



GPT- 2 + 3

Language Models are Unsupervised
Multitask Learners

DngBack

GPT 2

Key Idea

- Unidirectional like GPT v1
- Compared with GPT v1
 - Larger Model
 - Larger unlabeled dataset
 - No Finetuning (using zero-shot transfer instead)

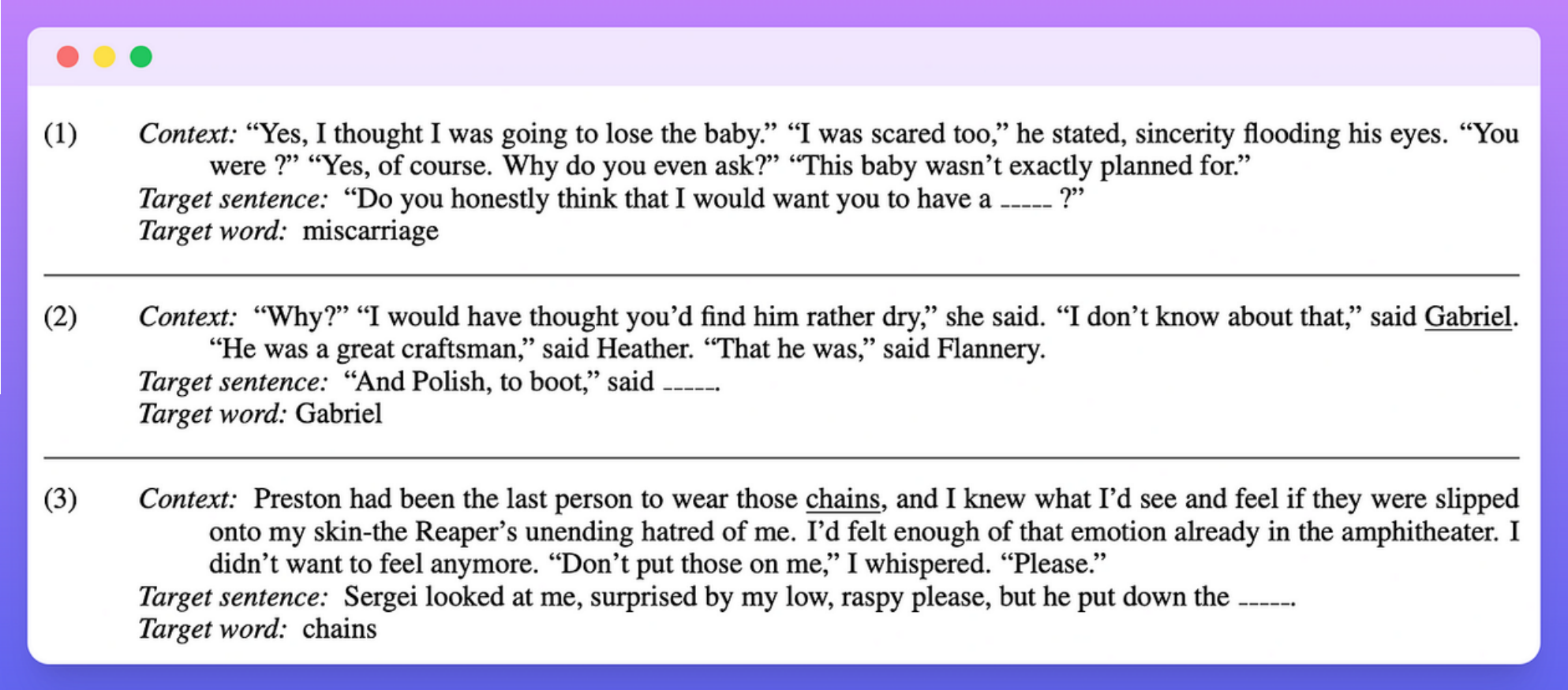
Architecture

- Similar with GPTv1 (based on decoder Transformer)
- Some small rearranging of layer norm and residual layer
- Increasing vocab size from 30000 to 50257
- Increasing context size from 512 to 1024

Parameters	Layers	d_{model}
117M	12	768
345M	24	1024
762M	36	1280
1542M	48	1600

Approach

Learning to perform a single task can be expressed in a probabilistic framework as estimating a conditional distribution $p(\text{output}|\text{input})$. Since a general system should be able to perform many different tasks, even for the same input, it should condition not only on the input but also on the task to be performed. That is, it should model $p(\text{output}|\text{input}, \text{task})$. This has been variously formalized in multitask and meta-learning settings. Task conditioning is often implemented at an architectural level, such as the task specific encoders and decoders in (Kaiser et al., 2017) or at an algorithmic level such as the inner and outer loop optimization framework of MAML (Finn et al., 2017). But as exemplified in McCann et al. (2018), language provides a flexible way to specify tasks, inputs, and outputs all as a sequence of symbols. For example, a translation training

- 
- (1) *Context:* “Yes, I thought I was going to lose the baby.” “I was scared too,” he stated, sincerity flooding his eyes. “You were ?” “Yes, of course. Why do you even ask?” “This baby wasn’t exactly planned for.”
Target sentence: “Do you honestly think that I would want you to have a _____ ?”
Target word: miscarriage
 - (2) *Context:* “Why?” “I would have thought you’d find him rather dry,” she said. “I don’t know about that,” said Gabriel. “He was a great craftsman,” said Heather. “That he was,” said Flannery.
Target sentence: “And Polish, to boot,” said _____
Target word: Gabriel
 - (3) *Context:* Preston had been the last person to wear those chains, and I knew what I’d see and feel if they were slipped onto my skin-the Reaper’s unending hatred of me. I’d felt enough of that emotion already in the amphitheater. I didn’t want to feel anymore. “Don’t put those on me,” I whispered. “Please.”
Target sentence: Sergei looked at me, surprised by my low, raspy please, but he put down the _____
Target word: chains

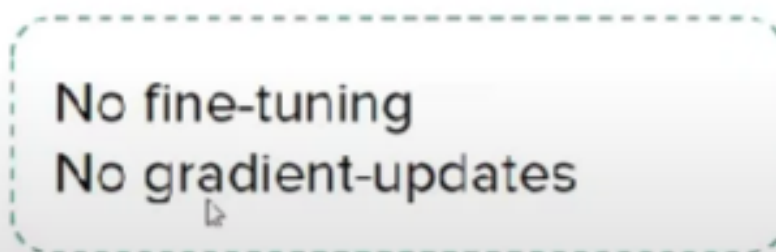
GPT 3

- Bigger than GPT 2
- Focus on in context learning

Traditional



In-context learning



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.

```
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
```

Reference

- <https://openai.com/research/better-language-models>
- <https://openai.com/research/better-language-models>
- <https://www.youtube.com/watch?v=UIFIVKBCG9U&t=1174s>
- https://www.youtube.com/watch?v=u1_qMdb0kYU&t=309s



Thanks for watching!