Poetry generation with GPT-2

Week 4 project by Erik Jenner and Alexandra Souly

MLAB 2022 Summer

The main goal of our project was to fine-tune GPT-2 small to generate rhyming poetry, based on a starting line given by the user. We have tried a few different approaches: fine-tuning with self-supervised learning using a rhyming poems dataset, rejection sampling to force rhymes, and using RL with a reward function that encourages rhymes and a stanza format.

We have found the combination of self-supervised learning and rejection sampling to be successful, while we haven't successfully improved on the model using RL. We have also created an interactive web interface that allows the user to generate poetry using our models.

Setup:

We have built a rhyme detector that checks if two words rhyme based on the words' pronunciation. We took two words to rhyme when their last vowels are pronounced the same.

As our dataset for fine-tuning, we preprocessed a Kaggle poems dataset, from which we have filtered out all non-English text. We have split them into 4 lines stanzas, and only kept the ones that had 2 pairs of rhymes, resulting in about 16000 stanzas to train with.

Self-supervised finetuning:

Self-supervised fine-tuning worked well without a hyper-parameter sweep, the model took only about 20 minutes to improve significantly. It reliably learned to output at least 4-line stanzas with linebreaks, used on-theme words, and mimiced prompt length in its lines. It was slightly better at rhyming as well, it produced rhyming lines with a 4% probability, while the base rate of GPT-2 is 1%. However, it did not learn to output EOS tokens after 4 lines despite our dataset only consisting of 4-line stanzas, so we automatically truncate the model outputs after 4 lines.

Rejection sampling:

To rhyme more reliably, we force rhymes in an 'aabb' pattern while sampling our stanza rhyme by rhyme using rejection sampling. We find a line that rhymes in 25 tries on average, but if no rhymes are found within a reasonable number of attempts, we just keep the most probable next line.

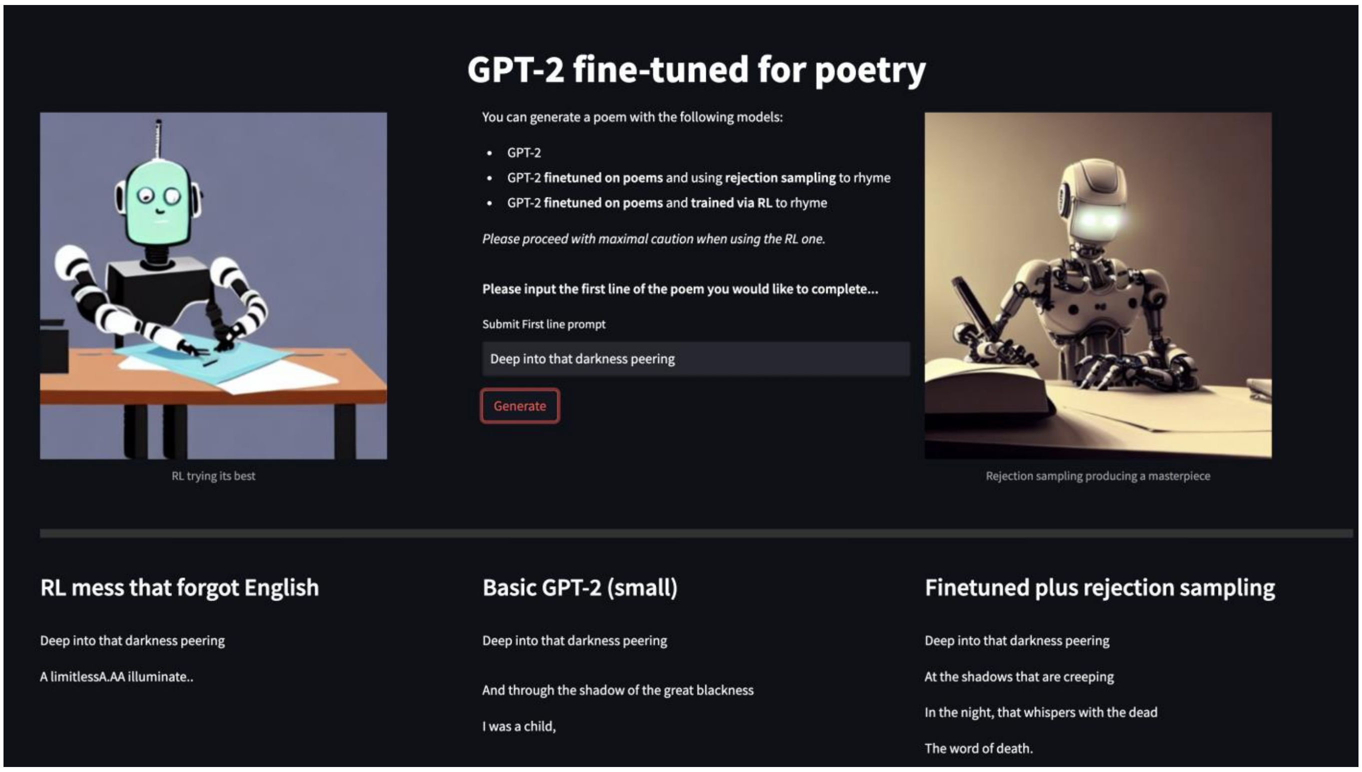
We have found this method to work well and produce acceptable poems that rhyme.

Reinforcement learning:

We have tried further fine-tuning our pretrained model with PPO using the trl library. We created a reward function that rewards rhymes and at least 4 lines in the output, hoping that PPO's KL divergence would keep the language capabilities of GPT-2. We did not find suitable hyperparameters for this, and after 10 hours of training our models had phase changes where they forgot how to speak in English and started outputting random text.

We have found that the short time frame made working with RL very difficult due to the long training times. Given more time, we think the RL approach could be fruitful and produce results similar to rejection sampling, but much faster. The RL fine-tuning seemed not very robust to different hyper-parameters, sweeping for good ones could help the model not forget English. We have also used a hand-crafted reward function with arbitrary coefficients, refining that or using actual human feedback would be helpful. Fine-tuning a bigger GPT-2 model night be a good approach as well.

Interface:
We have created a web interface for poem generation using Streamlit, which allows a user to generate a poem from a prompt using GPT-2, our self-supervised fine-tuned rejection sampling model, and our RL model. Only the second model produces real poetry, but the other two provide entertaining comparisons. See https://github.com/alexandrasouly/rhyme_finetuning on trying it out!



GPT-2 fine-tuned for poetry

You can generate a poem with the following models:

- GPT-2
- GPT-2 **finetuned on poems** and using **rejection sampling** to rhyme
- GPT-2 **finetuned on poems** and **trained via RL** to rhyme

Please proceed with maximal caution when using the RL one.

Please input the first line of the poem you would like to complete...

Submit First line prompt

Deep into that darkness peering

RL mess that forgot English

Deep into that darkness peering

A limitlessA.AA illuminate..

Basic GPT-2 (small)

Deep into that darkness peering

And through the shadow of the great blackness

I was a child,

Finetuned plus rejection sampling

Deep into that darkness peering

At the shadows that are creeping

In the night, that whispers with the dead

The word of death.

Screenshot of our user interface