

Generative Pre-Trained Transformers

GPT-3 (2020) & X-shot learning



Learning goals

- Recap GPT and the ideas behind standard language modeling
- Understand the difference between fine-tuning and X-shot learning

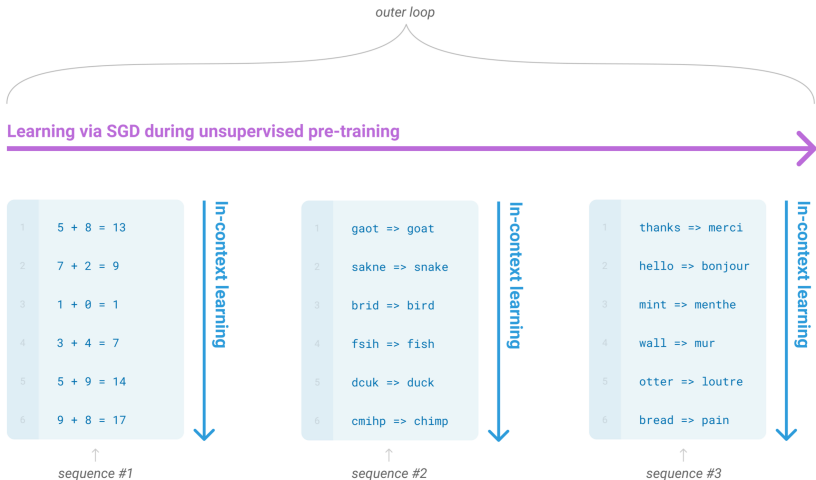
GPT RECAP

- Like BERT, GPTs are called “language models”.
- Like BERT, GPTs are trained on huge corpora, actually on even huger (closed-source) corpora.
- Like BERT, GPTs are based on the transformer architecture.
- Unlike GPTs, BERT is *not* a language model in the conventional sense, i.e. not an ARLM
- BERT instead relies on the cloze tasks, i.e. it is an MLM
- GPTs are conventional ARLMs: they are just trained on predicting the next word (or subword).

GPT RECAP

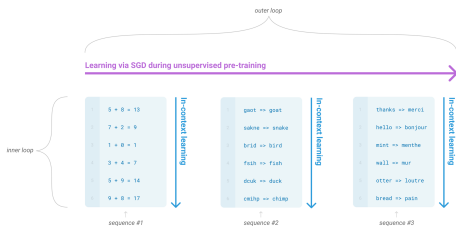
- Difference 1: GPTs rely on the transformer decoder
 - They are called “*generative*” Large Language Models (LLM)
 - BERT relies on the encoder, hence not perfectly suited for generation
- Difference 2: GPT is a **single model** that aims to solve **all tasks**.
 - It can also switch back and forth between tasks and solve tasks within tasks, another human capability that is important in practice. “**fluidity**”
- Difference 3: GPT leverages **task descriptions**.
- Difference 4: GPT is effective at **few-shot learning**.

IN-CONTEXT LEARNING



► Source: Brown et al., 2020

IN-CONTEXT LEARNING



► Source: Brown et al., 2020

- **Pre-training (Outer loop):**

- Model develops broad set of skills and abilities
- Trained via gradient descent

- **Inference (Inner loop):**

- It uses these abilities to rapidly adapt to a desired task
(= *in-context learning*)
- Just a single forward pass w/o any gradient updates

LEARNING IN BERT & CO.

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.




► Source: Brown et al., 2020

ZERO-SHOT LEARNING

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 rectangular box containing two lines of text. The first line is labeled '1' and reads 'Translate English to French:'. The second line is labeled '2' and reads 'cheese =>'. To the right of the box, there are two arrows pointing to the lines. The first arrow points to the first line and is labeled 'task description'. The second arrow points to the second line and is labeled 'prompt'. Below the second line of text, there is a horizontal dashed line.

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

← *task description*

← *prompt*

► Source: Brown et al., 2020

- No gradient updates
- Learning happens “on the fly”
- Model has to “understand” the task description

ONE-SHOT LEARNING

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

.....

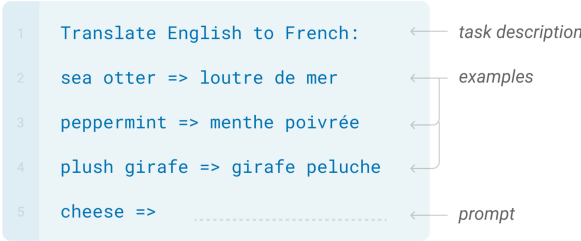
► Source: Brown et al., 2020

- No gradient updates
- Model has to “understand” the task description
- Understanding supported by *one* demonstration

FEW-SHOT LEARNING

Few-shot

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



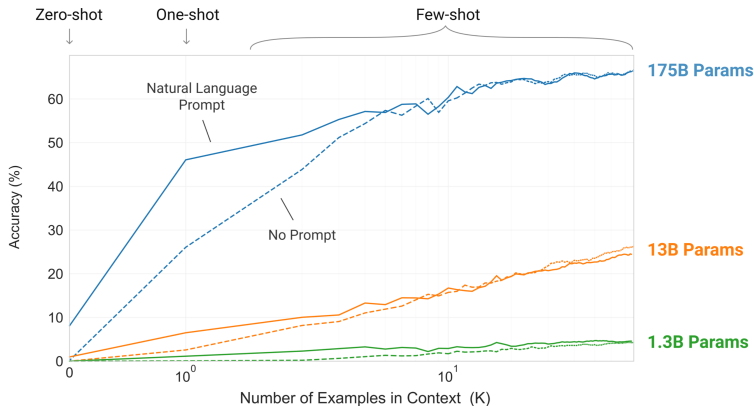
The diagram illustrates the structure of a few-shot learning prompt. It consists of five lines of text, each preceded by a number in a light blue box. The first line is the task description. The next three lines are examples of the task. The final line is the prompt to be completed. Arrows on the right side point to each line, with labels: 'task description' for the first line, 'examples' for the next three lines (indicated by a bracket), and 'prompt' for the final line.

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

► Source: Brown et al., 2020

- No gradient updates
- Model has to “understand” the task description
- Understanding supported by *few* demonstrations

EFFECTIVE IN-CONTEXT LEARNING*



► Source: Brown et al., 2020

- Number/Selection of demonstrations = Hyperparameter
- Larger model → Better in-context learning capabilities

*on an artificial, simple word scrambling/manipulation task

ARCHITECTURE

- Various sizes released; GPT-3 usually refers to largest one
- Both width (d_{model}) and depth (n_{layers}) are scaled
- Number of heads adjusted to d_{model}

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}

► Source: Brown et al., 2020

TRAINING CORPUS

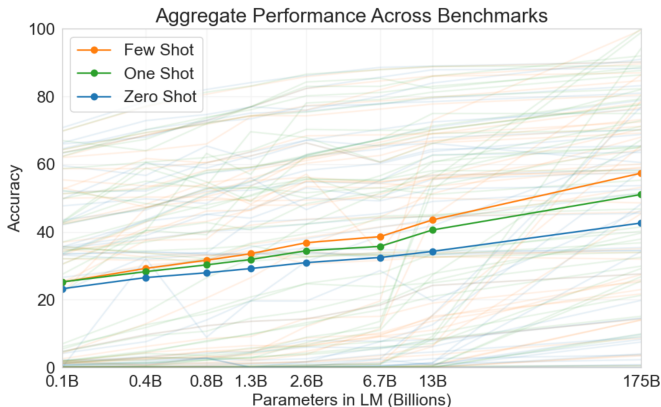
- BERT: 3.3B words (roughly 4B – 6B tokens)
- GPT-3: 499B tokens
- Increased by two orders of magnitude within < 2yrs
- Content: Mostly the internet (incl. test sets of popular benchmarks)

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

► Source: Brown et al., 2020

X-SHOT COMPARISON

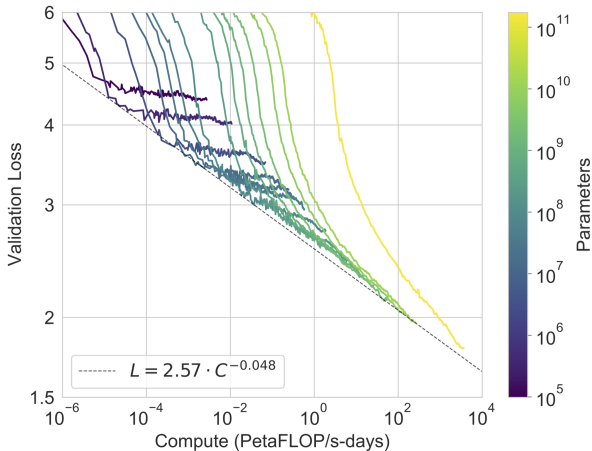
- 42 accuracy-denominated benchmarks
- Few-shot performance increases faster than zero-shot



► Source: Brown et al., 2020

LOSS AS A FUNCTION OF COMPUTE

Power-law trend



► Source: Brown et al., 2020