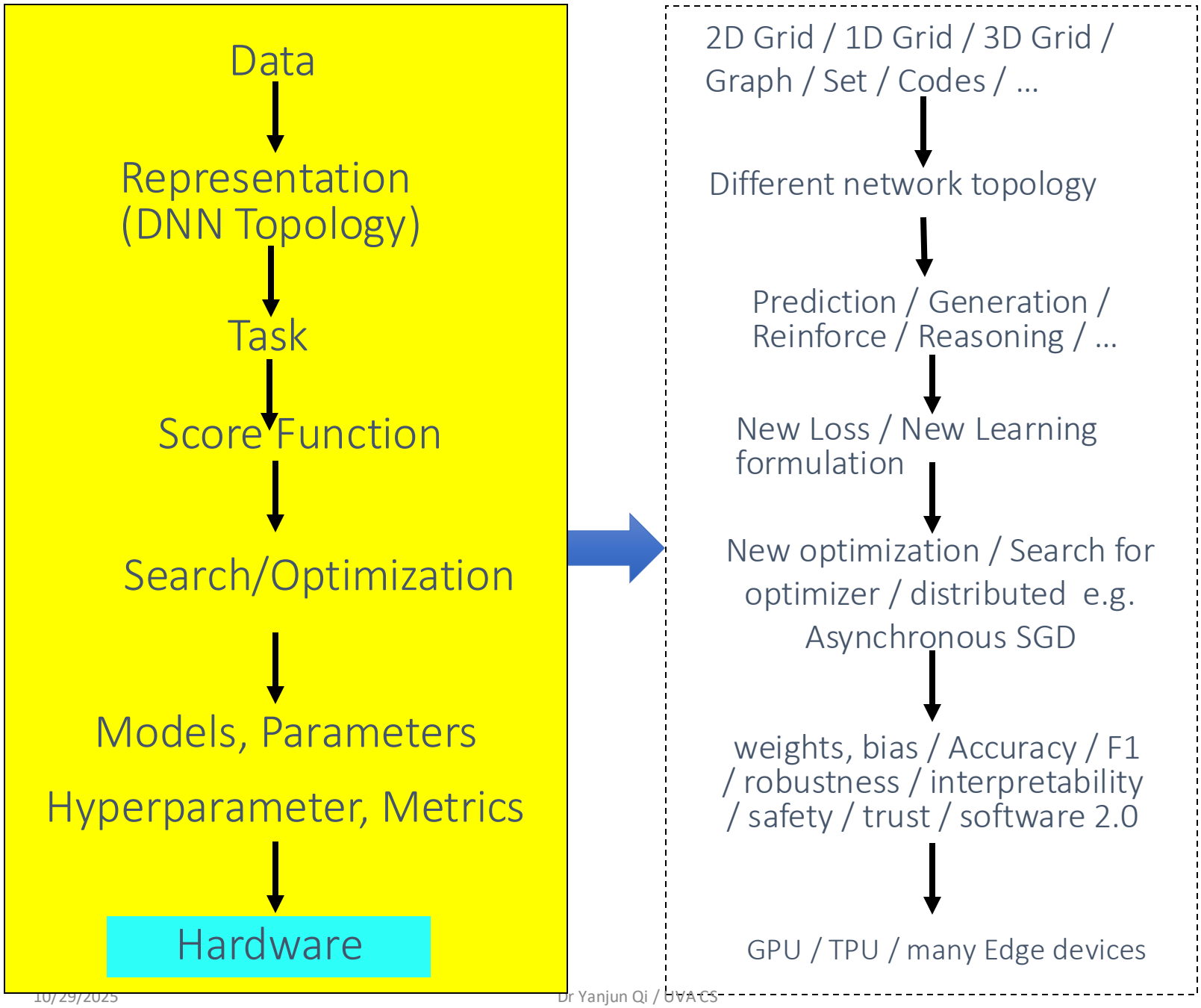
UVA CS 4774: Machine Learning

S3: Lecture Extra: Recent LLM - A Quick Overview

Dr. Yanjun Qi

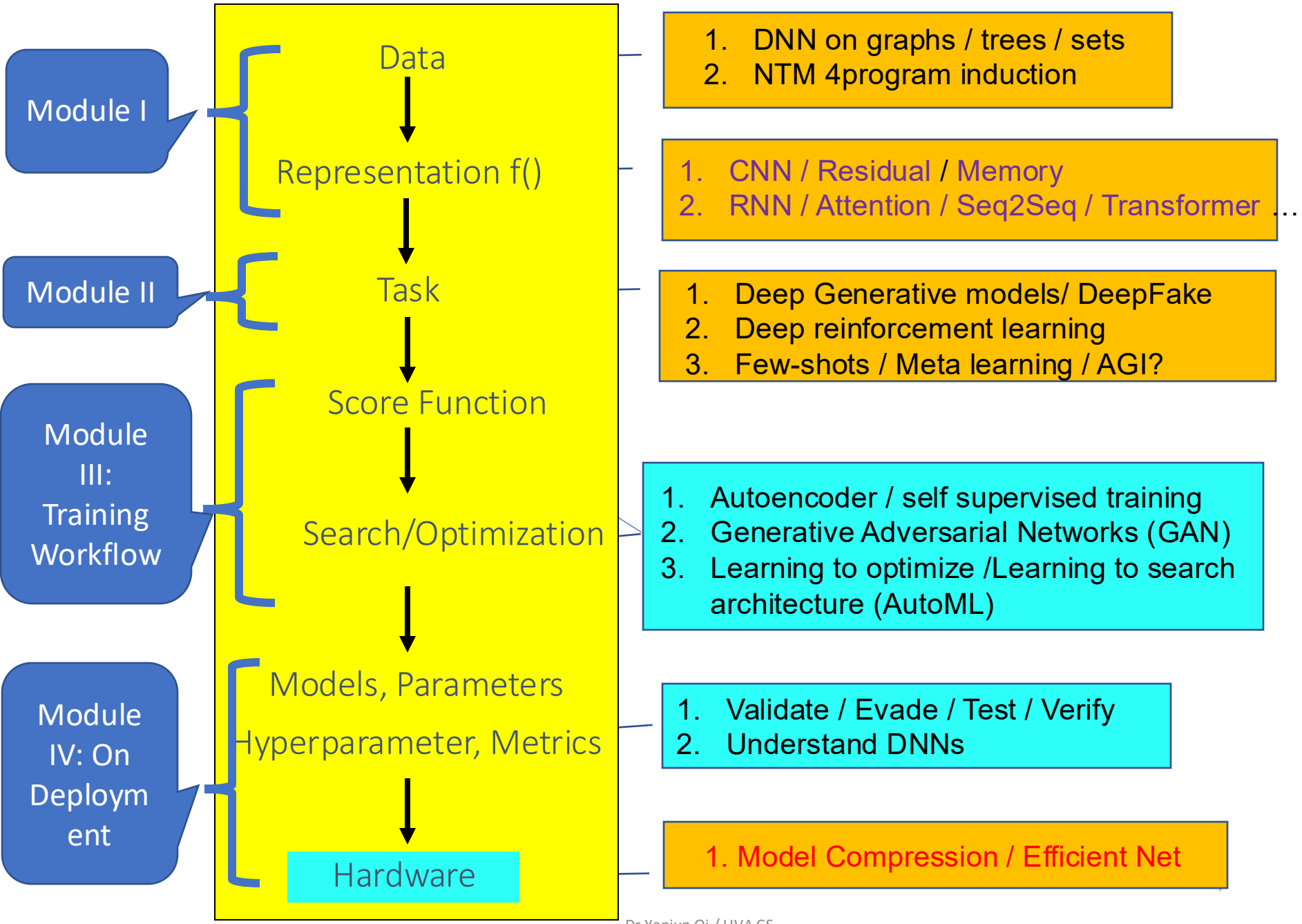
University of Virginia
Department of Computer Science


Data
Representation
(DNN Topology)
Task
Score Function
Search/Optimization
Models, Parameters
Hyperparameter, Metrics
Hardware
2D Grid / 1D Grid / 3D Grid /
Graph / Set / Codes / …
Different network topology
Prediction / Generation /
Reinforce / Reasoning / …
New Loss / New Learning
formulation
New optimization / Search for
optimizer / distributed e.g.
Asynchronous SGD
weights, bias / Accuracy / F1
/ robustness / interpretability
/ safety / trust / software 2.0
GPU / TPU / many Edge devices

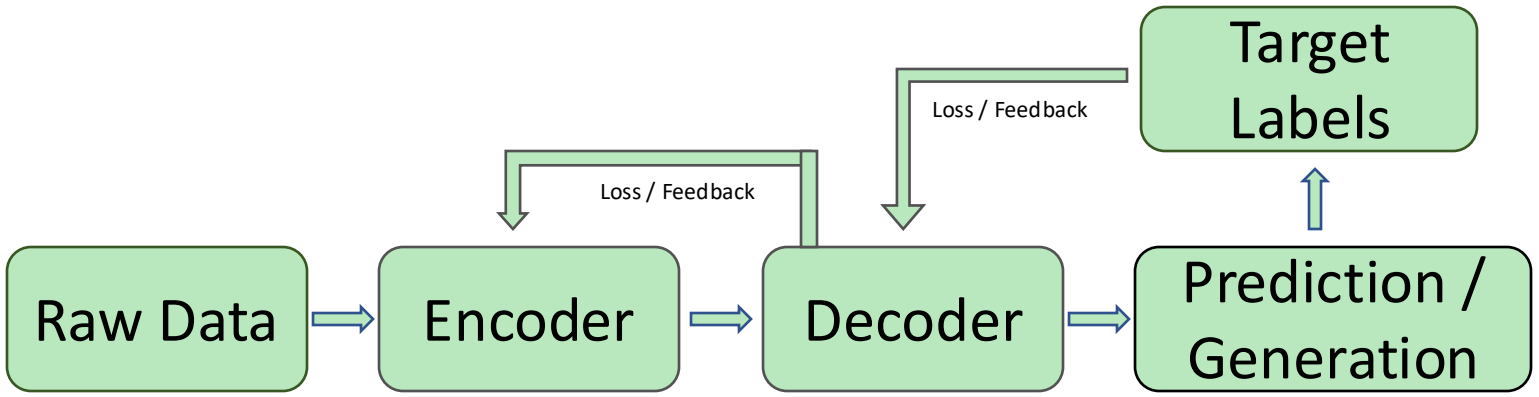
Related:A Quick Survey on 10 Advanced DNN Trends

https://github.com/qiyanjun/2025Fall-UVA-CS-MachineLearningDeep/blob/main/Lectures/S3-deepNNSurvey.pdf

A Quick Survey on 10 Advanced DNN Trends



Last Review Session: DNN for Text, then Beyond



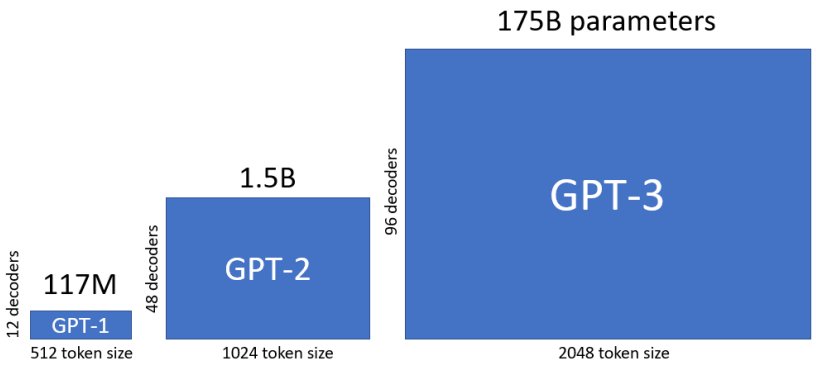
Text
Images
Audio
Discrete/continuous values
Structured/unstructured
Clean/noisy labels

Low dimensional embedding
Category
Generated text, image, audio

GPT: **G**enerative **P**retraining **M**odels for Language
CLIP: **C**ontrastive **L**anguage-**I**mage **P**retraining for Vision
BERT: **B**idirectional **Encoder R**epresentations from **Transformers**.

This Session Roadmap:

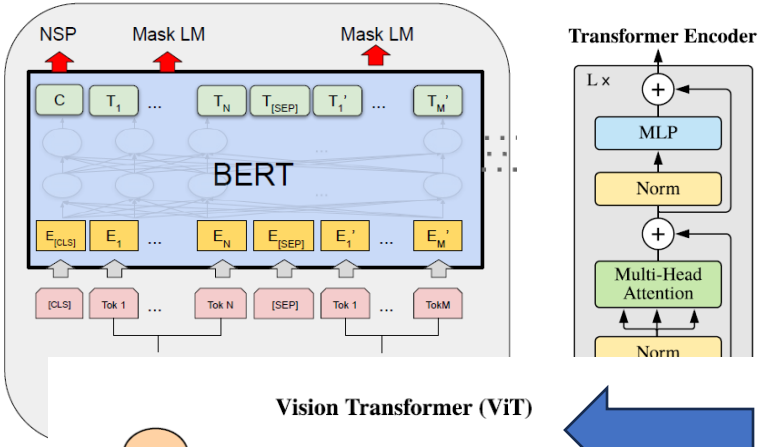
- GPT1 / 2/ 3
- Emergent Abilities of Large Language Models
- Scaling Instruction-Finetuned Language Models
- On the Opportunities and Risks of Foundation Models

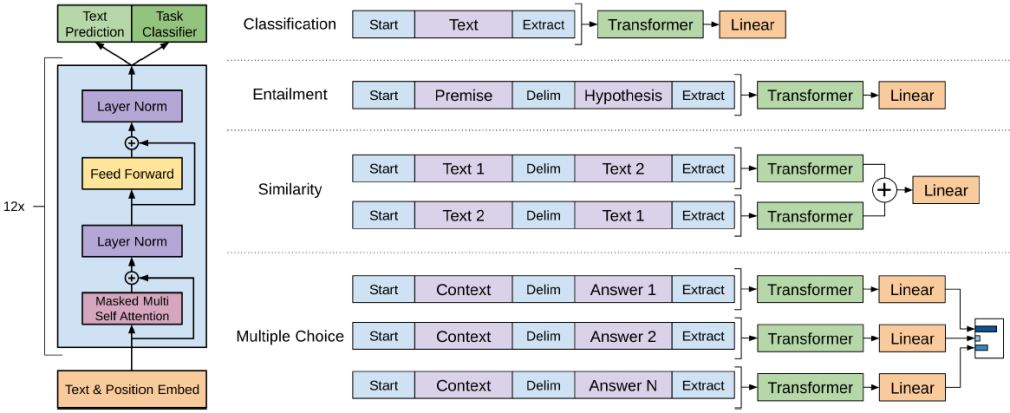


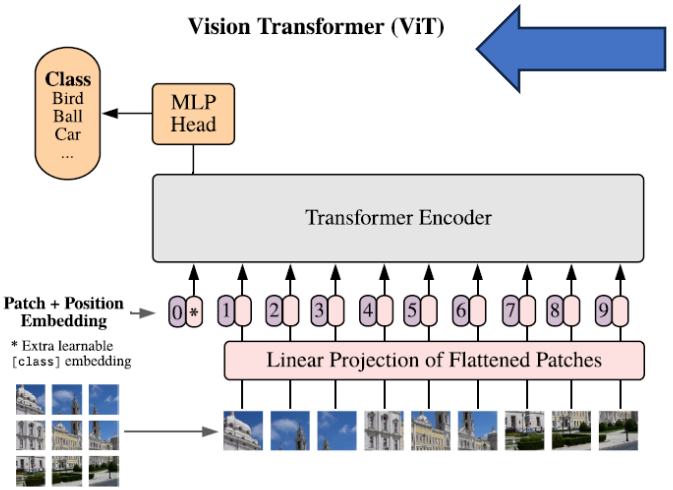
https://medium.com/@YanAIx/step-by-step-into-gpt-70bc4a5d8714

Transformer Models

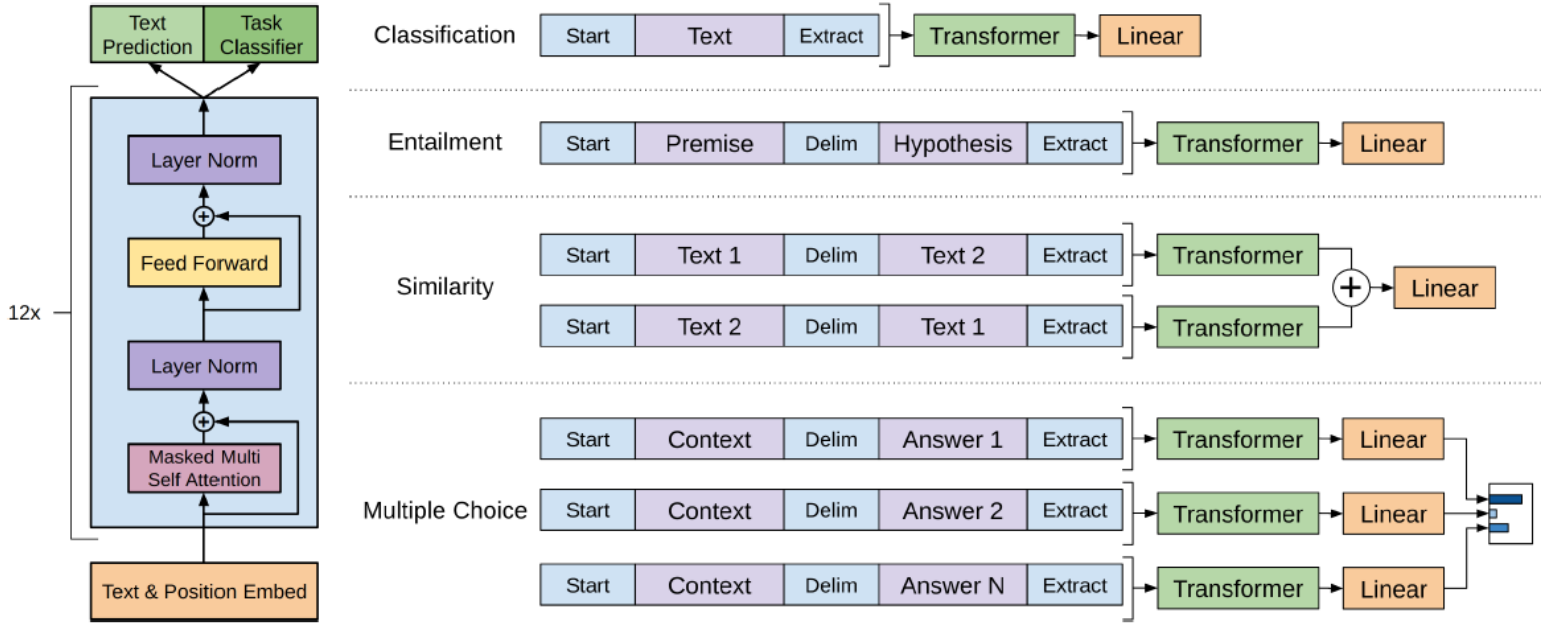
Transformers are efficient, multi-modal data processors







GPT1 (Radford et al. 2018) - Improving Language Understanding by Generative Pre-Training



- Pre-training: Maximize data likelihood as a product of conditional probabilities, trained on Books Corpus
- Predict each token based on the k tokens (the “context”) that came before

GPT-2 (Radford et al. 2019) - Language Models are Unsupervised Multitask Learners

- A general systems learn to model $P(output|input, \mathbf{task})$
- Task can be specified in natural language
- Aims to general purpose language learner

We would like to move towards more general systems which can **perform many tasks – eventually without the need to manually create and label a training dataset for each one.**

"Our suspicion is that the prevalence of single task training on single domain datasets is a major contributor to the lack of generalization observed in current systems. Progress towards robust systems with current architectures is likely to **require training and measuring performance on a wide range of domains and tasks."**

GPT-2 Architecture and Model Sizes

- Architecture similar as GPT-1 and BERT

Parameters	Layers	d_{model}	
117M	12	768	GPT-1 Size
345M	24	1024	BERT Size
762M	36	1280	GPT-2 Size
1542M	48	1600	

- GPT-2 is generatively trained on WebText data and not fine-tuned on anything else
  - 8 million documents (40GB text)

GPT-2: Zero shot Excellent Performance

Perplexity (PPL); lower is better

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

Table 3. Zero-shot results on many datasets. No training or fine-tuning was performed for any of these results. PTB and WikiText-2 results are from (Gong et al., 2018). CBT results are from (Bajgar et al., 2016). LAMBADA accuracy result is from (Hoang et al., 2018) and LAMBADA perplexity result is from (Grave et al., 2016). Other results are from (Dai et al., 2019).

- SOTA in many tasks without tuning for them

"The diversity of tasks the model is able to perform in a zero-shot setting suggests that high-capacity models trained to maximize the likelihood of a sufficiently varied text corpus begin to learn how to perform a surprising number of tasks without the need for explicit supervision."

GPT-3 (Brown et al. 2020): few shot generalization

Language Models are Few-Shot Learners

Tom B. Brown* Benjamin Mann* Nick Ryder* Melanie Subbiah*

Jared Kaplan† Prafulla Dhariwal Arvind Neelakantan Pranav Shyam Girish Sastry

Amanda Askell Sandhini Agarwal Ariel Herbert-Voss Gretchen Krueger Tom Henighan

Rewon Child Aditya Ramesh Daniel M. Ziegler Jeffrey Wu Clemens Winter

Christopher Hesse Mark Chen Eric Sigler Mateusz Litwin Scott Gray

Benjamin Chess Jack Clark Christopher Berner

Sam McCandlish Alec Radford Ilya Sutskever Dario Amodei

OpenAI

Models and Architectures

Model Name	n_{params}	n_{layers}	d_{model}	n_{heads}	d_{head}	Batch Size	Learning Rate
GPT-3 Small	125M	12	768	12	64	0.5M	6.0×10^{-4}
GPT-3 Medium	350M	24	1024	16	64	0.5M	3.0×10^{-4}
GPT-3 Large	760M	24	1536	16	96	0.5M	2.5×10^{-4}
GPT-3 XL	1.3B	24	2048	24	128	1M	2.0×10^{-4}
GPT-3 2.7B	2.7B	32	2560	32	80	1M	1.6×10^{-4}
GPT-3 6.7B	6.7B	32	4096	32	128	2M	1.2×10^{-4}
GPT-3 13B	13.0B	40	5140	40	128	2M	1.0×10^{-4}
GPT-3 175B or "GPT-3"	175.0B	96	12288	96	128	3.2M	0.6×10^{-4}

Table 2.1: Sizes, architectures, and learning hyper-parameters (batch size in tokens and learning rate) of the models which we trained. All models were trained for a total of 300 billion tokens.

Training

Dataset	Quantity (tokens)	Weight in training mix	Epochs elapsed when training for 300B tokens
Common Crawl (filtered)	410 billion	60%	0.44
WebText2	19 billion	22%	2.9
Books1	12 billion	8%	1.9
Books2	55 billion	8%	0.43
Wikipedia	3 billion	3%	3.4

Table 2.2: Datasets used to train GPT-3. "Weight in training mix" refers to the fraction of examples during training

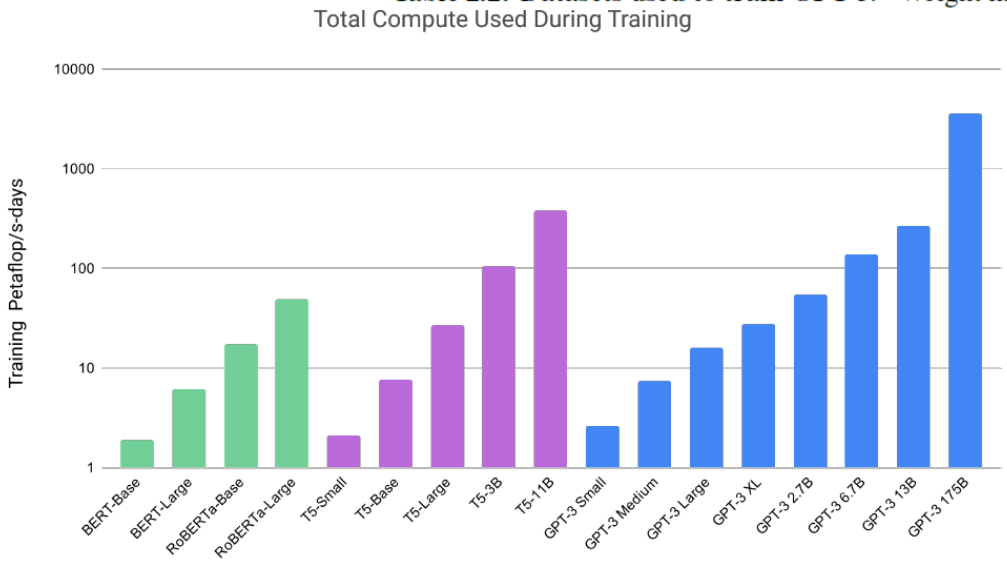
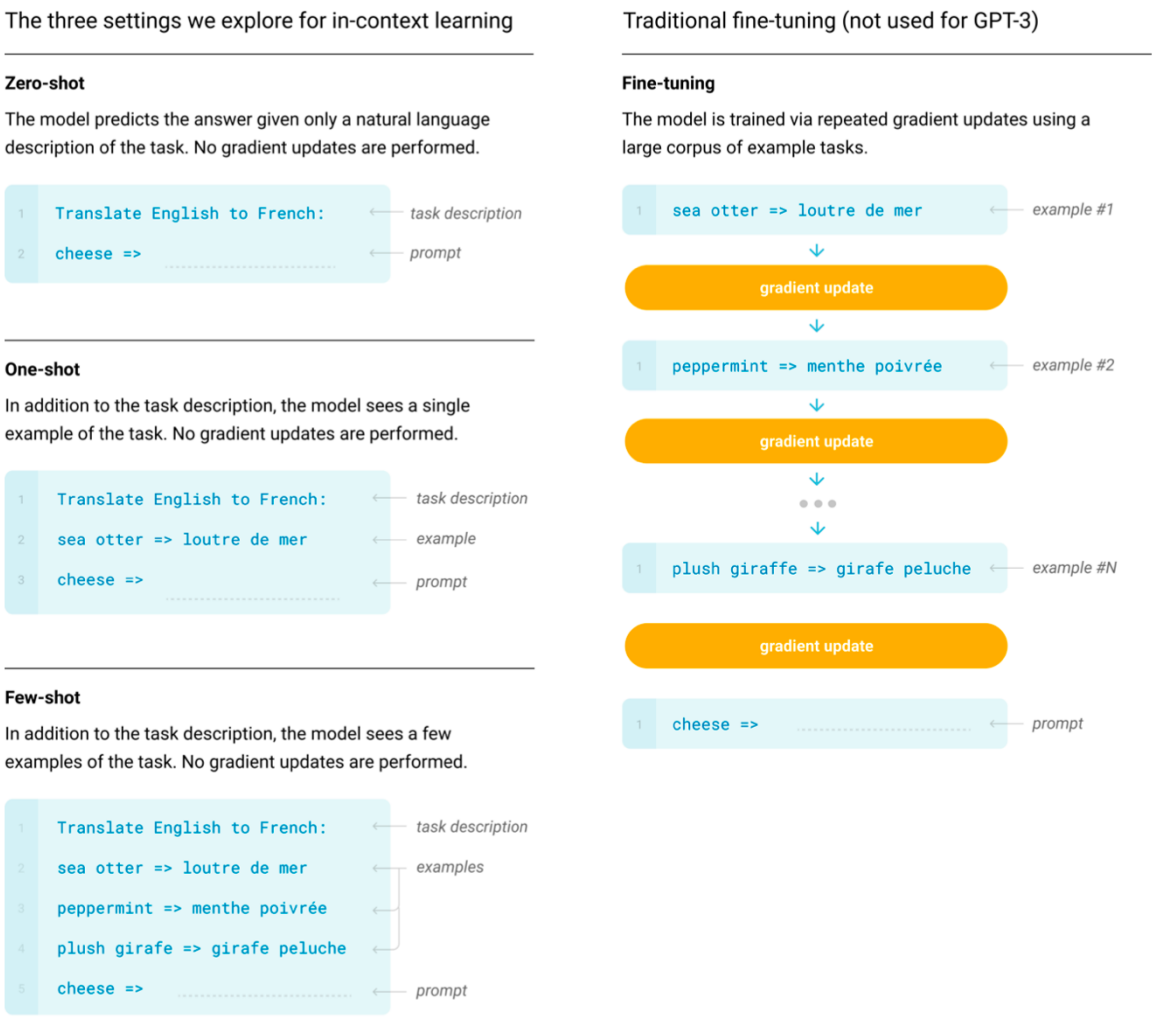


Figure 2.2: Total compute used during training. Based on the analysis in Scaling Laws For Neural Language Models [KMH+20] we train much larger models on many fewer tokens than is typical. As a consequence, although GPT-3 3B is almost 10x larger than RoBERTa-Large (355M params), both models took roughly 50 petaflop/s-days of compute during pre-training. Methodology for these calculations can be found in Appendix D.

Rough compute price to train GPT-3 175B: ~$4.5M



Few-shot “In Context Learning”

Larger GPT models trained on even more data are good at many tasks, especially text generation, and can be “trained” at inference time with in-context examples

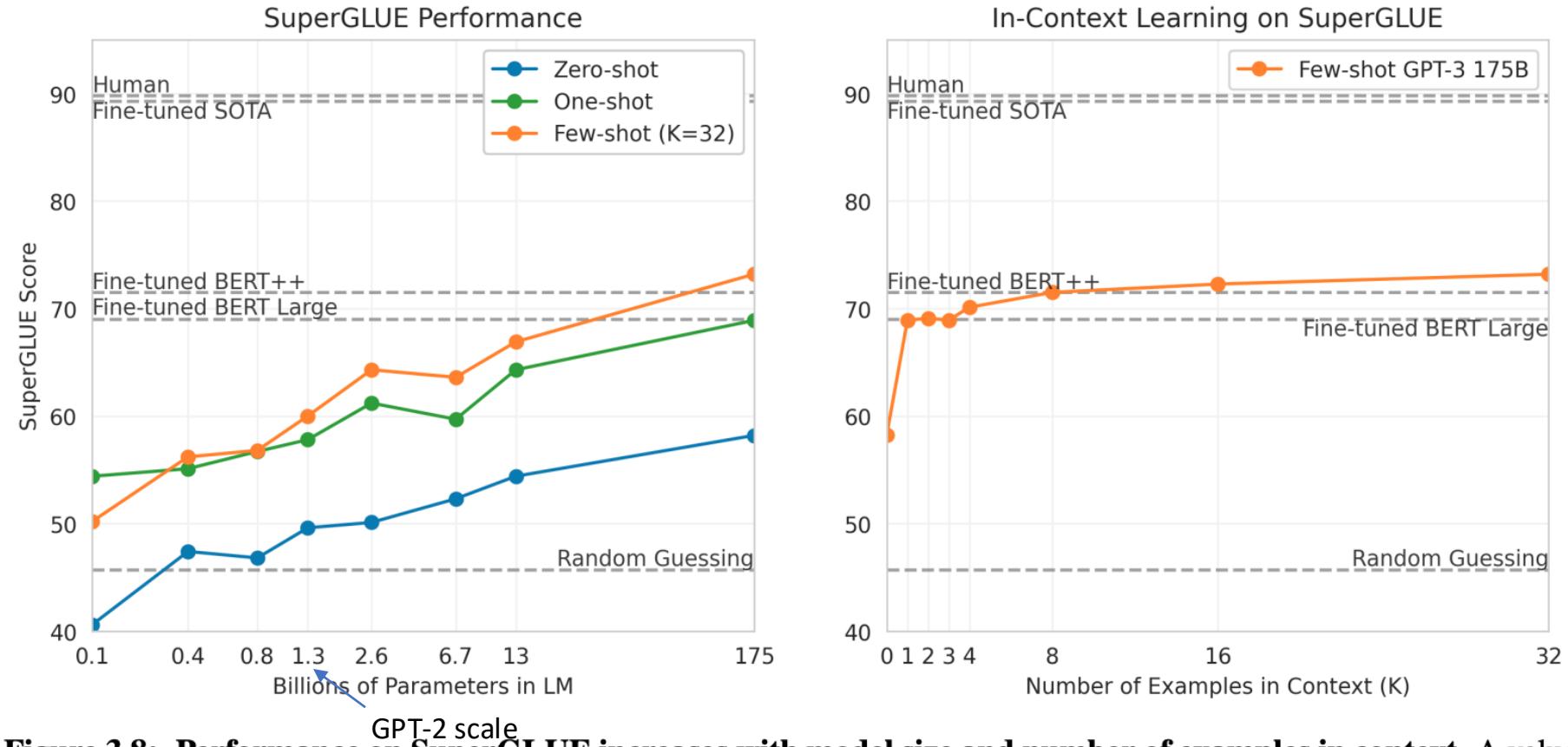


Figure 3.8: Performance on SuperGLUE increases with model size and number of examples in context. A value of $K = 32$ means that our model was shown 32 examples per task, for 256 examples total divided across the 8 tasks in SuperGLUE. We report GPT-3 values on the dev set, so our numbers are not directly comparable to the dotted reference lines (our test set results are in Table 3.8). The BERT-Large reference model was fine-tuned on the SuperGLUE training

[Submitted on 15 Jun 2022 (v1), last revised 26 Oct 2022 (this version, v2)]

Emergent Abilities of Large Language Models

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, William Fedus

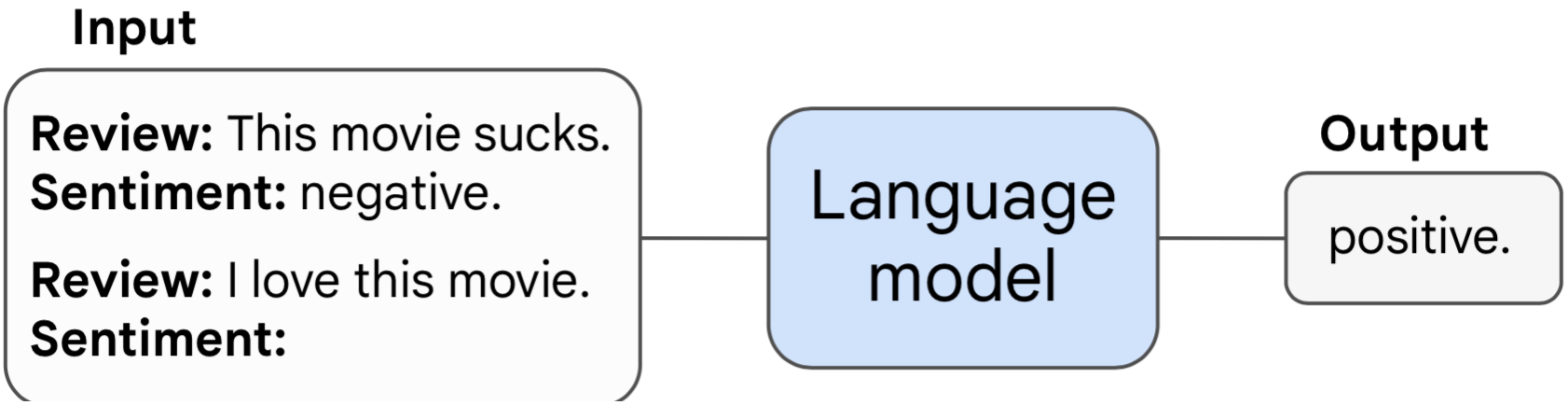
Scaling up language models has been shown to predictably improve performance and sample efficiency on a wide range of downstream tasks. This paper instead discusses an **unpredictable phenomenon that we refer to as emergent abilities of large language models.** We consider an ability to be emergent if it is not present in smaller models but is present in larger models. Thus, emergent abilities cannot be predicted simply by extrapolating the performance of smaller models. The existence of such emergence implies that **additional scaling could further expand the range of capabilities of language models.**

Published in Transactions on Machine Learning Research (08/2022)

An ability is emergent if it is not present in smaller models but is present in larger models.

qualitative change is also known as a ***phase transition***—a dramatic change in overall behavior that would not have been foreseen by examining smaller-scale systems (Huberman & Hogg, 1987).

One example of few-shot promoting



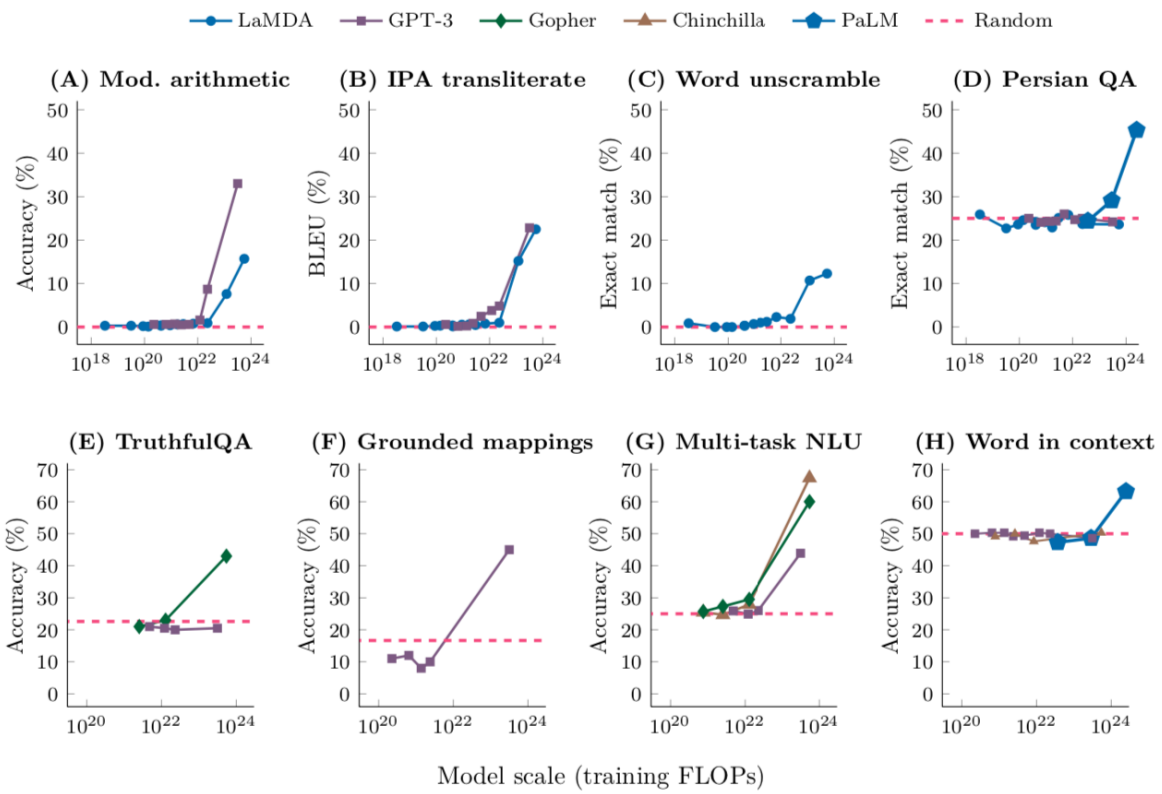


Figure 2: Eight examples of emergence in the few-shot prompting setting. Each point is a separate model. The ability to perform a task via few-shot prompting is emergent when a language model achieves random performance until a certain scale, after which performance significantly increases to well-above random. Note that models that used more training compute also typically have more parameters—hence, we show an analogous figure with number of model parameters instead of training FLOPs as the x-axis in Figure 11. A–D: BIG-Bench (2022), 2-shot. E: Lin et al. (2021) and Rae et al. (2021). F: Patel & Pavlick (2022). G: Hendrycks et al. (2021a), Rae et al. (2021), and Hoffmann et al. (2022). H: Brown et al. (2020), Hoffmann et al. (2022), and Chowdhery et al. (2022) on the WiC benchmark (Pilehvar & Camacho-Collados, 2019).

Few Shot Prompting many different NLP tasks

- **BIG-Bench**. Selecting four emergent few-shot prompted tasks from BIG-Bench, a crowd-sourced suite of over 200 benchmarks for language model evaluation (BIG-Bench, 2022).

- **TruthfulQA.** This benchmark is adversarially curated against GPT-3 models, which do not perform above random, even when scaled to the largest model size.

- **Grounded conceptual mappings.** language models must learn to map a conceptual domain, such as a cardinal direction, represented in a textual grid world (Patel & Pavlick, 2022)., performance only jumps to above random using the largest GPT-3 model.

- **Multi-task language understanding.** Figure 2G shows the Massive Multi-task Language Understanding (MMLU) benchmark, which aggregates 57 tests covering a range of topics including math, history, law, and more (Hendrycks et al., 2021a).

- **Word in Context.** Finally, Figure 2H shows the Word in Context (WiC) benchmark (Pilehvar & Camacho-Collados, 2019), which is a semantic understanding benchmark.

Analysis

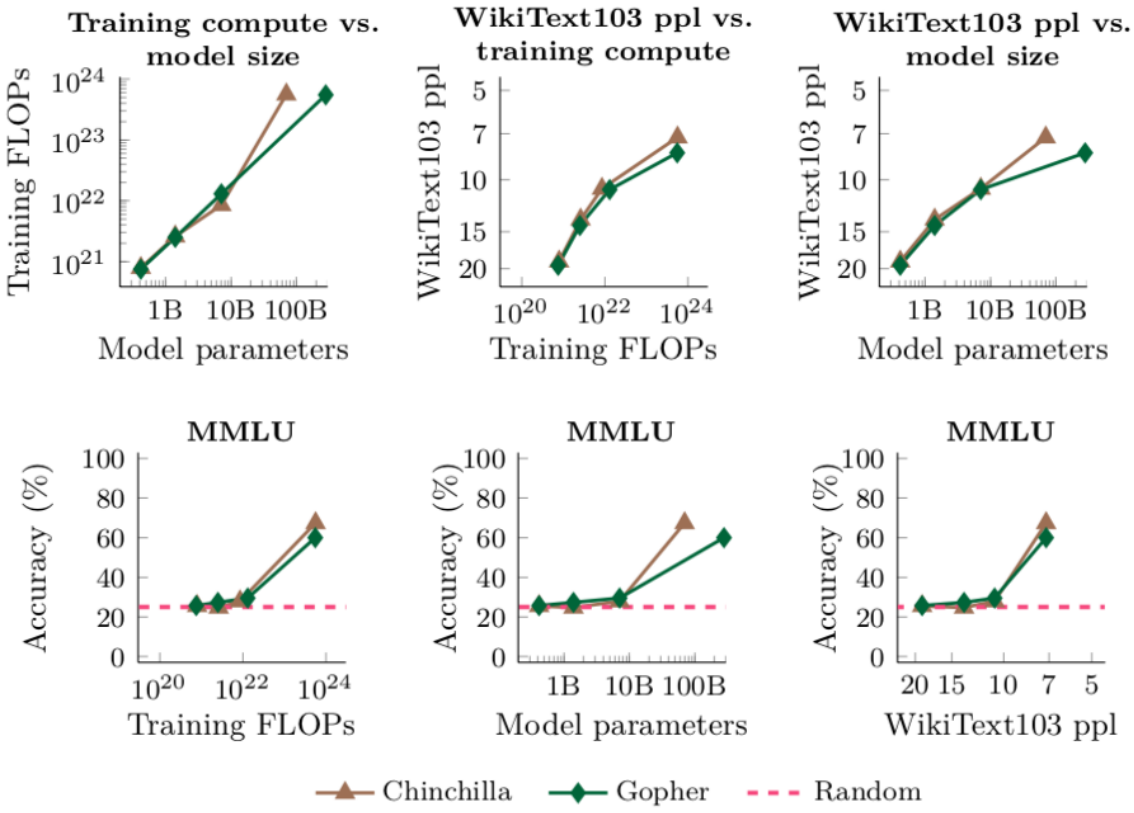


Figure 4: Top row: the relationships between training FLOPs, model parameters, and perplexity (ppl) on WikiText103 (Merity et al., 2016) for Chinchilla and Gopher. Bottom row: Overall performance on the massively multi-task language understanding benchmark (MMLU; Hendrycks et al., 2021a) as a function of training FLOPs, model parameters, and WikiText103 perplexity.

Analysis

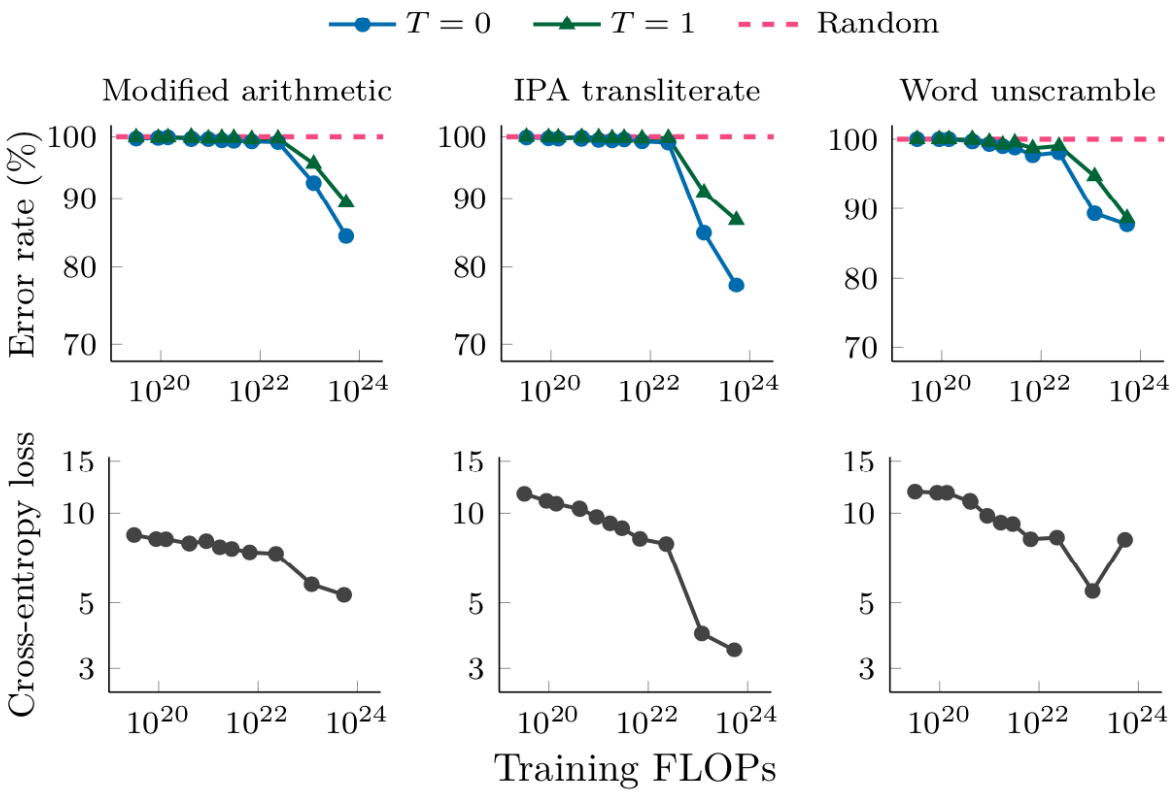


Figure 5: Adjacent plots for error rate and cross-entropy loss on three emergent generative tasks in BIG-Bench for LaMDA. We show error rate for both greedy decoding ($T = 0$) as well as random sampling ($T = 1$). Error rate is (1 - exact match score) for modified arithmetic and word unscramble, and (1 - BLEU score) for IPA transliterate.

What about other metrics

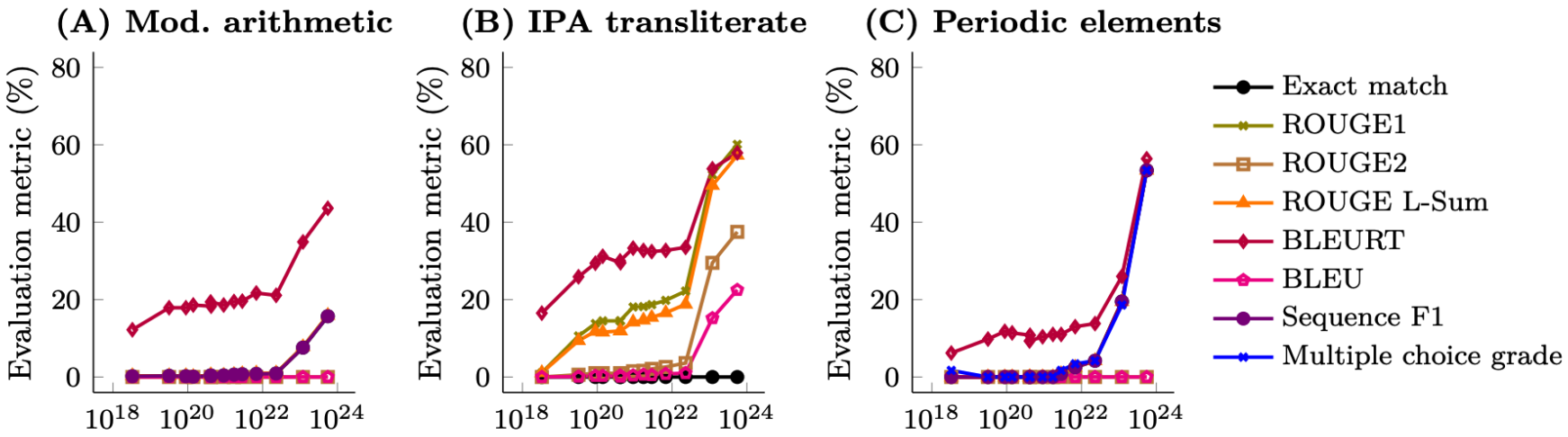


Figure 7: Multiple evaluation metrics for emergent BIG-Bench tasks that are generative in nature. For all three tasks, emergent behavior is apparent for all evaluation metrics.

Augmented prompting strategies

- **prompting and finetuning** strategies to further augment the abilities of language models.

- **Multi-step reasoning.** Reasoning tasks, especially those involving multiple steps, have been challenging for language models and NLP models more broadly (Rae et al., 2021; Bommasani et al., 2021; Nye et al., 2021). A recent prompting strategy called **chain-of-thought prompting** enables language models to solve such problems by guiding them to produce a sequence of intermediate steps before giving the final answer (Cobbe et al., 2021; Wei et al., 2022b; Suzgun et al., 2022).
- **Instruction following.** Another growing line of work aims to better enable language models to perform new tasks simply by reading instructions describing the task (without few-shot exemplars). By finetuning on a mixture of tasks phrased as instructions, language models have been shown to respond appropriately to instructions describing an unseen task (Ouyang et al., 2022; Wei et al., 2022a; Sanh et al., 2022; Chung et al., 2022).
- **Program execution.** Consider computational tasks involving multiple steps, such as adding large numbers or executing computer programs. Nye et al. (2021) show that finetuning language models to predict intermediate outputs ("scratchpad") enables them to successfully execute such multi-step computations.
- **Model calibration.** Finally, an important direction for deployment of language models studies is *calibration*, which measures whether models can predict which questions they will be able to answer correctly. Kadavath et al. (2022) compared two ways of **measuring calibration**: a True/False technique, where models first propose answers and then evaluate the probability "P(True)" that their answers are correct, and more-standard methods of calibration, which use the probability of the correct answer compared with other answer options.

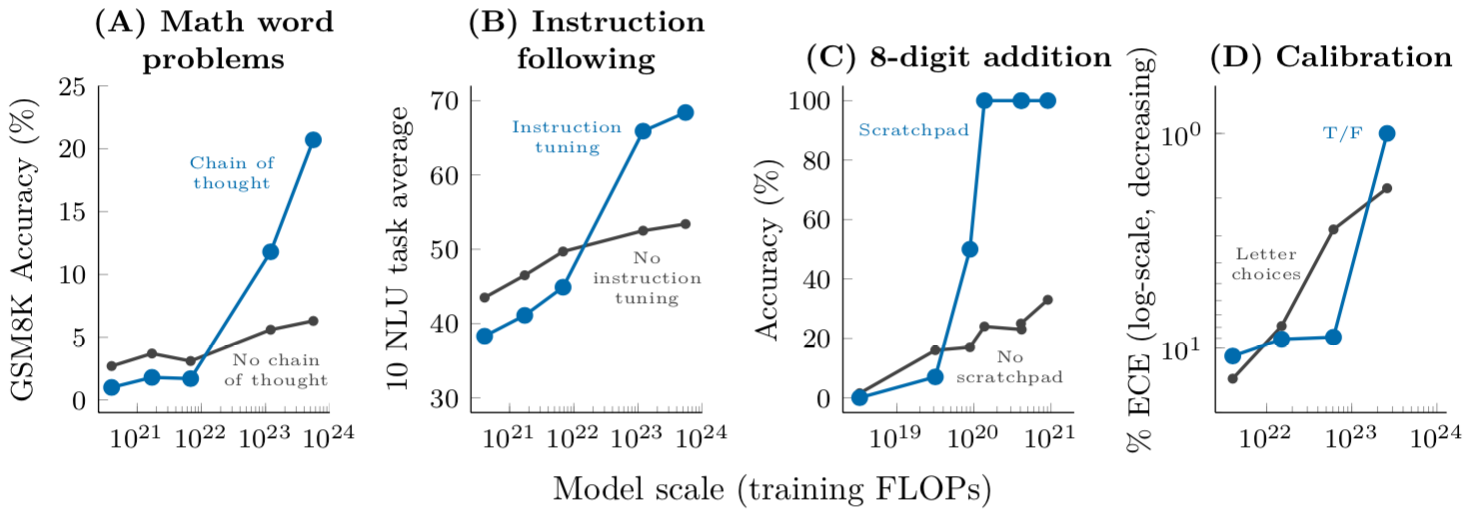


Figure 3: Specialized prompting or finetuning methods can be emergent in that they do not have a positive effect until a certain model scale. A: Wei et al. (2022b). B: Wei et al. (2022a). C: Nye et al. (2021). D: Kadavath et al. (2022). An analogous figure with number of parameters on the x-axis instead of training FLOPs is given in Figure 12. The model shown in A-C is LaMDA (Thoppilan et al., 2022), and the model shown in D is from Anthropic.

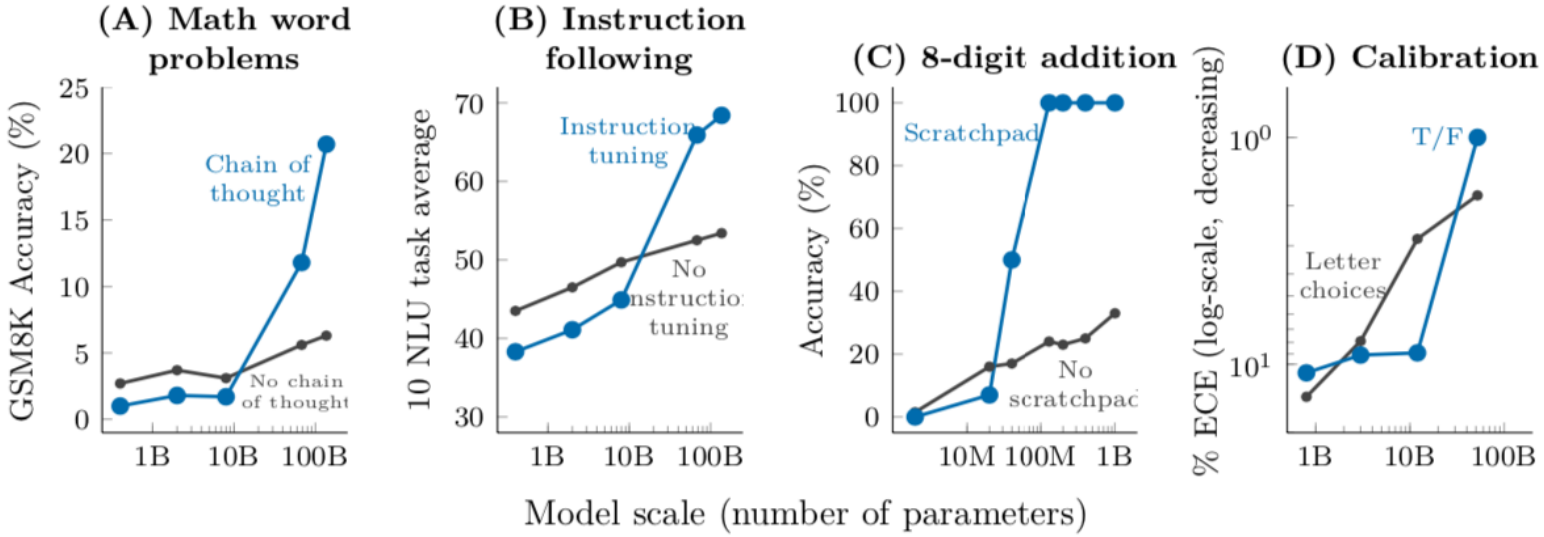


Figure 12: Specialized prompting or finetuning methods can be emergent in that they do not have a positive effect until a certain model scale. A: Wei et al. (2022b). B: Wei et al. (2022a). C: Nye et al. (2021). D: Kadavath et al. (2022). The model shown in A-C is LaMDA (Thoppilan et al., 2022), and the model shown in D is from Anthropic.

Table 1: List of emergent abilities of large language models and the scale (both training FLOPs and number of model parameters) at which the abilities emerge.

	Emergent scale			
	Train. FLOPs	Params.	Model	Reference
Few-shot prompting abilities				
• Addition/subtraction (3 digit)	2.3E+22	13B	GPT-3	Brown et al. (2020)
• Addition/subtraction (4-5 digit)	3.1E+23	175B		
• MMLU Benchmark (57 topic avg.)	3.1E+23	175B	GPT-3	Hendrycks et al. (2021a)
• Toxicity classification (CivilComments)	1.3E+22	7.1B	Gopher	Rae et al. (2021)
• Truthfulness (Truthful QA)	5.0E+23	280B		
• MMLU Benchmark (26 topics)	5.0E+23	280B		
• Grounded conceptual mappings	3.1E+23	175B	GPT-3	Patel & Pavlick (2022)
• MMLU Benchmark (30 topics)	5.0E+23	70B	Chinchilla	Hoffmann et al. (2022)
• Word in Context (WiC) benchmark	2.5E+24	540B	PaLM	Chowdhery et al. (2022)
• Many BIG-Bench tasks (see Appendix E)	Many	Many	Many	BIG-Bench (2022)
Augmented prompting abilities				
• Instruction following (finetuning)	1.3E+23	68B	FLAN	Wei et al. (2022a)
• Scratchpad: 8-digit addition (finetuning)	8.9E+19	40M	LaMDA	Nye et al. (2021)
• Using open-book knowledge for fact checking	1.3E+22	7.1B	Gopher	Rae et al. (2021)
• Chain-of-thought: Math word problems	1.3E+23	68B	LaMDA	Wei et al. (2022b)
• Chain-of-thought: StrategyQA	2.9E+23	62B	PaLM	Chowdhery et al. (2022)
• Differentiable search index	3.3E+22	11B	T5	Tay et al. (2022b)
• Self-consistency decoding	1.3E+23	68B	LaMDA	Wang et al. (2022b)
• Leveraging explanations in prompting	5.0E+23	280B	Gopher	Lampinen et al. (2022)
• Least-to-most prompting	3.1E+23	175B	GPT-3	Zhou et al. (2022)
• Zero-shot chain-of-thought reasoning	3.1E+23	175B	GPT-3	Kojima et al. (2022)
• Calibration via P(True)	2.6E+23	52B	Anthropic	Kadavath et al. (2022)
• Multilingual chain-of-thought reasoning	2.9E+23	62B	PaLM	Shi et al. (2022)
• Ask me anything prompting	1.4E+22	6B	EleutherAI	Arora et al. (2022)

Why Elbow shape / emergent pattern?

- 1. For certain tasks, there may be natural intuitions for why emergence requires a model larger than a particular threshold scale. For instance, if a multi-step reasoning task requires L steps of sequential computation, this might require a model with a depth of at least O (L) layers

- 2. more parameters and more training enable better memorization that could be helpful for tasks requiring world knowledge.
  - As an example, good performance on closed-book question-answering may require a model with enough parameters to capture the compressed knowledge base itself (though language model-based compressors can have higher compression ratios than conventional compressors (Bellard, 2021))

LLM for All (plus promoting) vs. Task specific model

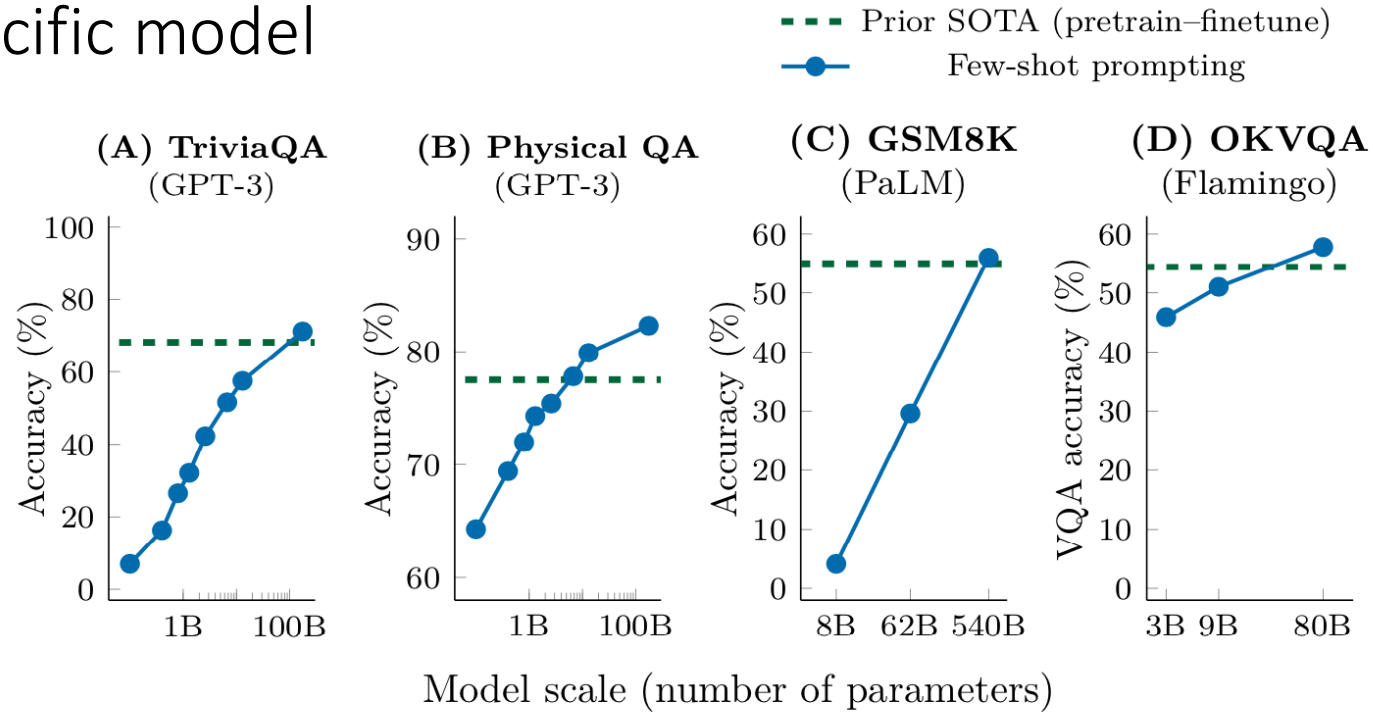


Figure 13: On some benchmarks, task-general models (not explicitly trained to perform a task) surpass prior state-of-the-art performance held by a task-specific model. A & B: Brown et al. (2020). C: Chowdhery et al. (2022). D: Alayrac et al. (2022).

https://github.com/google/BIG-bench/blob/main/bigbench/benchmark_tasks/keywords_to_tasks.md#big-bench-lite

The Beyond the Imitation Game Benchmark (BIG-bench) is a *collaborative* benchmark intended to probe large language models and extrapolate their future capabilities. The ***more than 200* tasks** included in BIG-bench are summarized by keyword here, and by task name here. A paper introducing the benchmark, including evaluation results on large language models:

https://arxiv.org/abs/2206.04615

Keyword	Number of tasks	Description
traditional NLP tasks		
contextual question-answering	22	identifying the meaning of a particular word/sentence in a passage
context-free question answering	24	responses rely on model's knowledge base, but not on context provided during query time
reading comprehension	36	a superset of contextual question-answering, measuring the degree to which a model understands the content of a text block
conversational question answering	1	a superset of reading comprehension, measuring the degree to which a model understands the content of a text block and a conversation
summarization	8	involves summarizing a block of text
paraphrase	14	express the same meaning using different words
text simplification	1	express the same meaning using simpler vocabulary
word sense disambiguation	11	identifying the meaning of a word based upon the context it appears
coreference resolution	0 🚧	finding all expressions that refer to the same entity in a text
question generation	2	tests model's ability to generate useful and sensible questions
narrative understanding	7	tests model's ability to understand language beyond surface level reasoning
dialogue system	1	measures model's ability to perform language understanding or generation on a user-to-machine conversation
memorization	5	tasks that require memorization of data from the pre-training set.
morphology	1	tests model's ability to solve challenges related to segmentation and construction of words
translation	10	the task involves translating between languages
writing style	2	measures model's ability to examine a text's writing style rather than its semantic meaning
grammar	2	tests model's ability to handle particular grammatical phenomena in the input or in the desired output

Many Large Scale PreTrained Language Model

- Basics (GPT, BERT, T5)
- PaLM
  - (decoder-only trained with next-token prediction)
- BLOOM
  - BLOOM is essentially similar to GPT3 (auto-regressive model for next token prediction), but has been trained on 46 different languages and 13 programming languages.
- Flan-PaLM / Flan-T5
- Many many new recent LLMs on huggingface: Llama3, Mistral

Build a Large Language Model (From Scratch)
Book by Sebastian Raschka

[Submitted on 20 Oct 2022 (v1), last revised 21 Oct 2022 (this version, v2)]

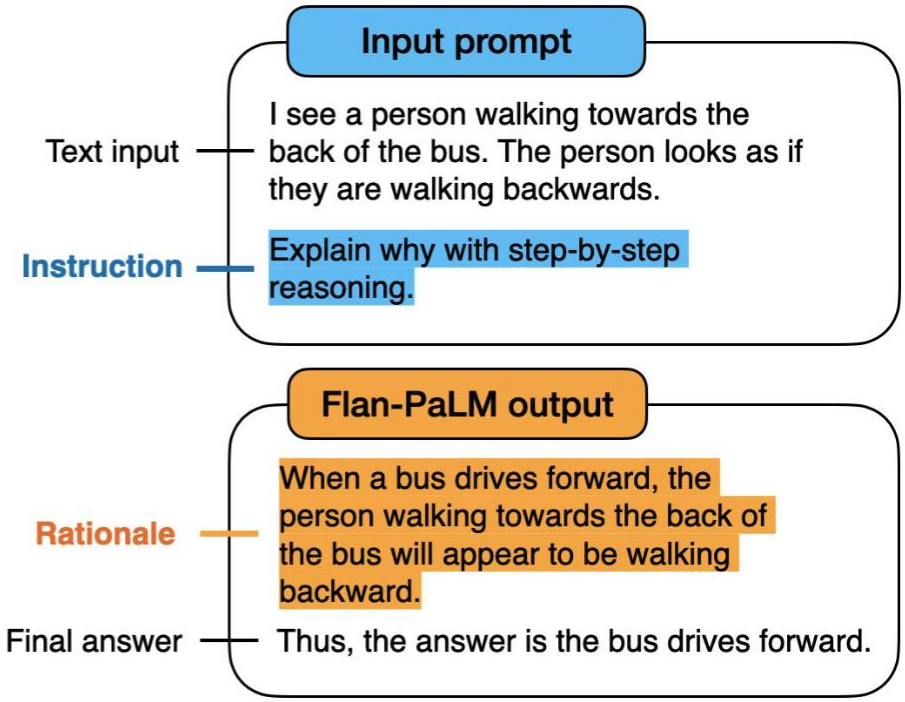
FLAN

Scaling Instruction-Finetuned Language Models

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, Jason Wei

Finetuning language models on a collection of datasets phrased as instructions has been shown to improve model performance and generalization to unseen tasks. In this paper we explore instruction finetuning with a particular focus on (1) scaling the number of tasks, (2) scaling the model size, and (3) finetuning on chain-of-thought data. We find that instruction finetuning with the above aspects dramatically improves performance on a variety of model classes (PaLM, T5, U-PaLM), prompting setups (zero-shot, few-shot, CoT), and evaluation benchmarks (MMLU, BBH, TyDiQA, MGSM, open-ended generation). For instance, Flan-PaLM 540B instruction-finetuned on 1.8K tasks outperforms PALM 540B by a large margin (+9.4% on average). Flan-PaLM 540B achieves state-of-the-art performance on several benchmarks, such as 75.2% on five-shot MMLU. We also publicly release Flan-T5 checkpoints, which achieve strong few-shot performance even compared to much larger models, such as PaLM 62B. Overall, instruction finetuning is a general method for improving the performance and usability of pretrained language models.

Chain-of-thought prompting is highly effective but having to write few-shot exemplars can be tedious and zero-shot CoT doesn't always work well. Our **CoT finetuning** significantly improves zero-shot reasoning abilities, such as on commonsense reasoning.



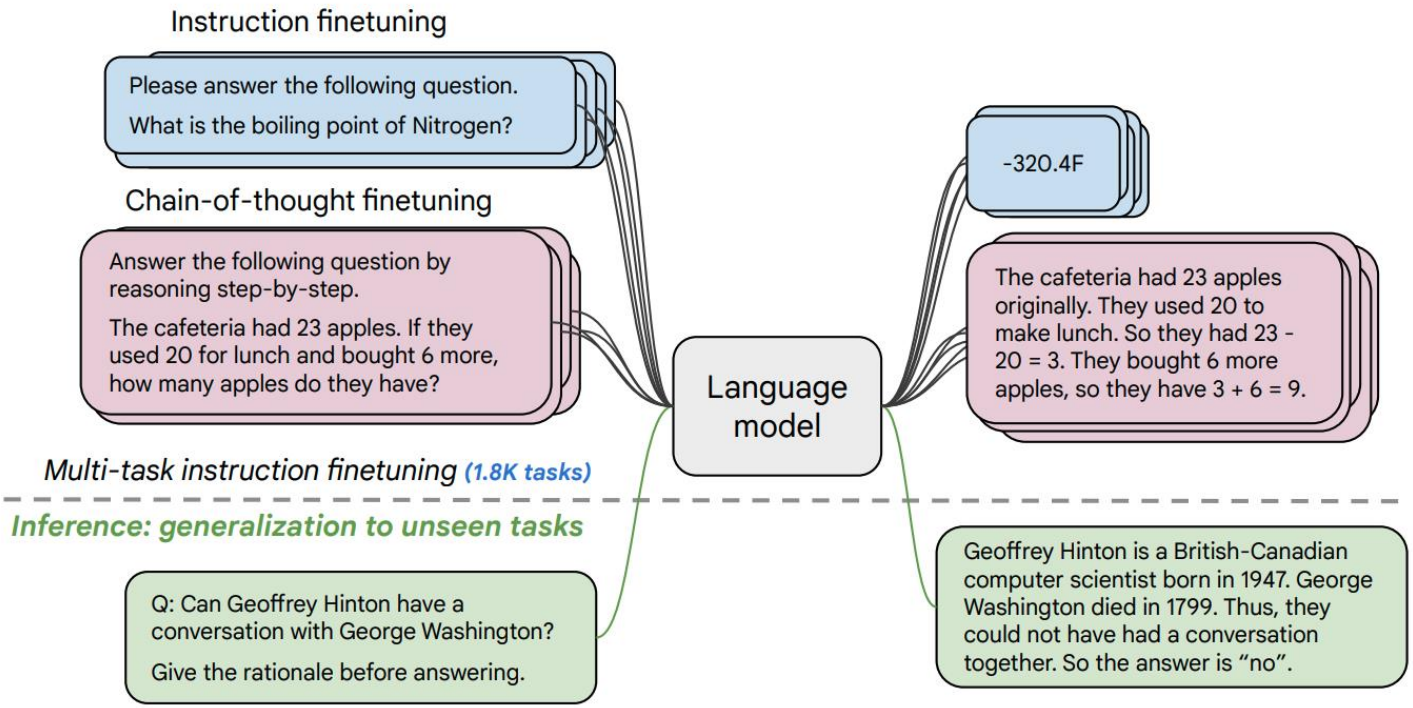
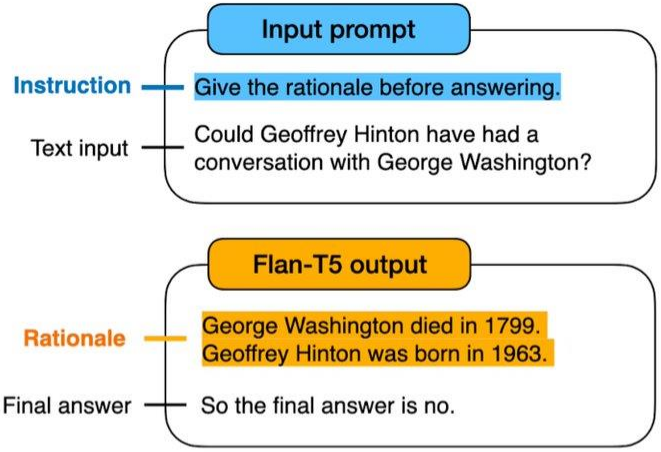
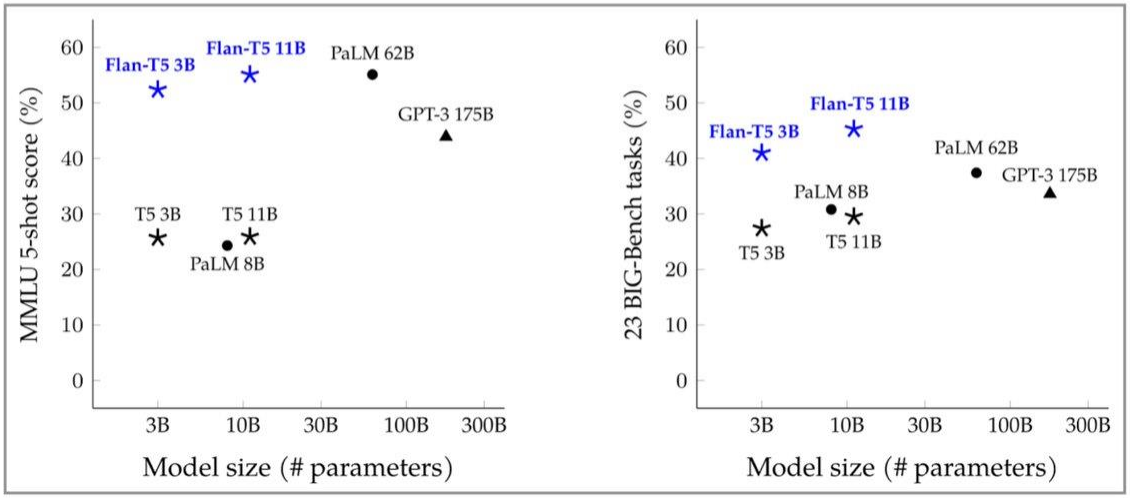
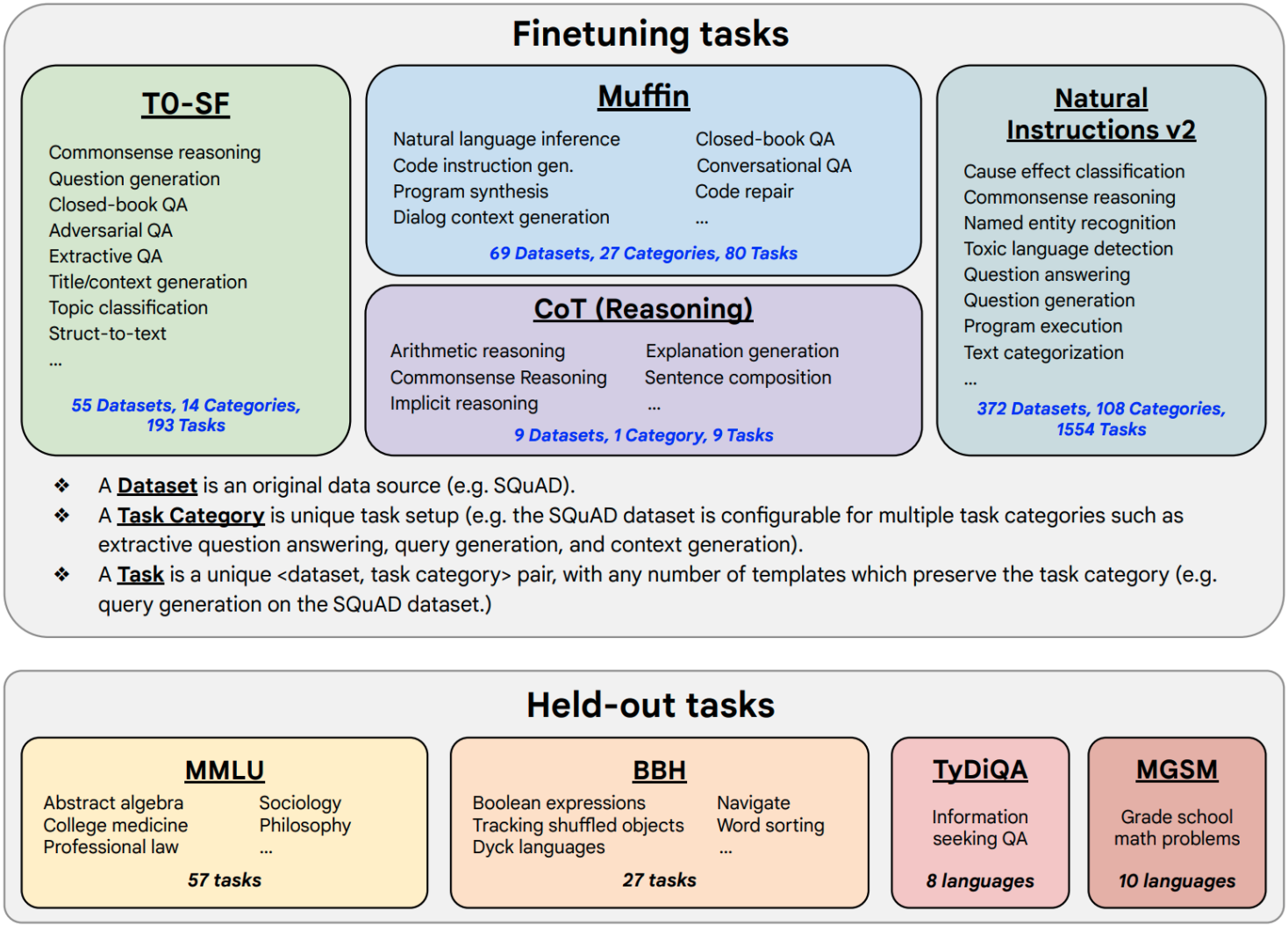


Figure 1: We finetune various language models on 1.8K tasks phrased as instructions, and evaluate them on unseen tasks. We finetune both with and without exemplars (i.e., zero-shot and few-shot) and with and without chain-of-thought, enabling generalization across a range of evaluation scenarios.

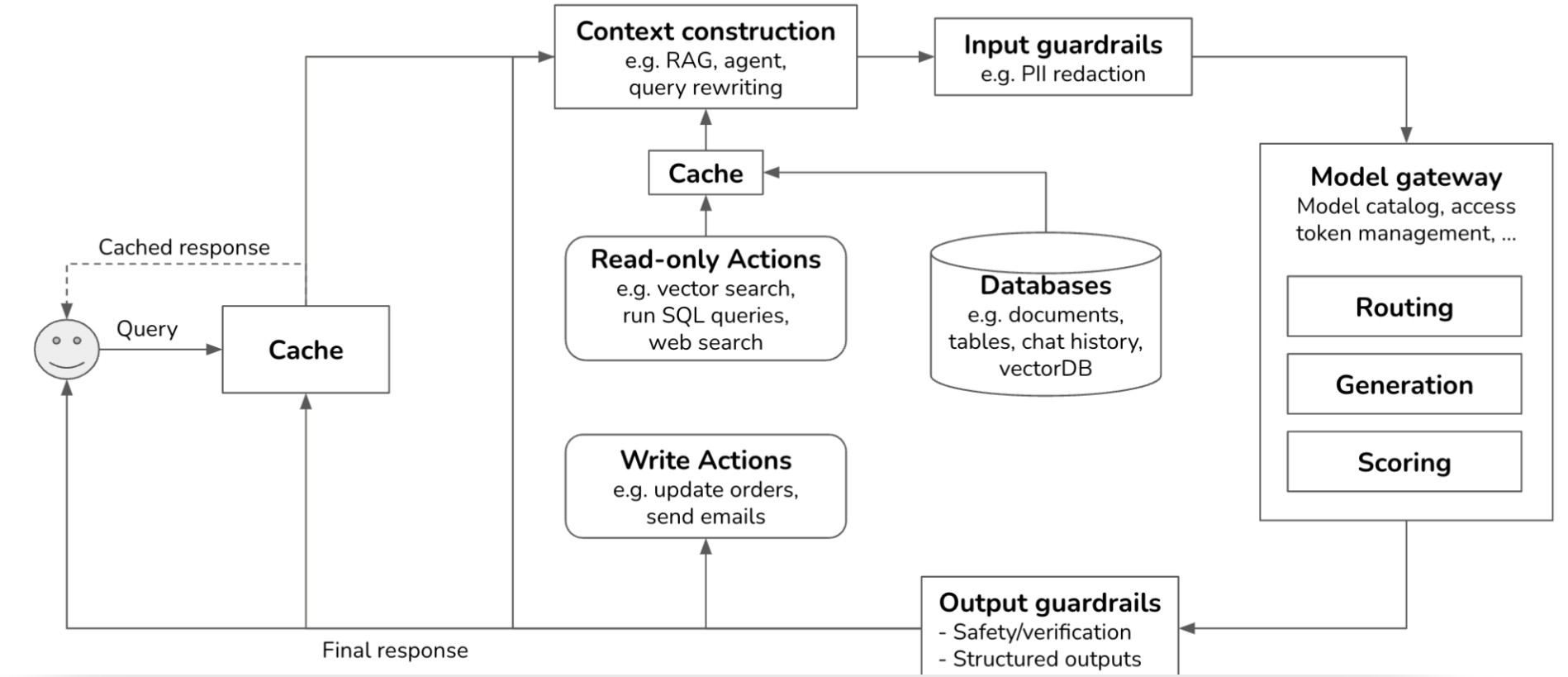

Flan-T5


Input prompt
Instruction
Give the rationale before answering.
Text input
Could Geoffrey Hinton have had a conversation with George Washington?
Flan-T5 output
Rationale
George Washington died in 1799. Geoffrey Hinton was born in 1963.
Final answer
So the final answer is no.


MMLU 5-shot score (%)
60
50
40
30
20
10
0
Flan-T5 3B
Flan-T5 11B
PaLM 62B
GPT-3 175B
T5 3B
T5 11B
PaLM 8B
3B
10B
30B
100B
300B
Model size (# parameters)
23 BIG-Bench tasks (%)
60
50
40
30
20
10
0
Flan-T5 11B
Flan-T5 3B
PaLM 62B
GPT-3 175B
PaLM 8B
T5 3B
T5 11B
3B
10B
30B
100B
300B
Model size (# parameters)


Finetuning tasks
T0-SF
Commonsense reasoning
Question generation
Closed-book QA
Adversarial QA
Extractive QA
Title/context generation
Topic classification
Struct-to-text
...
55 Datasets, 14 Categories, 193 Tasks
Muffin
Natural language inference
Code instruction gen.
Program synthesis
Dialog context generation
Closed-book QA
Conversational QA
Code repair
...
69 Datasets, 27 Categories, 80 Tasks
CoT (Reasoning)
Arithmetic reasoning
Commonsense Reasoning
Implicit reasoning
Explanation generation
Sentence composition
...
9 Datasets, 1 Category, 9 Tasks
Natural Instructions v2
Cause effect classification
Commonsense reasoning
Named entity recognition
Toxic language detection
Question answering
Question generation
Program execution
Text categorization
...
372 Datasets, 108 Categories, 1554 Tasks
A Dataset is an original data source (e.g. SQuAD).
A Task Category is unique task setup (e.g. the SQuAD dataset is configurable for multiple task categories such as extractive question answering, query generation, and context generation).
A Task is a unique <dataset, task category> pair, with any number of templates which preserve the task category (e.g. query generation on the SQuAD dataset.)
Held-out tasks
MMLU
Abstract algebra
College medicine
Professional law
Sociology
Philosophy
...
57 tasks
BBH
Boolean expressions
Tracking shuffled objects
Dyck languages
Navigate
Word sorting
...
27 tasks
TyDiQA
Information seeking QA
8 languages
MGSM
Grade school math problems
10 languages

Building FM based GenAI Platforms



https://huyenchip.com/2024/07/25/genai-platform.html#ai_pipeline_orchestration

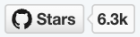
Data Pipelines

Name (site)	Description	Github	Pip Installs
Databricks	A unified data platform for building, deploying, and maintaining enterprise data solutions, including products (like MosaicML and MLflow) purpose-built for AI	Stars 40k	downloads 7.6M/week
Airflow	A data pipeline framework to programmatically author, schedule, and monitor data pipelines and workflows, including for LLMs	Stars 38k	downloads 5.6M/week
Unstructured.io	Open-source components for pre-processing documents such as PDFs, HTML, and Word documents for usage with LLM apps	Stars 9.8k	downloads 492k/week
Fivetran	A platform that extracts, loads, and transforms data from various sources for analytics, AI, and operations	N/A	downloads 262/week
Airbyte	An open-source data integration engine that helps consolidate data in data warehouses, lakes, and databases	Stars 17k	downloads 123K/week
Anyscale	An AI compute platform that allows developers to scale data ingest, preprocessing, embedding, and inference computations using Ray	Stars 35k	downloads 1.4M/week
Alluxio	An open-source data platform at the intersection of compute and storage, bringing data closer to compute, to accelerate model training and serving, boost GPU utilization, and reduce costs for AI workloads	Stars 6.9k	downloads 64/week

Vector Databases

Name (site)	Description	Github	Pip Installs
Pinecone	A managed, cloud-native vector database with a simple API for high-performance AI applications	n/a	downloads 728k/week
Weaviate	An open-source vector database that stores both objects and vectors	Stars 12k	downloads 1.6M/week
ChromaDB	An AI-native, open-source embedding database platform for developers	Stars 17k	downloads 578k/week
Pgvector	An open-source vector similarity search for Postgres, allowing for exact and approximate nearest-neighbor search	Stars 14k	downloads 949k/week
Zilliz (Milvus)	An open-source vector database, built for developing and maintaining AI applications	Stars 32k	downloads 270k/week
Qdrant	A vector database and vector similarity search engine	Stars 21k	downloads 791k/week
Metal io	A managed service for developers to build applications with ML embeddings	N/A	downloads 2.7k/week
LanceDB	A serverless vector database for AI applications	Stars 5.3k	downloads 173k/week

Playgrounds

Name (site)	Description	Github	Pip Installs
OpenAI Playground	A web-based platform for experimenting with various machine-learning models developed by OpenAI	N/A	N/A
nat.dev	A platform that allows users to test prompts with multiple language models and compare their performance		
Humanloop	A platform that helps developers build applications on top of LLMs		
Parea AI	Platform and SDK for AI Engineers providing tools for LLM evaluation, observability, and a version-controlled enhanced prompt playground.		

Orchestrators

Name (site)	Description	Github	Pip Installs
Langchain	An open-source library that gives developers the tools to build applications powered by LLMs	Stars 99k	downloads 7.1M/week
LlamaIndex	A data framework for LLM applications to ingest, structure, and access private or domain-specific data	Stars 38k	downloads 810k/week
Autogen	A framework for automating and streamlining LLM workflows using customizable, conversable agents for complex AI applications	Stars 38k	downloads 105k/week
Microsoft Semantic Kernel	A lightweight open-source orchestration SDK	Stars 23k	downloads 23k/week
Haystack	LLM orchestration framework to build customizable, production-ready LLM applications	Stars 19k	downloads 20k/week
Vercel AI SDK	An open-source library for developers to build streaming UIs in JavaScript and TypeScript	Stars 11k	downloads 887k/week (node/npm)
Vectara AI	A search and discovery platform for AI conversations utilizing your own data	Stars 155	N/A
ChatGPT	An AI chatbot that uses natural language processing to create humanlike conversational dialogue	N/A	N/A

APIs / Plugins

Name (site)	Description	Github	Pip Installs
Serp API	A real-time API to access Google search results, as well as handling proxies, solving captchas, and parsing structured data		
Wolfram Alpha API	A web-based API providing computational and presentation capabilities for integration into various applications	N/A	
Zapier API AI Plugin	A plugin that allows you to connect 5,000+ apps and interact with them directly inside ChatGPT	N/A	N/A

LLM Caches

Name (site)	Description	Github	Pip Installs
Redis	An in-memory data structure store used as a database, cache, message broker, and streaming engine	Stars 68k	downloads 13M/week
SQLite	A self-contained, serverless, zero-configuration, transactional SQL database engine	Stars 7.2k	downloads 72k/week
GPTCache	An open-source tool for improving the efficiency and speed of GPT-based applications by implementing a cache to store the responses	Stars 7.3k	N/A

^ Back to Contents ^

Logging / Monitoring / Eval

Name (site)	Description	Github	Pip Installs
Braintrust Data	An AI product stack featuring evaluations, prompt playgrounds, continuous integration, dataset management, and access to various AI models through a single API	Stars 296	downloads 81k/week
Arize AI	An observability platform for both LLMs and supervised ML	Stars 5k	downloads 53k/week
Weights & Biases	An MLOps platform for streamlining ML workflows	Stars 9.4k	downloads 3.8M/week
MLflow	A platform to streamline ML development	Stars 19k	downloads 3.2M/week
PromptLayer	A platform for tracking, managing, and sharing LLM prompt engineering	Stars 541	downloads 7.7k/week

Validators

Name (site)	Description	Github	Pip Installs
Guardrails AI	An open-source Python package for specifying structure and type, validating, and correcting the outputs of LLMs	Stars 4.4k	downloads 9.2k/week
Rebuff	An open-source framework designed to detect and protect against prompt injection attacks in LLM apps	Stars 1.2k	downloads 547/week
Microsoft Guidance	A guidance language for controlling LLMs, providing a syntax for architecting LLM workflows	Stars 19k	downloads 9.2k/week
LMQL	An open-source programming language and platform for language model interaction	Stars 3.8k	downloads 897/week
Outlines	A tool for helping developers guide text generation to build robust interfaces with external systems and guarantee that outputs match a regex or JSON schema	Stars 10k	downloads 416k/week
LLM Guard	An open-source, comprehensive tool designed to fortify the security of Large Language Models (LLMs).	Stars 1.4k	downloads 12k/week

https://arxiv.org/pdf/2412.17686

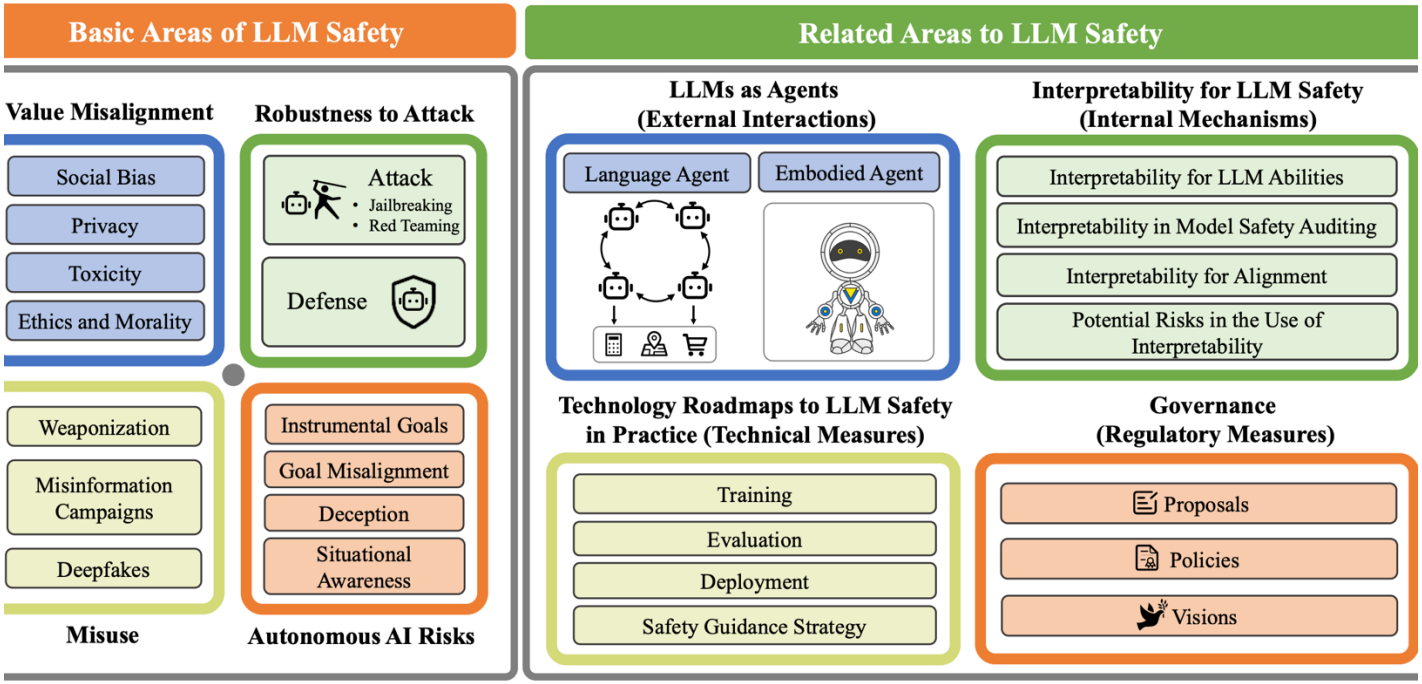
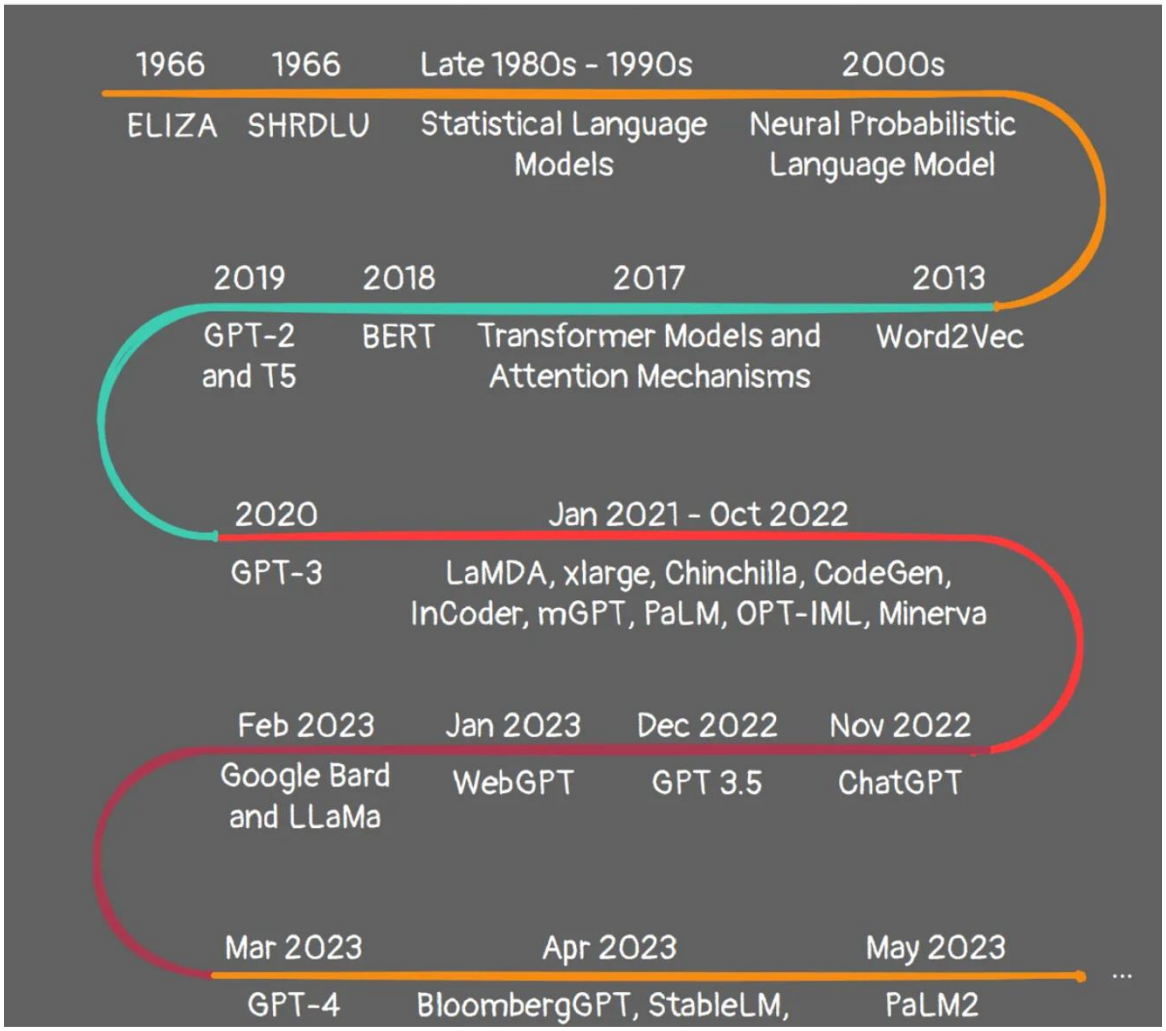


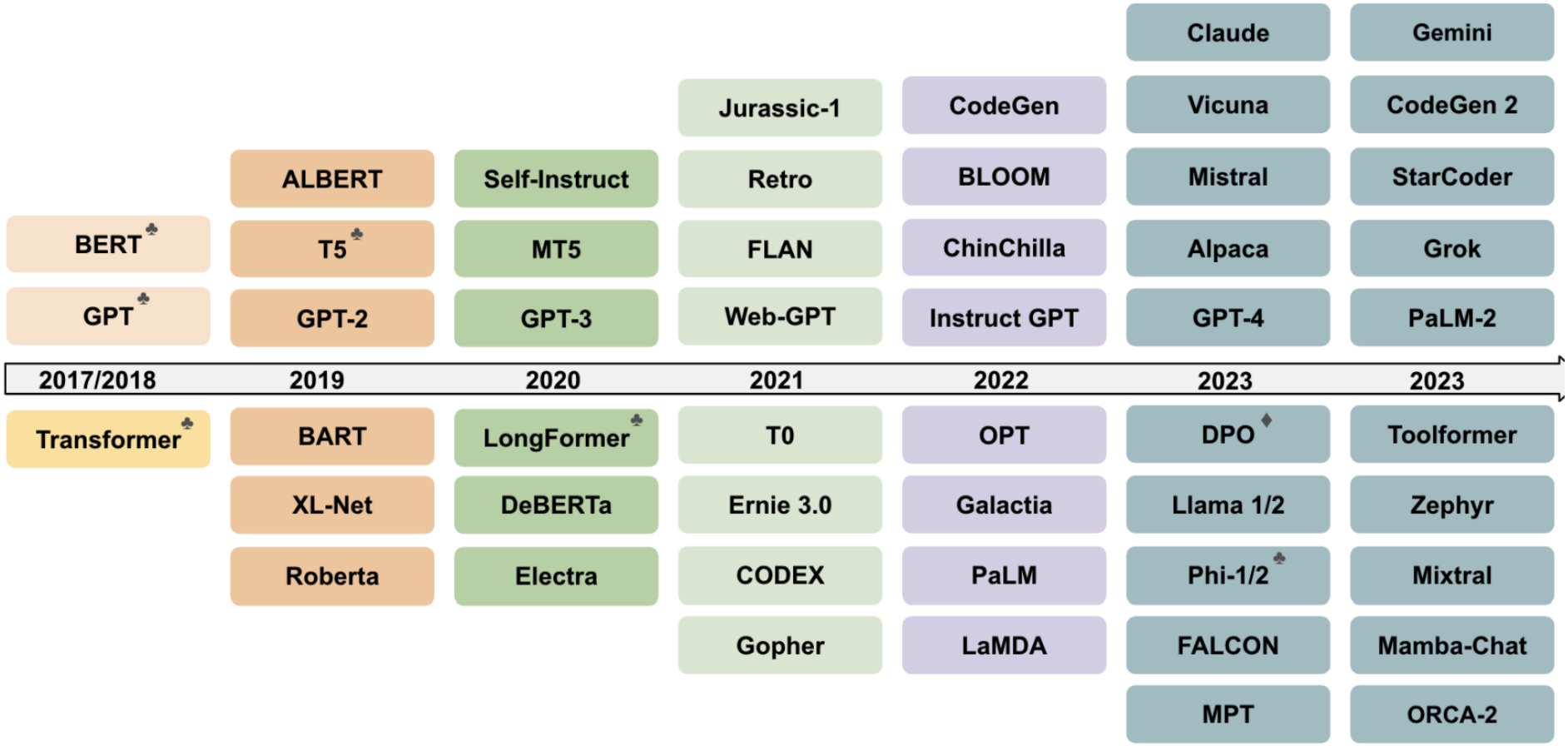
Figure 1: The taxonomy of LLM safety proposed in this survey.

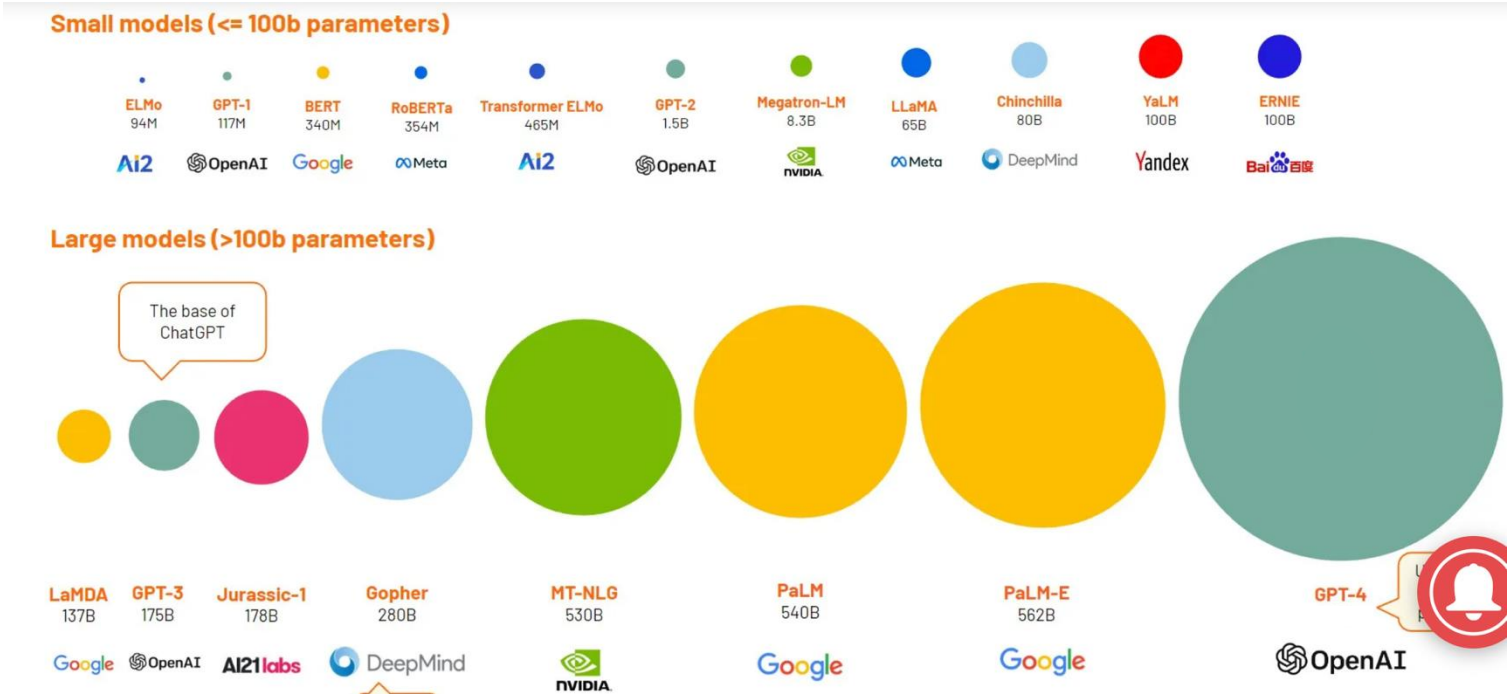
Many new LLMs in 2022-2023 ->2024 -> now



https://levelup.gitconnected.com/the-brief-history-of-large-language-models-a-journey-from-eliza-to-gpt-4-and-google-bard-167c614af5af

Timeline of some of the most representative LLM frameworks





https://levelup.gitconnected.com/the-brief-history-of-large-language-models-a-journey-from-eliza-to-gpt-4-and-google-bard-167c614af5af

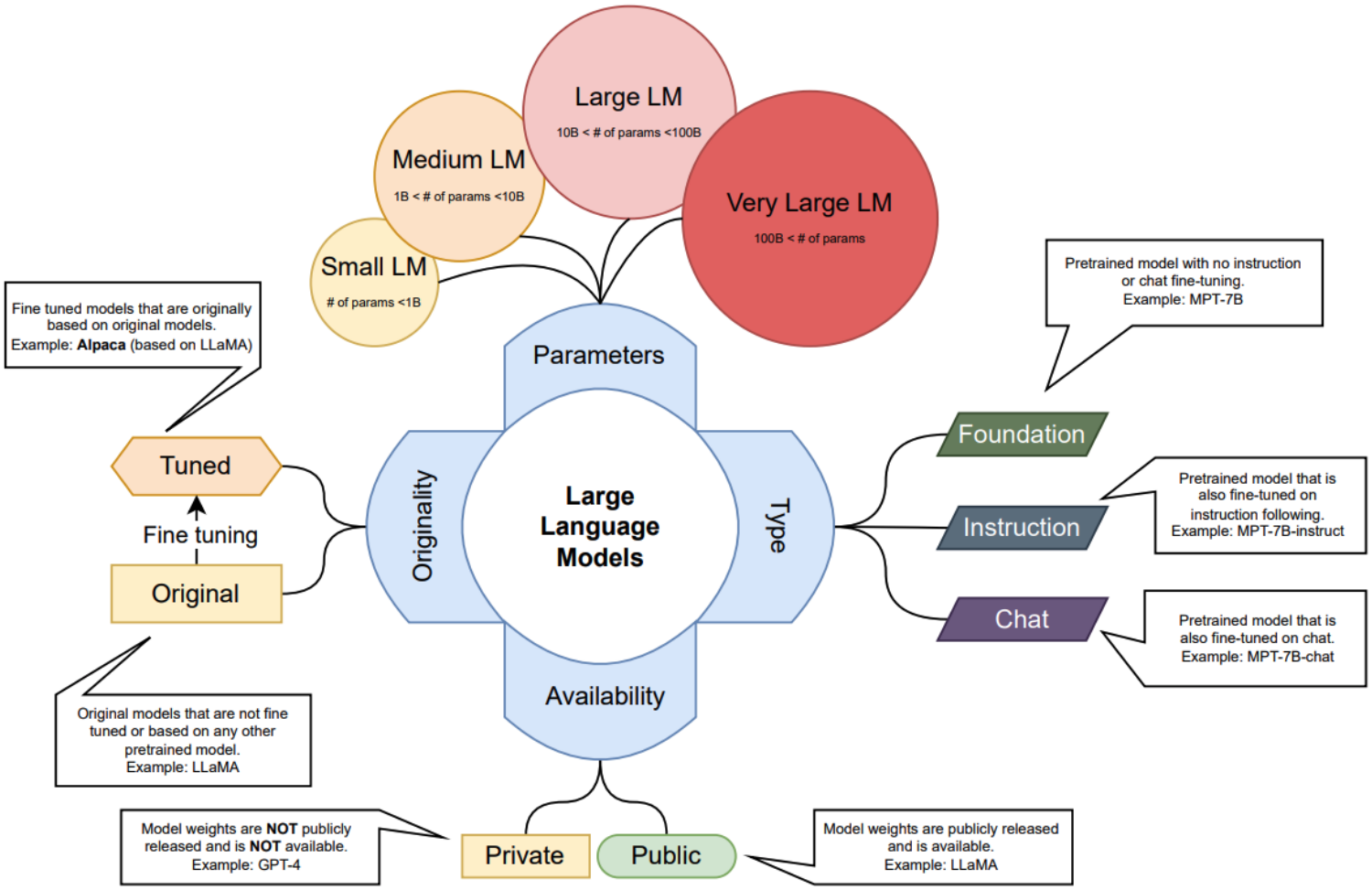
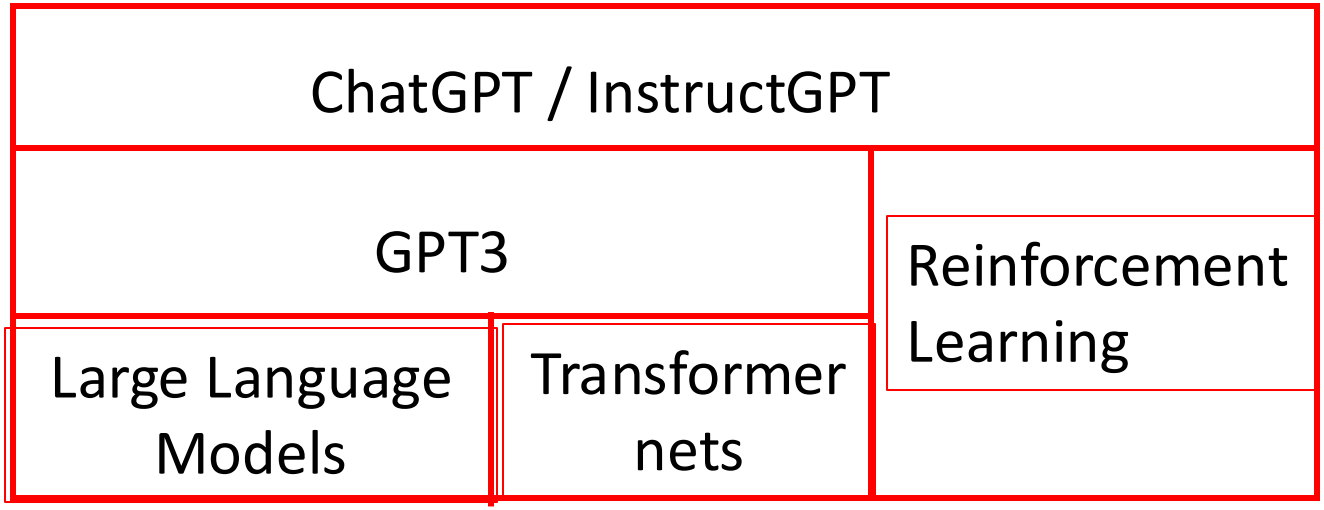


Fig. 43: LLM categorizations.

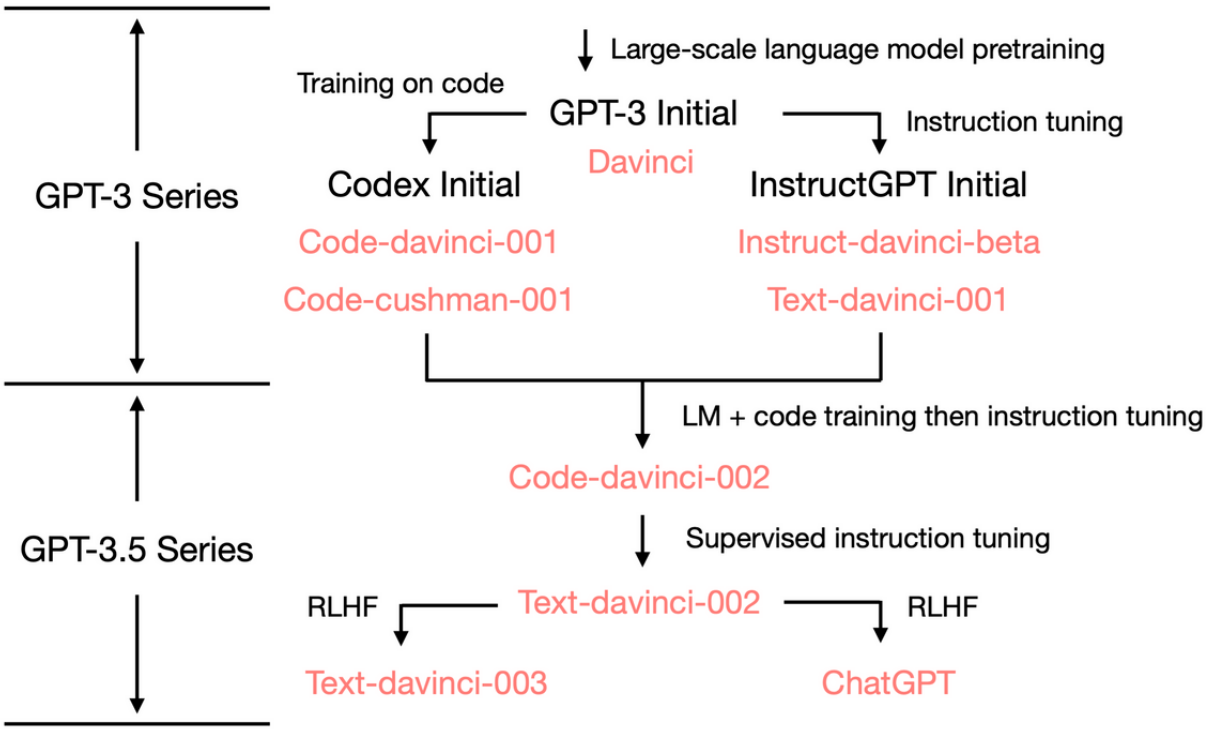
ChatGPT: Optimizing Language Models for Dialogue” by OpenAI

- No paper / Just a blog / Released Nov 30 2022
- It took 5 days to reach 1M users

Concepts that ChatGPT builds on



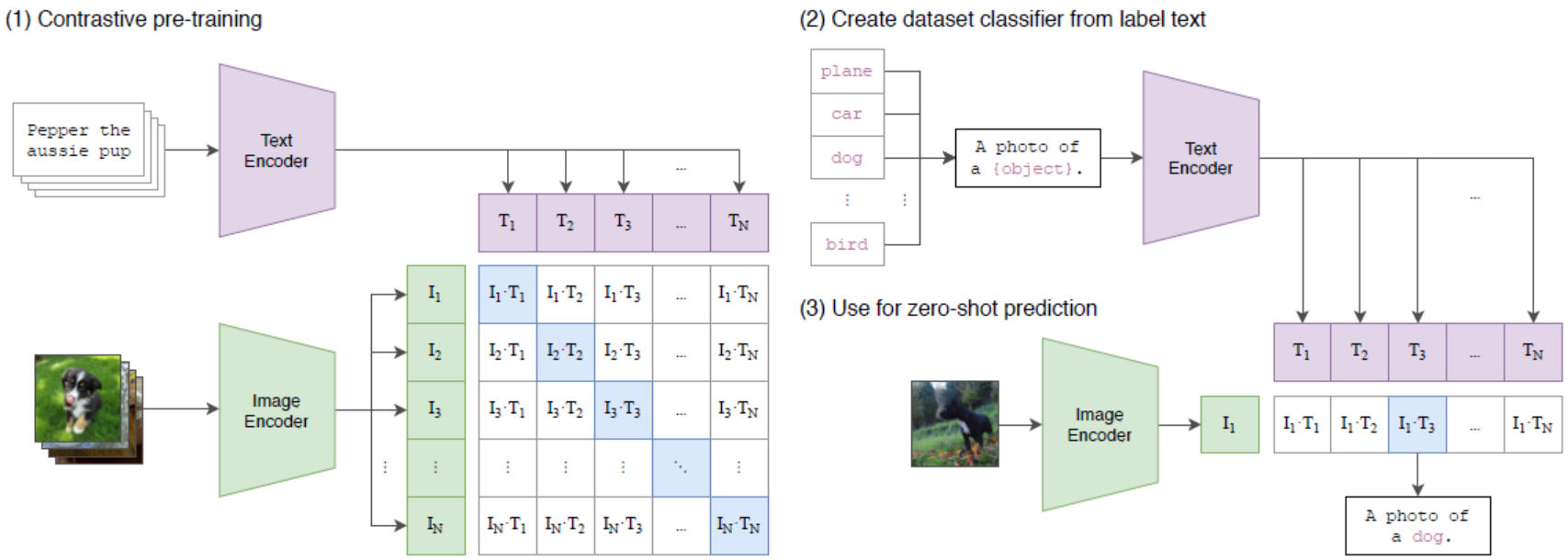
Family of GPT-3.5



Blog: [How does GPT Obtain its Ability? Tracing Emergent Abilities of Language Models to their Sources]

CLIP: **C**ontrastive **L**anguage-**I**mage **P**retraining for Vision

CLIP: Learning Transferrable Visual Models from Natural Language Supervision (Radford et al. 2021)

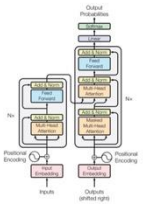


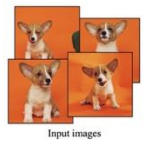
use a text encoder as a classifier

Related: Stable diffusion (Proposed 2022-08)

- High-Resolution Image Synthesis with Latent Diffusion Models
  CVPR'22 Rombach et al.
- Code and models released Open Source
- The CreativeML OpenRAIL M license
- by Stability AI and Runway

History of Image Generation





GAN

Style-GAN

DALL-E 2

Imagen

DALL-E3

2012 2014 2015 2017 2018 2020 2021 2021 2022 2022 2022 2022 2023

ImageNet w. CNN

DCGAN

Transformer

ViT (VQGAN)

DALL-E

GLIDE

Stable Diffusion

DreamBooth

Blog: [Ten Years of Image Synthesis]