

# GPT-3 and the future of language modeling

CS685 Fall 2020

Advanced Natural Language Processing

Mohit Iyyer

College of Information and Computer Sciences  
University of Massachusetts Amherst

# Stuff from last time

- How is the [CLS] token pretrained (e.g., how does it learn a contextualized vector during pretraining?) Is it shared across all pretraining sentences?
- We get multiple embeddings per token in ELMo and BERT (different layers), how do we choose which to use?
- Project proposal feedback by the end of the week!
- Practice exams available on Piazza

Today, an alternative to  
“pretrain+finetune”, which involves  
simply getting rid of fine-tuning

# The language model “scaling wars”!

ELMo: 93M params, 2-layer biLSTM

BERT-base: 110M params, 12-layer Transformer

BERT-large: 340M params, 24-layer Transformer

# The language model “scaling wars”!

ELMo: 93M params, 2-layer biLSTM

BERT-base: 110M params, 12-layer Transformer

BERT-large: 340M params, 24-layer Transformer

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

# The language model “scaling wars”!

ELMo: 93M params, 2-layer biLSTM

BERT-base: 110M params, 12-layer Transformer

BERT-large: 340M params, 24-layer Transformer

Model Name	$n_{\text{params}}$	$n_{\text{layers}}$	$d_{\text{model}}$	$n_{\text{heads}}$	$d_{\text{head}}$	Batch Size	Learning Rate
GPT-3 Small	125M	12	768	12	64	0.5M	$6.0 \times 10^{-4}$
GPT-3 Medium	350M	24	1024	16	64	0.5M	$3.0 \times 10^{-4}$
GPT-3 Large	760M	24	1536	16	96	0.5M	$2.5 \times 10^{-4}$
GPT-3 XL	1.3B	24	2048	24	128	1M	$2.0 \times 10^{-4}$
GPT-3 2.7B	2.7B	32	2560	32	80	1M	$1.6 \times 10^{-4}$
GPT-3 6.7B	6.7B	32	4096	32	128	2M	$1.2 \times 10^{-4}$
GPT-3 13B	13.0B	40	5120	40	128	2M	$1.0 \times 10^{-4}$
GPT-3 175B or “GPT-3”	175.0B	96	12288	96	128	3.2M	$0.6 \times 10^{-4}$

# The language model “scaling wars”!

ELMo: 1B training tokens

BERT: 3.3B training tokens

RoBERTa: ~30B training tokens

# The language model “scaling wars”!

ELMo: 1B training tokens

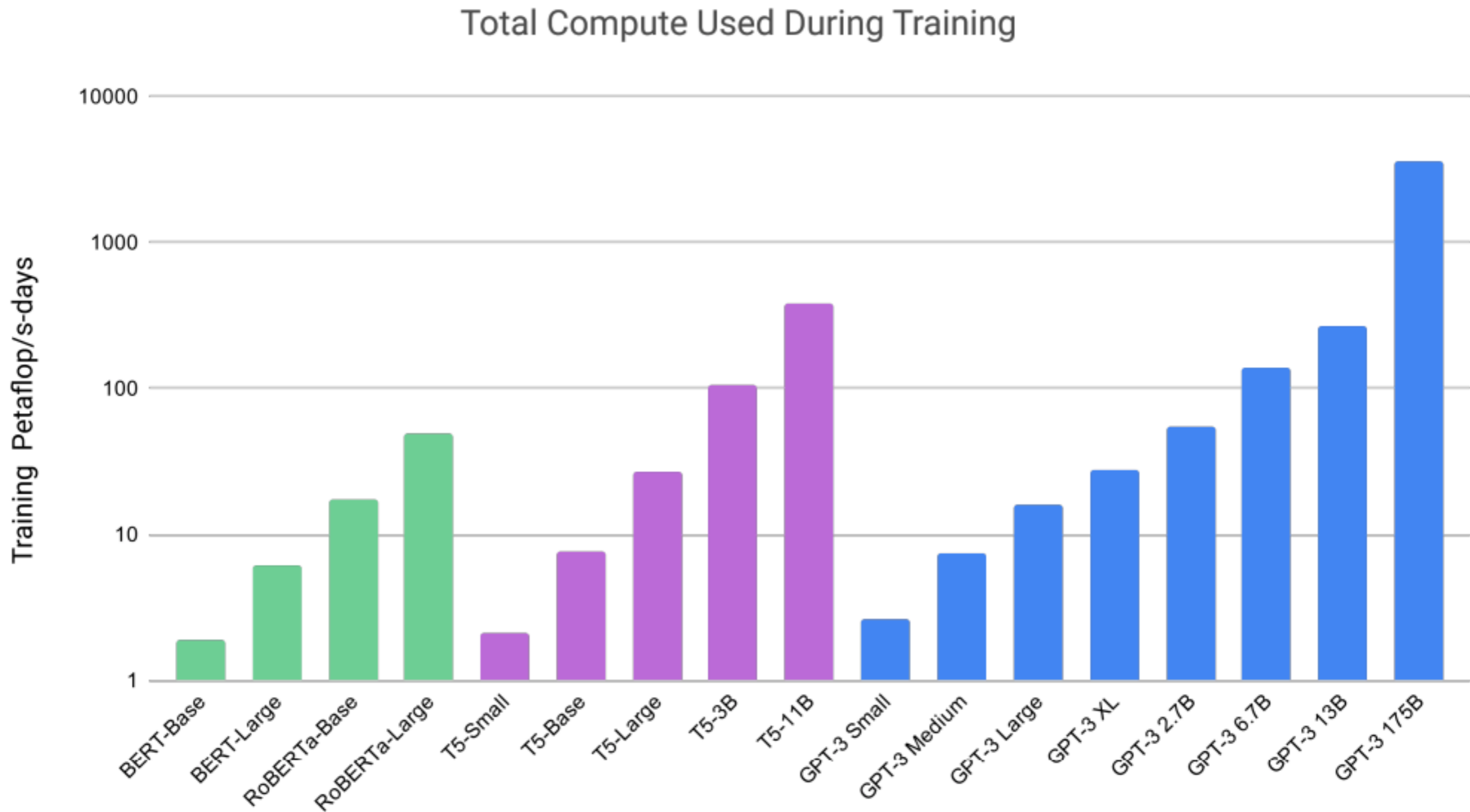
BERT: 3.3B training tokens

RoBERTa: ~30B training tokens

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



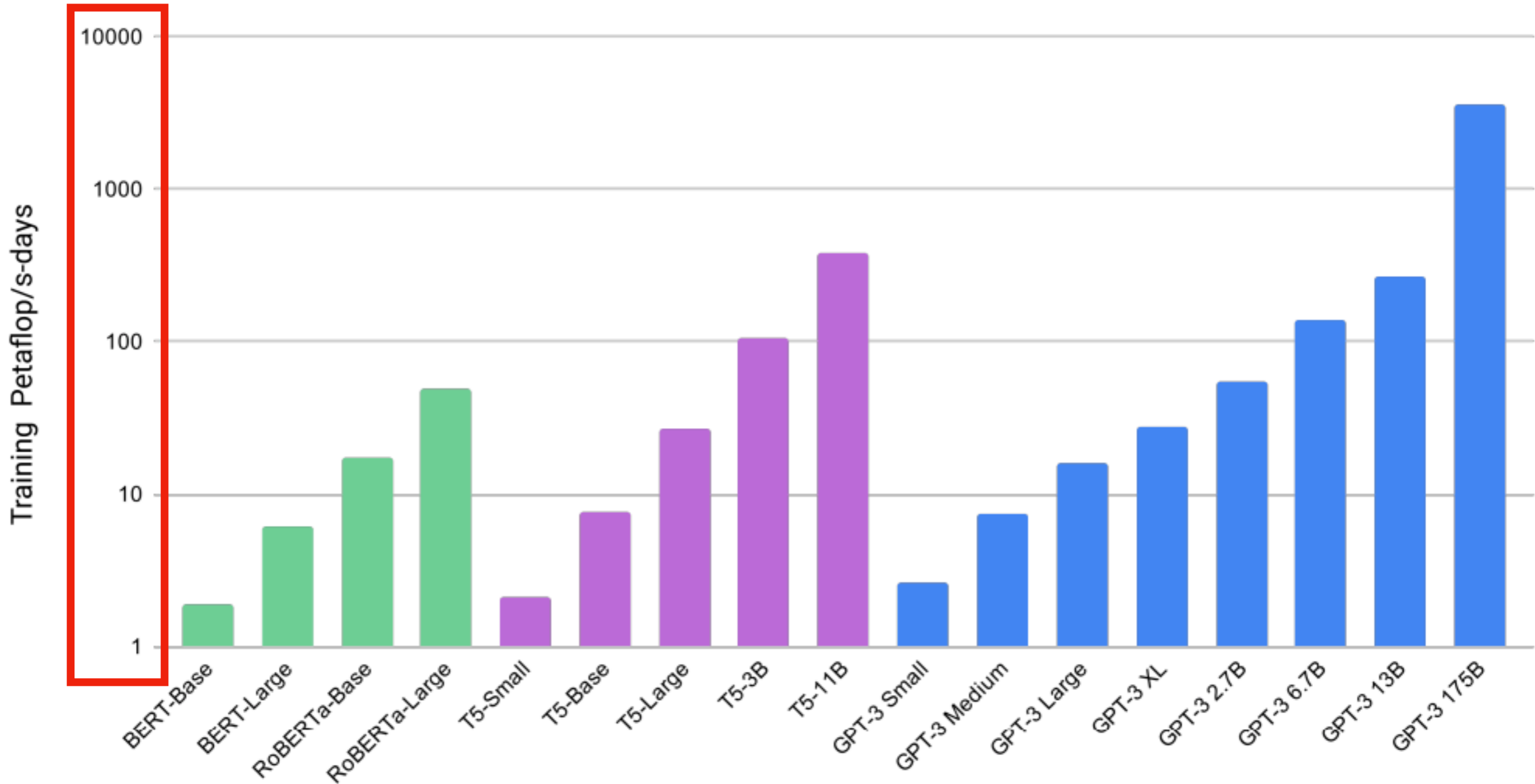
# The language model “scaling wars”!



# The language model “scaling wars”!

Log scale!

Total Compute Used During Training



so... what does all of this scaling buy us?

# 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.

1 sea otter => loutre de mer ← example #1



gradient update



1 peppermint => menthe poivrée ← example #2



gradient update



1 plush giraffe => girafe peluche ← example #N

gradient update

1 cheese => ..... ← prompt

Downstream training data

Downstream test data

## 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
```

## Zero-shot

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



The diagram shows a light blue rounded rectangle containing two lines of text. The first line is '1 Translate English to French:' with an arrow pointing to it from the text 'task description' on the right. The second line is '2 cheese =>' followed by a dotted line, with an arrow pointing to it from the text 'prompt' on the right.

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

**No fine-tuning!!! Literally just take a pretrained LM and give it the following prefix:**

**“Translate English to French: cheese =>”**

## One-shot

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

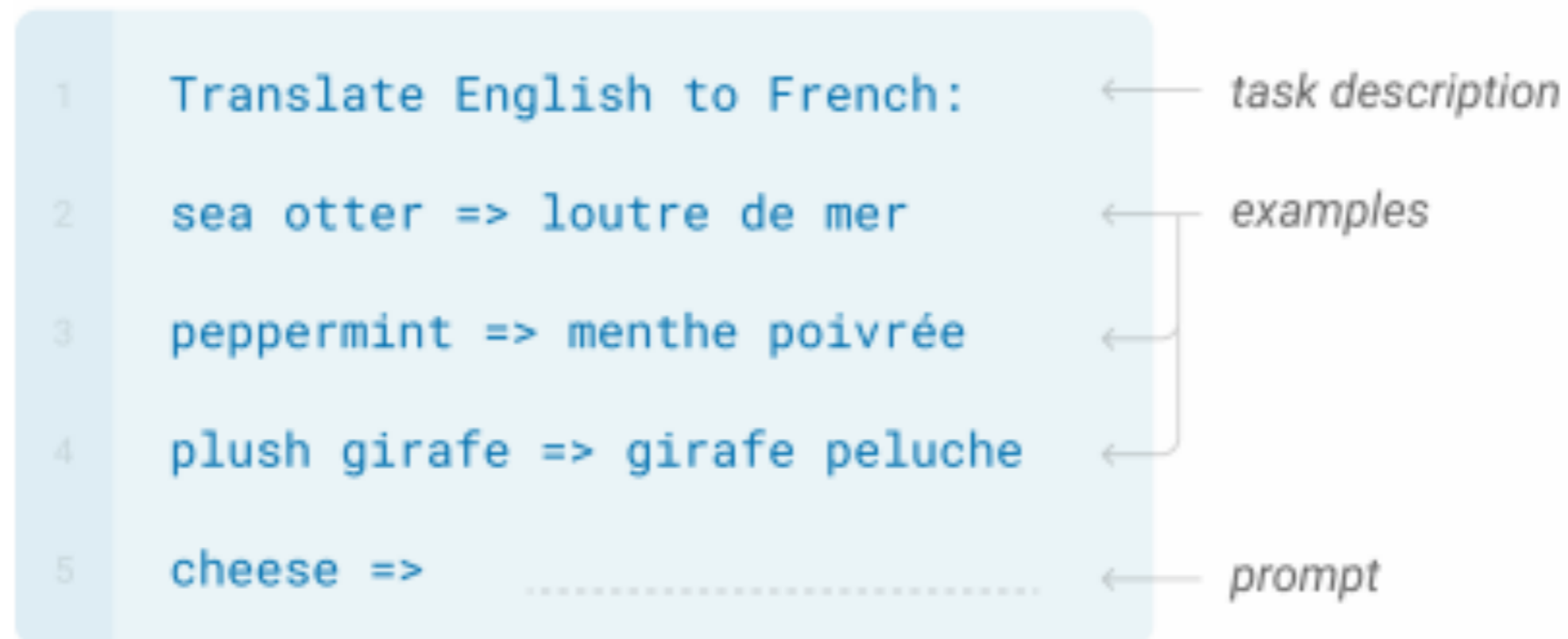


**No fine-tuning!!! Literally just take a pretrained LM and give it the following prefix:**

**“Translate English to French: sea otter => loutre de mer, cheese =>”**

## Few-shot

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



**No fine-tuning!!! Literally just take a pretrained LM and give it the following prefix:**

**“Translate English to French: sea otter => loutre de mer, peppermint => ... (few more examples), cheese =>”**

**Max of 100 examples fed into the prefix in this way**



How does this new paradigm compare to “pretrain + finetune”?

# TriviaQA

---

## Question

---

Miami Beach in Florida borders which ocean?

---

What was the occupation of Lovely Rita according to the song by the Beatles

---

Who was Poopdeck Pappys most famous son?

---

The Nazi regime was Germany's Third Reich; which was the first Reich?

---

At which English racecourse did two horses collapse and die in the parade ring due to electrocution, in February 2011?

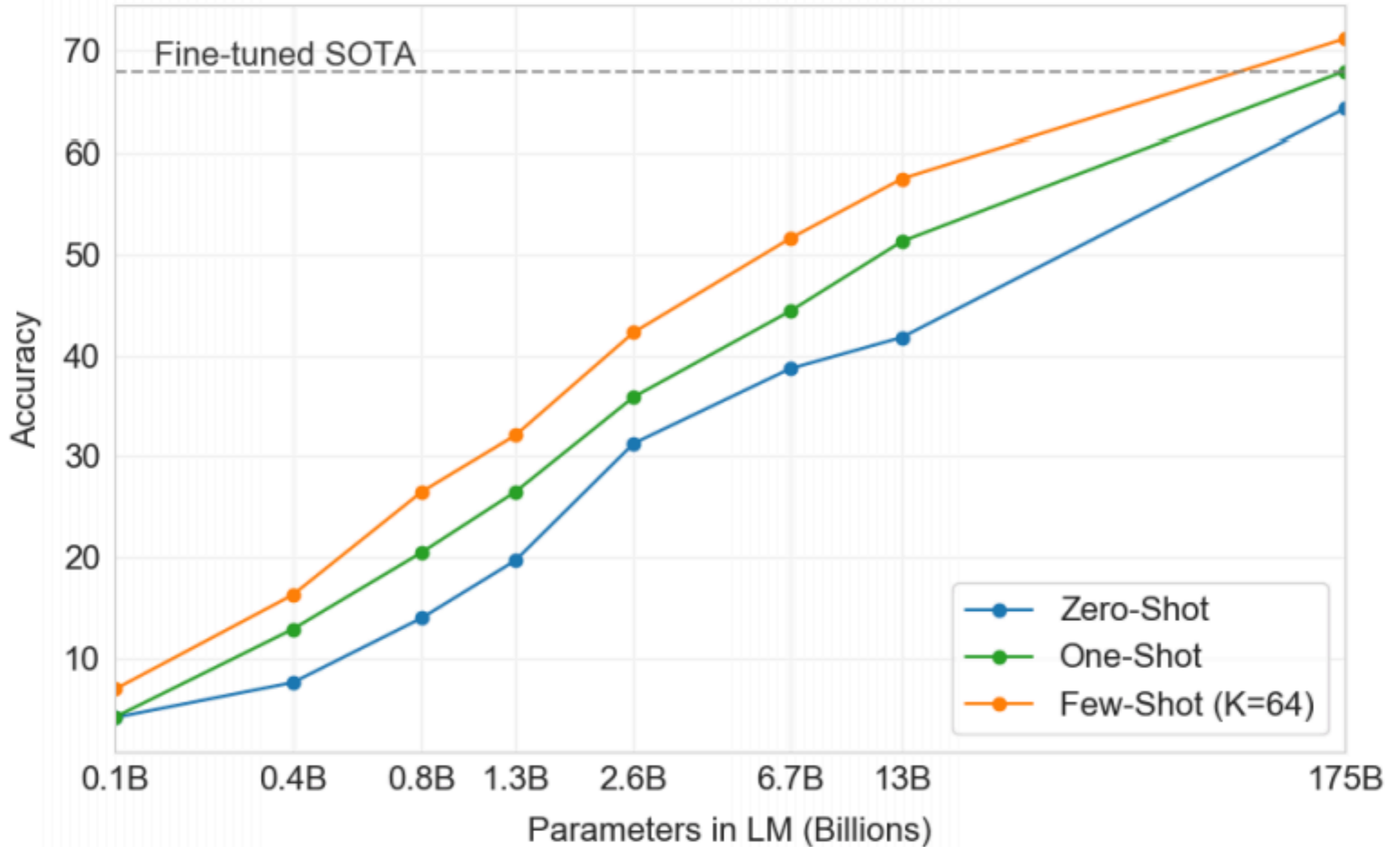
---

Which type of hat takes its name from an 1894 novel by George Du Maurier where the title character has the surname O'Ferrall ?

---

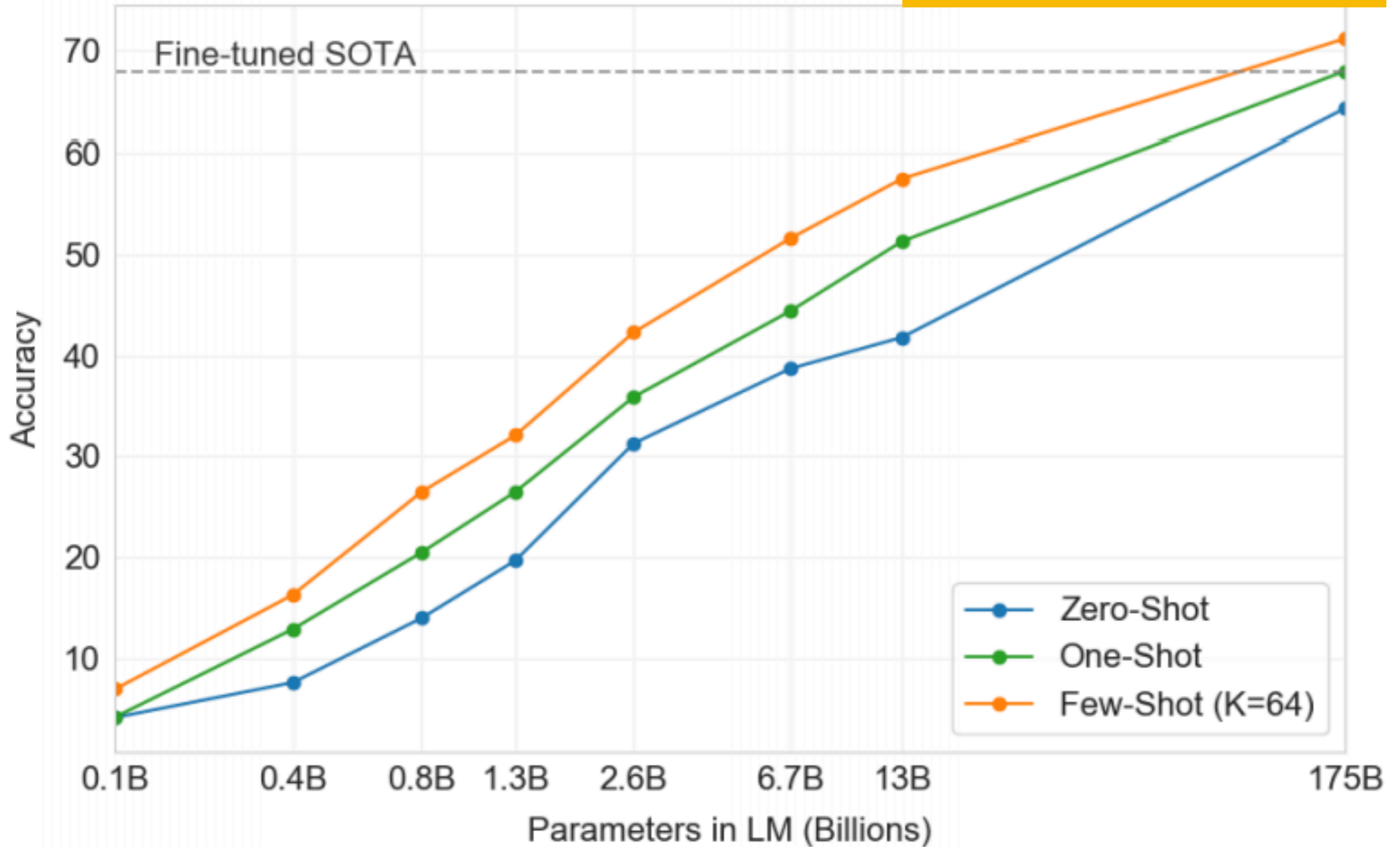
What was the Elephant Man's real name?

# TriviaQA



# TriviaQA

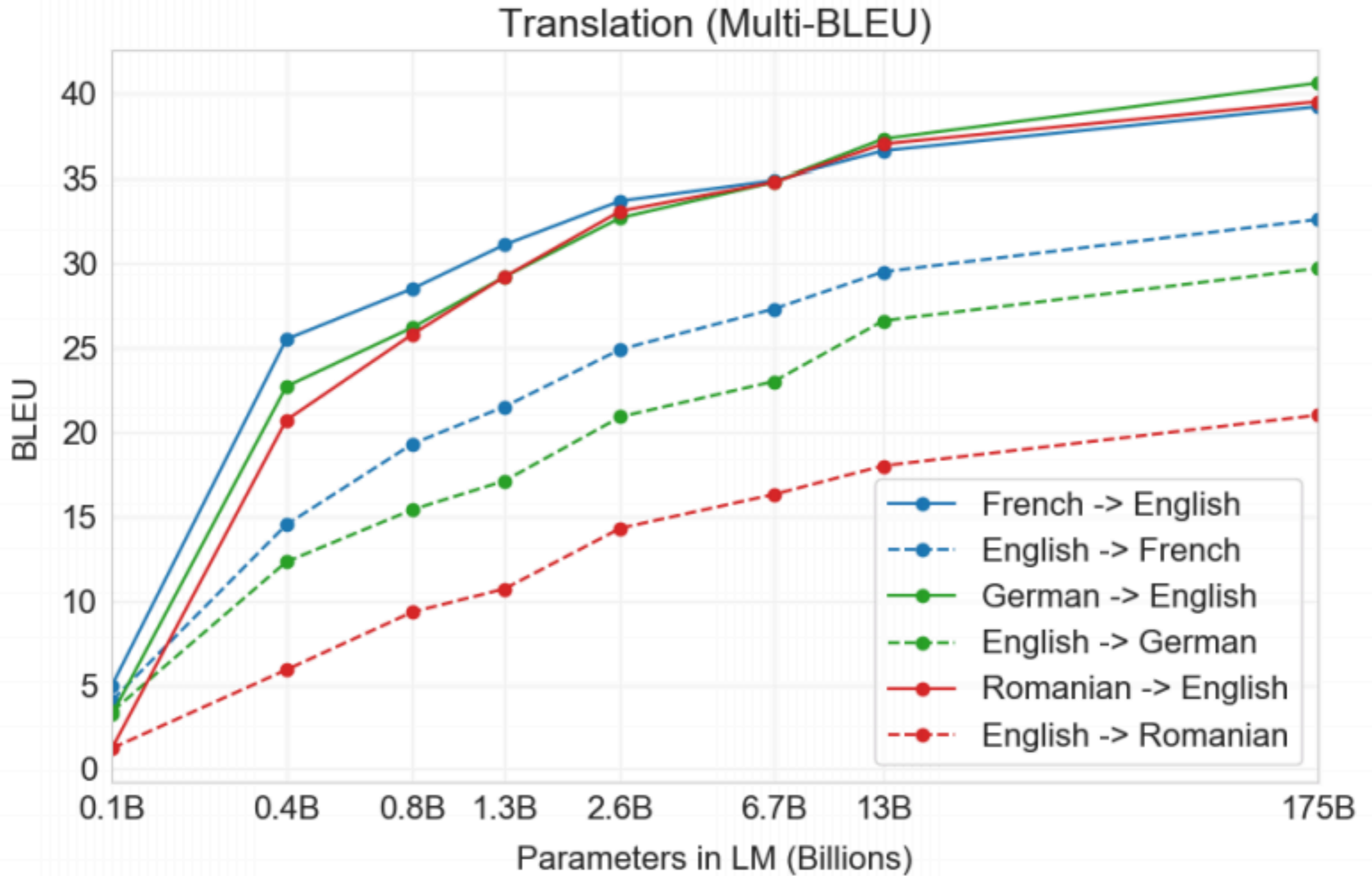
What does this mean?



What about translation? (7% of  
GPT3's training data is in  
languages other than English)

Setting	En→Fr	Fr→En	En→De	De→En	En→Ro	Ro→En
SOTA (Supervised)	<b>45.6<sup>a</sup></b>	35.0 <sup>b</sup>	<b>41.2<sup>c</sup></b>	40.2 <sup>d</sup>	<b>38.5<sup>e</sup></b>	<b>39.9<sup>e</sup></b>
XLM [LC19]	33.4	33.3	26.4	34.3	33.3	31.8
MASS [STQ <sup>+</sup> 19]	<u>37.5</u>	34.9	28.3	35.2	<u>35.2</u>	33.1
mBART [LGG <sup>+</sup> 20]	-	-	<u>29.8</u>	34.0	35.0	30.5
GPT-3 Zero-Shot	25.2	21.2	24.6	27.2	14.1	19.9
GPT-3 One-Shot	28.3	33.7	26.2	30.4	20.6	38.6
GPT-3 Few-Shot	32.6	<u>39.2</u>	29.7	<u>40.6</u>	21.0	<u>39.5</u>

Improvements haven't plateaued!



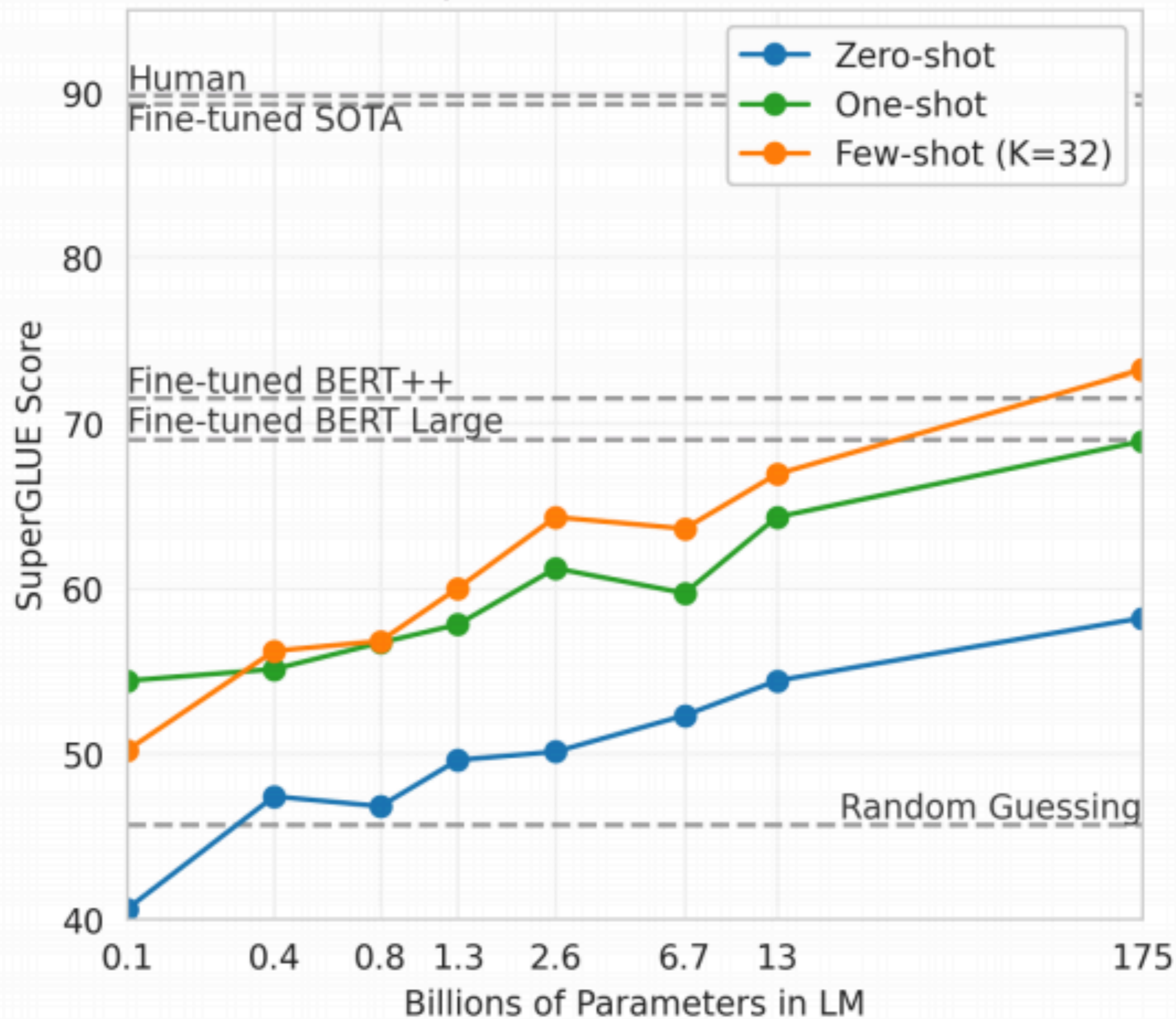
What about reading  
comprehension QA?



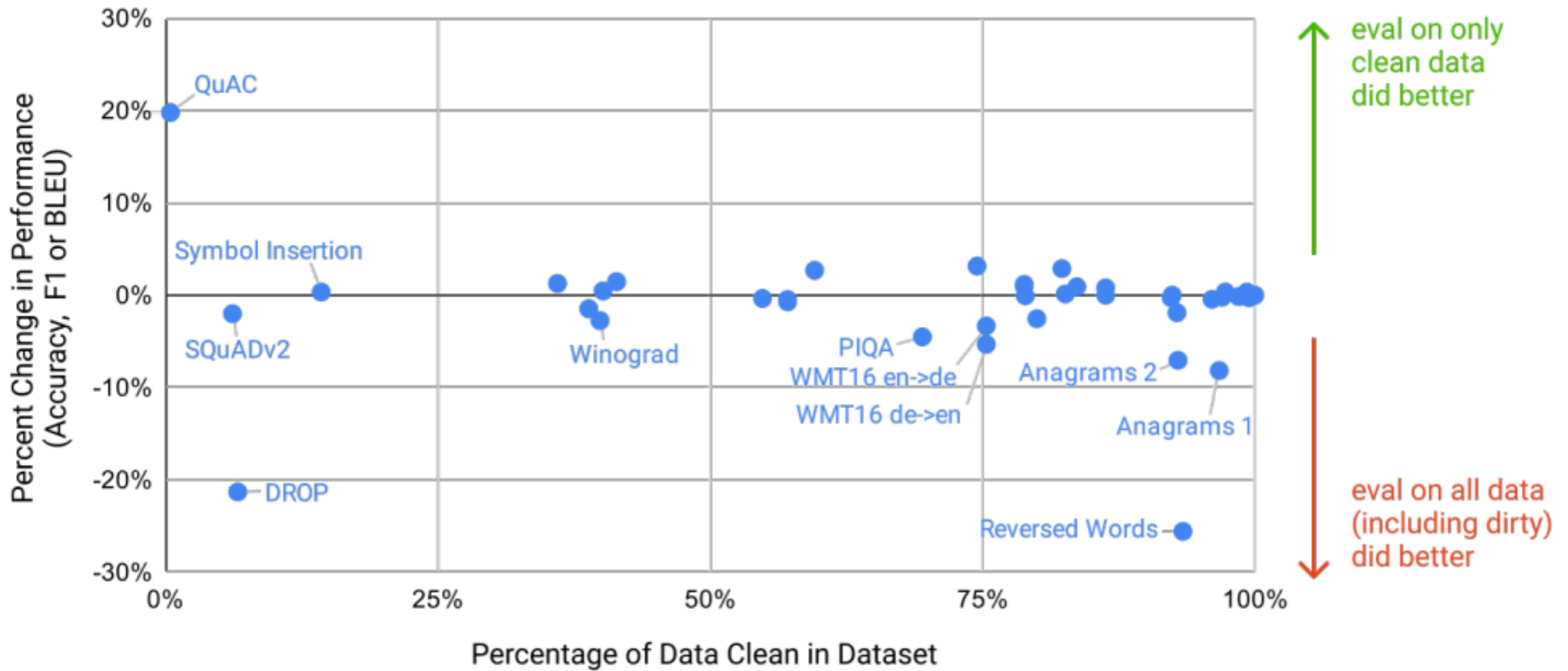
Setting	CoQA	DROP	QuAC	SQuADv2	RACE-h	RACE-m
Fine-tuned SOTA	<b>90.7<sup>a</sup></b>	<b>89.1<sup>b</sup></b>	<b>74.4<sup>c</sup></b>	<b>93.0<sup>d</sup></b>	<b>90.0<sup>e</sup></b>	<b>93.1<sup>e</sup></b>
GPT-3 Zero-Shot	81.5	23.6	41.5	59.5	45.5	58.4
GPT-3 One-Shot	84.0	34.3	43.3	65.4	45.9	57.4
GPT-3 Few-Shot	85.0	36.5	44.3	69.8	46.8	58.1

Struggles on “harder” datasets

### SuperGLUE Performance



# Data contamination



- **2 digit addition (2D+)** – The model is asked to add two integers sampled uniformly from  $[0, 100)$ , phrased in the form of a question, e.g. “Q: What is 48 plus 76? A: 124.”
- **2 digit subtraction (2D-)** – The model is asked to subtract two integers sampled uniformly from  $[0, 100)$ ; the answer may be negative. Example: “Q: What is 34 minus 53? A: -19”.
- **3 digit addition (3D+)** – Same as 2 digit addition, except numbers are uniformly sampled from  $[0, 1000)$ .
- **3 digit subtraction (3D-)** – Same as 2 digit subtraction, except numbers are uniformly sampled from  $[0, 1000)$ .
- **4 digit addition (4D+)** – Same as 3 digit addition, except uniformly sampled from  $[0, 10000)$ .
- **4 digit subtraction (4D-)** – Same as 3 digit subtraction, except uniformly sampled from  $[0, 10000)$ .
- **5 digit addition (5D+)** – Same as 3 digit addition, except uniformly sampled from  $[0, 100000)$ .
- **5 digit subtraction (5D-)** – Same as 3 digit subtraction, except uniformly sampled from  $[0, 100000)$ .
- **2 digit multiplication (2Dx)** – The model is asked to multiply two integers sampled uniformly from  $[0, 100)$ , e.g. “Q: What is 24 times 42? A: 1008”.
- **One-digit composite (1DC)** – The model is asked to perform a composite operation on three 1 digit numbers, with parentheses around the last two. For example, “Q: What is  $6+(4*8)$ ? A: 38”. The three 1 digit numbers are selected uniformly on  $[0, 10)$  and the operations are selected uniformly from  $\{+,-,*\}$ .

Setting	2D+	2D-	3D+	3D-	4D+	4D-	5D+	5D-	2Dx	1DC
GPT-3 Zero-shot	76.9	58.0	34.2	48.3	4.0	7.5	0.7	0.8	19.8	9.8
GPT-3 One-shot	99.6	86.4	65.5	78.7	14.0	14.0	3.5	3.8	27.4	14.3
GPT-3 Few-shot	100.0	98.9	80.4	94.2	25.5	26.8	9.3	9.9	29.2	21.3

A "whatpu" is a small, furry animal native to Tanzania. An example of a sentence that uses the word whatpu is:

We were traveling in Africa and we saw these very cute whatpus.

---

To do a "farduddle" means to jump up and down really fast. An example of a sentence that uses the word farduddle is:

**One day when I was playing tag with my little sister, she got really excited and she started doing these crazy farduddles.**

A "yalubalu" is a type of vegetable that looks like a big pumpkin. An example of a sentence that uses the word yalubalu is:

**I was on a trip to Africa and I tried this yalubalu vegetable that was grown in a garden there. It was delicious.**

---

A "Burringo" is a car with very fast acceleration. An example of a sentence that uses the word Burringo is:

**In our garage we have a Burringo that my father drives to work every day.**

---

A "Gigamuru" is a type of Japanese musical instrument. An example of a sentence that uses the word Gigamuru is:

**I have a Gigamuru that my uncle gave me as a gift. I love to play it at home.**

---

To "screeg" something is to swing a sword at it. An example of a sentence that uses the word screeg is:

**We screeghed at each other for several minutes and then we went outside and ate ice cream.**

So... should we drop everything  
and focus all of our efforts on  
training bigger and bigger LMs?

# Distinction between “form” and “meaning”

- **Form:** characters / words making up some text (or sounds etc for spoken language)
- **Meaning:** How the form of a given text relates to something outside of language (e.g., grounded in some world)

# Distinction between “form” and “meaning”

- Thought experiment (from Emily Bender):
  - **Training data:** All well-formed Java code on GitHub, but only the text of the code; no output; no understanding of what unit tests mean
  - **Test input:** A single Java program, possibly even from the training data
  - **Expected output:** Result of executing that program

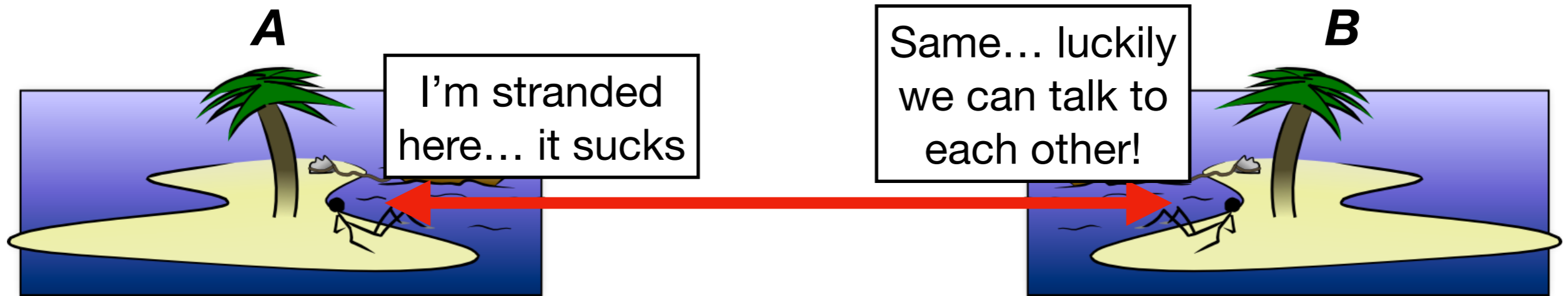


# Distinction between “form” and “meaning”

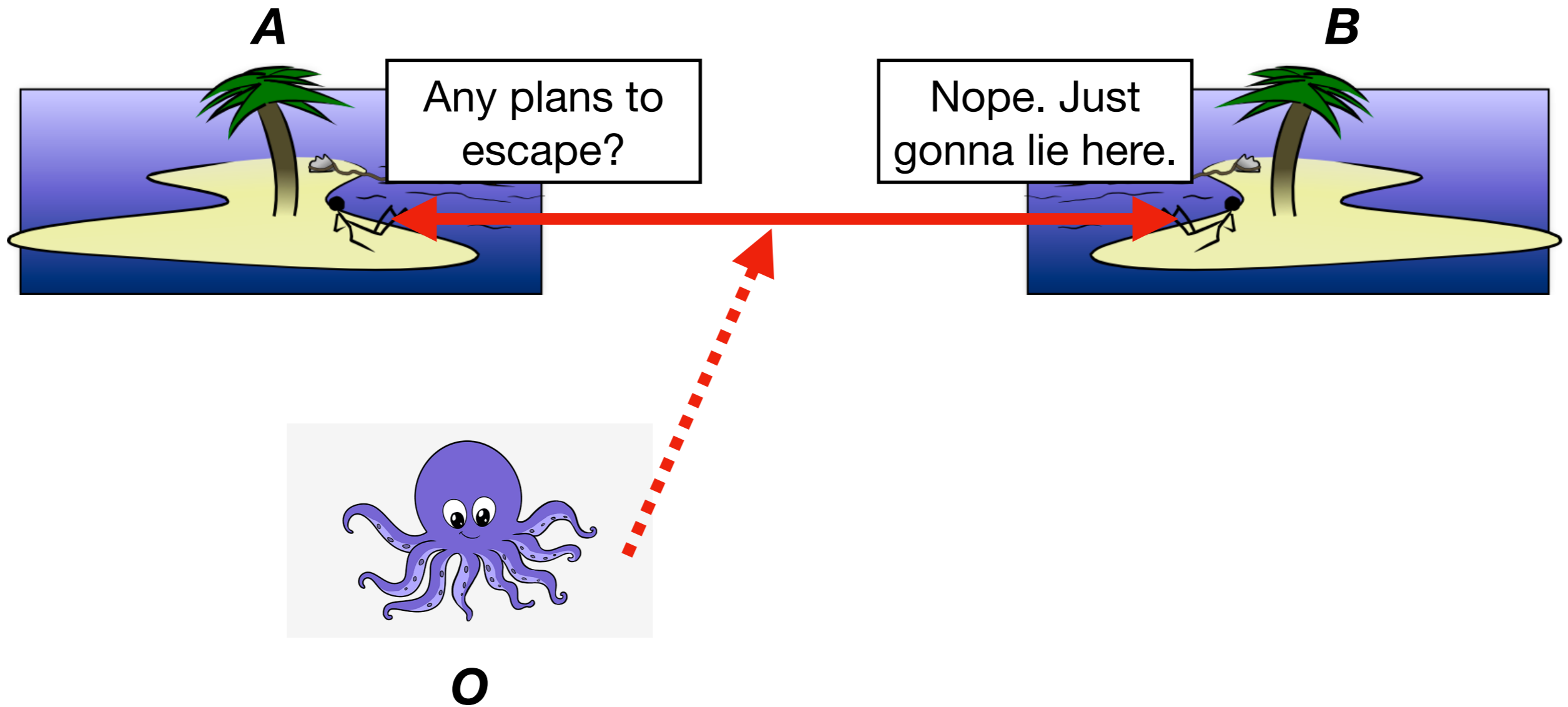
- Thought experiment (from Emily Bender):
  - Training data: All well-formed Java code on GitHub, but only the text of the code; no output; no understanding of what unit tests mean
  - Test input: A single Java program, possibly even from the training data
  - Expected output: Result of executing that program

What's missing is the *meaning*... what is the program supposed to do, given just the form (code)?

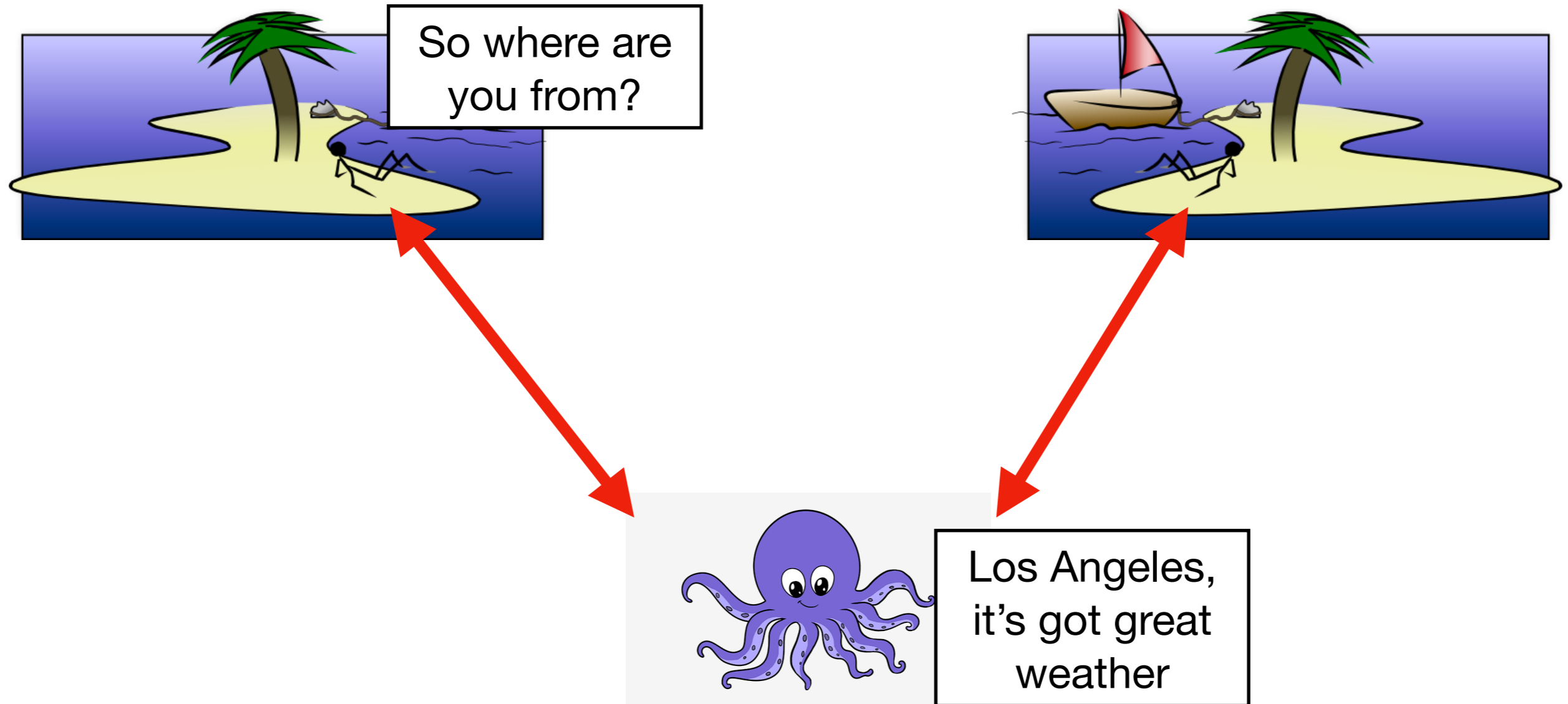
# The octopus test



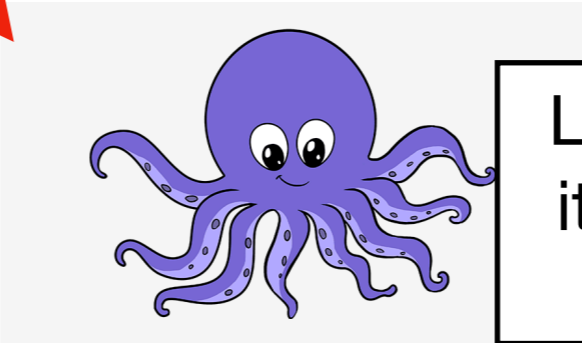
# The octopus test



# The octopus test



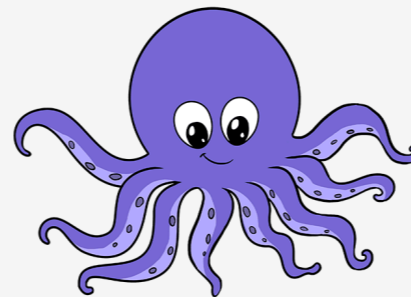
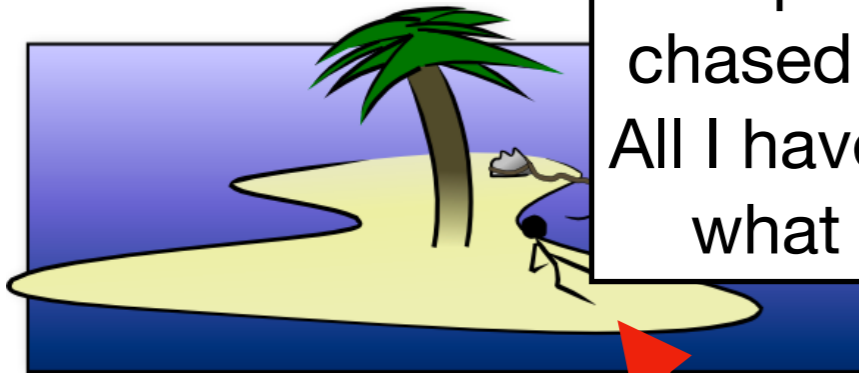
So where are you from?



Los Angeles, it's got great weather

# The octopus test

Help! I'm being  
chased by a bear!  
All I have is a stick,  
what do I do?



Not sure, sorry!

*(No idea what a bear or stick is...)*

# **O** did not learn “meaning”

- **O** only observed form, without any grounding in the world on these islands
- **A** could find meaning from **O**'s utterances, even though **O** did not “understand” what it was saying
- What if **B** didn't know what a bear was either? They might respond similarly to **O**. However, **B** can ground their responses in their own world/experience, and as such are formulating their response totally differently from **O**

# So what now?

- We need more datasets that are grounded in different modalities and ways of interaction!
- We need ways to test a model's ability to generalize or adapt to new tasks
- Take some inspiration from human language learning: children do not learn from *form* alone, why should we force our machines to do so?