Instruction Tuning

AI with Deep Learning (EE4016)

Lai-Man Po

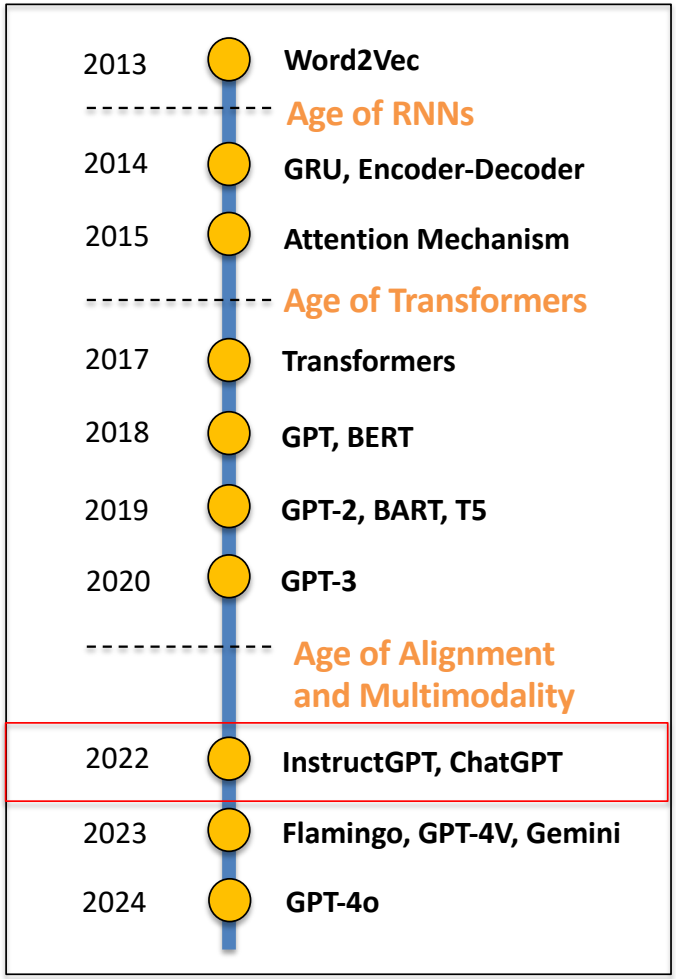
Department of Electrical Engineering

City University of Hong Kong

https://medium.com/@lmpo/an-overview-instruction-tuning-for-llms-440228e7edab

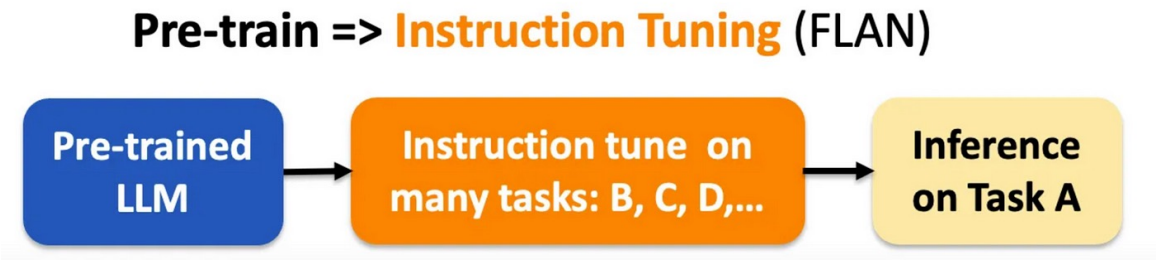
From Word2Vec to GPT (10 Years of Deep NLP)

1. Tokenization and Word2Vec
2. RNNs, LSTM, GRU, Encoder-Decoder and Attention Mechanism
3. Self-Attention and Transformers
4. LLMs: BERT, GPT, BART, T5
5. **Alignment: InstructGPT, ChatGPT**
6. Multimodality: GPT-4V, LLaVA, Gemini



Content

- Large Language Models (LLMs) Training Approaches
- Fine-Tuning vs Instruction Tuning
- Brief History of Instruction Tuning: T5, FLAN and T0
- Instruction Dataset Constructions
- Instruction Fine-tuned LLMs
- Domain-specific Instruction Fine-tuning



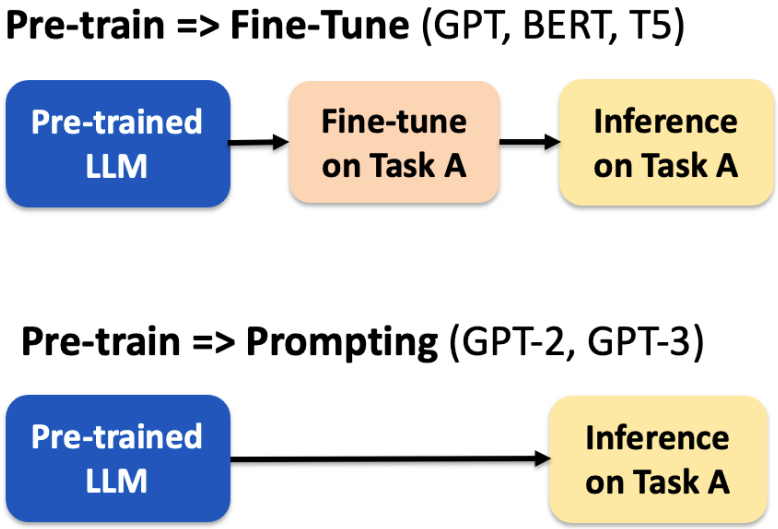
Instruction Tuning for Large Language Models: A Survey https://arxiv.org/abs/2308.10792

OpenAI's GPT LLM Series

- **GPT** (2018-06): 117M parameters
  - **Pioneered decoder-only transformer architecture** and improved language understanding through text-completion objective pre-training (self-supervised learning)
- **GPT-2** (2019-02): 1.5B parameters
  - Demonstrated language models as unsupervised multitask learners and showcased **zero-shot learning capabilities**
- **GPT-3** (2020-05): 175B parameters.
  - **Introduced few-shot learning with in-context learning** and exhibited remarkable versatility in performing various tasks without task-specific training
- **ChatGPT/GPT-3.5** (2022-11)
  - **Optimized for conversational AI and Preference Alignment using RLHF**
- **GPT-4V** (2023-03)
  - **Enhanced multimodal capabilities** and context understanding
- **GPT-4o** (2024-05)
  - A more efficient and optimized version of GPT-4
- **GPT-o1-preview** (2024-09)
  - Designed to **enhance reasoning capabilities**, particularly in complex tasks such as math, science, and coding

The Dawn of LLMs (2018–2020)

- The introduction of OpenAI's GPT in 2018 marked a new era in natural language processing.
  - This transformer-based model used a decoder-only architecture and was trained with text completion objectives.
  - Later models, such as BERT, BART, and T5, adopted a similar **pre-training and fine-tuning approach**.
  - The release of GPT-2 in 2019 pioneered **zero-shot prompting**, while GPT-3 in 2020 demonstrated the capabilities of large language models with 175 billion parameters, enabling **few-shot prompting**.



Limitations of Pre-Training and Fine-Tuning

Pre-train => Fine-Tune (GPT, BERT, T5)

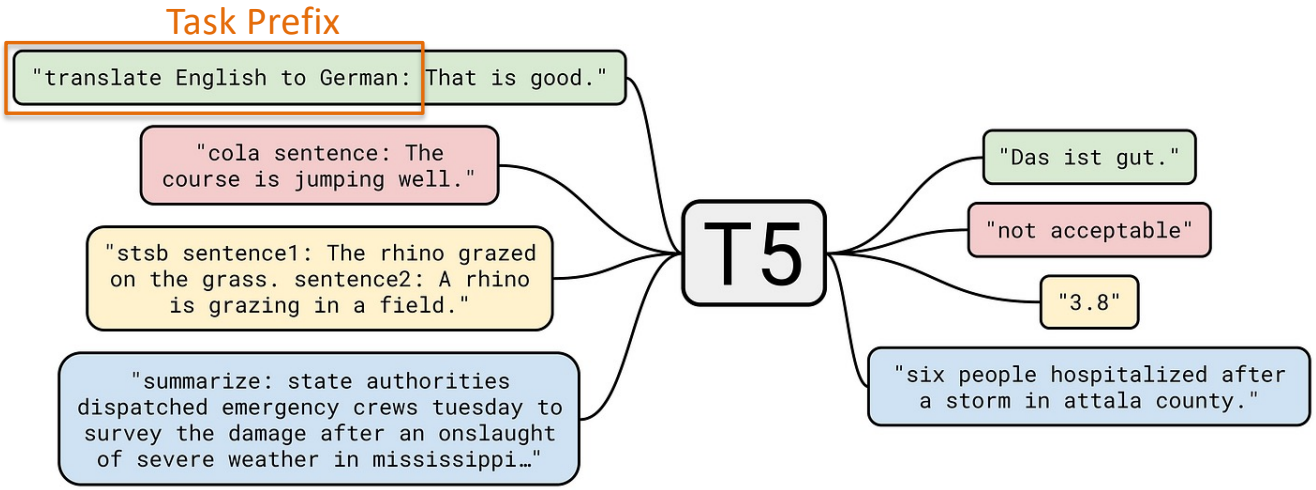
Domain Adaptation

- Typically requires many task-specific examples
- One specialized model for each task

- Models like GPT and BERT have limitations due to their pre-training and fine-tuning approaches.
- **Text completion objectives** may not provide sufficient context understanding, resulting in superficial comprehension.
- **The lack of detailed task descriptions** can **limit the model's ability to generalize**, making it less adaptable to new tasks.

T5 Multitask Learning and Its Limitations

- The T5 model's multitask learning approach, introduced in 2019, was a step forward, but still had limitations.
- By using a **task-specific prefix**, **it improved performance on training tasks**, but failed to enable effective **cross-task generalization**, leaving models unable to adapt to new, unseen tasks.

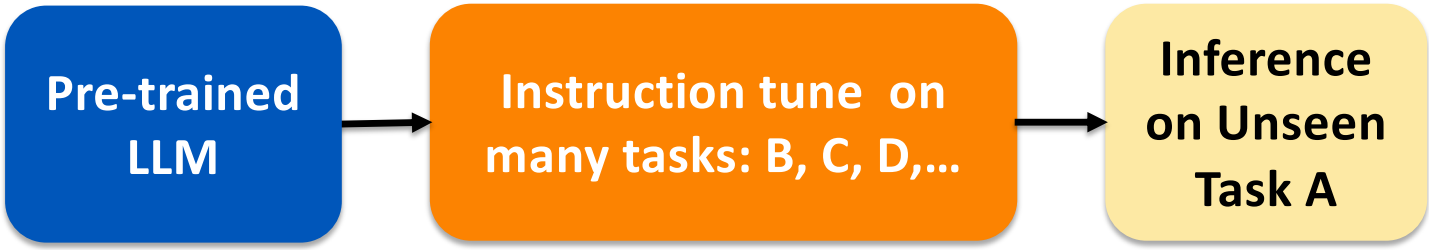


The T5 data format consists of a text-to-text format, where the input is a sequence of text with a task-specific prefix, and the output is a generated sequence of text.

Instruction Tuning (Mid-2021)

- In 2021, **Instruction Tuning** emerged as a new approach to overcome traditional pre-training and fine-tuning limitations.
- **Instruction Tuning** involves **providing detailed, natural language instructions** during fine-tuning, enabling models to learn nuanced task context and relationships, and **generalize better to new tasks and contexts.**

Pre-train => Instruction Tuning (FLAN)



Instruction tuning is a process that refines LLMs to better understand and follow user instructions, enhancing their usefulness and responsiveness, and cross-task generalization capabilities.

Key Concepts of Instruction Tuning

1. **Detailed Instructions:** Providing comprehensive, natural language descriptions of tasks to help models understand and produce accurate responses. Example of Instruction-following data pair:
   - **Instruction:** "Summarize the following text in 20 words or less: 'The new restaurant in town serves a variety of Italian dishes, including pasta, pizza, and risotto.'"
   - **Output:** "New Italian restaurant serves pasta, pizza, and risotto in a variety of dishes."

2. **Generalization:** Training on various tasks with detailed instructions to enable models to generalize their understanding to new, unseen tasks.

Domain Adaptation and Generalization Approaches

Pre-train => Fine-Tune (GPT, BERT, T5)

Domain Adaptation

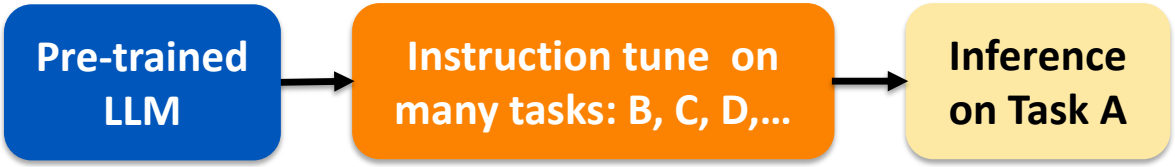
- Typically requires many task-specific examples
- One specialized model for each task

Pre-train => Prompting (GPT-2,GPT-3)

No Adaptation of the Model

- Just improve performance via few-shot prompting or prompt engineering

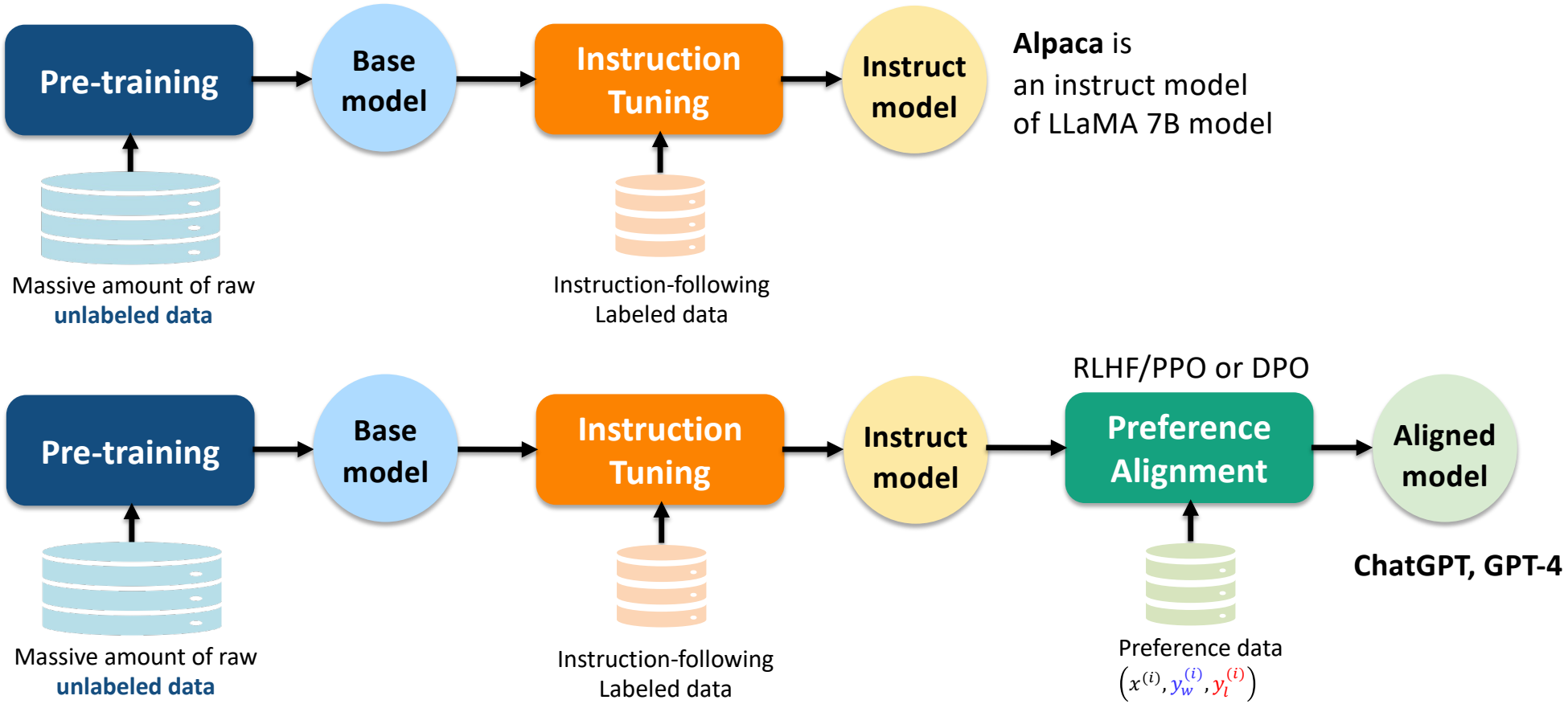
Pre-train => Instruction Tuning (FLAN)



Cross-Task Generalization

- **Model learns to perform many tasks via natural language instruction**
- Inference on unseen task

Instruction Tuning in Modern LLM Training Pipelines

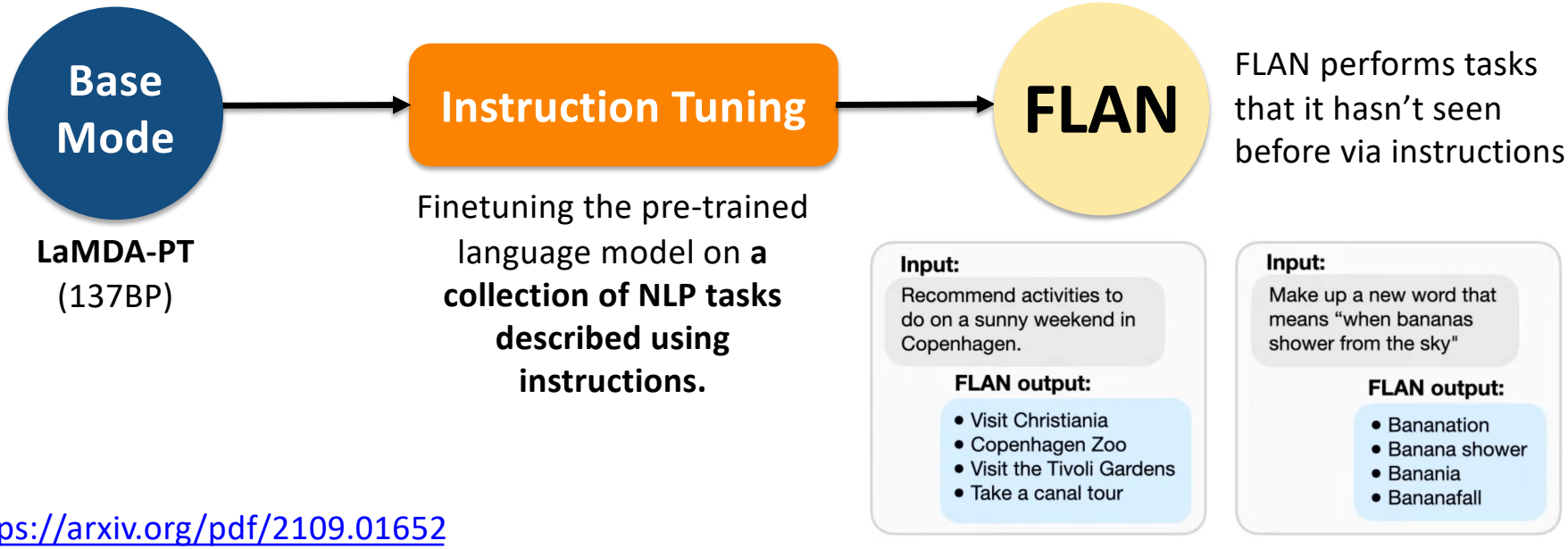


Instruction Tuning Emerges (2021)

- The concept of Instruction Tuning took shape as researchers fine-tuned LLMs on diverse (instruction, output) pairs. Example;
  - **Instruction:** "Summarize the following text in 20 words or less: 'The new restaurant in town serves a variety of Italian dishes, including pasta, pizza, and risotto.'"
  - **Output:** "New Italian restaurant serves pasta, pizza, and risotto in a variety of dishes."
- **Two papers** helped to establish instruction tuning as a distinct area of research.
  - **FLAN**: “A Framework for Natural Language Instruction Tuning" (2021-09)
  - **T0**: “Multitask Prompted Training Enables Zero-Shot Task Generalization” (2021-10)

FLAN: Fine-tuned LAnguage Net (2021–09)

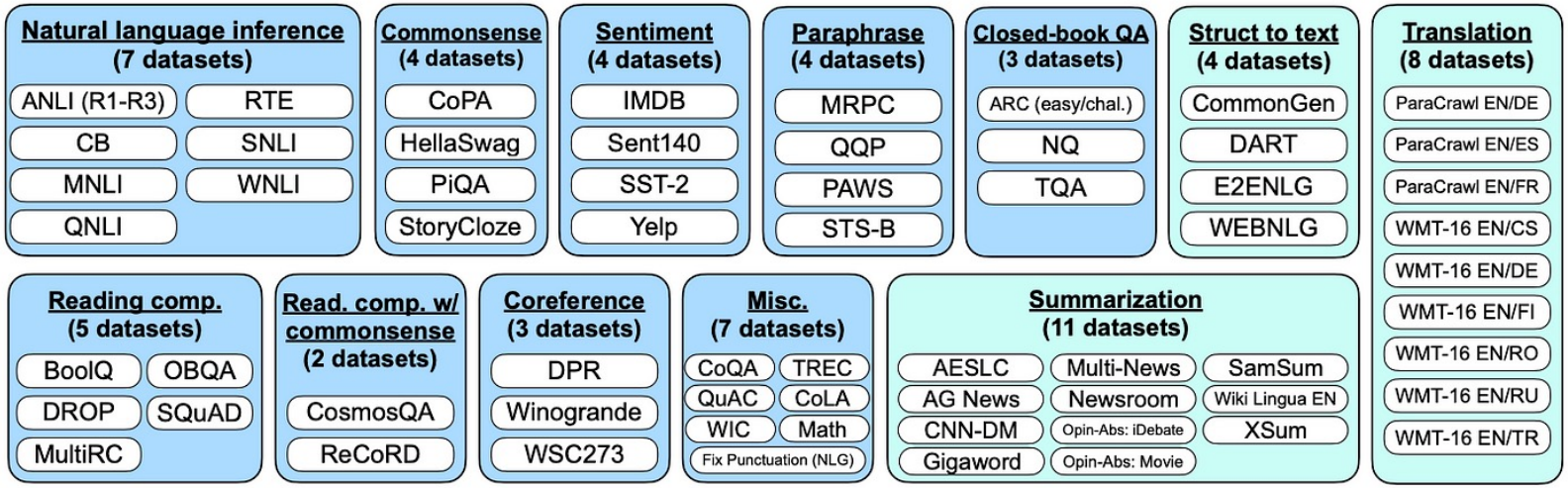
- **FLAN** marks a breakthrough in NLP by **leveraging instruction-following data with detailed instructions**, significantly enhancing the performance and versatility of large language models.



https://arxiv.org/pdf/2109.01652

FLAN Dataset

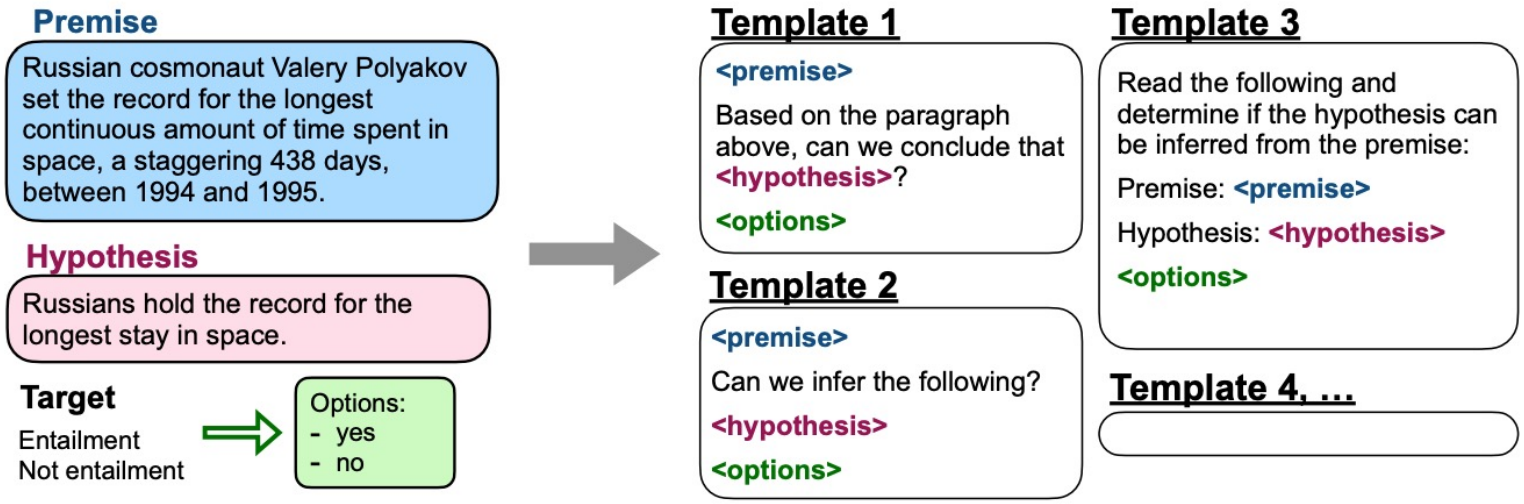
- **FLAN is trained on a diverse range of tasks and datasets**, including 62 NLP datasets grouped into 12 task clusters, covering both Natural Language Understanding and Generation tasks.
- This diverse training data enables FLAN to generalize well to new and unseen tasks, improving its robustness and adaptability.



Datasets and task clusters used in the FLAN paper (NLU tasks in blue; NLG tasks in teal).

FLAN Multiple Instruction Templates

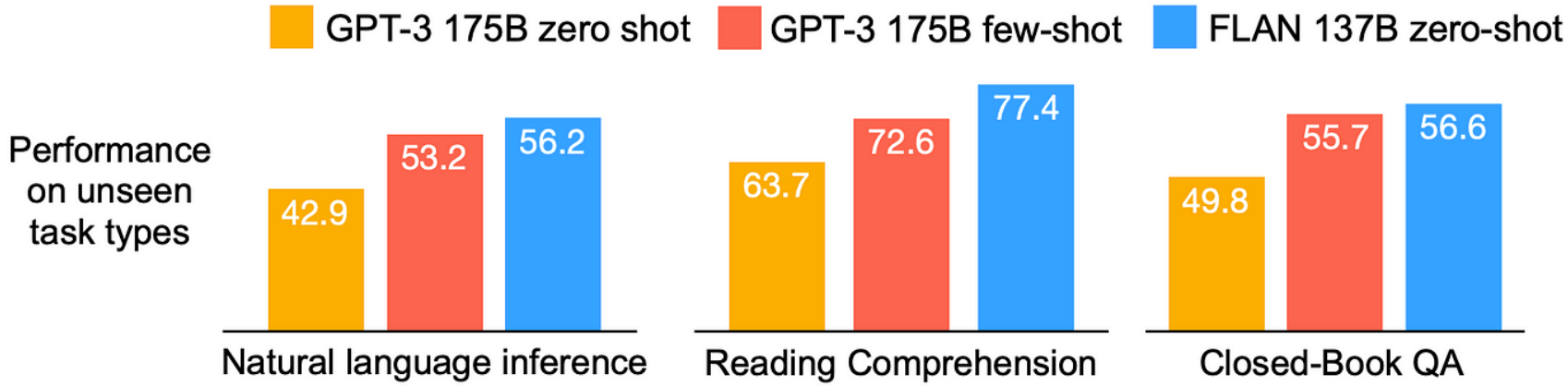
- For each dataset, **10 natural language instruction templates are created**, such as "Given the premise, can we conclude that [hypothesis]?" to provide the model with varied and clear instructions.



Multiple instruction templates describing a natural language inference task.

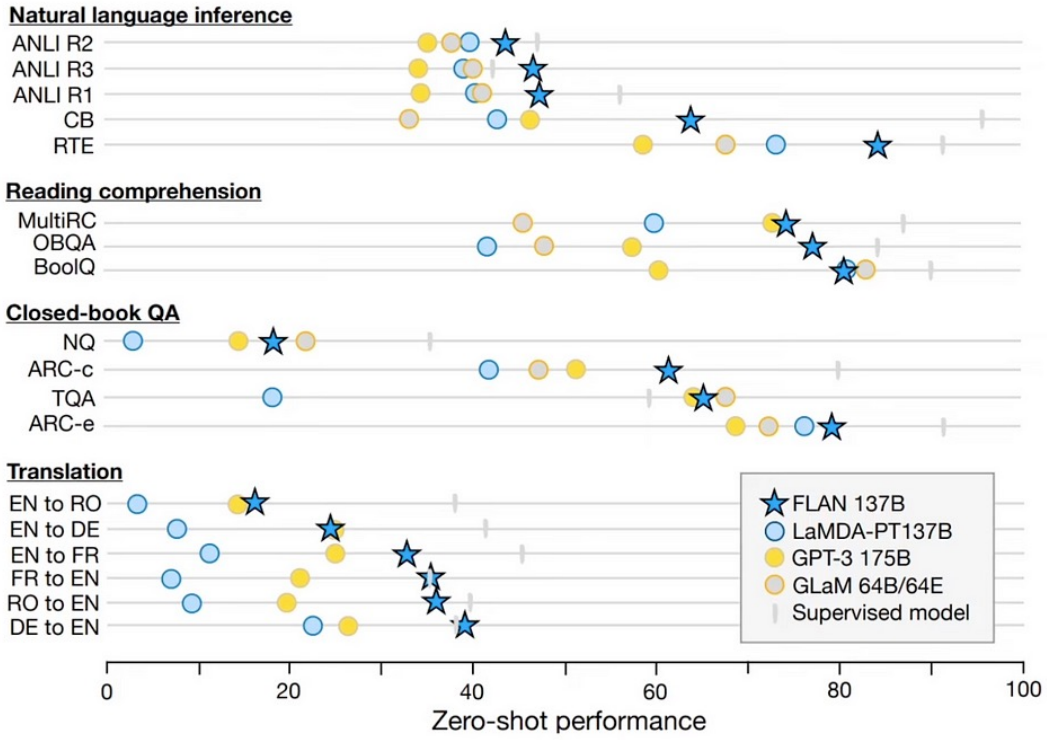
FLAN: Improved Generalization

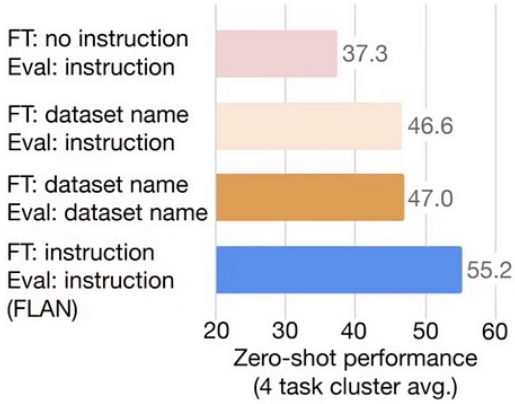
- **FLAN fine-tunes a 137 billion parameter LaMDA-PT model on an instruction-following dataset**, enabling it to extend its capabilities to new tasks with suitable instructions.
- This results in improved zero-shot learning performance and generalization capabilities, allowing the model to perform well on unseen tasks and understand new instructions effectively.



FLAN: Enhanced Performance

- **Zero-shot performance of FLAN compared to LaMDA-PT 137B, GPT-3 175B, and GLaM 64B/64E** on natural language inference, reading comprehension, closed-book QA, and translation. Performance of FLAN is the mean of up to 10 instructional templates per task. Supervised models were either T5, BERT, or translation models.



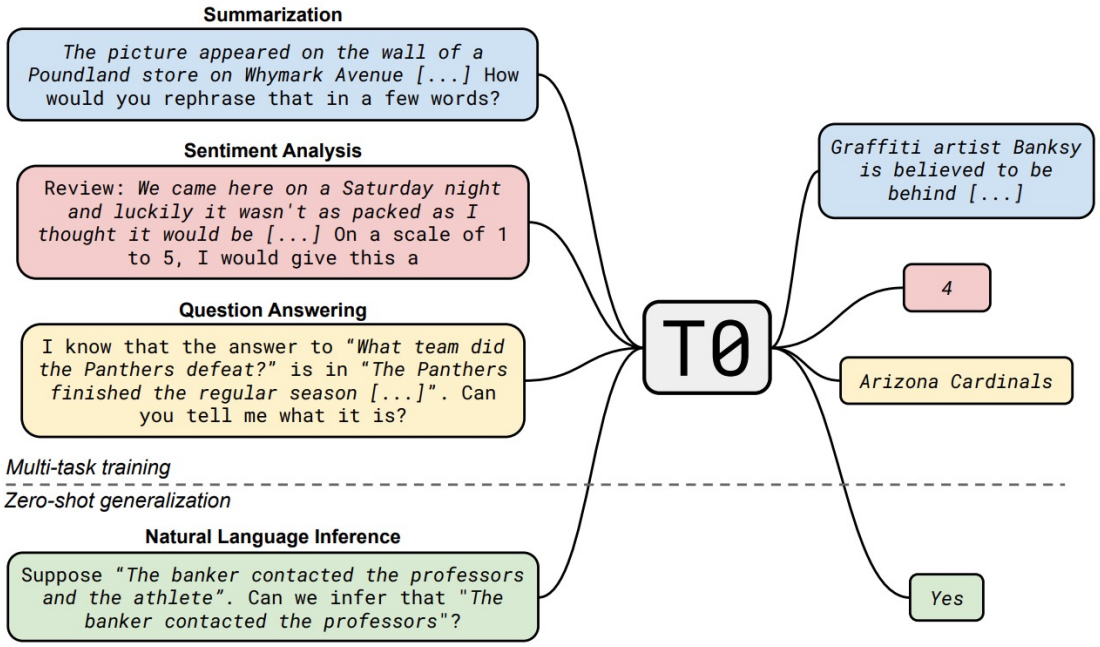


Impact of FLAN

- **Google's FLAN represents a significant step forward in the field of instruction tuning and natural language processing.**
- By training models to follow natural language instructions and generalize to new tasks, FLAN has demonstrated improved performance and versatility.
- Its impact on the development of more intuitive and adaptable language models is substantial, paving the way for future advancements in the field.

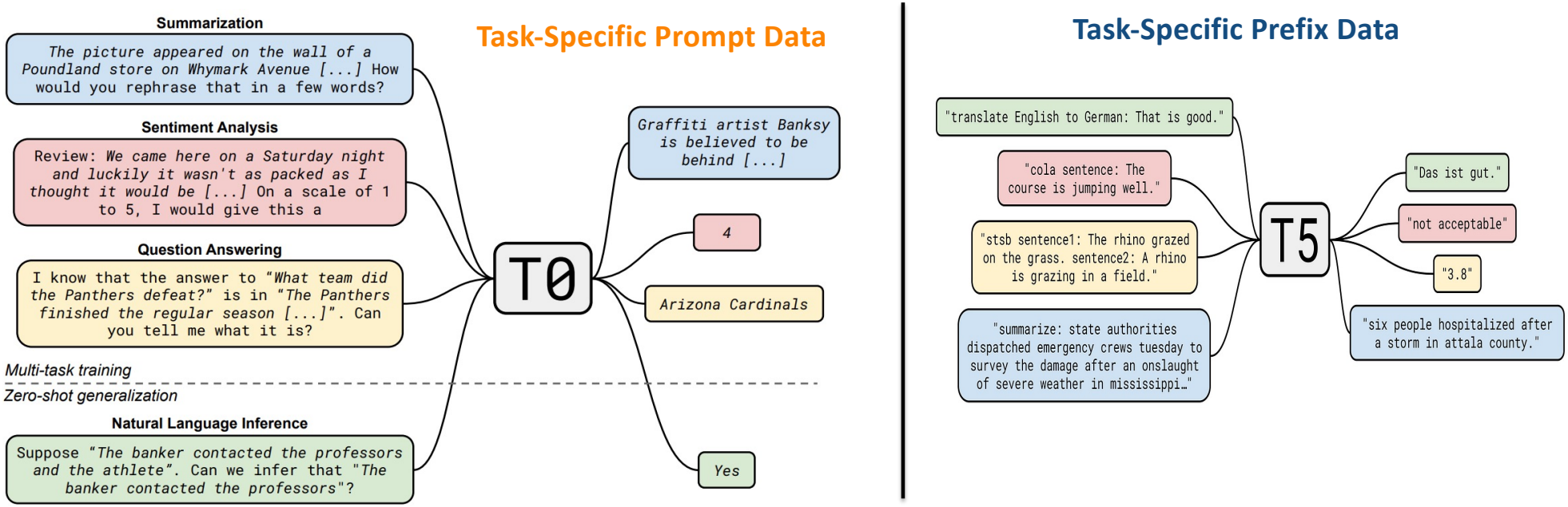
T0: Task-Specific Prompts (2021–10)

- T0 is a follow-up to the T5 model, which reframed NLP tasks as text-to-text problems. To overcome the T5 limitation, T0 was designed to enhance zero-shot learning through **task-specific prompts** during training.



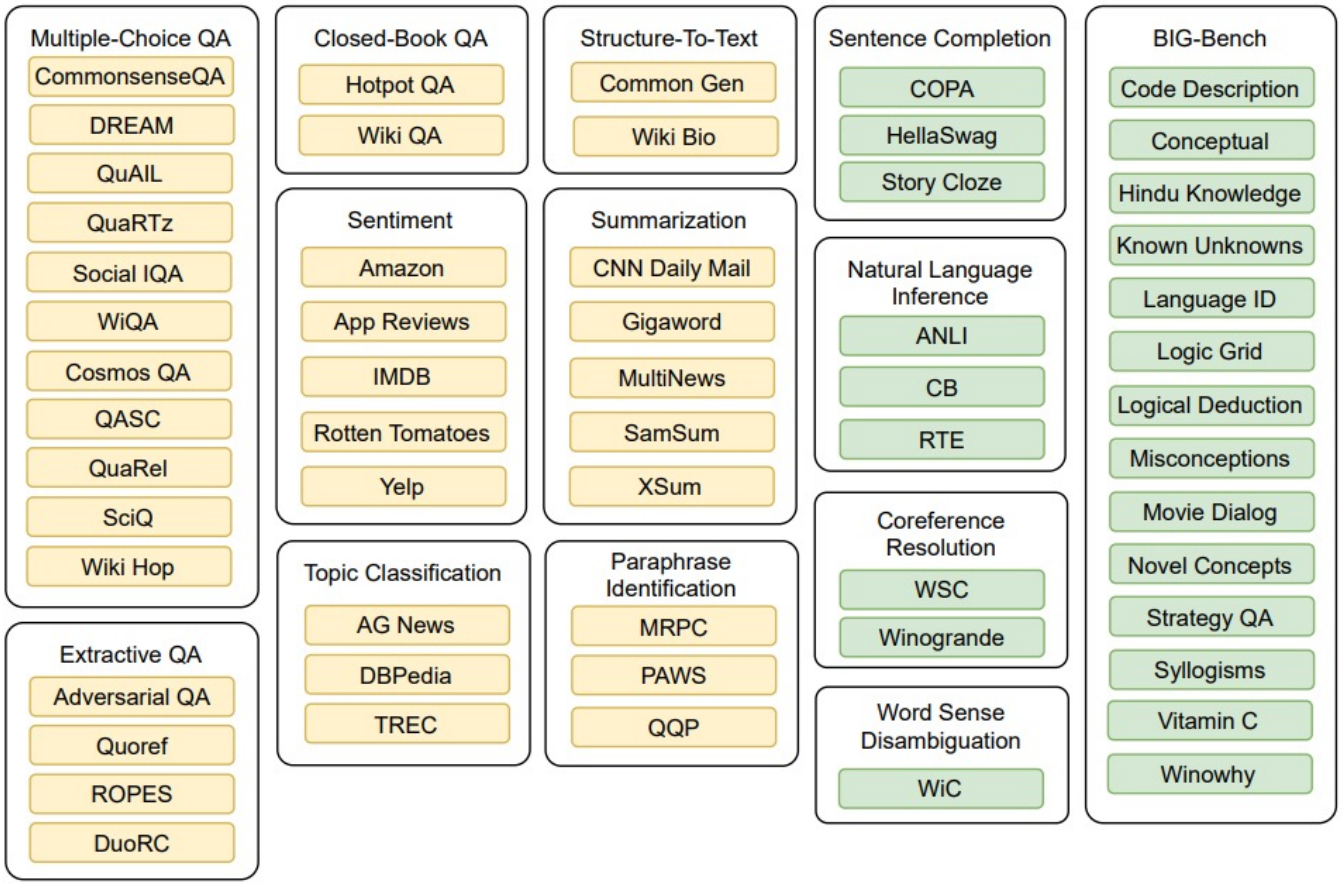
T0 vs T5 Data

- **T0 introducing task-specific prompts** during training, enabling zero-shot learning and generalization across a wide range of tasks.
- By exposing the model to diverse tasks with unique prompts, T0 can internalize context and requirements, allowing it to excel on unseen tasks.



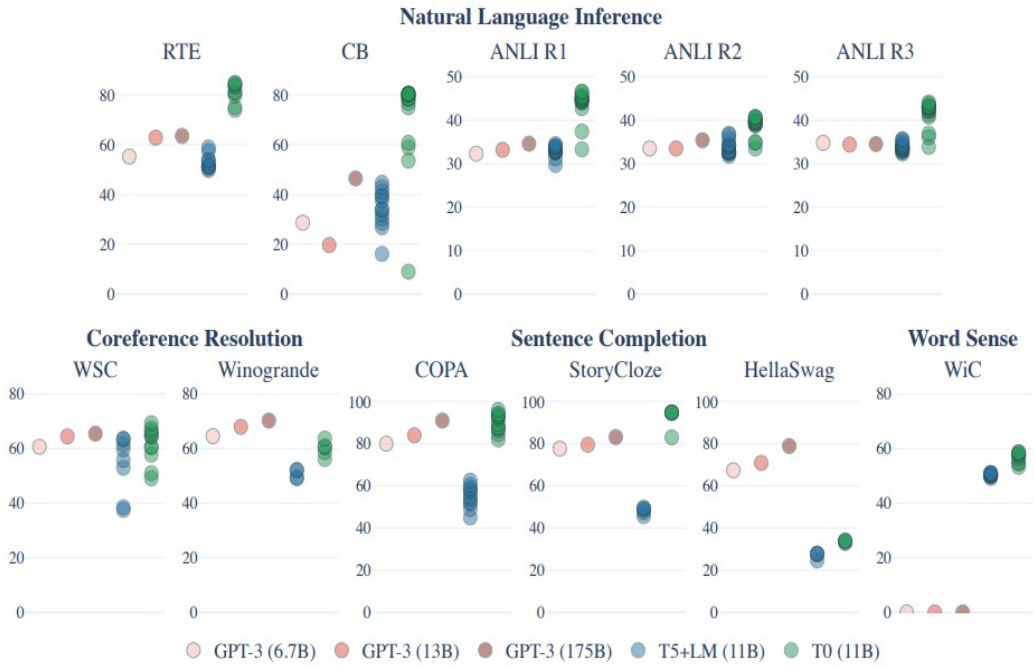
T0 Dataset

- The researchers evaluated their approach by pre-training the model on a standard T5 architecture, then fine-tuning it on a wide range of tasks (yellow in the diagram).
- They tested the model on unseen tasks (green in the diagram) and carefully held out specific task families to prevent overfitting.



T0 Results

- T0 is evaluated on 12 tasks, including text generation, question answering, and language translation
- The model shows strong zero-shot performance on several standard datasets, often outperforming models up to 16x its size
- T0 is compared to GPT-3 and other baseline models on held-out tasks
- Results show that T0 matches or exceeds the performance of all GPT-3 models on 9 out of 11 held-out datasets
- T0 also outperforms GPT-3 on all NLI datasets despite not being trained on natural language inference



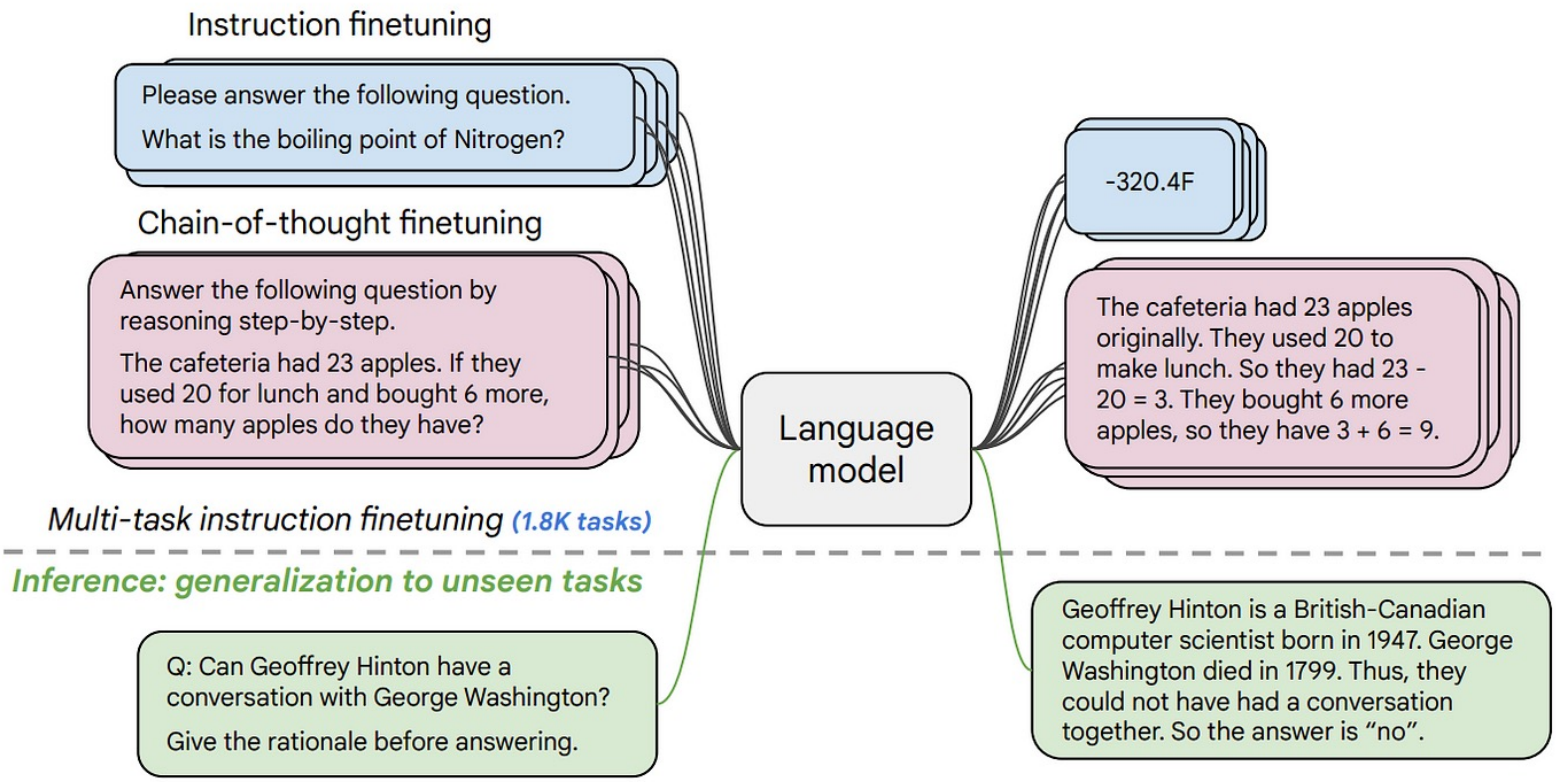
T0 Performance

- The T0 model is noteworthy because it demonstrates that it is possible to achieve comparable generalization performance with a smaller LLM, rivalling that of models with hundreds of billions of parameters.
- Examples of T0's applications, such as cooking recommendations and answering world knowledge questions, are presented, and further research on zero-shot learning and novel applications is eagerly anticipated.

Flan-PaLM: Scaling Instruction Tuning (2022–10)

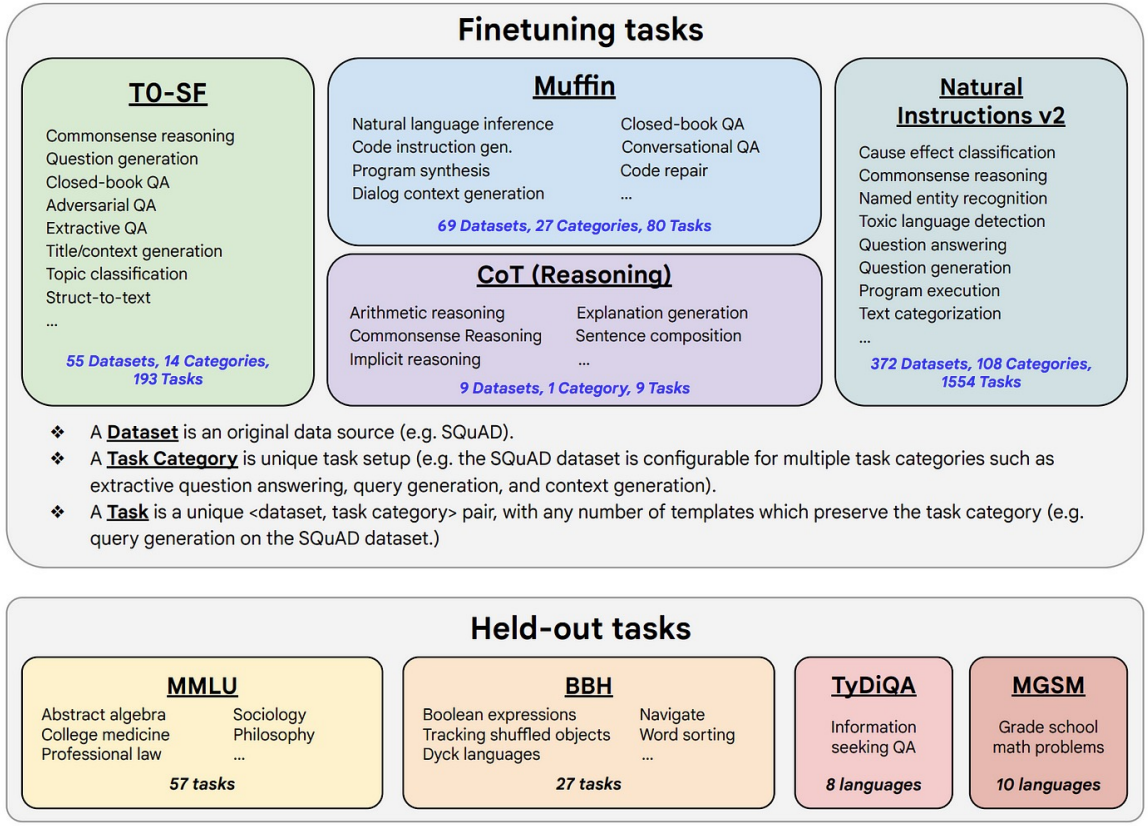
- Researchers at Google built upon the successes of FLAN and T0 to **investigate the effects of scaling instruction fine-tuning in LLMs**. Their study, Flan-PaLM, explored three key factors:
  1. **Scaling the number of tasks**: How does increasing the variety of tasks impact the model's performance?
  2. **Scaling the model size**: What benefits or drawbacks arise from using larger models?
  3. **Fine-tuning on Chain-of-Thought (CoT) data**: Can fine-tuning on CoT data improve the model's reasoning and arithmetic abilities?

Chain-of-Thought (CoT) data



Flan-PaLM Dataset

- This study utilized a comprehensive fine-tuning dataset consisting of **473 datasets, 146 task categories and 1,836 total tasks**. The dataset combines four mixtures from prior work Muffin, T0-SF, NIV2 and CoT. For each task in Muffin, T0-SF, and NIV2, instructional templates were used.
- The researchers manually created approximately ten instruction templates for each of the nine datasets in the CoT mixture. To create few-shot templates, various exemplar delimiters were written and applied randomly at the example level.

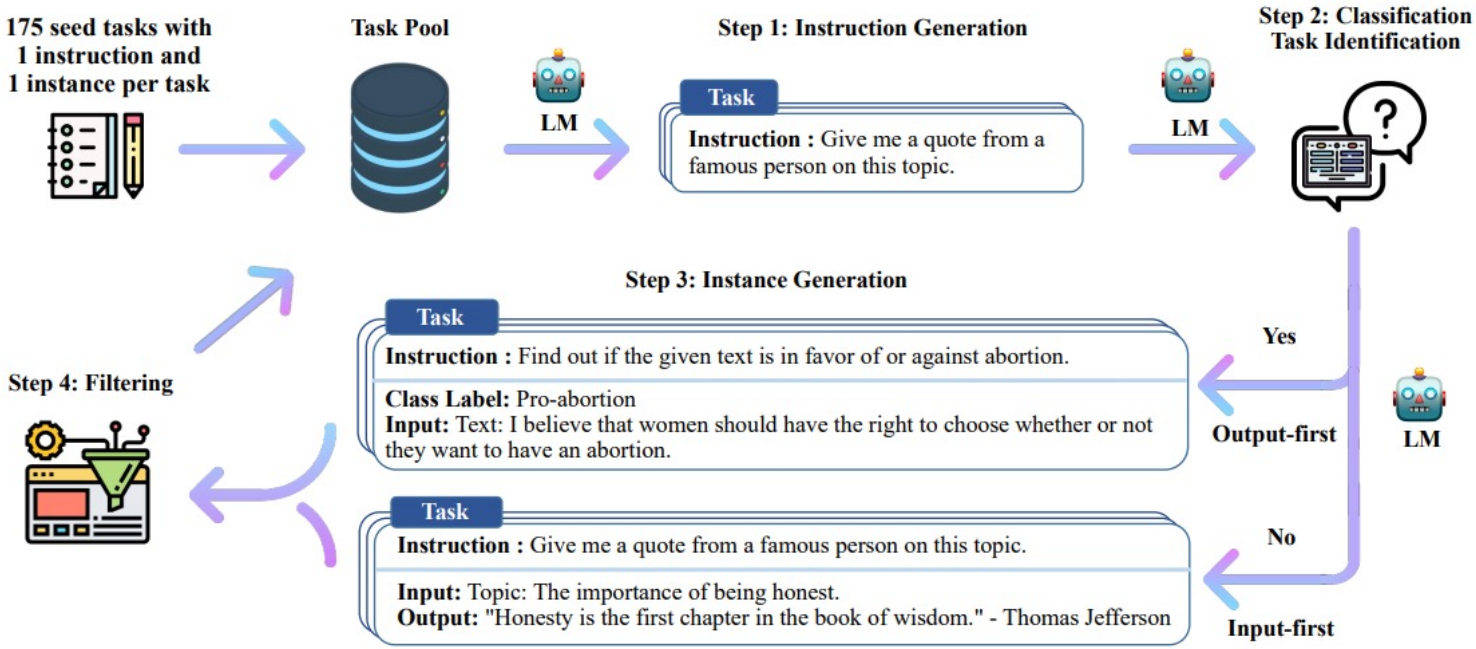


Flan-PaLM: Key Findings

1. **Instruction Tuning Results**: Instruction Tuning on PaLM models (8B, 62B, and 540B) showed significant improvements over no fine-tuning, with performance gains ranging from 9.4% to 15.5%.
2. **CoT + Instruction Tuning**: Including CoTt annotations in the fine-tuning mixture enhanced reasoning abilities. Flan-PaLM outperformed the original PaLM on four held-out evaluation benchmarks.
3. **Zero-Shot Performance**: Flan-PaLM models demonstrated improved performance on the BBH benchmark, which includes 23 unseen challenging BIG-Bench tasks. This improvement was achieved by leveraging CoT reasoning, activated by the phrase “let’s think step-by-step”.
4. **Comparison with Other Model**s: Instruction Tuning improved normalized average performance significantly across all model types tested.
5. **Zero-Shot Prompting**: In zero-shot settings, Flan-PaLM showed superior performance compared to the original PaLM, which often struggled with repetitions and failed to properly respond to instructions.

Self-Instruct (2022–12)

- The Self-Instruct paper investigated whether a LLM can generate instances for a given instruction. The answer is yes, if the model is prompted correctly.



Self-Instruct Method

- **Data Generation Pipeline**
  - A pipeline was built to generate input and output instances for various tasks using the GPT-3 model.
  - 52,000 instructions were generated and filtered to ensure diversity and relevance.
  - Analysis showed that 40-50% of the instructions generated meaningful and valid instances.
- **Fine-Tuning and Evaluation**
  - The model was fine-tuned using the generated data and evaluated on various tasks.
  - Strong performance was achieved, demonstrating that language models can generate high-quality instances for a given instruction.

Self-Instruct Results

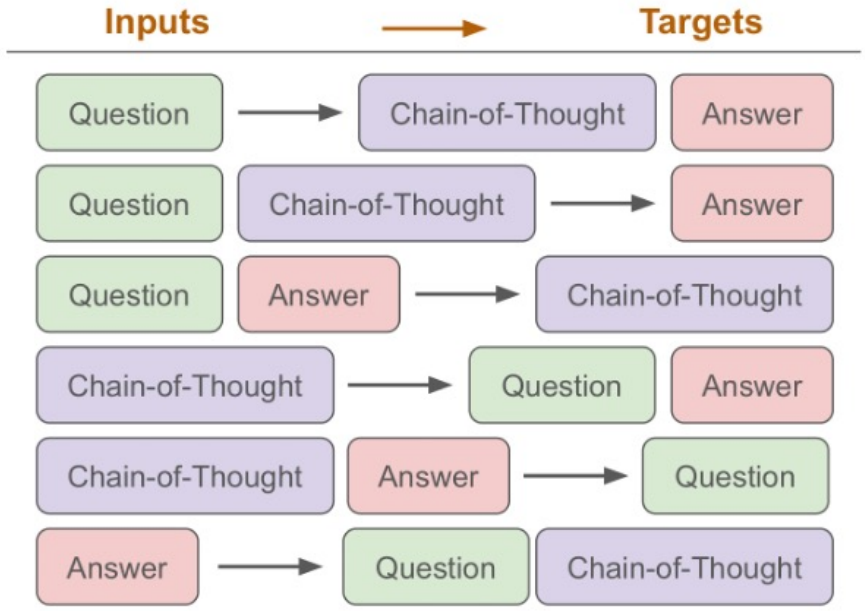
1. Self-Instruct improves the instruction-following ability of GPT-3 by a large margin on unseen tasks.
2. The generated data is diverse, with a decent number of new instructions and instances.
3. Data quality is good, with most instructions being meaningful, but some instances containing noise.
4. SELF-INSTRUCT outperforms models trained on other datasets (T0, SUPERNI) and is comparable to InstructGPT.
5. Increasing the data size leads to consistent improvement, but the improvement plateaus after 16K.

The FLAN Collection (2022-10)

- A publicly available collection of tasks, templates, and methods for instruction tuning, released to address the gap in accessible data for instruction tuning research.
- The collection is designed to facilitate effective instruction tuning and advance language model capabilities in processing new tasks
- **Dataset Structure:**
  - Task Sources: Flan 2021, T0-SF, Super-Natural Instructions, Chain-of-Thought, Dialog, Program Synthesis
  - Task Types: Zero-shot, few-shot, chain-of-thought, held-in, held-out tasks
  - Prompt Settings: Zero-shot, few-shot, mixed prompts
  - Input Inversion: Generating questions from answers
  - Weighted Mixture: Optimized task source weighting

Inversions

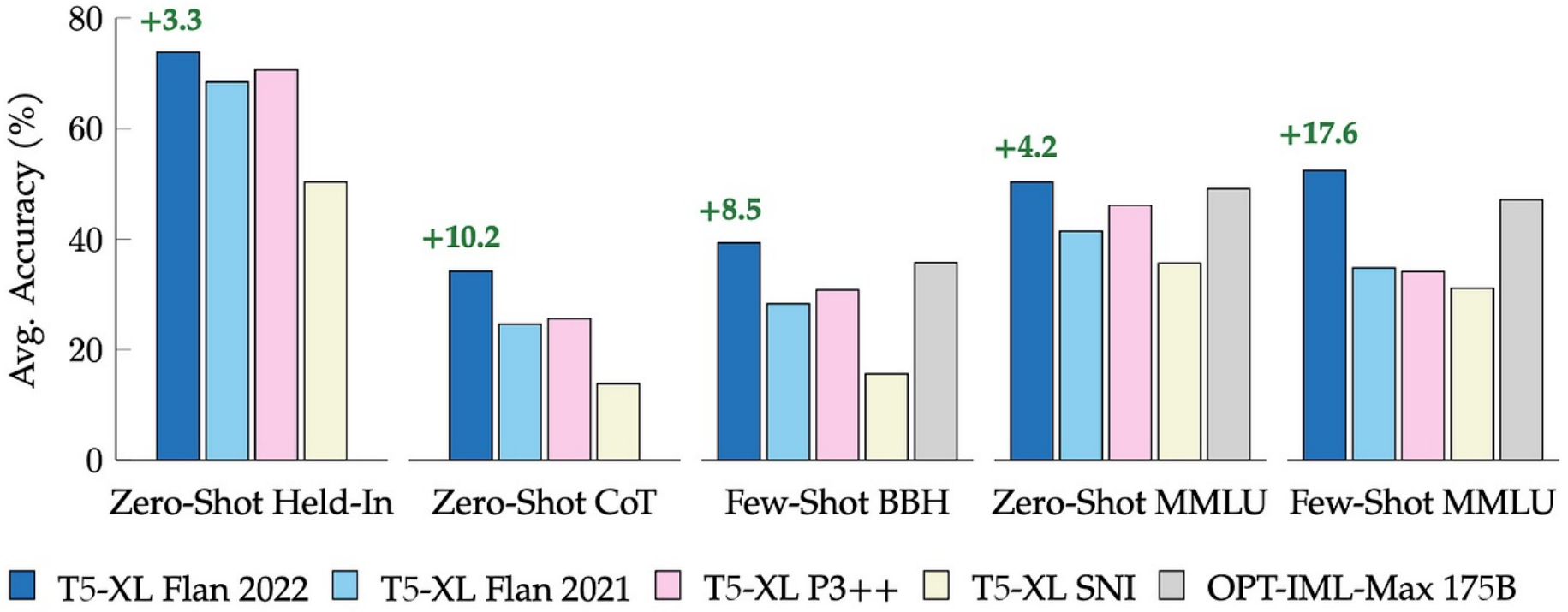
- Creating inversions for other datasets is straightforward when compared with CoT datasets.
- While others have Question-Answer pairs CoT datasets have an additional intermediate step.
- Because of that, for CoT datasets, inversions are created as a permutation of three components as shown below.



Experiments with the FLAN Collection

- The authors conducted several experiments to demonstrate the effectiveness of the FLAN Collection:
  - Method ablations showed that Flan-T5 outperforms alternative instruction tuning collections.
  - Training with mixed prompt settings improved zero-shot and few-shot performance.
  - Fine-tuning small models on many tasks improved Held-In and Held-Out performance.
  - Task enrichment with input inversion benefited Held-Out performance.
  - Optimizing mixture weighting improved results.
  - Instruction tuning enhanced single-task finetuning, leading to better performance and faster convergence.

- Comparing public instruction tuning collections on Held-In, Held-Out (BIG-Bench Hard (Suzgun et al., 2022) and MMLU (Hendrycks et al., 2020)), and Chain-of-Thought evaluation suites, detailed in Appendix A.3. All models except OPT-IML-Max (175B) are T5-XL with 3B parameters. Green text indicates absolute improvement over the next best comparable T5-XL (3B) model.

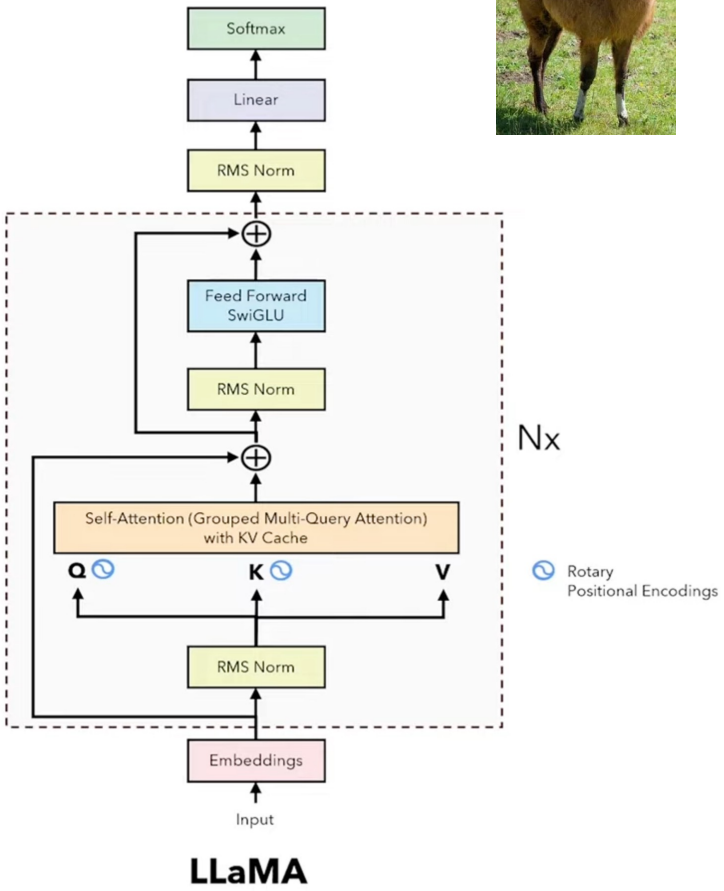


Summary of the Flan Collection 2022

- Overall, the Flan Collection and the FLAN-T5 model represent a significant breakthrough in instruction tuning for LLMs.
- The results demonstrate the potential of instruction generalization without human feedback and the effectiveness of the FLAN Collection in improving performance across a wide range of tasks.
- The paper suggests that instruction-tuned models can serve as a new standard starting point for single-task finetuning, offering faster convergence and computational benefits.

LLaMA (2023-03) and LLaMA2 (2023-07)

- **LLaMA** is an Open LLM developed by **Meta AI**, designed to be versatile and more responsible than other large language models.
- It is an **Autoregressive Language Model** with **transformer decoder-only architecture** using some advanced techniques:
  - **RMS Norm**
  - **Rotary Positional Encodings**
  - **Grouped Multi-Query Attention**
  - **KV Cache**
  - **SwiGLU**
- **Model sizes**: 7B to 65B parameters
- **Links**: Meta AI Blog, Github Repo (Official), Github Repo (Unofficial), Demo (Unofficial), OpenLLaMA: An Open Reproduction of LLaMA



LLaMA

Open Instruction Tuned Models

MT Bench 13B: 4.53

Alpaca

13 Mar. 2023

- 52k self-instruct style data distilled from text-davinci-003
- Model weight diff. to **LLaMA 7B**

https://crfm.stanford.edu/2023/03/13/alpaca.html

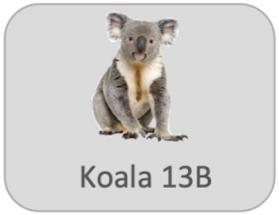
MT Bench 7B: 6.69

Vicuna (lmsys/vicuna-7b-delta-v0)

30 Mar. 2023

- Fine-tunes ChatGPT data from ShareGPT
- **LLaMA 7B and 13B** diff's
- Introduces LLM-as-a-judge

https://lmsys.org/blog/2023-03-30-vicuna/



Koala

MT Bench 13B: 6.08

3 Apr. 2023

- Diverse dataset (Alpaca, Anthropic HH, ShareGPT, WebGPT…)
- Human evaluation
- **LLaMA 7B** diff.

https://bair.berkeley.edu/blog/2023/04/03/koala/

Dolly

MT Bench 12B: 3.28

12 Apr. 2023

- 15k human written data
- Trained on **Pythia 12B**

https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm

Instruction Finetuned LLaMA Models

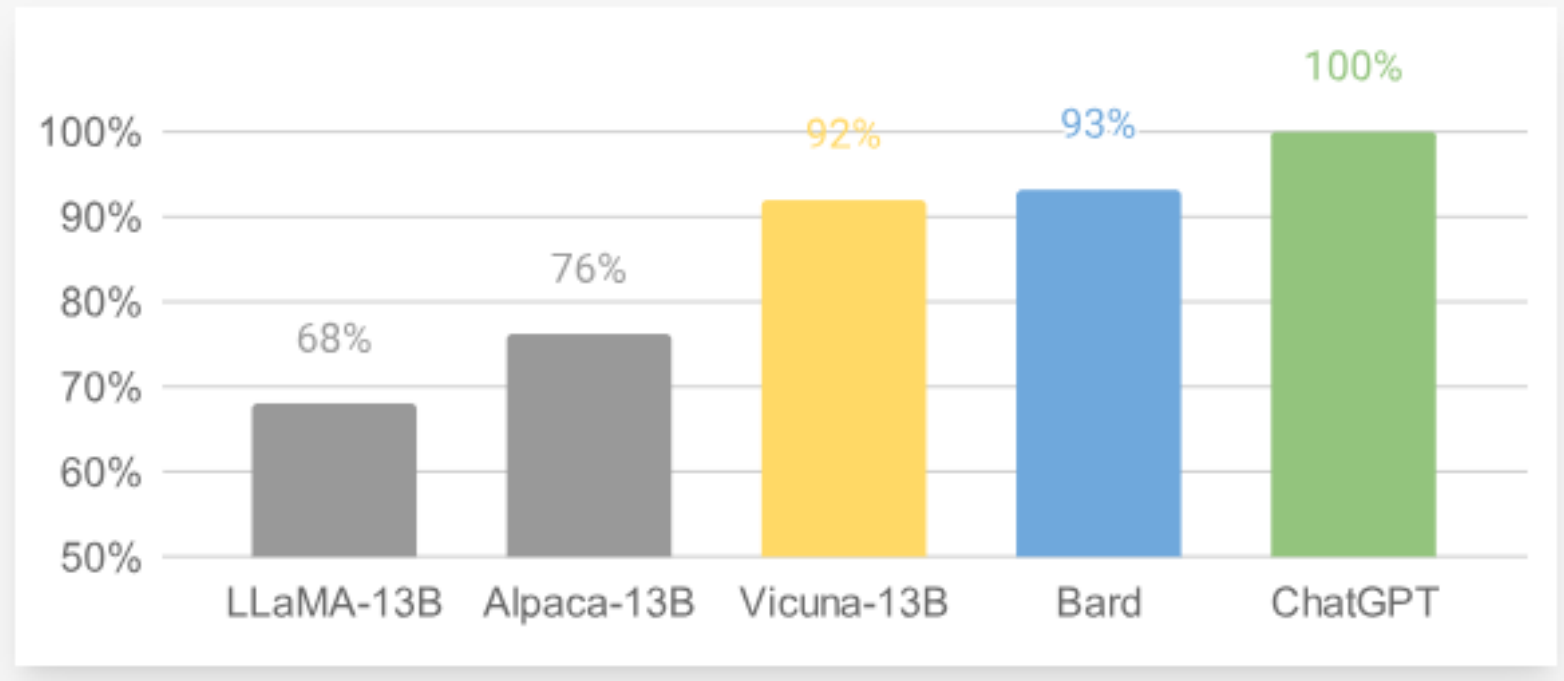
- **Alpaca** (Stanford)
  - **Alpaca** is a small instruction fine-tuned model based on LLaMA 7B model. It is designed to be easy and cheap to reproduce while performing well on various instruction-following tasks.
  - Alpaca was trained on a dataset of 52K demonstrations of instruction-following.

- **Vicuna** (UC Berkeley, CMU, Stanford, and UC San Diego)
  - **Vicuna** is a chatbot LLM model that trained by fine-tuning LLaMA on user-shared conversations collected from ShareGPT. It is trained for offering natural and engaging conversation capabilities

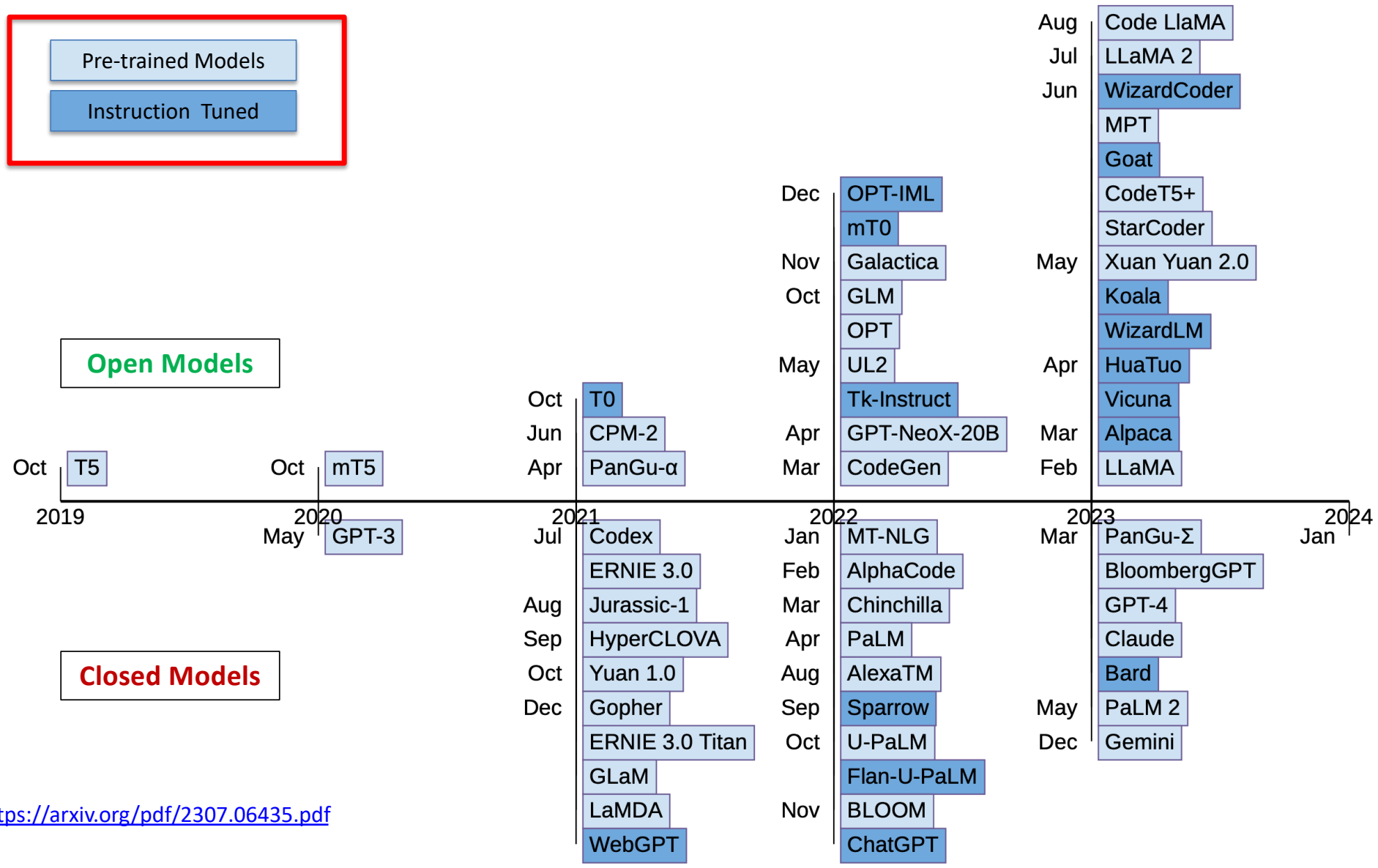
https://blog.gopenai.com/top-open-sourced-large-language-models-revolutionizing-conversational-ai-dc4a2e35c8e0

LLaMA vs Alpaca vs Vicuna Models

- To evaluate the performance of these models, a set of basic questions were created, and GPT-4 was used to score the responses of the LLaMA-13B, Alpaca-13B, Vicuna-13B, Google Bard and OpenAI ChatGPT:



Instruction fine-tuned LLMs	# Params	Base Model	Fine-tuning Trainset		
			Self-build	Dataset Name	Size
Instruct-GPT (Ouyang et al., 2022)	176B	GPT-3 (Brown et al., 2020b)	Yes	-	-
BLOOMZ (Muennighoff et al., 2022)[1]	176B	BLOOM (Scao et al., 2022)	No	xP3	-
FLAN-T5 (Chung et al., 2022)[2]	11B	T5 (Raffel et al., 2019)	No	FLAN 2021	-
Alpaca (Taori et al., 2023a)[3]	7B	LLaMA (Touvron et al., 2023a)	Yes	-	52K
Vicuna (Chiang et al., 2023)[4]	13B	LLaMA (Touvron et al., 2023a)	Yes	-	70K
GPT-4-LLM (Peng et al., 2023)[5]	7B	LLaMA (Touvron et al., 2023a)	Yes	-	52K
Claude (Bai et al., 2022b)	-	-	Yes	-	-
WizardLM (Xu et al., 2023a)[6]	7B	LLaMA (Touvron et al., 2023a)	Yes	Evol-Instruct	70K
ChatGLM2 (Du et al., 2022)[7]	6B	GLM (Du et al., 2022)	Yes	-	1.1 Tokens
LIMA (Zhou et al., 2023)	65B	LLaMA (Touvron et al., 2023a)	Yes	-	1K
OPT-IML (Iyer et al., 2022)[8]	175B	OPT (Zhang et al., 2022a)	No	-	-
Dolly 2.0 (Conover et al., 2023a)[9]	12B	Pythia (Biderman et al., 2023)	No	-	15K
Falcon-Instruct (Almazrouei et al., 2023a)[10]	40B	Falcon (Almazrouei et al., 2023b)	No	-	-
Guanaco (JosephusCheung, 2021)[11]	7B	LLaMA (Touvron et al., 2023a)	Yes	-	586K
Minotaur (Collective, 2023)[12]	15B	Starcoder Plus (Li et al., 2023f)	No	-	-
Nous-Hermes (NousResearch, 2023)[13]	13B	LLaMA (Touvron et al., 2023a)	No	-	300K+
TÜLU (Wang et al., 2023c)[14]	6.7B	OPT (Zhang et al., 2022a)	No	Mixed	-
YuLan-Chat (YuLan-Chat-Team, 2023)[15]	13B	LLaMA (Touvron et al., 2023a)	Yes	-	250K
MOSS (Tianxiang and Xipeng, 2023)[16]	16B	-	Yes	-	-
Airoboros (Durbin, 2023)[17]	13B	LLaMA (Touvron et al., 2023a)	Yes	-	-
UltraLM (Ding et al., 2023a)[18]	13B	LLaMA (Touvron et al., 2023a)	Yes	-	-



https://arxiv.org/pdf/2307.06435.pdf

Conclusion

- **Instruction tuning research show that LLMs can perform reasonably well without massive amounts of labeled data**. This is because these models have already learned many tasks and skills during pre-training.
- Recent work has also focused on replicating the success of GPT and other LLMs using self-instruct methods, distilling knowledge from more powerful models into smaller ones.
- However, concerns about licensing terms and restricted use of model outputs need to be addressed.
- To overcome this, we can use open-source models or invest in human labeling.
- Future research directions include understanding instruction tuning, building open-source models, and developing new algorithms to improve performance.

Instruction Tuning is a core of Modern LLM Training Pipelines