Review of GPT-3
CS 410 Fall 2020
Pierson Wodarz (wodarz2)

Introduction

In May 2020 researchers at OpenAI published a paper detailing an NLP system, GPT-3 [1]. This system was an evolution of a prior system, GPT-2, which was itself a successor to a previous system, GPT. This paper caused a lot of excitement for several reasons. First, GPT-2 was considered a major step forward for NLP and it was initially considered too dangerous by the authors to release in its entirety [2]; whereas GPT-3 was expected to be fully released. Second, GPT-3 made several improvements over its predecessor which enabled new applications. It is the improvements over GPT-2 and the new applications for GPT-3 which will be discussed in this review.

Overview of GPT

GPT is a transformer-based language model [3]. Specifically, a transformer decoder, which uses preceding tokens to generate an output based on a probability maximization. These tokens could be in the form of words (e.g. the preceding and ending words in a sentence), question/answers (e.g. the preceding question and subsequent answer), similarity (e.g. the possible orderings of a sentence), or other forms. Put simply: the model is used to predict the next word in a corpus given the previous words [2].

The model is generated by performing unsupervised training on various tokenized datasets. GPT-1 was trained on several different datasets to perform different tasks, for example RACE and Story Cloze datasets for Question Answering tasks, and the Stanford Sentiment Treebank-2 and CoLA for the Classification tasks [3]. GPT-2 used the WebText dataset generated by scraping the web [2]. GPT-3 used the Common Crawl dataset, a web-scraped dataset, a dataset of unpublished books, and Wikipedia [1]. Each subsequent model of GPT used larger datasets to train.

Ultimately the model is utilized to generate text based on user input. Possible forms include answering a question, finishing a sentence, and telling a story based on a prompt. These tasks can be performed in a variety of settings including Few-shot, One-Shot, and Zero-Shot settings, which refer to the number of demonstrations the model is provided before being asked to complete a task.

Comparison between GPT-2 and GPT-3

In short, the difference between GPT-2 and GPT-3 comes down to size and performance. Namely, the same model and architecture was used in GPT-3 as was used in GPT-2, so the difference lies in the datasets used to train the models and the size of the resultant models.

GPT-2 was trained on 40GB of text data scraped from the internet resulting in 1.5 billion parameters in the model [2]. GPT-3 was trained on 10's of terabytes of data (45TB for the Common Crawl dataset alone) resulting in over 300 billion training tokens and 175 billion parameters in the model [1]. Overall, in terms of training size, and model size, GPT-3 is about 2 orders of magnitude larger than GPT-2. In particular, it is the difference in the number of parameters in the model (175 billion for GPT-3 vs 1.5 billion for GPT-2) that allows for better performance and additional applications of GPT-3 that were not possible in GPT-2.

The performance improvements can be quantified by comparing the results of the models on various tasks. For example, on the LAMBADA language modeling task, zero-shot GPT-2 achieves a perplexity score (lower is better) of 8.63 [2] whereas zero-shot GPT-3 achieves a perplexity score of 3.0 [1]. Additionally, zero-shot GPT-2 achieves an accuracy score of 63.24% [2] whereas zero-shot

GPT-3 achieves an accuracy score of 76.2% [1]. This demonstrates a huge increase in performance for GPT-3. A summary of various tests/tasks and performance comparison can be seen below:

Dataset	Metric*	GPT-2	GPT-3
Winograd Schema	Accuracy (+)	70.7%	90.1%
LAMBADA	Accuracy (+)	63.24%	76.2%
LAMBADA	Perplexity (-)	8.63	3.00
Penn Tree Bank	Perplexity (-)	35.76	20.5

*(+): Higher is better; (-): Lower is better

The above quantifies the difference in performance between the two models, resulting in a clear distinction and improvement.

Capabilities/Applications of GPT-3

In their paper on GPT-3, the authors provide several example applications and quantitative results of GPT-3 when used to perform these tasks [1]. Several example applications are described below:

- Closed Book Question Answering: The model can be used to answer factual questions.
- Translation: The model can be used to translate languages (e.g. from English to German or French to English).
- Common Sense Reasoning: The model can reason scientifically and answer questions about how things work.
- News Article Generation: The model can generate news articles based on a prompt.

However, beyond the applications described by the authors, several independent researchers and individuals have identified novel uses, applications, and implementations of GPT-3:

- Code Interpretation [4]: The model interprets and describes blocks of code. This includes what the code does, the functions within the code, and the data types.
- Javascript Layout Generator [5]: The model takes as input a description of a page layout, and outputs the javascript code to generate the described layout.
- Machine Learning Code Generator [6]: The model takes as input a description of the dataset and desired output, and outputs the code to generate the machine learning model.

The above were chosen as examples because they represent novel applications of GPT-3, unthought of by the original authors. In particular, GPT-3 was mostly considered useful as a language text generator (answering questions, writing articles, categorization, etc). However, its ability to comprehend meaning and follow syntactical structures allows it to be applied beyond solely natural languages. In the case of the javascript layout generator, the model can understand the meaning behind the input description, and translate this meaning into code by mapping from an English description (e.g. “in” a “circle”), to the meaning behind that description, to code that represents the desired physical structure.

However, there exist limitations to the performance [1]. In the paper, the author’s note that the model loses coherence over the course of the document when required to generate lengthy text. Additionally, the model has inherent biases along racial, gender, and religious lines as a result of the training data, which may hinder performance in the sense that the model will be constrained by those built-in biases.

Finally, in terms of broader application, there may be potential for misuse of such a model [1], ranging from misinformation to social engineering. However, to mitigate such potential for misuse,

OpenAI is limiting access to the tool, which in turn limits the potential applications of the tool to those which the approved users have in mind.

Summation

GPT-3 represents an improvement beyond its predecessor in terms of size, complexity, and performance. This improvement unlocked new applications and opportunities for NLP. While there are still limits to the abilities of the model, its clear that advancements will continue to be made and the model offers additional performance improvements and opportunities with further training. The implications of such a powerful model, and the impact which it will have in the real-world, remain open questions that will be answered as the model is implemented over time.

References

[1] Brown, T. B., “Language Models are Few-Shot Learners”, *arXiv e-prints*, 2020.
[2] Radford, Alec, "Language Models are Unsupervised Multitask Learners", *OpenAI Blog,* 2019.
[3] Radford, Alec, “Improving Language Understanding by Generative Pre-Training”, *OpenAI Blog*, 2018.
[4] Masad, Amjad, https://twitter.com/amasad/status/1285789362647478272, *Twitter*, 2020.
[5] Shameem, Sharif, https://twitter.com/sharifshameem/status/1282676454690451457, *Twitter*, 2020.
[6] Scumer, Matt, https://twitter.com/mattshumer_/status/1287125015528341506, *Twitter*, 2020.