On the power of special-purpose GPT models to create and evaluate new poetry in old styles (DRAFT VERSION)

Piotr Sawicki[1], Marek Grześ[1], Fabricio Goes[3], Dan Brown[2], Max Peeperkorn[1], Aisha Khatun[2], Simona Paraskevopoulou

[1] School of Computing, University of Kent, Canterbury, UK
[2] Cheriton School of Computer Science, University of Waterloo, Canada
[3] Computing and Mathematical Sciences Department, University of Leicester, UK
p.sawicki@kent.ac.uk, m.grzes@kent.ac.uk, fabricio.goes@leicester.ac.uk, dan.brown@uwaterloo.ca,
m.peeperkorn@kent.ac.uk, aisha.khatun@uwaterloo.ca, simonaparaskevopoulou@gmail.com

Abstract

This study investigates the possibility of using GPT-3 models to generate high-quality poems in a specific author's style, through fine-tuning on datasets of poems accompanied by their metadata and automatically generated summaries. Our experiments show that a dataset of only 300 poems is sufficient to generate new poems in the style of a specific author. The evaluation was done through GPT-3 models fine-tuned for binary classification of GPT-3 outputs against the works of the original author. To establish the accuracy of GPT-3-based binary classifiers, we first tested them on a variety of texts and a range of classes, and found that their predictive accuracy is 99% on average. Using this method for poetry evaluation showed that the GPT-3 generated poems were indistinguishable from the original works of Walt Whitman and Rudyard Kipling in an average of 30% and 21% of the cases, respectively. This suggests that GPT-3 can be a useful tool in assisting authors, while further research is needed to turn it into an independent creator. Additionally, the workflow used in this study can be applied to other types of text and provides a way of using GPT-3 models for generating new content from user-provided summaries, when prompt engineering alone is insufficient.

Introduction

With the emergence of Large Language Models (LLMs), there has been tremendous growth, not only in Natural Language Processing (NLP) but also in Computational Creativity (CC). In particular, the GPT-series (Radford et al. 2018; 2019; Brown et al. 2020) is the main contributor to the progress. LLMs have an astonishing capacity for capturing and mimicking features from massive amounts of data. Although LLMs have attracted expected criticism (Birhane and Raji 2022; van Dis et al. 2023), e.g., with respect to stylistic reproduction (Floridi and Chiriatti 2020; Falk 2021), their reception has overall been positive (Brown and Jordanous 2022). Their remarkable generative capabilities warrant further exploration in Computational Creativity research (Dale 2021; Köbis and Mossink 2021).

Poetry creation, as a CC task, has been explored over the years using a wide variety of techniques (Lamb, Brown, and Clarke 2017; Oliveira 2017). There are expert systems (Misztal and Indurkhya 2014; Corneli et al. 2015), constraints-based approaches (Rashel and Manurung 2014; Toivanen et al. 2013), and linguistic models (Veale 2013; Hämäläinen 2018) that can imitate styles and produce novel poems. Moreover, machine learning techniques (Toivanen et al. 2014; Lamb and Brown 2019) and evolutionary approaches (Rahman and Manurung 2011) have achieved some success. Text-generation techniques have also been applied beyond poetry generation (Pachet and Roy 2014; Ens and Pasquier 2018) and can assist in the development of techniques to deliberately deviate from learned styles (Elgammal et al. 2017).

Large Language Models can generate high-quality texts, paragraphs, and short creative artifacts, such as poems or lyrics. The current applicability of LLMs goes beyond autonomous generation of novel artefacts, and practitioners use them in co-creative ways to explore, get inspired, or as a tool to overcome writer's block (Gwern Branwen 2019; 2022). Regardless, many creative tasks still require human moderation to filter out nonsensical responses and subpar results. To improve the quality of creative output of transformer-based systems, we need to explore what is possible, understand the challenges involved, and devise computer-based methods for verifying if the system is performing well. Likewise, it is crucial to determine differences in performance and associated costs between the various sizes and architectures of LLMs, allowing us to make informed decisions on model selection for the creative task at hand. In this paper, we present a preliminary exploration of these challenges, and offer current state-of-the-art recommendations.

As NLP research increasingly focuses on transformer-based approaches, computational creativity is starting to follow suit. Notable examples of using GPT-2 or BERT for poetry generation include fine-tuning GPT-2 for Chinese classical poetry (Liao et al. 2019), conducting an extensive human evaluation of GPT-2 generated English poetry (Köbis and Mossink 2021), experimenting with rigid constraints in poetry generation in both Chinese and English (Li et al. 2020), analysing the challenges of maintaining rigid stylistic constraints while using RNN and GPT-2 (Wöckener et al. 2021), exploring a transformative BERT-based approach to lyrics generation (Nikolov et al. 2020; Oliveira 2021), and generating lyrics from GPT-2 and evaluating with BERT (Wesek 2019). Hämäläinen *et al.* (2022)

experimented with combined encoder-decoder setup using RoBERTa and GPT-2 for modern French poetry generation. The methodology of human-computer co-creation of poetry have been explored in (Boggia et al. 2022), while (Stevenson et al. 2022) attempted to evaluate the creative abilities of GPT-3. Fine-tuning GPT-2 for poetry generation in the style of Emily Dickinson was explored in (Dai 2021). In (Lo, Ariss, and Kurz 2022) the authors have fine-tuned GPT-2 for limerick generation with special attention to maintaining the limerick rhyming scheme. (Chakrabarty, Padmakumar, and He 2022) worked on fine-tuning T0 and T5 LLMs for collaborative poetry generation.

In (Bons 2022) the author experimented first with generating song lyrics using prompt engineering with GPT-3, and subsequently with fine-tuning GPT-3 on a dataset of songs accompanied by songs' descriptions, artist biographies and song titles. The fine-tuning process allowed the author to generate higher quality lyrics than using prompt engineering only.

A similar approach from outside the field of computational creativity is the work of (Lee 2019; Lee and Hsiang 2020b; 2020a) who fine-tuned GPT-2 and BERT models for patent claim generation and evaluation. The authors fine-tuned GPT-2 on a dataset consisting of US patent claims, where each claim is accompanied by its summary and title. The system was subsequently able to generate patent claims from summaries provided by the user.

What those two works, song lyrics generation and patent claim generation, have in common is fine-tuning the models on the datasets where each entry is accompanied by its summary and other metadata. This allows the user to control the content of the output through a summary and other metadata provided in the prompt.

GPT-3 is capable of generating poetry through prompt engineering alone. It can generate poems that are not only grammatically correct and have appropriate structure, but also tell a coherent story and can appear meaningful and evocative (Gwern Branwen 2022). However, the poems generated through prompt engineering alone, always appear to be written in the same style and use plain and simple language that lacks the unique personal perspective and emotional nuance that are hallmarks of human-generated poetry. Our initial experiments have shown that prompting GPT-3 Davinci (`text-davinci-003` version, which is the recommended version for text generation) to generate poems in the style of a specific author, e.g. Walt Whitman, does not lead to the desired outcome. A well-structured poem is generated, and the narrative requested in the prompt is followed, but the style in an obvious way does not match the style of the requested author. One can assume that the works of all classical authors were part of the GPT-3 training dataset, but the style of a specific author cannot be reliably invoked through prompts because the GPT-3 model aggregates too many styles.

Objectives and Methods

Our long-term objective is to build a system which can generate poems in the style of a specific author and with the subject and narrative provided by the user, thus allowing the user maximum control over the outcome. We fine-tune GPT-3 models on datasets of poems accompanied by their summaries and other metadata. We show that when GPT-3 is fine-tuned on the poetry of poet A, it will produce outputs in A's style even if the summary will request topics/content that the poet A has never written about. For example, we obtain poems written in the style of poet A about topics or content that appeared in the works of poet B.

Our second objective is to show that GPT-3 can also evaluate the correctness of style. We use GPT-3 to evaluate generated poetry. This automated approach is motivated by the methodology presented in our previous work (Sawicki et al. 2022), where we have fine-tuned BERT models for binary classification of fragments from the works of an original author (Byron and Shelley in their case), against samples produced from GPT-2 models fine-tuned on the works of that author. The idea is that if the classifier cannot distinguish between those two categories, (i.e. the accuracy of the classifiers is around 50%), then the text has been successfully generated with high quality in the desired style.

This way of evaluation resembles the GAN argument: the produced item is regarded as "good" when the classifier cannot distinguish it from the set of items used to train the generator (Goodfellow et al. 2020). This approach, however, comes with a caveat: it can be argued that when the evaluation results are approaching 50%, instead of indicating the high quality of the generated poems, it may simply mean that the classifier is of poor quality. For that reason, we conduct a number of experiments to establish that GPT-3-based classifiers can succeed on this task. We classify using fine-tuned GPT-3 models instead of BERT (which was the classifier used in our previous work (Sawicki et al. 2022)), because BERT requires large data sets to achieve good classification accuracy, and our poetry datasets are too small. We demonstrate that GPT-3-based binary classifiers achieve 99% accuracy when fine-tuned on only 200 samples per label.

The main contributions of this paper are:

1. We present a workflow that allows for generation of poems with a specific narrative and in a specific author's style through fine-tuning GPT-3 models.This approach could be extended to other categories of text than poetry, where prompt engineering alone does not give desired results.

2. We demonstrate that GPT-3 models fine-tuned for classification are highly accurate as text classifiers and can be used as a tool for poetry evaluation.

3. We provide a dataset of 2100 out-of-copyright poems (7 authors and 300 poems per author) where each poem has a summary and theme. This dataset can be used for further research on poetry generation.

4. We show new insights into the performance of various versions of GPT-3 models on poetry generation. The smallest models (Ada and Babbage) produce results comparable to larger models (Curie and Davinci), thus considerably reducing the costs of fine-tuning GPT-3 for poetry generation and evaluation. This indicates that some tasks,

like poetry creation, do not require the use of largest models.

The paper is organised as follows. Our dataset and the process of fine-tuning GPT-3 for poetry generation are presented in Part 1 on poetry generation. Poetry evaluation using GPT-3 as a classifier is the subject of Part 2 of the paper, where the results are also presented and analysed. The main findings of the paper are highlighted in Conclusion, where ideas for future work are also discussed.

Part 1—Poetry Generation

GPT-3 should not be thought of as a single system. It is available in four different sizes (Ada, Babbage, Curie and Davinci) and a multitude of fine-tuned versions (for example: `text-davinci-003`, `code-davinci-002`, `curie-instruct-beta` , etc.). Fine-tuning is only available for the vanilla versions of the four sizes:

- Ada (2.7 Billion parameters),
- Babbage (6.7 Billion parameters),
- Curie (13 Billion parameters),
- Davinci (175 Billion parameters).

In this work, we use all four sizes of GPT-3 models fine-tuned separately for poetry generation and evaluation. We also use `text-davinci-003` (one of the latest of Davinci models fine-tuned by OpenAI specifically for text generation) to create summaries of the existing poems.

While GPT-2 models can be fine-tuned on any text file, GPT-3 requires a fine-tuning dataset to be organized in a specific way, i.e., each entry must be in the form of:

```
{"prompt":"BODY_OF_PROMPT",
 "completion":"BODY_OF_COMPLETION"}
```

couplets. Therefore, GPT-3 cannot be fine-tuned on the dataset of poems alone. If the body of the poem is in the completion, we must decide what to put in the prompt.

While it is possible to fine-tune GPT-3 on a dataset where the prompt contains only the name of the author and the title of the poem, this does not give the user much influence over the narrative of the generated poem. As the body of the poem is the expected completion of the model, it is required that the instructions are provided to the model through the prompt by describing the narrative of the poem. Since this prompt is missing in the original dataset (and, in fact, in all publicly available datasets at the time of writing this paper), we use GPT-3 to create summaries of the poems' bodies and use them to control the narrative. This means that we first use `text-davinci-003` to generate summaries for our corpus of poems, and then the original poems and their summaries are used to fine-tune another instance of GPT-3 for poetry generation.

Data Preparation

To prepare our dataset, we scraped 2100 poems from publicly available sources (Project Gutenberg 2022; Poetry Foundation 2022). To lower the cost of running the experiments, we used only the poems that are more than 100 words and less than 500 words in length. This dataset contains the works of seven classical poets, and we selected 300 poems per author. These authors are:

- Ella Wheeler Wilcox (American, 1850–1919),
- Rudyard Kipling (English, 1865–1936),
- Emily Dickinson (American, 1830–1886),
- Lord Byron (English, 1788–1824),
- William Wordsworth (English, 1770–1850),
- Walt Whitman (American, 1819–1892),
- Thomas Hardy (English, 1840–1928).

We use only the works of authors who passed away more than 75 years ago due to copyright limitations. For all these poems, we generated summaries and main themes using `text-davinci-003`, and this process is explicated below.

Summary Generation For the generation of summaries and themes, we experimented with all available GPT-3 sizes (Ada, Babbage, Curie, and Davinci), and we found that only Davinci is capable of generating good quality summaries. The version of Davinci we used was `text-davinci-003`, which, at the time of writing this paper is the latest Davinci model dedicated to text generation.

Initially, each entry in the original dataset contains the following data: author, title, dates of author's birth and death (separated with a hyphen), author's country and finally, the body of the poem.

To generate the summary of the poem, we have used the following prompt:

```
"This is the poem:" +
BODY_OF_THE_POEM +
"This is the poem's summary:"
```

Theme Generation The rationale behind adding the main theme of the poem is to give an additional way of influencing the content of the generated poem. For example, we can provide a summary that describes a poem about love, and set the main theme as "Love". The same prompt could have the main theme set to "Sadness" thus affecting the poem's tone.

To generate the main theme of a poem (from the body of the poem), we have used the following prompt, which also includes the full list of themes that `text-davinci-003` was selecting from.

```
"These are the categories: Mysticism, Childhood,
God, Love, Life, Art, Poetry, Sadness, Despair,
Depression, Death, Religion, Nature, Beauty,
Aging, Desire, Travel, Dreams, Birth, War,
Failure, Immortality, Fantasy.
Choosing from these categories select one that
best describes this poem:" +
BODY_OF_THE_POEM
```

Poems Annotated with Summaries and Themes Each entry in our dataset is augmented with the main theme of the poem and the poem's summary. Thus, each entry in the final dataset has the following format:

```
<|startofauthor|>AUTHOR<|endofauthor|>
<|startofdates|>BORN - DIED<|endofdates|>
<|startofcountry|>COUNTRY<|endofcountry|>
<|startoftitle|>TITLE<|endoftitle|>
<|startofthemes|>THEME<|endofthemes|>
<|startofsummary|>
...
BODY OF THE SUMMARY
...
<|endofsummary|>
<|startofpoem|>
...
BODY OF THE POEM
...
<|endofpoem|>
```

The added tags are used to clearly delineate the specific items in each entry in the dataset. This is to enhance the fine-tuning process. These tags are used both during fine-tuning of the GPT-3 models and during the generation of the poems later on. Our complete dataset that includes the original poems, their metadata, summaries, themes and tags are available on our GitHub repository[1].

Fine-tuning GPT-3 for Poetry Generation

OpenAI documentation (OpenAI-Documentation 2023) suggests using a dataset with a minimum of 500 entries (i.e. poems) for fine-tuning. Our dataset has only 300 entries for each specific author. This limitation is common in poetry analysis because, in general, poets do not produce a high volume of work. For this reason, we consider two approaches to fine-tuning GPT-3 on our data:

1. Fine-tune individual GPT-3 models for each author. Here, every model is based on 300 samples.
2. Fine-tune GPT-3 models on a combined dataset of all seven authors. Here, every GPT-3 model is fine-tuned on 2100 poems of 7 poets.

Additionally, we examine which GPT-3 models produce the best results when fine-tuned on our poetry data. The general guideline from OpenAI is to fine-tune smaller models for more epochs, and larger models for fewer epochs (given a dataset of a fixed size). We fine-tune Ada and Babbage models for four epochs, and Curie and Davinci for one epoch and four epochs when using 300 samples. When fine-tuning the models on 2100 samples, we fine-tune all models for four epochs.

The cost of fine-tuning GPT-3 for poetry generation at the time of writing this paper are as follows:

1. Davinci 300 samples 1 epoch - $6
2. Davinci 300 samples 4 epochs - $24
3. Davinci 2100 samples 4 epochs - $169

Model	Acronym	Fine-tuning epochs
GPT-3-Ada	4e	4
GPT-3-Babbage	4e	4
GPT-3-Curie	1e	1
GPT-3-Curie	4e	4
GPT-3-Davinci	1e	1
GPT-3-Davinci	4e	4

Table 1: Fine-tuning GPT-3 models for every poet separately. This method uses 300 samples per model.

Model	Acronym	Fine-tuning epochs
GPT-3-Ada	7A 4e	4
GPT-3-Babbage	7A 4e	4
GPT-3-Curie	7A 4e	4
GPT-3-Davinci	7A 4e	4

Table 2: Fine-tuning GPT-3 models for all poets. This method uses 2100 samples per model.

The cost of using Ada, Babbage, and Curie models are respectively 50, 40 and 10 times lower than using Davinci (OpenAI-Pricing 2023).

The summary of our fine-tuning configurations is presented in Tables 1 and 2. Table 1 shows that we fine-tune 6 models for every poet considered, and Table 2 shows that we create 4 models using the combined dataset of 2100 poems of all poets. All the hyperparameters of the GPT-3 models are left at their default values.

The following prompt-completion tuple structure is used for fine-tuning our GPT-3 models:

```
PROMPT:

<|startofauthor|>AUTHOR<|endofauthor|>
<|startofdates|>DATES<|endofdates|>
<|startofcountry|>COUNTRY<|endofcountry|>
<|startoftitle|>TITLE<|endoftitle|>
<|startofthemes|>THEME<|endofthemes|>
<|startofsummary|>
...
BODY OF THE SUMMARY
...
<|endofsummary|>
<|startofpoem|>

COMPLETION:

...
BODY OF THE POEM
...
<|endofpoem|>
```

Generating Poems from the Fine-tuned GTP-3 Models

Because of the high cost of running GPT-3 on the OpenAI's servers (OpenAI-Pricing 2023), we limited our fine-tuning for poetry generation to two authors. We have ran-

[1] https://github.com/PeterS111/Fine-tuning-GPT-3-for-Poetry-Generation-and-Evaluation

AUTHOR: Walt Whitman
AUTHOR'S DATES: 1819-1892
COUNTRY: United States
TITLE: Paradise: In A Dream
THEME: Mysticism
SUMMARY: This poem describes a dream of Paradise, in which the narrator sees and hears beautiful things such as fragrant flowers, birds singing, a river of gold sand, the Tree of Life and the gateway to Heaven. The beautiful sight fills them with hope of seeing these heavenly things again one day.

Table 3: Example prompt used to generate the poem presented in Table 4. The complete tags are not shown in order to save space. Original tags are presented above.

A DREAM of Paradise,
As in a chamber, in the middle of the woods, I stood alone;
A few faint stars shining in the window,
And a trace of red in the eastern window, giving out the light,
As if some fire had been quenched there.

As I so stood, suddenly appeared before me,
As it had been a huge transparent angel, in shape
not different from a man,
But of such exceeding beauty, that I knew him to be nothing less than a god.

He stood before me silent;
His eyes fixed on me, full, full of meaning looked the god;
Not to speak his name he said,
But beckoning with his left hand, he drew me after him into the woods,
Till we came to the edge of a river, shallow and rapid,
And stood by the banks of the river, where he stood silent.

Then he said–"Shall I show you Paradise?"
And he reached his right hand into the water,
And drew out with his fingers a handful of white sand,
And put the sand upon his left hand,
And made the sign of the cross upon me with the sand,
and gave me the handful of sand, and said,
"Look!"
And I looked–and behold a Paradise,
Birds, and fragrant flowers, and streams of gold sand,
And the River of Life, flowing and glittering,
With trees, and buildings, and places of refreshment,
And on the banks, beautiful women, and attractively
dressed
men.

And this was the River of Life, which flowed and sparkled,
And the trees were decorated with light,
And the buildings with beautiful colors were suffused with light,
And the refreshment places were set forth with plate and glass,
And the home of glory and delight, and the place of perfect love.

And thence I saw all the earth,
And the cities, and the country with its farms,
And the roads and the teeming population,
And the farms with their fields and crops,
And the working of the machinery, and the noise of the whirr of the machines,
And the new products in the markets,
And the great ships at the wharves, and the steamers on the sea.

And I said, "How beautiful, how beautiful!"
And then I awoke, and I remembered the dream.

Table 4: Example poem generated by our system using the prompt shown in Table 3.

domly chosen Walt Whitman and Rudyard Kipling. This applies both to our single-author approach and when generating from the models fine-tuned on the seven authors' dataset. Given the information shown in Tables 1 and 2, and our fine-tuning on two poets, the number of fine-tuned models for poetry generation is 16 in our experiments (2 poets times 6 models in Table 1 plus 4 models in Table 2).

From each fine-tuned model, we generate 300 poems to be later used in evaluation in Part 2. In the case of models fine-tuned on the seven authors' dataset, we generate 300 poems in the styles of both of our selected authors. Generating a poem requires a summary and theme in the prompt. To make the poem generation exercise fair, we did not use summaries of the poems that were in any of the fine-tuning datasets. Instead, we summarised 150 poems for two additional authors, William Ernest Henley (English, 1849–1903) and Christina Rossetti (English, 1830–1894). We use those summaries as part of prompts for generating poems. Thus, for example, the prompt for generating poems in the style of Walt Whitman will have author, author's dates of life and author's country all set to Whitman's details, but the title, theme, and summary will be selected randomly from either Henley or Rosetti.

We have found that apart from generating poems from our fine-tuned models using the prompt including all the metadata used above, it is also possible to add a starting fragment of the poem after the <|startofpoem|> tag. This allows the user to keep improving the poem in an iterative way. We do not report any results on this way of generating poetry, but we mention it here to emphasize the practical applicability of our methodology because it could form the basis of a practical iterative poetry generation system.

An example prompt used to generate poetry from our system is presented in Table 3, and the poem generated from this prompt is presented in Table 4.

Part 2—Poetry Evaluation

Part 1 of this paper explains our method for poetry generation in the style of a specific poet. As explained above, we generated 300 poems using each of the 16 GPT-3 models that we fine-tuned. This part of the paper evaluates the quality of the generated poems.

We use automated evaluation following the methodology presented in our previous work (Sawicki et al. 2022), where BERT was used to distinguish between the poems written by the original author and the poems generated by fine-tuned GPT-2 models. Since BERT requires a large number of training data samples, those authors used 20-lines-long fragments of poems in their experiments. In this paper, we evaluate entire poems, and this gives a limited amount of samples per label. As a result, using BERT-based classifiers is not viable. We show that GPT-3 can instead be used to classify poems when the amount of training data is small. We fine-tune GPT-3 to classify if a poem is written by the original author or generated by the fine-tuned GPT-3. This means that we have two orthogonal fine-tuning processes of the GPT-3 models in our paper.

Using GPT-3 for classification requires the implementation of the logit bias during inference. Logit bias is an optional parameter passed to GPT models during text generation. It modifies the likelihood of specified tokens appearing in the generated text. This parameter is represented as a mapping from tokens to their associated bias values, which are between -100 (a ban) to 100 (exclusive selection of the token). Moderate values between -100 and 100 will change the probability of a token being selected to a lesser degree. When this parameter is used, the bias changes the original probabilities of tokens generated by the model prior to sampling. Thus, passing the logit bias parameter for only two tokens, representing our classes “0” and “1”, both with a value of 100, will result in the models being able to output only these two tokens (OpenAI-Documentation 2023). Without this modification, the model may produce answers that will not indicate any of the classes, giving inconclusive classification results.

Our methodology for classification-based evaluation of poems consists of two steps:

1. Establishing the accuracy of GPT-3-based classifiers by conducting a series of experiments classifying various types of texts.
2. Evaluating GPT-3-generated poetry against the works of original authors using GPT-3-based classifiers.

Step 1—Establishing the Accuracy of GPT-3-based Classifiers

To establish the accuracy of the GPT-3-based classifiers, we trained classifiers on two-class text classification problems where the similarity between classes was ranging from completely dissimilar to increasingly similar. First, we classified Walt Whitman’s poetry against the extracts from a book on machine learning (Sutton and Barto 2018). This was an example text that is very different from poetry. Then, we proceeded to classify Whitman’s poetry against fragments of prose from the Collected Works of Leo Tolstoy (Project Gutenberg 2022), and finally we classified Whitman’s poetry against the poetry of Rudyard Kipling as an example of two classes of text that are similar to each other. Additionally, we also classified extracts from the book on machine learning against fragments of prose by Tolstoy. Since all the poems in our dataset are between 100 and 500 words in length, when the samples from the book on machine learning or from the prose by Tolstoy are used, they have the random length between 100 and 500 words.

Walt Whitman vs book on machine learning			
	Correct	Incorrect	Accuracy
Ada	199	1	99.5%
Babbage	**200**	**0**	**100%**
Curie	**200**	**0**	**100%**
Davinci	199	1	99.5%
Walt Whitman vs Leo Tolstoy			
	Correct	Incorrect	Accuracy
Ada	**200**	**0**	**100%**
Babbage	199	1	99.5%
Curie	**200**	**0**	**100%**
Davinci	196	4	98%
Leo Tolstoy vs book on machine learning			
	Correct	Incorrect	Accuracy
Ada	196	4	98%
Babbage	**200**	**0**	**100%**
Curie	189	11	94.5%
Davinci	180	20	90%
Walt Whitman vs Rudyard Kipling			
	Correct	Incorrect	Accuracy
Ada	196	4	98%
Babbage	**200**	**0**	**100%**
Curie	197	3	98.5%
Davinci	199	1	99.5%

Table 5: Results of evaluating the accuracy of GPT-3-based binary classifiers in Step 1.

In all four of these experiments, the training/test split is 0.67/0.33. The training dataset consists of 200 samples per label, and the test dataset consists of 100 samples per label.

In order to determine which fine-tuned model produces the best results, for each experiment, we fine-tuned each of the four GPT-3 sizes: Ada, Babbage, Curie and Davinci. As per the instructions on the OpenAI website, we fine-tune Ada and Babbage classifiers for four epochs, and Curie and Davinci classifiers for one epoch.

The results of these experiments are presented in Table 5, and they show that there is almost no difference between the outcome from four different model sizes. This is a very useful finding, since it eliminates the need for using the largest Davinci-based models, thus greatly reducing experimental cost. Consistently, we find that GPT-3 can be a highly accurate text classifier. In almost every case, the accuracy of the classifiers was 98% or more, both on similar as well as dissimilar classes. The lowest score in all of these experiments was due to the Davinci model fined-tuned to classify the book on machine learning against the prose by Tolstoy, with the accuracy of 90.0%. The second worst performing model was Curie, also on the task of classifying the book on machine learning against the prose by Tolstoy, where it scored 94.5%. The scores for Ada- and Babbage-based classifiers were very similar. Overall, these experiments show that fine-tuned GPT-3 models are reliable as binary text classifiers to distinguish between different authors of poetry and different categories of text.

Since GPT-3-Babbage-based classifiers were most accurate on average, we chose the Babbage model as the basis for fine-tuning the classifiers for our poetry evaluation experiments below.

Step 2—Evaluating GPT-3-generated Poetry Against the Works of the Original Author Using GPT-3-based Classifiers

Now we describe our evaluation of GPT-3-generated poetry against the works of the original authors using GPT-3 as a classifier. We use the poems generated by our process of generating new poems described in Part 1 of the paper.

As in Step 1, the training/test split for each classification was 0.67/0.33. Each training dataset consists of 200 samples per label, each validation dataset consists of 100 samples per label. Our evaluation defines a two-class classification problem, where label 0 represents generated poems, and label 1 denotes the works of the original author. The results are presented in Table 6. All the classifiers in this experiment are fine-tuned GPT-3 Babbage models, built as we explained above. Entries in the first column in the table tell us which fine-tuned GPT-3 model's output was label 0 (these are the poetry generator models obtained in Part 1), and this output was evaluated against the works of original author placed in label 1.

The results show that the accuracy of the classifiers varied from 61.5% to 87.5%. A higher accuracy indicates that the classifier was able to distinguish the GPT-3-generated poetry from the original works of the authors with a higher degree of success. On the other hand, a lower accuracy implies that the classifier struggled to distinguish between the two and that the GPT-3-generated poetry was similar to the original work of the human authors. An accuracy of 50% would mean that the classifier cannot differentiate between generated and original poems. The best result that we obtained on Whitman's style is 61.5%, and it demonstrates quite high quality in the generated poems. The best result obtained on Kipling's style is 67%, which is less pronounced, but given the very high accuracy of this classification method in our calibration experiment reported in Table 5, one can argue that a large number of high quality poems was obtained on Kipling's style as well.

The results of classification show some differences in the quality between poetry generated from different models and different dataset sizes. Interestingly, we should note that poetry generated from Davinci-based models did not achieve the highest results for both authors. Smaller GPT-3 models are sufficiently powerful to generate high-quality poetry in a selected style. The justification for good performance of the smaller models may be the fact that the largest Davinci model may require more fine-tuning data to capture the style more faithfully.

The results in Table 6 also vary between the works of the two poets. Because of the high costs of running these experiments, we were limited to generating and classifying poetry of only two authors. Repeating these experiments with the works of other authors would provide more insights into style preservation of GPT-3 models, but our current results on the style of two poets indicate that our method has merit, and that it is possible to generate new poems in the style of a specific author.

Walt Whitman GPT-3 vs Walt Whitman original			
Model	Correct	Incorrect	Accuracy
Ada 4e	127	73	63.5%
Ada 7A 4e	140	60	70%
Babbage 4e	131	69	65.5%
Babbage 7A 4e	134	66	67%
Curie 1e	150	50	75%
Curie 4e	**123**	**77**	**61.5%**
Curie 7A 4e	131	69	65.5%
Davinci 1e	144	56	72%
Davinci 4e	174	26	87%
Davinci 7A 4e	137	63	68.5%

Rudyard Kipling GPT-3 vs Rudyard Kipling original			
Model	Correct	Incorrect	Accuracy
Ada 4e	170	30	85%
Ada 7A 4e	147	53	73.5%
Babbage 4e	**134**	**66**	**67%**
Babbage 7A 4e	142	58	71%
Curie 1e	173	27	86.5%
Curie 4e	160	40	80%
Curie 7A 4e	150	50	75%
Davinci 1e	175	25	87.5%
Davinci 4e	161	39	80.5%
Davinci 7A 4e	163	37	81.5%

Table 6: Results of experiments in Step 2 where GPT-3-generated poetry is compared against the works of the original author. Entries in the first column in the table indicate which fine-tuned GPT-3 model's output was evaluated against the works of the original author. 7A refers to the dataset of seven authors (2100 samples), 1e or 4e indicate that the model was fine-tuned for one or four epochs, respectively.

In conclusion, the results of the experiments in Step 2 suggest that the quality of GPT-3-generated poetry is high when smaller GPT-3 models are used as a base for fine-tuning, and that a dataset of only 300 samples is sufficient to fine-tune a poetry generator that can preserve the style.

Our results show that there is no significant difference between models fine-tuned on 300 samples vs models fine-tuned on 2100 samples. However, fine-tuning on a dataset consisting of many poets' works could open the possibility of mixing poets' styles in the output. Instead of setting all the author's metadata in the prompt to, for example, Kipling's or Whitman's details, we could declare the author as ”Rudyard Whitman”. This approach, however, requires further research.

These results should be interpreted with caution in the light of the fact that binary classifiers used are entirely black-box systems, i.e. we do not know how the classification was performed. However, having established the high accuracy of these classifiers in Step 1, we can, to some extent, rely on these results. Further investigation, especially including human evaluations, is necessary to thoroughly determine the quality of the GPT-3-generated poetry.

Discussion

Ventura (2016) suggests that to evaluate the generative system in the context of computational creativity, we should consider the factors of **novelty**, **value** and **intentionality**.

The system we proposed is capable of producing **novel** works, benefiting from the enormous amount of data contained in the original training dataset of the GPT-3 models. The prompting choices made by a human collaborator may also contribute to novelty.

As for **value**, the quality of the output was deemed indistinguishable from the works of the original authors on average in 25% of cases. Our workflow allows for precise control over the output, and therefore can be a valuable tool for collaborative poetry creation.

Intentionality is more complicated, but overall the outputs are mostly inline with its inputs, suggesting the products correlate with the system's process. However, the fine-tuned GPT-3 poetry generator does not produce anything on its own, and every generated poem is the result of user decisions. Whether a computer can at all be deemed creative is a matter of ongoing discussion (Guckelsberger, Salge, and Colton 2017); clearly, the machine does what it is told to do by its programmer and its user. Regardless of that, we can strive towards reducing the need for human input in producing the artifacts, or cherry-picking them from the multitude of system's outputs, and our system contributes strongly toward these goals.

It is also worth considering the model as containing the intentionality of its creators, in building a general-purpose language system, amongst its implicit goals is the creation of high-quality topical poetry, since poetry is a major identifier of success for creativity in humans.

Our workflow of augmenting the dataset with summaries and themes, followed by fine-tuning GPT-3 models allows to generate poems in the specific author's **style**, which has proved impossible through prompt engineering alone.

The status of the overall task of style preservation as computational creativity task has been considered by Brown and Jordanous (2022), who give an overall positive answer. Certainly, building new poems in an existing style can delight readers, and in this sense alone, it surely provides novelty and value; also, writing in an existing style can avoid concerns of building ”atypical” poems.

Conclusion

The main contributions of this paper are:

1. We create a dataset of out-of-copyright poems augmented with summaries and themes generated by GPT-3. This dataset can be used by researchers for further experiments with poetry generation.
2. We demonstrate that GPT-3 models fine-tuned on as few as 300 poems are effective poetry generators, able to generate poems in a desired style, and with a given theme and narrative. Smaller GPT-3 models fine-tuned for poetry generation perform as well as larger models (as evaluated by our method of binary classification), meaning that the task may not be as challenging as some other language tasks, and that it can be done fairly inexpensively.
3. We demonstrate that GPT-3 models fine-tuned for binary text classification on as little as 200 samples per label achieve on average 99% accuracy in separating those two classes, with smaller models performing equally good, or better, than much larger models.
4. Overall, we provide a system that is capable of generating high quality poetry, in user-controlled style and content. Our system can also be used in an iterative way: after providing the summary and metadata, the user can also provide a poem fragment, and continue generation from that point in the poem. Thus, our system can be a valuable "poet's assistant."

The workflow used in this paper might be a way to train specialised language models in general: to fine tune on an appropriate corpus where each item is accompanied by its summary, in order to generate new items from the user-provided summaries. We could see this workflow as a general-purpose way of taking advantage of the language fluency of GPT models, while also allowing for some focus on specific topics. This approach can still run afoul of standard concerns about artificial intelligence and knowledge, like Searle's Chinese Room argument (Searle 1980). More research is needed to explore this topic.

In future work, we will experiment with other ways of encoding the poems than by using summaries and themes.

We can also examine a poet's style change over the course of their career (Gervás 2011). Applying our current workflow to this task will require reducing the size of the fine-tuning dataset for poetry-generators even further, by splitting it into subsets, for example: 'EARLY WHITMAN', 'MIDDLE WHITMAN', 'LATE WHITMAN'. The question that will have to be answered first is: how small a dataset is sufficient to fine-tune GPT-3 for poetry generation?

In our dataset, the summary was almost always shorter than the poem. It would be interesting to test our approach on shorter poetic forms, like haiku, where the length of the summary would exceed that of the poem. It would be interesting to see how GPT expands the haiku into a summary, but also how it would generate the concise haiku from long summaries, especially to see if it can capture the structure of the haiku consistently.

Automated evaluation of poetry is an open problem. Our approach that uses GPT-3 is an encouraging one with a great potential for highly accurate results, but it is a "black-box" classifier. A promising alternative could be evaluation by virtual crowd, presented in (Goes et al. 2022), where the authors have examined the possibility of GPT-3 simulating the members of the jury that evaluates jokes, through answering the same questions that human evaluators were asked. The results were compared to the ground-truth of the human evaluation and found to be similar. This approach, however, has not been tested yet on poetry evaluation, and therefore, it is left for future research.

Author Contributions

Experimental design: PS with MG, FG, DB, AK, MP, SP; experimental implementation: PS with SP; writing: PS with MG, DB, FG, MP, AK, editing: MG, DB, FG, MP, AK.

Acknowledgements

We thank Anna Jordanous for useful discussions and feedback during the process of the development of this paper. We also thank the anonymous reviewers for their feedback and suggested references. The work of DB is supported by a Discovery Grant from the Natural Sciences and Engineering Council of Canada. MP is supported by the University of Kent GTA Studentship Award, Prins Bernhard Cultuurfonds, Hendrik Mullerfonds, and Vreedefonds.

References

Birhane, A., and Raji, D. 2022. ChatGPT, Galactica, and the progress trap. *Wired*.

Boggia, M.; Ivanova, S.; Linkola, S.; Kantosalo, A.; and Toivonen, H. 2022. One line at a time—generation and internal evaluation of interactive poetry. In *Proceedings of the 13th International Conference on Computational Creativity*. The Association for Computational Creativity.

Bons, V. 2022. GPT-3 as a muse: generating lyrics. https://wingedsheep.com/songwriting-with-gpt-3/.

Brown, D. G., and Jordanous, A. 2022. Is style reproduction a computational creativity task? In *Proceedings of the 13th International Conference on Computational Creativity*. The Association for Computational Creativity.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. In *Advances in neural information processing systems*, volume 33, 1877–1901.

Chakrabarty, T.; Padmakumar, V.; and He, H. 2022. Help me write a poem: Instruction tuning as a vehicle for collaborative poetry writing. *arXiv preprint arXiv:2210.13669*.

Corneli, J.; Jordanous, A.; Shepperd, R.; Llano, M. T.; Misztal, J.; Colton, S.; and Guckelsberger, C. 2015. *Computational poetry workshop:* making sense of work in progress. In *Proceedings of the Sixth International Conference on Computational Creativity*, 268–275.

Dai, A. 2021. GPT-2 for Emily Dickinson poetry generation. Stanford University, Unpublished Dissertation.

Dale, R. 2021. GPT-3: What's it good for? *Natural Language Engineering* 27(1):113–118.

Elgammal, A.; Liu, B.; Elhoseiny, M.; and Mazzone, M. 2017. CAN: Creative adversarial networks, generating "Art" by learning about styles and deviating from style norms. In *Proceedings of the Eight International Conference on Computational Creativity*, 96–103.

Ens, J., and Pasquier, P. 2018. Caemsi: A cross-domain analytic evaluation methodology for style imitation. In *ICCC*, 64–71.

Falk, M. 2021. Artificial stupidity. *Interdisciplinary Science Reviews* 46(1-2):36–52.

Floridi, L., and Chiriatti, M. 2020. GPT-3: Its nature, scope, limits, and consequences. *Minds and Machines* 30(4):681–694.

Gervás, P. 2011. Dynamic inspiring sets for sustained novelty in poetry generation. In *ICCC*, 111–116.

Goes, F.; Zhou, Z.; Sawicki, P.; Grzes, M.; and Brown, D. G. 2022. Crowd score: A method for the evaluation of jokes using large language model AI voters as judges. *arXiv preprint arXiv:2212.11214*.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2020. Generative adversarial networks. *Communications of the ACM* 63(11):139–144.

Guckelsberger, C.; Salge, C.; and Colton, S. 2017. Addressing the" why?" in computational creativity: A non-anthropocentric, minimal model of intentional creative agency. In *Proceedings of the Eight International Conference on Computational Creativity*, 128–135.

Gwern Branwen. 2019. GPT-2 neural network poetry. https://gwern.net/gpt-2.

Gwern Branwen. 2022. GPT-3 creative fiction. https://gwern.net/gpt-3.

Hämäläinen, M.; Alnajjar, K.; and Poibeau, T. 2022. Modern french poetry generation with RoBERTa and GPT-2. In *Proceedings of the 13th International Conference on Computational Creativity*. The Association for Computational Creativity.

Hämäläinen, M. 2018. Harnessing NLG to create Finnish poetry automatically. In *Proceedings of the ninth international conference on computational creativity*. Association for Computational Creativity (ACC).

Köbis, N., and Mossink, L. D. 2021. Artificial intelligence versus Maya Angelou: Experimental evidence that people cannot differentiate AI-generated from human-written poetry. *Computers in human behavior* 114:106553.

Lamb, C., and Brown, D. G. 2019. Twitsong 3.0: Towards semantic revisions in computational poetry. In *ICCC*, 212–219.

Lamb, C.; Brown, D. G.; and Clarke, C. L. 2017. A taxonomy of generative poetry techniques. *Journal of Mathematics and the Arts* 11(3):159–179.

Lee, J.-S., and Hsiang, J. 2020a. Patent claim generation by fine-tuning openai GPT-2. *World Patent Information* 62:101983.

Lee, J.-S., and Hsiang, J. 2020b. PatentTransformer-2: Controlling patent text generation by structural metadata. *arXiv preprint arXiv:2001.03708*.

Lee, J.-S. 2019. Personalized patent claim generation and measurement. *arXiv preprint arXiv:1912.03502*.

Li, P.; Zhang, H.; Liu, X.; and Shi, S. 2020. Songnet: Rigid formats controlled text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 742–751.

Liao, Y.; Wang, Y.; Liu, Q.; and Jiang, X. 2019. GPT-based generation for classical Chinese poetry. *arXiv preprint arXiv:1907.00151*.

Lo, K.-L.; Ariss, R.; and Kurz, P. 2022. GPoeT-2: A GPT-2 based poem generator. *arXiv preprint arXiv:2205.08847*.

Misztal, J., and Indurkhya, B. 2014. Poetry generation system with an emotional personality. In *ICCC*, 72–81.

Nikolov, N. I.; Malmi, E.; Northcutt, C. G.; and Parisi, L. 2020. Rapformer: Conditional rap lyrics generation with denoising autoencoders. In *Proceedings of the 13th International Conference on Natural Language Generation*, 360–373.

Oliveira, H. G. 2017. A survey on intelligent poetry generation: Languages, features, techniques, reutilisation and evaluation. In *Proceedings of the 10th international conference on natural language generation*, 11–20.

Oliveira, H. G. 2021. Exploring a masked language model for creative text transformation. In *Proceedings of the Twelfth International Conference on Computational Creativity*, 62–71.

OpenAI-Documentation. 2023. OpenAI API documentation. https://platform.openai.com/docs/introduction.

OpenAI-Pricing. 2023. OpenAI pricing. https://openai.com/api/pricing/.

Pachet, F., and Roy, P. 2014. Non-conformant harmonization: the real book in the style of take 6. In *ICCC*, 100–107.

Poetry Foundation. 2022. https://www.poetryfoundation.org/.

Project Gutenberg. 2022. http://gutenberg.org/.

Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I.; et al. 2018. Improving language understanding by generative pre-training.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1(8):9.

Rahman, F., and Manurung, R. 2011. Multiobjective optimization for meaningful metrical poetry. In *Proceedings of the Second International Conference on Computational Creativity*, 4–9.

Rashel, F., and Manurung, R. 2014. Pemuisi: a constraint satisfaction-based generator of topical Indonesian poetry. In *ICCC*, 82–90.

Sawicki, P.; Grzes, M.; Jordanous, A.; Brown, D.; and Peeperkorn, M. 2022. Training GPT-2 to represent two romantic-era authors: challenges, evaluations and pitfalls. In *Proc. of ICCC*. Association for Computational Creativity (ACC).

Searle, J. R. 1980. Minds, brains, and programs. *Behavioral and Brain Sciences* 3(3):417–424.

Stevenson, C.; Smal, I.; Baas, M.; Grasman, R.; and van der Maas, H. 2022. Putting GPT-3's creativity to the (alternative uses) test. In *Proceedings of the 13th International Conference on Computational Creativity*. The Association for Computational Creativity.

Sutton, R. S., and Barto, A. G. 2018. *Reinforcement learning: An introduction*. MIT press.

Toivanen, J.; Järvisalo, M.; Toivonen, H.; et al. 2013. Harnessing constraint programming for poetry composition. In *The Fourth International Conference on Computational Creativity*. The University of Sydney.

Toivanen, J.; Gross, O.; Toivonen, H.; et al. 2014. "The Officer Is Taller Than You, Who Race Yourself!": Using Document Specific Word Associations in Poetry Generation. In *Proceedings of the Fifth International Conference on Computational Creativity*. Jožef Stefan Institute.

van Dis, E. A.; Bollen, J.; Zuidema, W.; van Rooij, R.; and Bockting, C. L. 2023. ChatGPT: five priorities for research. *Nature* 614(7947):224–226.

Veale, T. 2013. Less rhyme, more reason: Knowledge-based poetry generation with feeling, insight and wit. In *ICCC*, 152–159.

Ventura, D. 2016. Mere generation: Essential barometer or dated concept. In *Proceedings of the Seventh International Conference on Computational Creativity*, 17–24. Sony CSL, Paris.

Wesek, A. 2019. A comprehensive study of state-of-the-art word embedding algorithms for natural language generation. University of Kent, Unpublished MSc Thesis.

Wöckener, J.; Haider, T.; Miller, T.; Nguyen, T. T. L.; Pham, M. V.; Belouadi, J.; Eger, S.; et al. 2021. End-to-end style-conditioned poetry generation: What does it take to learn from examples alone? In *Proceedings of the 5th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, 57–66.