Tech Review: How GPT has evolved

Yutaro Nishiyama (yutaron2@illinois.edu)

2022-10-31

1 Introduction

GPT (Generative Pre-trained Transformer) was developed by OpenAI in 2018. Before GPT, major NLP models were trained on large annotated data to perform specific tasks, such as sentiment classification and textual entailment, which had two major limitations:

1. Fail to generalize beyond specific tasks
2. Require much manual labor to label a training dataset.

OpenAI has developed GPT, which addresses these issues, and released GPT-2 in 2019 and GPT-3 in 2020. In this review, we will investigate the three GPT models and clarify how GPT has evolved.

2 GPT-1

GPT-1 realizes generalization with fewer labeled training data by

1. training the model on unlabeled data as unsupervised learning
2. tuning parameters as supervised learning with fewer data.(Radford et al. (2018))

2.1 Unsupervised pre-training

GPT-1 used the BooksCorpus and 1 Billion Word Language Model Benchmark datasets as training dataset for pre-training. Given this unsupervised corpus of tokens $u = u_1, ..., u_n$, we maximize the following likelihood to obtain a language model.

$$L_1(u) = \sum_i \log P(u_i | u_{i-k}, ..., u_{i-1}; \theta)$$

where k is the size of the context window and $k = 512$ in GPT-1.

To train its parameters, GPT-1 uses a multi-layer Transformer decoder as follows.

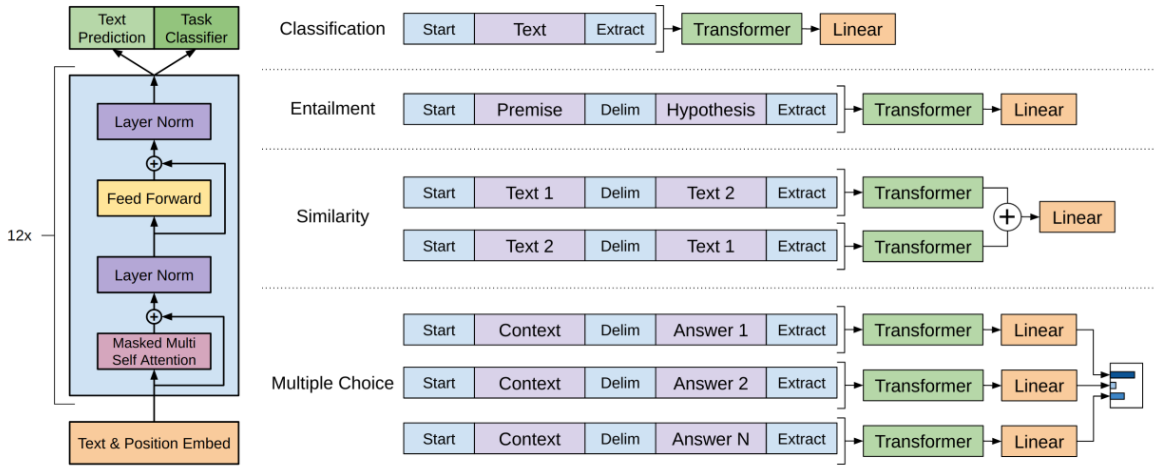

Figure 1: Radford et al. (2018)

The input h_0 is defined as

$$h_0 = UW_e + W_p$$

where $U = (u_{-k}, ..., u_{-1},)$, W_e is the token embedding matrix, and W_p is the position embedding matrix. U starts with `Start`, ends with `Extract`, and is separated by `Delim`.

Then, GPT-1 processes the input through 12 decoder layers, i.e., transformer blocks, each of which has

1. Masked Multi-Head attention
2. Normalization
3. Feed Forward Network
4. Layer Normalization.

$$h_l = \text{transformer-block}(h_{l-1}) \forall i \in [1, n]$$

Finally, it predicts the next word with a linear and softmax layer.

$$P(u) = \text{softmax}(h_n W_e^T)$$

GPT-1 train the model for 100 epochs on minibatches of 64 randomly sampled, contiguous sequences of 512 tokens.

2.2 Supervised fine-tuning

After unsupervised pre-training, GPT-1 adapt the parameters to the supervised target task by substitute the last linear & softmax layer with another layer for a specific task.

Given a labeled dataset C composed of input tokens $x^1, ...x^m$ with a label y, the last layer is defined as

$$P(y|x^1, ..., x^m) = \text{softmax}(h_l^m W_y).$$

GPT-1 tunes the parameter to maximize

$$L_2(C) = \sum \log P(y|x^1, ..., x^m) + \lambda L_1(C)$$

where $\lambda = 0.5$, since adding L_1 improves generalization and accelerates convergence.

Surprisingly, GPT-3 finetunes the parameters for only 3 epochs in most cases.

2.3 Results

GPT-1 improved the state of the art on 9 of the 12 following datasets.

Table 1: **Bold datasets** are those GPT-1 improved SoTA for.

Task	Datasets
Natural language inference	**SNLI**, **MultiNLI**, **Question NLI**, RTE, **SciTail**
Question Answering	**RACE**, **Story Cloze**
Sentence similarity	MSR Paraphrase Corpus, **Quora Question Pairs, STS Benchmark**
Classificaiton	Stanford Sentiment Treebank-2, **CoLA**

In addition, zero-shot performance of GPT-1 indicates that language model can realize the generalization. Zero-shot performance means no fine-tuning to a specific task after pre-training.

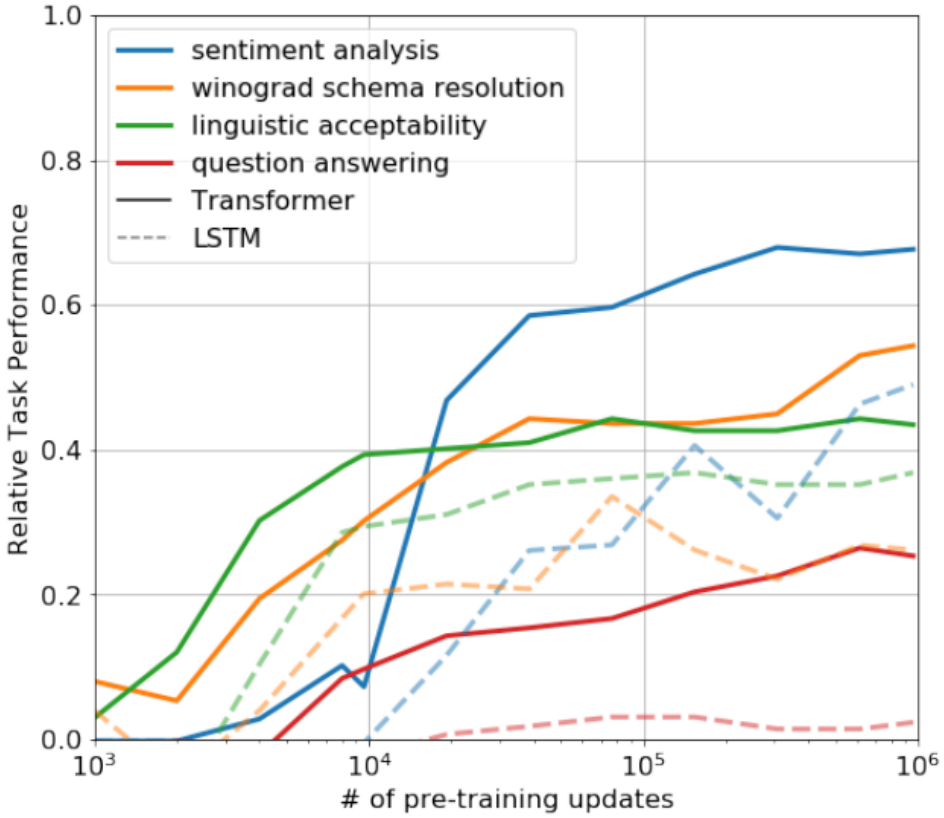

Figure 2: The evolution of zero-shot performance on different tasks as a function of LM pre-training updates. Performance per task is normalized between a random guess baseline and the current state-of-the-art with a single model. (Radford et al. (2018))

3 GPT-2

In 2019, OpenAI developed GPT to GPT-2 trained on a larger dataset with 1.5 billion parameters, which is over 10 times larger than GPT-1 with 117 million parameters (Radford et al. (2019)). Although GTP-2 has many similarities with GPT-1, it makes the following changes.

3.0.1 Dataset for pre-training

GPT-2 used the WebText dataset for pre-training, which is specifically created for this model. To assure the quality of corpus, the researchers scraped all outbound links from Reddit, a social media, with at least 3 karma.

3.0.2 Input layer

The context token size increases from 512 to 1024. The batch size also increases from 64 to 512.

3.0.3 Decoder layers

The number of decoder layers increases to 48. The order of layers also changed as follows.

1. Layer Normalization
2. Masked Multi-Head attention
3. Layer Normalization
4. Feed Forward Network

3.1 Results

GTP-2 achieved SoTA on 7 out of 8 datasets in zero-shot.

GPT-2 also achieved SoTA in Children's book test (Hill et al. (2015)), LAMBADA (LAnguage Modeling Broadened to Account for Discourse Aspects, Paperno et al. (2016)), and Winoward Schema Challenge (Levesque, Davis, and Morgenstern (2012)).

Language Models are Unsupervised Multitask Learners

	LAMBADA (PPL)	LAMBADA (ACC)	CBT-CN (ACC)	CBT-NE (ACC)	WikiText2 (PPL)	PTB (PPL)	enwik8 (BPB)	text8 (BPC)	WikiText103 (PPL)	1BW (PPL)
SOTA	99.8	59.23	85.7	82.3	39.14	46.54	0.99	1.08	18.3	**21.8**
117M	**35.13**	45.99	**87.65**	**83.4**	**29.41**	65.85	1.16	1.17	37.50	75.20
345M	**15.60**	55.48	**92.35**	**87.1**	**22.76**	47.33	1.01	**1.06**	26.37	55.72
762M	**10.87**	**60.12**	**93.45**	**88.0**	**19.93**	**40.31**	**0.97**	**1.02**	22.05	44.575
1542M	**8.63**	**63.24**	**93.30**	**89.05**	**18.34**	**35.76**	**0.93**	**0.98**	**17.48**	42.16

Figure 3: Radford et al. (2019)

4 GPT-3

In 2020, GPT-3 was released as the third version of GPT (Brown et al. (2020)). GPT-3 increases the number of parameters drastically to 175 billion parameters, since the scaling law indicates that a larger transformer performs better (Kaplan et al. (2020)). Scaling Law quantifies how cross-entropy is estimated as follows.

Given L is cross-entropy,

$$L \propto N^{-\alpha_N}$$

where N is the number of parameters and $\alpha_N \sim 0.076$.

$$L \propto D^{-\alpha_D}$$

where D is the data size and $\alpha_D \sim 0.095$.

$$L \propto C^{-\alpha_C}$$

where N is the computation cost and $\alpha_C \sim 0.057$.

In addition to the increase in parameters, GPT-3 evolves from GPT-2 in several ways.

4.1 Dataset for pre-training

GPT-3 used a five different corpora, i.e., Common Crawl, WebText2, Books1, Books2, and Wikipedia.

4.1.1 Input layer

The context token size increases from 1024 to 2048. The batch size also increases from 512 to 3.2 million.

4.2 Decoder Layer

GPT-3 adopts Sparse Transformer (Child et al. (2019)), which pay attention to limited number of preceding tokens (i.e., sparse) to lighten Multi-Head attention layers. Also, the number of decoder layers increases to 96.

4.3 No need for fine-tuning

GPT-3 requires no fine-tuning. In this sense, GPT-3 achieved full generalization, independent from any specific tasks. Instead, we provide examples of a specific task to the model before performing it, although the model updates no parameters with the examples.

- Few-shot: provides a few examples
- One-shot: provides one example
- Zero-shot: provides no example

4.4 Results

Among the results of GPT-3 the paper (Brown et al. (2020)) demonstrates, we introduces some interesting ones here.

First, GPT-3 achieved SoTA in LAMBADA, while not in StoryCloze and HellaSwag.

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

Figure 4: Brown et al. (2020)

For QA tasks, GPT-3 beat SoTA of Fine-tuned models in Trivia QA.

Setting	NaturalQS	WebQS	TriviaQA
RAG (Fine-tuned, Open-Domain) [LPP+20]	**44.5**	**45.5**	**68.0**
T5-11B+SSM (Fine-tuned, Closed-Book) [RRS20]	36.6	44.7	60.5
T5-11B (Fine-tuned, Closed-Book)	34.5	37.4	50.1
GPT-3 Zero-Shot	14.6	14.4	64.3
GPT-3 One-Shot	23.0	25.3	**68.0**
GPT-3 Few-Shot	29.9	41.5	**71.2**

Figure 5: Brown et al. (2020)

GPT-3 surprisingly achieved SoTA in the translation from French to English and Dutch to English, although it is trained on dataset, 93% of which is English.

Setting	En→Fr	Fr→En	En→De	De→En	En→Ro	Ro→En
SOTA (Supervised)	**45.6**[a]	35.0 [b]	**41.2**[c]	40.2[d]	**38.5**[e]	**39.9**[e]
XLM [LC19]	33.4	33.3	26.4	34.3	33.3	31.8
MASS [STQ+19]	37.5	34.9	28.3	35.2	35.2	33.1
mBART [LGG+20]	-	-	29.8	34.0	35.0	30.5
GPT-3 Zero-Shot	25.2	21.2	24.6	27.2	14.1	19.9
GPT-3 One-Shot	28.3	33.7	26.2	30.4	20.6	38.6
GPT-3 Few-Shot	32.6	39.2	29.7	40.6	21.0	39.5

Figure 6: Brown et al. (2020)

The following table shows the accuracy of arithmetic tasks. E.g., 2D+ means 2 digit addition.

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 7: Brown et al. (2020)

GPT-3 seems to understand arithmetic calculation as humans do, since only 17 out of 2,000 addition problems and 2 out of 2,000 subtraction problems in the training dataset match the test dataset.

We can find another interesting result in SAT analogies, where GPT-3 beat the average score

of 57% among college applicants.

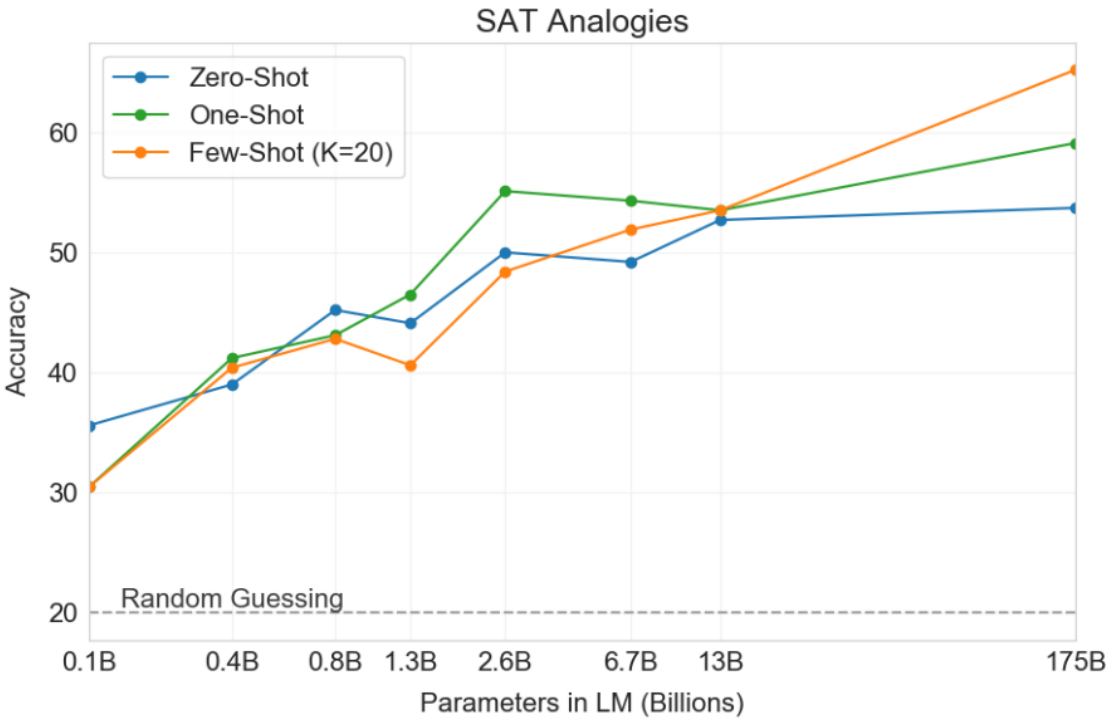

Figure 8: Brown et al. (2020)

Finally, mean human accuracy at detecting model generated articles is about 50%, which is chance level performance. I.e., articles generated by GPT-3 achieved the human level.

	Mean accuracy	95% Confidence Interval (low, hi)	t compared to control (p-value)	"I don't know" assignments
Control (deliberately bad model)	86%	83%–90%	-	3.6 %
GPT-3 Small	76%	72%–80%	3.9 (2e-4)	4.9%
GPT-3 Medium	61%	58%–65%	10.3 (7e-21)	6.0%
GPT-3 Large	68%	64%–72%	7.3 (3e-11)	8.7%
GPT-3 XL	62%	59%–65%	10.7 (1e-19)	7.5%
GPT-3 2.7B	62%	58%–65%	10.4 (5e-19)	7.1%
GPT-3 6.7B	60%	56%–63%	11.2 (3e-21)	6.2%
GPT-3 13B	55%	52%–58%	15.3 (1e-32)	7.1%
GPT-3 175B	52%	49%–54%	16.9 (1e-34)	7.8%

Figure 9: Brown et al. (2020)

5 Conclusion

Thus, GPT has evolved from the first version to the third one to address preceding issues while the model size monotononically has increased as follows.

	GPT-1	GPT-2	GPT-3
#parameters	117 million	1.5 billion	175 billion
#decoder layers	12	48	96
#context token size	512	1024	2048
#hidden layer	768	1600	12288
#batch size	64	512	3.2 million

GPT-1's idea of unsupervised pre-training and supervised fine-tuning with fewer data realizes the breakthrough in NLP. Moreover, the fact that GPT-3 without fine-tuning beat several SoTAs indicates the potential to solve various kinds of NLP problems with such huge language models with much less manual labour.

References

Brown, Tom, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, et al. 2020. "Language Models Are Few-Shot Learners." *Advances in Neural Information Processing Systems* 33: 1877–1901.

Child, Rewon, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. "Generating Long Sequences with Sparse Transformers." *arXiv Preprint arXiv:1904.10509*.

Hill, Felix, Antoine Bordes, Sumit Chopra, and Jason Weston. 2015. "The Goldilocks Principle: Reading Children's Books with Explicit Memory Representations." *arXiv Preprint arXiv:1511.02301*.

Kaplan, Jared, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. "Scaling Laws for Neural Language Models." *arXiv Preprint arXiv:2001.08361.*

Levesque, Hector, Ernest Davis, and Leora Morgenstern. 2012. "The Winograd Schema Challenge." In *Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning.*

Paperno, Denis, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. "The LAMBADA Dataset: Word Prediction Requiring a Broad Discourse Context." *arXiv Preprint arXiv:1606.06031.*

Radford, Alec, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. "Improving Language Understanding by Generative Pre-Training."

Radford, Alec, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. "Language Models Are Unsupervised Multitask Learners." *OpenAI Blog* 1 (8): 9.