Augmenting Textual Datasets Using Generative Pretrained Transformers

Almog Zur, Ari Granevich

January 27, 2023

1 Introduction

Labeling data can get expensive, and oftentimes we have a smaller dataset than we would like. In addition, we might want to add certain invariants to the data, teaching our model to disregard irrelevant traits in the data. For these purposes augmentation is widely used. Text is an especially complex type of data, and analyzing it is a well studied but not completely solved problem. This makes purposeful augmenting of textual datasets difficult, unlike images which have a variety of easy augmentations. Generative textual models store a lot of information about natural language and its invariants, which makes them ideal for such tasks.

Some research has already been done on the subject. Yoo et al. [YPK+21] Employed GPT3 [BMR+20] along with a small number of randomly selected samples to create new samples for classification. In addition they used soft labels for the generated samples, allowing for better knowledge distillation for the larger model to the small student. Balkus et al. [BY22] has tested using GPT3's completion endpoint to augment a dataset by randomly selecting samples with the same label, encouraging GPT3 to generate a sample of the same label.

The goal of this project is to perform transfer learning by using a large generative pretrained transformer (GPT) to augment a small number of samples into a larger dataset which should allow for better learning. Due to technical limitations, in this project specifically we will use GPT2 [RWC+19].

2 Methodology

The code can be found here

Our test is composed of two parts. First, we use GPT2 and the original dataset to generate new samples for our dataset. Next, we combine our original dataset with the synthetic dataset and train a simple transformer model on it to measure the increase in performance compared to the control, which is the original dataset alone.

2.1 Dataset

We used the IMDB movie reviews sentiment analysis dataset [MDP+11]. This dataset is fairly large so we only used a small subset of samples. GPT models have a token limitation so for compatibility we opted to reduce the dataset to its shortest samples. We reduced the dataset to samples of at most 300 characters. In total, we used 776 samples.

2.2 Augmentation

We used GPT2-large, as a compromise between speed and quality. GPT2 has 4 versions from "small" to "xl" in an exponentially increasing number of parameters.

In the augmentation process we fed 2 samples to GPT2 In the format " $< positive/negative > movie\,review :< review >$ ", and let GPT2 complete the sequence. We then got a sequence of samples with a similar format. We cleaned the samples by taking only samples which fit the original format and discarding, for example, samples like "amazing movie review : ...". In total after an hour of generation we got 1615 samples.

2.3 Classification

For the classification task we used the transformer classifier used in tutorial 7. This model has shows good performance for the IMDB dataset and we hoped it would also function well for our subset of samples. The model is composed of a transformer encoder, performing self-attention on the sequence's embedding. The output of the encode is passed through a *max* function and then a linear layer which converts it to the output dimension. For hyper parameters we used the default parameters: embedding dimension: 32, 1 transformer layer, 2 heads, and a transformer hidden dimension of 128. For the training process we trained for 10 epochs using adamW with a learning rate of 0.01 and batch size of 128. We used the standard cross entropy loss function.

3 Results

Training the model on the original 776 samples barely allows the model to learn, with a test accuracy of 52% on average. Training the model on the augmented data, all 2391 samples, results in worse performance with a test accuracy of 50% on average. Since the label is binary this is equal to purely guessing. While we cannot conclusively determine the model worsened, we can confidently assert the augmented data is not beneficial to test accuracy.

3.1 Possible Issues

Since the literature already contains examples of augmentation succeeding we were surprised to see these results. After considering it we reached two possible explanations:

1. More data means more iterations which could lead the model to overfit
2. The augmentation caused a label imbalance, which decreased classifier accuracy.
3. The augmentation generated sub-par data, perhaps with wrong labels or nonsensical text, resulting in a decrease in model performance.

We tested each hypothesis:

3.1.1 Overfitting

Decreasing the number of epochs did not change test accuracy, suggesting the model is not overfitting due to an increased number of iterations. In addition when using a similarly sized amount of the original dataset we get a test accuracy of 65% on average, which means the augmented data itself is also a major factor.

3.1.2 Label Imbalance

After forcefully balancing the augmented data using duplication we tested again, and got a test accuracy of 52% on average. This suggests the label imbalance is indeed an issue. On the other hand we still did not get an improvment even after balancing the labels, suggesting the generated data is at best useless.

3.1.3 Sub-Par Data

This seems to be the biggest factor. As mentioned a data set with 2500 samples should get a test accuracy of 65% whereas the augmented dataset got a test accuracy of 50%. Looking closely at the data we can see some examples of confusing reviews, such as:

"not a bad movie IMHO, just one I would not be too in the mood for." For a positive review, and "Not what it could have been as it got pushed to a better release date. This movie was about a man who becomes a "demon hunter" and after two years of practice destroys all of the demons and is accepted back into the society. He tries to protect his wife (a woman) but there are some demons who will do anything to capture and torment her. The problem is that he is not always a good "demon hunter" because he must fight against all the human enemies who would do what he would just to get the girl. When things go wrong he is forced to re-evaluate the whole relationship and his choices. It is

not exactly "good" movie but I really liked it. It got an 8/10 but didn't meet my expectations." For a negative review.

While most reviews are good, it is possible they are too close to the generating samples to be any use.

4 Conclusion

We have tested the idea of augmenting text datasets using GPT models, and got discouraging results. With the evidence of success in the literature we still believe this method has a place and think the following might be good directions to study:

- Generating more data to see if more sub-par data can be useful in spite of its quality
- Trying to generate better data by using a stronger GPT model (GPT3 was developed with the goal of improving learning from a small number of samples, which is what we need in this situation)
- Balkus et al. examined a genetic algorithm for selecting which samples to feed into GPT3. Perhaps a similar algorithm can allow for the generation of high quality data using a simple model like GPT2.

References

[BMR+20] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[BY22] Salvador Balkus and Donghui Yan. Improving short text classification with augmented data using gpt-3. *arXiv preprint arXiv:2205.10981*, 2022.

[MDP+11] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[RWC+19] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[YPK+21] Kang Min Yoo, Dongju Park, Jaewook Kang, Sang-Woo Lee, and Woomyeong Park. Gpt3mix: Leveraging large-scale language models for text augmentation. *arXiv preprint arXiv:2104.08826*, 2021.