CS505 Project Report: AI Poetica

Junyi Li, Tianchi Huang

link to repo:
https://github.com/JaydenLi0206/AI-Poetica.git

Abstract

This report introduces our AI Poetica, a GPT-2 based model fine-tuned on a curated corpus of 20th-century poetry to autonomously generate poems. The model's novelty lies in its dataset—a collection of profound poetic works, enabling the generation of stylistically diverse and coherent poetry. To assess the model's efficacy, perplexity scores and Flesch-Kincaid Readability Tests were employed, revealing high variability indicative of creative depth. Future work includes experimenting with more advanced models and expanding the dataset to refine quality and creativity in poem generation.

Introduction

Poetry stands as a revered channel of human expression, melding the intricacies of language with the depth of emotion and imagination. In this age of digital transformation, the intersection of artificial intelligence and literature has introduced intriguing possibilities in the realm of creative writing. Our project, AI Poetica, harnesses the power of Natural Language Processing (NLP) to craft poetry, blending the timeless art form with cutting-edge technology. We have embarked on an ambitious journey to develop a model based on GPT-2, one of the most advanced language processing algorithms, to generate poems that resonate with both style and substance.

Our venture is characterized by the assembly of a unique corpus that encompasses a selection of the most esteemed poetic compositions of the 20th century. This compilation is not merely a dataset; it is a tapestry of literary masterpieces, carefully woven to reflect a spectrum of styles and themes that define an era of poetic richness. By fine-tuning GPT-2 on this meticulously curated corpus, we aim to capture the essence of poetry's structural elegance and thematic resonance. We have tailored our model to recognize and emulate the nuances that give poetry its rhythmic and evocative power. This report details the conscientious process from initial dataset preparation to the intricate fine-tuning of the model, setting the stage for AI Poetica to generate poetry that is coherent, stylistically faithful, and thematically compelling.

Methods Process

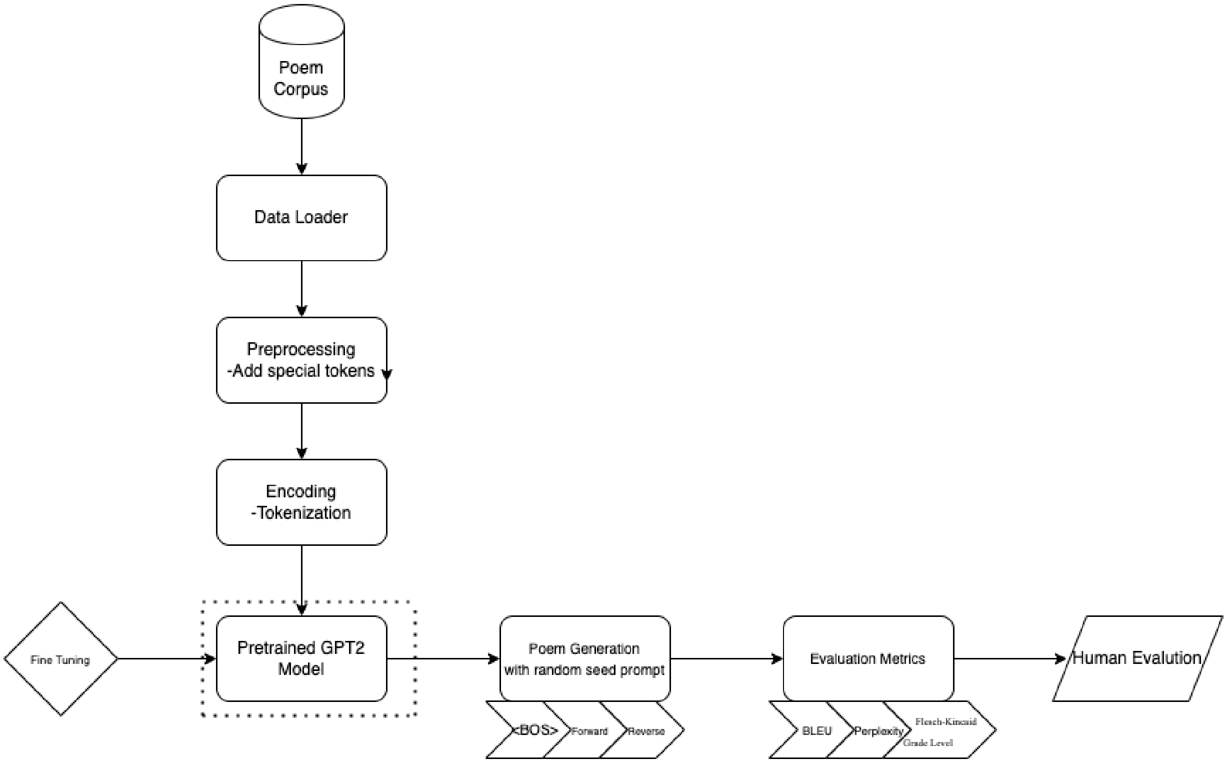

Figure 1: Poem Generation Process for AI Poetica

Dataset Preparation

The dataset utilized for this project is a manually selected collection of poems scraped from Project Gutenberg, featuring works by prominent 20th-century poets such as T.S. Eliot, Cyril Scott, Ezra Pound, and others. Each record in the dataset is comprised of two primary fields: 'title', which indicates the name of the poem or the segment, and 'stanza_text', which contains the actual text of the poem or excerpt. The texts are rich with imagery and complex themes, reflecting the diverse styles and profound expressions characteristic of early to mid-20th-century poetry. This dataset serves as a valuable resource for literary analysis, natural language processing tasks, or anyone interested in the evolution and depth of modern poetry.

We created a class for a custom dataset designed for a PyTorch model, which takes poem stanzas as input and preprocesses them for training or inference with a GPT-2 model. It tokenizes the input text with special beginning and end tokens, truncates or pads them to a maximum length, and creates attention masks for the model, providing a standardized way to load and handle poem stanza data for training.

Data Loader

The data loading portion of the code initializes two `DataLoader` objects for the training and validation datasets of poem stanzas, respectively. The training data loader uses a

`RandomSampler` to shuffle and batch the data, enhancing the model's generalizability, while the validation data loader employs a `SequentialSampler` to iterate through the data in order, both with a specified batch size which is 3 to determine the number of samples processed in each training step.

Encoding (Tokenization)

In the tokenization process, a GPT-2 tokenizer is initialized from a pre-trained 'gpt2' model to ensure that the structure and vocabulary of the original model are preserved. The tokenizer is then equipped with three special tokens: BOS> for the beginning of a sentence, EOS> for the end of a sentence, and PAD> for padding shorter sequences. A poem's stanzas need to be indicated by these tokens in order to maintain consistency in sequence lengths during training. In training and generation, the number of new tokens added to the tokenizer allows the model to understand and interpret these newly introduced special symbols.

Model Selection

For this project, we chose to use a powerful language model GPT-2 which is a transformer-based language model developed by OpenAI in 2019. At its core, GPT-2 is an unsupervised transformer language model. Language models are designed to understand and predict text sequences, leveraging probability distributions to forecast the subsequent word in a sequence. They discern and utilize the contextual relationships between words to enhance text predictions. By fine-tuning it on our poem dataset, it can achieve the goal of generating poems for both conditional and unconditional generation.

Model Training

Fine tuning GPT-2 involves iteratively updating the model parameters to minimize the loss function. The process is iterative and typically relies on gradient descent. Here is a condensed version of what a training cycle looks like:
Initialization with pre-trained parameters; for each iteration(epoch), involving forward pass, calculate loss, backward pass, and then update parameters; then repeat until convergence.
We use Adam optimizer to update model weights during training which is a prevailing adaptive gradient descent algorithm. Also, we define key hyperparameters: learning_rate, which controls the step size in optimization; eps, a small number to maintain numerical stability; and warmup_steps, which gradually increases the learning rate at the start of training to stabilize early iterations. Next, it specifies that computations should be performed on a GPU for efficiency. Finally, it prepares the initial input for text generation by encoding a seed prompt ("<BOS>" for beginning of sentence) into a tensor and moving it to the GPU, setting the stage for the model to generate text or continue with fine-tuning.
After 80 epochs, the trained model is saved for generating new poem stanzas.

Poem Generation

Setting the model to evaluation mode indicates it is not in the training process anymore but in the use of generating new content. Then, with the use of the generate method, the model is going to create a new poem based on the initial seed text provided. This method employs a sampling strategy, where top_k and top_p parameters control the diversity and randomness of the generated text, ensuring a creative and varied output. Each generated sequence is then decoded from token IDs back into readable text, and the resulting poem stanzas are printed out, showcasing the model's ability to emulate the style and structure of the training data it was fine-tuned on.

Qualitative Results

Below we present some selected poems generated by our AI Poetica.

1. The Chair is dumbfounded, seeming to slumber; I hear Beside a Creolean lady talking In quaint and quaint tenements, Loud and proud, What music these lovely souls hear; and she, The dame of situations, plays alone The harp on the cell.
The air is heavy with viols, which throughout the whole of the Show are terrific; The viols, which throughout the whole of the Show are terrific, Lie on the banister-organ, some five feet in'd, Broad-shouldered elephants, which always cower Within the banister.
I say these things to myself, To luncheon guests, in the midst of their Being polite and to children;— At the violet hour, when all Is clench'd within the violet walls, All propitious smoke rises In the air, like an incense-wreathed rose; The lady's fingers are sensitized, The foaming mouth, like a lectern, is full Of memories profound.
I say these things to myself, As the evening torches fill My brain, which will burn up

2. The South-wind brings Life with it,— Life with a crust, A clinging mist on its brow.
While Time warms, And the South-wind makes its voice.
The Music swells, And we are sucked in.
The South-wind calls,— "Bard!"— "Courage! come this way!" Our hearts are at their height.
The South-wind makes his art.

3. Asleep, stung by visions, Till the ebbing sea grants one the terror of a lost night; Till, in the calm of some forgotten nest, One poet laments alone The shaft of sleeping Jove's wings, Far from the fiery noon....
I cherish thee thus!
O starless night!
I cherish thee thus!
O starless night!
Stay where!
I cherish thee so!
As I adore thee, Beloved of the Infinite!
To win, to parley With the wintry rabble; To hold, and win, and sink or swim With the tide of history, And at last be man, Earth-bitten!
And live and let live!

4. I bring all I am, certitude, Of power that bring me forth, And make known all I am to say.
To all men at heart, kindred tongue and eye, I say, I am the Avenger, And I am the dawn.
Of old I have stood athwart the lash of the North Star With inflated frenzy, And I will not slumber, Nor whine and moan, Nor whine and smile.
When each with his momentous art Outshines mine, I sway With the will of the Sprite as through endless swoons, That I give them speech and joy.
There were but one way to me,——Speak, and I am done.

5. I ask of thee, O Beauty!
To know thy secret, And dive deep in thy depths!
While a star glows at my back, And deep in thy spirit I desire Thee.
Thy form is vain, And thy beauty is shallow.
Thou cannot hide thy sorrows, Nor weep at thy brow, Because I have a prize in my hand, Nor sit by and brood over thy griefs.
Thou delight'st to garnish With gems of wine my whole, And, while my soul prays for thee, I must reign alone.
Thou delight'st to garnish With gems of gold my whole, And when my eyes are full of tears, I weep alone.
Oh, miserable me!
Stoop! and enfold!
Thou delight'st to garnish With gems of gold my whole!

Evaluation Metrics

After generating poems, we have identified key metrics to evaluate our results effectively. The first metric is the calculation of perplexity, a standard measure in our previous assignments. In this approach, we assign a probability of 0.01 to words not found in our poem corpus, ensuring a more realistic assessment of the language model's performance. This method helps gauge how well the model predicts or understands the structure of the generated poems.

The second set of metrics we've incorporated is the Flesch-Kincaid Readability Tests, a well-regarded tool for assessing the readability of English texts. These tests consist of two components:

Flesch Reading-Ease: This test evaluates the text on a 100-point scale, where higher scores indicate greater ease of reading. Texts with simple, short sentences, and familiar words tend to score higher, reflecting their accessibility to a broader audience.

Flesch-Kincaid Grade Level: This aspect of the tests assigns a U.S. school grade level to the text, suggesting the educational level required for comprehension. For example, a score of 8 implies that the text is suitable for eighth-grade readers. The test takes into account sentence length and word complexity, providing insight into the text's complexity.

Employing both the perplexity measure and the Flesch-Kincaid Readability Tests allows us to evaluate our generated poems from different perspectives. While perplexity focuses on the predictability and linguistic consistency of the poems, the Flesch-Kincaid tests offer an understanding of their readability and accessibility. These tests are instrumental in educational and publishing contexts, ensuring that written materials align with the intended audience's comprehension abilities. Together, these metrics provide a comprehensive framework for assessing the quality and effectiveness of our poem generation model.

By generating 50 sentences and calculating the perplexity, we get the graph in Figure 2.

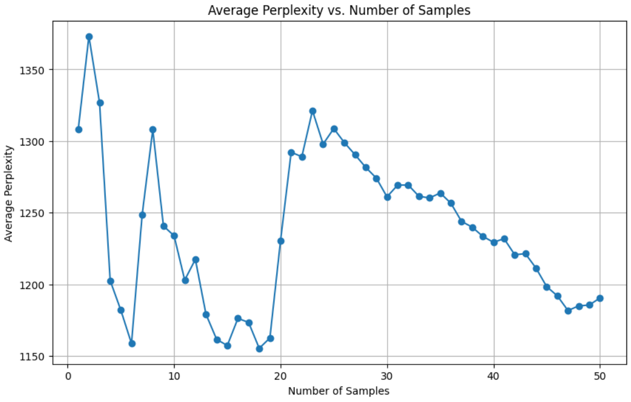

Figure 2: Trend of Average Perplexity Across Generated Poems

A perplexity of around 1200 is relatively high, indicating that the model found the text to be unpredictable or complex, possibly due to the creative and non-standard use of language often found in poetry.

By generating 80 sentences and calculating the scores based on the two tests in the Flesch-Kincaid Readability Test, we get the graph in Figure 3.

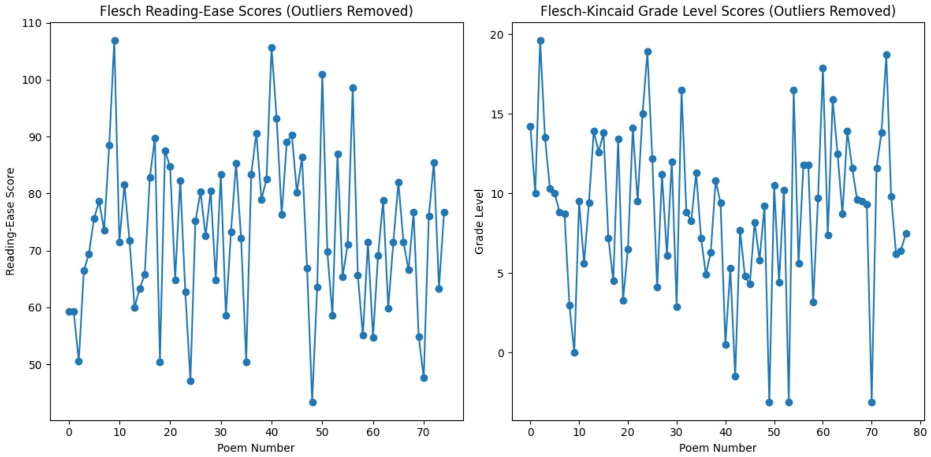

Figure 3: Flesch Reading-Ease and Flesch-Kincaid Grade Level scores graphs

In short, the graphs display a high degree of variability in the readability of the generated poems, which could be indicative of diverse styles and vocabularies being used by the poem generation model. While higher variability in scores could suggest creativity and a wide range of expression, it may also point to inconsistency in the model's output, especially if there is no intentional control over the complexity of the generated poems. If the goal is to maintain a certain readability level, these results could indicate the need to fine-tune the model to produce more consistently targeted outputs.

For human evaluation, we ask for some opinions from our friends since they are the easiest to approach. To conclude their opinions, overall, these poems are rich in language and emotion, each creating a distinct world. Despite their impressive linguistic flair, their effectiveness varies, with some parts being more impactful and coherent than others. Although the poems' vivid imagery and emotional depth are commendable, they might be undermined by occasional obscurity or repetition.
That is to say, we can define our model’s performance to be successful.

Discussion

At all, we believe our model is doing a great job in generating poems. It obviously learned the pattern and format of how to write a poem. Although the evaluation score may not looks good, that could be because poetry often relies on creative structure, metaphor, and rhythm, which are not always captured in the text generation where those metrics evaluate. That is to say, even with an extremely low evaluation score, the poem itself still can be qualitatively good. Also, unlike informative texts, poetry is highly subjective. What one person considers a masterpiece, another might not understand or appreciate. Therefore, the evaluation of poetry can take human evaluation into account where multiple humans could judge the quality of poetry.

To clarify, we only use the smallest version of GPT-2 which is less than a tenth of the parameters in the available XL version of it. However, it still displays a really powerful ability to generate text which shows plenty of room for improvement.

Future Work

For future work, building on the promising results of our GPT-2-based AI Poetica, we propose to experiment with larger-scale models, such as the more parameter-dense GPT-2 XL version, to potentially enhance the quality and depth of the generated poetry. The larger scale version has more powerful capabilities. The increased capacity could better capture the nuances and complexities inherent in poetic language, offering a more refined understanding and generation of poetic structure, metaphor, and rhythm. Additionally, we plan to explore more advanced training techniques, data augmentation strategies, and the integration of multimodal inputs to enrich the creative process. Also, we can expand the size of the dataset to include more poem samples for the model to better capture the writing skills.

This expansion, coupled with more sophisticated evaluation metrics and continued human evaluation, will drive our AI Poetica toward a new horizon of artificial creativity, pushing the boundaries of what's possible in the automated generation of poetic art.

Conclusion

We present AI Poetica, an automated poem generator incorporating a pre-trained GPT-2 model and our custom poetry corpus, displaying the ability to generate well-formatted poems with rich content. We also explore and define metrics to evaluate the results such as perplexity, Flesch-Kincaid Readability Tests, and human evaluations. Finally, we generate 100 poems and store them into one file for fun and pass them through to our friends for entertainment.

Personal Contribution

Tianchi Huang: Model Evaluation, constructing report, making visualizations, construct dataset, construct model

Junyi Li: Model training, Data preprocessing, Poem generation, constructing report, constructing dataset