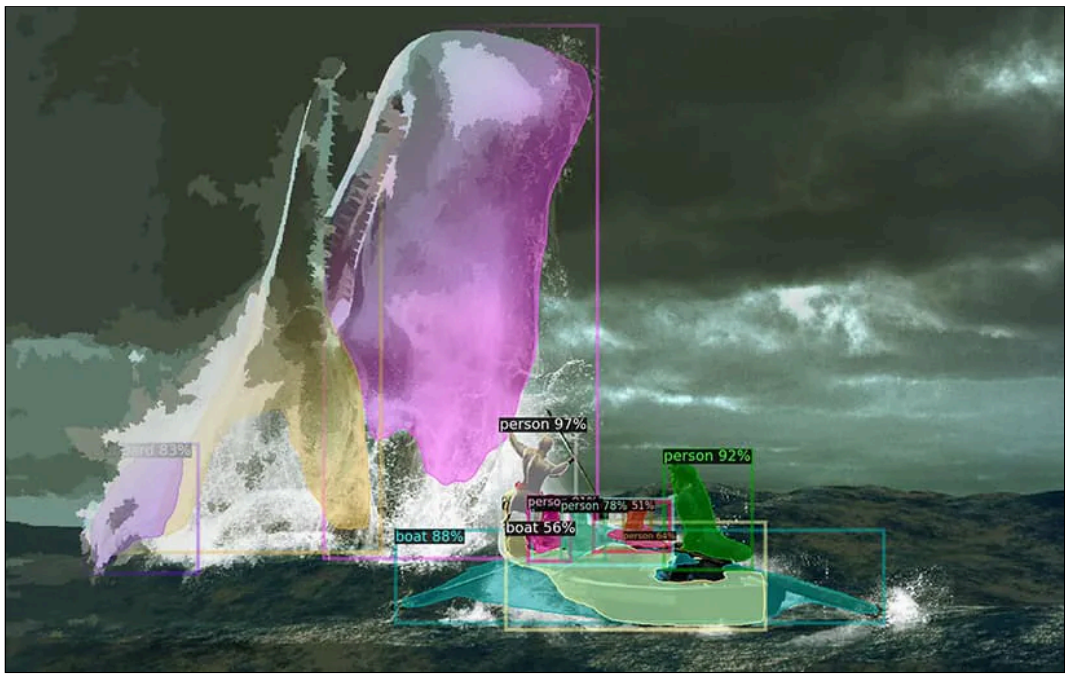
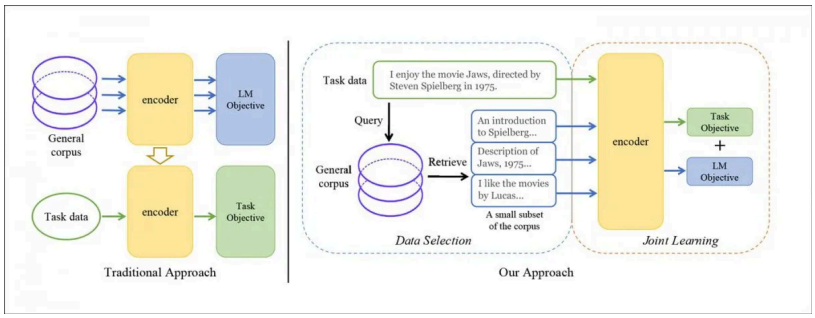
Creating a GPT-Style Language Model for a Single Question

Originally published at https://www.unite.ai/creating-a-gpt-style-language-model-for-a-single-question/



Researchers from China have developed an economical method for creating GPT-3-style Natural Language Processing systems while avoiding the increasingly prohibitive expense of time and money involved in training up high volume datasets – a growing trend which otherwise threatens to eventually relegate this sector of AI to FAANG players and high-level investors.

The proposed framework is called *Task-Driven Language Modeling* (TLM). Instead of training a huge and complex model on a vast corpus of billions of words and thousands of labels and classes, TLM instead trains a far smaller model that actually incorporates a query directly inside the model.



Left, a typical hyperscale approach to high volume language models; right, TLM's slimline method to explore a large language corpus on a per-topic or per-questio
https://arxiv.org/pdf/2111.04130.pdf

Effectively, a unique NLP algorithm or model is produced in order to answer a single question, instead of creating an enormous and unwieldy general language model that can answer a wider variety of questions.

In testing TLM, the researchers found that the new approach achieves results that are similar or better than Pretrained Language Models such as RoBERTa-Large, and hyperscale NLP systems such as OpenAI's GPT-3, Google's TRILLION Parameter Switch Transformer Model, Korea's HyperClover, AI21 Labs' Jurassic 1, and Microsoft's Megatron-Turing NLG 530B.

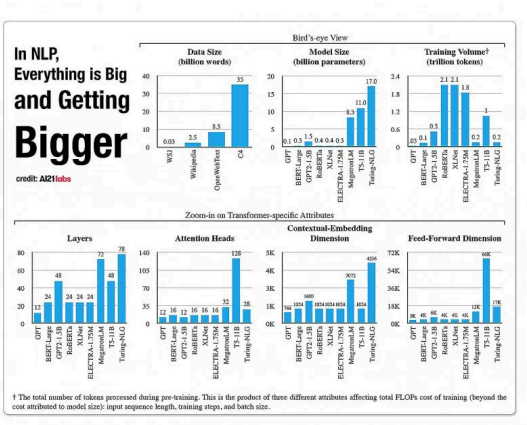
In trials of TLM over eight classification datasets across four domains, the authors additionally found that the system reduces the training FLOPs (floating point operations per second) required by two orders of magnitude. The researchers hope that TLM can 'democratize' a sector that is becoming increasingly elite, with NLP models so large that they cannot realistically be installed locally, and instead sit, in the case of GPT-3, behind the expensive and limited-access APIs of OpenAI and, now, Microsoft Azure.

The authors state that cutting training time by two orders of magnitude reduces training cost over 1,000 GPUs for one day to a mere 8 GPUs over 48 hours.

The new report is titled *NLP From Scratch Without Large-Scale Pretraining: A Simple and Efficient Framework*, and comes from three researchers at Tsinghua University in Beijing, and a researcher from China-based AI development company Recurrent AI, Inc.

Unaffordable Answers

The cost of training effective, all-purpose language models is increasingly becoming characterized as a potential 'thermal limit' on the extent to which performant and accurate NLP can really become diffused in culture.



Statistics on the growth of facets in NLP model architectures, from a 2020 report by A121 Labs. Source: https://arxiv.org/pdf/2004.08900.pdf

In 2019 a researcher calculated that it costs $61,440 USD to train the XLNet model (reported at the time to beat BERT in NLP tasks) over 2.5 days on 512 cores across 64 devices, while GPT-3 is estimated to have cost $12 million to train – 200 times the expense of training its predecessor, GPT-2 (though recent re-estimates claim it could be trained now for a mere $4,600,000 on the lowest-priced cloud GPUs).

Subsets of Data Based on Query Needs

Instead, the new proposed architecture seeks to derive accurate classifications, labels and generalization by using a query as a kind of filter to define a subset of information from a large language database that will be trained, together with the query, in order to provide answers on a limited topic.

The authors state:

> *'TLM is motivated by two key ideas. First, humans master a task by using only a small portion of world knowledge (e.g., students only need to review a few chapters, among all books in the world, to cram for an exam).*
>
> *'We hypothesize that there is much redundancy in the large corpus for a specific task. Second, training on supervised labeled data is much more data efficient for downstream performance than optimizing the language modeling objective on unlabeled data. Based on these motivations, TLM uses the task data as queries to retrieve a tiny subset of the general corpus. This is followed by jointly optimizing a supervised task objective and a language modeling objective using both the retrieved data and the task data.'*

Besides making highly effective NLP model training affordable, the authors see a number of advantages to using task-driven NLP models. For one, researchers can enjoy greater flexibility, with custom strategies for sequence length, tokenization, hyperparameter tuning and data representations.

The researchers also foresee the development of hybrid future systems which trade off limited pre-training of a PLM (which is otherwise not anticipated in the current implementation) against greater versatility and generalization against training times. They consider the system a step forward for the advancement of in-domain zero-shot generalization methods.

Testing and Results

TLM was tested on classification challenges in eight tasks over four domains – biomedical science, news, reviews and computer science. The tasks were divided into high-resource and low-resource categories. High resource tasks included over 5,000 task data, such as AGNews and RCT, among others; low-resource tasks included ChemProt and ACL-ARC, as well as the HyperPartisan news detection dataset.

The researchers developed two training sets titled Corpus-BERT and Corpus-RoBERTa, the latter ten times the size of the former. The experiments compared general Pretrained Language Models BERT (from Google) and RoBERTA (from Facebook) to the new architecture.

The paper observes that though TLM is a general method, and should be more limited in scope and applicability than broader and higher-volume state-of-the-art models, it is able to perform close to domain-adaptive fine-tuning methods.

Model	#Param	FLOPs[1]	Data[2]	AGNews	Hyp.	Help.	IMDB	ACL.	SciERC	Chem.	RCT	Avg.
BERT-Base[3]	109M	2.79E19	16GB	93.50 ±0.15	91.93 ±1.74	69.11 ±0.17	93.77 ±0.22	69.45 ±2.90	80.98 ±1.07	81.94 ±0.38	87.00 ±0.06	83.46
BERT-Large[3]	355M	9.07E19	16GB	93.51 ±0.40	91.62 ±0.69	69.39 ±1.14	**94.76** ±0.09	69.13 ±2.93	**81.37** ±1.35	**83.64** ±0.41	**87.13** ±0.09	**83.82**
TLM *(small-scale)*	109M	2.74E18	0.91GB	**93.74** ±0.20	**93.53** ±1.61	**70.54** ±0.39	93.08 ±0.17	**69.84** ±3.69	80.51 ±1.53	81.99 ±0.42	86.99 ±0.03	83.78
RoBERTa-Base[3]	125M	1.54E21	160GB	**94.02** ±0.15	93.53 ±1.61	70.45 ±0.24	**95.43** ±0.16	68.34 ±7.27	81.35 ±0.63	82.60 ±0.53	87.23 ±0.09	84.12
TLM *(medium-scale)*	109M	8.30E18	1.21GB	93.96 ±0.18	**94.05** ±0.96	**70.90** ±0.73	93.97 ±0.10	**72.37** ±2.11	**81.88** ±1.92	**83.24** ±0.36	**87.28** ±0.10	**84.71**
RoBERTa-Large[3]	355M	4.36E21	160GB	94.30 ±0.23	**95.16** ±0.00	70.73 ±0.62	**96.20** ±0.19	**72.80** ±0.62	82.62 ±0.68	84.62 ±0.50	**87.53** ±0.13	85.50
TLM *(large-scale)*	355M	7.59E19	3.64GB	**94.34** ±0.12	**95.16** ±0.00	**72.49** ±0.33	95.77 ±0.24	72.19 ±1.72	**83.29** ±0.95	**85.12** ±0.85	87.50 ±0.12	**85.74**

[1] The total training compute (FLOPs) is calculated by (6 × Total_Training_Tokens × Parameter_Size) as in (Brown et al., 2020). For TLM, FLOPs are reported as the averaged result over eight tasks.
[2] The size of data selected from general corpus that are actually used in training. For TLM, it is reported by averaging over eight tasks.
[3] The BERT-Base and BERT-Large are pretrained by (Devlin et al., 2018) and RoBERTa-Base and RoBERTa-Large are pretrained by (Liu et al., 2019). We finetuned them to obtain the results over the eight tasks.

Results from comparing the performance of TLM against BERT and RoBERTa-based sets. The results list an average F1 score across three different training scal of parameters, total training compute (FLOPs) and size of training corpus.

The authors conclude that TLM is capable of achieving results that are comparable or better than PLMs, with a substantial reduction in FLOPs needed, and requiring only 1/16th of the training corpus. Over medium and large scales, TLM apparently can improve performance by 0.59 and 0.24 points on average, while reducing training data size by two orders of magnitude.

> *'These results confirm that TLM is highly accurate and much more efficient than PLMs. Moreover, TLM gains more advantages in efficiency at a larger scale. This indicates that larger-scale PLMs might have been trained to store more general knowledge that is not useful for a specific task.'*

Writer on machine learning, domain specialist in human image synthesis. Former head of research content at Metaphysic.ai, until its dissolution into DNEG's Brahma.ai.
Portfolio site: martinanderson.ai
Contact: martin@martinanderson.ai