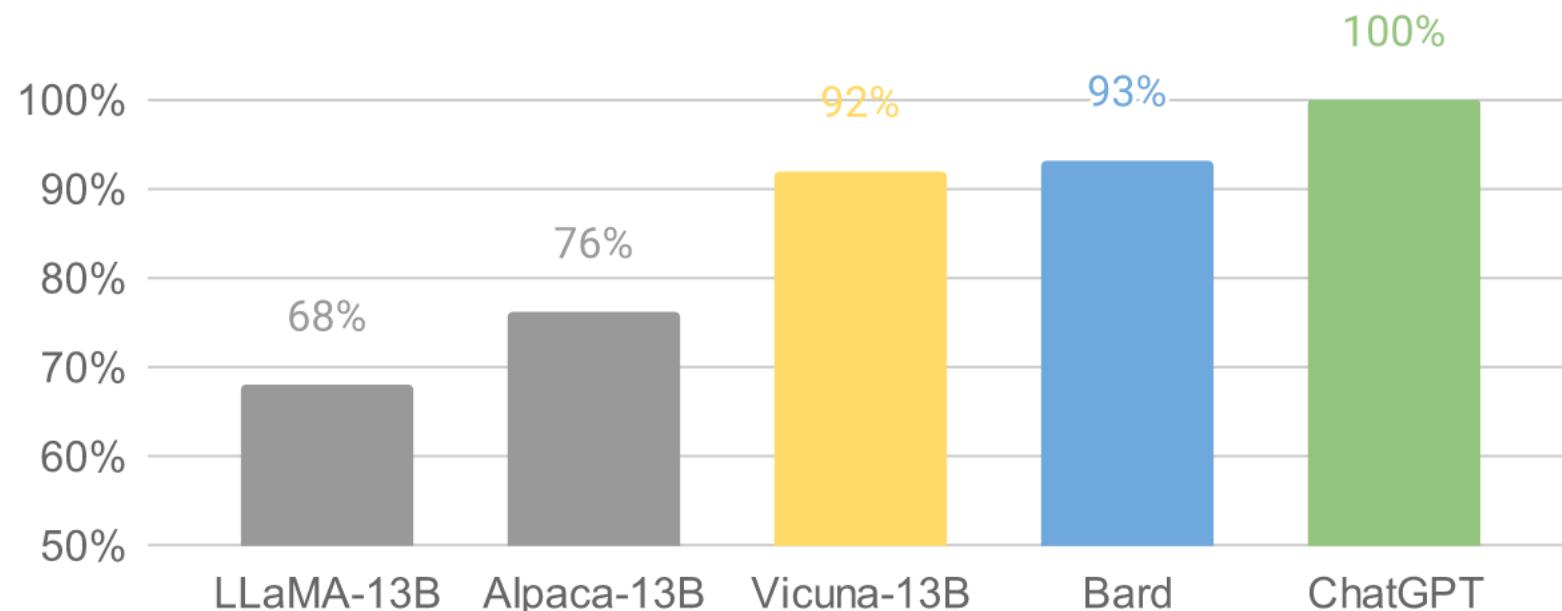


Vicuna: An Open-Source Chatbot

Impressing GPT-4 with 90%* ChatGPT Quality

by the Team with members from UC Berkeley, CMU, Stanford, and UC San Diego

- **Vicuna-13B: an open-source chatbot trained by fine-tuning LLaMA on user-shared conversations collected from ShareGPT.**
- **The cost of training Vicuna-13B is around \$300.**



Chinese-Vicuna: A Chinese Instruction-following LLaMA-based Model 一個中文低資源的 llama+lora 方案

Dataset Preview

instruction (string)	input (string)	output (string)
"用一句话描述地球为什么是独一无二的。 \n "	""	"地球上适宜生命存在的条件和多样化的生命形式。"
"给出一段对话，要求GPT模型使用合适的语气和回答方式继续对话。 对话： A：你今... "基于以下提示填写以下句子的空格。 提示： - 提供多种现实世界的场景 - 空格... "请生成一个新闻标题，描述一场正在发生的大型自然灾害。 \n "	""	"A: 这太好了！你的新工作听起来很令人兴奋。你对接下来的日子有什么期望吗？ B: 是啊，我非..." "多种形容词可填，以下是其中一些例子： - 愉快的 - 惬意的 - 轻松的 - 安静的 - 美妙的" " ""强烈飓风肆虐，数百万人疏散！" "人工智能是一种科技，通过使用计算机程序和算法实现类似于人类智能的能力，例如学习、推理和自..."
"为指定的词汇创建一个关于该词汇的简短解释。 "人工智能" "在给定的文本中查找并替换特定的单词。 原始文本： "这个句子中有一个错误的单词... "根据给定的年份，计算该年是否为闰年。 \n \n1996\n"	""	"替换后的文本： "这个句子中有一个修改的单词" "是闰年。"



Chinese-Vicuna based on Guanaco Dataset and Belle Dataset

Source: https://huggingface.co/datasets/Chinese-Vicuna/guanaco_belle_merge_v1.0

Source: <https://github.com/Facico/Chinese-Vicuna>

RedPajama

a project to create leading open-source models,
starts by reproducing LLaMA training dataset of over 1.2 trillion tokens



Dataset	RedPajama	LLaMA*
CommonCrawl	878 billion	852 billion
C4	175 billion	190 billion
Github	59 billion	100 billion
Books	26 billion	25 billion
ArXiv	28 billion	33 billion
Wikipedia	24 billion	25 billion
StackExchange	20 billion	27 billion
Total Tokens	1.2 trillion	1.25 trillion

ChatPDF

www.chatpdf.com



ChatPDF & Jenni AI: Write your next paper with AI - Unlimited access to the [Jenni AI Writer](#) with ChatPDF Plus! [i](#)

Chat with any PDF

Join millions of students, researchers and professionals to instantly answer questions and understand research with AI



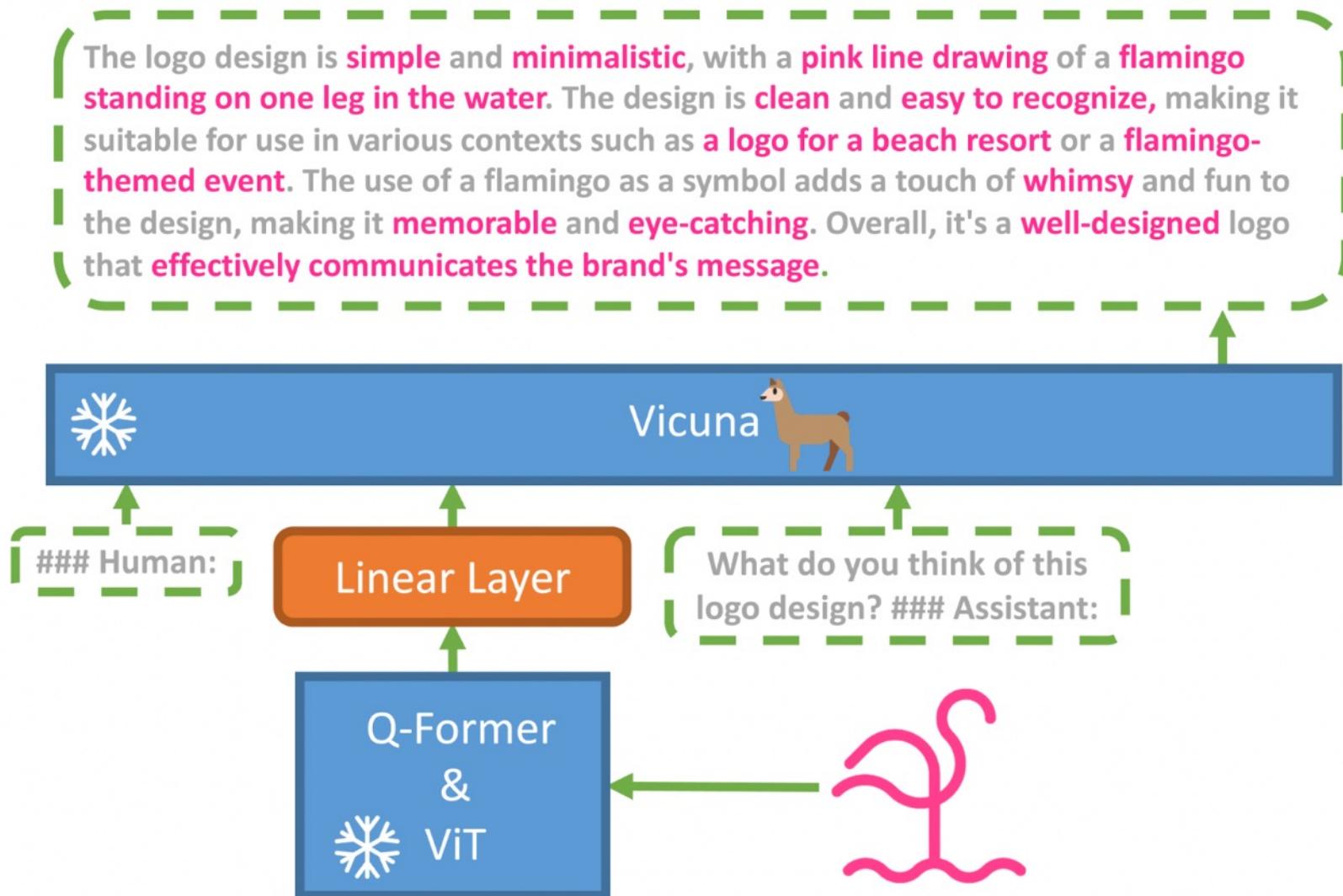
Drop PDF here

[Browse my Computer](#)

[From URL](#)

MiniGPT-4:

Enhancing Vision-language Understanding with Advanced Large Language Models



LLaVA: Large Language and Vision Assistant

llava-vl.github.io

 LLaVA: Large Language and Vision Assistant

[Project Page](#) [Paper](#) [Code](#) [Model](#)

LLaVA-13B-v0

Image

Drop Image Here
- or -
Click to Upload

Preprocess
When the image is not square, you may try adjusting this .

Crop Resize Pad

Parameters

Terms of use
By using this service, users are required to agree to the following terms:
The service is a research preview intended for non-commercial use only. It only provides limited safety measures and may generate offensive content. It must not be used for any illegal, harmful, violent, racist, or sexual purposes. The service may collect user dialogue data for future research.

LLaVA Chatbot

What are the things I should be cautious about when I visit here?



When visiting this location, one should be cautious about the weather conditions and the state of the wooden pier. In the image, the wooden pier is extended over a large, still lake under cloudy skies, which could suggest the possibility of sudden changes in weather, such as wind, rain, or even high waves. Visitors should pay attention to weather forecasts and remain vigilant while on the pier to ensure their safety. Additionally, they should be mindful of the pier's condition and the surrounding water, as wood can be slippery, especially when wet or in windy conditions. It is crucial to maintain proper footing and follow any posted safety guidelines, as the quiet waters of the lake can be deceptively

Enter text and press ENTER

Submit

Upvote

Downvote

Flag

Regenerate

Visual Instruction Tuning

LLaVA: Large Language and Vision Assistant

University of Wisconsin-Madison, Microsoft Research, Columbia University

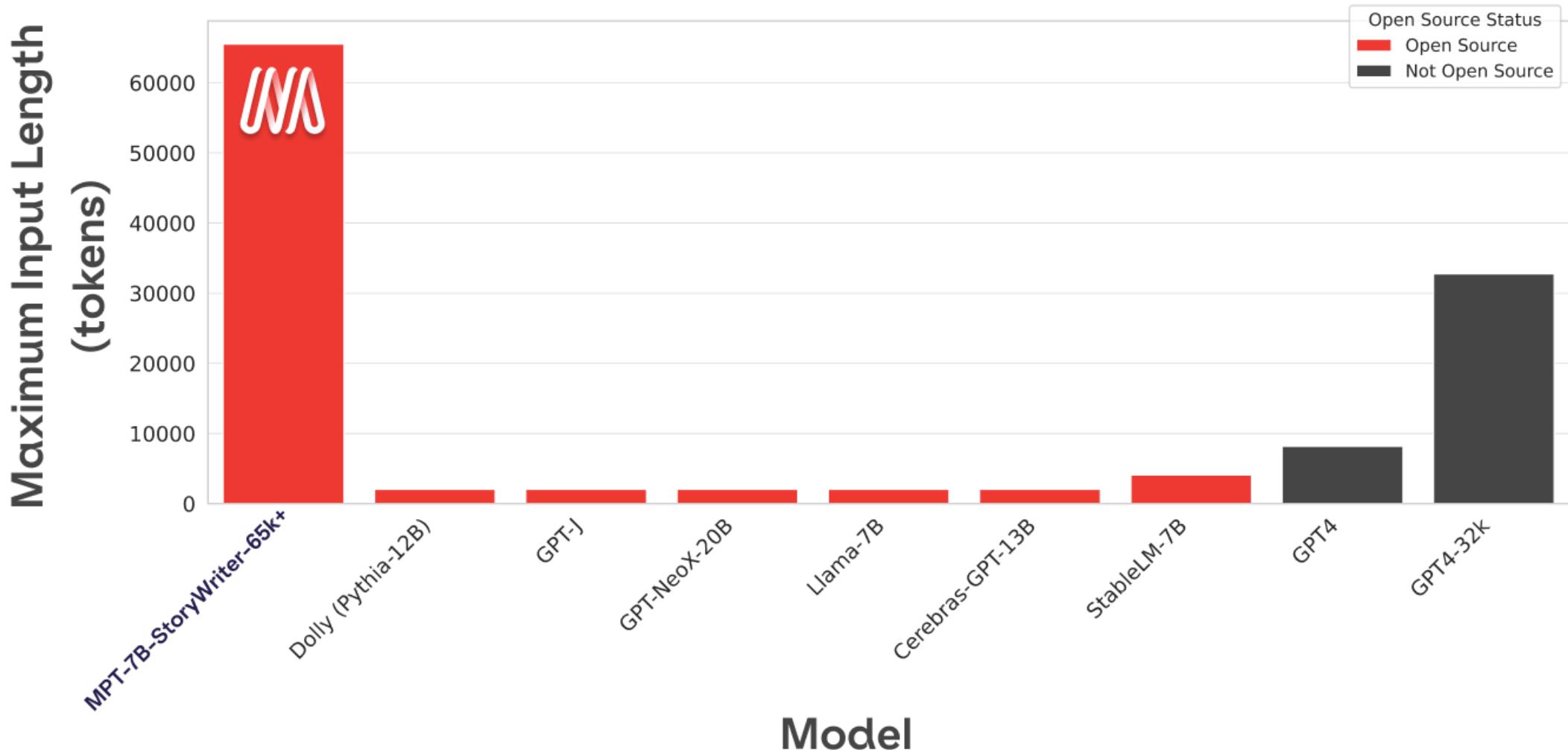
Science QA:
New SoTA with
the synergy of
LLaVA with
GPT-4



LLaVA represents a novel end-to-end trained large multimodal model that combines a vision encoder and Vicuna for general-purpose visual and language understanding, achieving impressive chat capabilities mimicking spirits of the multimodal GPT-4 and setting a new state-of-the-art accuracy on Science QA.

MPT-7B-StoryWriter-65k+

Maximum Input Lengths of Different LLMs





MPT-30B

OPEN-SOURCE LLM.
COMMERCIALY LICENSED.
MORE POWERFUL.

MPT-30B, MPT-7B LLaMa-30B, LLaMa-7B

Model Purpose	Model Series	Model	Sequence Length	Accuracy (Pass@1)	Externally Reported Pass@1 & [Source]
General Purpose	MPT	MPT-30B	1024	25.00%	N/A
		MPT-30B Chat	1024	37.20%	N/A
		MPT-30B Instruct	1024	26.20%	N/A
		MPT-7B	1024	15.90%	N/A
		MPT-7B Instruct	1024	16.50%	N/A
General Purpose	LLaMa	LLaMa-7B	1024	10.10%	10.5% [1]
		LLaMa-13B	1024	16.50%	15.8% [1]
		LLaMa-30B	1024	20.10%	21.7% [1]
General Purpose	Falcon	Falcon-40B	1024	1.2%* (did not generate code)	N/A
		Falcon-40B Instruct	1024	0.6%* (did not generate code)	18.9% [2]

Meta Llama-2 70B: Best Open Source and Commercial LLM (Llama-2, Falcon, MPT)

 MetaAI



Introducing Llama 2

The next generation of our
open source large language model

Llama 2 is available for free for research and commercial use.

[Download the Model](#)

Meta Llama-2 70B: Best Open Source and Commercial LLM (Llama-2, Falcon, MPT)

MODEL SIZE (PARAMETERS)	PRETRAINED	FINE-TUNED FOR CHAT USE CASES
7B	Model architecture:	Data collection for helpfulness and safety:
13B	Pretraining Tokens: 2 Trillion	Supervised fine-tuning: Over 100,000
70B	Context Length: 4096	Human Preferences: Over 1,000,000

Llama 2 pretrained models are trained on 2 trillion tokens, and have double the context length than Llama 1. Its fine-tuned models have been trained on over 1 million human annotations.

Meta Llama-2 70B: Best Open Source and Commercial LLM (Llama-2, Falcon, MPT)

Benchmark (Higher is better)	MPT (7B)	Falcon (7B)	Llama-2 (7B)	Llama-2 (13B)	MPT (30B)	Falcon (40B)	Llama-1 (65B)	Llama-2 (70B)
MMLU	26.8	26.2	45.3	54.8	46.9	55.4	63.4	68.9
TriviaQA	59.6	56.8	68.9	77.2	71.3	78.6	84.5	85.0
Natural Questions	17.8	18.1	22.7	28.0	23.0	29.5	31.0	33.0
GSM8K	6.8	6.8	14.6	28.7	15.2	19.6	50.9	56.8
HumanEval	18.3	N/A	12.8	18.3	25.0	N/A	23.7	29.9
AGIEval (English tasks only)	23.5	21.2	29.3	39.1	33.8	37.0	47.6	54.2
BoolQ	75.0	67.5	77.4	81.7	79.0	83.1	85.3	85.0

Llama 2 outperforms other open source language models on many external benchmarks, including reasoning, coding, proficiency, and knowledge tests.

Llama-2: Comparison to closed-source models (GPT-3.5, GPT-4, PaLM) on academic benchmarks

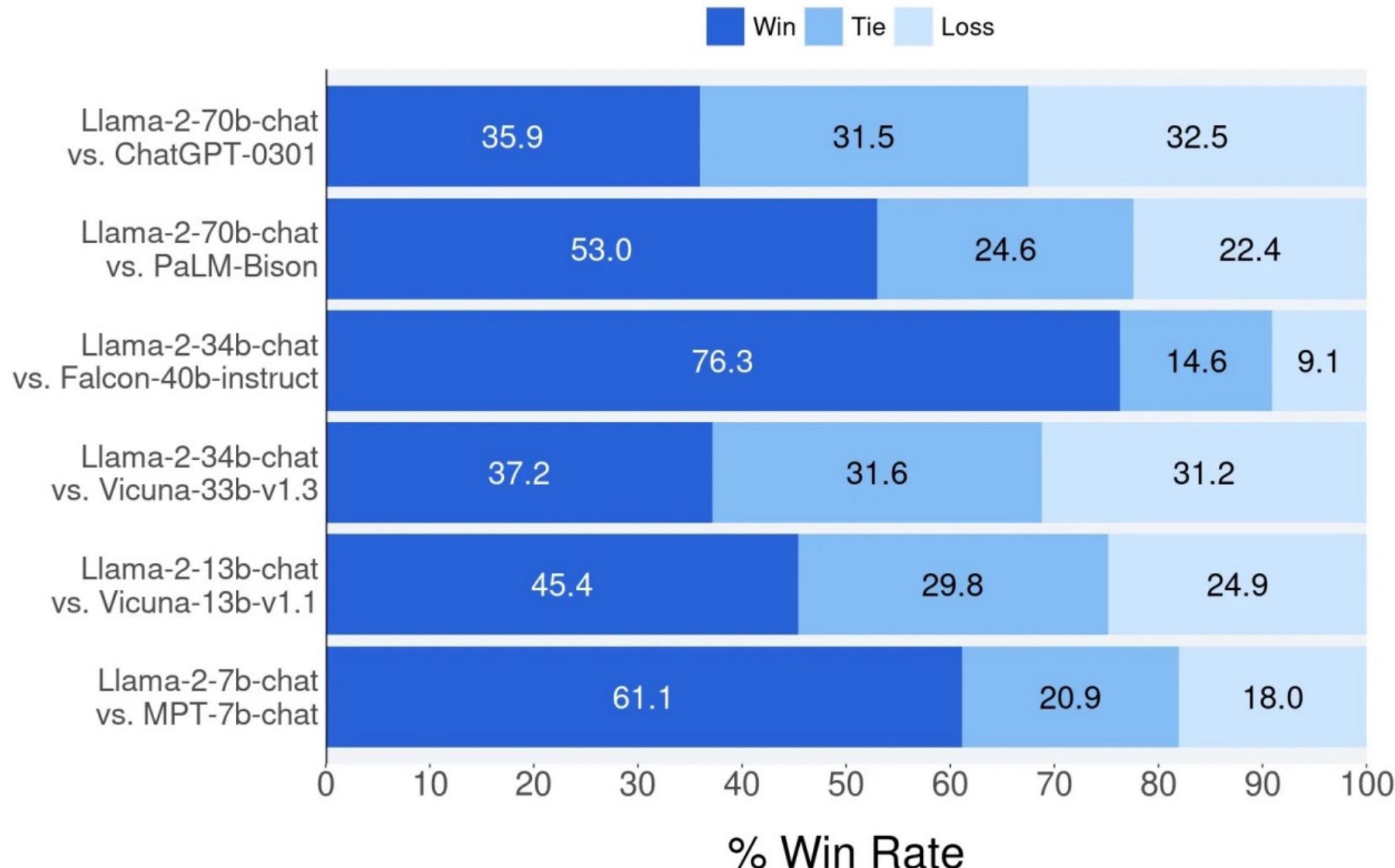
Benchmark (shots)	GPT-3.5	GPT-4	PaLM	PaLM-2-L	LLAMA 2
MMLU (5-shot)	70.0	86.4	69.3	78.3	68.9
TriviaQA (1-shot)	–	–	81.4	86.1	85.0
Natural Questions (1-shot)	–	–	29.3	37.5	33.0
GSM8K (8-shot)	57.1	92.0	56.5	80.7	56.8
HumanEval (0-shot)	48.1	67.0	26.2	–	29.9
BIG-Bench Hard (3-shot)	–	–	52.3	65.7	51.2

Results for GPT-3.5 and GPT-4 are from OpenAI (2023).

Results for the PaLM model are from Chowdhery et al. (2022).

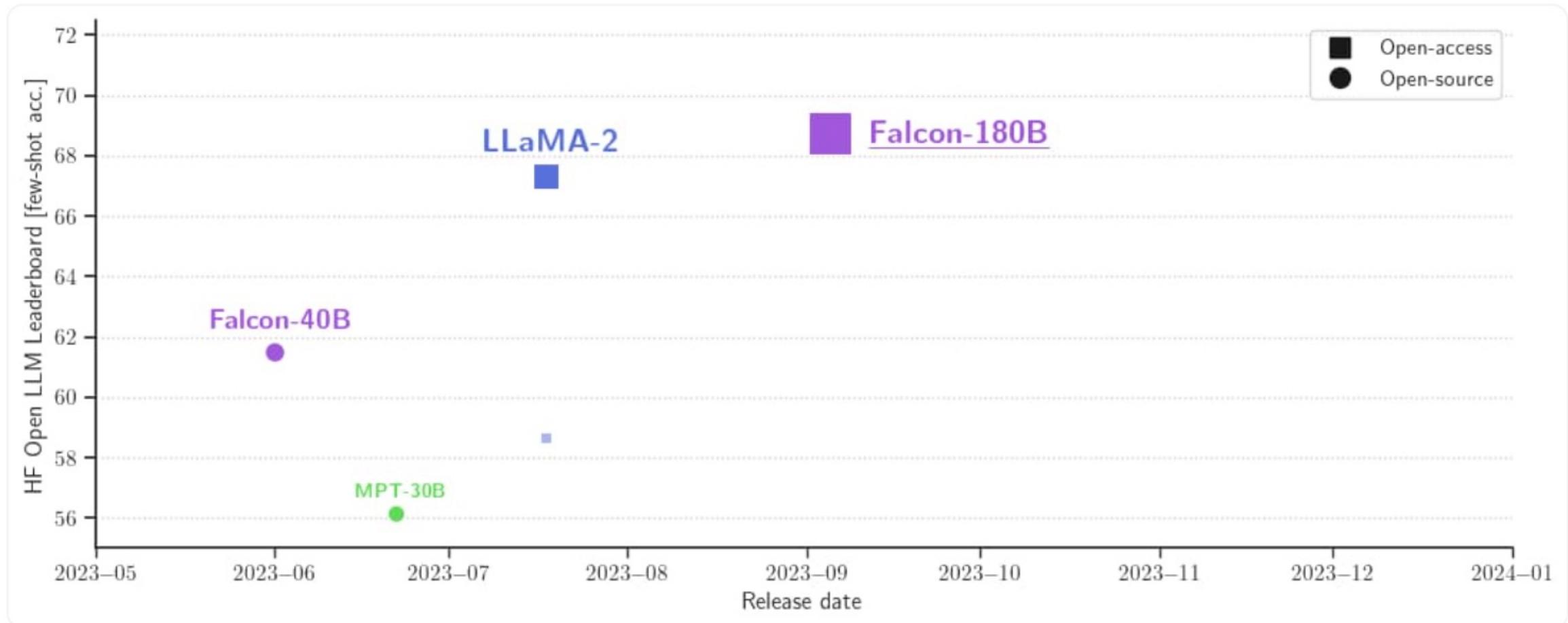
Results for the PaLM-2-L are from Anil et al. (2023).

Llama-2 Chat: Helpfulness Human Evaluation





Falcon 180B





Falcon 180B, LLaMA 65B, MPT 30B

Model	Size	Leaderboard score	Commercial use or license	Pretraining length
Falcon	180B	68.74	🟡	3,500B
Llama 2	70B	67.35	🟡	2,000B
LLaMA	65B	64.23	🔴	1,400B
Falcon	40B	61.48	🟢	1,000B
MPT	30B	56.15	🟢	1,000B



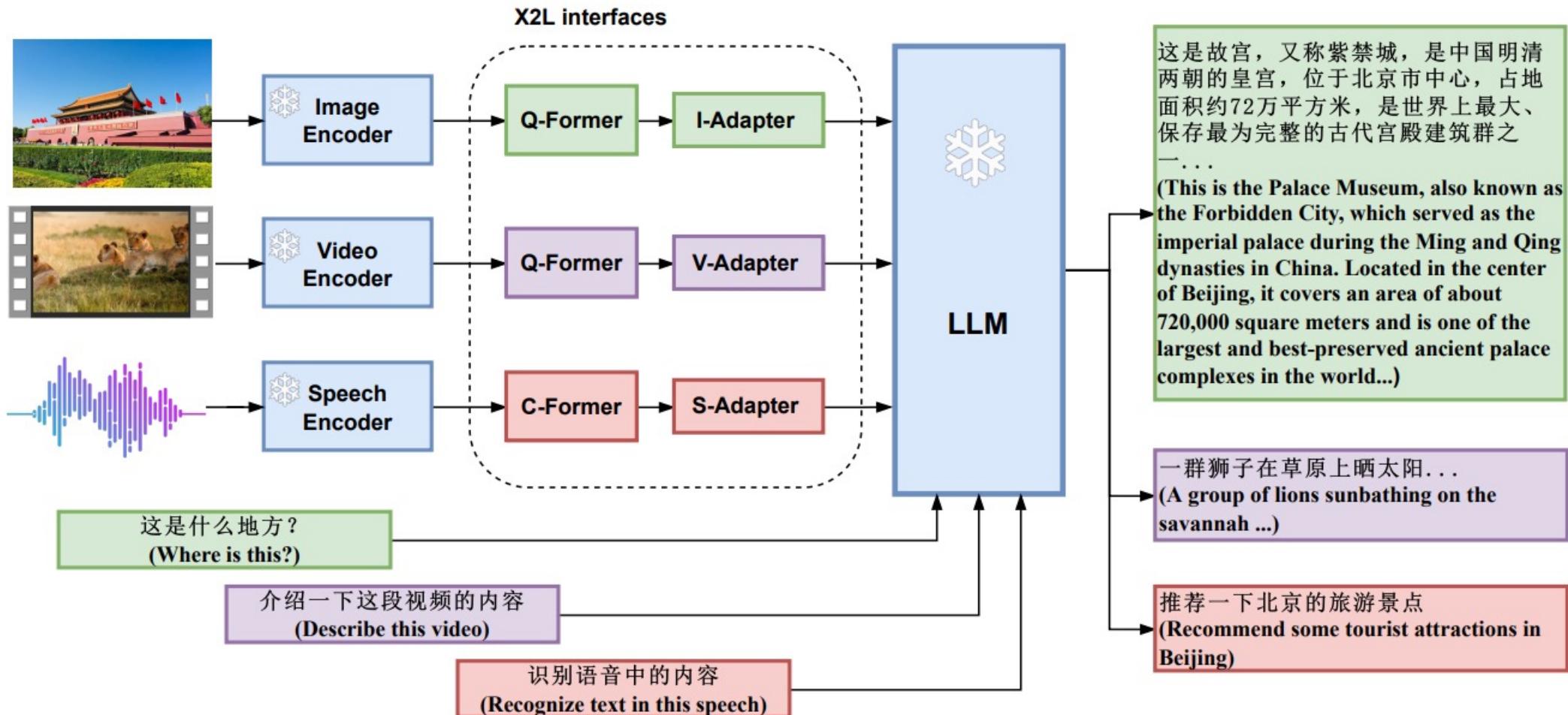
Falcon 180B

Hardware requirements

NVIDIA A100 80 GB:
\$16,135

Type	Kind	Memory	Example
Falcon 180B	Training	Full fine-tuning	5120GB 8x 8x A100 80GB
Falcon 180B	Training	LoRA with ZeRO-3	1280GB 2x 8x A100 80GB
Falcon 180B	Training	QLoRA	160GB 2x A100 80GB
Falcon 180B	Inference	BF16/FP16	640GB 8x A100 80GB
Falcon 180B	Inference	GPTQ/int4	320GB 8x A100 40GB

X-LLM: Bootstrapping Advanced Large Language Models by Treating Multi-Modalities as Foreign Languages



Stable Diffusion

Hugging Face Models Datasets Spaces Docs Solutions Pricing

Spaces: stabilityai/stable-diffusion like 1.89k Running

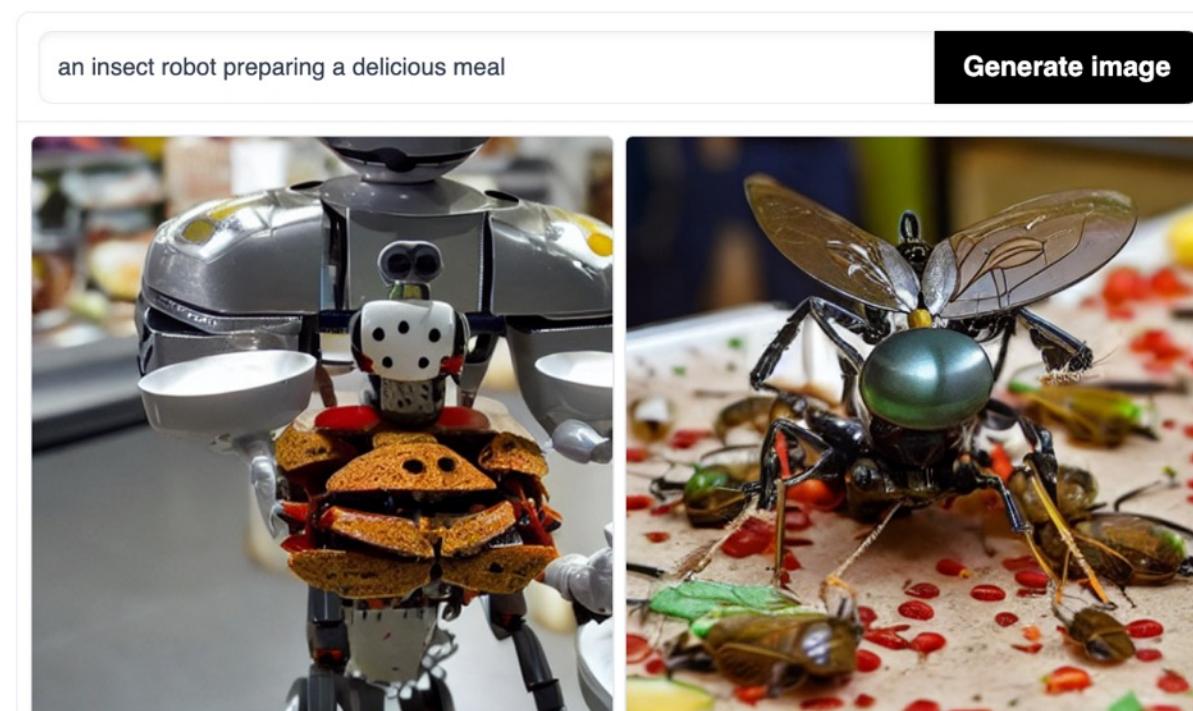
App Files Community 241

Linked Models

Stable Diffusion Demo

Stable Diffusion is a state of the art text-to-image model that generates images from text.

For faster generation and forthcoming API access you can try [DreamStudio Beta](#)



<https://huggingface.co/spaces/stabilityai/stable-diffusion>

Stable Diffusion Colab

woctezuma / stable-diffusion-colab Public

Notifications Fork 7 Star 31

<> Code Issues Pull requests Actions Projects Wiki Security Insights

main ▾ 1 branch 0 tags Go to file Code ▾

woctezuma README: add a reference for sampler schedules 37bc02d 24 days ago 18 commits

LICENSE Initial commit 27 days ago

README.md README: add a reference for sampler schedules 24 days ago

stable_diffusion.ipynb Allow to choose the scheduler 25 days ago

☰ README.md

Stable-Diffusion-Colab

The goal of this repository is to provide a Colab notebook to run the text-to-image "Stable Diffusion" model [1].

Usage

- Run [stable_diffusion.ipynb](#) 

About

Colab notebook to run Stable Diffusion.

[github.com/CompVis/stable-diffusion](#)

deep-learning colab image-generation
text-to-image diffusion text2image
colaboratory google-colab
colab-notebook google-colaboratory
google-colab-notebook
text-to-image-synthesis huggingface
diffusion-models
text-to-image-generation latent-diffusion
stable-diffusion huggingface-diffusers
diffusers stable-diffusion-diffusers

Readme MIT license 31 stars 2 watching

<https://github.com/woctezuma/stable-diffusion-colab>

Stable Diffusion Reimagine

 Clipdrop ▶ Stable diffusion Reimagine
by stability.ai

Apps ▾

API

Blog

Pricing

Sign-in / Sign-up

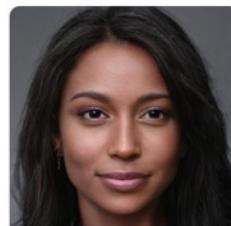
Stable diffusion reimagine

Create multiple variations from a single image.



Click, paste, or drop a file here to start.

↓ Or click on an example below



<https://clipdrop.co/stable-diffusion-reimagine>

Lexica Art: Search Stable Diffusion images and prompts

Lexica

Search images

Search

Lexica

The Stable Diffusion search engine

Join the Discord

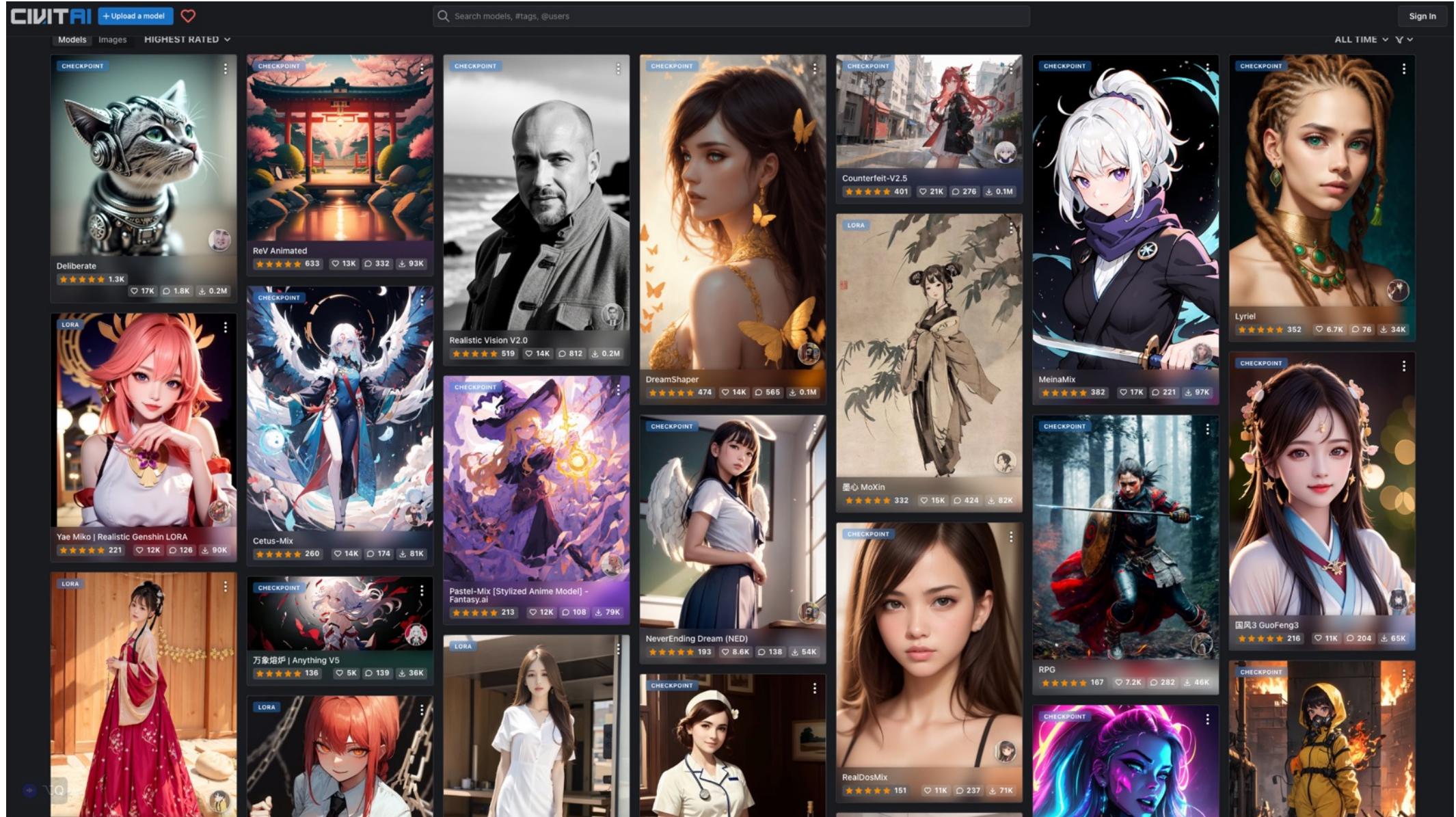
Search for an image

Columns: 10

Search

https://lexica.art/

Civitai: Stable Diffusion AI Art Models



<https://civitai.com/>

D-ID Text to Video

[Products](#)[Technology](#)[Ethics](#)[Pricing](#)[Company](#)[Start Free Trial](#)[Log in](#)

Turn Text To Video In 30 Seconds

Save time and money and enrich your content with engaging videos. Try it today!

[Start Free Trial](#)

Synthesia: #1 AI Video Generation

[Features](#)[Use cases](#)[Pricing](#)[Resources](#)[Company](#)[Log in](#)[Create account](#)

#1 AI VIDEO GENERATION PLATFORM ⓘ

Turn your text into videos in minutes

- Get natural sounding AI voices in 120+ languages
- Make your videos more engaging with 140+ AI Avatars
- Edit as simply as a slide-deck, no experience required

[Create a free AI video](#)[▶ Watch 2 min demo](#)

No credit card required.

The screenshot shows the Synthesia software interface. At the top, there's a toolbar with icons for Avatar, Text, Shape, Media, Record, and communication. Below the toolbar is a sidebar titled "Scenes" with a thumbnail for "SCENE 1". The main area displays a video frame featuring a woman with dark hair, smiling, against a blue abstract background. A volume icon is visible in the bottom left corner of the video frame. At the bottom of the video frame, there's a caption: "Hey, I'm Alex, and this is what an AI Avatar looks and sounds like. Create a free AI video and see for yourself." On the right side of the interface, there's a vertical panel with sections for "Scene 1", "Replace la", "Background", "Background", "Opacity", "Blur", "Music", and "Transition". At the very bottom, there's a progress bar showing "0:00 / 0:09".

<https://www.synthesia.io/>

Speechify: #1 AI Voice Over Generator



Text to Speech

AI Voice Studio

Products

Teams

Edu

About

Log in

Talk to Sales

Try for free

The #1 AI Voice Over Generator

Natural sounding, human-quality voice generator for all your content. Try our AI voice today, for free!

Type text here

Introducing the ultimate voiceover tool for professionals and amateurs – a powerful and easy-to-use software that lets you easily create high-quality voiceovers.

Narrate text, videos, explainers, slides, books – anything – in any style.

Our voiceover product is perfect for businesses, content creators, podcasters, video editors, and anyone else who needs to add professional-quality voiceovers to their projects.

Select Voice

Davis General

Aria Chat

Guy Friendly

Clone My Voice

Add Pause

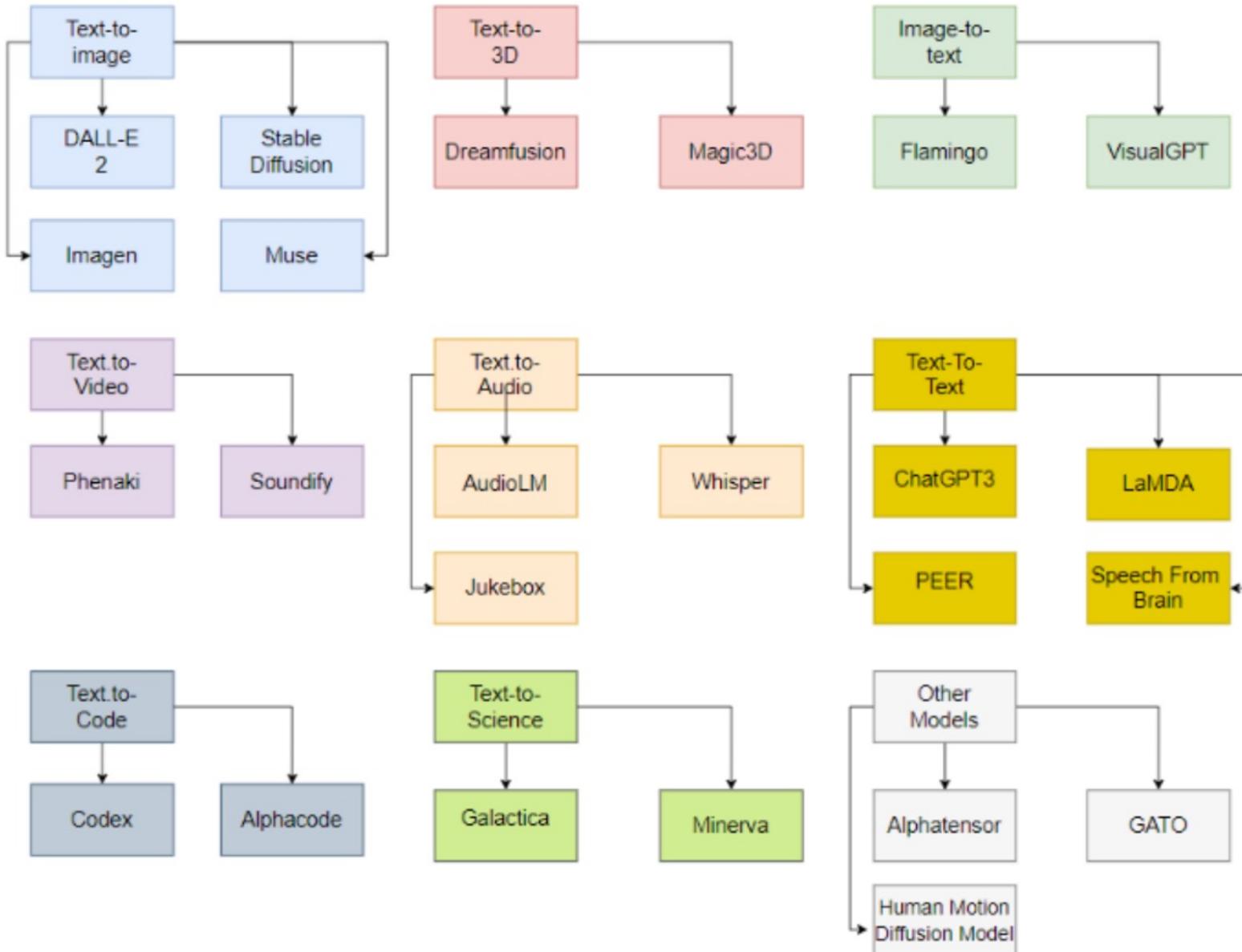
Pause for a few seconds to add emphasis.

Listen With Music

Change music ▾

Create an account to access 200+ high-quality voices and Granular controls on the pitch, tone and speed.

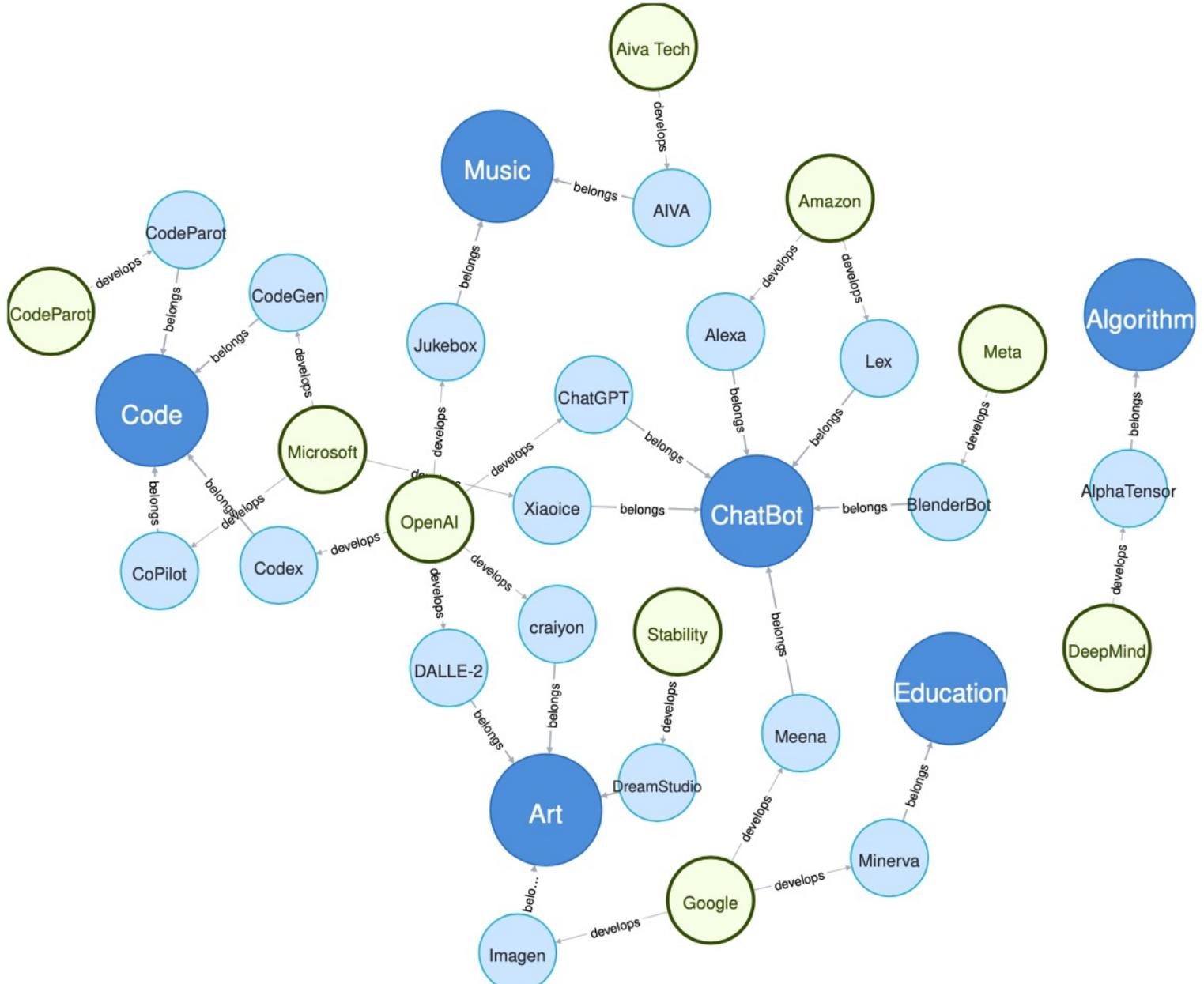
Generative AI Models



**ChatGPT
is not
all you need**

**Attention
is
all you need**

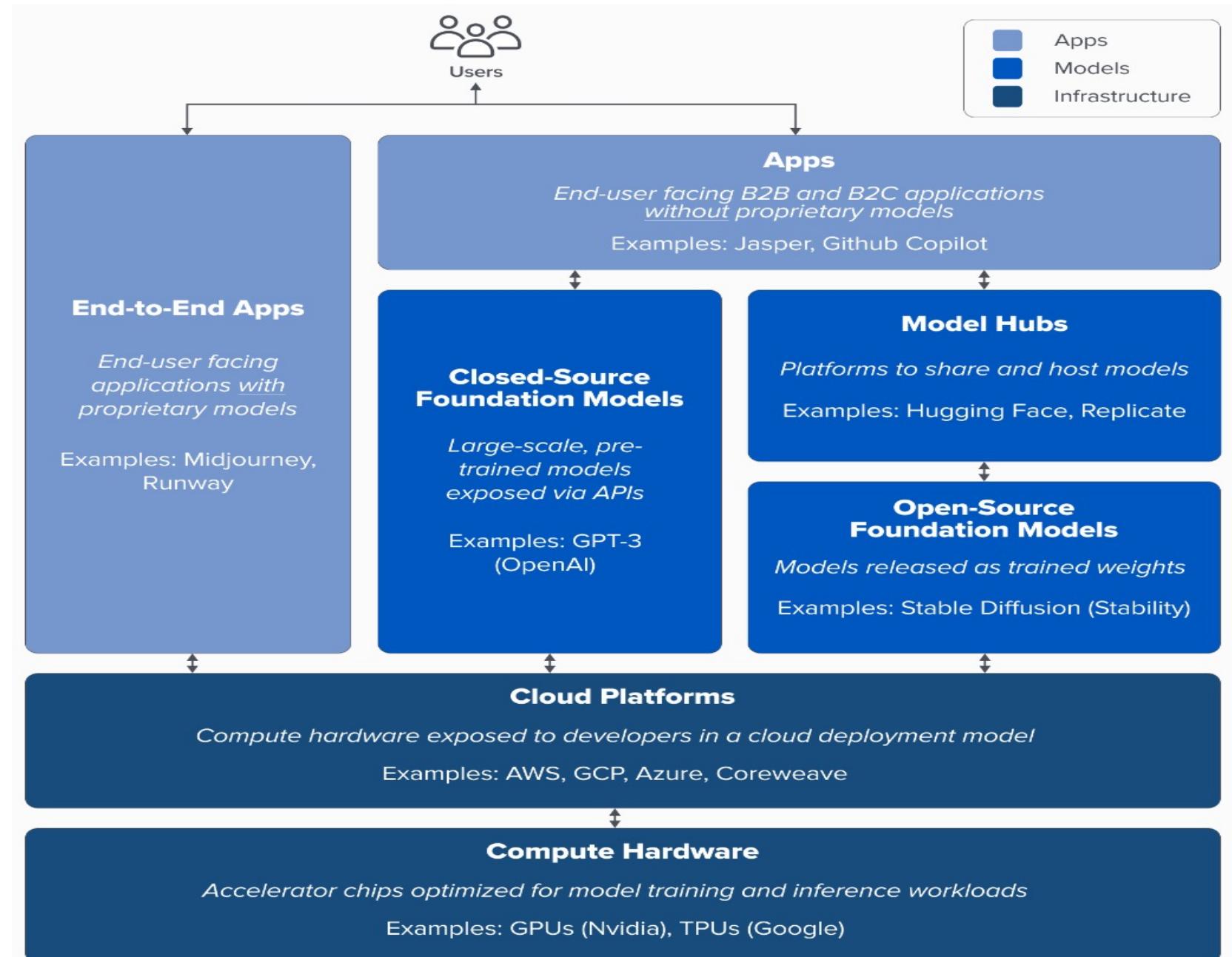
Generative AI Research Areas, Applications and Companies



Applications of Generative AI Models

Application	Platform/Software	Company	Year	Papaer	Link
ChatBot	Xiaoice	Microsoft	2018	[200]	Xiaoice
ChatBot	Meena	Google	2020	[201]	Meena Blog
ChatBot	BlenderBot	Meta	2022	[202]	Blenderbot
ChatBot	ChatGPT	OpenAI	2022	[10]	ChatGPT
ChatBot	Alexa	Amazon	2014	-	Amazon Alexa
ChatBot	Lex	Amazon	2017	-	Amazon Lex
Music	AIVA	Aiva Tech	2016	-	AIVA
Music	Jukebox	OpenAI	2020	[203]	Jukebox
Code	CodeGPT	Microsoft	2021	[204]	CodeGPT
Code	CodeParrot	CodeParrot	2022	[205]	CodeParrot
Code	Codex	OpenAI	2021	[206]	Codex blog
Code	CoPilot	Microsoft	2021	[206]	CoPilot
Art	DALL-E-2	OpenAI	2022	[5]	DALL-E-2 Blog
Art	DreamStudio	Stability	2022	[13]	Dreamstudio
Art	craiyon	OpenAI	2021	[1]	Craiyon
Art	Imagen	Google	2022	[152]	Imagen
Education	Minerva	Google	2022	[207]	Minerva Blog
Algorithm	AlphaTensor	DeepMind	2022	[208]	AlphaTensor

Generative AI Tech Stack



Generative AI Software and Business Factors

Business
Factors



Software

Application

A product utilizing and managing model inputs and outputs

Models

Large language models, image generation, or other ML models

Data

Labeling, evaluation

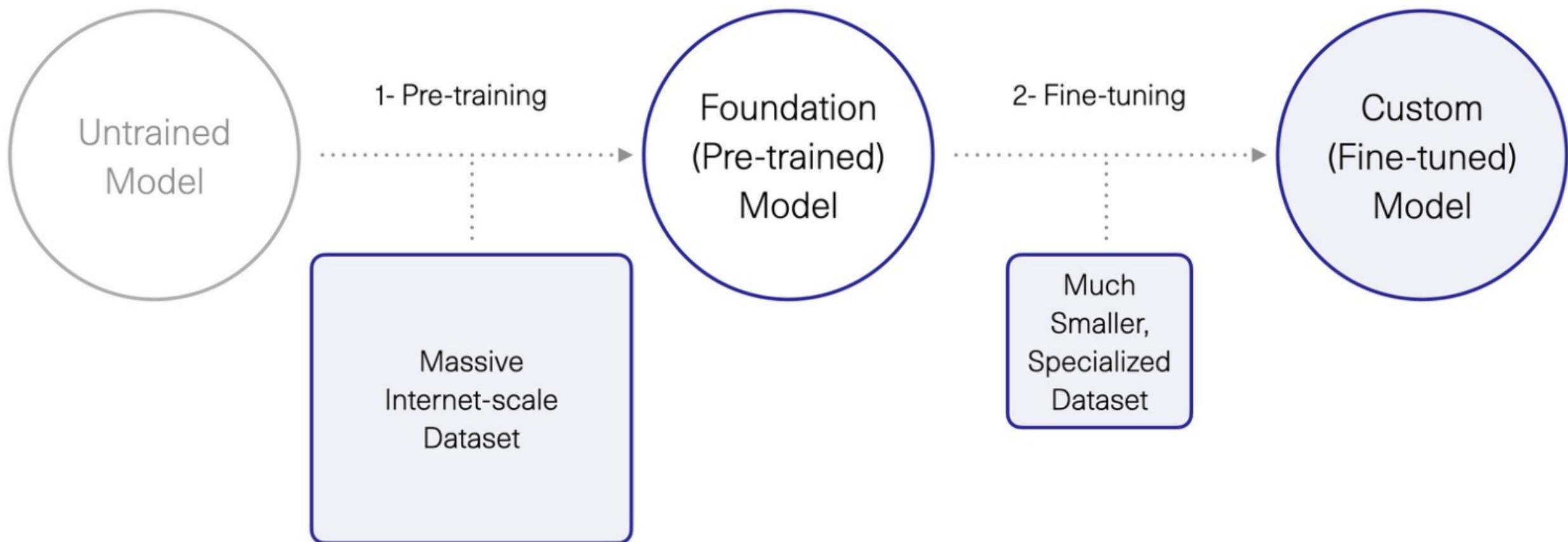
MLOps Model management, tracking

Cloud Platform

Hosting, compute, model deployment and monitoring

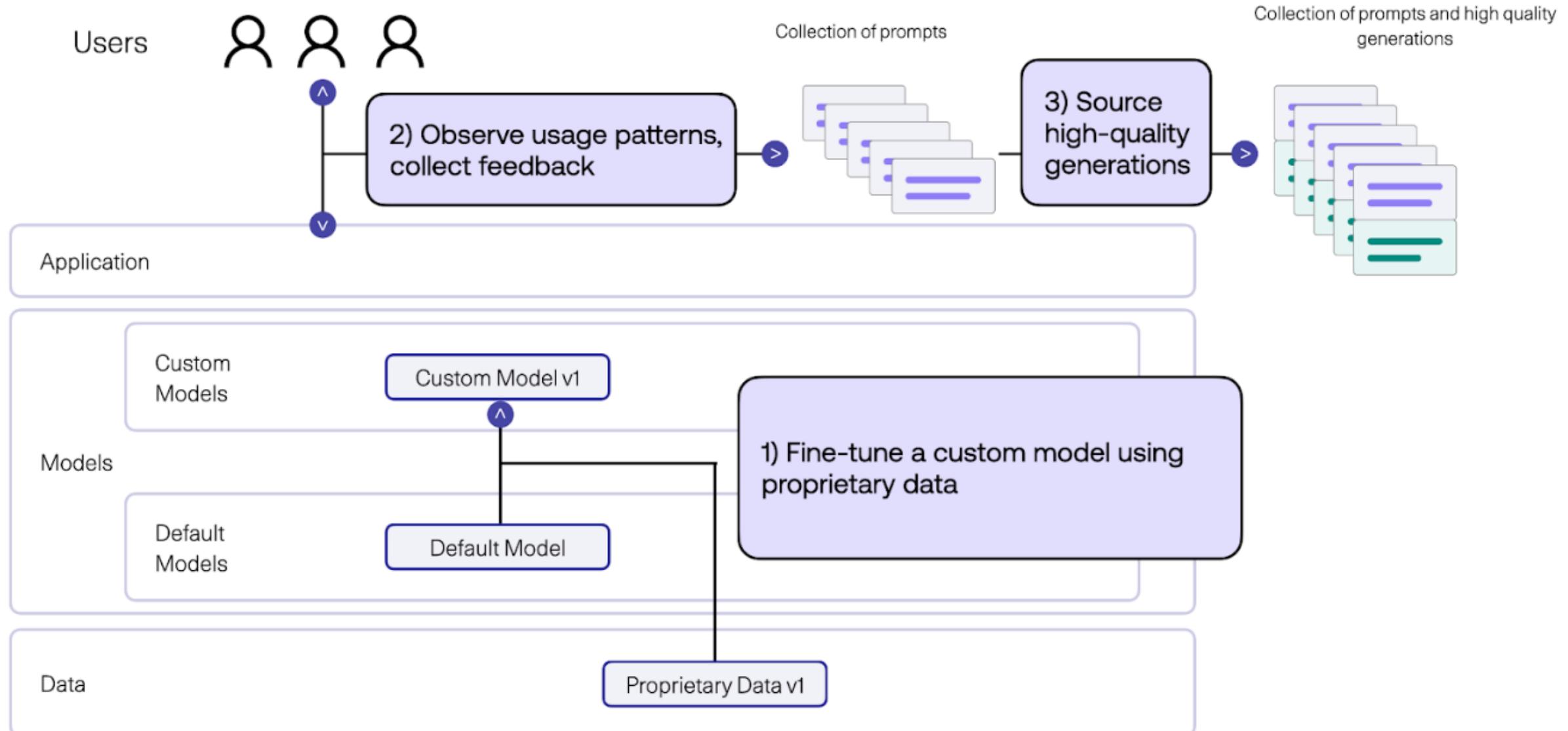
Generative AI

1. Pre-training Foundation (Pre-trained) Model
2. Fine-turning Custom (Fine-tuned) Model



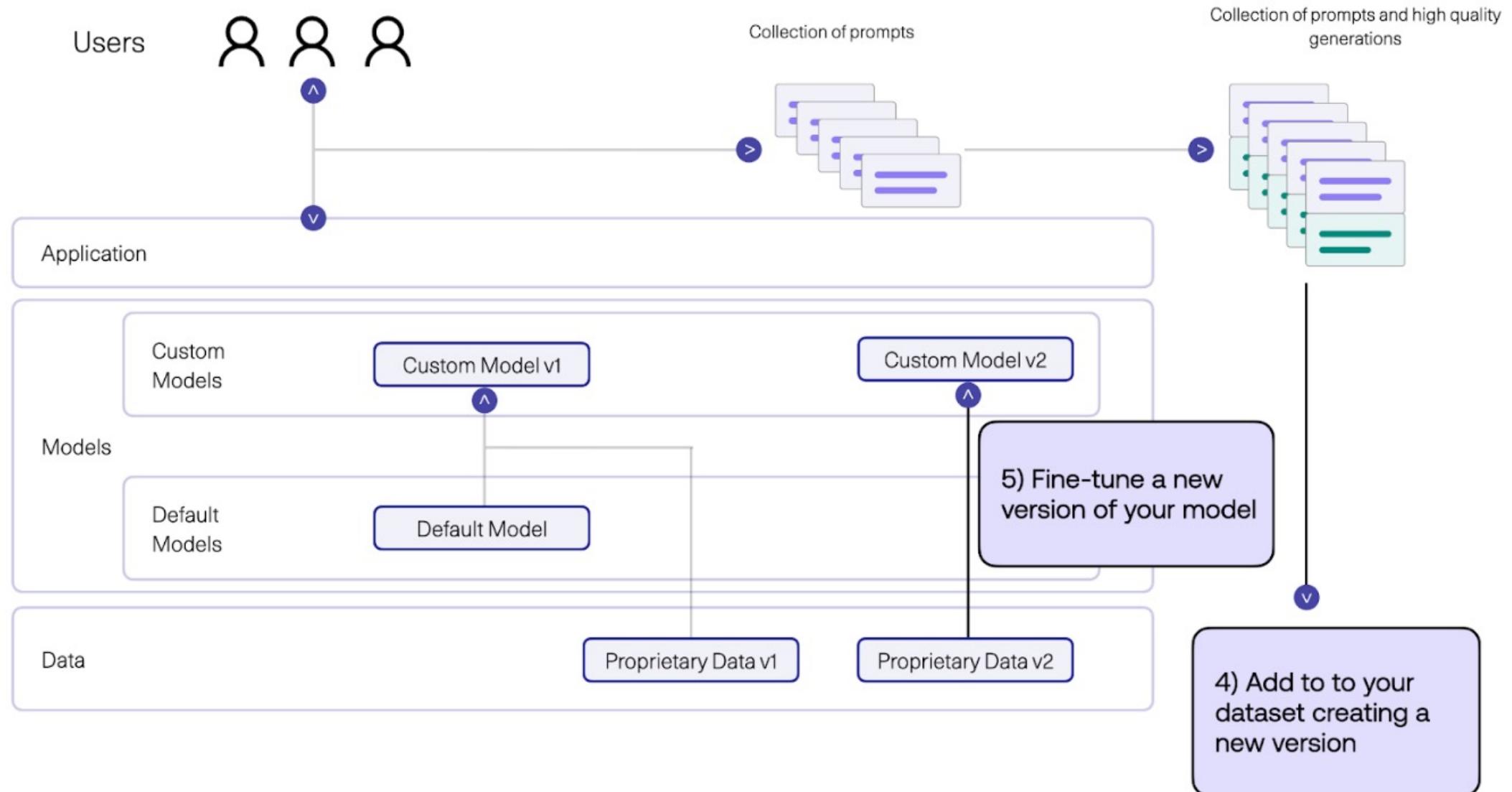
Generative AI

Fine-tune Custom Models using Proprietary Data

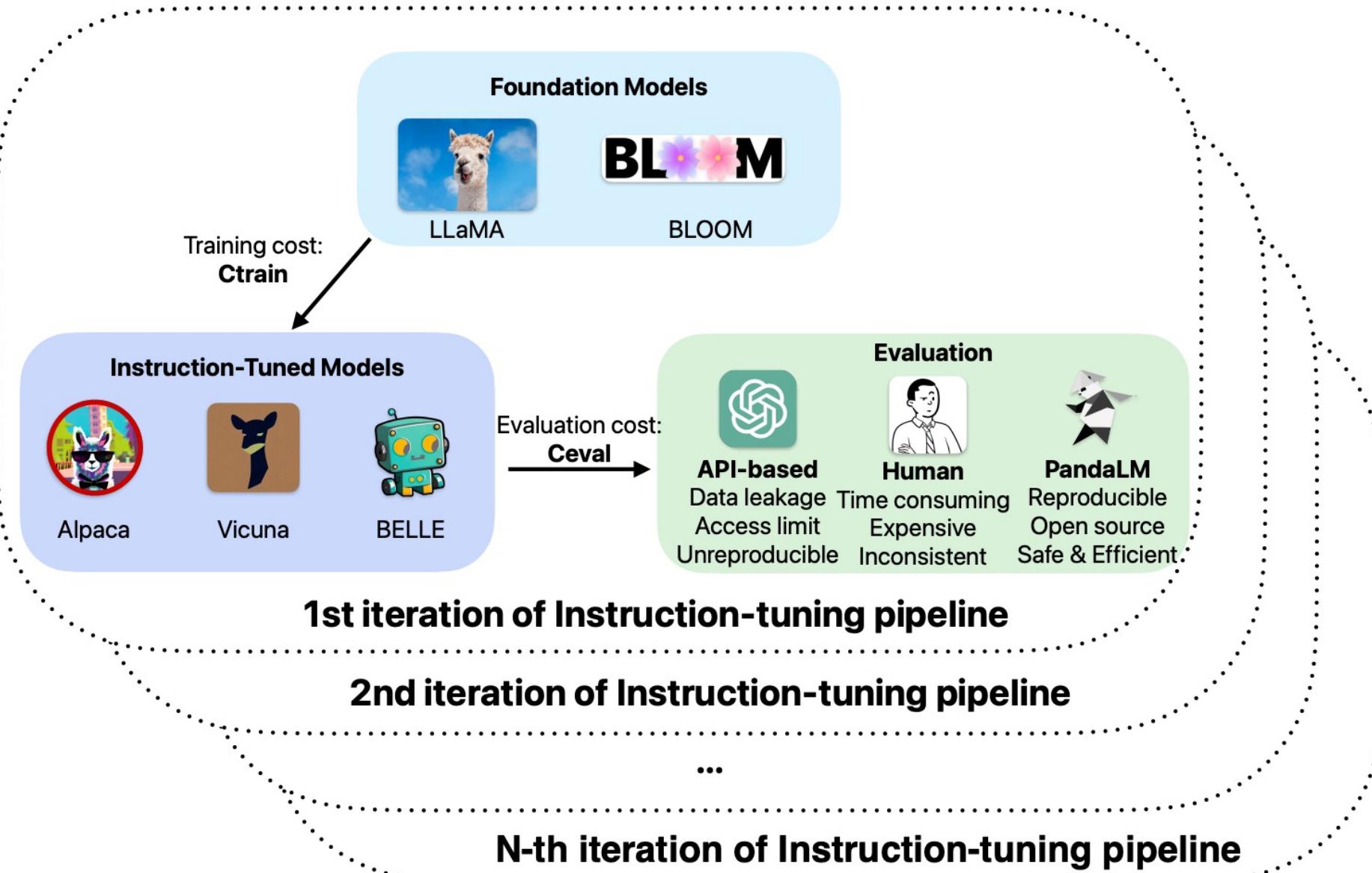


Generative AI

Fine-tune Custom Models using Proprietary Data



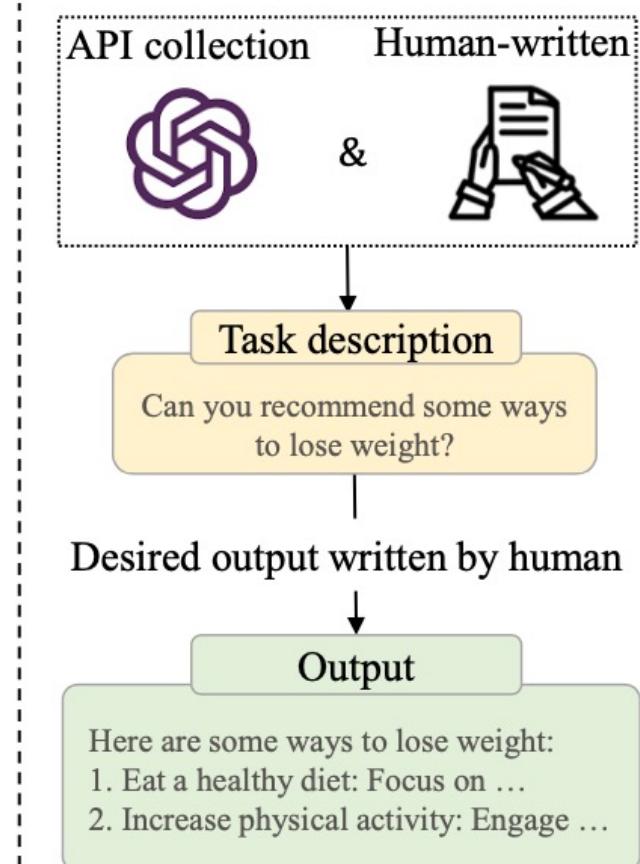
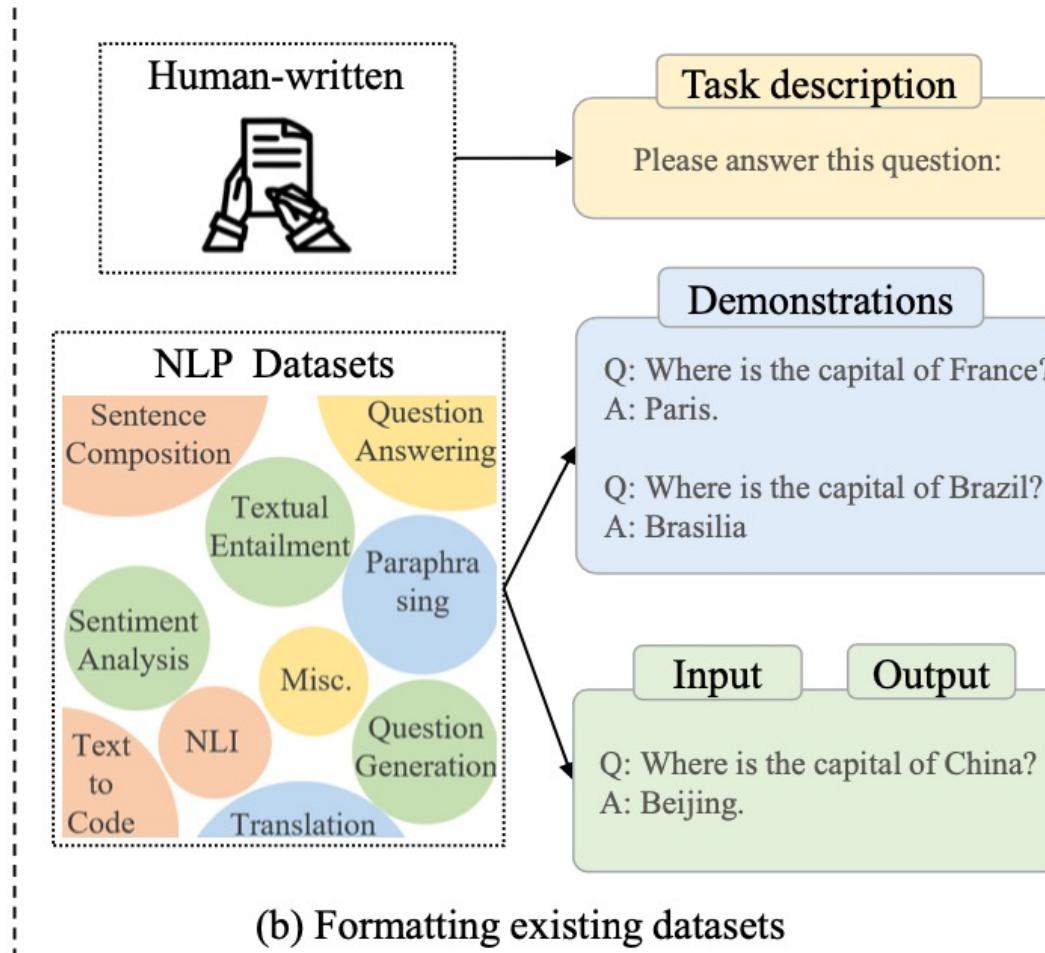
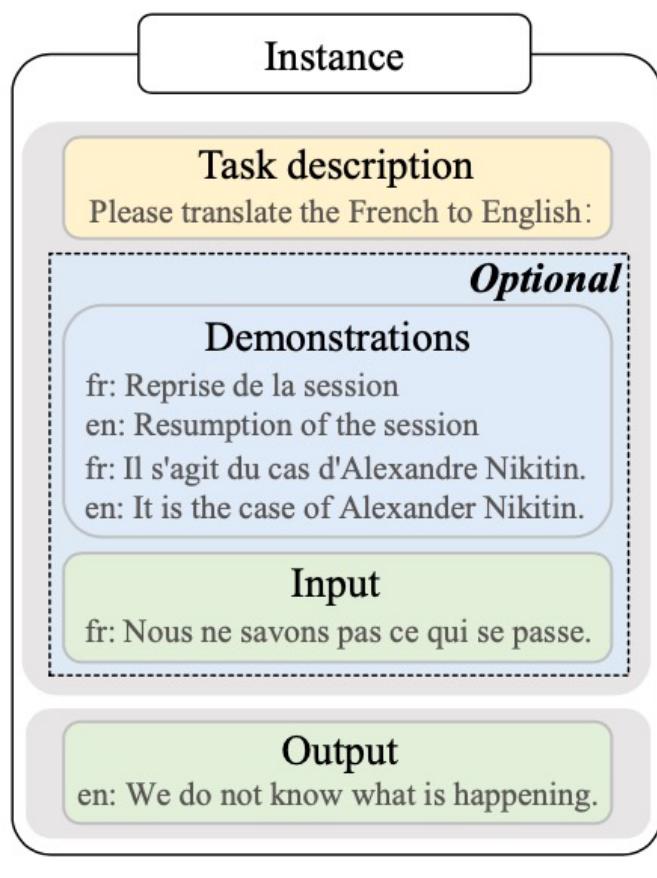
Pipeline of Instruction Tuning LLMs.



Available Task Collections for Instruction Tuning

Collections	Time	#Task types	#Tasks	#Examples
Nat. Inst. [193]	Apr-2021	6	61	193K
CrossFit [194]	Apr-2021	13	160	7.1M
FLAN [62]	Sep-2021	12	62	4.4M
P3 [195]	Oct-2021	13	267	12.1M
ExMix [196]	Nov-2021	11	107	18M
UnifiedSKG [197]	Jan-2022	6	21	812K
Super Nat. Inst. [78]	Apr-2022	76	1616	5M
MVPCorpus [198]	Jun-2022	11	77	41M
xP3 [84]	Nov-2022	17	85	81M
OIG ¹⁴	Mar-2023	-	-	43M

Instance Formatting and Two Different Methods for Constructing the Instruction-formatted Instances



In-context Learning (ICL) and Chain-of-thought (CoT) Prompting

In-Context Learning

Answer the following mathematical reasoning questions:

Q: If you have 12 candies and you give 4 candies to your friend, how many candies do you have left?

A: The answer is 8.

Q: If a rectangle has a length of 6 cm and a width of 3 cm, what is the perimeter of the rectangle?

A: The answer is 18 cm.

Q: Sam has 12 marbles. He gives 1/4 of them to his sister. How many marbles does Sam have left?

A: The answer is 9.

N x

Chain-of-Thought Prompting

Answer the following mathematical reasoning questions:

Q: If a rectangle has a length of 6 cm and a width of 3 cm, what is the perimeter of the rectangle?

A: For a rectangle, add up the length and width and double it. So, the perimeter of this rectangle is $(6 + 3) \times 2 = 18$ cm.

The answer is 18 cm.

Q: Sam has 12 marbles. He gives 1/4 of them to his sister. How many marbles does Sam have left?

A: He gives $(1 / 4) \times 12 = 3$ marbles. So Sam is left with $12 - 3 = 9$ marbles.
The answer is 9.

N x

LLM

: Task description

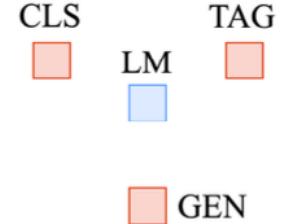
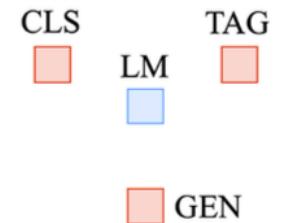
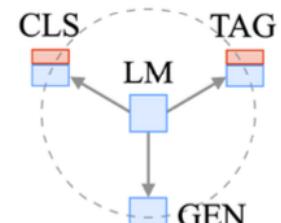
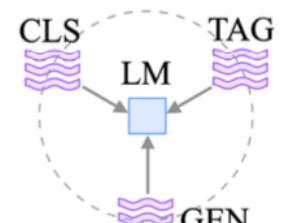
: Demonstration

: Chain-of-Thought

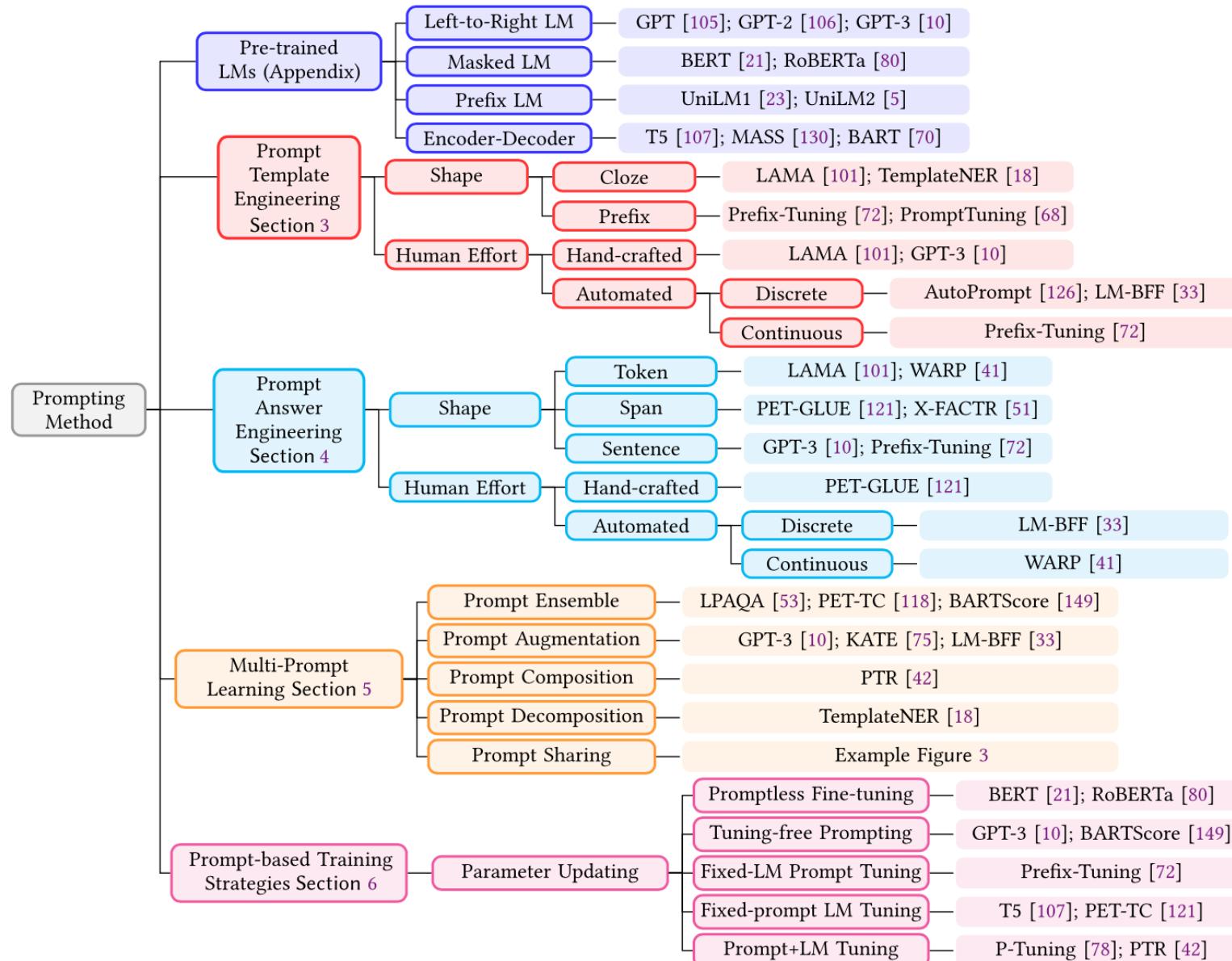
: Query

Pre-train, Prompt, and Predict: Prompting Methods in Natural Language Processing (LLMs)

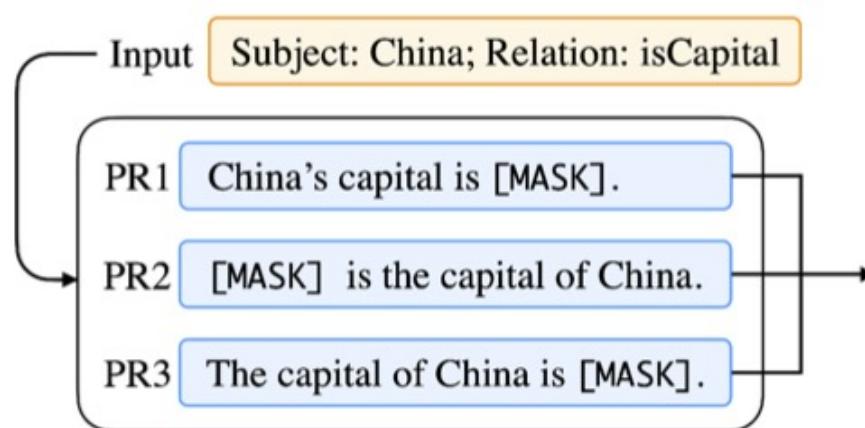
Four Paradigms in NLP

Paradigm	Engineering	Task Relation
a. Fully Supervised Learning (Non-Neural Network)	Feature (e.g. word identity, part-of-speech, sentence length)	
b. Fully Supervised Learning (Neural Network)	Architecture (e.g. convolutional, recurrent, self-attentional)	
c. Pre-train, Fine-tune	Objective (e.g. masked language modeling, next sentence prediction)	
d. Pre-train, Prompt, Predict	Prompt (e.g. cloze, prefix)	

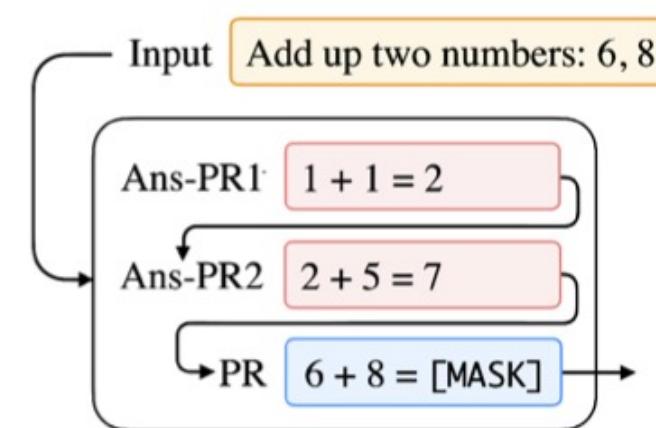
Typology of Prompting Methods



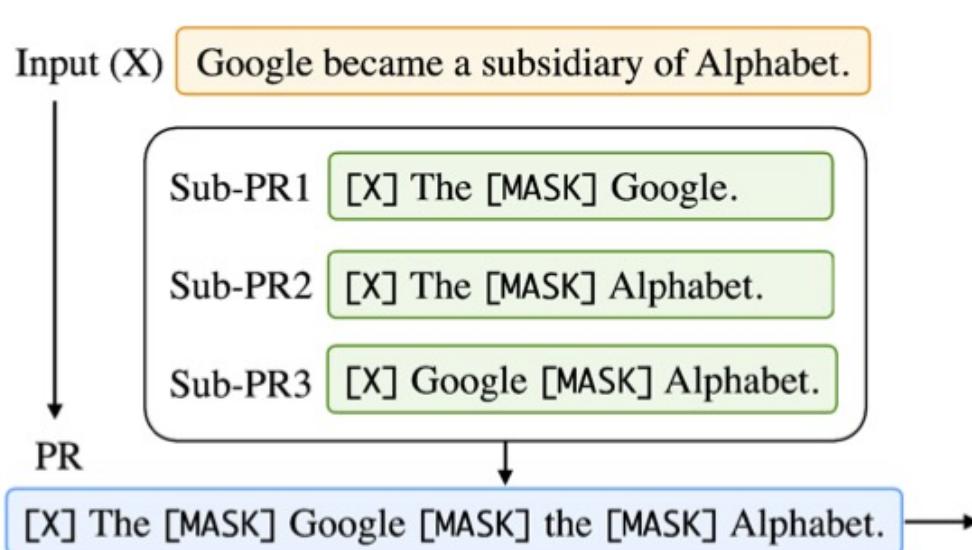
Different Multi-Prompt Learning Strategies



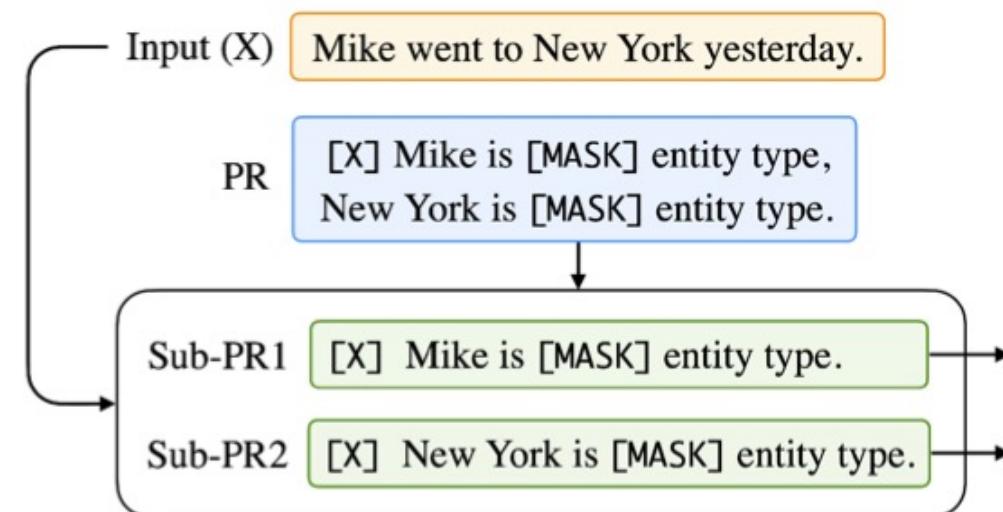
(a) Prompt Ensembling.



(b) Prompt Augmentation.



(c) Prompt Composition.



(d) Prompt Decomposition.

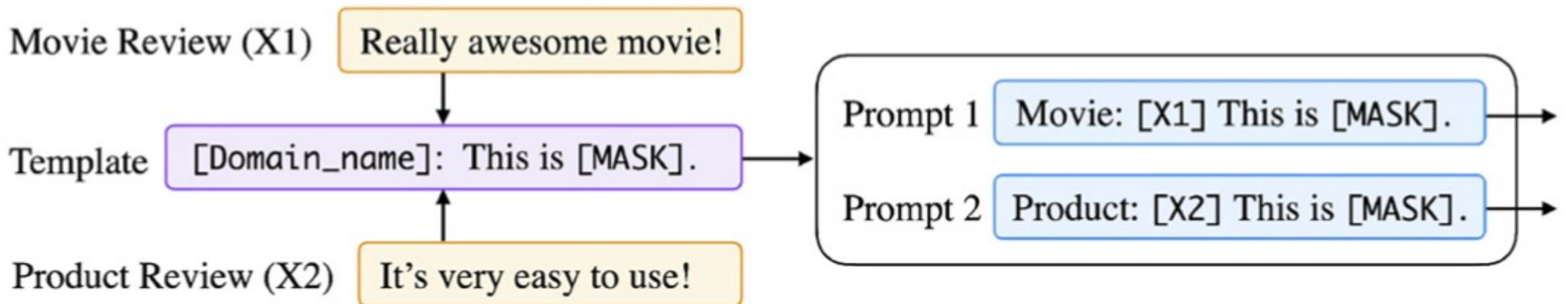
Examples of Input, Template, and Answer for Different Tasks

Type	Task Example	Input ([X])	Template	Answer ([Z])
Text Classification	Sentiment	I love this movie.	[X] The movie is [Z].	great fantastic ...
	Topics	He prompted the LM.	[X] The text is about [Z].	sports science ...
	Intention	What is taxi fare to Denver?	[X] The question is about [Z].	quantity city ...
Text-span Classification	Aspect Sentiment	Poor service but good food.	[X] What about service? [Z].	Bad Terrible ...
				Yes No ...
				...
Text-pair Classification	Natural Language Inference	[X1]: An old man with ...	[X1]? [Z], [X2]	Yes No ...
		[X2]: A man walks
		[X1]: Mike went to Paris.		organization location ...
Tagging	Named Entity Recognition	[X2]: Paris	[X1][X2] is a [Z] entity.	...
		[X1]: Mike went to Paris.		The victim ... A woman
		[X2]: Paris		...
Text Generation	Summarization	Las Vegas police ...	[X] TL;DR: [Z]	The victim ... A woman
				...
				I love you. I fancy you. ...
Regression	Textual Similarity	[X1]: A man is smoking.	[X1] [Z], [X2]	Yes No ...
		[X2]: A man is skating.		...
		[X1]: A man is smoking.		...

Characteristics of Different Tuning Strategies

Strategy	LM Params	Prompt Params		Example
		Additional	Tuned	
Promptless Fine-tuning	Tuned	—	—	ELMo [97], BERT [20], BART [69]
Tuning-free Prompting	Frozen	✗	✗	GPT-3 [9], AutoPrompt [125], LAMA [100]
Fixed-LM Prompt Tuning	Frozen	✓	Tuned	Prefix-Tuning [71], Prompt-Tuning [67]
Fixed-prompt LM Tuning	Tuned	✗	✗	PET-TC [117], PET-Gen [118], LM-BFF [32]
Prompt+LM Fine-tuning	Tuned	✓	Tuned	PADA [5], P-Tuning [77], PTR [41]

Multi-prompt Learning for Multi-task, Multi-domain, or Multi-lingual Learning



GPT-3.5 Prompt Engineering for Question Answering

- Find the answer to the question from the given context.
- When the question cannot be answered with the given context, say "unanswerable".
- Just say the answer without repeating the question.
- Context: {context}
- Question:{question}
- Answer:

Prompts and QA Inference For FLAN T5

Question Answering

- Prompts and QA Inference For FLAN T5, we follow [41] and use the following prompt:
- Context: {context}\nQuestion: {question}\nAnswer:
- Context: {context}
- Question: {question}
- Answer:

Prompt Engineering

- **Prompts For FLAN models**
 - Longpre, S., Hou, L., Vu, T., Webson, A., Chung, H. W., Tay, Y., ... & Roberts, A. (2023). The flan collection: Designing data and methods for effective instruction tuning. *arXiv preprint arXiv:2301.13688*.
- **MNLI, NLI-FEVER, VitaminC:**
 - "Premise: {premise}\n\nHypothesis: {hypothesis}\n\nDoes the premise entail the hypothesis?\n\nA yes\nB it is not possible to tell\nC no"
- **ANLI:**
 - "{context}\n\nBased on the paragraph above can we conclude that\"{hypothesis}\"?\n\nA Yes\nB It's impossible to say\nC No"
- **SNLI:**
 - "If \"{premise}\", does this mean that \"{hypothesis}\"?\n\nA yes\nB it is not possible to tell\nC no"

Fine-tuning LLM for Dialogue System

**Reinforcement Learning from
Human Feedback (RLHF)**

ChatGPT:
Optimizing Language Models for Dialogue

Reinforcement Learning from Human Feedback (RLHF)

ChatGPT: Optimizing Language Models for Dialogue

Step 1

Collect demonstration data and train a supervised policy.

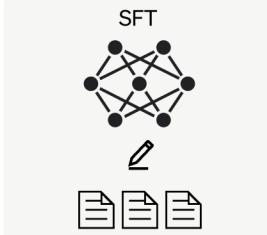
A prompt is sampled from our prompt dataset.



A labeler demonstrates the desired output behavior.



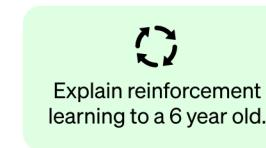
This data is used to fine-tune GPT-3.5 with supervised learning.



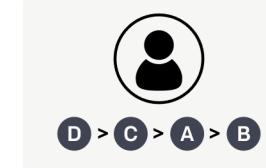
Step 2

Collect comparison data and train a reward model.

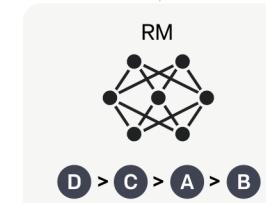
A prompt and several model outputs are sampled.



A labeler ranks the outputs from best to worst.



This data is used to train our reward model.



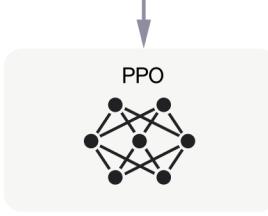
Step 3

Optimize a policy against the reward model using the PPO reinforcement learning algorithm.

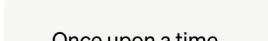
A new prompt is sampled from the dataset.



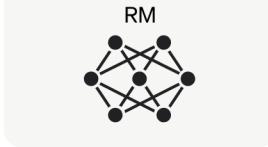
The PPO model is initialized from the supervised policy.



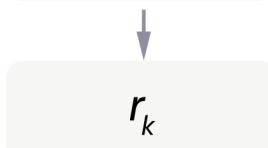
The policy generates an output.



The reward model calculates a reward for the output.



The reward is used to update the policy using PPO.



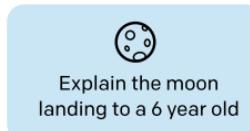
Training language models to follow instructions with human feedback

InstructGPT and GPT 3.5

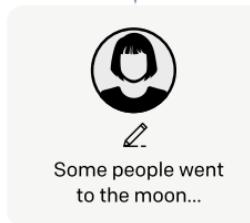
Step 1

Collect demonstration data, and train a supervised policy.

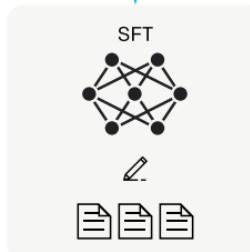
A prompt is sampled from our prompt dataset.



A labeler demonstrates the desired output behavior.



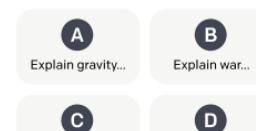
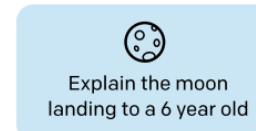
This data is used to fine-tune GPT-3 with supervised learning.



Step 2

Collect comparison data, and train a reward model.

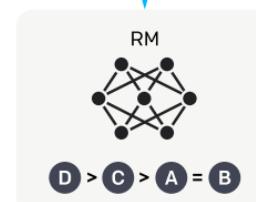
A prompt and several model outputs are sampled.



A labeler ranks the outputs from best to worst.



This data is used to train our reward model.



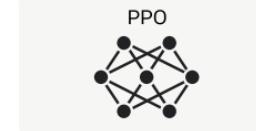
Step 3

Optimize a policy against the reward model using reinforcement learning.

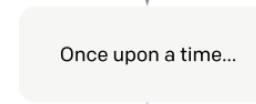
A new prompt is sampled from the dataset.



The policy generates an output.



The reward model calculates a reward for the output.



The reward is used to update the policy using PPO.



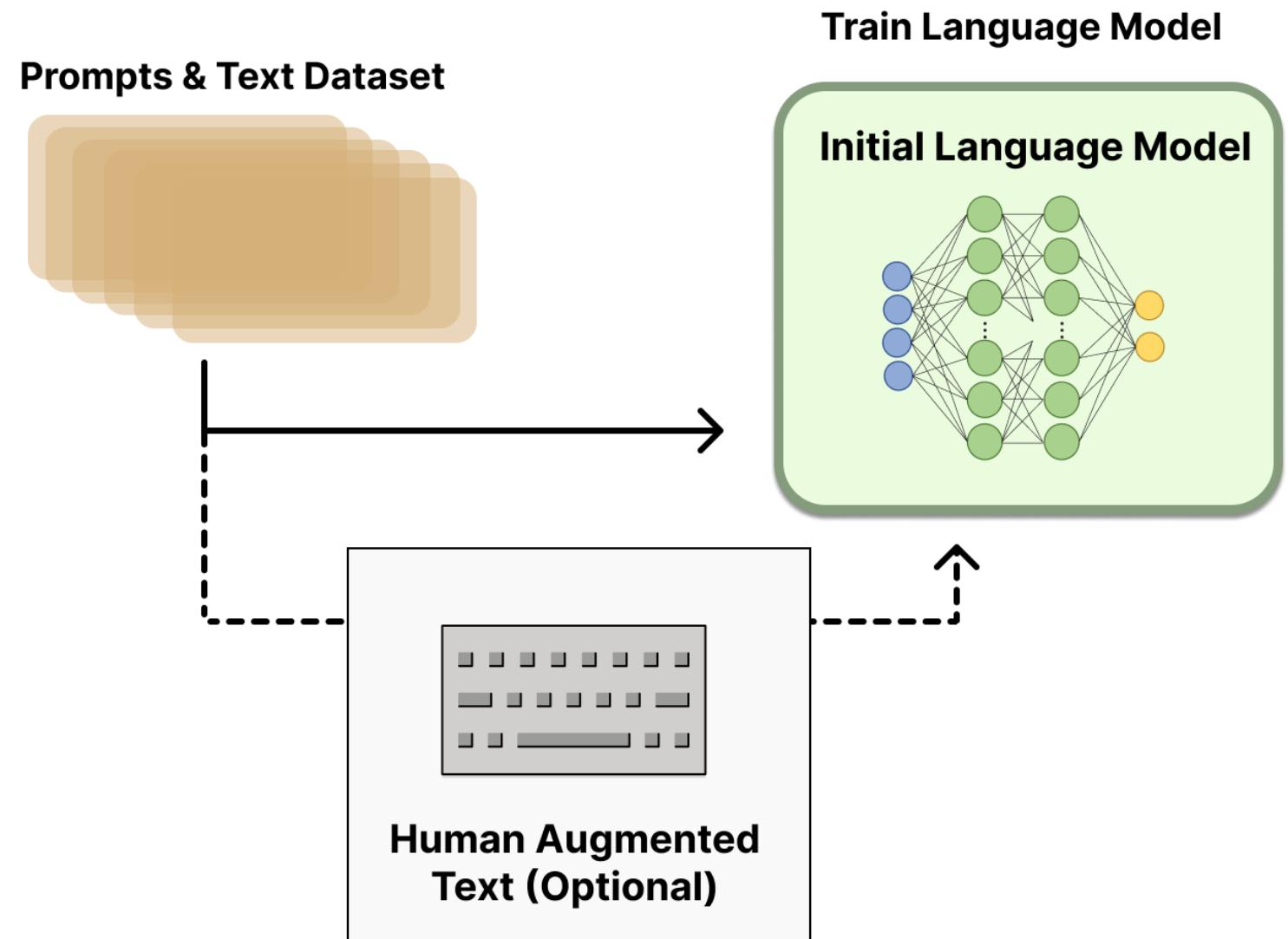
r_k

Reinforcement Learning from Human Feedback (RLHF)

- 1. Pretraining a Language Model (LM)**
- 2. Gathering Data and Training a Reward Model**
- 3. Fine-tuning the LM with Reinforcement Learning**

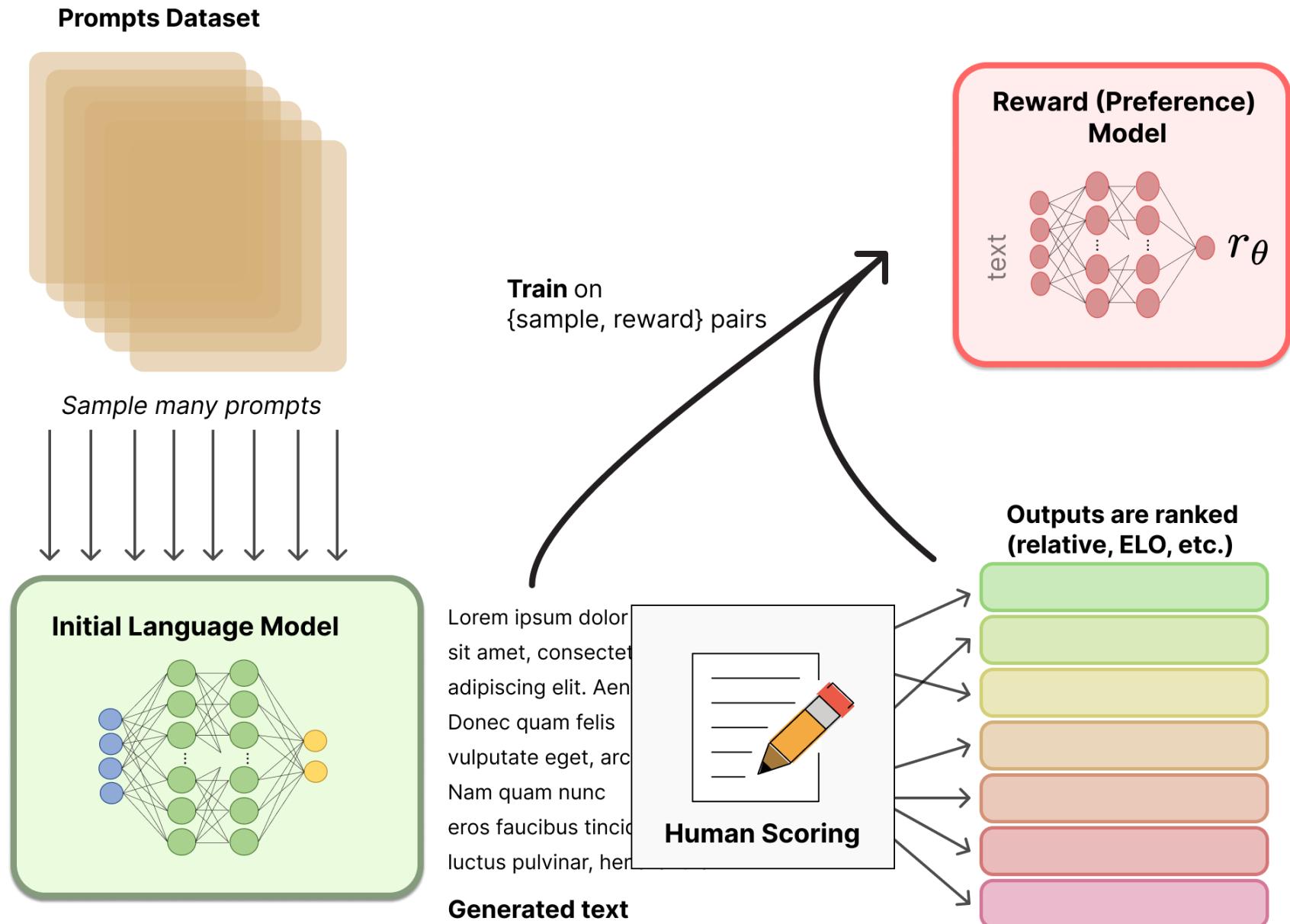
Reinforcement Learning from Human Feedback (RLHF)

**Step 1. Pretraining
a Language Model
(LM)**



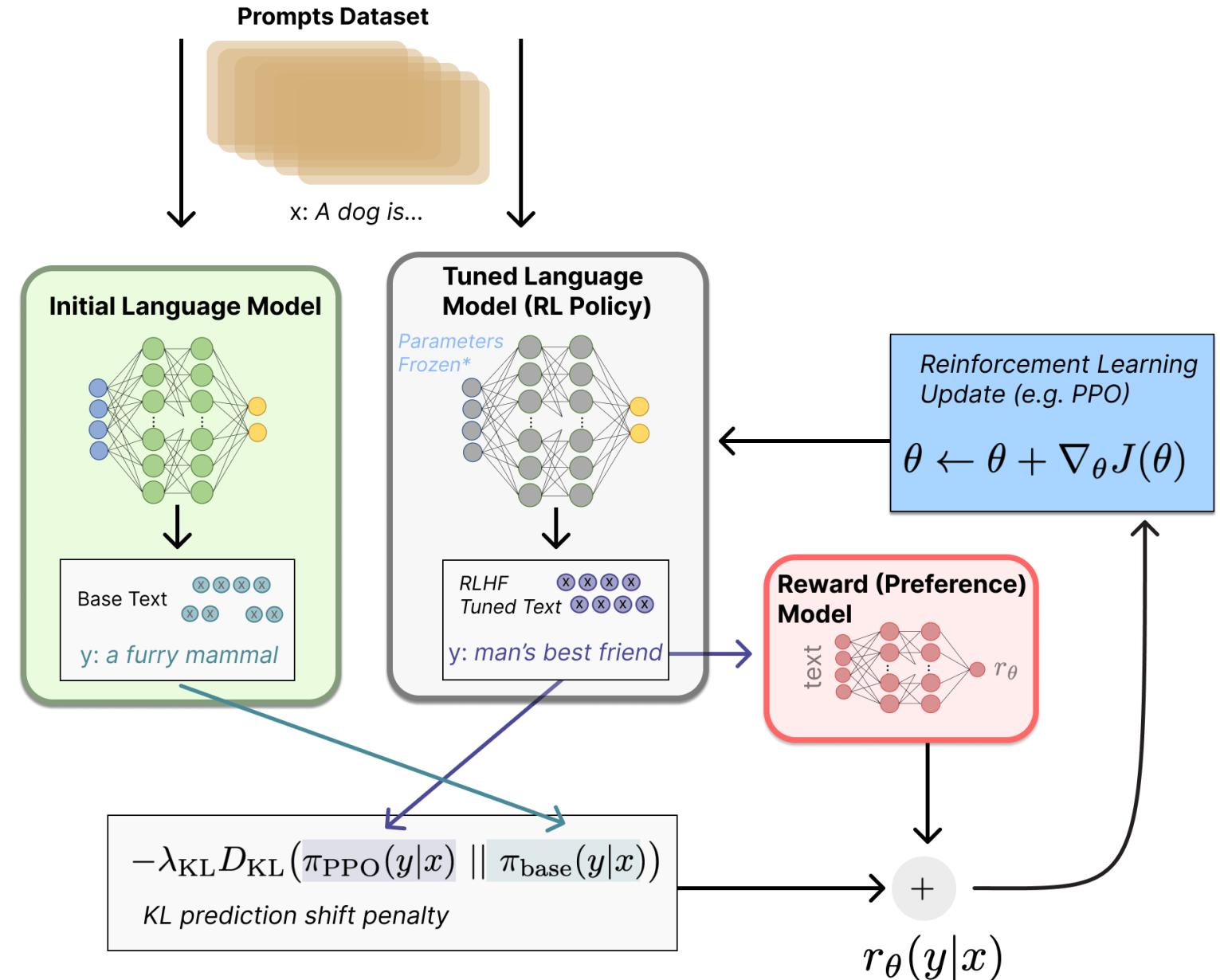
Reinforcement Learning from Human Feedback (RLHF)

Step 2. Gathering Data and Training a Reward Model

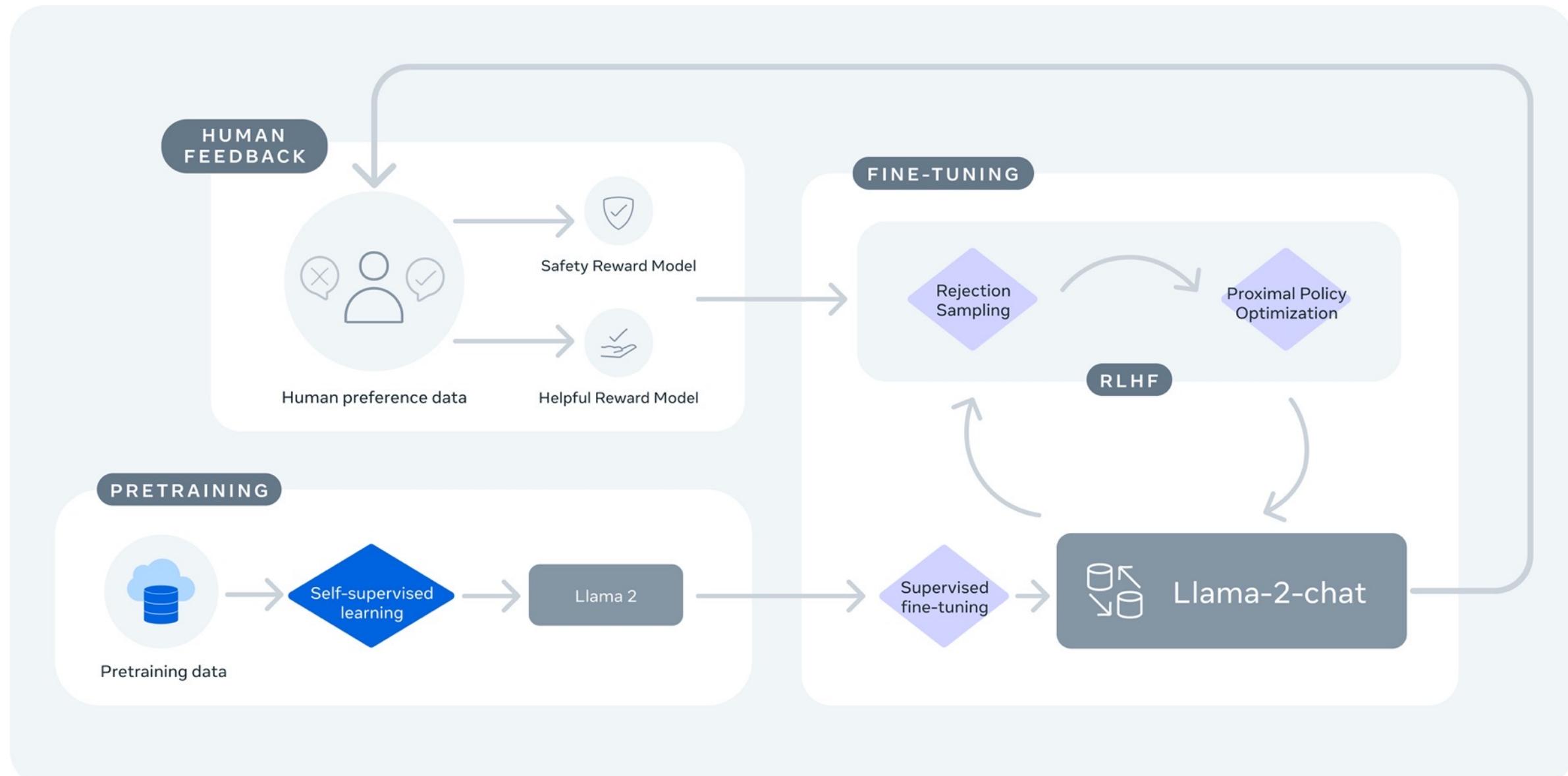


Reinforcement Learning from Human Feedback (RLHF)

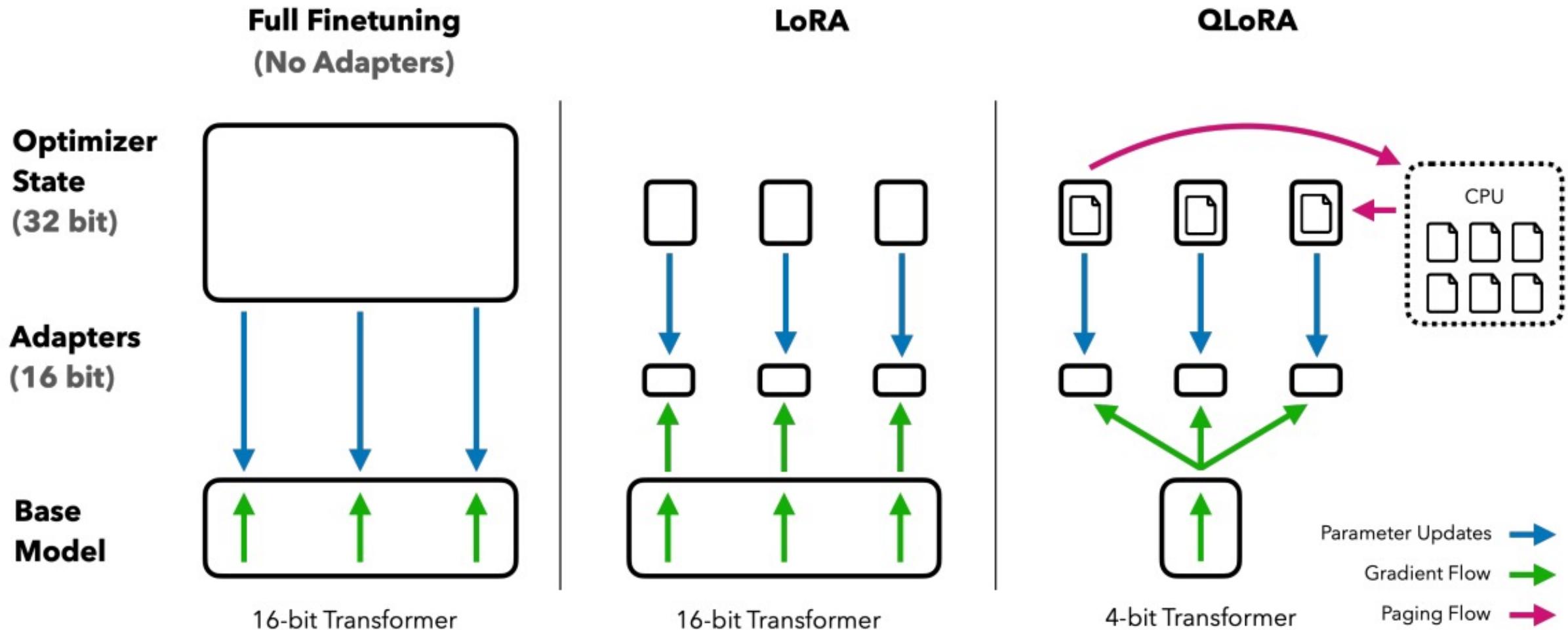
Step 3. Fine-tuning the LM with Reinforcement Learning



Llama-2-chat uses RLHF to ensure safety and helpfulness



QLoRA: Efficient Finetuning of Quantized LLMs



QLoRA: Efficient Finetuning of Quantized LLMs

QLoRA reduces the average memory requirements of finetuning a **65B** parameter model from >780GB of **GPU memory** to **<48GB**

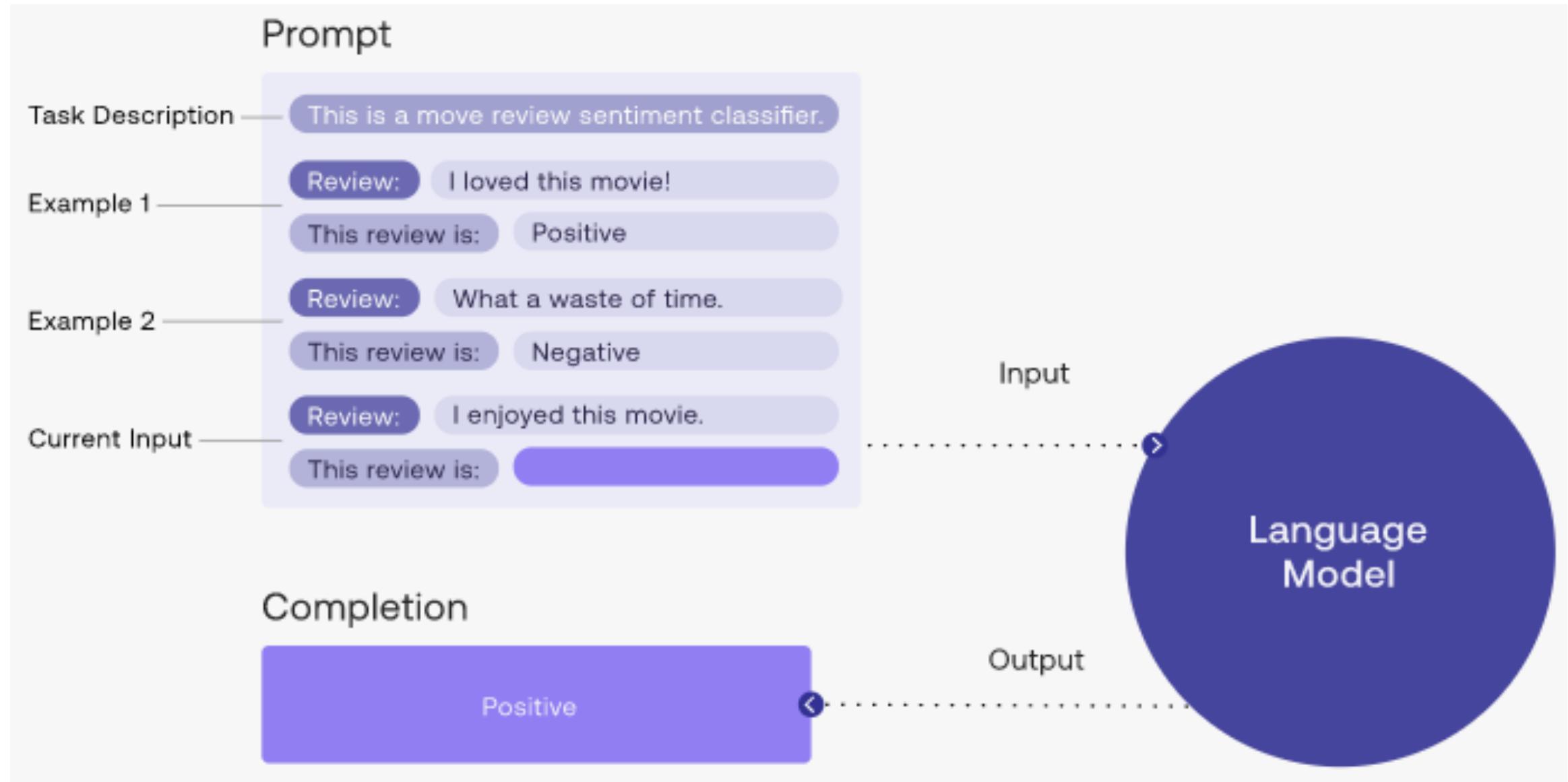
Model	Size	Elo
GPT-4	-	1348 ± 1
Guanaco 65B	41 GB	1022 ± 1
Guanaco 33B	21 GB	992 ± 1
Vicuna 13B	26 GB	974 ± 1
ChatGPT	-	966 ± 1
Guanaco 13B	10 GB	916 ± 1
Bard	-	902 ± 1
Guanaco 7B	6 GB	879 ± 1

QLoRA: Efficient Finetuning of Quantized LLMs

Model / Dataset	Params	Model bits	Memory	ChatGPT vs Sys	Sys vs ChatGPT	Mean	95% CI
GPT-4	-	-	-	119.4%	110.1%	114.5%	2.6%
Bard	-	-	-	93.2%	96.4%	94.8%	4.1%
Guanaco	65B	4-bit	41 GB	96.7%	101.9%	99.3%	4.4%
Alpaca	65B	4-bit	41 GB	63.0%	77.9%	70.7%	4.3%
FLAN v2	65B	4-bit	41 GB	37.0%	59.6%	48.4%	4.6%
Guanaco	33B	4-bit	21 GB	96.5%	99.2%	97.8%	4.4%
Open Assistant	33B	16-bit	66 GB	91.2%	98.7%	94.9%	4.5%
Alpaca	33B	4-bit	21 GB	67.2%	79.7%	73.6%	4.2%
FLAN v2	33B	4-bit	21 GB	26.3%	49.7%	38.0%	3.9%
Vicuna	13B	16-bit	26 GB	91.2%	98.7%	94.9%	4.5%
Guanaco	13B	4-bit	10 GB	87.3%	93.4%	90.4%	5.2%
Alpaca	13B	4-bit	10 GB	63.8%	76.7%	69.4%	4.2%
HH-RLHF	13B	4-bit	10 GB	55.5%	69.1%	62.5%	4.7%
Unnatural Instr.	13B	4-bit	10 GB	50.6%	69.8%	60.5%	4.2%
Chip2	13B	4-bit	10 GB	49.2%	69.3%	59.5%	4.7%
Longform	13B	4-bit	10 GB	44.9%	62.0%	53.6%	5.2%
Self-Instruct	13B	4-bit	10 GB	38.0%	60.5%	49.1%	4.6%
FLAN v2	13B	4-bit	10 GB	32.4%	61.2%	47.0%	3.6%
Guanaco	7B	4-bit	5 GB	84.1%	89.8%	87.0%	5.4%
Alpaca	7B	4-bit	5 GB	57.3%	71.2%	64.4%	5.0%
FLAN v2	7B	4-bit	5 GB	33.3%	56.1%	44.8%	4.0%

Source: Dettmers, Tim, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. "QLoRA: Efficient Finetuning of Quantized LLMs." arXiv preprint arXiv:2305.14314 (2023).

Prompt Engineering with ChatGPT for NLP



NLP with Transformers Github

Why GitHub? Team Enterprise Explore Marketplace Pricing

Search / Sign in Sign up

[nlp-with-transformers / notebooks](#) Public

Notifications Fork 170 Star 1.1k

Code Issues Pull requests Actions Projects Wiki Security Insights

main 1 branch 0 tags Go to file Code

lewtn Merge pull request #21 from JingchaoZhang/patch-3 ... ae5b7c1 15 days ago 71 commits

.github/ISSUE_TEMPLATE Update issue templates 25 days ago

data Move dataset to data directory 4 months ago

images Add README last month

scripts Update issue templates 25 days ago

.gitignore Initial commit 4 months ago

01_introduction.ipynb Remove Colab badges & fastdoc refs 27 days ago

02_classification.ipynb Merge pull request #8 from nlp-with-transformers/remove-display-df 26 days ago

03_transformer-anatomy.ipynb [Transformers Anatomy] Remove cells with figure references 22 days ago

04_multilingual-ner.ipynb Merge pull request #8 from nlp-with-transformers/remove-display-df 26 days ago

05_text-generation.ipynb Merge pull request #8 from nlp-with-transformers/remove-display-df 26 days ago

About

Jupyter notebooks for the Natural Language Processing with Transformers book

[transformersbook.com/](#)

Readme Apache-2.0 License 1.1k stars 33 watching 170 forks

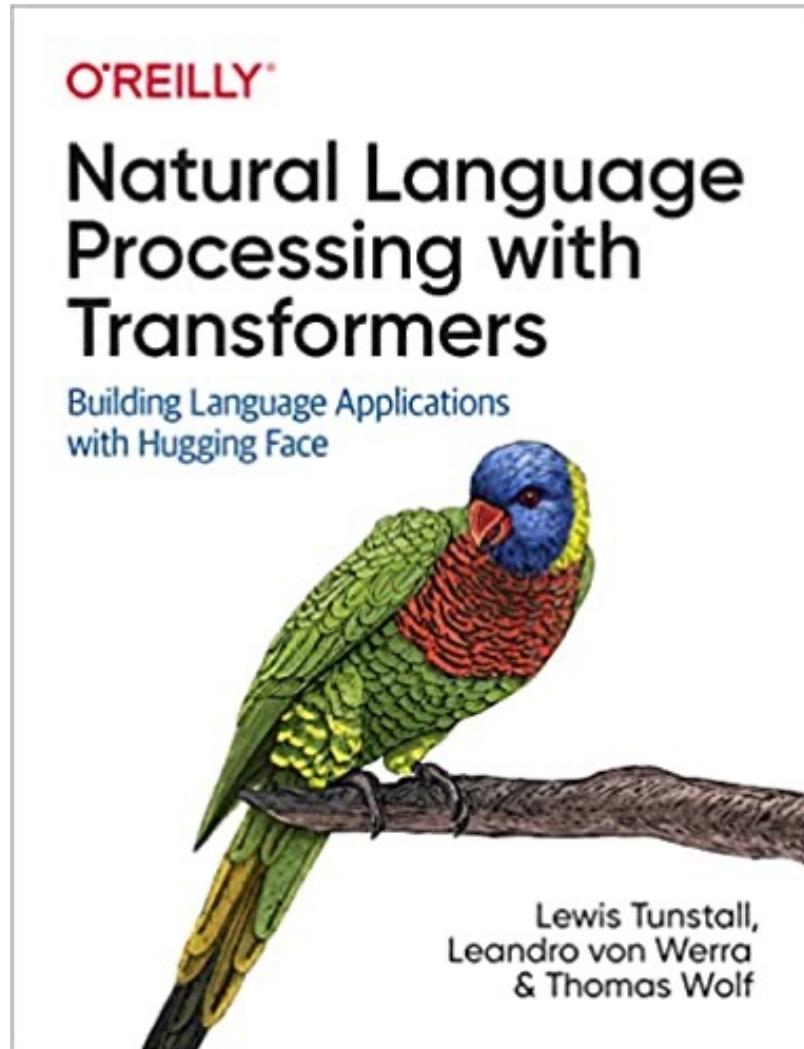
O'REILLY® Natural Language Processing with Transformers Building Language Applications with Hugging Face

Releases No releases published

Packages

<https://github.com/nlp-with-transformers/notebooks>

NLP with Transformers Github Notebooks



Running on a cloud platform

To run these notebooks on a cloud platform, just click on one of the badges in the table below:

Chapter	Colab	Kaggle	Gradient	Studio Lab
Introduction	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Text Classification	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Transformer Anatomy	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Multilingual Named Entity Recognition	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Text Generation	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Summarization	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Question Answering	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Making Transformers Efficient in Production	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Dealing with Few to No Labels	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Training Transformers from Scratch	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Future Directions	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab

Nowadays, the GPUs on Colab tend to be K80s (which have limited memory), so we recommend using [Kaggle](#), [Gradient](#), or [SageMaker Studio Lab](#). These platforms tend to provide more performant GPUs like P100s, all for free!

Python in Google Colab (Python101)

<https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT>

The screenshot shows a Google Colab notebook titled "python101.ipynb". The notebook interface includes a toolbar at the top with icons for file operations, a search bar, and user settings. Below the toolbar is a code editor with two tabs: "+ Code" and "+ Text". The main content area displays a section titled "Text Summarization" with a list of sources and a code cell demonstrating text summarization using the Hugging Face Transformers library.

• Source: Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022), Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O'Reilly Media.
• Github: <https://github.com/nlp-with-transformers/notebooks>

```
1 #Source: https://huggingface.co/tasks/summarization
2 !pip install transformers
3 from transformers import pipeline
4 classifier = pipeline("summarization")
5 text = "Paris is the capital and most populous city of France, with an estimated population of 2,175,601 residents as of 2018, in an area of more than
6 classifier(text, max_length=30)

No model was supplied, defaulted to sshleifer/distilbart-cnn-12-6 (https://huggingface.co/sshleifer/distilbart-cnn-12-6)
Your min_length=56 must be inferior than your max_length=30.
[{'summary_text': ' Paris is the capital and most populous city of France, with an estimated population of 2,175,601 residents . The City of Paris'}]
```

```
1 #!pip install transformers
2 text = """Dear Amazon, last week I ordered an Optimus Prime action figure \
3 from your online store in Germany. Unfortunately, when I opened the package, \
4 I discovered to my horror that I had been sent an action figure of Megatron \
5 instead! As a lifelong enemy of the Decepticons, I hope you can understand my \
6 dilemma. To resolve the issue, I demand an exchange of Megatron for the \
7 Optimus Prime figure I ordered. Enclosed are copies of my records concerning \
8 this purchase. I expect to hear from you soon. Sincerely, Bumblebee."""
9 from transformers import pipeline
10 summarizer = pipeline("summarization")
11 outputs = summarizer(text, max_length=45, clean_up_tokenization_spaces=True)
12 print(outputs[0]['summary_text'])
```

<https://tinyurl.com/aintpuppython101>

NLP with Transformers

```
!git clone https://github.com/nlp-with-transformers/notebooks.git  
%cd notebooks  
from install import *  
install_requirements()
```

```
from utils import *  
setup_chapter()
```

Text Classification

```
text = """Dear Amazon, last week I ordered an Optimus Prime action figure \
from your online store in Germany. Unfortunately, when I opened the package, \
I discovered to my horror that I had been sent an action figure of Megatron \
instead! As a lifelong enemy of the Decepticons, I hope you can understand my \
dilemma. To resolve the issue, I demand an exchange of Megatron for the \
Optimus Prime figure I ordered. Enclosed are copies of my records concerning \
this purchase. I expect to hear from you soon. Sincerely, Bumblebee."""
```

Text Classification

```
text = """Dear Amazon, last week I ordered an Optimus Prime action figure \
from your online store in Germany. Unfortunately, when I opened the package, \
I discovered to my horror that I had been sent an action figure of Megatron \
instead! As a lifelong enemy of the Decepticons, I hope you can understand my \
dilemma. To resolve the issue, I demand an exchange of Megatron for the \
Optimus Prime figure I ordered. Enclosed are copies of my records concerning \
this purchase. I expect to hear from you soon. Sincerely, Bumblebee."""
```

```
from transformers import pipeline
classifier = pipeline("text-classification")

import pandas as pd
outputs = classifier(text)
pd.DataFrame(outputs)
```

	label	score
0	NEGATIVE	0.901546

Text Classification

```
from transformers import pipeline
classifier = pipeline("text-classification")

import pandas as pd
outputs = classifier(text)
pd.DataFrame(outputs)
```

	label	score
0	NEGATIVE	0.901546

Named Entity Recognition

```
ner_tagger = pipeline("ner", aggregation_strategy="simple")
outputs = ner_tagger(text)
pd.DataFrame(outputs)
```

	entity_group	score	word	start	end
0	ORG	0.879010	Amazon	5	11
1	MISC	0.990859	Optimus Prime	36	49
2	LOC	0.999755	Germany	90	97
3	MISC	0.556570	Mega	208	212
4	PER	0.590256	##tron	212	216
5	ORG	0.669692	Decept	253	259
6	MISC	0.498349	##icons	259	264
7	MISC	0.775362	Megatron	350	358
8	MISC	0.987854	Optimus Prime	367	380
9	PER	0.812096	Bumblebee	502	511

Question Answering

```
reader = pipeline("question-answering")
question = "What does the customer want?"
outputs = reader(question=question, context=text)
pd.DataFrame([outputs])
```

	score	start	end	answer
0	0.631292	335	358	an exchange of Megatron

Summarization

```
summarizer = pipeline("summarization")
outputs = summarizer(text, max_length=45, clean_up_tokenization_spaces=True)
print(outputs[0]['summary_text'])
```

Bumblebee ordered an Optimus Prime action figure from your online store in Germany. Unfortunately, when I opened the package, I discovered to my horror that I had been sent an action figure of Megatron instead.

Text Summarization

```
text = """Dear Amazon, last week I ordered an Optimus Prime action figure \
from your online store in Germany. Unfortunately, when I opened the package, \
I discovered to my horror that I had been sent an action figure of Megatron \
instead! As a lifelong enemy of the Decepticons, I hope you can understand my \
dilemma. To resolve the issue, I demand an exchange of Megatron for the \
Optimus Prime figure I ordered. Enclosed are copies of my records concerning \
this purchase. I expect to hear from you soon. Sincerely, Bumblebee."""
```

```
from transformers import pipeline
summarizer = pipeline("summarization")
outputs = summarizer(text, max_length=45, clean_up_tokenization_spaces=True)
print(outputs[0]['summary_text'])
```

Bumblebee ordered an Optimus Prime action figure from your online store in Germany. Unfortunately, when I opened the package, I discovered to my horror that I had been sent an action figure of Megatron instead.

Translation

```
translator = pipeline("translation_en_to_de",
                     model="Helsinki-NLP/opus-mt-en-de")
outputs = translator(text, clean_up_tokenization_spaces=True, min_length=100)
print(outputs[0]['translation_text'])
```

Sehr geehrter Amazon, letzte Woche habe ich eine Optimus Prime Action Figur aus Ihrem Online-Shop in Deutschland bestellt. Leider, als ich das Paket öffnete, entdeckte ich zu meinem Entsetzen, dass ich stattdessen eine Action Figur von Megatron geschickt worden war! Als lebenslanger Feind der Decepticons, Ich hoffe, Sie können mein Dilemma verstehen. Um das Problem zu lösen, Ich fordere einen Austausch von Megatron für die Optimus Prime Figur habe ich bestellt. Anbei sind Kopien meiner Aufzeichnungen über diesen Kauf. Ich erwarte, bald von Ihnen zu hören. Aufrichtig, Bumblebee.

Text Generation

```
from transformers import set_seed  
set_seed(42) # Set the seed to get reproducible results  
  
generator = pipeline("text-generation")  
response = "Dear Bumblebee, I am sorry to hear that your order was mixed up."  
prompt = text + "\n\nCustomer service response:\n" + response  
outputs = generator(prompt, max_length=200)  
print(outputs[0]['generated_text'])
```

Customer service response:

Dear Bumblebee, I am sorry to hear that your order was mixed up. The order was completely mislabeled, which is very common in our online store, but I can appreciate it because it was my understanding from this site and our customer service of the previous day that your order was not made correct in our mind and that we are in a process of resolving this matter. We can assure you that your order

Text Generation

Dear Amazon, last week I ordered an Optimus Prime action figure from your online store in Germany. Unfortunately, when I opened the package, I discovered to my horror that I had been sent an action figure of Megatron instead! As a lifelong enemy of the Decepticons, I hope you can understand my dilemma. To resolve the issue, I demand an exchange of Megatron for the Optimus Prime figure I ordered. Enclosed are copies of my records concerning this purchase. I expect to hear from you soon. Sincerely, Bumblebee.

Customer service response:

Dear Bumblebee, I am sorry to hear that your order was mixed up. The order was completely mislabeled, which is very common in our online store, but I can appreciate it because it was my understanding from this site and our customer service of the previous day that your order was not made correct in our mind and that we are in a process of resolving this matter. We can assure you that your order

Question Answering

```
!pip install transformers
from transformers import pipeline
qamodel = pipeline("question-answering")
question = "Where do I live?"
context = "My name is Michael and I live in Taipei."
qamodel(question = question, context = context)
```

```
{'answer': 'Taipei', 'end': 39, 'score': 0.9730741381645203, 'start': 33}
```

Question Answering

```
from transformers import pipeline
qamodel = pipeline("question-answering", model ='deepset/roberta-base-squad2')
question = "Where do I live?"
context = "My name is Michael and I live in Taipei."
output = qamodel(question = question, context = context)
print(output['answer'])
```

Taipei

Text Generation with LLM (zephyr-7b-beta)

```
# Install transformers from source - only needed for versions <= v4.34
!pip install git+https://github.com/huggingface/transformers.git
!pip install accelerate
```

Text Generation with LLM (zephyr-7b-beta)

```
import torch
from transformers import pipeline

pipe = pipeline("text-generation",
model="HuggingFaceH4/zephyr-7b-beta",
torch_dtype=torch.bfloat16,
device_map="auto")
```

Text Generation with LLM (zephyr-7b-beta)

```
# We use the tokenizer's chat template to format each message - see
# https://huggingface.co/docs/transformers/main/en/chat_templating
messages = [
    {
        "role": "system",
        "content": "You are a friendly chatbot who always responds in the style of a pirate",
    },
    {"role": "user", "content": "How many helicopters can a human eat in one sitting?"},
]
prompt = pipe.tokenizer.apply_chat_template(messages,
tokenize=False, add_generation_prompt=True)

outputs = pipe(prompt, max_new_tokens=256,
do_sample=True, temperature=0.7, top_k=50, top_p=0.95)
print(outputs[0]["generated_text"])
```

Text Generation with LLM (zephyr-7b-beta)

```
import torch
from transformers import pipeline

pipe = pipeline("text-generation", model="HuggingFaceH4/zephyr-7b-beta",
torch_dtype=torch.bfloat16, device_map="auto")

# We use the tokenizer's chat template to format each message - see
# https://huggingface.co/docs/transformers/main/en/chat_templating
messages = [
    {
        "role": "system",
        "content": "You are a friendly chatbot who always responds in the style of a pirate",
    },
    {"role": "user", "content": "How many helicopters can a human eat in one sitting?"},
]

prompt = pipe.tokenizer.apply_chat_template(messages, tokenize=False,
add_generation_prompt=True)
outputs = pipe(prompt, max_new_tokens=256, do_sample=True, temperature=0.7,
top_k=50, top_p=0.95)
print(outputs[0]["generated_text"])
```

Summary

- **Text Generation**
- **Large Language Models (LLMs)**
- **Prompt Engineering**
- **Fine-tuning**
- **Retrieval Augmented Generation (RAG)**

References

- Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022), Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O'Reilly Media.
- Denis Rothman (2021), Transformers for Natural Language Processing: Build innovative deep neural network architectures for NLP with Python, PyTorch, TensorFlow, BERT, RoBERTa, and more, Packt Publishing.
- Savaş Yıldırım and Meysam Asgari-Chenaglu (2021), Mastering Transformers: Build state-of-the-art models from scratch with advanced natural language processing techniques, Packt Publishing.
- Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C. D., & Finn, C. (2023). Direct preference optimization: Your language model is secretly a reward model. arXiv preprint arXiv:2305.18290.
- Tunstall, Lewis, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang et al. "Zephyr: Direct Distillation of LM Alignment." arXiv preprint arXiv:2310.16944 (2023).
- Yihan Cao, Siyu Li, Yixin Liu, Zhiling Yan, Yutong Dai, Philip S. Yu, and Lichao Sun (2023). "A Comprehensive Survey of AI-Generated Content (AIGC): A History of Generative AI from GAN to ChatGPT." arXiv preprint arXiv:2303.04226.
- Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. (2023) "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing." ACM Computing Surveys 55, no. 9 (2023): 1-35.
- Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min et al. (2023) "A Survey of Large Language Models." arXiv preprint arXiv:2303.18223.
- Touvron, Hugo, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov et al. (2023) "Llama 2: Open Foundation and Fine-Tuned Chat Models." arXiv preprint arXiv:2307.09288 (2023).
- Junliang Wang, Chuqiao Xu, Jie Zhang, and Ray Zhong (2022). "Big data analytics for intelligent manufacturing systems: A review." Journal of Manufacturing Systems 62 (2022): 738-752.
- Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., ... & Lowe, R. (2022). Training language models to follow instructions with human feedback. arXiv preprint arXiv:2203.02155.
- Gozalo-Brizuela, Roberto, and Eduardo C. Garrido-Merchan (2023). "ChatGPT is not all you need. A State of the Art Review of large Generative AI models." arXiv preprint arXiv:2301.04655 (2023).
- Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. (2023) "InstructBLIP: Towards General-purpose Vision-Language Models with Instruction Tuning." arXiv preprint arXiv:2305.06500 (2023).
- Shahab Saquib Sohail, Faiza Farhat, Yassine Himeur, Mohammad Nadeem, Dag Øivind Madsen, Yashbir Singh, Shadi Atalla, and Wathiq Mansoor (2023). "The Future of GPT: A Taxonomy of Existing ChatGPT Research, Current Challenges, and Possible Future Directions." Current Challenges, and Possible Future Directions (April 8, 2023) (2023).
- Longbing Cao (2022). "Decentralized ai: Edge intelligence and smart blockchain, metaverse, web3, and desci." IEEE Intelligent Systems 37, no. 3: 6-19.
- Qinglin Yang, Yetong Zhao, Huawei Huang, Zehui Xiong, Jiawen Kang, and Zibin Zheng (2022). "Fusing blockchain and AI with metaverse: A survey." IEEE Open Journal of the Computer Society 3 : 122-136.
- Russell Belk, Mariam Humayun, and Myriam Brouard (2022). "Money, possessions, and ownership in the Metaverse: NFTs, cryptocurrencies, Web3 and Wild Markets." Journal of Business Research 153: 198-205.
- Thien Huynh-The, Quoc-Viet Pham, Xuan-Qui Pham, Thanh Thi Nguyen, Zhu Han, and Dong-Seong Kim (2022). "Artificial Intelligence for the Metaverse: A Survey." arXiv preprint arXiv:2202.10336.
- Thippa Reddy Gadekallu, Thien Huynh-The, Weizheng Wang, Gokul Yenduri, Pasika Ranaweera, Quoc-Viet Pham, Daniel Benevides da Costa, and Madhusanka Liyanage (2022). "Blockchain for the Metaverse: A Review." arXiv preprint arXiv:2203.09738.
- Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., ... & Lowe, R. (2022). Training language models to follow instructions with human feedback. arXiv preprint arXiv:2203.02155.
- Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." arXiv preprint arXiv:1810.04805.
- OpenAI (2023), A Survey of Techniques for Maximizing LLM Performance, <https://www.youtube.com/watch?v=ahnGLM-RC1Y>
- The Super Duper NLP Repo, <https://notebooks.quantumstat.com/>
- NLP with Transformer, <https://github.com/nlp-with-transformers/notebooks>
- Min-Yuh Day (2023), Python 101, <https://tinyurl.com/aintpupython101>