Dmitriy Parshin

May 2022

Abstract

The aim of the project is a simple introduction to GPT and its advanced training technology. With the subsequent generation of text by the trained model.

Link to my project https://github.com/The-Illusive-Man-2000/NLP-course.-Spring-21-22.-Project.

1. Introduction.
   A few months ago, I read an article about how GPT was retrained by correspondence of a girl who had previously died. This article hooked me and I decided to get acquainted with this technology.

1.1 Team.

Dmitriy Parshin searched for information, collected a small dataset and retrained the model.

2. Related Work.

Previously, text was generated using recurrent neural networks. After the advent of the attention mechanism, text generation began to be done using transformers. A whole zoo of different models appeared.

3. Model Description.

Having heard about the power of GPT, I decided to start my acquaintance with text generation with GPT. Through the hugging face library. Specifically, I used a small Russian-language ruGPT3 model from Sberbank. Small, because colab did not pull large and medium versions of the model.

4. Dataset.

Initially, I wanted to use correspondence to retrain the model. But using correspondence for a public project was unethical. So I decided to use poetry

texts. At first I used Pushkin's poetry, but then I decided to focus on Mayakovsky's poetry, as he had a more expressive and recognizable style. Let me emphasize right away that I am not a fan of Mayakovsky in particular and of poetry in general. I chose 6 poems by Mayakovsky among the most popular.

Poems taken from here:
https://rustih.ru/populyarnye-stixi-pushkina/
https://rustih.ru/populyarnye-stixi-mayakovskogo/

5. Experiments.

5.1 Metrics.

I read about the bertscore and BLEURT metrics, but judging by the articles I read, they required that, in addition to the generated text, there should also be text with which to compare. Those, these metrics are more suitable for the tasks of translating text from one language to another or paraphrasing text. I had the task of generating text from scratch, so the quality of the resulting text had to be assessed by eye.

5.2 Experiment Setup.

At first I tried the large and medium version of ruGPT3, but free colab did not pull these models. I had to stop at a small version of the model. Experimentally, I noticed that with a large number of epochs, more than 200-300, the model, instead of adopting the style, began to remember entire fragments of the text on which it was trained. When the number of epochs is less than 100, the model generated text poorly. Therefore, I decided to stop at a smaller number of epochs 100-200. Also, when generating text with an already trained model, I experimented a bit with parameters such as beam width, temperature, top-k and top-p.

5.3 Baselines.

3 years ago I tried to teach LSTM on Pushkin's poems, but then I didn't succeed. And who now needs LSTMs for such tasks when there are GPT and Hugging Face?

6. Results.

Some examples of the text that the model generated after learning on Mayakovsky's poems. The line that the model tried to continue «В лесу родилась ёлочка»:

```
text = "В лесу родилась ёлочка\n"
input_ids = tokenizer.encode(text, return_tensors="pt").to(DEVICE)
model2.eval()
with torch.no_grad():
    #out = model2.generate(input_ids, do_sample=True, num_beams=2, temperature=1.5, top_p=0.9, max_length=80)
    out = model2.generate(input_ids, do_sample=True, top_k=5, max_length=100)

generated_text = list(map(tokenizer.decode, out))[0]
print()
print(generated_text)
```

```
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.

В лесу родилась ёлочка
и в лесу она…
Знай: сначала
ты
плохо играешь,
потом -
очень хорошо платишь.
Если
ты вдруг заплачешь -
плохо не будет
тебе.
Если
катишься по улице
и вдруг
октябрята закричат,-
катись скорей
и не плачь.
Если
катишься ты по городу,-
катись скорей
и
не зво
```

+ Код + Текст

```
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.

В лесу родилась ёлочка
и в лесу
поняла
мою
любимую
работу.
Я — гордый орел
и смелый пацан.
Скажи, кто твой
ответ
на этот
вечный
язык
тех,
кто
жив
и дышит
ветром
из преисподней?

Мальчик
радостный
шел
с работы,
а
за ним
швед
и двое
ватников
```

7. Conclusion.
   I collected a small dataset of Mayakovsky's poems and trained ruGPT3 on it. Experimented with epochs and generation parameters (beam width, temperature, etc.). Of course, there is no rhyme in the texts received, and the meaning is schizophrenia. But in my opinion, the pre-trained model was able to capture Mayakovsky's style, his sharp, torn, short sentences. I suspect that if it were possible to upgrade the medium or large version of the model, then the result would be better. However, I set out to touch this technology. And I achieved my main goal. I touched the technology and experimented a little.

References

https://habr.com/ru/news/t/576952/

https://www.theregister.com/2021/09/08/project_december_openai_gpt_3/

https://huggingface.co/blog/how-to-generate

https://huggingface.co/docs/transformers/main_classes/trainer

https://neurohive.io/ru/novosti/bleurt-metrika-dlya-ocenki-modelej-dlya-generacii-teksta/

https://spb.hse.ru/mirror/pubs/share/480745430.pdf

https://colab.research.google.com/drive/1kpL8Y_AnUUiCxFjhxSrxCsc6-sDMNb_Q

https://paperswithcode.com/paper/bertscore-evaluating-text-generation-with

https://github.com/huggingface/datasets/blob/master/metrics/bertscore/bertscore.py

https://towardsdatascience.com/decoding-strategies-that-you-need-to-know-for-response-generation-ba95ee0faadc

https://colab.research.google.com/github/philschmid/fine-tune-GPT-2/blob/master/Fine_tune_a_non_English_GPT_2_Model_with_Huggingface.ipynb