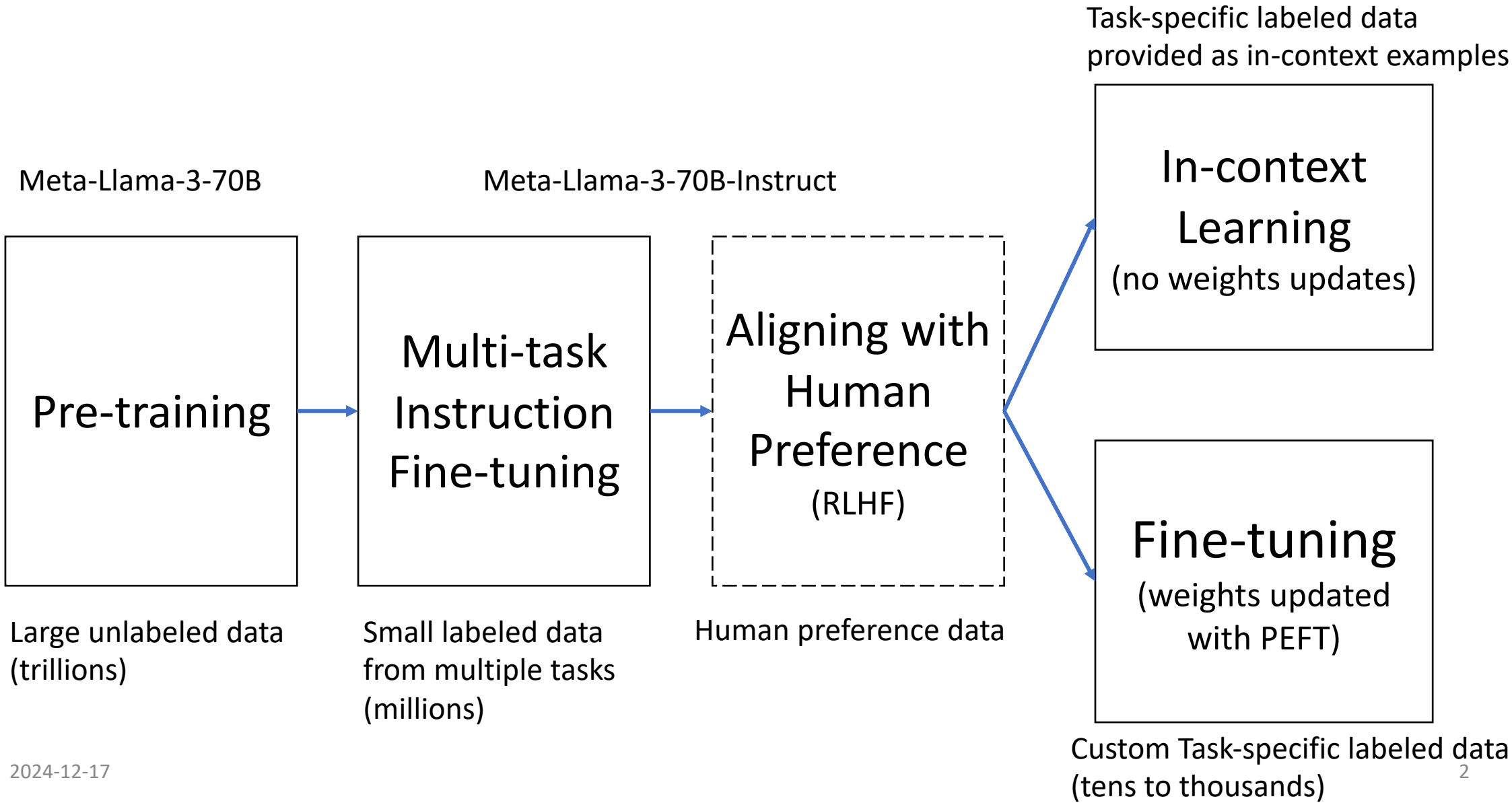
Few-Shot Parameter-Efficient Fine-Tuning is Better and Cheaper than In-Context Learning

LLM: Few-Shot Fine-Tuning V.S. In Context Learning

Houston Machine Learning Meetup

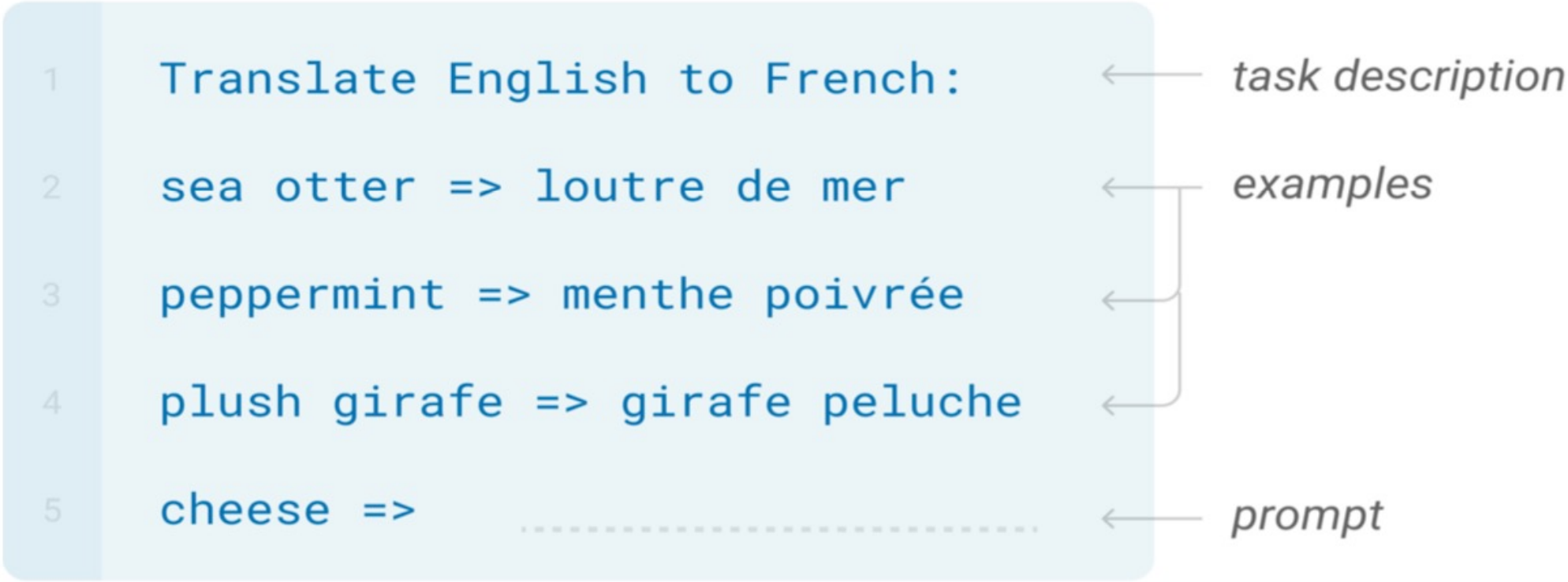
12/13/2024

LLM Training Paradigm



Few-shot In-Context Learning (ICL)

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



Few-shot Fine-tuning (PEFT)

- Fine-tuning is a process that involves adjusting a pretrained model's weights to better fit your specific data or task. This is different from in-context learning, as it actually trains a new model on your custom data.

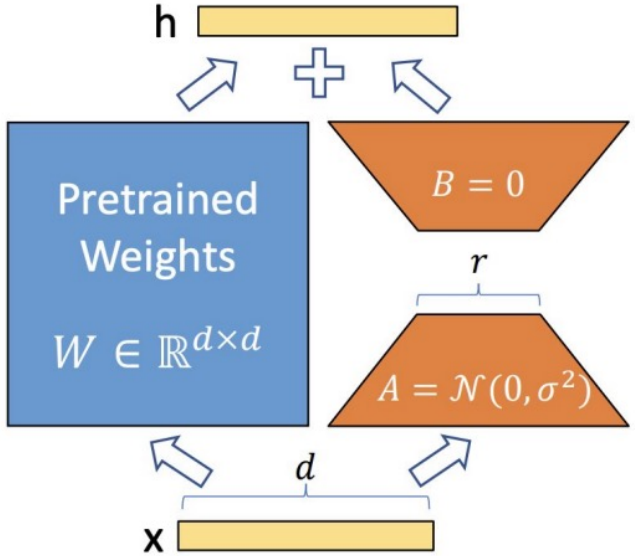


Figure 1: Our reparametrization. We only train A and B.

$$h = W_0 x + \Delta W x = W_0 x + BAx$$

- Only update the low-rank matrix
- 10000x less trainable parameter
- 3x less GPU memory requirement
- Apply to any linear layer
- No inference overhead

Comparison: ICL v.s. PEFT

ICL

PROS:

- ICL enables single models to perform many tasks without additional training

CONS:

- Processing the k in-context examples increases the inference cost by approximately k times compared to processing the unlabeled example alone.
- ICL typically produces inferior performance compared to fine-tuning.
- The exact formatting of the prompt (including the wording and ordering of examples) can have significant and unpredictable impact on the model's performance.

PEFT

PROS:

- No additional inference cost compared to processing the unlabeled example directly
- Have significant better performance comparing to ICL

CONS:

- One-time fine-tuning cost to train the model

Few-Shot Parameter-Efficient Fine-Tuning is Better and Cheaper than In-Context Learning

Haokun Liu* **Derek Tam*** **Mohammed Muqeeth***

Jay Mohta **Tenghao Huang** **Mohit Bansal** **Colin Raffel**

Department of Computer Science
University of North Carolina at Chapel Hill

- Published at NIPS 2022, ~760 citations so far.
- This paper introduces $(IA)^3$: Infused Adapter by Inhibiting and Amplifying Inner Activations
- Argues that PEFT is better than ICL in few-shot settings
- Among the first to answer:

whether PEFT methods work well when very little labeled data is available

Overview

(IA)3: “Infused Adapter by Inhibiting and Amplifying Inner Activations”

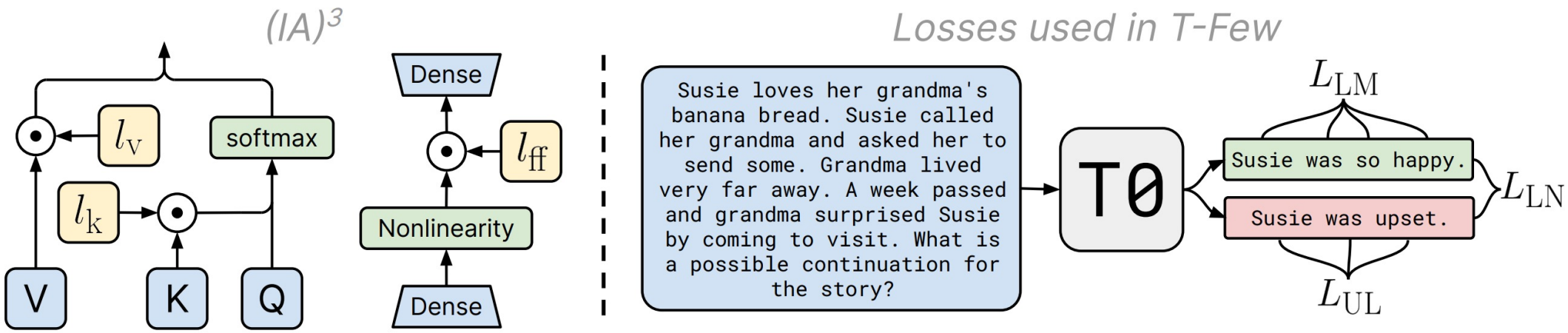
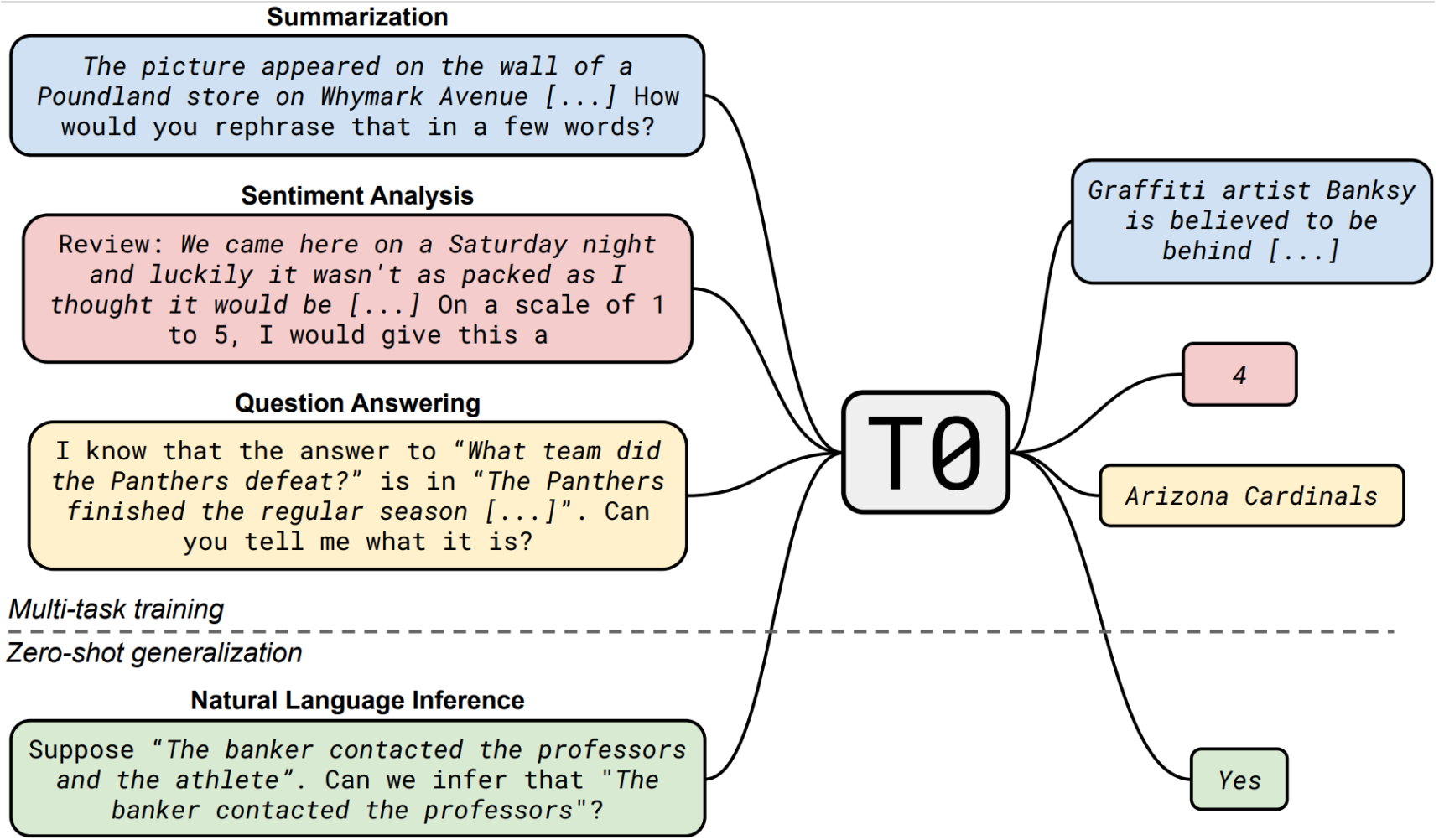


Figure 1: Diagram of `(IA)3` and the loss terms used in the `T-Few` recipe. *Left:* `(IA)3` introduces the learned vectors l_k, l_v, and l_{ff} which respectively rescale (via element-wise multiplication, visualized as $\odot$) the keys and values in attention mechanisms and the inner activations in position-wise feed-forward networks. *Right:* In addition to a standard cross-entropy loss L_{LM}, we introduce an unlikelihood loss L_{UL} that lowers the probability of incorrect outputs and a length-normalized loss L_{LN} that applies a standard softmax cross-entropy loss to length-normalized log-probabilities of all output choices.

Choose a pre-trained model: T0



Choose a pre-trained model: T0

- T0 is based on T5, an encoder-decoder Transformer model that was pre-trained via a masked language modeling objective on a large corpus of unlabeled text data.
- T0 was created by fine-tuning T5 on a multitask mixture of datasets in order to enable zero-shot generalization, i.e. the ability to perform tasks without any additional gradient-based training.
- T0 was released in 3B and 11B parameter variants, referred to as “T0-3B” and simply “T0” respectively
- T0 shows zero-shot task generalization on English natural language prompts, outperforming GPT-3 (175B) on many tasks, while being 16x smaller.

Model loss: Unlikelihood loss

cross-entropy loss $L_{\mathrm{LM}} = -\frac{1}{T}\sum_t \log p(y_t|\mathbf{x}, y_{<t})$

$$L_{\mathrm{UL}} = -\frac{\sum_{n=1}^{N}\sum_{t=1}^{T^{(n)}} \log(1 - p(\hat{y}_i^{(n)}|\mathbf{x}, \hat{y}_{<t}^{(n)}))}{\sum_{n=1}^{N} T^{(n)}}$$

We discourage the model from predicting tokens from incorrect target sequences, where ^y (n) = (ˆy1(n), ^y2(n), . . . , yˆT(n)) is the n-th of N incorrect target sequences.

Model loss: Length Normalization

$$\beta(\mathbf{x}, \mathbf{y}) = \frac{1}{T} \sum_{t=1}^{T} \log p(y_t | \mathbf{x}, y_{<t}).$$

$$L_{\text{LN}} = -\log \frac{\exp(\beta(\mathbf{x}, \mathbf{y}))}{\exp(\beta(\mathbf{x}, \mathbf{y})) + \sum_{n=1}^{N} \exp(\beta(\mathbf{x}, \hat{\mathbf{y}}^{(n)}))}$$

we maximize the length-normalized log probability of the correct answer choice

Model PEFT: Recaling vectors

We found it was sufficient to introduce **rescaling vectors on the keys and values** in **self-attention and encoder-decoder attention** mechanisms and on the **intermediate activation of the position-wise feed-forward** networks.

the notation from Vaswani et al. [33], we introduce three learned vectors $l_{\mathrm{k}} \in \mathbb{R}^{d_{\mathrm{k}}}, l_{\mathrm{v}} \in \mathbb{R}^{d_{\mathrm{v}}}$, and $l_{\mathrm{ff}} \in \mathbb{R}^{d_{\mathrm{ff}}}$, which are introduced into the attention mechanisms as:

$$\mathrm{softmax}\left(\frac{Q(l_{\mathrm{k}} \odot K^T)}{\sqrt{d_k}}\right)(l_{\mathrm{v}} \odot V)$$

and in the position-wise feed-forward networks as $(l_{\mathrm{ff}} \odot \gamma(W_1 x))W_2$, where γ is the feed-forward network nonlinearity. We introduce a separate set of $l_{\mathrm{k}}, l_{\mathrm{v}}$, and l_{ff} vectors in each Transformer layer

$\odot$ represents element-wise multiplication

Comparisons with other PEFT methods

We compare with 9 strong PEFT methods:

- **BitFit**: updates only the bias parameters
- **Adapters**: introduce task-specific layers after the self-attention and position-wise feed-forward network
- **Compacter and Compacter++**: improve upon adapters by using low-rank matrices and hypercomplex multiplication
- **Prompt tuning**: learns task-specific prompt embeddings that are concatenated to the model's input
- **FISH Mask**: chooses a subset of parameters to update based on their approximate Fisher information
- **Intrinsic SAID (Structure-aware intrinsic dimension)**: performs optimization in a low-dimensional subspace
- **Prefix-tuning**: learns task-specific vectors that are concatenated to the model's activations
- **LoRA**: assigns low-rank updates to parameter matrices.

Additionally, we include the baselines of **full-model fine-tuning**.

Evaluation

- Evaluation is done using the T0 Model
- For each dataset tested, the number of few-shot examples used varies from 20 to 70, matching the same numbers as the evaluation from the paper: “Language models are few-shot learners” focusing on few-shot in context learning.

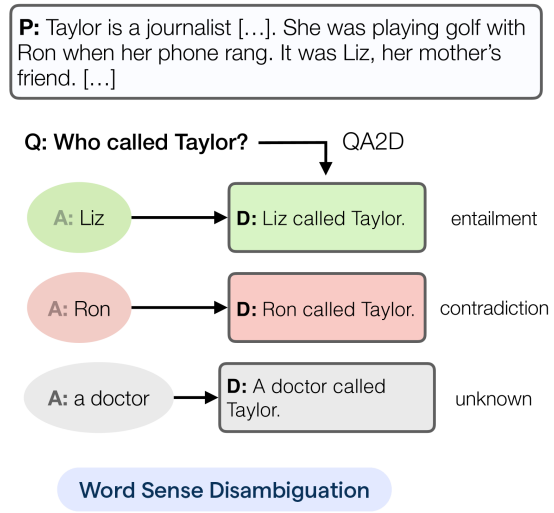
Evaluation Datasets

- Datasets used for testing are sentence completion (COPA, H-SWAG, and Story Cloze), natural language inference (ANLI, CB, and RTE), coreference resolution (WSC and Winogrande), and word sense disambiguation (WiC).

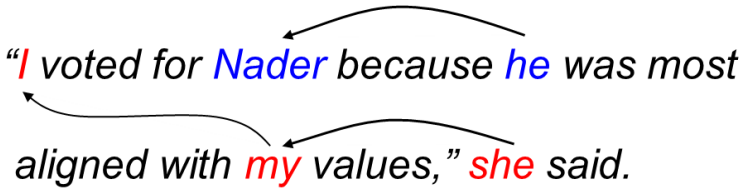
Sentence Completion

Premise: I tipped the bottle. What happened as a RESULT?
Alternative 1: The liquid in the bottle froze.
Alternative 2: The liquid in the bottle poured out.

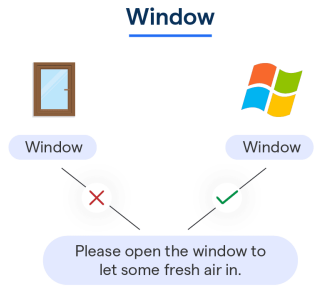
Natural Language Inference



Coreference Resolution



Word Sense Disambiguation



Results

Situations With Adversarial Generations

	COPA	H-Swag	Story
T-Few	$93.0_{2.0}$	$67.1_{6.0}$	97.
T0	90.8	33.7	9
T5+LM	68.0	60.95	6
GPT-3 (175B)	92.0	79.3	8
GPT-3 (13B)	86.0	71.3	8
GPT-3 (6.7B)	83.0	67.3	8

	RTE	CB	A
T-Few	$85.6_{2.9}$	$87.5_{3.6}$	
T0	81.2	78.6	
T5 + LM	53.4	32.1	
GPT-3 (175B)	72.9	82.1	
GPT-3 (13B)	60.6	66.1	
GPT-3 (6.7B)	49.5	60.7	55.1 55.1 55.9

On stage, a woman takes a seat at the piano. She
a) sits on a bench as her sister plays with the doll.
b) smiles with someone as the music plays.
c) is in the crowd, watching the dancers.
d) nervously sets her fingers on the keys.

A girl is going across a set of monkey bars. She
a) jumps up across the monkey bars.
b) struggles onto the monkey bars to grab her head.
c) gets to the end and stands on a wooden plank.
d) jumps up and does a back flip.

The woman is now blow drying the dog. The dog
a) is placed in the kennel next to a woman's feet.
b) washes her face with the shampoo.
c) walks into frame and walks towards the dog.
d) tried to cut her face, so she is trying to do something very close to her face.

Table 9: Comparing T-Few with few-shot ICL methods. All GPT-3 numbers are from Brown et al. [4] and all T0 numbers are from Sanh et al. [1]. Subscripts are IQR.

T-Few shine in Inference Cost

- The T-Few method shines in terms of Inference FLOPS
- Inference FLOPs listed are FLOPS to make a prediction for a single example

Method	Inference FLOPs	Training FLOPs	Disk space	Acc.
`T-Few`	1.1e12	2.7e16	4.2 MB	72.4%
T0 [1]	1.1e12	0	0 B	66.9%
T5+LM [14]	4.5e13	0	16 kB	49.6%
GPT-3 6.7B [4]	5.4e13	0	16 kB	57.2%
GPT-3 13B [4]	1.0e14	0	16 kB	60.3%
GPT-3 175B [4]	1.4e15	0	16 kB	66.6%

T-Few training costs is as much as using GPT-3 175B to process 20 inferences with few-shot ICL.

Training cost is not as big as you thought

- Training an eleven billion parameter encoder-decoder model for 1,000 steps with a batch size of 8 length-103 sequences requires approximately 3 × 11e9 × 1, 000 × 8 8 × 103 = 2.7e16 FLOPs.
- While not insignificant, this is only about 20 times larger than the FLOPs required to process a single example with few-shot ICL using GPT-3 175B.
- **Training T-Few costs as much as using GPT-3 175B to process 20 examples with few-shot ICL**.
- We also found that fine-tuning T0 with T-Few on a single dataset only takes about half an hour on a single NVIDIA A100 GPU. As of writing, this would cost about $2 USD using Microsoft Azure.

Ablation study on the effects of pre-training (PT) and losses

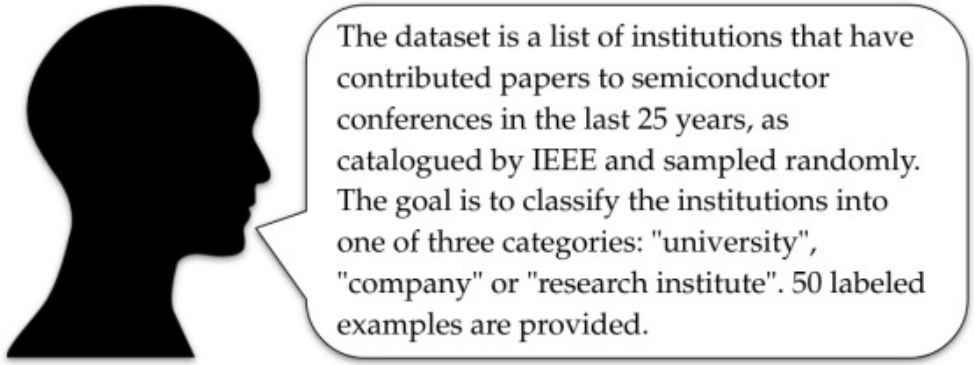
	COPA	H-Swag	StoryCloze	Winogrande	WSC	WiC
T-Few	$93.0_{2.0}$	$67.1_{6.0}$	$97.9_{0.3}$	$74.3_{1.5}$	$75.0_{5.5}$	$62.15_{7.8}$
- PT	$92.0_{2.0}$	$64.5_{6.6}$	$97.8_{0.8}$	$72.7_{1.0}$	$73.1_{6.3}$	$60.8_{6.4}$
- L_{UL} - L_{LN}	$91.0_{2.0}$	$52.1_{2.7}$	$97.4_{0.5}$	$71.9_{1.1}$	$71.2_{1.0}$	$62.2_{2.4}$
- PT - L_{UL} - L_{LN}	$94.0_{2.3}$	$52.7_{4.9}$	$98.0_{0.3}$	$74.0_{1.1}$	$72.6_{4.8}$	$62.6_{5.0}$

	RTE	CB	ANLI-R1	ANLI-R2	ANLI-R3	**Acc.**
T-Few	$85.6_{2.9}$	$87.5_{3.6}$	$59.3_{3.6}$	$49.8_{2.6}$	$44.8_{8.0}$	72.4
- PT	$84.5_{2.8}$	$83.9_{5.4}$	$57.9_{3.2}$	$48.6_{3.0}$	$43.1_{5.7}$	70.8
- L_{UL} - L_{LN}	$82.0_{0.7}$	$82.1_{3.6}$	$54.8_{0.4}$	$46.1_{0.6}$	$40.8_{5.2}$	68.3
- PT - L_{UL} - L_{LN}	$84.5_{2.9}$	$80.4_{3.6}$	$57.1_{3.1}$	$47.1_{2.4}$	$43.8_{5.9}$	69.7

PT: simply **pre-train** the new parameters introduced by (IA)3 on the same multitask mixture used to train T0

Table 10: T-Few ablation results when omitting $(\text{IA})^3$ pre-training (PT) and/or the L_{UL} and L_{LN} losses. Subscripts are IQR.

RAFT benchmark: Real-world Annotated Few-shot Tasks



The dataset is a list of institutions that have contributed papers to semiconductor conferences in the last 25 years, as catalogued by IEEE and sampled randomly. The goal is to classify the institutions into one of three categories: "university", "company" or "research institute". 50 labeled examples are provided.

Task Specification in Natural Language

Organization	Paper Title	Label
North Carolina State Univ.,Raleigh,NC,USA	3Gb/s AC-coupled chip-to-chip communication using a low-swing pulse receiver	university
Advanced LCD Technology Development Center...	Sub-Micron CMOS / MOS-Bipolar Hybrid TFTs for System Displays	company
imec,Heverlee,Belgium	24.4 A 680nA fully integrated implantable ECG-acquisition IC with analog...	research institute
...	...	...

Partially-labeled Dataset

Label the sentence based on whether it is related to an adverse drug effect (ADE)

The following is a banking customer service query. Classify it as one of the following 77 classes.

Label the impact statement as "mentions a harmful application" or "doesn't mention a harmful application" based on whether it mentions a harmful application of the research done in the paper.

Identify whether this paper should be included in a meta-review which includes the findings of systematic reviews on interventions designed to promote charitable donations.

...

Other Tasks

Figure 1: RAFT includes naturally occurring classification datasets, mimicking work that is usually given to human research assistants. Each task comes with natural language instructions and labels in addition to 50 training examples.

RAFT benchmark: Real-world Annotated Few-shot Tasks

Dataset Name	*Long inputs*	*Domain expertise*	*Detailed instructions*	*Number of classes*	*Test set size*
ADE Corpus V2 (*ADE*)	–	✓	–	2	5,000
Banking77 (*B77*)	–	–	–	77	5,000
NeurIPS impact statement risks (*NIS*)	✓	–	–	2	150
OneStopEnglish (*OSE*)	✓	–	–	3	516
Overruling (*Over*)	–	✓	–	2	2,350
Semiconductor org types (*SOT*)	–	–	–	3	449
Systematic review inclusion (*SRI*)	✓	–	✓	2	2,243
TAI safety research (*TAI*)	✓	✓	✓	2	1,639
Terms of Service (*ToS*)	–	✓	✓	2	5,000
TweetEval Hate (*TEH*)	–	–	✓	2	2,966
Twitter complaints (*TC*)	–	–	–	2	3,399

Table 1: Overview of the tasks in RAFT. *Long inputs*, *Domain expertise*, and *Detailed instructions* are some of the real-world challenges posed by RAFT.

RAFT results

To collect human baselines, we use the Surge crowdsourcing platform. We randomly select 100 data points from each test set and use a 2-step labeling process: qualification then annotation. The crowdsourced label is the plurality vote of 5 labelers.

Method	Ade Corpus V2	Banking 77	Neurips Impact Statement Risks	One Stop English	Overruling	Semiconductor Org Types	Systematic Review Inclusion	Tai Safety Research	Terms Of Service	Tweet Eval Hate	Twitter Complaints
T-Few	80.4	69.5	83.3	67.6	95.0	91.5	50.8	73.6	75.0	58.6	87.9
Human baseline [2]	83.0	60.7	85.7	64.6	91.7	90.8	46.8	60.9	62.7	72.2	89.7
PET [50]	82.2	59.3	85.7	64.6	90.8	81.6	49.3	63.8	57.6	48.3	82.4
SetFit [51]	72.6	53.8	87.2	52.1	90.7	68.2	49.3	62.8	62.0	53.2	83.7
GPT-3 [4]	68.6	29.9	67.9	43.1	93.7	76.9	51.6	65.6	57.4	52.6	82.1

Table 11: Detailed per-dataset results for T-Few and the other top-5 methods on RAFT.

Metric: Macro-average F1

Method	Acc.
T-Few	75.8%
Human baseline [2]	73.5%
PET [50]	69.6%
SetFit [51]	66.9%
GPT-3 [4]	62.7%

Table 2: Top-5 best methods on RAFT as of writing. T-Few is the first method to outperform the human baseline and achieves over 6% higher accuracy than the next-best method.

RAFT: Comparison to other PEFT methods

	# of Param	COPA	H-Swag	StoryCloze	Winogrande	WSC	WiC	RTE	CB	ANLI-R1	ANLI-R2	ANLI-R3
Full Model Fine-tuning	3B	$81.0_{11.0}$	$46.4_{8.8}$	$93.8_{2.7}$	$56.5_{1.5}$	$65.4_{7.7}$	$57.7_{3.9}$	$79.8_{3.6}$	$87.5_{5.4}$	$46.6_{2.5}$	$41.3_{0.9}$	$40.2_{5.3}$
BitFit (with LayerNorm)	1.3M	$75.0_{2.0}$	$29.5_{3.6}$	$88.6_{0.7}$	$49.6_{1.3}$	$61.5_{11.5}$	$51.7_{2.2}$	$72.2_{1.1}$	$57.1_{1.8}$	$36.5_{0.8}$	$35.3_{2.2}$	$36.6_{0.8}$
LayerNorm	250K	$76.0_{2.0}$	$29.6_{3.4}$	$88.7_{0.9}$	$49.4_{1.4}$	$63.5_{12.5}$	$52.2_{1.6}$	$71.8_{0.4}$	$57.1_{1.8}$	$36.5_{0.7}$	$35.1_{2.6}$	$36.3_{1.0}$
Adapter	12.9M	$84.0_{3.0}$	$41.9_{3.8}$	$91.7_{3.7}$	$54.7_{3.6}$	$65.4_{1.0}$	$55.5_{2.7}$	$76.2_{3.6}$	$87.5_{3.6}$	$45.1_{2.6}$	$40.4_{1.2}$	$35.3_{1.3}$
Compacter	807K	$84.0_{5.0}$	$46.4_{2.5}$	$93.5_{2.2}$	$55.5_{2.9}$	$64.4_{6.7}$	$55.2_{3.8}$	$75.8_{6.1}$	$82.1_{3.6}$	$40.8_{3.3}$	$37.4_{0.2}$	$35.8_{3.3}$
Compacter++	540K	$86.0_{3.0}$	$46.3_{3.0}$	$93.5_{1.2}$	$55.1_{1.1}$	$65.4_{3.9}$	$54.1_{2.2}$	$76.9_{0.4}$	$82.1_{3.6}$	$41.7_{0.4}$	$38.3_{1.8}$	$36.9_{1.5}$
Prompt tuning (10)	41K	$67.0_{5.0}$	$29.9_{0.6}$	$84.2_{0.8}$	$51.9_{1.6}$	$54.8_{10.6}$	$51.6_{2.0}$	$52.7_{5.4}$	$66.1_{1.8}$	$34.2_{1.9}$	$33.5_{1.1}$	$33.5_{1.3}$
Prompt tuning (100)	409K	$60.0_{19.0}$	$26.8_{0.6}$	$74.0_{3.4}$	$51.1_{0.8}$	$60.6_{4.8}$	$50.0_{1.1}$	$48.0_{2.9}$	$53.6_{17.9}$	$33.4_{1.2}$	$33.8_{0.5}$	$33.3_{0.8}$
Prefix tuning	576K	$71.0_{8.0}$	$42.1_{4.0}$	$90.2_{3.1}$	$52.0_{1.3}$	$56.7_{3.3}$	$54.2_{3.3}$	$68.6_{3.3}$	$84.0_{1.8}$	$43.3_{4.1}$	$37.5_{1.2}$	$36.5_{1.5}$
FishMask (0.2%)	6M	$82.0_{5.0}$	$44.1_{4.2}$	$94.2_{1.8}$	$54.5_{2.1}$	$63.5_{4.8}$	$52.5_{3.3}$	$76.9_{4.7}$	$83.9_{3.6}$	$43.7_{0.3}$	$39.7_{1.4}$	$37.2_{1.1}$
FishMask (0.02%)	600K	$84.0_{6.0}$	$38.2_{3.6}$	$93.6_{0.7}$	$53.9_{2.8}$	$61.5_{1.0}$	$53.5_{1.3}$	$75.5_{5.4}$	$76.8_{3.6}$	$39.9_{0.9}$	$38.1_{2.0}$	$36.2_{1.8}$
Intrinsic SAID	500K	$77.0_{4.0}$	$36.7_{4.5}$	$89.3_{2.3}$	$52.7_{2.1}$	$61.5_{8.7}$	$55.0_{2.7}$	$69.0_{7.6}$	$80.4_{0.0}$	$40.4_{3.3}$	$35.4_{4.1}$	$35.5_{1.6}$
Intrinsic SAID	20K	$76.0_{4.0}$	$38.3_{6.4}$	$89.7_{2.7}$	$50.9_{1.0}$	$55.8_{6.7}$	$55.3_{0.5}$	$66.1_{5.4}$	$83.9_{1.8}$	$41.3_{1.3}$	$38.5_{1.8}$	$35.8_{2.0}$
LoRA	9.1M	$88.0_{5.0}$	$47.1_{3.2}$	$93.6_{2.1}$	$56.8_{3.3}$	$60.6_{5.8}$	$55.2_{5.0}$	$78.3_{7.6}$	$85.7_{1.8}$	$45.1_{2.5}$	$41.0_{1.4}$	$39.5_{4.8}$
$(IA)^3$	540K	$87.0_{3.0}$	$49.4_{4.6}$	$94.7_{2.7}$	$59.8_{0.6}$	$68.3_{6.7}$	$56.0_{4.6}$	$78.0_{2.5}$	$87.5_{1.8}$	$48.6_{2.0}$	$40.8_{1.5}$	$40.8_{2.3}$

Conclusion

- We introduced T-Few, a parameter-efficient few-shot learning recipe that attains **higher accuracy than few-shot ICL** at a lower computational cost and provide a new perspective on how best to **perform few-shot learning** with large language models
- When applying T-Few to the RAFT benchmark, we **attained super-human performance** for the first time and outperformed prior submissions by a large margin.
- T-Few **reduces inference cost with not-so-big one-time training cost**: uses over 1,000× fewer FLOPs during inference than few-shot ICL with GPT-3 and only requires 30 minutes to train on a single NVIDIA A100 GPU (equivalent to 20 GPT-3 inference calls)
- Caveat: all of the experiments were on multiple choices, classification tasks, we are interested in applying T-Few to generative tasks like as summarization and question answering in future work.

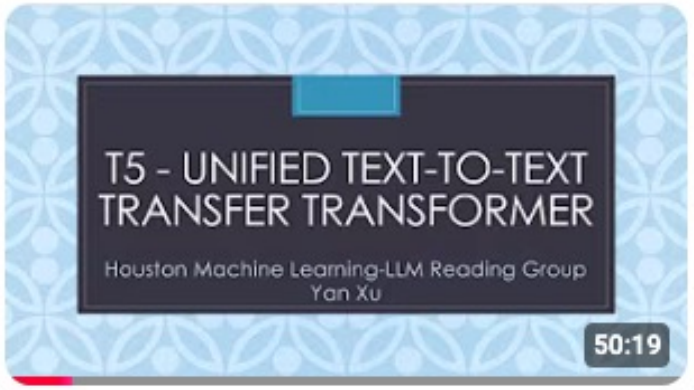
Thank you

YanAITalk

@yanaitalk · 2.2K subscribers · 61 videos

Make machine learning easy to understand! .

Customize channel | Manage videos

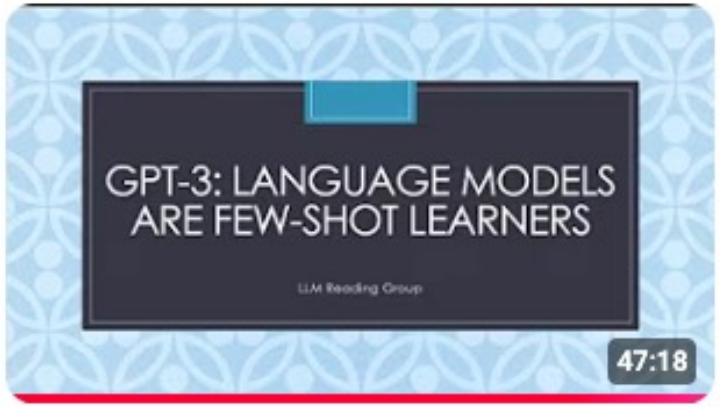


LLM: Exploring the Limits of Transfer Learning with a unified Text-to-Text...

673 views • 1 year ago

LLM: Pretraining, Instruction fine-tuning and RLHF

5.8K views • 1 year ago



LLM: GPT-3

161 views • 1 year ago

Parameter-efficient Fine-tuning of LLMs with LoRA

308 views • 3 weeks ago