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

In this paper, we explore a semi-supervised approach for language understanding tasks using a combination of unsupervised pre-training and supervised fine-tuning. Our goal is to learn a universal representation that transfers with little adaptation to a wide range of tasks. We assume access to a large corpus of unlabeled text and several datasets with manually annotated training examples (target tasks). Our setup does not require these target tasks to be in the same domain as the unlabeled corpus. We employ a two-stage training procedure. First, we use a language modeling objective on the unlabeled data to learn the initial parameters of a neural network model. Subsequently, we adapt these parameters to a target task using the corresponding supervised objective. For our model architecture, we use the Transformer [62], which has been shown to perform strongly on various tasks such as machine translation [62], document generation [34], and syntactic parsing [29]. This model choice provides us with a more structured memory for handling long-term dependencies in text, compared to alternatives like recurrent networks, resulting in robust transfer performance across diverse tasks. During transfer, we utilize task-specific input adaptations derived from traversal-style approaches [52], which process structured text input as a single contiguous sequence of tokens. As we demonstrate in our experiments, these adaptations enable us to fine-tune effectively with minimal changes to the architecture of the pre-trained model. We evaluate our approach on four types of language understanding tasks – natural language inference, question answering, semantic similarity, and text classification. Our general task-agnostic model outperforms discriminatively trained models that employ architectures specifically crafted for each task, significantly improving upon the state of the art in 9 out of the 12 tasks studied. For instance, we achieve absolute improvements of 8.9% on commonsense reasoning (Stories Cloze Test) [40], 5.7% on question answering (RACE) [30], 1.5% on textual entailment (MultiNLI) [66] and 5.5% on the recently introduced GLUE multi-task benchmark [64]. We also analyzed zero-shot behaviors of the pre-trained model on four different settings and demonstrate that it acquires useful linguistic knowledge for downstream tasks.

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.

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):

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.

We demonstrate that scaling up language models greatly improves task-agnostic, few-shot performance, sometimes even becoming competitive with prior state-ofthe-art fine-tuning approaches. Specifically, we train GPT-3, an autoregressive language model with 175 billion parameters, 10x more than any previous nonsparse language model, and test its performance in the few-shot setting. For all tasks, GPT-3 is applied without any gradient updates or fine-tuning, with tasks and few-shot demonstrations specified purely via text interaction with the model. GPT-3 achieves strong performance on many NLP datasets, including translation, question-answering, and cloze tasks. We also identify some datasets where GPT3’s few-shot learning still struggles, as well as some datasets where GPT-3 faces methodological issues related to training on large web corpora.

In this work, we empirically test whether scaling continues to improve performance by extrapolating the previously identified phenomena another two orders of magnitude. We train a 175 billion parameter autoregressive language model, which we call GPT-3, and measure its transfer learning abilities. Our findings are summarized in Figure 1.1. We observe that one- and few-shot performance is often much higher than true zero-shot performance leading us to suggest that language models can also be understood as meta-learners where slow outer-loop gradient descent based learning is combined with fast “in-context” learning implemented within the context activations of the model. Broadly, on NLP tasks GPT-3 achieves promising results in the zero- and one-shot settings, and in the few-shot setting is sometimes competitive with or even occasionally surpasses state-of-the-art (despite state-of-the-art being held by fine-tuned models). For example, GPT-3 achieves 81.5 F1 on CoQA in the zero-shot setting, 84.0 F1 on CoQA in the one-shot setting, and 85.0 F1 in the few-shot setting. Similarly, GPT-3 achieves 64.3% accuracy on TriviaQA in the zero-shot setting, 68.0% in the one-shot setting, and 71.2% in the few-shot setting, the last of which is state-of-the-art relative to fine-tuned models operating in the same closed-book setting. We additionally train a series of smaller models (ranging from 125 million parameters to 13 billion parameters) in order to compare their performance to GPT-3 in the zero-, one- and few-shot settings. In general, we find relatively smooth scaling for most tasks with model capacity in all three settings; one notable pattern is that the gap between zero-, one-, and few-shot performance often grows with model capacity, perhaps suggesting that larger models are more proficient meta-learners.

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 β\_1=0.9,β\_2=0.95 and ε= 10^(-8).

Alternating dense and locally banded sparse attention patterns were used.