Evaluating Performance of Different Domains for Pretraining Language Models

Noah Ponto
ponton@uw.edu

Rthvik Raviprakash
rravipra@uw.edu

Shreshth Kharbanda
skhar@uw.edu

1. Introduction

1.1. State of related work

Generative language models are an area of study which stand to improve our world by making it easier for people to write their thoughts and share their ideas. State of the art models such at BERT [3] and GPT-3 [2] have proven to perform extremely well on a wide variety of NLP tasks. Pretraining language models is a valuable tool which can allow complex models to be more quickly repurposed for more specific tasks [8]. There has been research which explores pretraining these language models on more specific domains such as computer code [4], scientific papers [1], or biomedical text [6].

1.2. Problem statement

Pretraining is clearly a useful tool for expanding the applications of language models, but what data will produce the best results when used in pretraining? Many of the leading language models have relied on the vast volume of data offered by sources like Wikipedia, novels, web crawl corpuses, and even social media. The commonality across these pieces of research is a complete focus on the volume of data with less thought about the exact pretraining data being used.

1.3. Unique insight

In our project, we would like to examine the effect of pretraining a generative language model on different text domains. We plan to take a fixed design for a language model with constant hyperparameters, and pretrain on a set of initial texts. We will then fine tune on a set of different text domains and evaluate the models' performance in terms of perplexity.

1.4. Technical challenges

One technical challenge we will likely encounter is the limited availability of cleaned and ready-to-use pretraining data for certain text domains. While large amounts of text are available for popular and common domains such as news articles and scientific papers, other domains such as legal documents or medical records will likely have limited pretraining data available that is ready-to-use. Since we're constrained by time, cleaning the data we retrieve likely won't be efficient so we would want to find ready-to-use datasets. This could impact the quality of the language models as well since we are able to train and fine-tune on the specific domains with ready-to-use datasets. Additionally, selecting appropriate hyperparameters for the language models can be a challenge. While we plan to keep hyperparameters constant across models to ensure consistency, selecting the initial hyperparameters can require trial and error to find the optimal settings, which is likely to be one challenge that we will face. Finally, while there are several useful metrics for evaluating a language model;s performance, one challenge might be to decide on how to objectively interpret and compare the models, which could make it challenging to draw clear conclusions from our results.

1.5. Experiment plan

We plan to objectively explore the differences between models pretrained on various domains. We will do this by pretraining our model on different domains, then fine-tuning and testing on one controlled domain. In our test set we plan to take the perplexity, which would be the possibility of getting the next token correct. We're leaning towards using perplexity for our objective score over accuracy because typically accuracy is ideal for classification tasks whereas when it comes to NLP it's nearly impossible to get the input and output precisely the same, but perplexity can enable us to get a better understanding of how the model is performing.

For pretraining the model, we plan to experiment with text from the domains of ancient greek philosophical writings, poetry, and movie dialogues. Finally, we plan to fine tune and test all three of those pretrained models on a corpus of text from the news reports. At the time of the project proposal we have already found text corpuses for all of our propsed domains.

Here are our steps:

1. Collect and preprocess data for the 3 steps involved,

ensuring that each dataset is representative of its respective domain

2. Pretrain a generative language model on a large standard text corpus in the decided domain using a fixed set of hyperparameters

3. Fine-tune the pre-trained language model on each of the collected datasets, ensuring that hyperparameters are kept constant across all models

4. Train each model until a predetermined number of epochs are completed

5. Evaluate the performance of each model by computing perplexity on a test dataset that is separate from the pretraining and fine-tuning

6. Perform a qualitative analysis of the generated text for each model to evaluate how well the model captures the specific domain-specific language and terminology

7. Compare the performance and evaluate the impact of pretraining on different text domains

1.6. Expected outcome

The expected outcome would be that the model performance after fine-tuning will be much more improved that the model performance without fine-tuning which is mainly because of it being pre-trained. Pretraining is most useful for small and medium-sized datasets, which are most common in commercial applications. However, even for large datasets, pre-training improves performance [5]. Usually multi-task fine-tuning followed by individual task fine-tuning of a model outperforms multitask fine-tuning alone [7].

Further, we expect to conclude that fine-tuning is mainly useful in domains where data are scarce such as medical records or legal documents. Even in language translation for a language that is so-called ‘isolated’ or not well known, fine-tuning it on a different or a similar domain can help improve the translation performance. Overall, we aim to conclude that fine-tuning will have an improvement in the performance or the performance might be the same but it will never be worse.

Another addition that we would like to add here is that pretraining on one domain and fine-tuning on another domain which is not that similar to the pre-trained domain might not necessarily improve the performance in this case. It would also depend on the final task on which we are evaluating our performance and the previous domains it has been trained on. This is something that we would like to explore as well as a part of our expected results. We expect that even if there is some similarity in the final task to the domain/domains that the model has been fine-tuned on there will be an improvement in the model performance.

References

[1] Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: A pretrained language model for scientific text. In *Conference on Empirical Methods in Natural Language Processing*, 2019. 1

[2] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, T. J. Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *ArXiv*, abs/2005.14165, 2020. 1

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805, 2019. 1

[4] Zhangyin Feng, Daya Guo, Duyu Tang, Nan Duan, Xiaocheng Feng, Ming Gong, Linjun Shou, Bing Qin, Ting Liu, Daxin Jiang, and Ming Zhou. Codebert: A pre-trained model for programming and natural languages. *ArXiv*, abs/2002.08155, 2020. 1

[5] Jeremy Howard and Sebastian Ruder. Fine-tuned language models for text classification. *CoRR*, abs/1801.06146, 2018. 2

[6] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36:1234 – 1240, 2019. 1

[7] Wei Liu, Lei li, Zuying Huang, and Yinan Liu. Multilingualwikipedia summarization and title generation on low resource corpus. pages 17–25, 12 2019. 2

[8] Alec Radford and Karthik Narasimhan. Improving language understanding by generative pre-training. 2018. 1