Machine learning for statistical NLP: Advanced LT2326

Victoria Filippatou
University of Gothenburg

1. Introduction

The following project represents an exploration of the capabilities and limitations of natural language generation with transformers, with a particular focus on the GPT2 model, which is largely recognized and used. More specifically, the objective of this endeavor was to delve into natural language generation by fine-tuning the GPT2 model to generate poems based on a prompt that would define the poem's theme or form. To achieve this, the GPT2 model was retrained on a curated dataset of poems that was sourced from the Kaggle platform. The expected result would be a poem-like text, relevant to the topic word given and somewhat creative, or at least more creative than the output given by the original raw model.

The motivation behind this undertaking was a sense of curiosity to explore how efficiently language models can grasp symbolic meaning, like metaphors, or abstract concepts, and rhyme schemes. Additionally, I wanted to understand how or if they can generate more easily such an output after fine-tuning. Furthermore, I believe it is interesting to see and compare the ways human vs artificial intelligence creativity materialize and if they intersect in any meaningful way. Having in mind that there is a plethora of similar projects that have been undertaken before and that this topic can be considered well researched, I wanted to experiment with language generation with transformers because it is something I had never done before, but also because it seemed fun.

2. Background

As mentioned earlier, delving into examining language models for creative language generation is not something new, but something that has developed over the years. GPT2 is the generative model that has garnered the most attention and has been tested to generate coherent and contextually meaningful text, especially when it comes to poetry. But even before that, deep neural networks, such as RNNs and CNNs have been tasked with the same goal.

Xian-chao et al (2017) explored the use of four different types of character-level language models (NLMs) for generating Japanese haiku poems. They collected their data from web collections and query logs derived from the Rinna chatbot, and they later evaluated the results by perplexity measurements. This evaluation demonstrated that the RNN-LSTM model performs slightly better than the other three models for the Haiku web dataset, whereas the RCNN model is performing the best in the haiku user dataset. These results indicate that they might have been the most effective at generating coherent haikus. However, the authors of the paper do not elaborate further on the quality of the produced haiku poems.

Moving to transformer generated poetry, Bena and Kalita (2020) finetuned the GPT2 model for creative poem generation, that expresses emotions and uses dream-like language. They first train different models for each of these tasks, and they evaluate the generated outputs based on automated and human evaluation methods. The results show that the models are capable of generating poetry that elicits emotions and exhibits dream poetry characteristics, while at the same time maintaining grammatical integrity, semantic cohesion, and high text quality in terms of lexical choice. Overall, this paper is one of the pioneers of contributing to the field of automatic poetry generation by incorporating creativity and emotion.

Additionally, Miceli discussed the development of a transformer-based model called 'Haikoo', for generating haiku poetry. This model consisted of a fine-tuned GPT2 model and a Plug and Play Language Model (PPLM) component for controlling the generated output. The results showed that 'Haikoo' outperforms previous state-of-the-art models as well as the baseline one, that was used in evaluation. The evaluation process included perplexity and human judgments as metrics, with the automated evaluation indicating the superiority of 'Haikoo' against the baseline model. A more comprehensive human evaluation was used for rating features like sensicality, wisdom, and overall quality, and ranked the 'Haikoo' model notably higher than the baseline one, especially for the sensicality feature. The generated haikus were distinguished for their cohesion and overall quality, showcasing the potential of transformer-approached methods in haiku and poem generation in general.

Based on these studies I expect achieving similar outcomes in this smaller scale project.

3. Methodology

The foundation of this project was sourcing my dataset from Kaggle, which included poems categorized according to their topic or form. The sub-datasets proved to be remarkably diverse, assembling a total amount of approximately 8000 poems. However, sometimes big datasets and long data can have their disadvantages, so after running into multiple 'CUDA out of memory' errors, I decided to only include poems of length no more than 300. This decision resulted in a significantly smaller dataset of only 295 poems in my topics dataset! This proved to solve the CUDA errors, but it was in the expense of the learning of the model.

As for the preprocessing of the data, I proceeded with cleaning any unwanted information in the poem, specifically the 'copyright' text or symbols. In addition, the topic for some of the poems was manually marked if it was not included already in the dataset. However, this procedure proved to be very timeconsuming, and in the end, I decided to skip it and proceed exclusively with the topics dataset, leaving the forms and the merged datasets for some future endeavor. Lastly, I split my dataset into train and testing.

The model I used was the GPT2-head-model, and it was sourced from the 🤗 Hugging Face library while PyTorch was used for fine-tuning it. The primary goal here was to train the model on the topics dataset in order to get it acquainted with the different themes and the topic-specific language. Ultimately, poems would be generated based on a given theme word and manually evaluated for their coherence and their relevance to the topic.

Originally, the evaluation of the generated poems was therefore decided to be manual. This way I would focus on evaluating the poems according to their relevance of the topic word, their coherence, and their creativity on a scale from 1 to 5, with 5 being the highest. Secondly, the project was initially intended to summon automated metrics as well, and specifically perplexity. The plan was to divide the dataset into training and testing subsets, and construct probability dictionaries for the words that are most likely to appear in poems with the target themes. Subsequently, perplexity calculations would be applied to the generated poems using these topic-specific dictionaries. Otherwise, I would compare perplexity scores between the generated poem and a poem from the testing data in the same topic. However, due to time constraints, but also because of the fine-tuned model's generated poems, I decided to not use any perplexity measures and stick to the manual evaluation for the scope of this project.

Lastly, this methodology intended to include a comparative analysis between the outputs of the fine-tuned model and the original, raw model. This comparison was expected to provide insights into the extent to which fine-tuning and retraining enhances the model's ability to perform well on downstream tasks, in this case generating coherent and topic-relevant poems. A reasonable expectation would be that the fine-tuned model would produce more poem-like and ideally more semantically complex, in other words more creative text.

4. Results

Firstly, I generated text using the original raw model. A comparison with the fine-tuned one would provide valuable insights into the models' behavior and the impact of fine-tuning. As anticipated, the text generated here was just a continuation of the prompt topic word, which sometimes was nonsensical such as ['butterfly" with $1,000 to spare. A second option was a two-year, $50,000 grant.\n\nBut'], or ['butterfly-1.jpg","large_image_url":"https:\\/\\/s3.amazonaws.com\\/brewerydbapi'], which does not resemble a conventional piece of prose. In cases where the content made sense, it looked something like ["butterfly-like creatures. But in reality, they're really the polar opposite of their real species... not quite a species, though. ("]. Based on these outputs, there were hopes that the retrained model would do at least somewhat better in generating truly relevant text to the prompt word in a poem-like form.

However, the resulting generated poems of the fine-tuned GPT2 model did not resemble anything of the expected outcomes, let alone actual poems. Instead of generating original poems, the model reproduced the prompt, which contained the topic word, without adding any creative or meaningful expansion. To address this issue, I tried decreasing the batch size to 5 (which I think is still quite small), increasing the number of epochs to 40 during training as well as the length of the poems that are allowed to exist in the dataset to 512 instead of 300, so that there would be more examples. However, the latter resulted in Value Errors when training, and ultimately neither of the approaches mentioned seemed to help in any significant way.

Based on these results, it is quite obvious that the perplexity measures were not really needed, and that the manual evaluation of the generated output can be something like this: relevance to the topic word: 5 (since it is actually the topic word), coherence: 5 (since there is no other context), and creativity: 1 (this one is more obvious). But, in a more serious manner, the generated outputs are by far not matching the expectations set in the beginning of this project, and in reality, the previous manual evaluation should not be taken into consideration at all. What is important is looking into the reasons why this kind of output was generated.

5. Discussion

Looking at the previous results we can make some assumptions as to why the output of the model was like this. As mentioned earlier, the dataset used for retraining was not very big after truncation in order to fix the CUDA errors. However, it is not certain that this was the reason why the model behaved like this. My prediction would be that even if the data were smaller and did not teach the model much, the model would still generate some kind of expansion to the prompt word even if it was not poem-like or very relevant or creative.

In order to further investigate the reason behind such an output, it would be beneficial to look into and probably augment the data by randomly replacing words with one of their synonyms. Combining lines from different poems would not suit this project very well, because it would be important to maintain the semantic coherence. In this sense, randomly merging poems cannot guarantee that. Moreover, I would try different ways of training my model. For example, I created a custom training loop but maybe it would be more beneficial to use the Trainer class from the transformers library. This would also allow me to save checkpoints of my model and investigate closer what it is learning at each step of the process. In addition, trying different decoding methods, such as beam search, could shed more light on the generation process as there is a chance of them having different results in the output.

6. Conclusion

To summarize, this project aimed at the exploration of language generation by transformers and more specifically poem generation. It included data collection and preprocessing in order to fine tune the GPT2 head model to generate poems that would showcase characteristics like creativity, semantic coherence, and poem-like form. It was expected that not all these goals would be achieved completely, but the results were even more disappointing, since the model not only did not learn but also could not generate any significant output other than the prompt word. Looking into the reasons why this happened would have a lot of interest and as mentioned previously some steps towards that direction are already defined. I hope that in the future I will have some time to look into it and experiment more with language generation outside of any course's scope.

References

B. Bena & J.K. Kalita. (2020). Introducing Aspects of Creativity in Automatic Poetry Generation. ArXiv, abs/2002.02511.

Micelli, G. Haiku Generation. A Transformer Based Approach With Lots Of Control

Wu, X., Momo, K., Kazushige, I., & Zhan, C. (2017). Haiku Generation Using Deep Neural Networks.