STEVENS
INSTITUTE of TECHNOLOGY
THE INNOVATION UNIVERSITY®

CS 584 Natural Language Processing

Large Language Models, Prompting

Ping Wang
Department of Computer Science
Stevens Institute of Technology

Reminder

- Homework 4: Dec 6
- Project Presentation: Dec 4 and Dec 11
- Final report and codes submission: Dec 13

Today's Lecture

- Large language models (LLMs)
- Prompting
- Instruction finetuning
- Limitations of LLMs
- Evaluation, interpretability, ethics

Larger and larger models

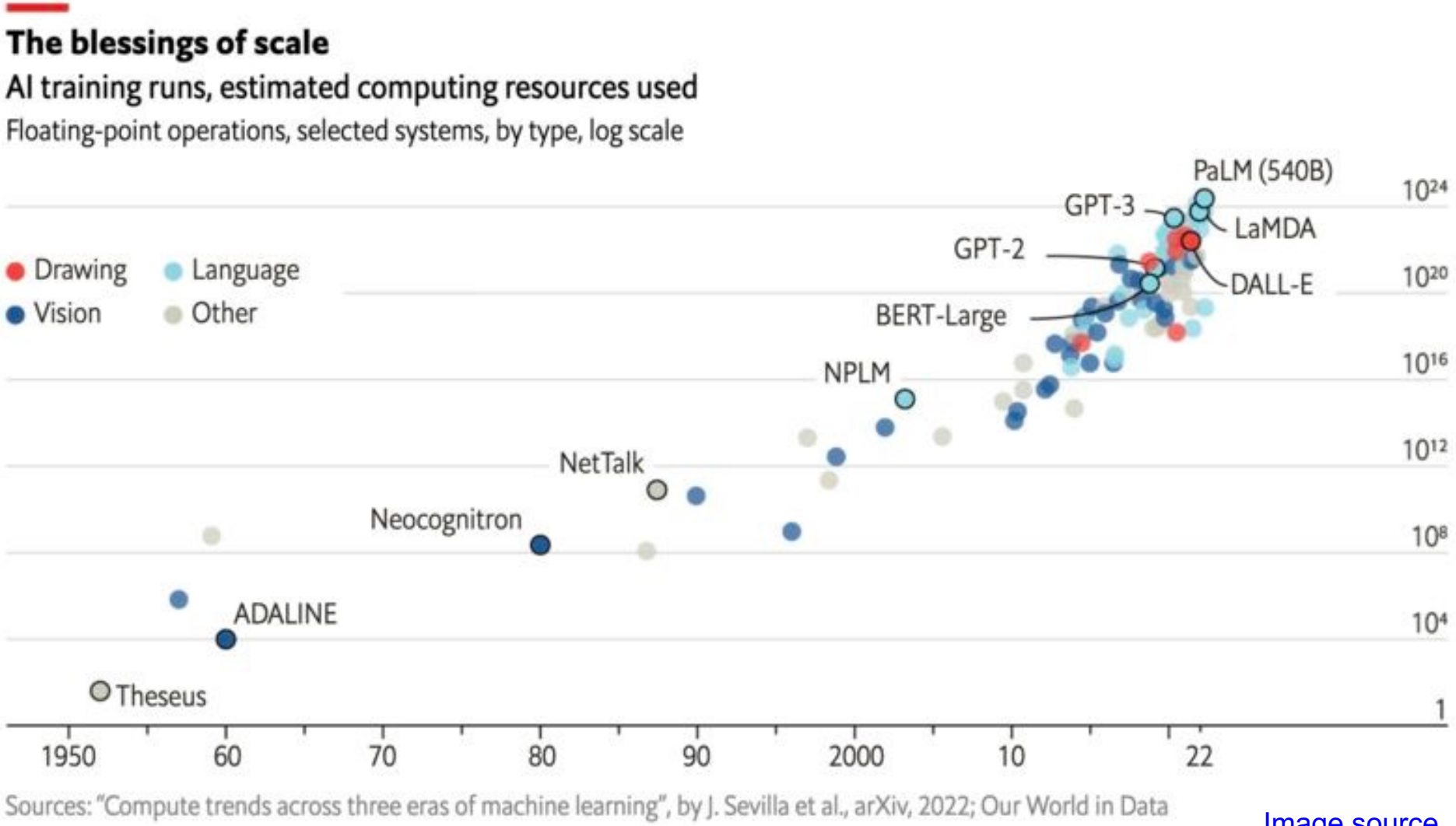
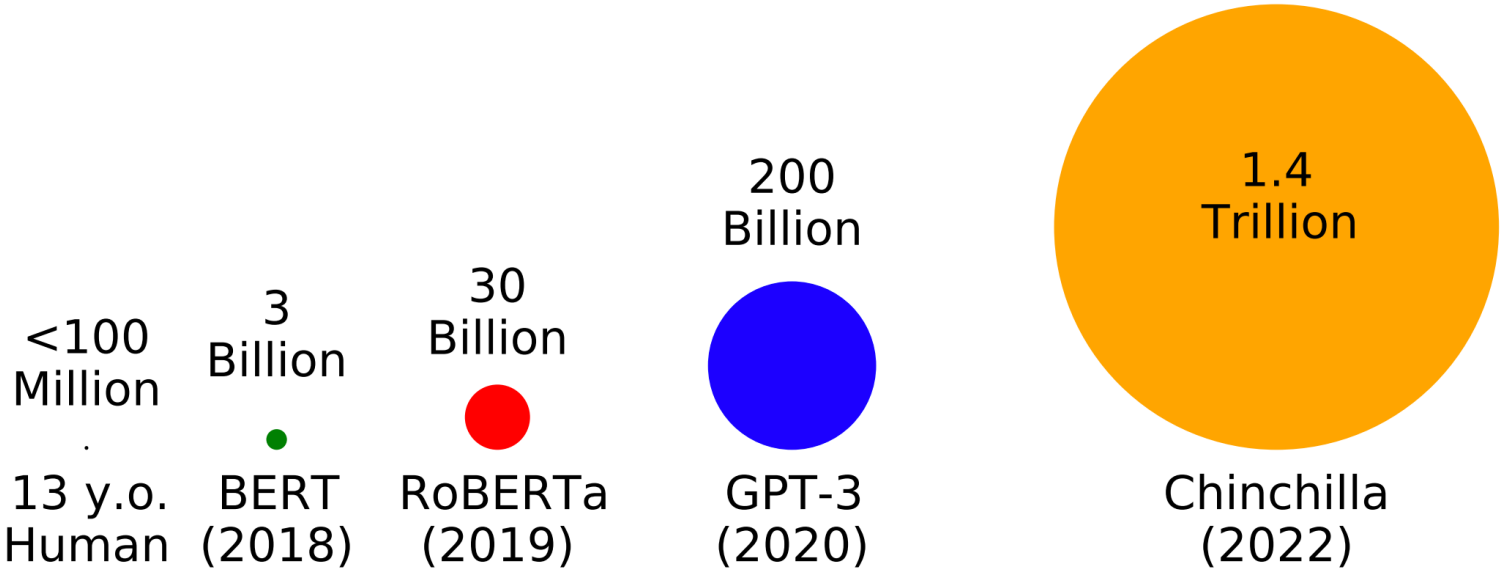


Image source

Trained on more and more data

tokens seen during training

- Huge effort has been put into optimizing LM pretraining at massive scales in the last several years.
- Datasets have also grown by orders of magnitude.



https://babylm.github.io/

Why Pretrained LM and Large LM?

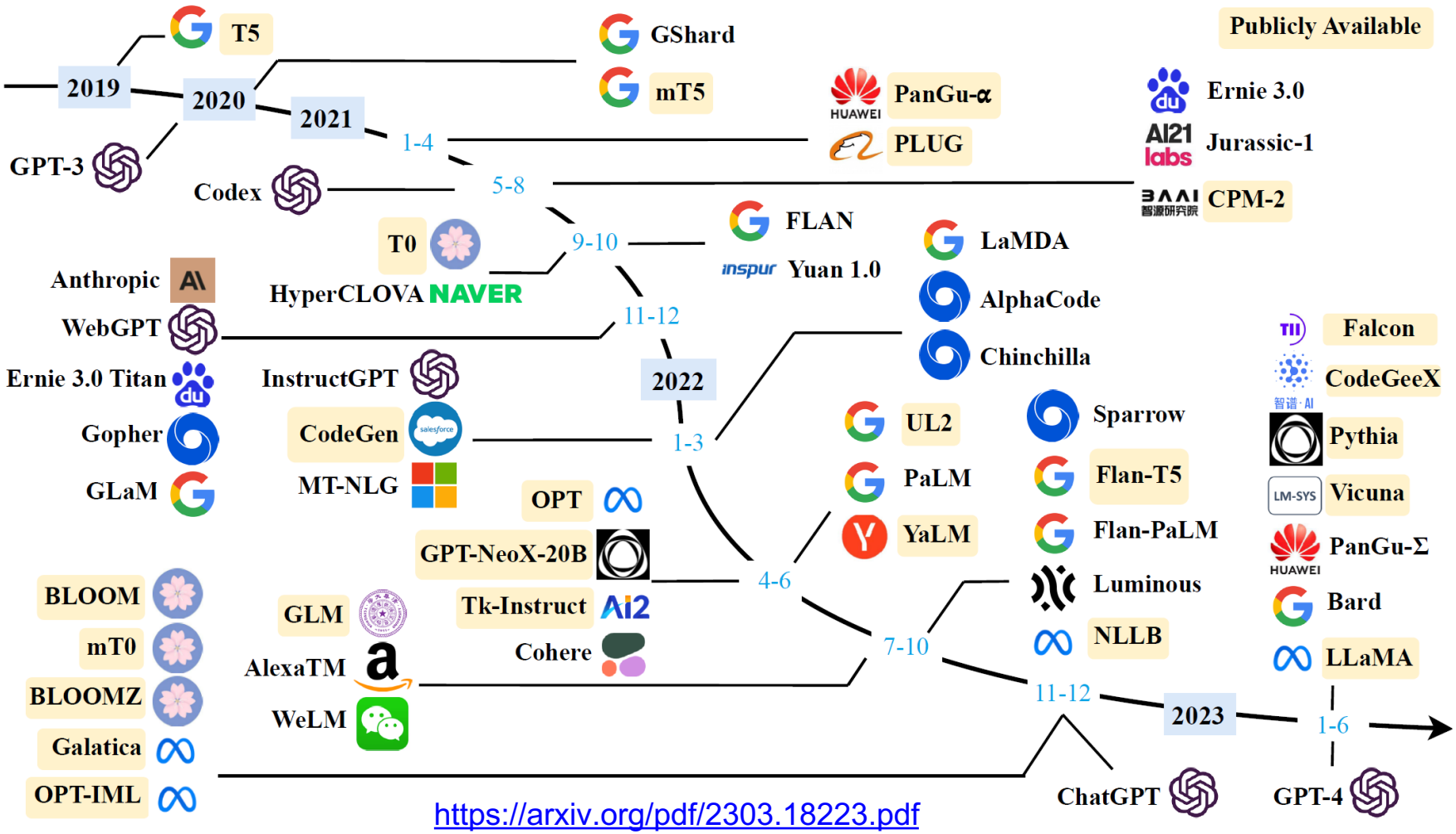
- Largely raised the performance bar of various NLP tasks.
- Pretrained context aware word representations are very effective as general-purpose semantic features.
- The abilities that are not present in small models but arise in large models.
- Leads to rethinking the possibilities of Artificial General Intelligence (AGI).
    - The ability to understand, learn, and apply knowledge in a way that is comparable to the cognitive capabilities of human beings.
    - As of now, AGI remains theoretical, and researchers continue to explore various approaches and methodologies to advance AI systems toward this ambitious goal.

What kinds of things does pretraining learn?

- Syntax
- Semantics
- Common sense
- Sentiment
- Reasoning
- Basic arithmetic
- And others …

Timeline of existing large language models

Mainly according to the release date of the technical paper for a model.



https://arxiv.org/pdf/2303.18223.pdf

Brief illustration for the technical evolution of GPT-series models

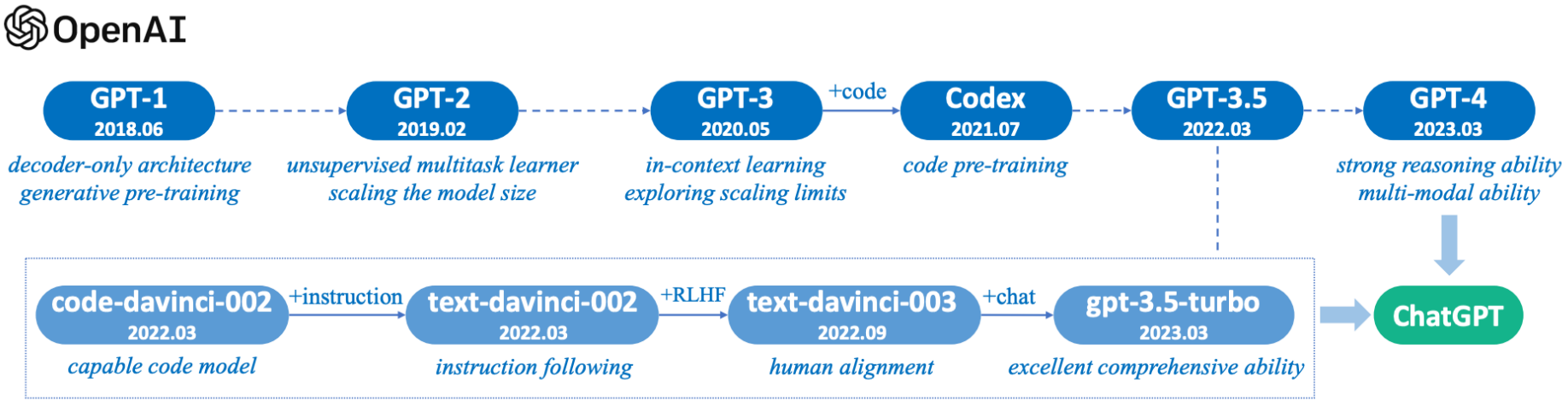
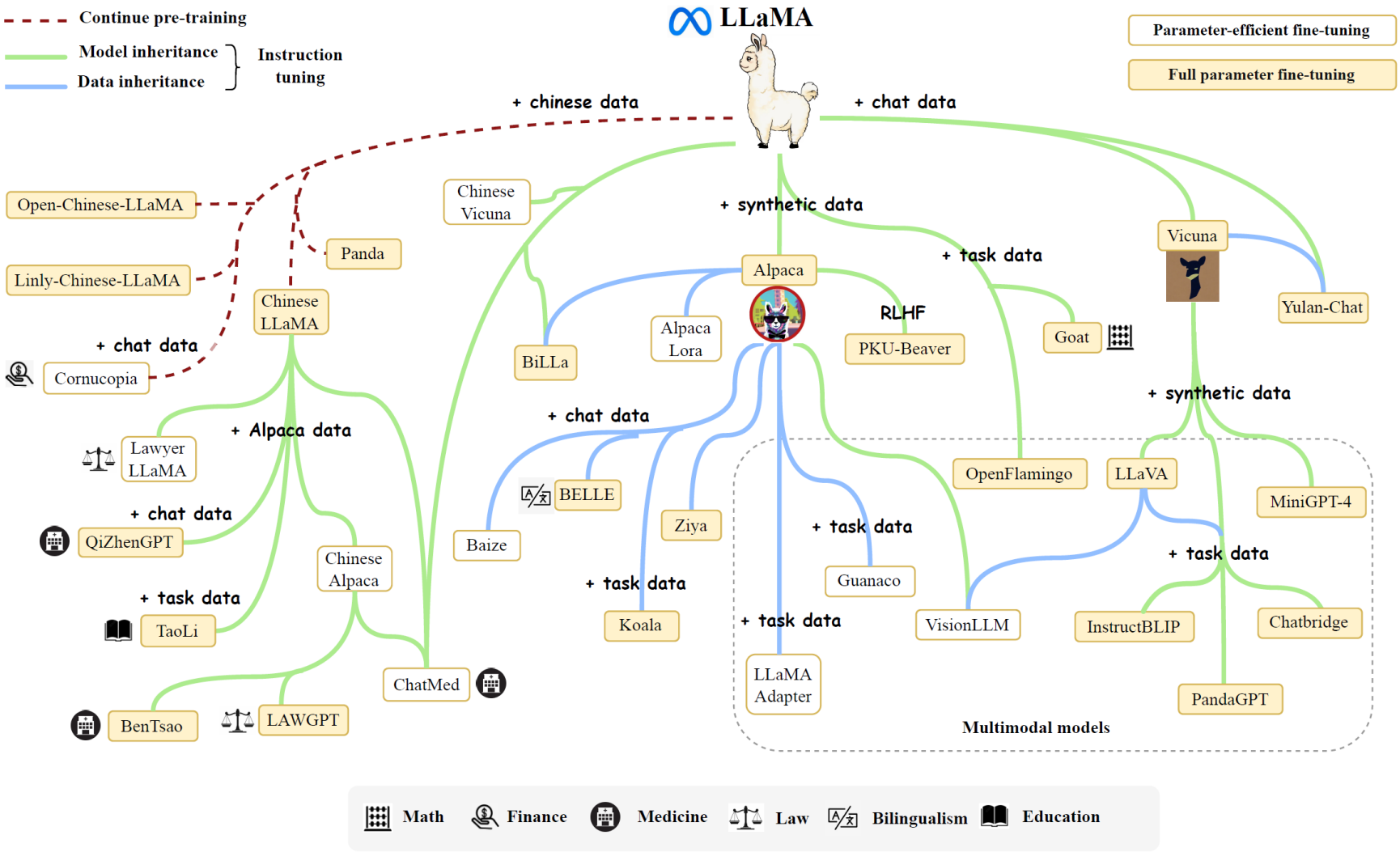


Fig. 3: A brief illustration for the technical evolution of GPT-series models. We plot this figure mainly based on the papers, blog articles and official APIs from OpenAI. Here, *solid lines* denote that there exists an explicit evidence (*e.g.*, the official statement that a new model is developed based on a base model) on the evolution path between two models, while *dashed lines* denote a relatively weaker evolution relation.

https://arxiv.org/pdf/2303.18223.pdf

Evolutionary of research work on LLaMA



https://arxiv.org/pdf/2303.18223.pdf

Various Aspects in LLMs

- **Model architecture:**
  - Encoder-only, Decoder-only, Encoder-Decoder
- **Layers components:**
  - Tokenization, Normalization, Activation, Attention, …
- **Model training:**
  - Data preprocessing, Batch training, Learning rate, Optimizer, Human alignment, Scalable training, …

Pre-training Tasks

- **Language Modeling (LM):**
  - Predicting next token
- **Denoising Autoencoding (DAE):**
  - Predicting corrupted text
- **Mixture-of-Denoisers:**
  - LM, DAE with short span, DAE with long span

Emergent zero-shot learning

- One key emergent ability in GPT-2 is zero-shot learning:
  - The ability to do many tasks with no examples, and no gradient updates.
- GPT-2 beats SoTA on language modeling benchmarks with no task-specific fine-tuning.
- You can get interesting zero-shot behavior if you're creative enough with how you specify your task!

Heart of the LLMs: In-context learning

The model learns about the context based on the examples provided.

- Very large language models seem to perform some kind of learning without gradient steps simply from examples you provide within their contexts.
- Enabling GPT models to understand/comprehend and create text that closely resembles human speech, based on the instructions and examples they're provided.

- Three approaches to in-context learning:
  - **Few-shot**: includes several examples in the prompt to demonstrate the expected answer format and content.
  - **One-shot**: only a single example is provided in the prompt to demonstrate the desired task.
  - **Zero-shot**: no examples are provided to the model during the generation call. Instead, only the task or request is given as input.

https://vitalflux.com/in-context-learning-gpt-3-concepts-examples/

New methods of “prompting” LMs

The three settings we explore for in-context learning

Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

```
1  Translate English to French:        ← task description
2  cheese =>                           ← prompt
```

One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

```
1  Translate English to French:        ← task description
2  sea otter => loutre de mer          ← example
3  cheese =>                           ← prompt
```

Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

```
1  Translate English to French:        ← task description
2  sea otter => loutre de mer          ← examples
3  peppermint => menthe poivrée
4  plush girafe => girafe peluche
5  cheese =>                           ← prompt
```

Traditional fine-tuning (not used for GPT-3)

Fine-tuning

The model is trained via repeated gradient updates using a large corpus of example tasks.

```
sea otter => loutre de mer             ← example #1
        ↓
   gradient update
        ↓
peppermint => menthe poivrée           ← example #2
        ↓
   gradient update
        ↓
        ...
        ↓
plush giraffe => girafe peluche        ← example #N
        ↓
   gradient update
        ↓
cheese =>                              ← prompt
```

Brown et al., 2020)

Emergent ability on Machine Translation

Zero-shot

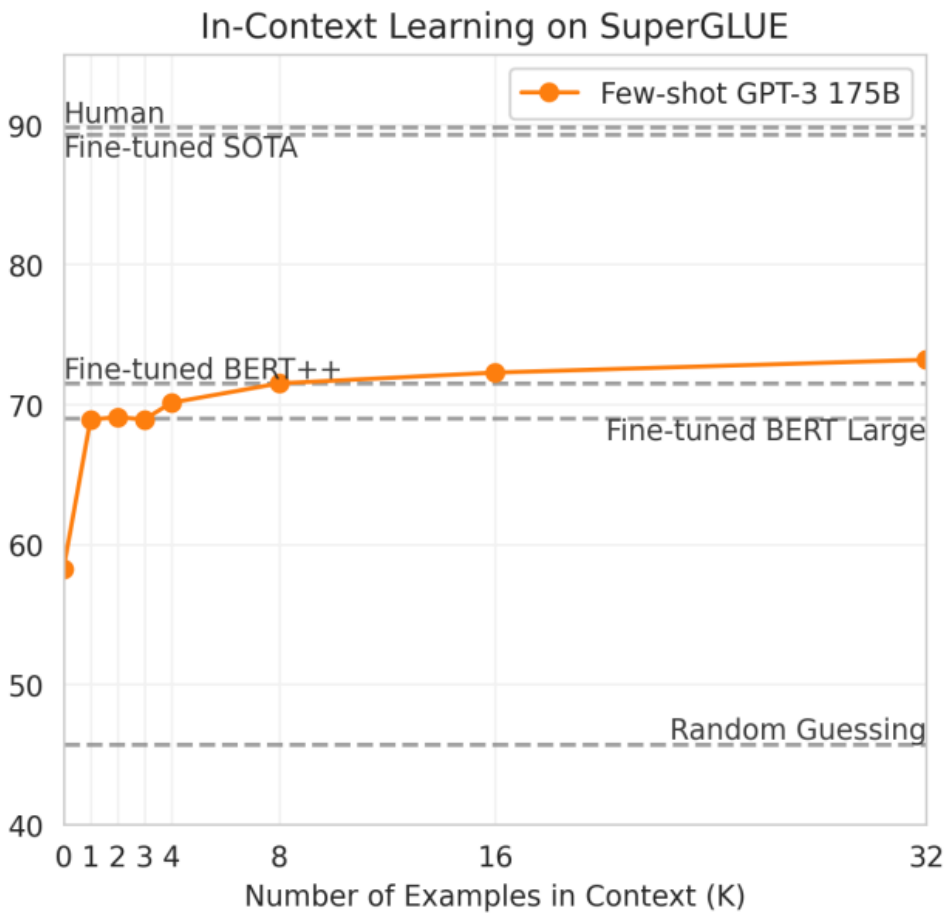
```
1  Translate English to French:
2  cheese =>
```

One-shot

```
1  Translate English to French:
2  sea otter => loutre de mer
3  cheese =>
```

Few-shot

```
1  Translate English to French:
2  sea otter => loutre de mer
3  peppermint => menthe poivrée
4  plush girafe => girafe peluche
5  cheese =>
```



Limits of prompting for harder tasks?

- Some tasks seem too hard for even large LMs to learn through prompting alone.

- Especially tasks involving richer, multi-step reasoning.
- (Humans struggle at these tasks too!)
  - 19583 + 29534 = 49117
  - 98394 + 49384 = 147778
  - 29382 + 12347 = 41729
  - 93847 + 39299 = ?

- Solution: change the prompt!

Prompt Optimization

- A number of methods exist for searching over prompts.
- Most of these do not lead to dramatically better results than doing some manual engineering (and they may be computationally intensive).
- Nevertheless, the choice of prompt is very important in general for zero-shot settings!

Four paradigms in NLP

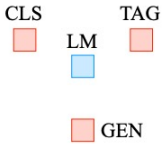
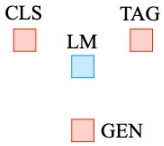
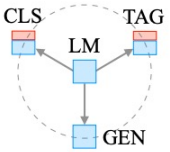
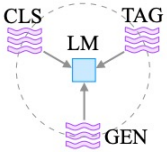
Paradigm	Engineering	Task Relation
a. Fully Supervised Learning (Non-Neural Network)	Features (e.g. word identity, part-of-speech, sentence length)	
b. Fully Supervised Learning (Neural Network)	Architecture (e.g. convolutional, recurrent, self-attentional)	
c. Pre-train, Fine-tune	Objective (e.g. masked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

Table 1: Four paradigms in NLP. The "**engineering**" column represents the type of engineering to be done to build strong systems. The "**task relation**" column, shows the relationship between language models (LM) and other NLP tasks (CLS: classification, TAG: sequence tagging, GEN: text generation). : fully unsupervised training. : fully supervised training. : Supervised training combined with unsupervised training. indicates a textual prompt. Dashed lines suggest that different tasks can be connected by sharing parameters of pre-trained models. "LM→Task" represents *adapting LMs (objectives) to downstream tasks* while "Task→LM" denotes *adapting downstream tasks (formulations) to LMs*.

Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing

Prompt Types

- **Cloze prompts**: which fill in the blanks of a textual string
  - I love this movie. Overall it was a [Z] movie.
- **Prefix prompts**: which continue a string prefix.
  - I love this movie. Overall the movie is [Z].

- How to choose: depends on the task and and the model
  - Generation task, or tasks using a standard auto-regressive LM: prefix prompts tend to be more conductive, as they mesh well with the left-to-right nature of the model.
  - Masked LMs: cloze prompts are a good fit, as they very closely match the form of the pre-training task.
  - Full text reconstruction models are more versatile, and can be used with either cloze or prefix prompts.

Chain-of-thought (CoT) prompting

- Chain-of-thought uses natural language as a scaffold for "reasoning"
- Unifies several ideas:
  - For math: relies on the fact that LLMs can do single steps of arithmetic. Builds on that to do multistep problems.
  - For QA: many problems involve reasoning decompositions
    - E.g., What's the capital of the country where Aristotle lived?
    - country = "country where Aristotle lived"
    - return What's the capital of [country]
  - For other tasks: capture the kinds of behavior written in rationales
- Typically, CoT is a few-shot prompting technique, where the in-context examples now contain explanations.
- Answer is not generated in one go, but comes after an explanation that "talks through" the reasoning.

Chain-of-thought prompting

Standard Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. ✗

Chain-of-Thought Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

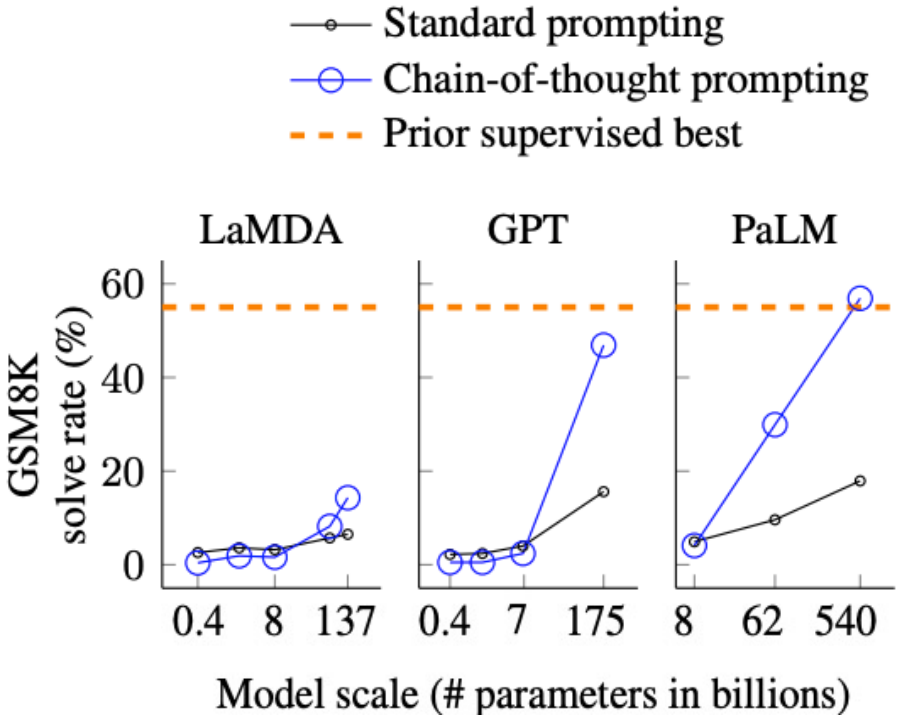
Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had 23 - 20 = 3. They bought 6 more apples, so they have 3 + 6 = 9. The answer is 9. ✓

[Wei et al., 2022; also see Nye et al., 2021]

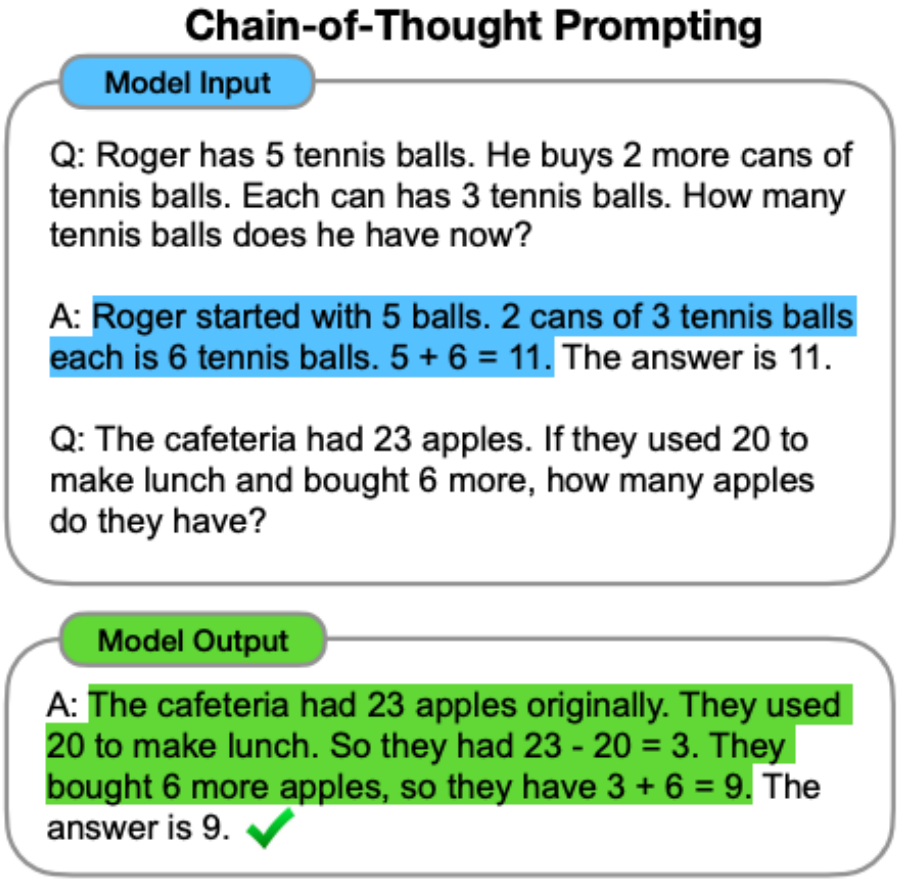
Chain-of-thought prompting performance

- Enables large language models to solve challenging middle school math word problems.
- Notably, chain-of-thought reasoning is an emergent ability of increasing model scale.



[Wei et al., 2022; also see Nye et al., 2021]

Chain-of-thought prompting

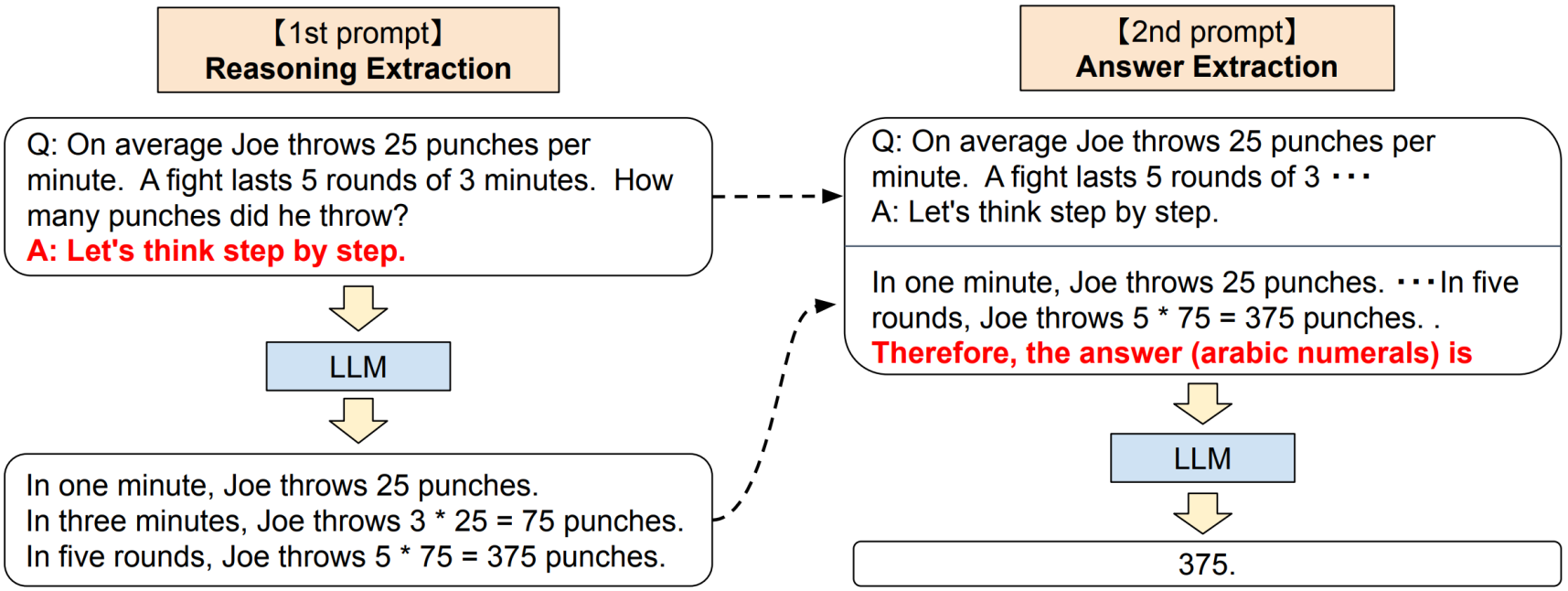


Do we even need examples of reasoning?

Can we just ask the model to reason through things?

[Wei et al., 2022; also see Nye et al., 2021]

Step-by-Step reasoning



Zero-shot CoT Prompting: step-by-step

Prompt for step-by-step reasoning: produces chains of thought without including demonstrations.

(a) Few-shot

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

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

(Output) The answer is 8. ✗

(b) Few-shot-CoT

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

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

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

(c) Zero-shot

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?
A: The answer (arabic numerals) is

(Output) 8 ✗

(d) Zero-shot-CoT (Ours)

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

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

Kojima et al. (2022)

Zero-shot COT Prompting: step-by-step

	MultiArith	GSM8K
Zero-Shot	**17.7**	**10.4**
Few-Shot (2 samples)	33.7	15.6
Few-Shot (8 samples)	33.8	15.6
Zero-Shot-CoT	**78.7**	**40.7**
Few-Shot-CoT (2 samples)	84.8	41.3
Few-Shot-CoT (4 samples : First) (*1)	89.2	-
Few-Shot-CoT (4 samples : Second) (*1)	90.5	-
Few-Shot-CoT (8 samples)	93.0	48.7

Greatly outperforms zero-shot →

Manual CoT still better →

[Kojima et al. (2022)]

Zero-shot COT Prompting: step-by-step

No.	Category	Template	Accuracy
1	instructive	Let's think step by step.	**78.7**
2		First, (*1)	77.3
3		Let's think about this logically.	74.5
4		Let's solve this problem by splitting it into steps. (*2)	72.2
5		Let's be realistic and think step by step.	70.8
6		Let's think like a detective step by step.	70.3
7		Let's think	57.5
8		Before we dive into the answer,	55.7
9		The answer is after the proof.	45.7
10	misleading	Don't think. Just feel.	18.8
11		Let's think step by step but reach an incorrect answer.	18.7
12		Let's count the number of "a" in the question.	16.7
13		By using the fact that the earth is round,	9.3
14	irrelevant	By the way, I found a good restaurant nearby.	17.5
15		Abrakadabra!	15.5
16		It's a beautiful day.	13.1
-		(Zero-shot)	17.7

In-Context Learning

- **Advantages:**
  - No finetuning needed
  - Prompt engineering (e.g. CoT) can improve performance
- **Limitations:**
  - Limits to what you can fit in context
  - Complex tasks will probably need gradient steps

Instruction Finetuning

PROMPT *Explain the moon landing to a 6 year old in a few sentences.*

COMPLETION GPT-3

```
Explain the theory of gravity to a 6 year old.

Explain the theory of relativity to a 6 year old in a few sentences.

Explain the big bang theory to a 6 year old.
```

PROMPT *Explain the moon landing to a 6 year old in a few sentences.*

COMPLETION **Human**

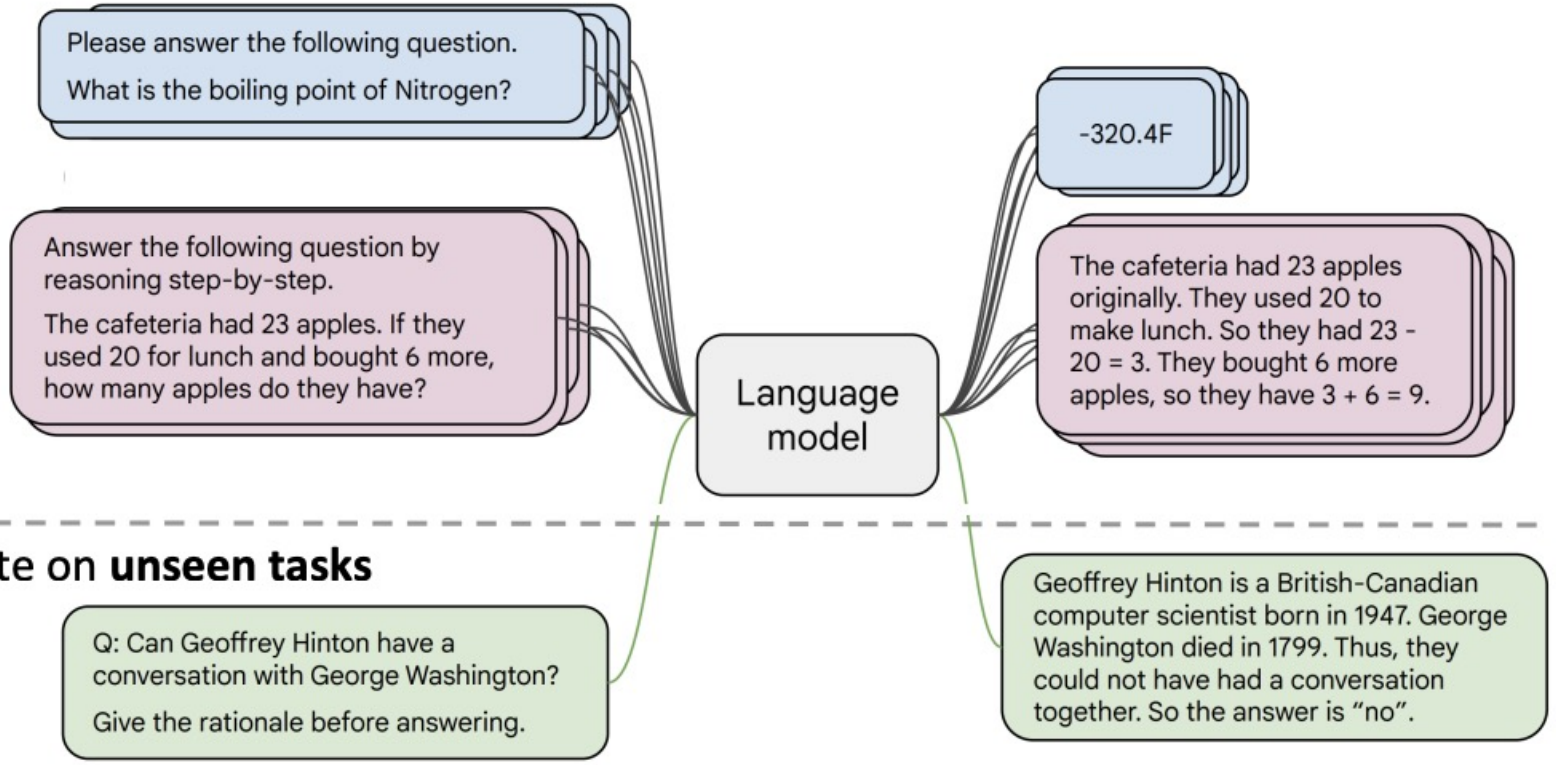
A giant rocket ship blasted off from Earth carrying astronauts to the moon. The astronauts landed their spaceship on the moon and walked around exploring the lunar surface. Then they returned safely back to Earth, bringing home moon rocks to show everyone.

Language models are not aligned with user intent.
Finetuning to the rescue!

[Ouyang et al., 2022]

Instruction Finetuning

- **Collect examples** of (instruction, output) pairs across many tasks and finetune an LM

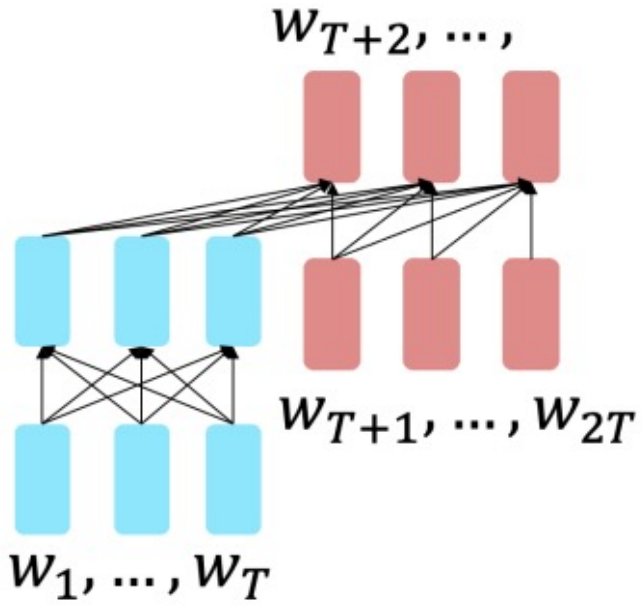


- Evaluate on **unseen tasks**

[FLAN-T5; Chung et al., 2022]

Instruction Finetuning

- Recall the T5 encoder-decoder model [Raffel et al., 2018], pretrained on the span corruption task
- Flan-T5 [Chung et al., 2020]: T5 models finetuned on 1.8K additional tasks.



Params	Model	BIG-bench + MMLU avg (normalized)
80M	T5-Small	-9.2
	Flan-T5-Small	-3.1 (+6.1)
250M	T5-Base	-5.1
	Flan-T5-Base	6.5 (+11.6)
780M	T5-Large	-5.0
	Flan-T5-Large	13.8 (+18.8)
3B	T5-XL	-4.1
	Flan-T5-XL	19.1 (+23.2)
11B	T5-XXL	-2.9
	Flan-T5-XXL	23.7 (+26.6)

Bigger model = bigger Δ

[Chung et al., 2022]

Instruction finetuning

Model input (Disambiguation QA)

Q: In the following sentences, explain the antecedent of the pronoun (which thing the pronoun refers to), or state that it is ambiguous.

Sentence: The reporter and the chef will discuss their favorite dishes.

Options:
(A) They will discuss the reporter's favorite dishes
(B) They will discuss the chef's favorite dishes
(C) Ambiguous

A: Let's think step by step.

Before instruction finetuning

The reporter and the chef will discuss their favorite dishes.
The reporter and the chef will discuss the reporter's favorite dishes.
The reporter and the chef will discuss the chef's favorite dishes.
The reporter and the chef will discuss the reporter's and the chef's favorite dishes.

✖ (doesn't answer question)

After instruction finetuning

The reporter and the chef will discuss their favorite dishes does not indicate whose favorite dishes they will discuss. So, the answer is (C). ✔

- Simple and straightforward, generalize to unseen tasks.
- Highly recommend trying FLAN-T5 out to get a sense of its capabilities.

Limitations of instruction finetuning

- Expensive to collect ground truth data for tasks.
- Tasks like open-ended creative generation have no right answer.
  - Write me a story about a dog and her pet grasshopper.
- Language modeling penalizes all token-level mistakes equally, but some errors are worse than others.
  - Such as choosing a completely unrelated word or changing the meaning of a sentence, can be more severe than minor mistakes in word choice.
- Even with instruction finetuning, there a mismatch between the LM objective and the objective of "satisfy human preferences"!

Can we explicitly attempt to satisfy human preferences?

Reinforcement Learning from Human Feedback (RLHF)

- RLHF employs reinforcement learning (RL) to adapt LLMs to human feedback by learning a reward model.
- Three components:
  - A pre-trained LM to be aligned: typically a generative model that is initialized with existing pretrained LM parameters.
  - A reward model learning from human feedback: provides (learned) guidance signals that reflect human preferences for the text generated by the LM, usually in the form of a scalar value.
    - The reward model can take on two forms: a fine-tuned LM or a LM trained using human preference data.
  - A RL algorithm training the LM: Proximal Policy Optimization (PPO) is a widely used.

[Ouyang et al., 2022]

InstructGPT: scaling up RLHF to tens of thousands of tasks

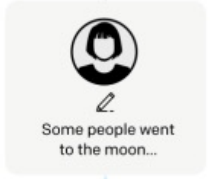
30k tasks!

Step 1

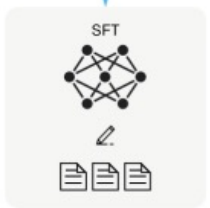
Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.



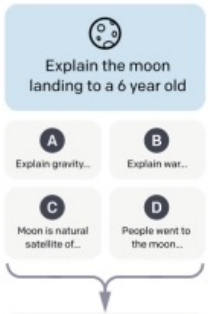
This data is used to fine-tune GPT-3 with supervised learning.



Step 2

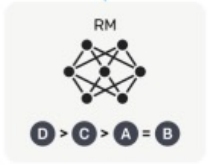
Collect comparison data, and train a reward model.

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

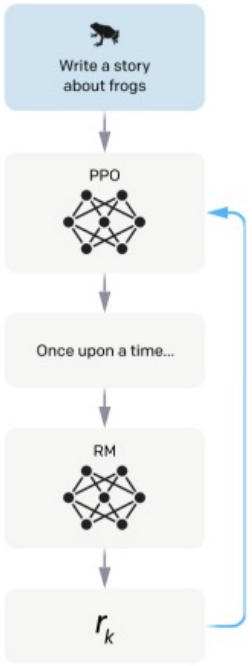
Optimize a policy against the reward model using reinforcement learning.

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.



[Ouyang et al., 2022]

64

InstructGPT

PROMPT *Explain the moon landing to a 6 year old in a few sentences.*

COMPLETION

GPT-3

```
Explain the theory of gravity to a 6 year old.

Explain the theory of relativity to a 6 year old in a few sentences.

Explain the big bang theory to a 6 year old.

Explain evolution to a 6 year old.
```

InstructGPT

```
People went to the moon, and they took pictures of what they saw, and
sent them back to the earth so we could all see them.
```

ChatGPT: optimizing language for dialogue

- Note: OpenAI (and similar companies) are keeping more details secret about ChatGPT training (including data, training parameters, model size)— perhaps to keep a competitive edge…

Instruction finetuning!

We trained this model using Reinforcement Learning from Human Feedback (RLHF), using the same methods as InstructGPT, but with slight differences in the data collection setup. We trained an initial model using supervised fine-tuning: human AI trainers provided conversations in which they played both sides—the user and an AI assistant. We gave the trainers access to model-written suggestions to help them compose their responses. We mixed this new dialogue dataset with the InstructGPT dataset, which we transformed into a dialogue format.

To create a reward model for reinforcement learning, we needed to collect comparison data, which consisted of two or more model responses ranked by quality. To collect this data, we took conversations that AI trainers had with the chatbot. We randomly selected a model-written message, sampled several alternative completions, and had AI trainers rank them. Using these reward models, we can fine-tune the model using Proximal Policy Optimization. We performed several iterations of this process.

RLHF!

Limitations of RL + Reward Modeling

- Human preferences are unreliable!
  - ”Reward hacking” is a common problem in RL
  - Chatbots are rewarded to produce responses that seem authoritative and helpful, regardless of truth
  - This can result in making up facts + hallucinations
- Models of human preferences are even more unreliable!
- There is a real concern of AI mis(alignment)!

Factuality and Hallucination

- When you fine-tune a bag-of-words model on sentiment
  - You learn word meanings from the data itself.
- When you fine-tune an embedding-based model or BERT on sentiment
  - You still learn from the data,
  - The pre-training helps generalize.
- When a language model is prompted to do a task like sentiment,
  - You really don't see enough data points to "learn" much.
  - You're relying on the model's pre-training.

- What implications does this have for producing factual knowledge from LMs?

Factuality and Hallucination

- Language models model distributions over text, not facts. There's no guarantee that what they generate is factual:
  - Language models are trained on the web. Widely-popularized falsehoods may be reproduced in language models.
  - A language model may not be able to store all rare facts, and as a result moderate probability (uncertainty or lack of strong evidence) is assigned to several options.
- There are many proposed solutions to factuality. How do we evaluate them? How do we check facts "explicitly"?

Hallucination

LLMs are prone to generate untruthful information that either conflicts with the existing source or cannot be verified by the available source. Even the most powerful LLMs such as ChatGPT face great challenges in migrating the hallucinations the generated texts. This issue can be partially alleviated by special approaches such as alignment tuning and tool utilization.

Factuality and Hallucination

There are still many limitations of large LMs (size, hallucination) that may not be solvable with RLHF!

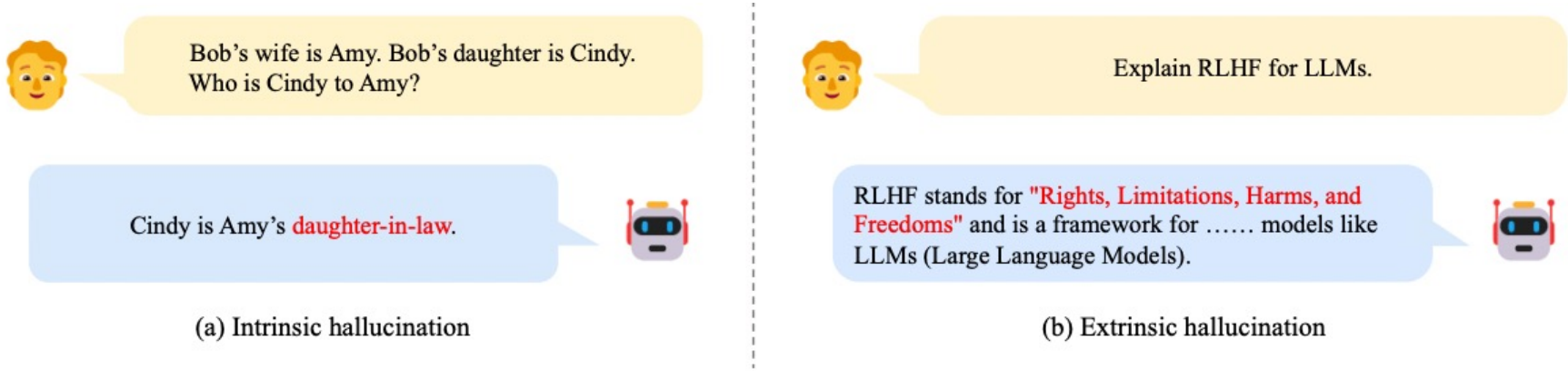


Fig. 14: Examples of intrinsic and extrinsic hallucination for a public LLM (access date: March 19, 2023). As an example of intrinsic hallucination, the LLM gives a conflicting judgment about the relationship between Cindy and Amy, which contradicts the input. For extrinsic hallucination, in this example, the LLM seems to have an incorrect understanding of the meaning of RLHF (reinforcement learning from human feedback), though it can correctly understand the meaning of LLMs (in this context).

Knowledge recency

- LLMs would encounter difficulties when solving tasks that require the latest knowledge beyond the training data.
- To tackle this issue, a straightforward approach is to regularly update LLMs with new data.
- However, it is very costly to fine-tune LLMs, and also likely to cause the catastrophic forgetting issue when incrementally training LLMs.
- Therefore, it is necessary to develop efficient and effective approaches that can integrate new knowledge into existing LLMs, making them up-to-date.

Knowledge Recency

The parametric knowledge of LLMs is hard to be updated in a timely manner. Augmenting LLMs with external knowledge sources is a practical approach to tackling the issue. However, how to effectively update knowledge within LLMs remains an open research problem.

Evaluation metrics: ROUGE

ROUGE (Recall-Oriented Understudy for Gisting Evaluation)

- Based on BLEU
- Not as good as human evaluation ('did this answer the user's question?')
- But much more convenient

Given a document D:

1. Have N humans produce a set of reference summaries of D
2. Run system, giving automatic summary X
3. What percentage of the N-grams from the reference summaries appear in X?

$$\text{ROUGE-N} = \frac{\sum_{S \in \text{References Summaries}} \sum_{\text{gram}_n \in S} \text{count}_{\text{match}}(\text{gram}_n)}{\sum_{S \in \text{References Summaries}} \sum_{\text{gram}_n \in S} \text{count}(\text{gram}_n)}$$

ROUGE: A Package for Automatic Evaluation of Summaries, Lin, 2004 http://www.aclweb.org/anthology/W04-1013

Evaluation metrics: ROUGE

ROUGE (Recall-Oriented Understudy for Gisting Evaluation)

$$\text{ROUGE-N} = \frac{\sum_{S \in \text{References Summaries}} \sum_{\text{gram}_n \in S} \text{count}_{\text{match}}(\text{gram}_n)}{\sum_{S \in \text{References Summaries}} \sum_{\text{gram}_n \in S} \text{count}(\text{gram}_n)}$$

Like BLEU, it's based on n-gram overlap, differences:

- ROUGE has no brevity penalty
- Rouge is based on recall, while BLEU is based on precision
  - Precision is more important for MT (then add brevity penalty to fix under-translation)
  - Recall is more important for summarization (assuming you have a max length constraint)
  - However, often a F1 (combination of precision and recall) version of ROUGE is reported anyway.

ROUGE: A Package for Automatic Evaluation of Summaries, Lin, 2004 http://www.aclweb.org/anthology/W04-1013

Evaluation metrics: ROUGE

- BLEU is reported as a single number, which is combination of the precisions for n=1,2,3,4 n-grams.
- ROUGE scores are reported separately for each n-gram.
- The most commonly-reported ROUGE scores are:
  - ROUGE-1:* unigram overlap
  - ROUGE-2: bigram overlap
  - ROUGE-L: longest common subsequence overlap
- There is now a convenient Python implementation of ROUGE!

Evaluation of NLG

- There are not ideal evaluation for open-ended tasks.
  - Word overlap based metrics, such as BLEU, ROUGE, perperlexity.
  - No automatic metrics to adequately capture overall quality (i.e. a proxy for human quality judgment).

- More focused metrics to capture particular aspects of generated text:
  - Fluency (compute probability w.r.t. well-trained LM)
  - Correct style (prob w.r.t. LM trained on target corpus)
  - Diversity (rare word usage, uniqueness of n-grams)
  - Relevance to input (semantic similarity measures)
  - Simple things like length and repetition
  - Task-specific metrics e.g. compression rate for summarization

- Though these don't measure overall quality, they can help us track some important qualities that we care about.

Human Evaluation

- Human judgments are regarded as the gold standard
- Of course, we know that human eval is slow and expensive
- ...but are those the only problems?
- Supposing you do have access to human evaluation:
- Does human evaluation solve all of your problems?

- No!
- Conducting human evaluation effectively is very difficult
- Humans:
  - are inconsistent
  - can be illogical
  - lose concentration
  - misinterpret your question
  - can't always explain why they feel the way they do

Automatic Evaluation + Human Evaluation

Interpretability

- The ability to understand and explain the decisions or predictions of NLP models.
- Ensuring model trustworthiness, identifying biases, and providing insights into model behavior.
  - **Feature Importance:** which features are most important for making predictions
  - **Local interpretability:** explain model predictions on specific instances
  - **Global Interpretability**: This involves understanding the overall behavior of the model
  - **Attention Mechanisms**: provide insights into which parts of the input are being attended to during processing.
  - **Word Embedding Analysis**: provide insights into the relationships between words in the model's learned space; t-SNE or PCA for visualization.
  - **Visualizations**: help in understanding its decision-making process. For example, heatmaps can be used to highlight important words or phrases.
  - **Bias Detection and Mitigation**: Interpretability methods can be used to identify and mitigate biases in NLP models, helping to ensure fairness and ethical use.
  - **Error Analysis**: Examining cases where the model makes mistakes can provide insights into its limitations and areas for improvement.
  - And many others …

Ethics

Here are some of the key concerns of ethical issues associated with NLP models.

- **Bias and Fairness**: learn biases present in the data they are trained on, leading to unfair or discriminatory outcomes.
- **Transparency and Explainability**: it is challenging to explain how and why a model makes certain predictions, which is crucial for trust and accountability.
- **Data Privacy and Security**: NLP models often require large amounts of data to be trained effectively. If not handled properly, sensitive information could be inadvertently disclosed.
- **Misinformation and Disinformation**: NLP models can be used to generate fake news or deceptive content. This poses a significant challenge for combating misinformation and disinformation online.
- **Environmental Impact**: Large NLP models often require significant computational resources to train and deploy. This can have a substantial environmental impact due to the energy consumption associated with high-performance computing.
- And many more …

Readings

1. Language Models are Few-Shot Learners
2. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models
3. Finetuned Language Models Are Zero-Shot Learners
4. Learning to summarize from human feedback

Course Outcomes

1. Implement gradient descent (GD) and stochastic gradient descent (SGD) techniques for learning problems and understand the theory behind them.
2. Apply word2vec models in real-world text corpora.
3. Understand the neural networks models and backpropagation optimization. Implement neural network models in TensorFlow or others.
4. Understand dependency parsing, recurrent neural networks and their application in NLP.
5. Understand convolutional neural networks and their application in NLP.
6. Understand sequence to sequence models and attention in NLP deep neural networks.
7. Understand the recent advances in NLP, such as transformers, pre-training, fine-tuning, prompting, etc.

STEVENS

INSTITUTE of TECHNOLOGY

THE INNOVATION UNIVERSITY®

stevens.edu

Thank You