COS 484

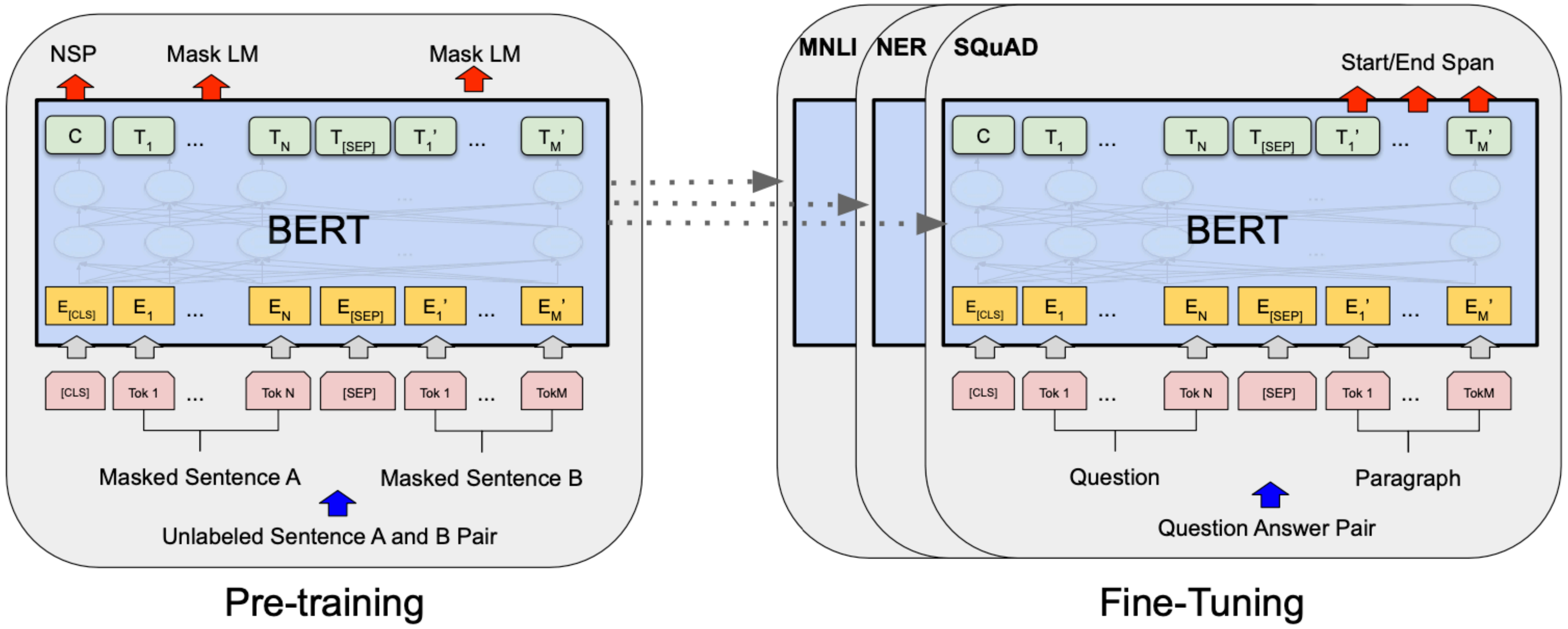
Natural Language Processing

L11: Large Language Models (LLMs)

Spring 2026

Recap: Pretraining / fine-tuning

"Pre-train" a model on a large dataset for task X, then "fine-tune" it on a dataset for task Y



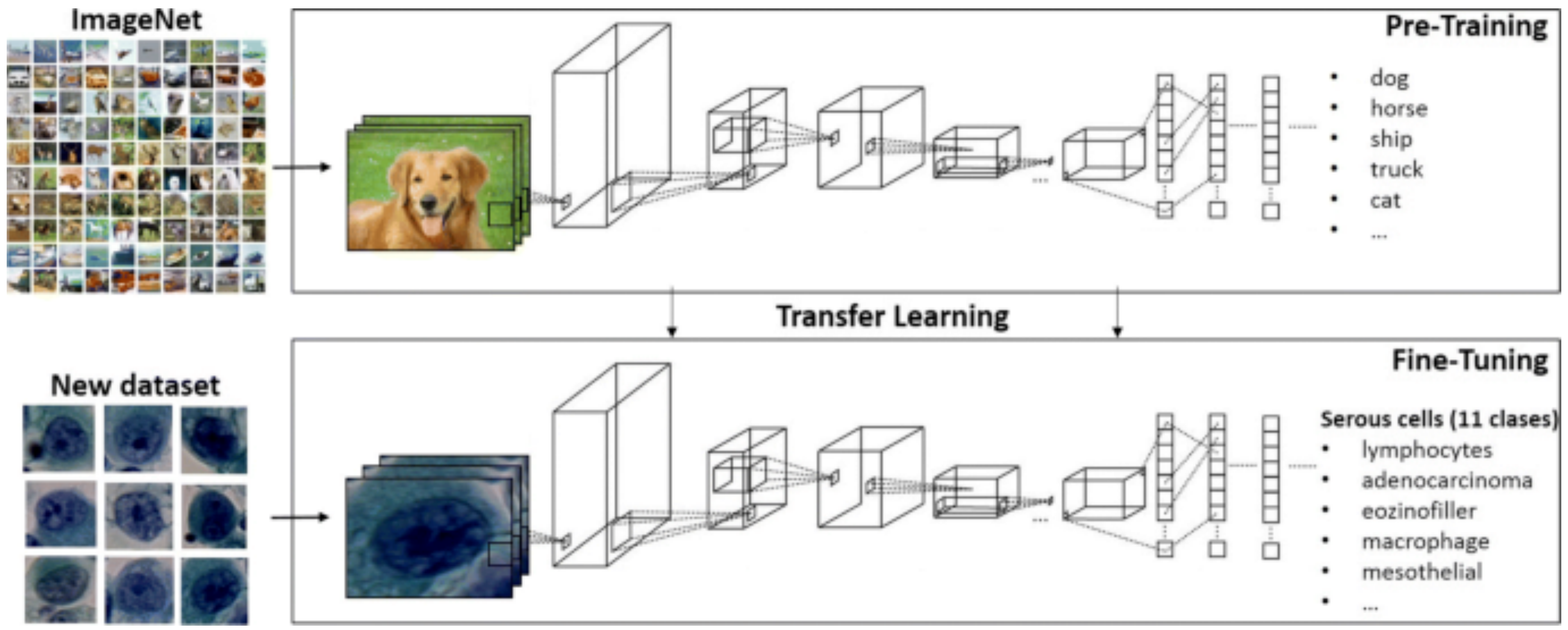
"**Fine-tuning** is the process of **taking the network learned by these pre-trained models**, and **further training the model**, often via an added neural net classifier that takes the top layer of the network as input, to perform some downstream task."

Fine-tuning is a training process and takes **gradient descent steps**!

Recap: Pretraining / fine-tuning

Pre-training

Fine-tuning



1.28M images, 1000 classes

3652 images, 11 classes

Pre-training

Natural language [MASK] (NLP) is an [MASK] subfield of linguistics, computer science, and artificial [MASK] concerned with the interactions [MASK] computers and human [MASK] … → processing, interdisciplinary, Intelligence, between, language

3.3B tokens
(512 tokens per segment)

Fine-tuning

contains no wit , only labored gags	negative
the greatest musicians	positive
very good viewing alternative	positive

67k examples, 2 classes

Recap: Pretraining / fine-tuning

Experiments on GLUE (Wang et al., 2019)

# of examples range between 2.5k and 392k examples

System	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	**Average**
	392k	363k	108k	67k	8.5k	5.7k	3.5k	2.5k	-
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

Today we are going to see other uses of pre-trained models:

1) few-shot examples (e.g., 32)
2) No fine-tuning (= no gradient updates)

This lecture

- Post-BERT models of pre-training / fine-tuning
- GPT-3: prompting and in-context learning
- Scaling laws

Post-BERT models for pre-training/fine-tuning

RoBERTa

- BERT is still under-trained
- Removed the next sentence prediction pre-training — it adds more noise than benefits!
- Trained longer with 10x data & bigger batch sizes
- Pre-trained on 1,024 V100 GPUs for one day in 2019

Model	**data**	**bsz**	**steps**	**SQuAD** (v1.1/2.0)	**MNLI-m**	**SST-2**
RoBERTa						
with BOOKS + WIKI	16GB	8K	100K	93.6/87.3	89.0	95.3
+ additional data (§3.2)	160GB	8K	100K	94.0/87.7	89.3	95.6
+ pretrain longer	160GB	8K	300K	94.4/88.7	90.0	96.1
+ pretrain even longer	160GB	8K	500K	**94.6/89.4**	**90.2**	**96.4**
$\text{BERT}_{\text{LARGE}}$						
with BOOKS + WIKI	13GB	256	1M	90.9/81.8	86.6	93.7

(Liu et al., 2019): RoBERTa: A Robustly Optimized BERT Pretraining Approach

ALBERT

Key idea: **parameter sharing** across different layers + smaller embedding sizes

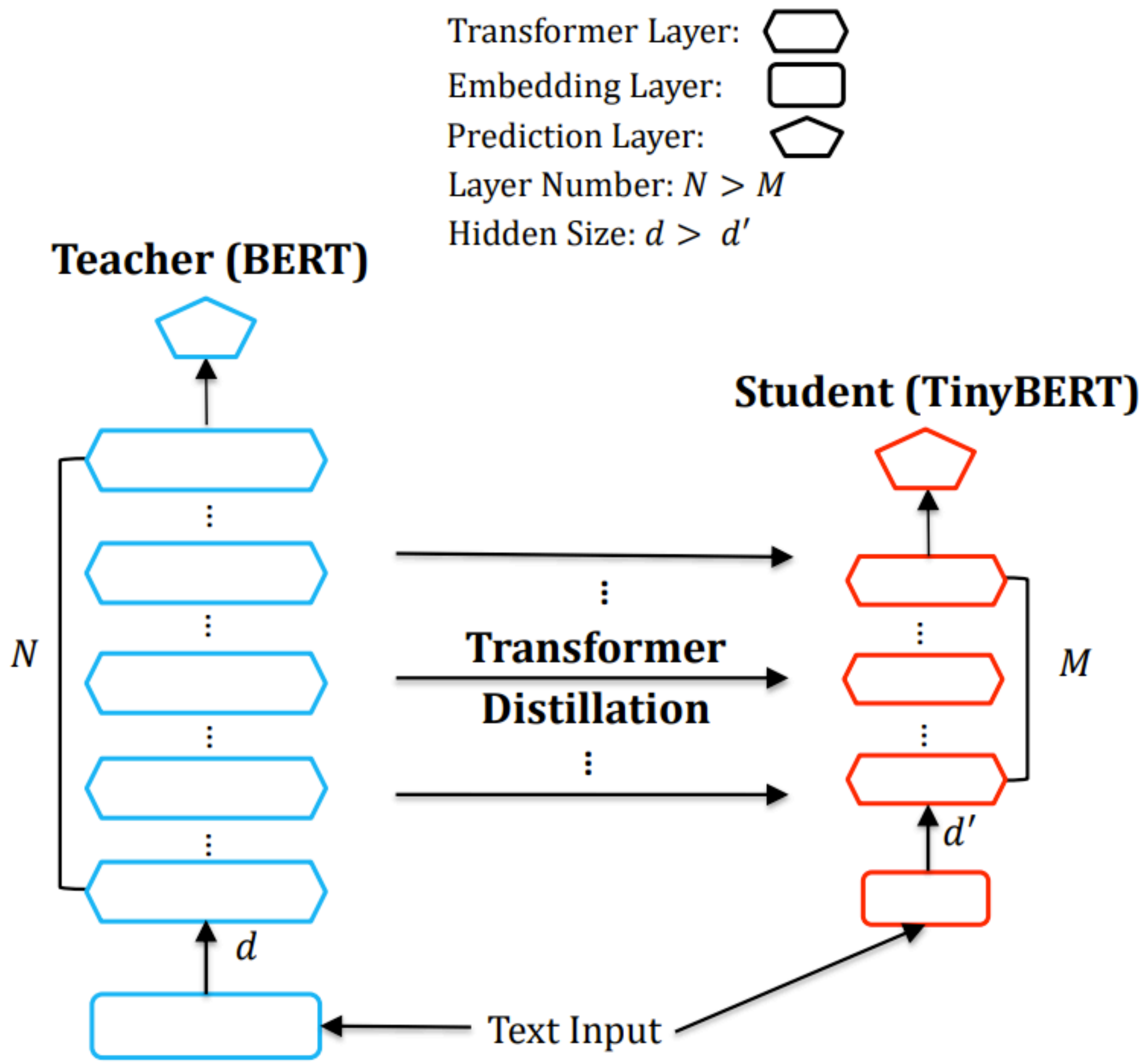
Model		Parameters	Layers	Hidden	Embedding	Parameter-sharing
BERT	base	108M	12	768	768	False
	large	334M	24	1024	1024	False
ALBERT	base	12M	12	768	128	True
	large	18M	24	1024	128	True
	xlarge	60M	24	2048	128	True
	xxlarge	235M	12	4096	128	True

Model		Parameters	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg	Speedup
BERT	base	108M	90.4/83.2	80.4/77.6	84.5	92.8	68.2	82.3	4.7x
	large	334M	92.2/85.5	85.0/82.2	86.6	93.0	73.9	85.2	1.0
ALBERT	base	12M	89.3/82.3	80.0/77.1	81.6	90.3	64.0	80.1	5.6x
	large	18M	90.6/83.9	82.3/79.4	83.5	91.7	68.5	82.4	1.7x
	xlarge	60M	92.5/86.1	86.1/83.1	86.4	92.4	74.8	85.5	0.6x
	xxlarge	235M	**94.1/88.3**	**88.1/85.1**	**88.0**	**95.2**	**82.3**	**88.7**	0.3x

AlBERT models have less # of parameters (less storage), but they can be slower because the model architectures are larger

(Lan et al., 2020): ALBERT: A Lite BERT for Self-supervised Learning of Language Representations

DistillBERT / TinyBERT / MobileBERT



https://github.com/abhilash1910/DistilBERT--SQuAD-v1-Notebook

Key idea: produce a smaller model (student) that distill information from the BERT models (teacher)

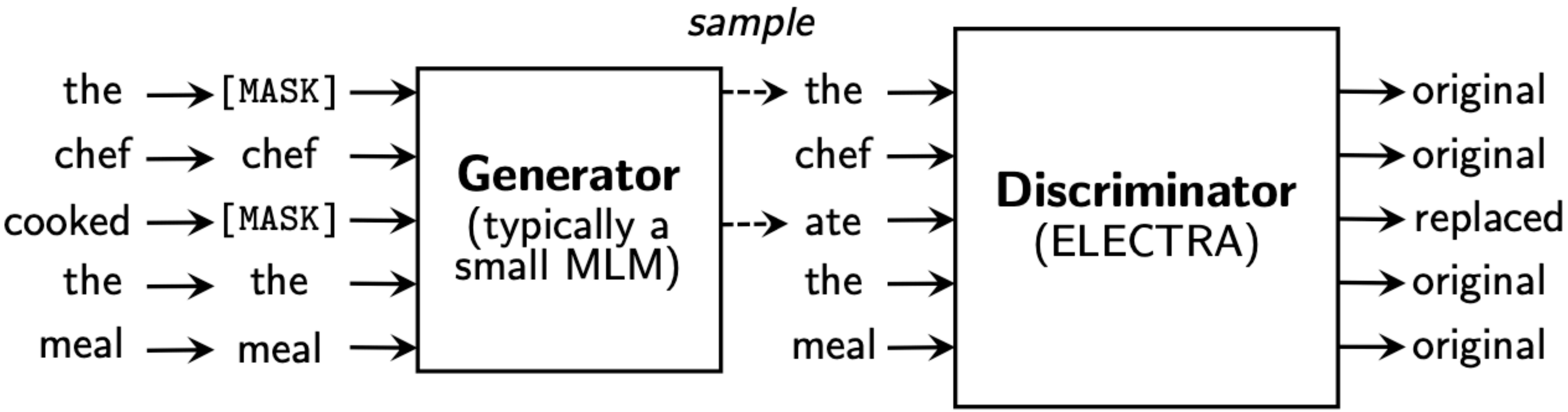
Table 1: **DistilBERT retains 97% of BERT performance.** Comparison on the dev sets of the GLUE benchmark. ELMo results as reported by the authors. BERT and DistilBERT results are the medians of 5 runs with different seeds.

Model	**Score**	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST-2	STS-B	WNLI
ELMo	68.7	44.1	68.6	76.6	71.1	86.2	53.4	91.5	70.4	56.3
BERT-base	79.5	56.3	86.7	88.6	91.8	89.6	69.3	92.7	89.0	53.5
DistilBERT	77.0	51.3	82.2	87.5	89.2	88.5	59.9	91.3	86.9	56.3

(Sanh et al., 2019): DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter

ELECTRA

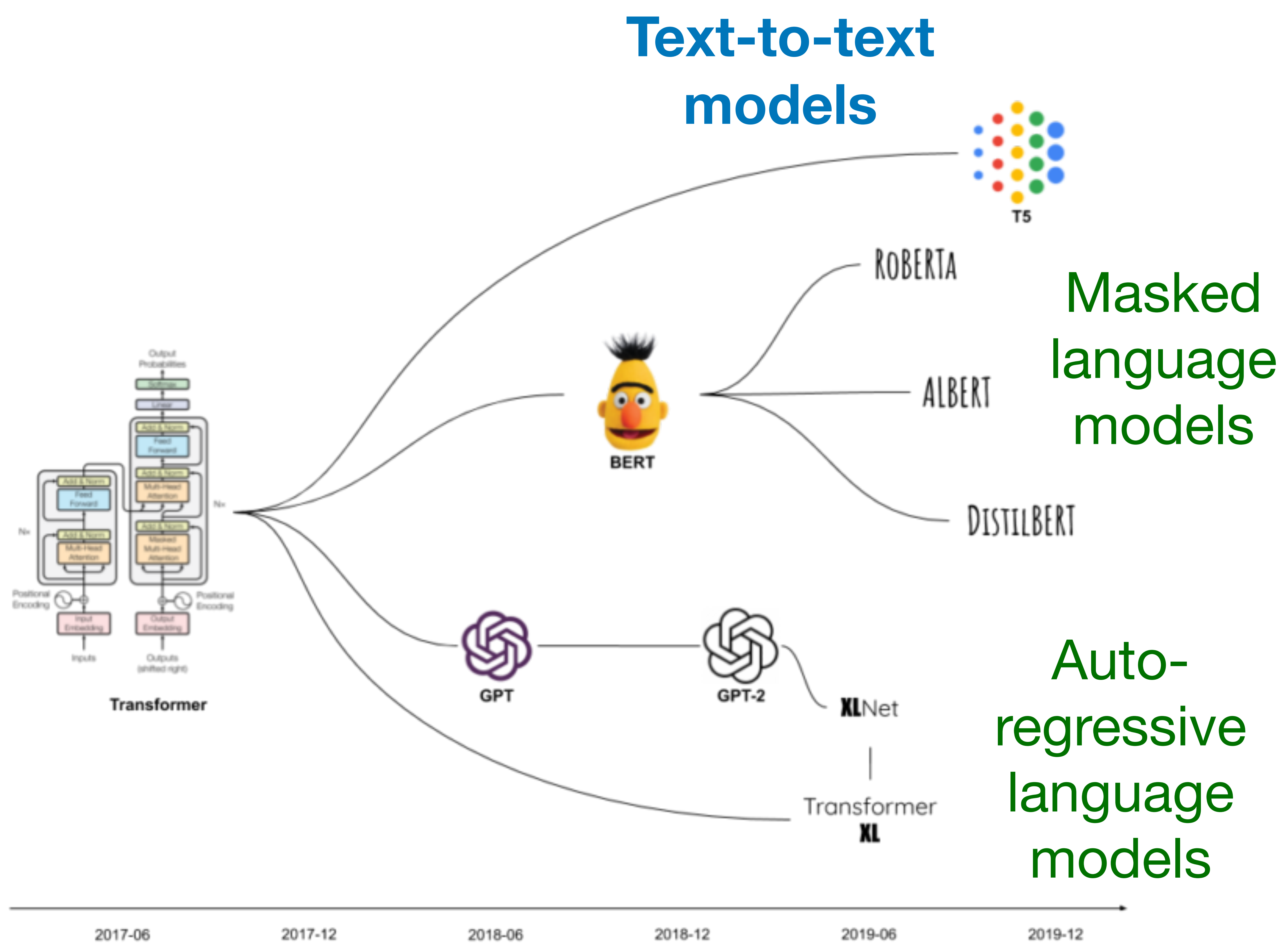
ELECTRA provides a more **efficient** training method,
because it predicts 100% of tokens (instead of 15%) every time



Only the discriminator will be used for downstream fine-tuning

(Clark et al., 2020): ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators

Three major forms of pre-training

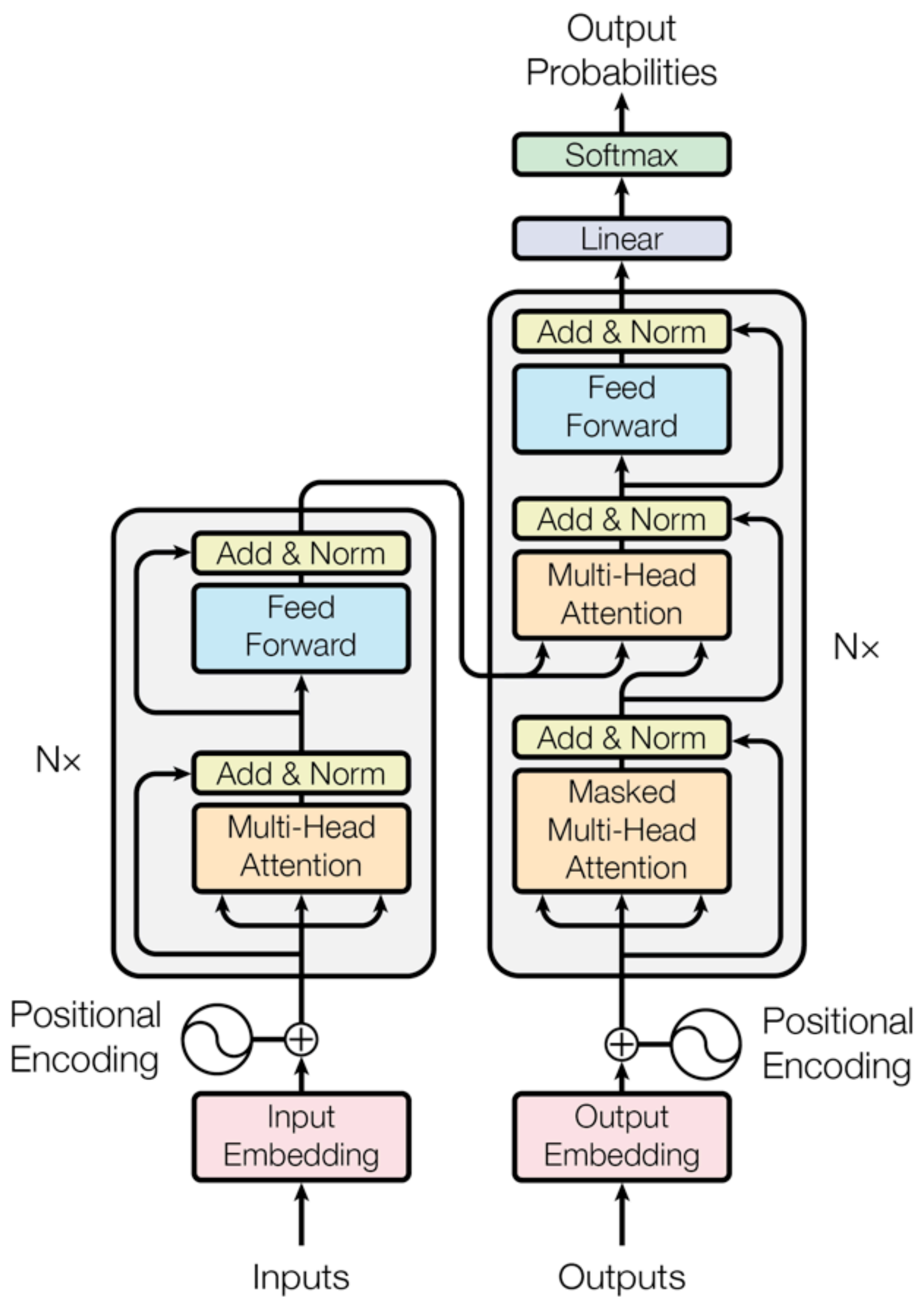


- Masked language models = Transformer encoder
- Autoregressive language models = Transformer decoder
- **Text-to-text models = Transformer encoder-decoder**

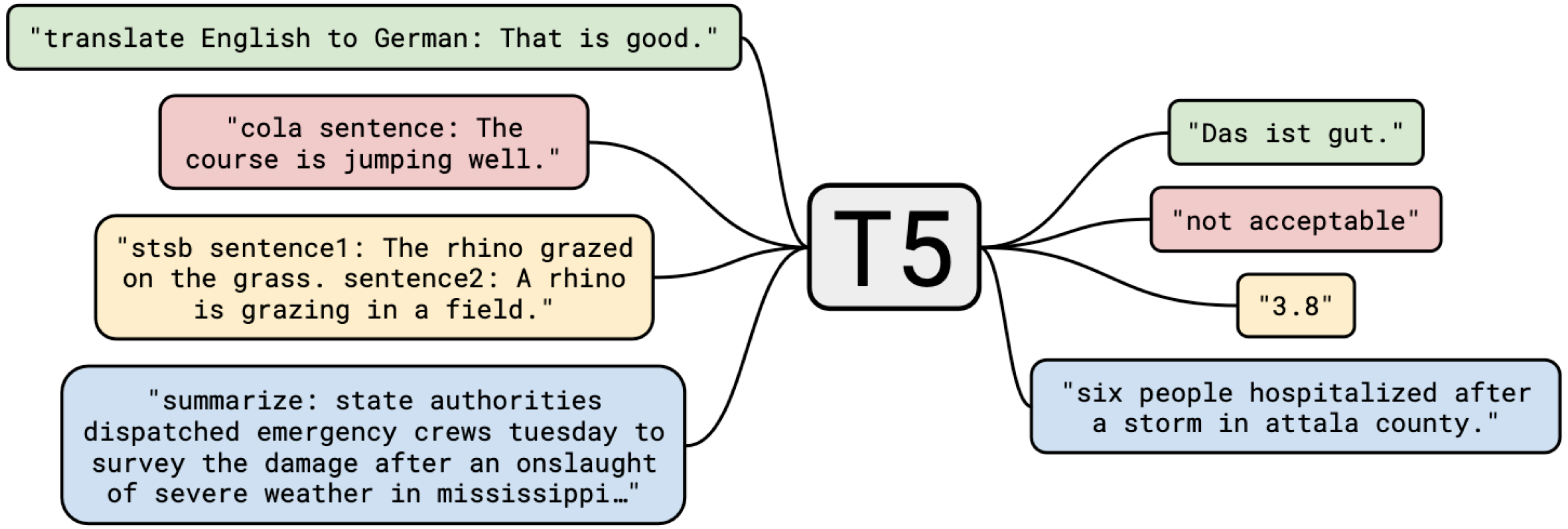
https://www.factored.ai/2021/09/21/an-intuitive-explanation-of-transformer-based-models/

Text-to-text models

- So far, **encoder-only models (e.g., BERT)** enjoy the benefits of **bidirectionality** but they can’t be used to generate text
- **Decoder-only models (e.g., GPT)** can do generation but they are left-to-right LMs..
- Text-to-text models combine the best of both worlds!

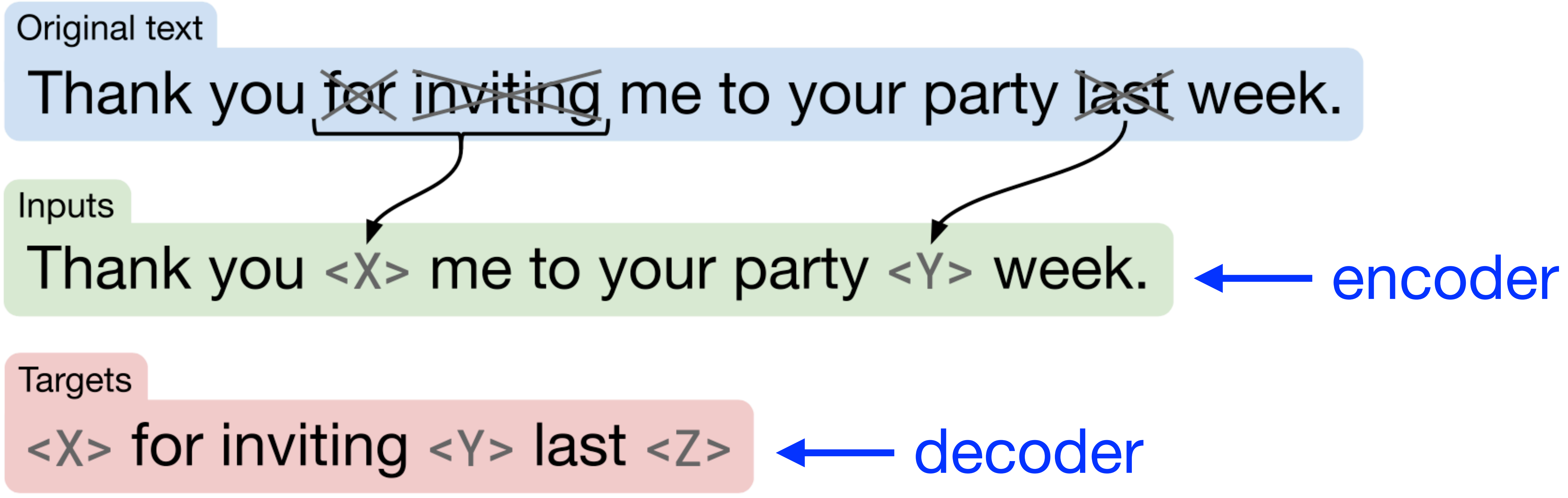


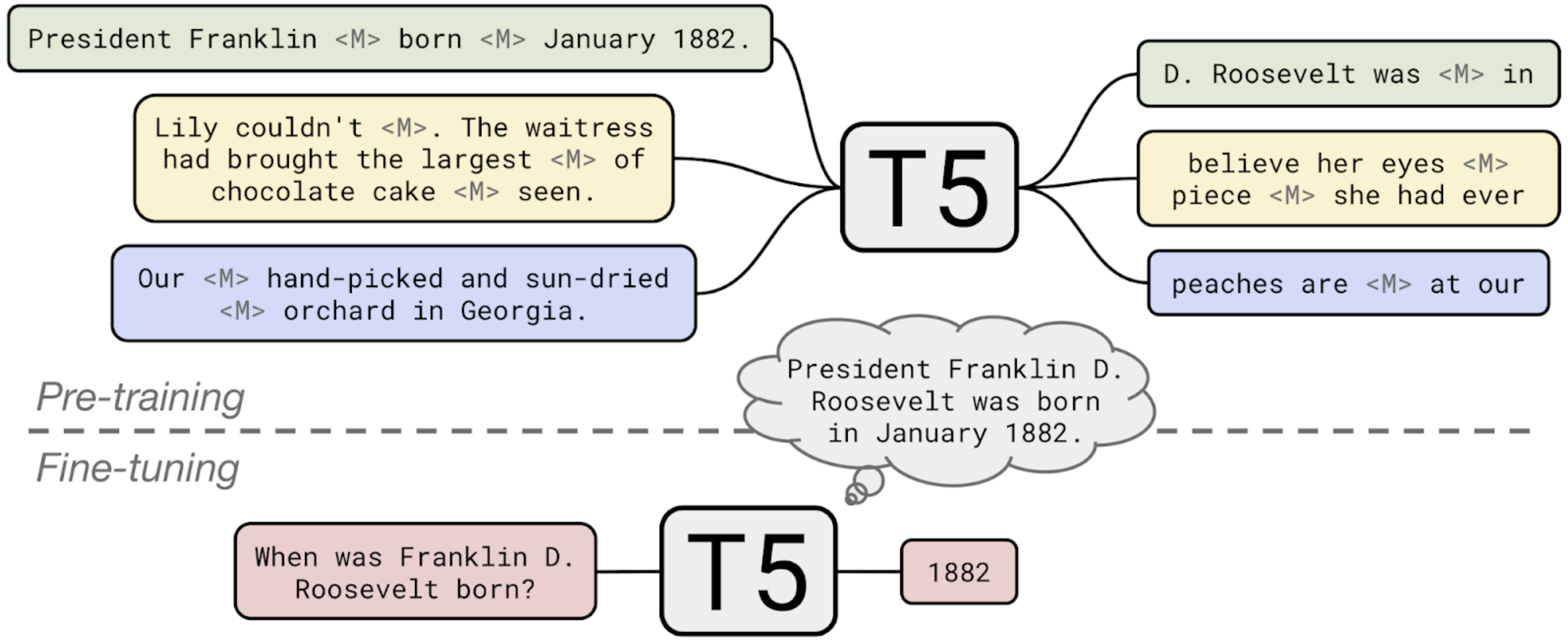
T5 = Text-to-Text Transfer Transformer



(Raffel et al., 2020): Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

T5 models





T5 comes in different sizes:

- t5-small.
- t5-base.
- t5-large.
- t5-3b.
- t5-11b.

(Raffel et al., 2020): Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

How to use these pre-trained models?

🤗 Transformers

• **Transformers**

Search documentation ⌘K

V4.27.2 EN 92,354

- CANINE
- CodeGen
- ConvBERT
- CPM
- CTRL
- DeBERTa
- DeBERTa-v2
- DialoGPT
- **DistilBERT**
- DPR
- ELECTRA

DistilBERT

All model pages distilbert | 🤗 Hugging Face Spaces

Overview

The DistilBERT model was proposed in the blog post Smaller, faster, cheaper, lighter: Introducing DistilBERT, a distilled version of BERT, and the paper DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. DistilBERT is a small, fast, cheap and light Transformer model trained by distilling BERT base. It has 40% less parameters than *bert-base-uncased*, runs 60% faster while preserving over 95% of BERT's performances as measured on the GLUE language understanding benchmark.

```
>>> from transformers import AutoTokenizer

>>> tokenizer = AutoTokenizer.from_pretrained("bert-base-cased")


>>> def tokenize_function(examples):
...     return tokenizer(examples["text"], padding="max_length", truncation=True)


>>> tokenized_datasets = dataset.map(tokenize_function, batched=True)
```

```
>>> from transformers import AutoModelForSequenceClassification

>>> model = AutoModelForSequenceClassification.from_pretrained("bert-base-cased", num_labels=5)
```

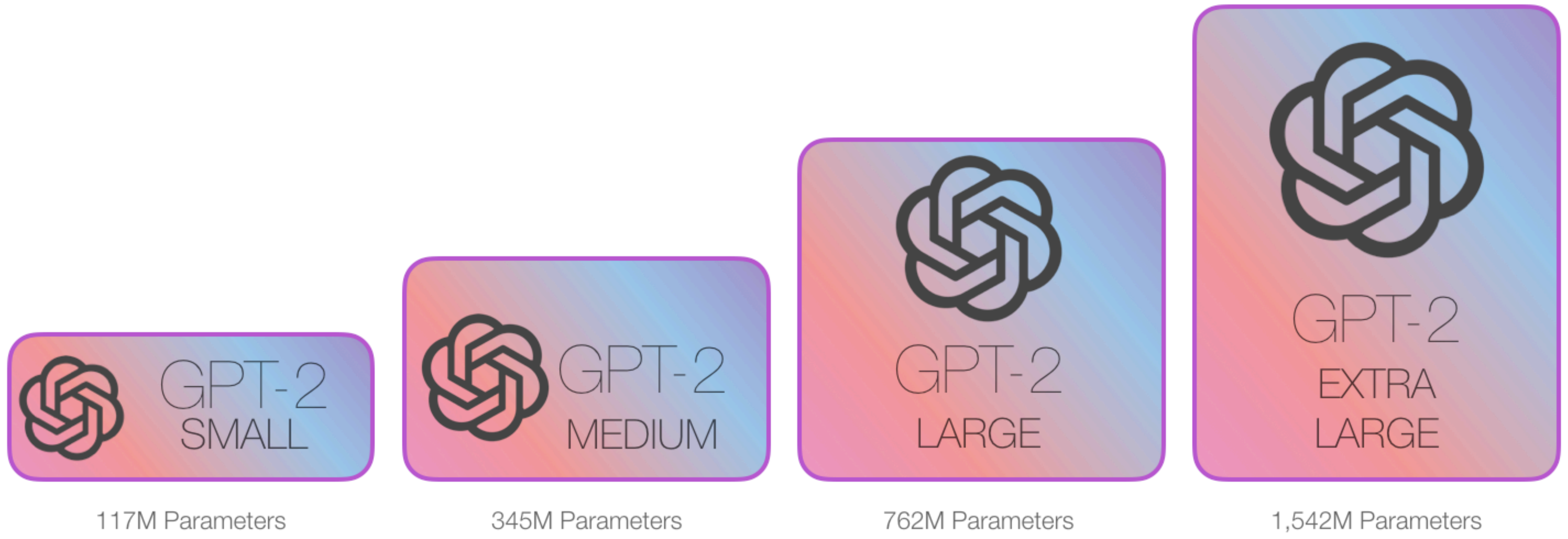
GPT-3: Prompting and In-context Learning

From GPT to GPT-2 to GPT-3

- All **decoder-only Transformer-based language models**
- Model size ↑, training corpora ↑

GPT-2

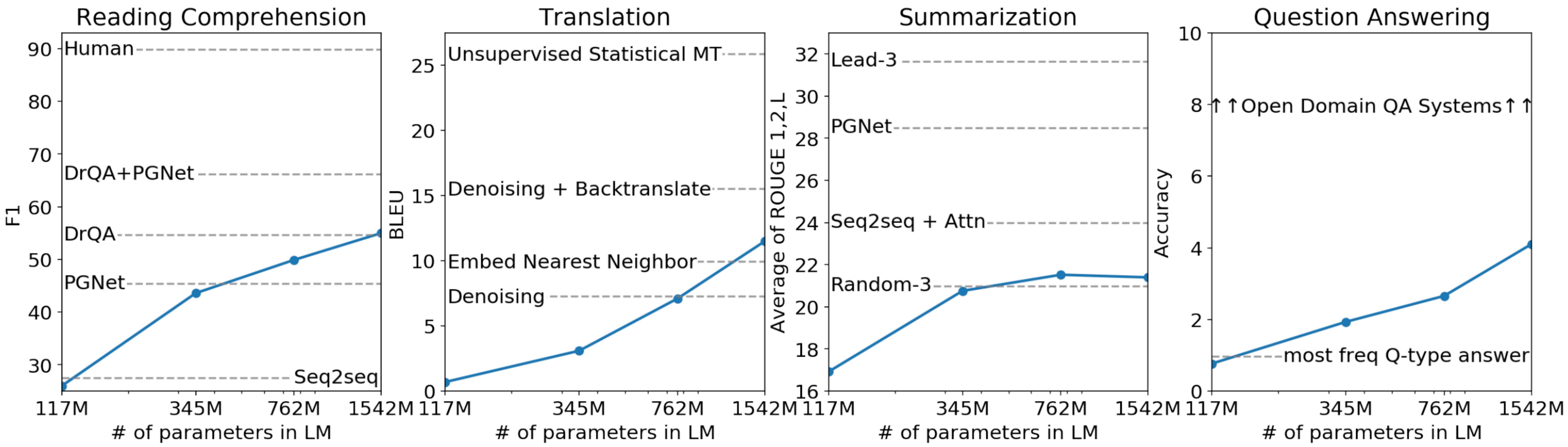
Context size = 1024



.. trained on 40Gb of Internet text ..

(Radford et al., 2019): Language Models are Unsupervised Multitask Learners

GPT-2 started to achieve strong zero-shot performance



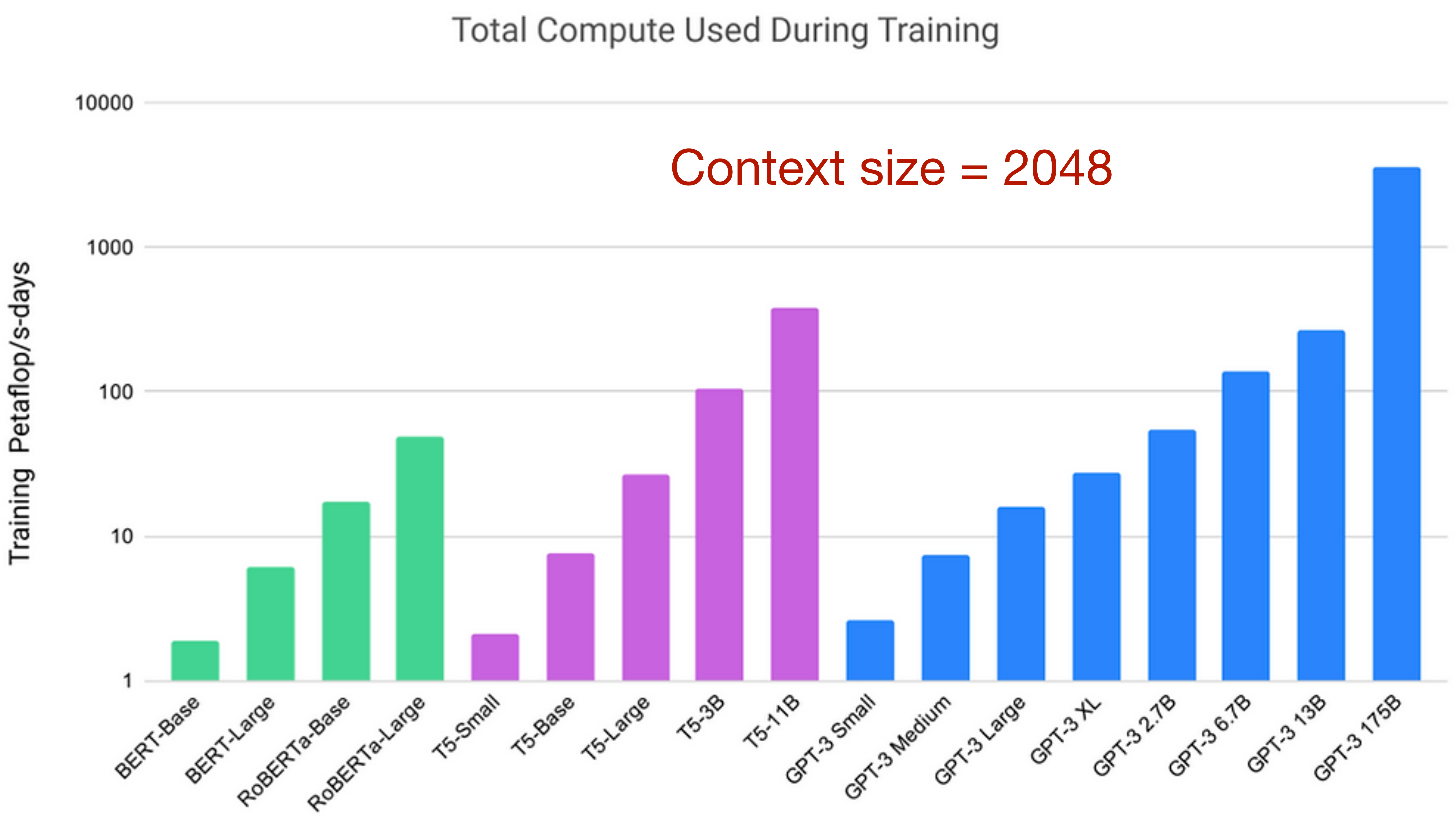
WASHINGTON - After defeating incumbent Donald Trump and Democratic candidate Joe Biden in the 2020 election, Edward Snowden has announced that his first action as President will be to declassify and release hundreds of thousands of pages of US government records about domestic surveillance operations and programs in the post-9/11 era . Snowden made the announcement in a short video address on Monday evening. He said that the release would help " move beyond the current narrative and myths of the American surveillance state to one of transparency , accountability , and truth ." The release of these records will enable a more open discussion of the US government 's surveillance practices as well as the impact that the programs had on citizens' privacy . Snowden's comments came one day after a federal judge unse aled a ruling from 2014 that the National Security Agency 's bulk collection of phone data and internet data was illegal .

https://transformer.huggingface.co/doc/gpt2-large

(Radford et al., 2019): Language Models are Unsupervised Multitask Learners

GPT-3: language models are few-shot learners

- GPT-2 → GPT-3: 1.5B → 175B (# of parameters), ~14B → 300B (# of tokens)



(Brown et al., 2020): Language Models are Few-Shot Learners

Paradigm shift since GPT-3

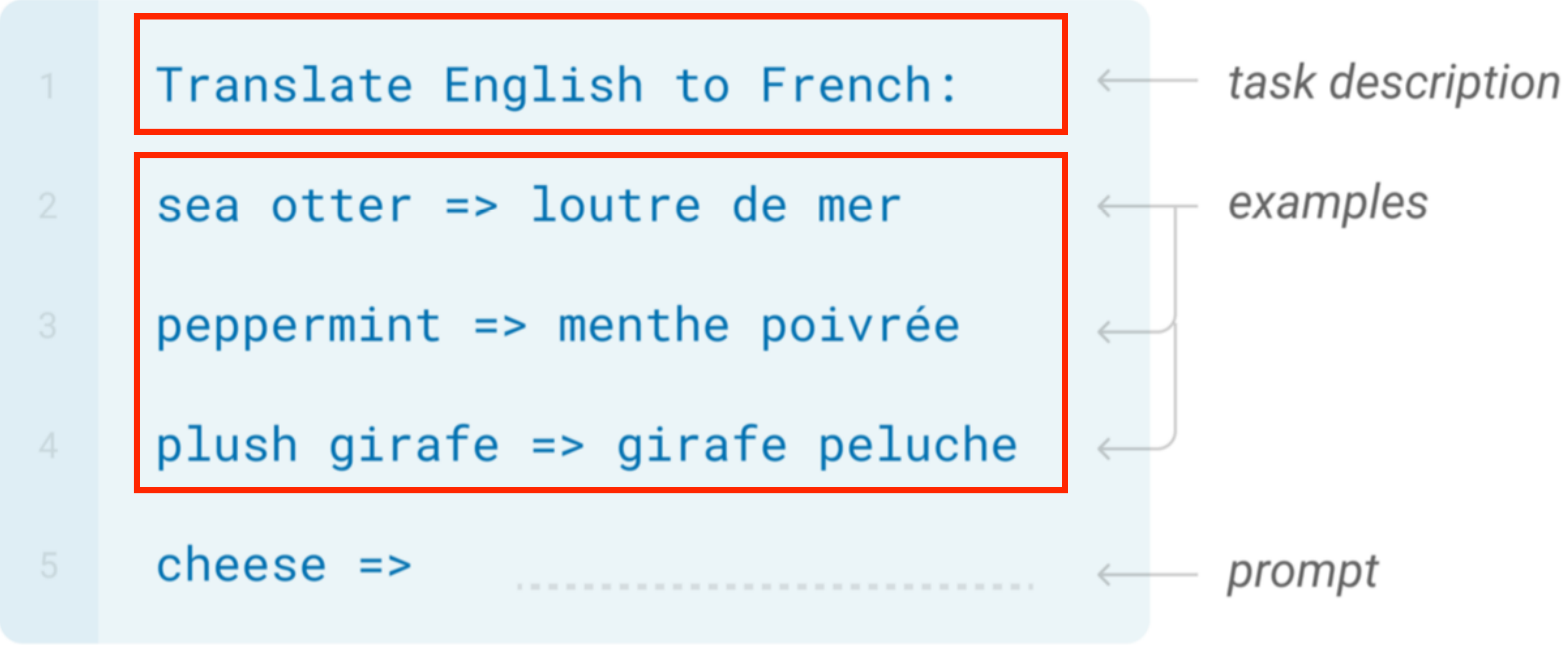
- Before GPT-3, **fine-tuning** is the default way of doing learning in models like BERT/T5/GPT-2
    - SST-2 has 67k examples, SQuAD has 88k (passage, answer, question) triples
- Fine-tuning requires computing the gradient and applying a parameter update on every example (or every K examples in a mini-batch)
- However, this is very expensive for the 175B GPT-3 model

GPT-3: Few-shot learning

- GPT-3 proposes an alternative: **in-context learning**

Few-shot

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



- This is just a forward pass, **no gradient update at all**!
- You only need to feed a small number of examples (e.g., 32)

(On the other hand, you can't feed many examples at once too as it is bounded by context size)

GPT-3: task specifications

Context →	Passage: Saint Jean de Brébeuf was a French Jesuit missionary who travelled to New France in 1625. There he worked primarily with the Huron for the rest of his life, except for a few years in France from 1629 to 1633. He learned their language and culture, writing extensively about each to aid other missionaries. In 1649, Brébeuf and another missionary were captured when an Iroquois raid took over a Huron village . Together with Huron captives, the missionaries were ritually tortured and killed on March 16, 1649. Brébeuf was beatified in 1925 and among eight Jesuit missionaries canonized as saints in the Roman Catholic Church in 1930. Question: How many years did Saint Jean de Brébeuf stay in New France before he went back to France for a few years? Answer:
Target Completion →	4

DROP
(a reading comprehension task)

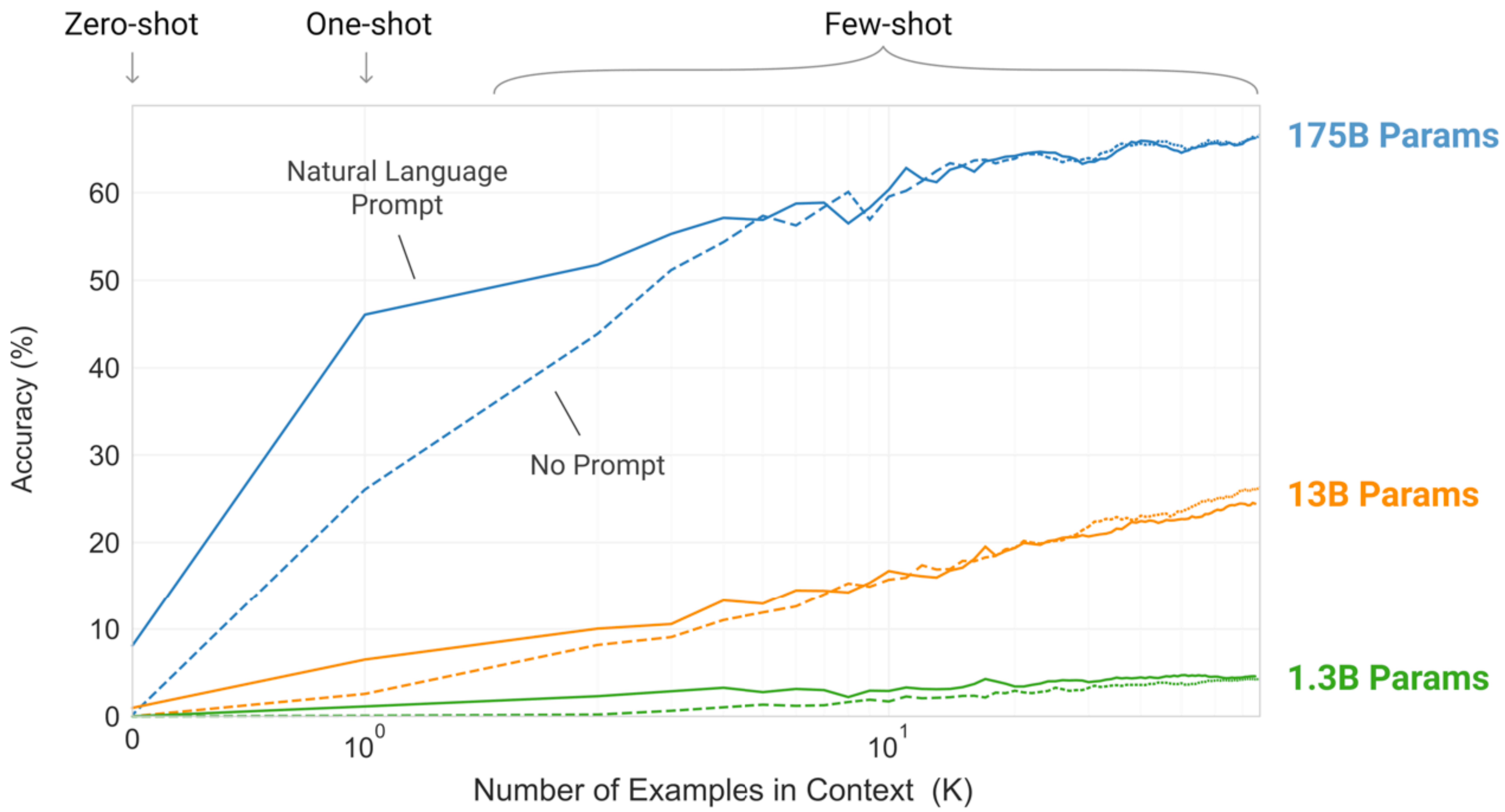
Context →	Please unscramble the letters into a word, and write that word: skicts =
Target Completion →	sticks

Unscrambling words

Context →	An outfitter provided everything needed for the safari. Before his first walking holiday, he went to a specialist outfitter to buy some boots. question: Is the word 'outfitter' used in the same way in the two sentences above? answer:
Target Completion →	no

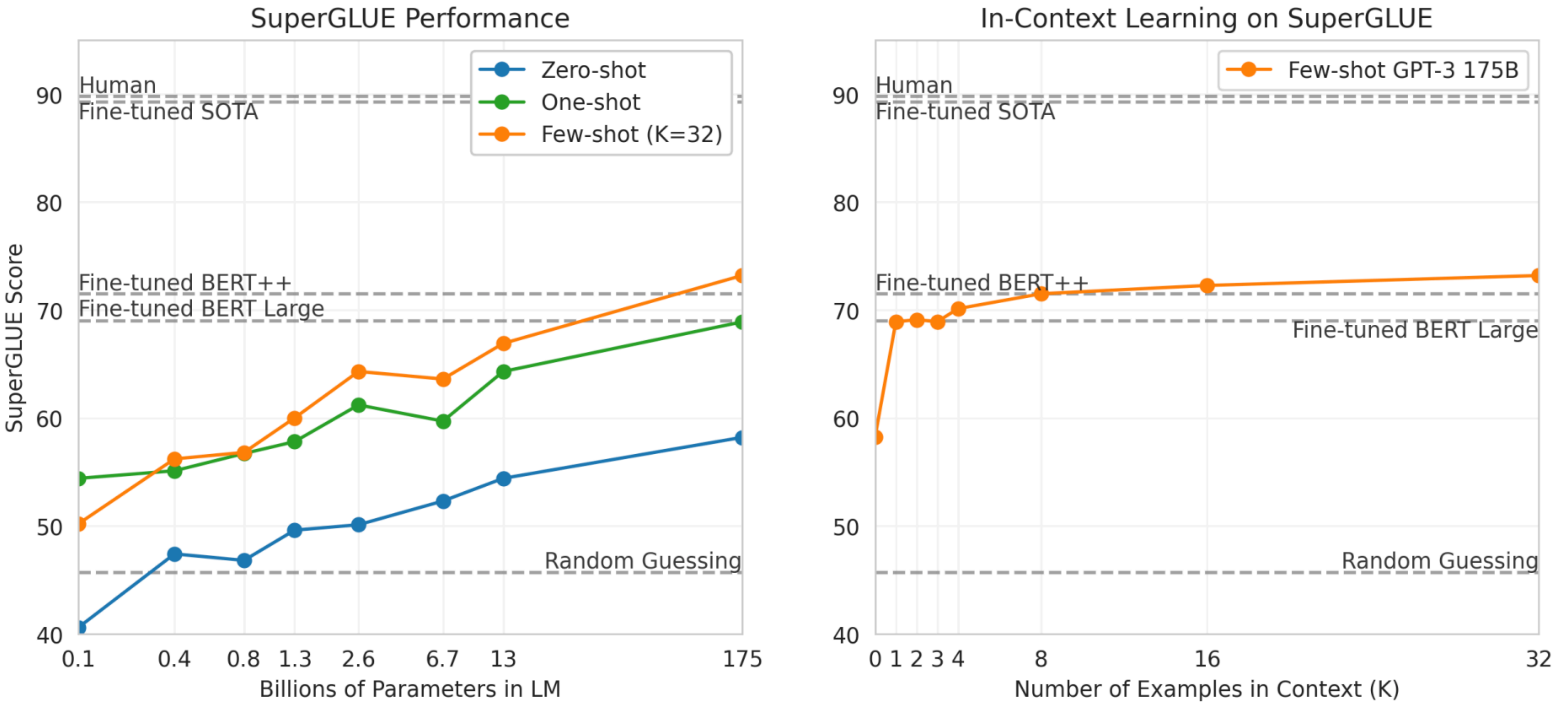
Word in context (WiC)

GPT-3's in-context learning



(Brown et al., 2020): Language Models are Few-Shot Learners

GPT-3 performance on SuperGLUE



(Wang et al., 2019) SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems

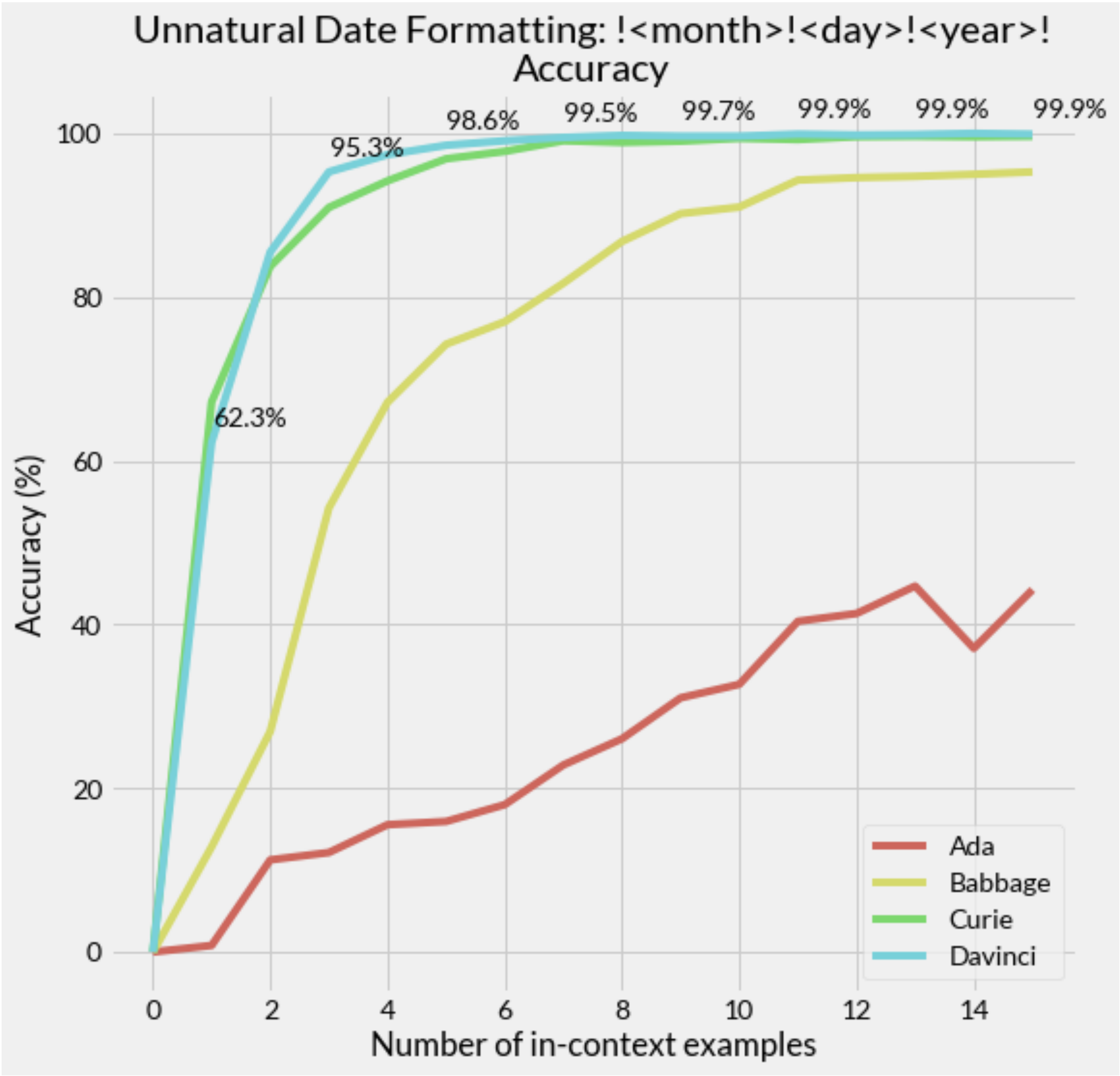
GPT-3's in-context learning

```
Input: 2014-06-01
Output: !06!01!2014!
Input: 2007-12-13
Output: !12!13!2007!
Input: 2010-09-23
Output: !09!23!2010!
Input: 2005-07-23
Output: !07!23!2005!
```

in-context examples

test example

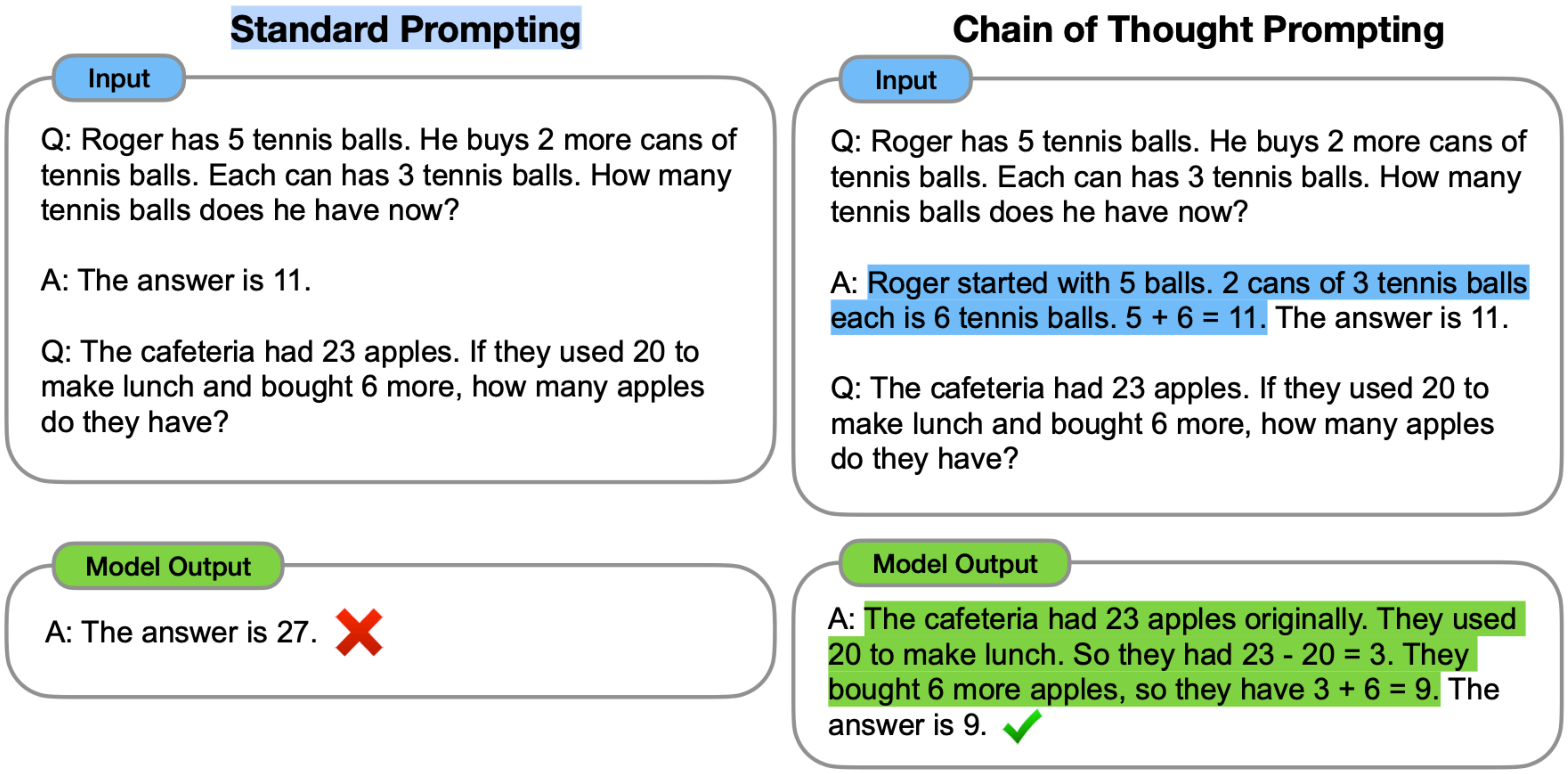
model completion



http://ai.stanford.edu/blog/in-context-learning/

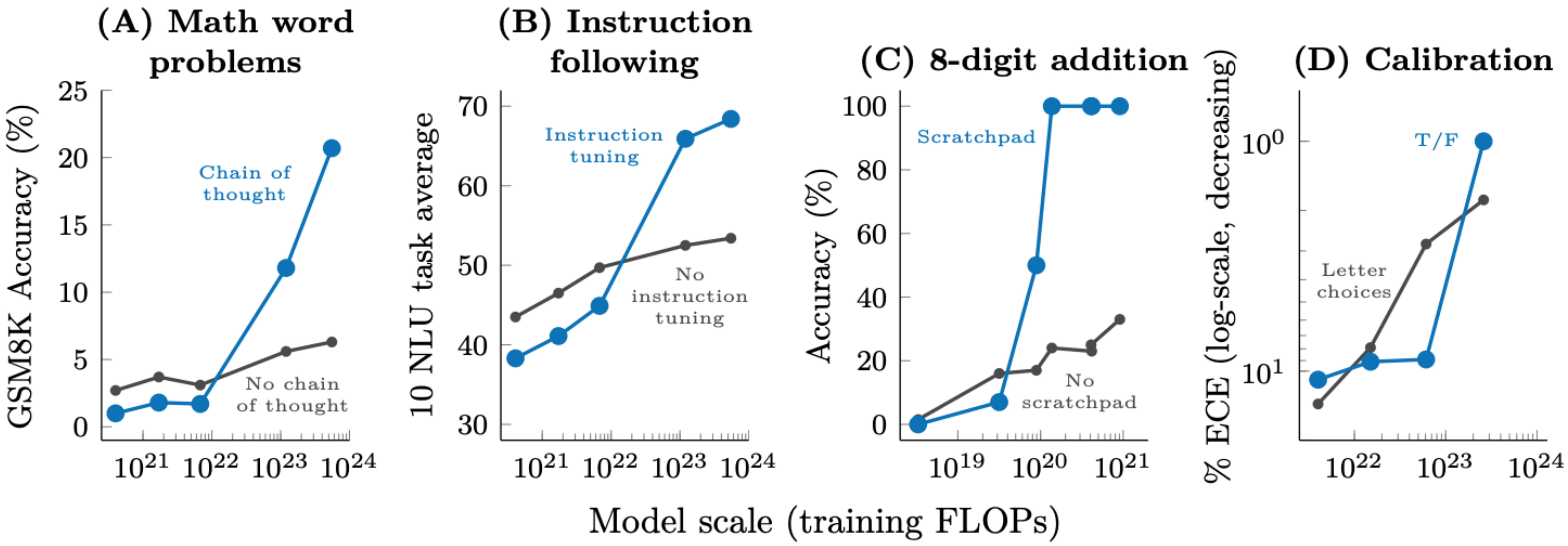
(Brown et al., 2020): Language Models are Few-Shot Learners

Chain-of-thought (CoT) prompting



(Wei et al., 2022): Chain-of-Thought Prompting Elicits Reasoning in Large Language Models

Emergent properties of LLMs

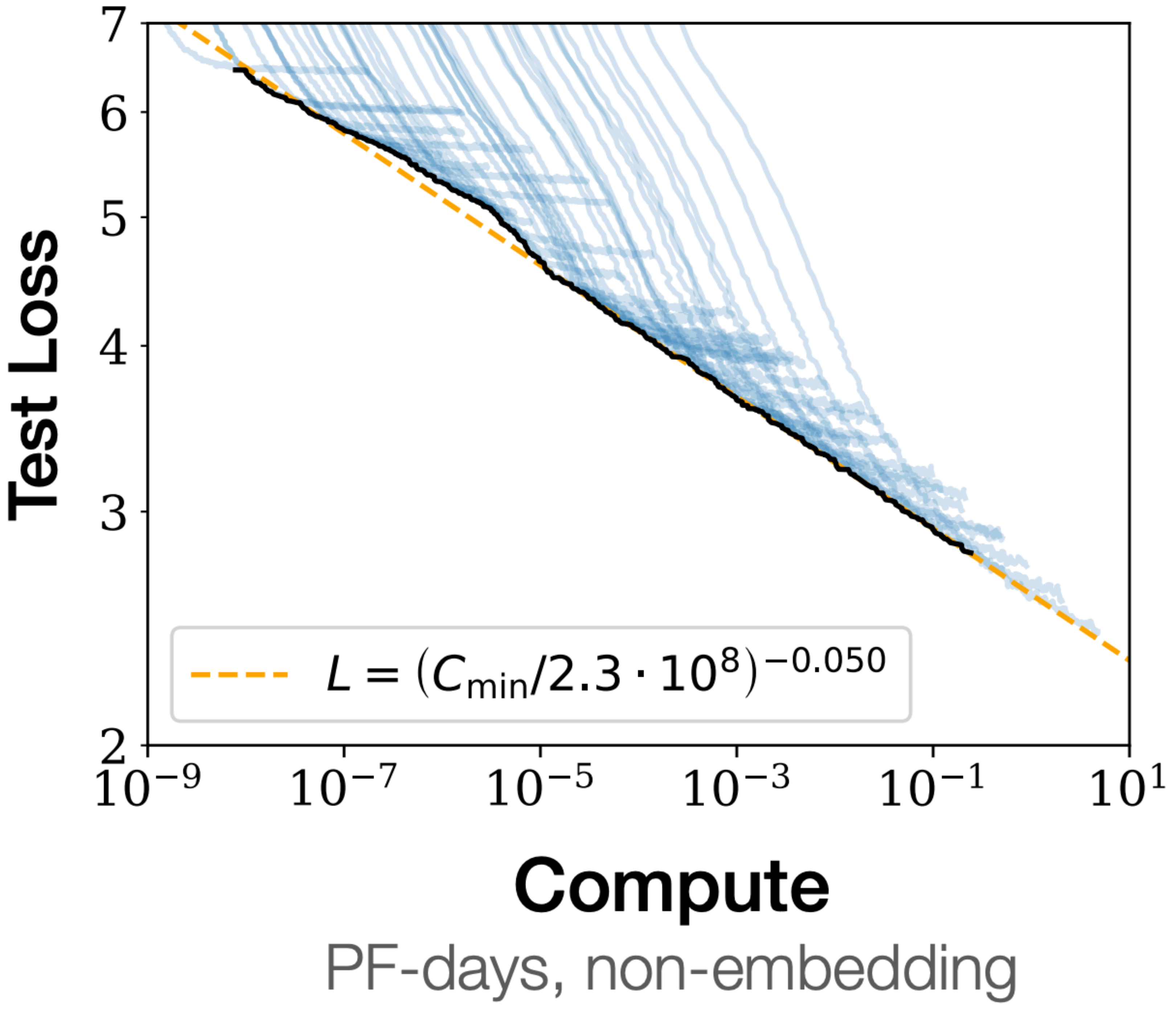


(Wei et al., 2022) Emergent Abilities of Large Language Models

What happened after GPT-3?

How to increase model size & training corpora?

Scaling Laws



Loss $\propto$ (Compute)$^{-\alpha}$ Loss $\propto$ (Data)$^{-\beta}$ Loss $\propto$ (Model params)$^{-\gamma}$

Loss goes down predictably wrt compute, data, model size!

(Kaplan et al., 2020) Scaling Laws for Neural Language Models

Chinchilla Scaling Laws: How to Optimally Allocate Compute: Model Params vs Dataset Size

$$\hat{L}(N, D) \triangleq E + \frac{A}{N^{\alpha}} + \frac{B}{D^{\beta}}.$$

L: loss
N: number of params
D: dataset size
E, A, B, α, β: fit based on data

Model	Size (# Parameters)	Training Tokens
LaMDA (Thoppilan et al., 2022)	137 Billion	168 Billion
GPT-3 (Brown et al., 2020)	175 Billion	300 Billion
Jurassic (Lieber et al., 2021)	178 Billion	300 Billion
Gopher (Rae et al., 2021)	280 Billion	300 Billion
MT-NLG 530B (Smith et al., 2022)	530 Billion	270 Billion
Chinchilla	70 Billion	1.4 Trillion

Rule of thumb: Increase dataset size proportional to model size
(e.g. 20 token per param)

(Hoffmann et al., 2022) Training Compute-Optimal Large Language Models

Open-Weight Models

RESEARCH

Introducing LLaMA: A foundational, 65-billion-parameter large language model

February 24, 2023

- **Smaller models** trained on **1.4T**, high-quality & publicly available data
- "LLaMA-13B outperforms GPT-3 (175B) on most benchmarks, and LLaMA-65B is competitive with the best models, Chinchilla-70B and PaLM-540B"

(Touvron et al., 2023): LLaMA: Open and Efficient Foundation Language Models

Recent models are trained for much longer

- Llama-3: 8B, 70B, 405B trained on **15T** tokens
- Qwen-2.5: 0.5B, 1.5B, 3B, 7B, 14B, 32B, 72B trained on **18T** tokens
- DeepSeek V3: 671B (37B active) trained on **14.8T** tokens

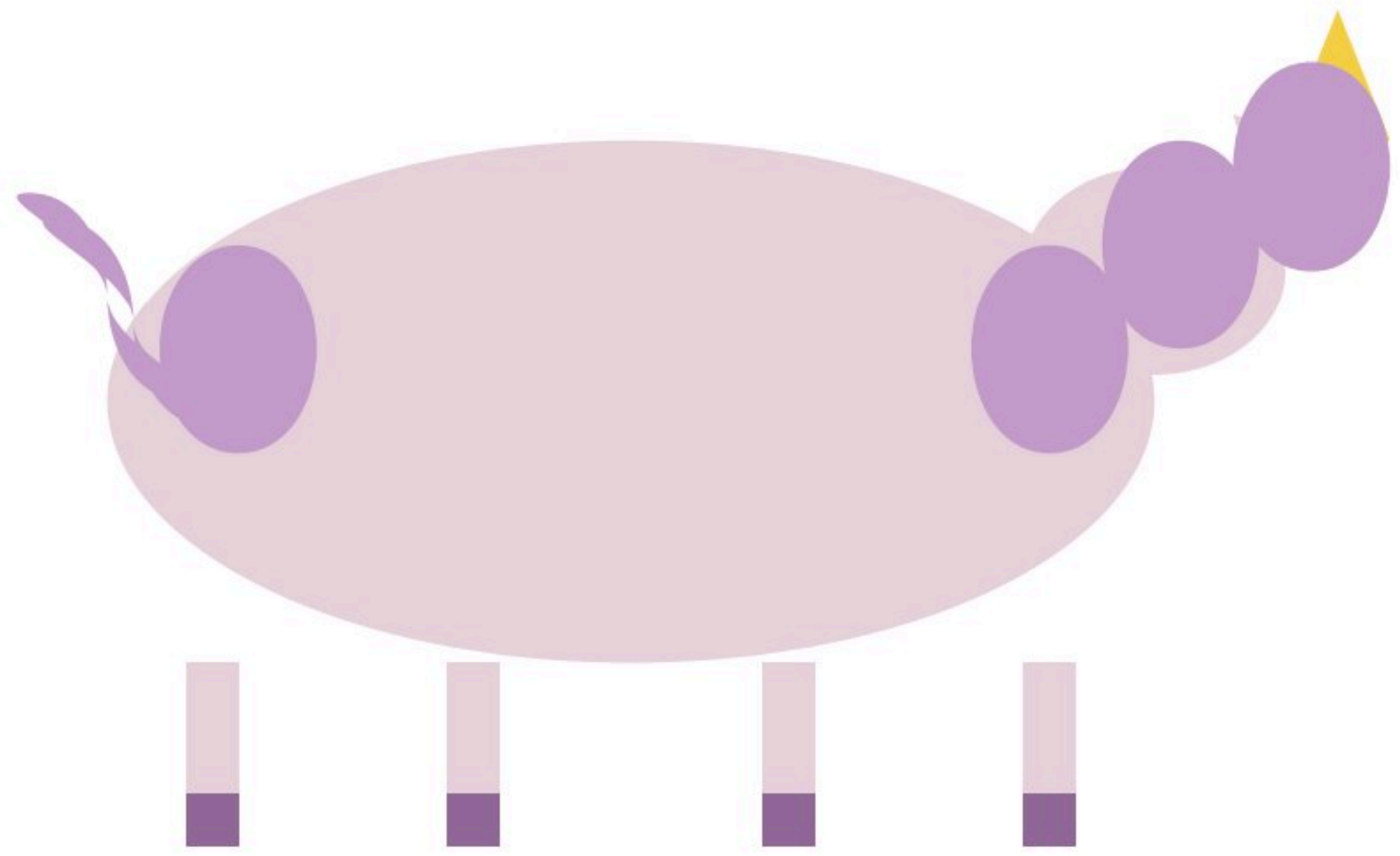
Optimize for Inference: “over-trained” smaller models are faster during inference

Multi-Modal: GPT-4

- GPT-4: a **multi-modal** model capable of processing image and text inputs and producing text outputs.
- Model size and training details unknown
- Can process up to 32k context size

Prompt: Draw a unicorn in TiKZ.

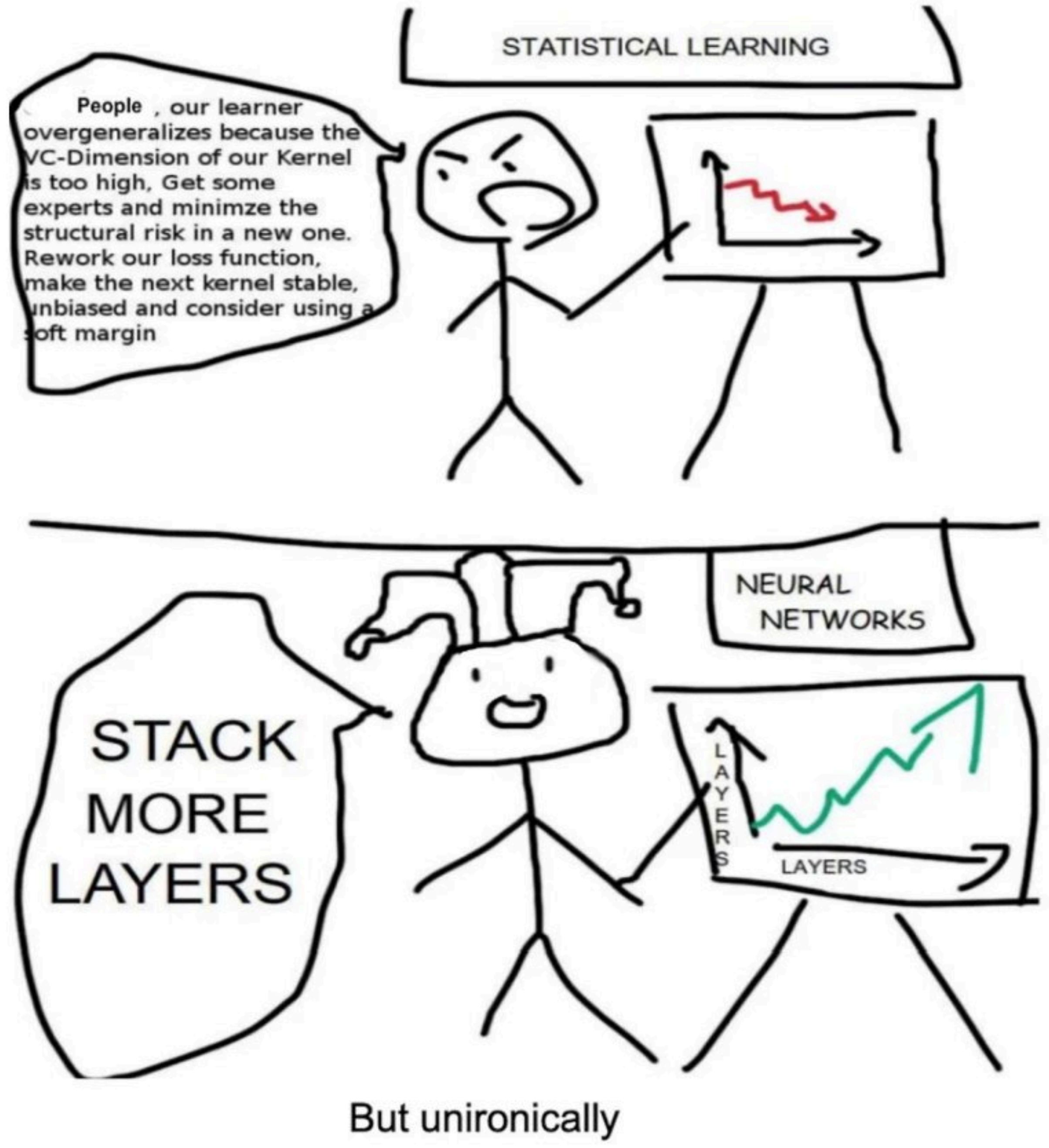
GPT-4: [Produces LaTeX compiling to following picture.]



GPT-4

GPT-4 visual input example, Moar Layers:

User Can you explain why this is funny. Think about it step-by-step.



GPT-4 The comic is satirizing the difference in approaches to improving model performance between statistical learning and neural networks.

In statistical learning, the character is shown to be concerned with overfitting and suggests a series of complex and technical solutions, such as minimizing structural risk, reworking the loss function, and using a soft margin.

In contrast, the neural networks character simply suggests adding more layers to the model. This is often seen as a common solution to improving

GPT-4 visual input example, Extreme Ironing:

User What is unusual about this image?

Source: https://www.barnorama.com/wp-content/uploads/2016/12/03-Confusing-Pictures.jpg

GPT-4 The unusual thing about this image is that a man is ironing clothes on an ironing board attached to the roof of a moving taxi.