Member-only story

OpenAI GPT: Generative Pre-Training for Language Understanding

Understanding Transformer-Based Self-Supervised Architectures

Rohan Jagtap · Follow
Published in DataSeries · 5 min read · Jul 4, 2020

Photo by Green Chameleon on Unsplash

Language Modeling is currently the biggest trend in NLP. All the major tasks in NLP follow the pattern of **self-supervised pre-training a corpus on the language model architecture followed by fine-tuning** the model for the required downstream task. Since this modeling is partially unsupervised (and partially supervised), this is also a use case of **semi-supervised training.**

In this article, I'll be delineating OpenAI GPT, which is one of the most important and fundamental models in language understanding that helped lay the foundation of language modeling. This model also is one of the pioneers in the burgeoning of NLP in a high number of training parameters with **110M parameters** (which may seem less at the date, however it was a great deal when it came out).

Generative Pre-Training

As mentioned earlier, GPT is one of the pioneers in Language Understanding and Modeling. Hence, it essentially proposes the concept of pre-training a language model on a huge corpus of data and then fine-tuning. This being said, we will further move on with the specifics of GPT.

The Architecture

Open AI GPT uses a **Transformer Decoder** architecture as opposed to BERT's Transformer Encoder architecture. I have already covered the difference between the Transformer Encoder and Decoder in this post; however, it is as follows:

- **The Transformer Encoder** is essentially a Bidirectional Self-Attentive Model, that uses all the tokens in a sequence to attend each token in that sequence

> *i.e. for a given word, the attention is computed using all the words in the sentence and not just the words preceding the given word in one of the left-to-right or right-to-left traversal order.*

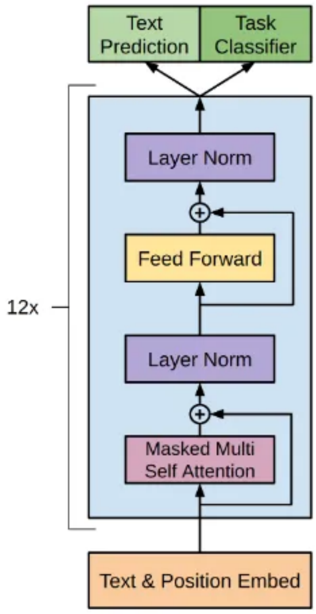
- While the **Transformer Decoder**, is a Unidirectional Self-Attentive Model, that uses only the tokens preceding a given token in the sequence to attend that token

> *i.e. for a given word, the attention is computed using only the words preceding the given word in that sentence according to the traversal order, left-to-right or right-to-left.*

— from BERT: Pre-Training of Transformers for Language Understanding

Thus, **GPT gets its auto-regressive nature from this directionality provided by the Transformer Decoder** as it uses just the previous tokens from the sequence to predict the next token.

Unsupervised Pre-Training



OpenAI GPT's Transformer Decoder Architecture from the Paper

The GPT model tries to maximize the following function:

$$L_1(\mathcal{U}) = \sum_i \log P(u_i | u_{i-k}, \ldots, u_{i-1}; \Theta)$$

Objective Function for Pre-training from the Paper

i.e. for a given corpus ***U,*** we maximize the probability that the token ***u_i,*** appears in the context given the tokens ***u_(i-k),…, u_(i-1)***. ***k*** is the window size for which we consider the previous tokens from the corpus and **Θ** are the model parameters.

Here, it calculates the probability and attention using the following:

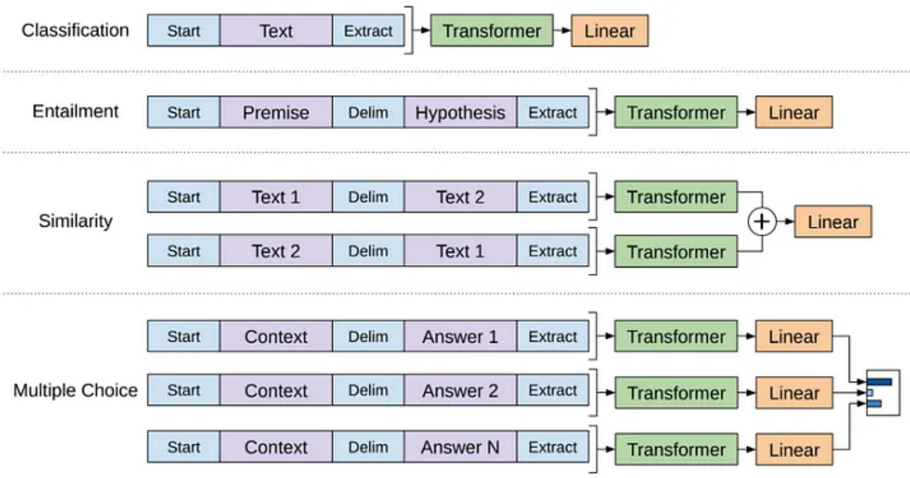
$$\begin{aligned} h_0 &= UW_e + W_p \\ h_l &= \texttt{transformer\_block}(h_{l-1}) \forall i \in [1, n] \\ P(u) &= \texttt{softmax}(h_n W_e^T) \end{aligned}$$

Attention and Probability of GPT from the Paper

which is basically the standard language model wherein, ***U*** is the input explained earlier, ***W_e*** are the embedding weights, ***W_p*** are the positional encodings, we calculate states for ***h_l*** using ***h_(l-1)*** using the transformer in accordance to the auto-regressive task, and finally smoothen the logits using a ***softmax.***

If you're unaware of these transformer specific terms, you can refer here to get an insight on the Transformer model.

Supervised Fine-Tuning



Input Representations for OpenAI GPT from the Paper

In these tasks, we consider a labeled dataset ***C*** and we try to maximize:

$$L_2(\mathcal{C}) = \sum_{(x,y)} \log P(y|x^1, \ldots, x^m).$$

Partial Objective Function for Fine-Tuning Task in GPT from the Paper

i.e. we maximize the ***log*** probability of the label ***y*** given the tokens ***x_1, …, x_m*** which is basically obtained using:

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

Function for Obtaining the Probability Scores using Attention from the Paper

One Important Fact about the OpenAI's GPT model is that by empirical studies, the authors have observed, that **before fine-tuning** the model, **unsupervised pre-training again on the labeled dataset** yield the best results. Though they admit that this didn't work out for smaller datasets.

So, combining both the objectives from the Unsupervised pre-training as well as the Supervised fine-tuning tasks, we get the combined objective function:

$$L_3(\mathcal{C}) = L_2(\mathcal{C}) + \lambda * L_1(\mathcal{C})$$

Complete Objective Function from the Paper

Furthermore, in the fine-tuning tasks, the inputs need to have some special organization to be clearly comprehensive for the model (as shown in the figure at the beginning of the section). We address these with respect to the corresponding downstream task:

- **Textual Entailment or Natural Language Inference (NLI):** The **text** and the **hypothesis** are concatenated and separated using a delimiter token ***$***.
- **Textual Similarity:** For this task, unlike entailment, there is no significance to the order in which the input sequences are fed to the network. Hence, both the attentions are calculated (i.e. in both the orders) and they are added element-wise.
- **Question Answering and Common Sense Reasoning:** For these tasks, we basically have a **context** with possibly an answer to the given **question** might be present; and optionally a **few options** of the possible answer of the given context. Here, the context (*z*) and question (*q*) are concatenated to each of the answer options (*a_k*); i.e. [*z; q; $; a_k*]. All these are fed to individual transformer decoders and then the scores are normalized using a ***softmax.***

Conclusion

We've seen the architecture and the working of the OpenAI GPT model. Although this model has kind of become obsolete in 2020, it is foundational for most of the modern language models and hence worth understanding.

You can find the pre-trained weights and the model architecture by huggingface transformers here

References

OpenAI GPT Original Paper: https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf

Transformers Explained

An exhaustive explanation of Google's Transformer model; from theory to implementation

towardsdatascience.com

huggingface/transformers

Artificial Intelligence · Deep Learning · NLP · Machine Learning · OpenAI

github.com

174

BERT: Pre-Training of Transformers for Language Understanding

Understanding Transformer-Based Self-Supervised Architectures

medium.com

Written by Rohan Jagtap

475 Followers · Writer for DataSeries

Follow

Immensely interested in AI Research | I read papers and post my notes on Medium

More from Rohan Jagtap and DataSeries

Rohan Jagtap in Towards Data Science

Transformers Explained

An exhaustive explanation of Google's Transformer model; from theory to...

· 9 min read · Jun 11, 2020

231 2

Raymond Ng in DataSeries

Build Your GPT Frontend

Leverage the OpenAI API to Create Your GPT Frontend

6 min read · Mar 26

90

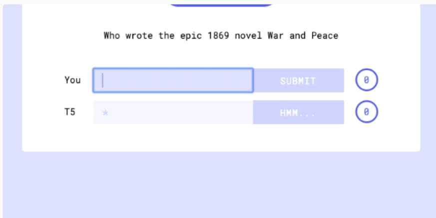
Muhammad Danyal in DataSeries

Public Claims and How to validate a JWT

JWT stand for JSON Web Token. It is a security validation mechanism widely used...

7 min read · Apr 14, 2020

437 3



Rohan Jagtap in Towards Data Science

T5: Text-To-Text Transfer Transformer

Understanding Transformer-Based Self-Supervised Architectures

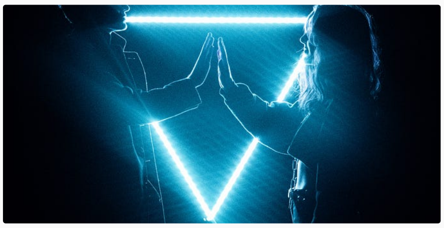
· 8 min read · Aug 1, 2020

293 2

See all from Rohan Jagtap

See all from DataSeries

Recommended from Medium



Wouter van Heeswijk, Ph... in Towards Data Scien...

Proximal Policy Optimization (PPO) Explained

The journey from REINFORCE to the go-to algorithm in continuous control

· 13 min read · Nov 29, 2022

171 2

E2Analyst in Predict

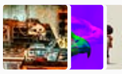
GPT-4: Everything you want to know about OpenAI's new AI model

From Text to Images, GPT-4 is Set to Revolutionize the Way We Interact with AI

· 8 min read · Mar 14

168 1

Lists

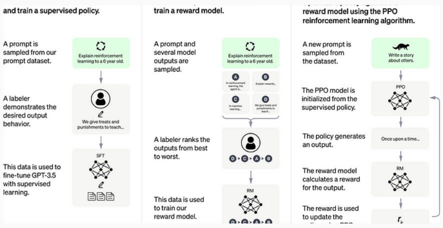


What is ChatGPT?

9 stories · 112 saves

Staff Picks

354 stories · 117 saves



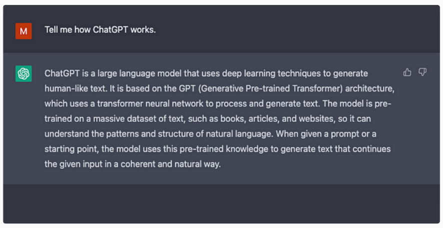
Isaac Kargar in AIGuys

Reinforcement Learning from Human Feedback, InstructGPT,...

Note: some parts of this blog post are generated by ChatGPT! :)

· 9 min read · Jan 7

71



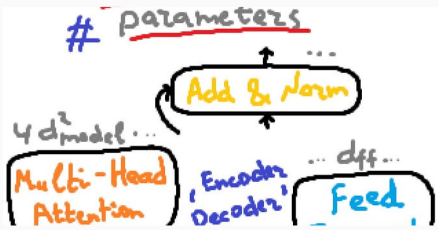
Molly Ruby in Towards Data Science

How ChatGPT Works: The Models Behind The Bot

A brief introduction to the intuition and methodology behind the chat bot you can't...

· 8 min read · Jan 30

7.5K 125



Cameron R. Wolfe, Ph.D. in Towards Data Science

Language Model Scaling Laws and GPT-3

Understanding why LLMs like GPT-3 work so well...

· 20 min read · Dec 10, 2022

52 2

Dmytro Nikolaiev (Dimid) in Towards Data Science

How to Estimate the Number of Parameters in Transformer models

An inside look at the Transformer Encoder/Decoder building blocks

· 10 min read · Jan 13

233 2

See more recommendations