OntarioTech
Engineering
& Applied Science

GPT–3

Adoption

Time to reach 100 million users;

- Mobile phone - 16 years
- Internet - 7 years
- Facebook - 4.5 years
- WhatsApp - 3.5 years
- Instagram - 2.5 years
- TikTok - 9 months
- ChatGPT - 2 months

Overview

1. Technical details on GPT-3
    a. Terminology
    b. What is a language model
    c. What is GPT-3
    d. Architecture
    e. Technical details
    f. Capabilities of GPT-3
    g. Limitation of GPT-3
    h. Applications
    i. Future of AI
2. Introduction to OpenAI API
3. Getting started with the API
    a. How to obtain the API
    b. Running few example codes

Terminology

- **Language Model:**
  - These models can predict the most likely next words, along with their probabilities, given a set of words.
- **Zero/One/Few shot learning:**
  - Refer to a model's ability to learn a new task with zero, one, or a few examples.
- **Transformer Models:**
  - Transformers are a family of deep learning models that are primarily used in natural language processing (NLP). They serve as the fundamental building block for many of the current state-of-the-art NLP architectures.
- **Token:**
  - Basic units of input and output text. Discrete units of text that model process. Can be a word of subwords.
- **Parameters:**
  - Refer to the numerical values that represent the weights and biases of the neural network architecture used by the model.

What is a language model?

- Language models are trained on large datasets of natural language texts
- These models learn the underlying patterns and structures of language to predict the likelihood of sequence of words
- These models are used for natural language processing, machine translation, text classification, speech recognition, text-to-speech conversion and question and answering tasks.
- Most language models are trained, using statistical methods such as n-grams and markov models, and deep learning techniques such as neural networks.
- The significance is that most modern language models have typically been designed, using deep learning techniques. The most commonly used method is the transformer architecture, used in GPT-3 models.

State of Language Models

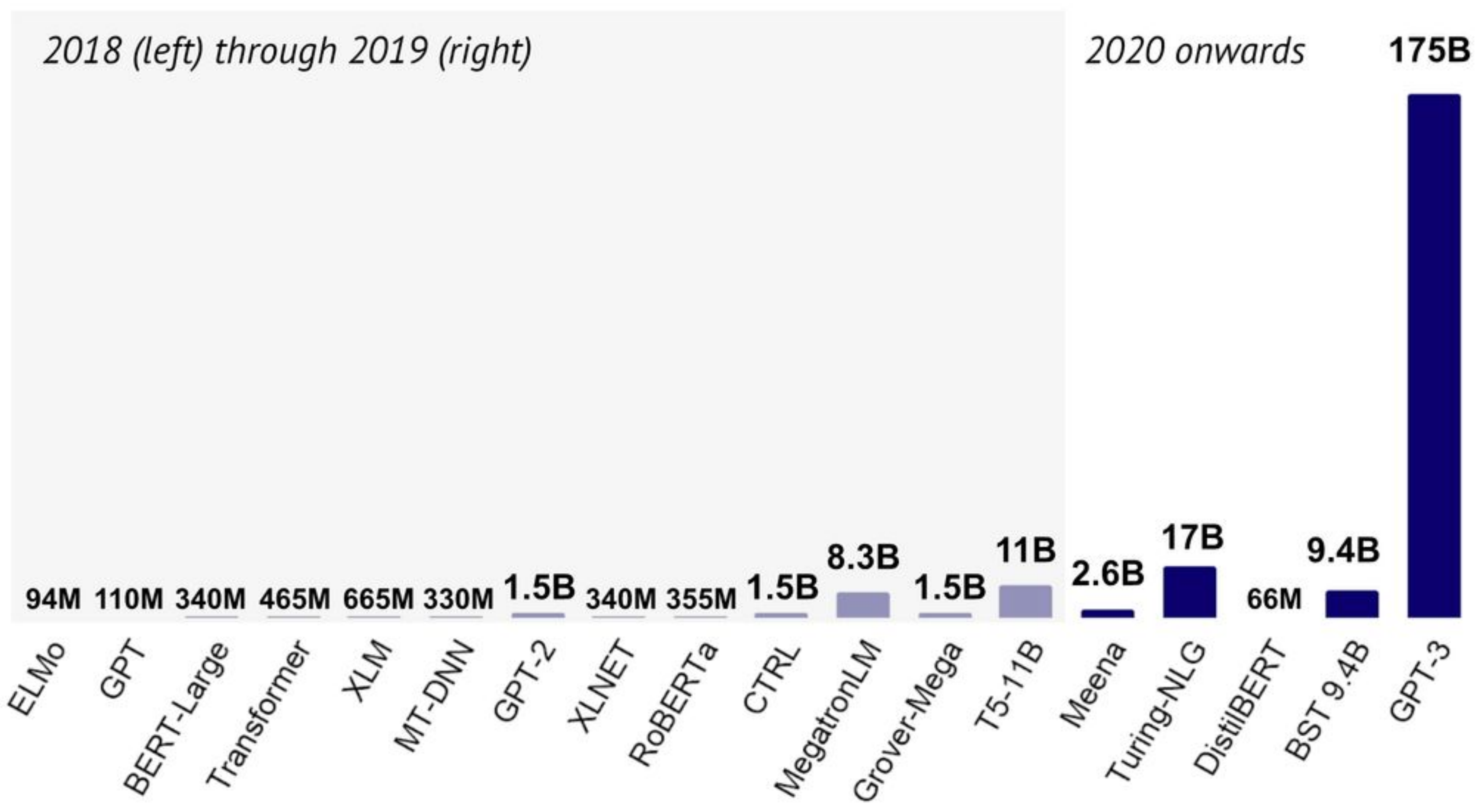


Figure 1: Evolution of Language Models and their sizes [1]

What is GPT-3

- GPT stands for “Generative Pre-trained Transformer”

GPT:

- GPT: first version of GPT, released in 2018. It had 117 million parameters and was trained on web-text

GPT-2:

- Advanced version of GPT, released in 2019, had 1.5 billion parameters.
- 8 million web pages

GPT-3:

- Is one of the best state-of-the-art language processing AI models developed
- GPT-3 is an autoregressive language model with 175 billion parameters, 10x time more than any previous method
- 45 terabyte of training data

Technical Details::Architecture

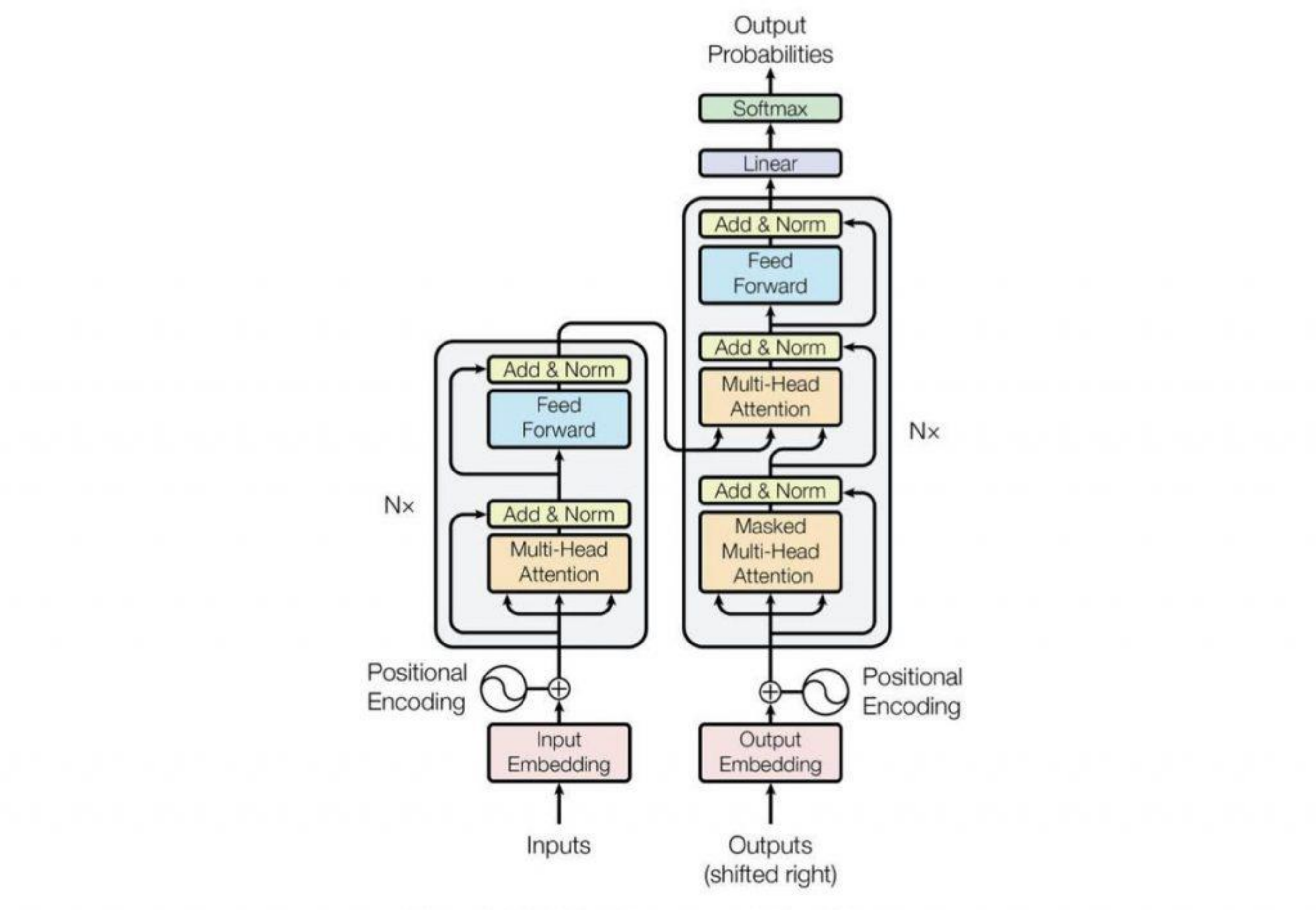
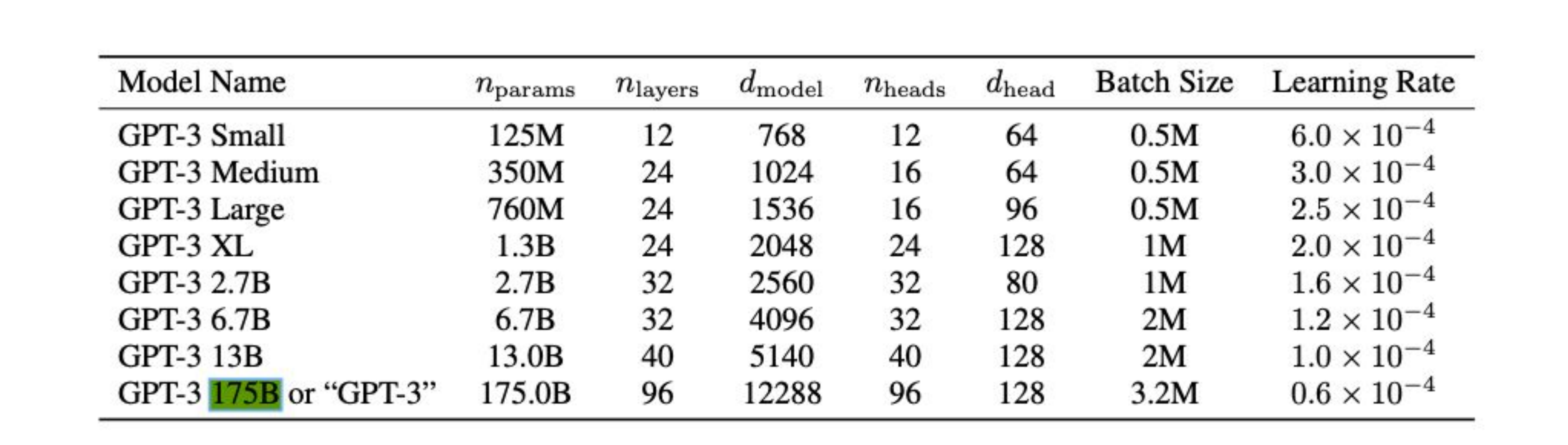


Figure 2: The transformer model architecture [2]

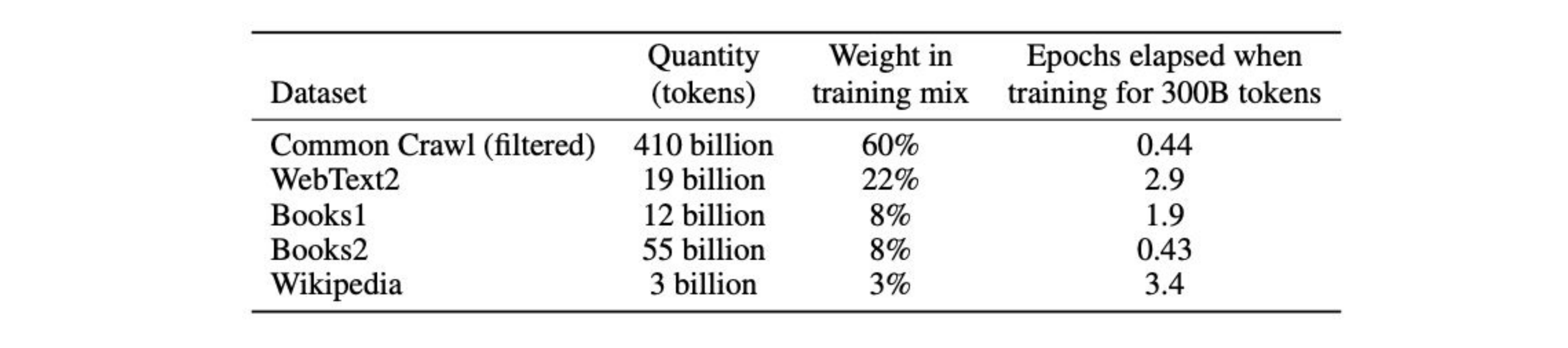
Technical Details::Models and parameters



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}

Figure 3: Model, size, architecture, and learning hyper-parameters (batch size in tokens)

Technical Details::Dataset used for GPT-3



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

Figure 4: Dataset used for training the model. Trained 300 billion tokens.

CommonCrawl: 45TB, between 2016 - 2019, after filtering 570GB

WebText2: 40GB, of outbound links from Reddit, that are interesting and educational

Wikipedia: 6 million articles.

Technical Details::Computation power used

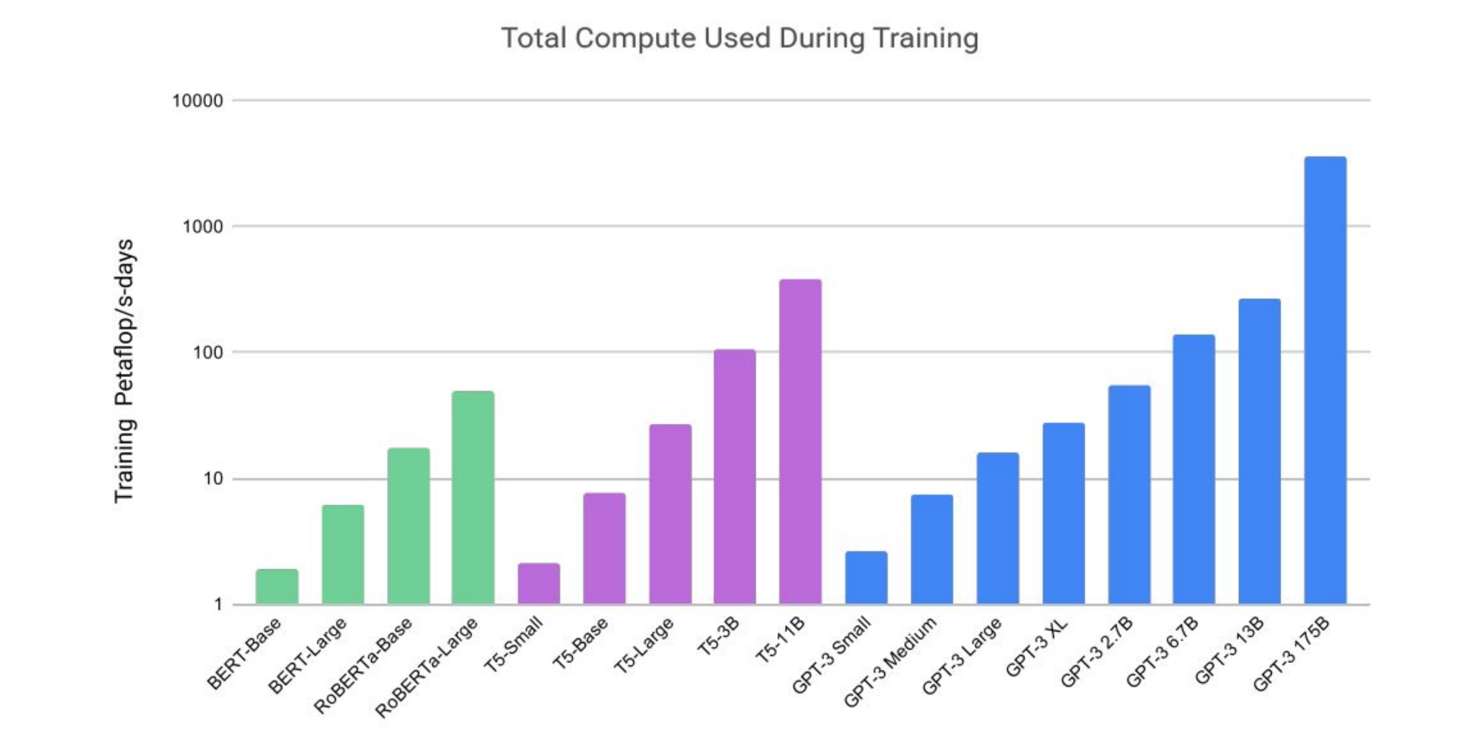


Figure 5: Total computation used training compare to other state-of-the-art models (presented in log scale)

"As an example, it requires approximately 9.2 days on 512 V100 GPUs to train a 8.3B GPT-2 (Shoeybi et al., 2019), and 14.8 days on 10000 V100 GPUs to train a 175B GPT-3 (Patterson et al., 2021)"

Technical Details::Accuracy and Performance

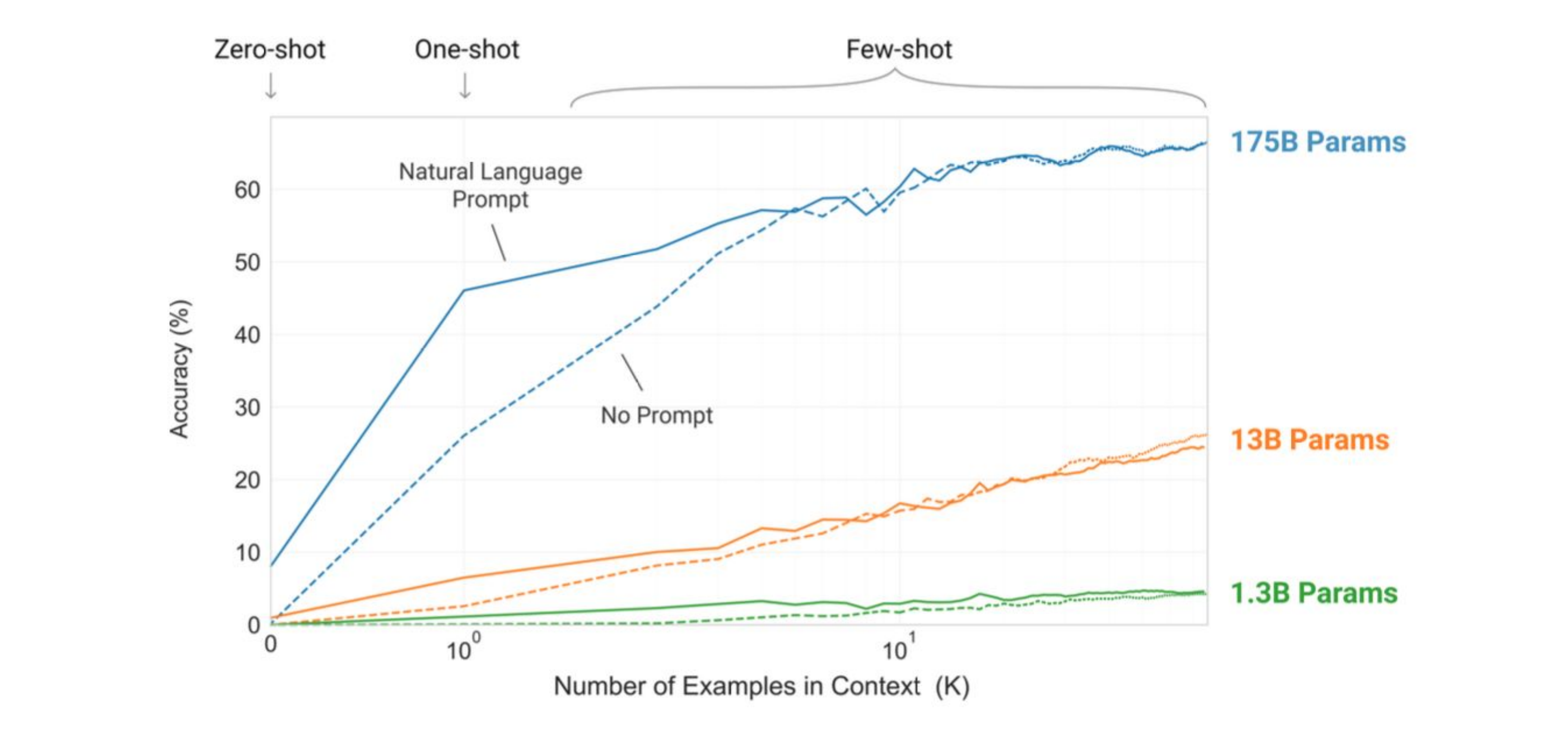
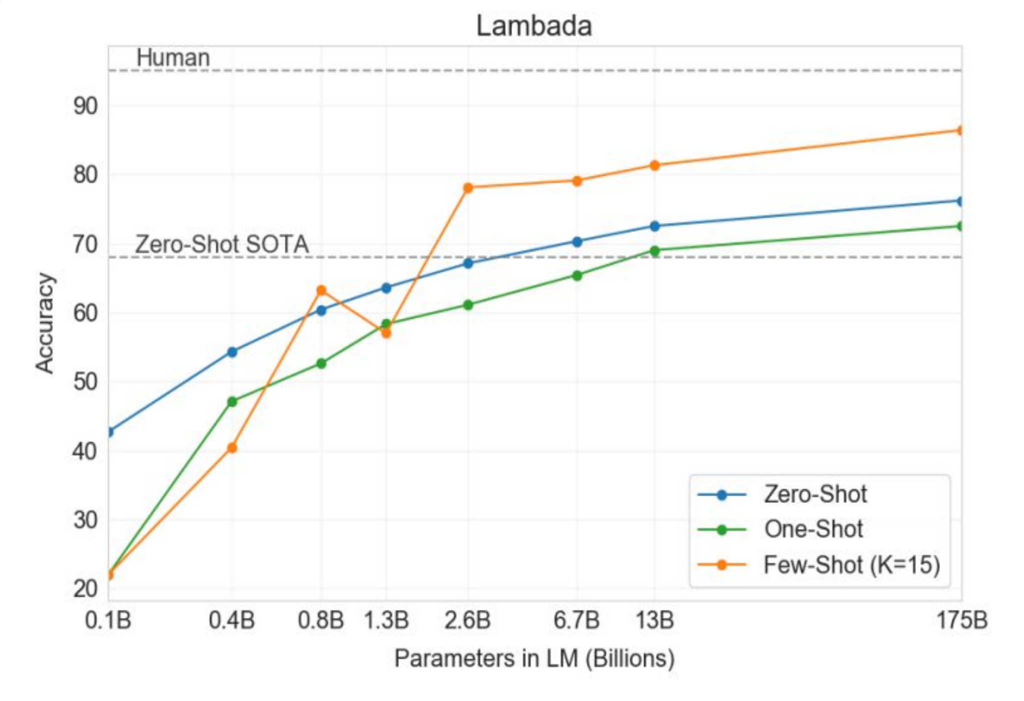
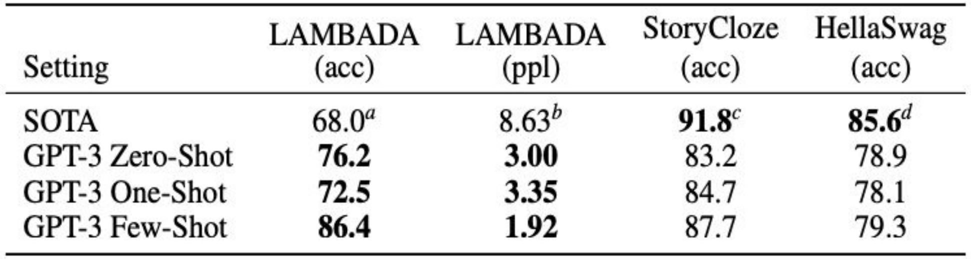


Figure 6: Performance over different in-context learning methods. Graph shows gains due to scaled up size of the model for zero-shot, one-shot and few-shot learning.

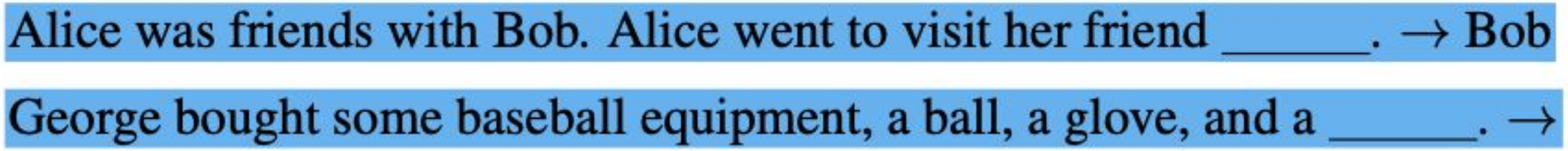
Technical Details::Evaluation and results

1. LAMBADA dataset (archived 86% accuracy)





Setting	LAMBADA (acc)	LAMBADA (ppl)	StoryCloze (acc)	HellaSwag (acc)
SOTA	68.0[a]	8.63[b]	**91.8**[c]	**85.6**[d]
GPT-3 Zero-Shot	**76.2**	**3.00**	83.2	78.9
GPT-3 One-Shot	**72.5**	**3.35**	84.7	78.1
GPT-3 Few-Shot	**86.4**	**1.92**	87.7	79.3



Alice was friends with Bob. Alice went to visit her friend ______. → Bob

George bought some baseball equipment, a ball, a glove, and a ______. →

Figure 7: Performance on the LAMBADA dataset

Technical Details::Evaluation and results

2. HellaSwag (79.3%)

3. StoryCloze (87.2%)

4. Closed Book Question Answering:

- This measures the ability to answer questions on broad factual knowledge.
    1. Natural Questions
    2. WebQuestions dataset
    3. TriviaQA datasets

5. Translations:

- From French to English, German to English, Romanian to English and vice versa.

6. Winograd Scheme-like tasks (89.7%)

7. Common Sense Reasoning: (82.8% Accuracy)

8. Comprehensive Reading tasks

9. SuperGLUE benchmark suite

10. Natural Language Inference (NLI) tasks

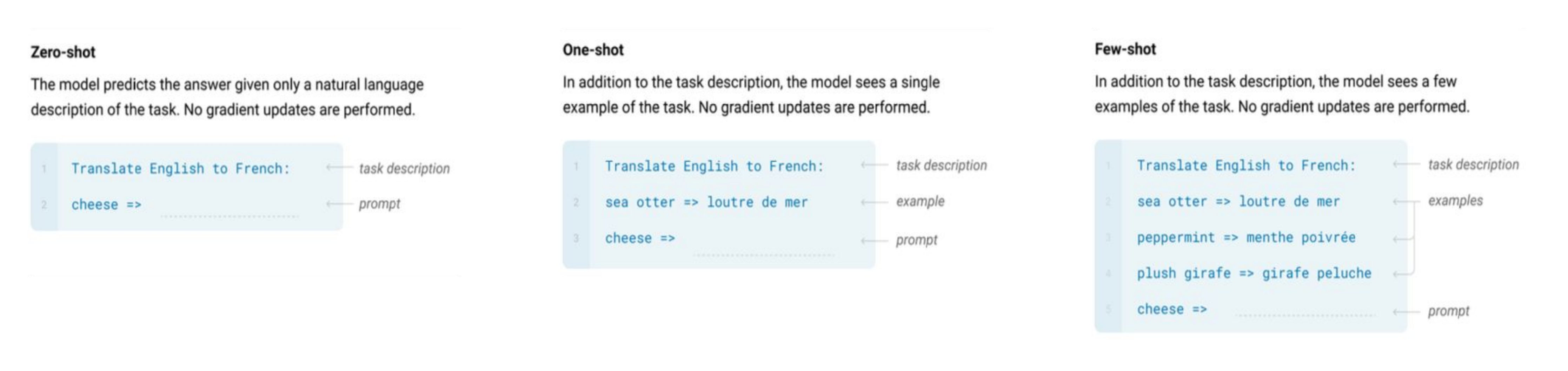
11. Synthetic and Qualitative Tasks

12. Arithmetic

State-of-the-art results archived by GPT-3 are;

1. **Cloze task and sentence and paragraph completion**
2. **Commonsense reasoning (PIQA dataset)**

Technical Details::Zero-shot, One-shot and Few-shot



Zero-shot

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

```
Translate English to French:        task description
cheese =>                           prompt
```

One-shot

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

```
Translate English to French:        task description
sea otter => loutre de mer          example
cheese =>                           prompt
```

Few-shot

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

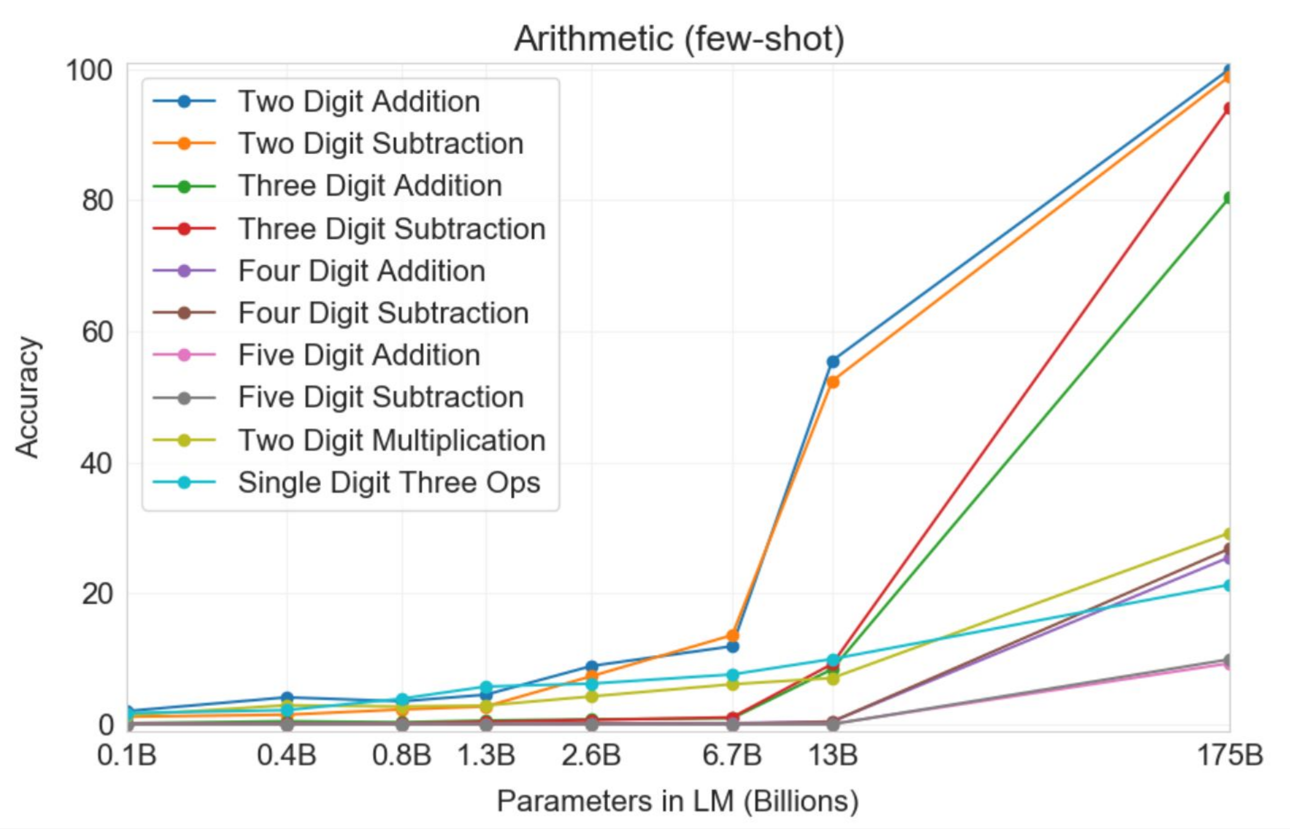
```
Translate English to French:        task description
sea otter => loutre de mer          examples
peppermint => menthe poivrée
plush girafe => girafe peluche
cheese =>                           prompt
```

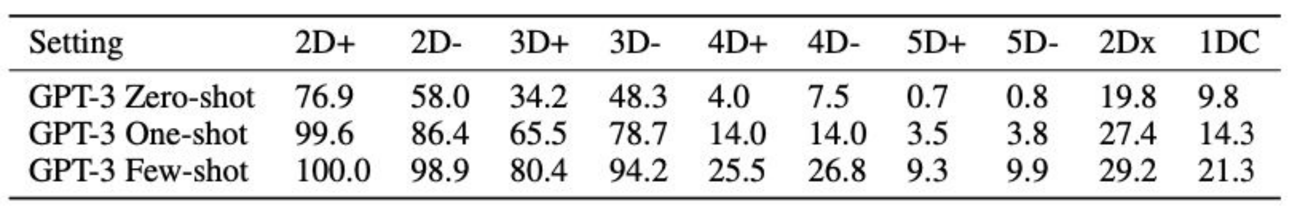
Figure 7: Zero-shot and one-short learning - on the language model

Advantages/Capabilities of GPT-3

- **Language Generation**:
  - able to generate high-quality text that is often difficult to distinguish from human-generated text.
- **Language Understanding:**
  - can perform a variety of language understanding tasks, such as sentiment analysis, question answering, and summarization. It can also understand and generate text in multiple languages.
- **Zero-Shot and Few-Shot Learning:**
  - able to learn new tasks with zero or very few examples, making it a highly flexible and adaptable model(generalizable).
- **Large-Scale Training:**
  - since it was trained on a massive dataset, the extensive training enables it to generate high-quality text with a rich vocabulary.
- **Fewer Preprocessing Requirements:**
  - Unlike other NLP models, GPT-3 requires minimal preprocessing of data, making it easier and faster to work with.
- **Multi-Task Learning:**
  - can perform multiple tasks, such as language translation, question answering, and summarization, without requiring separate training for each task.

Limitations of GPT-3





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

Figure 8: Results on basic arithmetic tasks for GPT-3

Ex: Q: What is 24 times 42?

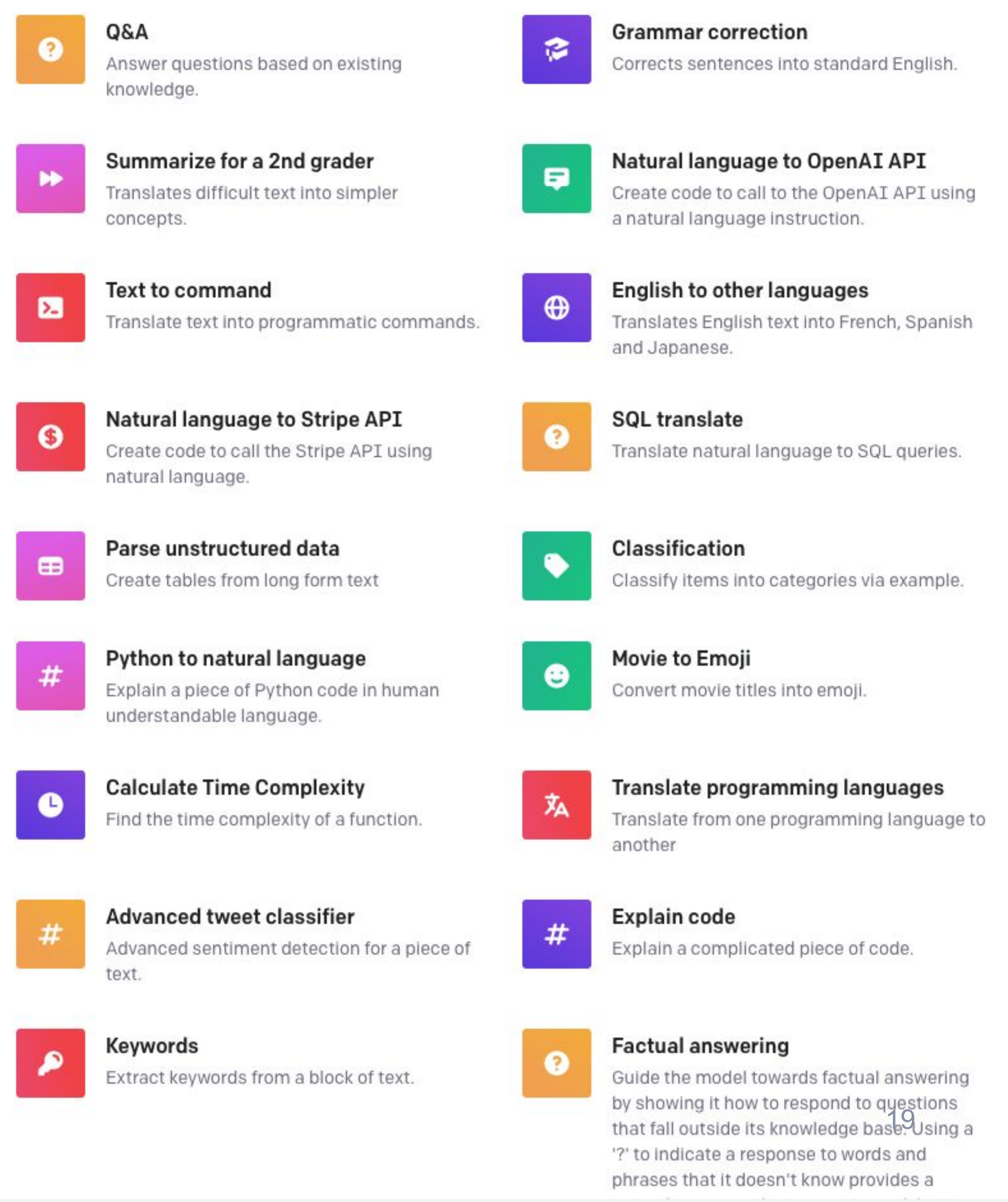
A: 1008

Limitations of GPT-3

- **Lack of real-world context:**
  - GPT-3 relies on statistical patterns in language to generate text, which means it lacks true understanding of the world and common sense reasoning.
- **Limited factual accuracy:**
  - GPT-3 can generate text that is factually incorrect or biased, as it learns from the patterns in the data it is trained on, which can be biased or contain errors.
- **Limited interpretability:**
  - GPT-3 is a complex model with millions of parameters, making it difficult to understand how it works or why it makes certain predictions.
- **Limited transferability:**
  - GPT-3 performs best on tasks that are similar to the ones it was trained on, and may not perform well on tasks that require knowledge from other domains.
- **Resource-intensive:**
  - GPT-3 requires significant computational resources to train and run, making it inaccessible to many researchers and developers.

Applications

1. Text generation
2. Translation
3. Question answering
4. Chatbots and virtual assistant
5. Sentiment analysis
6. Summarization
7. Creative writing
8. Code generation
9. Personalization
10. Education
11. Voice assistants
12. Search Engines (Bing)



Q&A
Answer questions based on existing knowledge.

Grammar correction
Corrects sentences into standard English.

Summarize for a 2nd grader
Translates difficult text into simpler concepts.

Natural language to OpenAI API
Create code to call to the OpenAI API using a natural language instruction.

Text to command
Translate text into programmatic commands.

English to other languages
Translates English text into French, Spanish and Japanese.

Natural language to Stripe API
Create code to call the Stripe API using natural language.

SQL translate
Translate natural language to SQL queries.

Parse unstructured data
Create tables from long form text

Classification
Classify items into categories via example.

Python to natural language
Explain a piece of Python code in human understandable language.

Movie to Emoji
Convert movie titles into emoji.

Calculate Time Complexity
Find the time complexity of a function.

Translate programming languages
Translate from one programming language to another

Advanced tweet classifier
Advanced sentiment detection for a piece of text.

Explain code
Explain a complicated piece of code.

Keywords
Extract keywords from a block of text.

Factual answering
Guide the model towards factual answering by showing it how to respond to questions that fall outside its knowledge base. Using a '?' to indicate a response to words and phrases that it doesn't know provides a

Future

1. Improved natural language understanding
2. Applications in new fields
3. Increased personalization
4. Collaboration between humans and AI
5. Ethical considerations

Image: Shutterstock

OpenAI API

The API provides access to various pre-trained AI models, including language processing, natural language understanding, and machine learning models. This enables developers to build AI-powered applications without having to invest significant time and resources in developing their own AI models.

> https://platform.openai.com/docs/quickstart

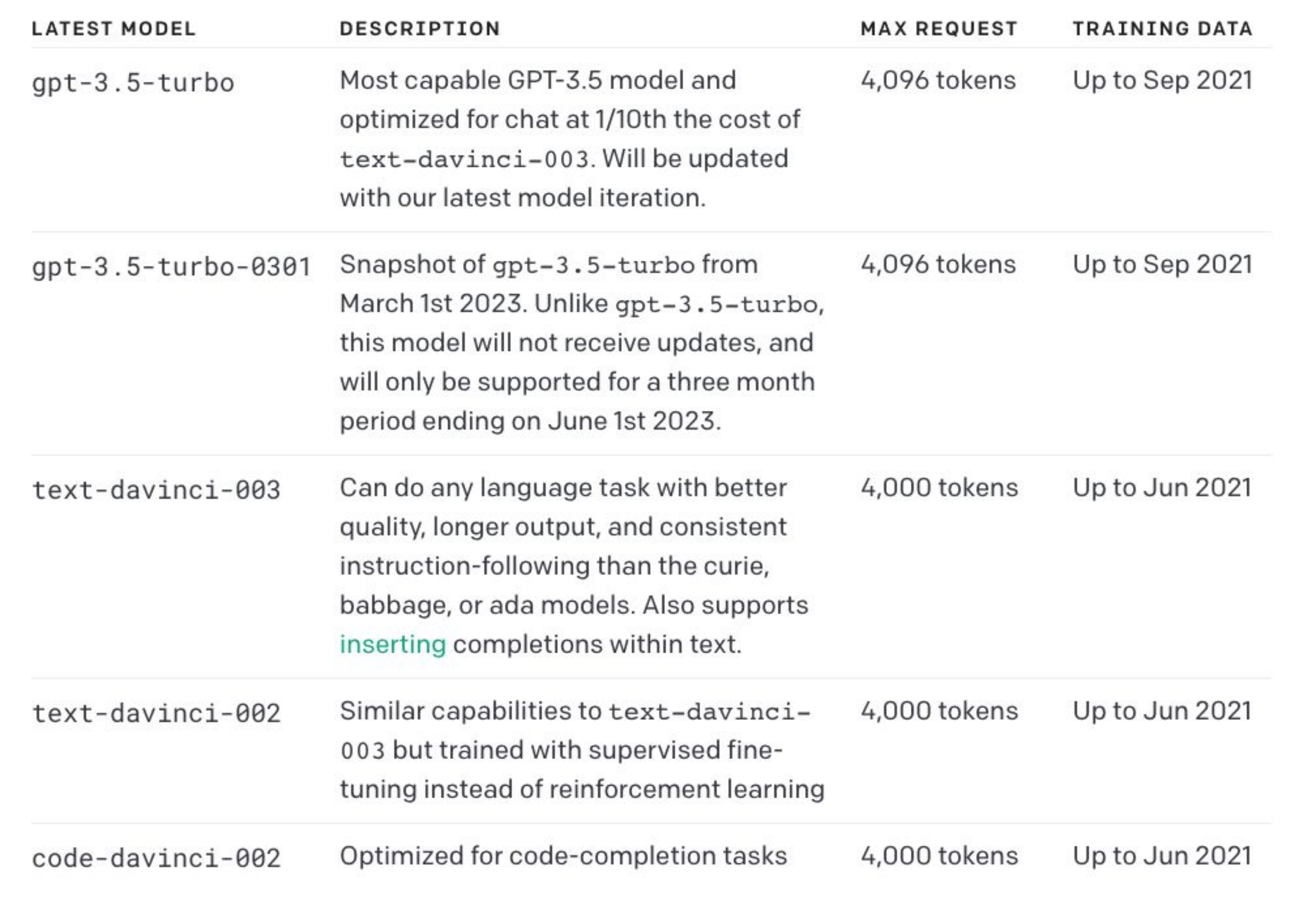
> https://platform.openai.com/docs/api-reference/introduction

OpenAI API is offered in different languages;

- Official Libraries are; python, Node.JS

There are Community written libraries for almost any of other languages

OpenAI API: Available Models



LATEST MODEL	DESCRIPTION	MAX REQUEST	TRAINING DATA
`gpt-3.5-turbo`	Most capable GPT-3.5 model and optimized for chat at 1/10th the cost of `text-davinci-003`. Will be updated with our latest model iteration.	4,096 tokens	Up to Sep 2021
`gpt-3.5-turbo-0301`	Snapshot of `gpt-3.5-turbo` from March 1st 2023. Unlike `gpt-3.5-turbo`, this model will not receive updates, and will only be supported for a three month period ending on June 1st 2023.	4,096 tokens	Up to Sep 2021
`text-davinci-003`	Can do any language task with better quality, longer output, and consistent instruction-following than the curie, babbage, or ada models. Also supports inserting completions within text.	4,000 tokens	Up to Jun 2021
`text-davinci-002`	Similar capabilities to `text-davinci-003` but trained with supervised fine-tuning instead of reinforcement learning	4,000 tokens	Up to Jun 2021
`code-davinci-002`	Optimized for code-completion tasks	4,000 tokens	Up to Jun 2021

Image: Shutterstock

OpenAI API : Basic usage

> pip install openai

```
import os
import openai

# Load your API key from an environment variable or secret management service
openai.api_key = os.getenv("OPENAI_API_KEY")

response = openai.Completion.create(model="text-davinci-003", prompt="Say this is a test", temperature=0, max_tokens=7)
```

OpenAI API: Getting started with API

API Reference: https://platform.openai.com/docs/api-reference/introduction

```
response = openai.Completion.create(
  model="text-davinci-003",
  prompt="prompt\n",
  temperature=0,
  max_tokens=100,
  top_p=1,
  frequency_penalty=0.0,
  presence_penalty=0.0,
  stop=["\n"]
)
```

model: ID of the GPT model to use “davinci, ada, curie”

prompt: text prompt to feed (upto 2048 tokens)

temperature [0, 2]: controls the “creativity”. Higher value gives diverse and unpredictable results, lower will give predictable result

max_tokens: maximum length of response (words)

n: number of responses to generate (choices)

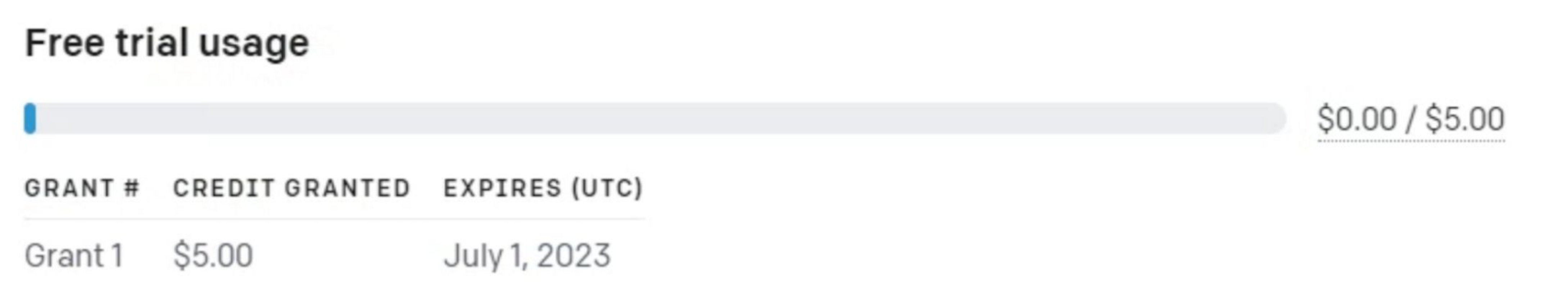
stop: specifies a string to serve as stop criteria for response

OpenAI API: Register to get the API (provide a $5 free trial)

1. https://platform.openai.com/

OpenAI API: Register to get the API (provide a $5 free trial)

2. https://platform.openai.com/account/usage



Free trial usage

$0.00 / $5.00

GRANT #	CREDIT GRANTED	EXPIRES (UTC)
Grant 1	$5.00	July 1, 2023

3. https://platform.openai.com/account/api-keys

OpenAI API: Development Environment

4. https://jupyterenv.metaai.dev/hub/login

Datasets for your own projects

OpenWebTextCorpus: Attempt to recreate the OpenAI’s WebText2 dataset

https://skylion007.github.io/OpenWebTextCorpus/

CommonCrawl: Open repository of Web crawl data,

https://commoncrawl.org/the-data/get-started/

References

[1] N. B. and I. Hogarth, “State of AI Report 2022.” https://www.stateof.ai/

[2] “OpenAI API.” https://platform.openai.com

Thank You