Published in Walmart Global Tech Blog

Priya Shree Follow

Nov 9, 2020 · 14 min read · Listen

Save

The Journey of Open AI GPT models

Photo Credit : Image by Free-Photos from Pixabay

Generative Pre-trained Transformer (GPT) models by OpenAI have taken natural language processing (NLP) community by storm by introducing very powerful

language models. These models can perform various NLP tasks like question answering, textual entailment, text summarisation etc. without any supervised training. These language models need very few to no examples to understand the tasks and perform equivalent or even better than the state-of-the-art models trained in supervised fashion.

In this article we will cover the journey of these models and understand how they have evolved over a period of 2 years. We will be covering the following topics here:

1. Discussion of GPT-1 paper (Improving Language Understanding by Generative Pre-training).

2. Discussion of GPT-2 paper (Language Models are unsupervised multitask learners) and its subsequent improvements over GPT-1.

3. Discussion of GPT-3 paper (Language models are few shot learners) and the improvements which have made it one of the most powerful models NLP has seen till date.

This article assumes familiarity with the basics of NLP terminologies and transformer architecture.

Let us begin by understanding these papers one by one. To make this journey more comprehendible, I have segmented each paper into four sections: objectives and concepts discussed in the papers, the datasets used, the model architecture and implementation details, and their performance evaluations.

Improving Language Understanding by Generative Pre-training (GPT-1):

Prior to this work, most state-of-the-art NLP models were trained specifically on a particular task like sentiment classification, textual entailment etc. using supervised learning. However, supervised models have two major limitations:

i. They need large amount of annotated data for learning a particular task which is often not easily available.

ii. They fail to generalize for tasks other than what they have been trained for.

> *Unsupervised learning served as pre-training objective for supervised fine-tuned models, hence the name Generative Pre-training.*

Let us walk through the concepts and approaches discussed in this paper.

1. **Learning Objectives and Concepts:** This *semi-supervised* learning (unsupervised pre-training followed by supervised fine-tuning) for NLP tasks has following three components:

a. **Unsupervised Language Modelling** (Pre-training): For unsupervised learning, standard language model objective was used.

$$L_1(T) = \sum_i log\, P(t_i|t_{i-k}, \ldots, t_{i-1}; \theta) \qquad (i)$$

where T was the set of tokens in unsupervised data {t_1,...,t_n}, k was size of context window, θ were the parameters of neural network trained using stochastic gradient descent.

b. **Supervised Fine-Tuning**: This part aimed at maximising the likelihood of observing label y, given features or tokens x_1,...,x_n.

$$L_2(C) = \sum_{x,y} log\, P(y|x_1, \ldots, x_n) \qquad (ii)$$

where C was the labeled dataset made up of training examples.

Instead of simply maximising the objective mentioned in equation (ii), the authors added an **auxiliary learning objective** for supervised fine-tuning to get better

generalisation and faster conv… …ning objective was stated as:

$$L_3(C) = L_2(C) + \lambda L_1(C) \qquad (iii)$$

where $L_1(C)$ was the auxiliary objective of learning language model and λ was the weight given to this secondary learning objective. λ was set to 0.5.

Supervised fine-tuning was achieved by adding a linear and a softmax layer to the transformer model to get the task labels for downstream tasks.

c. **Task Specific Input Transformations**: In order to make minimal changes to the architecture of the model during fine tuning, inputs to the specific downstream tasks were transformed into ordered sequences. The tokens were rearranged in following manner:

— Start and end tokens were added to the input sequences.

— A delimiter token was added between different parts of example so that input could be sent as ordered sequence. For tasks like question answering, multiple choice questions etc. multiple sequences were sent for each example. E.g. a training example comprised of sequences for context, question and answer for question answering task.

2. **Dataset**: GPT-1 used the BooksCorpus dataset to train the language model. BooksCorpus had some 7000 unpublished books which helped training the language model on unseen data. This data was unlikely to be found in test set of downstream tasks. Also, this corpus had large stretches of contiguous text, which helped the model learn large range dependencies.

3. **Model Architecture and Implementation Details**: GPT-1 used 12-layer decoder only transformer structure with masked self-attention to train language model. The architecture of model remained same to a large extent as described in the original work on transformers. Masking helped achieve the language model objective wherein the language model did not have access to subsequent words to the right of current word.

Following are the implementation details:

a. **For Unsupervised Training:**

- Byte Pair Encoding (BPE) vocabulary with 40,000 merges was used.
- Model used 768-dimensional state for encoding tokens into word embeddings. Position embeddings were also learnt during training.
- 12 layered model was used with 12 attention heads in each self-attention layer.
- For position wise feed forward layer 3072-dimensional state was used.
- Adam optimiser was used with learning rate of 2.5e-4.
- Attention, residual and embedding dropouts were used for regularisation, with dropout rate of 0.1. Modified version of L2 regularisation was also used for non-bias weights.
- GELU was used as activation function.
- The model was trained for 100 epochs on mini-batches of size 64 and sequence length of 512. The model had 117M parameters in total.

b. **For Supervised Fine-tuning:**

- Supervised fine-tuning took as few as 3 epochs for most of the downstream tasks. This showed that the model had already learnt a lot about the language during pre-training. Thus, minimal fine-tuning was enough.
- Most of the hyper parameters from unsupervised pre-training were used for fine-tuning.

4. **Performance and Summary:**

> *GPT-1 performed better than specifically trained supervised state-of-the-art models in 9 out of 12 tasks the models were compared on.*

Another significant achievement by this model was its decent **zero-shot performance** on various tasks. The paper demonstrated that model had evolved in zero shot

performance on different NLP tasks like question-answering, schema resolution, sentiment analysis etc. due to pre-training.

GPT-1 proved that language model served as an effective pre-training objective which could help model generalize well. The architecture facilitated transfer learning and could perform various NLP tasks with very little fine-tuning. This model showed the power of generative pre-training and opened up avenues for other models which could unleash this potential better with larger datasets and more parameters.

Language Models are unsupervised multitask learners (GPT-2):

The developments in GPT-2 model were mostly in terms of using a larger dataset and adding more parameters to the model to learn even stronger language model. Let us look at the significant developments in GPT-2 model and the concepts discussed in the paper:

1. **Learning Objectives and Concepts**: Following are the two important concepts discussed in this paper in the context of NLP.

- **Task Conditioning**: We had seen that training objective of language model is formulated as P(output|input). However, GPT-2 aimed at learning multiple tasks using the same unsupervised model. To achieve that, the learning objective should be modified to P(output|input, task). *This modification is known as task conditioning, where the model is expected to produce different output for same input for different tasks.* Some models implement task conditioning at an architectural level where the model is fed both, the input and the task. For language models, the output, input and task, all are sequences of natural language. Thus, *task conditioning for language models* is performed by providing examples or natural language instructions to the model to perform a task. Task conditioning forms the basis for zero-shot task transfer which we will cover next.

- **Zero Shot Learning and Zero Short Task Transfer**: An interesting capability of GPT 2 is zero shot task transfer. Zero shot learning is a special case of zero shot task transfer where no examples are provided at all and the model understands the task based on the given instruction. Instead of rearranging the sequences, as was done for GPT-1 for fine-tuning, input to GPT-2 was given in a format which expected the model to understand the nature of task and provide answers. This was done to

emulate zero-shot task transfer behaviour. E.g. for English to French translation task, the model was given an English sentence followed by the word French and a prompt (:). The model was supposed to understand that it is a translation task and give French counterpart of English sentence.

2. **Dataset:** To create an extensive and good quality dataset the authors scraped the Reddit platform and pulled data from outbound links of high upvoted articles. The resulting dataset called WebText, had 40GB of text data from over 8 million documents. This dataset was used for training GPT-2 and was huge compared to Book Corpus dataset used for training GPT-1 model. All Wikipedia articles were removed from WebText as many test sets contain Wikipedia articles.

3. **Model architecture and Implementation Details:** GPT-2 had 1.5 billion parameters. which was 10 times more than GPT-1 (117M parameters). Major differences from GPT-1 were:

- GPT-2 had 48 layers and used 1600 dimensional vectors for word embedding.
- Larger vocabulary of 50,257 tokens was used.
- Larger batch size of 512 and larger context window of 1024 tokens were used.
- Layer normalisation was moved to input of each sub-block and an additional layer normalisation was added after final self-attention block.
- At initialisation, the weight of residual layers was scaled by 1/√N, where N was the number of residual layers.

The authors trained four language models with 117M (same as GPT-1), 345M, 762M and 1.5B (GPT-2) parameters. Each subsequent model had lower perplexity than previous one. *This established that the perplexity of language models on same dataset decreases with an increase in the number of parameters.* Also, the model with the highest number of parameters performed better on every downstream task.

4. **Performance and Summary:** GPT-2 was evaluated on several datasets of downstream tasks like reading comprehension, summarisation, translation, question answering etc. Let us look at some of those tasks and GPT-2's performance on them in detail:

- GPT-2 improved the then existing state-of-the-art for 7 out of 8 language modelling datasets in zero shot setting.
- Children's Book Dataset evaluates the performance on language models on categories of words like nouns, prepositions, named entities etc. GPT-2 increased the state-of-the-art accuracy approximately by 7% for common noun and named entity recognition.
- LAMBADA dataset evaluates the performance of models in identifying long range dependencies and predicting last word of a sentence. GPT-2 reduced the perplexity from 99.8 to 8.6 and improved the accuracy significantly.
- GPT-2 outperformed 3 out 4 baseline models in reading comprehension tasks in zero shot setting.
- In French to English translation task, GPT-2 performed better than most unsupervised models in zero shot setting but did not outperform the state-of-the-art unsupervised model.
- GPT-2 could not perform well on text summarisation and its performance was similar or lesser than classic models trained for summarisation.

> *GPT-2 was able to achieve state-of-the-art results on 7 out of 8 tested language modelling datasets in zero-shot.*

GPT-2 showed that training on larger dataset and having more parameters improved the capability of language model to understand tasks and surpass the state-of-the-art of many tasks in zero shot settings. The paper stated that with increase in the capacity of the model, the performance increased in log-linear fashion. Also, the drop in perplexity of language models did not show saturation and kept on decreasing with increase in number of parameters. As a matter of fact, GPT-2 under fitted the WebText dataset and training for more time could have reduced the perplexity even more. This showed that model size of GPT-2 was not the limit and building even larger language models would reduce the perplexity and make language models better at natural language understanding.

Language models are few shot learners (GPT-3):

In its quest to build very strong and powerful language models which would need no fine-tuning and only few demonstrations to understand tasks and perform them, Open AI built the GPT-3 model with 175 billion parameters. This model had 10 times more parameters than Microsoft's powerful Turing NLG language model and 100 times more parameters than GPT-2. Due to large number of parameters and extensive dataset GPT-3 has been trained on, it performs well on downstream NLP tasks in zero-shot and few-shot setting. Owing to its large capacity, it has capabilities like writing articles which are hard to distinguish from ones written by humans. It can also perform on-the-fly tasks on which it was never explicitly trained on, like summing up numbers, writing SQL queries and codes, unscrambling words in a sentence, writing React and JavaScript codes given natural language description of task etc. Let's understand the concepts and developments mentioned in GPT-3 paper along with some broader impacts and limitations of this model:

1. **Learning Objectives and Concepts**: Let us discuss the two concepts discussed in this paper.

- **In-context learning**: Large language models develop pattern recognition and other skills using the text data they are trained on. While learning the primary objective of predicting the next word given context words, the language models also start recognising patterns in data which help them minimise the loss for language modelling task. Later, this ability helps the model during zero-shot task transfer. When presented with few examples and/or a description of what it needs to do, the language models matches the pattern of the examples with what it had learnt in past for similar data and uses that knowledge to perform the tasks. This is a powerful capability of large language models which increases with the increase in the number of parameters of the model.

- **Few-shot, one-shot and zero-shot setting**: As discussed earlier, few, one and zero-shot settings are specialised cases of zero-shot task transfer. In few-shot setting, the model is provided with task description and as many examples as fit into the context window of model. In one-shot setting the model is provided exactly one example and in zero-shot setting no example is provided. With increase in capacity of model, few, one and zero-shot capability of model also improves.

2. **Dataset**: GPT-3 was trained on a mix of five different corpora, each having certain weight assigned to it. High quality datasets were sampled more often, and model was trained for more than one epoch on them. The five datasets used were Common Crawl, WebText2, Books1, Books2 and Wikipedia.

3. **Model and Implementation details**: The architecture of GPT-3 is same as GPT-2. Few major differences from GPT-2 are:

- GPT-3 has 96 layers with each layer having 96 attention heads.
- Size of word embeddings was increased to 12888 for GPT-3 from 1600 for GPT-2.
- Context window size was increased from 1024 for GPT-2 to 2048 tokens for GPT-3.
- Adam optimiser was used with $\beta_1=0.9, \beta_2=0.95$ and $\varepsilon= 10^{-8}$.
- Alternating dense and locally banded sparse attention patterns were used.

4. **Performance and Summary**: GPT-3 was evaluated on a host of language modelling and NLP datasets. GPT-3 performed better than state-of-the-art for language modelling datasets like LAMBADA and Penn Tree Bank in few or zero-shot setting. For other datasets it could not beat the state-of-the-art but improved the zero-shot state-of-the-art performance. GPT-3 also performed reasonably well in NLP tasks like closed book question answering, schema resolution, translation etc., often beating the state-of-the-art or performing comparable to fine-tuned models. For most of the tasks, the model performed better in few-shot setting as compared to one and zero-shot.

Apart from evaluating the model on conventional NLP task, the model was also evaluated on synthetic tasks like arithmetic addition, unscrambling of words, news article generation, learning and using novel words etc. For these tasks too, the performance increased with increase in number of parameters and the model performed better in few-shot setting than one and zero-shot.

5. **Limitations and Broader Impacts**: The paper discusses several weaknesses of GPT-3 model and areas open for improvement. Let's summarise them here.

- Though GPT-3 is able to produce high quality text, at times it starts losing coherency while formulating long sentences and repeats sequences of text over

and over again. Also, GPT-3 does not perform very well on tasks like natural language inference (determining that if a sentence implies other sentence), fill in the blanks, some reading comprehension tasks etc. The paper cites unidirectionality of GPT models as the probable cause for these limitations and suggests training bidirectional models at this scale to overcome these problems.

- Another limitation pointed by the paper is GPT-3's generic language modelling objective which weighs each token equally and lacks the notion of task or goal-oriented prediction of tokens. To counter this, the paper suggests approaches like augmentation of learning objective, use of reinforcement learning to fine tune models, addition of other modalities etc.

- Other limitations of GPT-3 include complex and costly inferencing from model due to its heavy architecture, less interpretability of the language and results generated by model and uncertainty around what helps the model achieve its few shot learning behaviour.

- Along with these limitations, GPT-3 carries potential risk of misuse of its human-like text generating capability for phishing, spamming, spreading misinformation or performing other fraudulent activities. Also, the text generated by GPT-3 possesses the biases of the language it is trained on. The articles generated by GPT-3 might have gender, ethnicity, race or religion bias. Thus, it becomes extremely important to use such models carefully and to monitor the text generated by them before its usage.

Ending Note:

This article summarises the journey and developments of OpenAI GPT models and their evolution over three papers. These models are undoubtedly very powerful language models and have revolutionised the domain of Natural Language Processing by performing plethora of tasks using just the instructions and few examples. Though these models are not at par with humans in natural language understanding, they have certainly shown a way forward to achieve that objective.

Glossary:

1. Auxiliary Learning Objective is an additional training objective or task that is learnt along with primary learning objective to improve the performance of the models

by making them more generic. This paper provides more details on this concept.

2. Masking refers to removing or replacing words in a sentence by some other dummy token such that the model does not have access to those words at the time of training.

3. Byte Pair Encoding is a data compression technique in which frequently occurring pairs of consecutive bytes are replaced with a byte not present in data to compress the data. To reconstruct the original data, a table containing mapping of replaced bytes is used. This blog explains BPE in detail.

4. Zero shot learning or behaviour refers to the ability of a model to perform a task without having seen any example of that kind in past. No gradients update happen during zero shot learning and the model is supposed to understand the task without looking at any examples.

5. Zero shot task transfer or meta-learning refers to the setting in which the model is presented with few to no examples, to make it understand the task. The term zero shot comes from the fact that no gradient updates are performed. The model is supposed to understand the task based on the examples and instruction.

6. Perplexity is the standard evaluation metric for language models. Perplexity is the inverse probability of test set which is normalised by number of words in test set. Language models with lower perplexity are considered to better than ones with higher perplexity. Read this blog for more explanation on perplexity.

References:

1. Radford, A., Narasimhan, K., Salimans, T. and Sutskever, I., 2018. Improving language understanding by generative pre-training.

2. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D. and Sutskever, I., 2019. Language models are unsupervised multitask learners. *OpenAI blog*, *1*(8), p.9.

3. Brown, Tom B., Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan et al. “Language models are few-shot learners.” *arXiv preprint arXiv:2005.14165* (2020).

4. Rei, M., 2017. Semi-supervised multitask learning for sequence labeling. *arXiv preprint arXiv:1704.07156*.

5. Waswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L. and Polosukhin, I., 2017. Attention is all you need. In *NIPS*.

Note: For purpose of brevity, the links to blogs have not been repeated in references.