

# GPT: Generalized Pre-Training

GPT is a sequence of increasingly powerful (and big) models of similar architecture.

We introduced the family and the original model [here](#)([NLP Recent.ipynb#GPT:-Generalized-Pre-Training](#)).

The second and third generation models are 10 and 1000 times bigger (number of parameters).

# GPT-2

[paper \(https://cdn.openai.com/better-language-models/language\\_models\\_are\\_unsupervised\\_multitask\\_learners.pdf\)](https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf)

[Model card \(https://github.com/openai/gpt-2/blob/master/model\\_card.md\)](https://github.com/openai/gpt-2/blob/master/model_card.md)

[Summary \(https://openai.com/blog/better-language-models/\)](https://openai.com/blog/better-language-models/)

- 48 Transformer blocks (4 times original)
  - $n_{\text{heads}} = 16(?)$ ,  $d_{\text{head}} = 96(?)$ 
    - $d_{\text{model}} = n_{\text{heads}} * d_{\text{head}} = 1536$
- 1.5 billion weights
- Trained on
  - Trained on 40GB of data, 10 times the amount of data as original GPT
  - Sequence of 1024 tokens (2 times original)

## Results: Zero shot

- Tested on 8 tasks
  - State of the art on 7 out of the 8

# GPT-3

[paper \(https://arxiv.org/abs/2005.14165\)](https://arxiv.org/abs/2005.14165)

[Model card \(https://github.com/openai/gpt-3/blob/master/model-card.md\)](https://github.com/openai/gpt-3/blob/master/model-card.md)

[Summary\(\)](#)

- 96 Transformer blocks(8 times original)
  - $n_{\text{heads}} = 96, d_{\text{head}} = 128(?)$
  - $d_{\text{model}} = n_{\text{heads}} * d_{\text{head}} = 12,288$
- 1.5 billion weights
- Trained on
  - 570 GB of data (100 times GPT)
  - [Common Crawl \(https://commoncrawl.org/the-data/get-started/\)](https://commoncrawl.org/the-data/get-started/)
    - web crawler over multiple years
    - 570 GB (100 times GPT)
    - 410 billion tokens
  - Additional training sets, for experiments
    - [Webtext2 \(https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf\)](https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf)
      - Web pages originating from highly ranked Reddit links

- 19 billion tokens
  - Books
    - 67 billion tokens -Wikipedia
    - 3 billion tokens
- Sequence of 2048 tokens
- 190K KWh of electricity used in training
  - \$ 0.22 per KW hour  $\approx$  \$42*K* electricity used to train

You can see from the following graph how the computation times increase by orders of magnitude over the generations of GPT

- GPT-3 small  $\approx$  GPT
- GPT-3 XL  $\approx$  GPT-2

**Compute time**

---

---

Picture from: <https://arxiv.org/pdf/2005.14165.pdf>

## How much would it cost you to train GPT-3 ?

- Amazon Cloud
  - G5 instance
    - NVidia A10G Tensor Core GPUs @ 250 Tflops/GPU
    - 8 GPU instance (2 Pflops) @\$10/hour (with yearly contract; \ \$16\hour on-demand)
      - \$240 per 2Pflops-day
- GPT-3  $\approx$  3000 Pflop-days
  - $3000/2 = 1500$  days G5 instances to get 3000 Pflops-days
  - Cost =  $1500 * \$240/\text{day} = \$360\text{K}$



# WebText: a new training set

One key to the success of GPT-2 (and later generations) was a newly created training set that was scraped from the Web.

The most common web-scraped dataset is Common Crawl (<https://commoncrawl.org/>).

- large, diversified
- quality problems ?
  - Large set of pages pointed to are "gibberish"

The GPT team tried to create a high-quality crawl by using a curated approach to links

- Based on Reddit
- Only follow links originating from highly-ranked (high "karma") Reddit pages

The result is called WebText

- 40GB; 8MM documents
- removed any Wikipedia
  - since it is included in many of the benchmark tasks whose performance we want to measure out of sample

# Multi-task learning

One area of recent interesting is *multi-task learning*

- Training a model to implement multiple tasks

A model that implements a single task computes

$$p(\text{output} \mid \text{input})$$

A model that implements several tasks computes

$$p(\text{output} \mid \text{input}, \text{task-id} )$$

When training a model for multiple tasks, the training examples would look something like:

(Translate to French, English text, French Text)  
(Answer the question, document, question, answer)

Text is almost a universal encoding so NLP is a natural way of expressing multiple tasks.

So a natural extensions of a Language Model is to solve multiple tasks

- Encode your specific task as an input that can be handled by a Language Model
- That's one advantage of Byte Pair Encoding
  - No special per-task pre-processing needed for a task's training set

We will take the idea of Multi-task learning one step further

- Learning how to solve a task **without** explicitly training a model !

# Learning to learn

The GPT family explores some deep questions.

We are familiar with teaching a NN a task via several approaches

- Completely Supervised Training
- Supervised Pre-Training with Fine-Tuning

But can a NN learn to solve a task *without having seen training examples for the task* ?

This question can be framed as follows

- Given a trained Language Model LM
- Can LM model be *used* for a new target task T, with examples  $(\mathbf{x}_T^{(i)}, \mathbf{y}_T^{(i)})$ 
  - By giving LM a set  $E$  consisting of  $k$  examples for task T

Notice the word *used* rather than *trained*

- the weights of LM are **not** changed
- the examples in  $E$  are only used to "prime" LM to understand the new task T



There are variations of the question dependent on the size  $k$  of examples

- **Few shot learning:**  $10 \leq k \leq 100$  typically
- **One shot learning:**  $k = 1$
- **Zero shot learning**  $k = 0$

A picture will help

---

Picture from: <https://arxiv.org/pdf/2005.14165.pdf>

Is this even possible ?!

Let's look at the reported results from the third generation GPT-3 model.

## Few/One/Zero shot learning

---

Picture from: <https://arxiv.org/pdf/2005.14165.pdf>

---

# How is that possible ? Some theories

## Theory 1

- The training set contains explicit instances of these out of sample tasks

## Theory 2

- The super-large training sets contain *implicit* instances of these out of sample tasks
  - For example: an English-language article quoting a French speaker in French with English translation

One thing that jumps out from the graph:

- Bigger models are more likely to exhibit meta-learning

### **Theory 3**

The training sets are so big that the model "learns" to create groups of examples with a common theme

- Even with the large number of parameters, the model capacity does not suffice for example memorization

Another thing to consider

- The behavior of an RNN depends on *all* previous inputs
  - It has memory (latent state, etc.)

So Few Shot Learning may work by "priming" the memory with parameters for a specific task

# Social concerns

The team behind GPT is very concerned about potential misuse of Language Models.

To illustrate, they conducted an experiment in having a Language Model construct news articles

- Select title/subtitle of a genuine news article
- Have the Language Model complete the article from the title/subtitle
- Show humans the genuine and generated articles and ask them to judge whether the article was written by a human



## Human accuracy in detecting model generated news articles

---

---

Picture from: <https://arxiv.org/pdf/2005.14165.pdf>

The bars show the range of accuracy across the 80 human judges.

- 86% accuracy detecting articles created by a really bad model (the control)
- 50% accuracy detecting articles created by the biggest models

It seems that humans might have difficulty distinguishing between genuine and generated articles.

The fear is that Language Models can be used

- to mislead
- to create offensive speech

```
In [1]: print("Done")
```

Done