Haiku Generation with Large Language Models

Stanford CS224N Custom Project

Victoria DiMelis
Department of Computer Science
Stanford University
vdimelis@stanford.edu

Brennan Megregian
Department of Computer Science
Stanford University
brennan4@stanford.edu

Mentor:
Hong Liu

Abstract

Generating poetry and other creative open-ended texts has been a long-standing challenge in the field of Natural Language Processing. In this paper, we present an approach to fine-tuning Large Language Models, specifically OpenAI's GPT-3 and Meta's OPT-125M, to generate haikus, a traditional form of Japanese poetry that consists of three lines and a total of 17 syllables. Our goal is to generate haikus that not only follow the traditional syllabic structure but also convey poetic imagery and evoke an emotional response in the reader. We evaluate our approach based on perplexity metrics in addition to qualitative human evaluation. We believe that our approach opens up new possibilities for using Large Language Models to generate creative and emotionally resonant text; our results show us that while it can be difficult for models to learn the structure of the haiku, they are surprisingly poetic after being fine-tuned on tens of thousands of haikus and using certain sampling and prompting techniques.

1 Introduction

This project investigates the performance of fine-tuned Large Language Models (LLMs) on haiku generation. A haiku is a type of short, syllabic poem originating in Japan. They follow a very specific structure, consisting of three lines, with each line holding 5, 7, and 5 syllables, respectively. Other notable features include a *kireji* ('cutting word') and a *kigo* ('seasonal reference'). While these components are characteristic of a traditional haiku, English haikus often allow for more artistic license and variation. Automatic poetry generation in general is a popular area of research within Natural Language Generation (Gonçalo Oliveira, 2017), and has been bolstered by the advent of increasingly large LLMs (Bena and Kalita, 2020).

Haiku generation is a particularly difficult and interesting task, as despite their short length, haikus must follow a specific structure and syllable pattern. Additionally, the creative and subjective nature of haikus add to this difficulty, as there is not a "correct" answer or output. Wu et al. (2017) explores haiku generation primarily through the use of Recurrent Neural Networks (RNN), while Miceli (2021) chooses to fine-tune OpenAI's GPT-2 model. Both papers present successful results, however Wu et al. focuses on Japanese haikus (whereas our focus is on English ones) and we build upon Miceli's work through the use of multiple LLMs, as well as additional fine-tuning and a larger dataset. The specific task of our project is to generate a haiku given an input word to be included in the poem. For example, if we input the word 'echoes', the generated haiku involving this might be:

The sound of the sea,
The sound of the wind and rain, [1]
Echoes in my mind

The overall goal of our project is to produce both fluent and creative haikus such as this by fine-tuning OpenAI's GPT-3 model and Meta's OPT-125M model. Our work produced semi-succesful AI-generated haikus and illustrated to us the difficulties of generating text that is both form-restrictive and creative.

[1] This is an example of an output generated by our fine-tuned GPT-3 model, given the zero-shot prompt: "write a haiku about echoes".

2 Related Work

Several studies have explored how language models can succesfully generate haikus and poetry in general.

In an attempt to generate poems of higher quality, Yi et al. (2018) implemented many different methods of training the model to write poetry. They obtain their best results when utilizing mutual reinforcement learning. Since their research was largely focused on quality of the generated poem, Yi et al. (2018) developed metrics for human evaluation to rate the generated poems. Inspired by the human evaluation metrics developed in this paper, our human evaluation metrics are nearly exactly the same.

Chakrabarty et al. (2022) also focuses on improving quality of general poetry generation. Instead of focusing on building the model, the authors focus on prompting (specifically instruction tuning). The authors consider prompts to language models that specify a topic to write about as 'collaborative'. The authors find that collaborative text generation can help performance when generating open-ended and creative text. We utilize 'collaborative' prompting when experimenting with different models (both fine-tuned and baseline) in an effort to produce better outputs. In addition to instruction tuning, we also utilize few-shot prompting as a way to potentially improve output.

In comparison, Wu et al. (2017) focuses on haikus specifically. The authors utilize Deep Neural Networks to train Rinna, an AI chatbot deployed by Microsoft Japan. They experiment with four different types of models: vanilla Recurrent Neural Network (RNN), a Recurrent Neural Network (RNN) with with Long Short-Term Memory (LSTM) blocks, a Recurrent Convolutional Neural Network (RCNN), and a Sequence Generative Adversarial Network (SeqGAN). This paper took a more technical approach to the task than we ended up taking, but it helped to motivate our initial research into the feasibility of generating haikus using language models trained on large corpora of haiku text.

Miceli (2021) also explores haiku generation. Rather than using Deep Neural Networks, Miceli fine-tunes GPT-2 to generate haikus. Miceli also explores plug and play language modeling (PPLM) to improve the performance of the fine-tuned model. The model generated, known as *Haikoo* successfully generates haikus and is evaluated on perplexity and human evaluation. Of all papers in this section, our paper most directly builds on Miceli's work, as we too focused on fine-tuning models and improving their performance. Like Miceli, we focus on perplexity and human evaluation to evaluate our model performance on the task. However, we build upon this research through experimenting with multiple LLMs and a larger dataset.

Given the many great papers we found in the space of poetry generation, we endeavored to take aspects from each paper to build a high-performing haiku generation model within our focus of fine-tuning LLMs.

3 Approach

Fine-tuning. Our primary approach involves fine-tuning Large Language Models on a haiku dataset for the purpose of generating unique and creative haikus given an input topic. Our first approach was to fine-tune and then utilize the Curie model form OpenAI's GPT-3 architecture. The GPT-3 Curie model is a Generative Pre-Trained Transformer with 175 billion parameters (Brown et al., 2020). For this fine-tuning, we used OpenAI's API which allowed us to interact with this pre-trained model, passing in the dataset and hyperparameters on which to fine-tune the model. The baseline model against which we compared our fine-tuned model is simply the standard Curie model. Most of the coding for fine-tuning GPT-3 was taken from OpenAI's documentation which provides examples that demonstrate how to interact with the API.

In addition to GPT-3, we also attempted fine-tuning a much smaller model, Meta's OPT-125M model - which is also a Pre-trained Transformer Language Model. This model has only 125 million parameters (Zhang et al., 2022), so it is notably smaller than the Curie model. Unlike GPT-3, which we had to access through an API, OPT-125M is a fully released pre-trained model, so we were able to directly write our own code to fine-tune the model. We utilized Hugging Face and the built-in trainer to conduct our fine-tuning. We were curious about how using a much smaller model could hypothetically improve our results: the OPT-125M model requires less data than GPT-3 to fine-tune effectively, as the smaller number of parameters means it is more easily able to learn from a smaller amount of data. Similar to our approach with GPT-3, we use the non-fine-tuned OPT-125M model as our baseline to help us evaluate the fine-tuned OPT-125M model.

Sampling. On top of just fine-tuning the models, we also implemented beam search on the OPT-125M fine-tuned model to see how this sampling technique affected performance as opposed to the default maximum likelihood estimate (MLE) generation. For the GPT-3 Curie model, we experimented with different temperature values. For GPT-3, possible temperature values range from 0 to 2. Temperature is a parameter used to increase or decrease the confidence a model has in its most likely response - higher temperature values correspond to more randomized and diverse text outputs.

Temperature is applied in decoding, during the softmax normalization step such that given some temperature value, τ, the equation becomes:

$$\sigma(z_i) = \frac{\exp(\frac{z_i}{\tau})}{\sum_{j=0}^{N} \exp(\frac{z_j}{\tau})}$$

Prompting. We also experimented with zero-shot and few-shot prompting on all of our baseline and fine-tuned models to explore how prompting affects the quality of our output. The zero-shot prompt is formatted as follows: "Write a haiku about {key word} -> ". To construct our few-shot prompt, we randomly selected 3 haikus from our initial fine-tuning dataset and prepended them to the zero-shot prompt.

4 Experiments

4.1 Data

For both of our approaches described above, we used subset of multiple haiku datasets, including one from Hugging Face (statworx, 2022), and two from Kaggle (Jhalani, 2021) bfbarry (2021). Initially, we just used a subset of the Hugging Face dataset, excluding haikus that did not follow the 5-7-5 syllabic structure and that included harmful or inappropriate content using OpenAI's content Moderation API. However, we realized this resultant dataset of $\sim 14,000$ examples was limiting the performance of our fine-tuned models. As a result, we collected haikus from two Kaggle datasets mentioned above, filtered them in a similar way as the Hugging Face dataset, and then combined all of these examples. This left us with a final dataset consisting of $59,483$ haikus. The Hugging Face dataset provided a key word/topic for each haiku, and we used these to reformat our dataset for fine-tuning such that each haiku was preceded by a 'prompt' of the form: "Write a haiku about {key word} -> ". The two Kaggle datasets however did not include such key words, so we first ran their respective haikus through OpenAI's `text-curie-001` with the following prompt: "Identify a single key word in the following poem: {insert haiku}". We then set the generated key word as the key word in the fine-tuning prompt. A single example of the final format of our fine-tuning data is: {"prompt": "Write a haiku about white sands ->", "completion": " White sands on the beach / And pink petals off a branch / Drifting in the wind"}.

4.2 Evaluation method

Quantitative. The quantitative evaluation metric used in both of our approaches is the perplexity of the generated haiku outputs given the fine-tuning dataset. Perplexity is a common intrinsic evaluation method used throughout the literature on haiku generation (Miceli, 2021)(Wu et al., 2017). It is defined as follows:

$$PP(W) = \left(\frac{1}{P(w_1, w_2, \ldots, w_N)}\right)^{\frac{1}{N}}$$

For both the GPT-3 and OPT-125M baseline and fine-tuned models, we provided the same prompt as described above, for each key word in a fixed list of topics we composed, including 'Wind', 'Nature', and 'Love'[2]. We then calculated the perplexity of each prompt's individual output, as well as the overall perplexity of all the test prompts' outputs. We used unigrams to compute the perplexity for both zero-shot and few-shot prompts. To implement this evaluation, we utilized the Natural Language Toolkit's (NLTK) `lm` package, specifically the Laplace Maximum Likelihood Estimation model, within our own written code to to identify the probabilities of each unigram and then to compute the relevant perplexity (Bird and Klein, 2009).

Qualitative. While this quantitative evaluation method provided some indication of the overall performance of our models, it is not holistic. Since haikus, and poetry in general, are inherently subjective in terms of quality and style, it was necessary that we supplement our method with qualitative human-evaluation. Human-evaluation is utilized frequently throughout creative text generation tasks, such as in Yi et al. (2018), Bena and Kalita (2020), and Miceli (2021). We built upon techniques outlined in Yi et al. and Miceli's research. In particular, we measured each of the generated outputs from the test prompts, for all baseline and fine-tuned models, against five metrics. Firstly, each generated haiku was given a binary score (1 or 0) indicating whether or not it was actually a valid haiku (i.e. that its syllabic structure was indeed 5-7-5). Next, we evaluated form, a metric of how close the output was to a haiku. Scores for this metric were integers between 1 and 5, with a score of 5 representing a perfect haiku and a score of 1 representing a nonsense chunk of text. Outputs that, for example, were three lined poems of incorrect syllabic structure were awarded a form score of 4. The third human evaluation metric used was topic, a metric of whether the poem was

[2] See Appendix A.1 for the full list of test prompt topics.

actually about the prompted word. Scores for this metric were integers between 1 and 5, with a score of 5 representing a poem that literally contained the prompted word and a score of 1 representing a poem that had nothing to do with the prompted word. Next, we evaluated fluency, a metric of whether the output made sense. Scores for this metric were also integers between 1 and 5, with a score of 5 representing a perfectly fluent output and a score of 1 representing a nonsense output. Finally, we evaluated the metric of quality, which we used to encode artistry and poeticism. Scores for this also ranged between 1 and 5, with a score of 5 representing an artistic poem on par with something written by a human poet and a score of 1 representing a poem that has no artistic merit whatsoever.

We utilized the binary haiku validity to compute the proportion of valid haikus generated, and we took the average of the other 4 human evaluation metrics to create an overall human evaluation score for each model evaluated. In addition to our perplexity metrics, we felt that this evaluation strategy did a good job at reflecting the actual quality of the outputs we inspected.

4.3 Experimental details

We used OpenAI's API to interact with and fine-tune the Curie model from GPT-3. Using a learning rate of 0.2, a prompt loss weight of 0.01, a batch size of 64, and 4 epochs, fine-tuning Curie took around 45 minutes. For this model, we also experimented with the effects of different temperature values during prompting time, specifically temperatures of 0, 0.2, 0.4, 0.8, 1, and 1.4. To fine-tune OPT-125M, we used the trainer tool from Hugging Face. We used a learning rate of 0.00005 and a batch size of 8. Fine-tuning the data took about 11 hours to run on a local device. When we implemented beam search on the fine-tuned OPT-125M model, we experimented with different beam sizes, but settled on 20 beams, as this gave us the best perplexity score.

4.4 Results and Analysis

Below, Table 1 demonstrates the perplexity scores calculated for each model, with different prompting and sampling techniques. The following values (rounded to one decimal place) were computed based on the outputs generated by each model for the given fixed list of test prompts (both zero-shot and few-shot prompts) using only unigrams. In this case, we used a temperature of 0. The lowest perplexities across all models for each column in shown in bold. Table 2 presents the qualitative evaluation scores for all the same models, with temperature 0. The highest scores across all models for each column in shown in bold.

Table 1: Perplexity Results

Model	Prompting/Sampling	Overall Perplexity	'Wind' Perplexity	'Echoes' Perplexity
GPT-3 Curie Baseline	Zero-Shot	1003.5	1065.8	850.6
	Few-Shot	633.7	840.8	998.0
GPT-3 Curie Fine-tune	Zero-Shot	494.6	460.0	584.8
	Few-Shot	1115.5	1717.9	1623.0
OPT-125M Baseline	Zero-Shot	1311.3	785.5	3167.5
	Few-Shot	3191.0	2740.1	2740.1
OPT-125M Fine-tune	Zero-Shot	**418.0**	1290.6	**411.4**
	Zero-Shot, Beam Search	524.1	**362.0**	726.8
	Few-Shot	644.0	1072.5	474.0

Table 2: Human Evaluations

Model	Prompting/Sampling	Overall Rating	% Haikus	Form	Topic	Fluency	Quality
GPT-3 Curie Baseline	Zero-Shot	3.1	0.0	3.8	4.5	2.2	2.0
	Few-Shot	4.0	4.3	3.9	4.7	4.3	3.1
GPT-3 Curie Fine-tune	Zero-Shot	**4.5**	**78.3**	**4.8**	4.8	**4.7**	3.9
	Few-Shot	4.1	0.0	4.0	4.8	4.3	3.3
OPT-125M Baseline	Zero-Shot	1.8	0.0	1.0	2.9	1.8	1.3
	Few-Shot	3.4	56.5	3.5	3.1	3.6	3.4
OPT-125M Fine-tune	Zero-Shot	4.0	13.0	4.0	4.7	4.0	3.4
	Zero-Shot, Beam Search	**4.5**	39.1	4.3	4.6	**4.7**	**4.4**
	Few-Shot	3.9	21.7	4.2	**4.9**	3.7	3.0

GPT-3. Looking at Table 1, the fine-tuned GPT-3 model was able to achieve a lower perplexity score than its respective baseline model - the lowest overall perplexity achieved by any GPT-3 fine-tuned model was 494.552, whereas for the baseline model it was 633.707. This result is expected, and demonstrates the success of our fine-tuning approach for the specified haiku-generation task. Examining the performance of GPT-3 further, we observed that for the baseline model, few-shot learning achieved much better scores than zero-shot learning, whereas for the fine-tuned model, zero-shot learning was more successful. While this behavior is somewhat surprising, it can perhaps be explained by the nature of the models and the prompts themselves. Given the baseline model has less exposure to our specific haiku data, it is understandable that a few-shot approach would be better, as this gives the model an increased ability to predict a haiku with the desired format and structure. For the fine-tuned model, the structure of the fine-tuning data was of the form of a zero-shot prompt, so it is perhaps understandable that this model was better able to predict what text to generate given this familiar form, rather than the less-seen few-shot prompt. The models' perplexities for the test prompts with key words 'wind' and 'echoes' are also given, and while there is understandably variation in the performance of the models across these prompts, nevertheless the trends remain the same as just described.

Beyond the perplexities given in Table 1, we also experimented with the affect of different sampling temperatures on the various GPT-3 models. Below, Figure 1 demonstrates the overall perplexity scores for the fine-tuned and baseline GPT-3 models with varying temperature values, using zero-shot learning. As we can see, perplexity increases with the temperature values. This is expected, considering higher temperature corresponds to more diverse outputs. For both models, temperatures of 0 to 0.4 have similar perplexities. This is reflected in the outputs themselves as from temperature 0.8 on the generated haikus become increasingly nonsensical. For all temperatures shown, the baseline model consistently performs worse than our fine-tuned model for zero-shot learning. As described in detail above, this demonstrates the success of our fine-tuned model.[3]

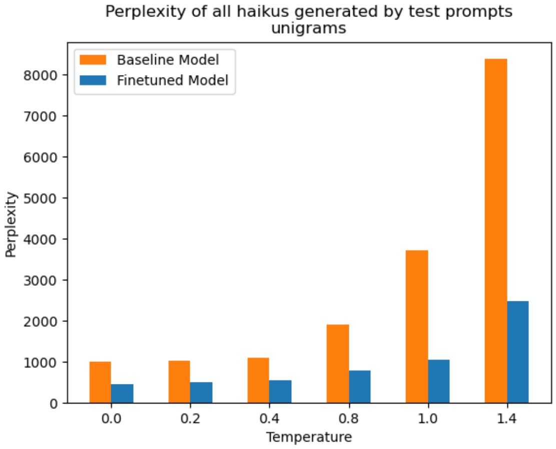

Figure 1: Perplexities with Temperature

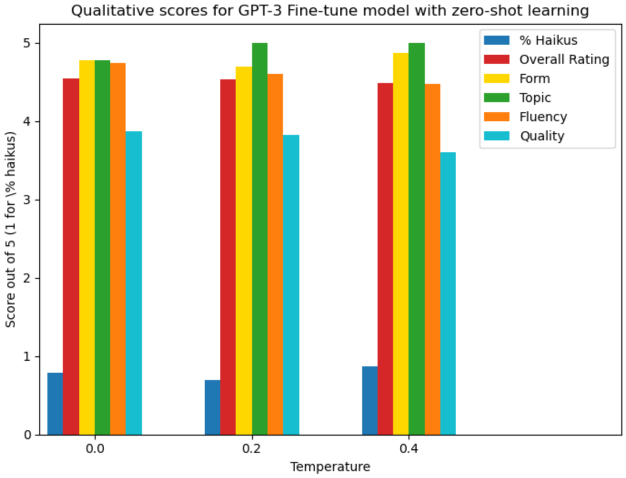

Figure 2: Qualitative Scores

Looking at Table 2, we see that the GPT-3 fine-tuned model with zero-shot learning achieves the highest score across all categories, for all GPT-3 models. This behaviour reflects the perplexity scores, and given the same reasoning as above, demonstrates the success of our fine-tuned model. 'Topic' was the highest scoring category, which shows how these models were better able to learn and predict the content of the haikus rather than structure, as demonstrated by the very low '% Haikus' scores (excluding the fine-tuned model with zero-shot learning). The human-evaluation results of varying temperature on the haikus generated by the fine-tuned GPT-3 model with zero-shot learning are presented in Figure 2. As with perplexity, temperatures 0, 0.2, and 0.4 perform very similarly. While temperature 0 achieves slightly better overall rating, fluency and quality scores, temperature 0.4 surprisingly produces the highest percentage of valid haikus and sticks to the input topic the most. This demonstrates that a temperature of 0.4 may be enough to slightly increase the diversity of the outputs, but that it is not a high enough value to make the output undesirable or nonsensical. Overall, all these temperatures result in impressive performance, with overall ratings of ~ 4.5.

[3] As described above, for few-shot learning, the baseline model of GPT-3 has lower perplexity than the fine-tuned model - this behavior remains for different temperatures, and a plot is given in Appendix A.2.

OPT-125M. For OPT-125M, we similarly found that the fine-tuned OPT-125M models achieved lower perplexity than the corresponding baseline models. The lowest perplexity achieved by a baseline OPT-125M model was 1,311.250; while the lowest perplexity achieved by a fine-tuned OPT-125M model was 418.021 (this was actually the lowest perplexity out of all models discussed, including GPT-3 models), as seen in Table 1. Even the *highest* overall perplexity scored by a fine-tuned OPT-125M model (644.0) is less than half of this lowest baseline OPT-125M model perplexity (1,311.250). The fact that the perplexity is lower than the fine-tuned models is expected, as this suggests that the fine-tuning efforts improved the haiku generation. In terms of human evaluation, the fine-tuned models again outperform the baseline. The highest overall human evaluation rating for an OPT-125M baseline model was 3.40, as compared to the highest overall evaluation for a fine-tuned OPT-125M model, which was 4.51; in fact, the outputs for the baseline model when using zero-shot prompts received a rating of 1.75 – the lowest rating of all models, including GPT-3 models, as seen in Table 2. Again, this suggests that the fine-tuning helped to generate better outputs.

When analyzing prompting strategies of OPT-125M, we find that zero-shot prompting (lowest perplexity of 418.0) actually achieves lower perplexity than few-shot prompting (lowest perplexity of 644.0). For the fine-tuned OPT-125M models, this makes sense since they were fine-tuned on zero-shot prompt/completion pairs. However, for the baseline this behavior is a bit more confusing. Upon inspection, it is clear that the few-shot prompting on the baseline model produced outputs that were all incredibly similar, sometimes producing the same exact haiku for different prompts. Therefore, this explains the lower perplexity when zero-shot prompting the baseline. In terms of human evaluation ratings, the baseline OPT-125M models produced higher quality models when given a few-shot prompt (overall human evaluation rating of 3.4). As discussed, the few-shot prompting on the baseline resulted in many repeat outputs, but these outputs were valid and sensical haikus, so the human evaluation rating was much higher. When zero-shot prompting the baseline, outputs were largely nonsensical and invalid haikus, so the zero-shot OPT 125-M model ended up with the lowest overall human evaluation rating of all models considered in the table (1.75). For fine-tuned OPT-125M models, we see that few-shot prompting performs worse than zero-shot prompting. The few-shot prompting received an overall human evaluation rating of 3.9, while the zero-shot prompting of the fine-tuned model received an average overall human evaluation rating of 4.25. Again, this is likely due to the training corpus used when fine-tuning.

Finally, we analyze the implementation of beam search for text generation on the OPT-125M fine-tuned model. While the model with beam search has a slightly higher perplexity (524.1 – still lower than most models analyzed) than the MLE fine-tuned model (with perplexity 418.0), we see that the human evaluation rating is higher when beam search is implemented (4.5 vs. 4.0). All human evaluation metrics are included in Figure 3 below. Therefore, we see that beam search helps to produce poems of higher quality and of comparable perplexity.

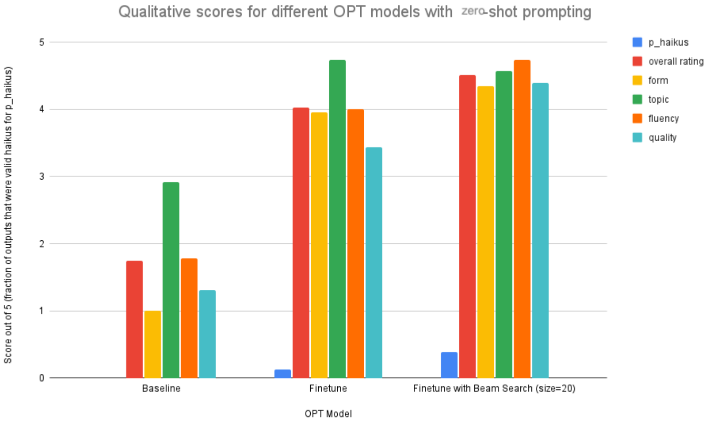

Figure 3: Qualitative Scores

Overall. Overall, we see that the OPT-125M models on average trend towards lower perplexity than the GPT-3 models, but they perform similarly well after fine-tuning in terms of human evaluation. The GPT-3 baseline models also outperformed the OPT-125M baseline models both quantitatively and qualitatively. This is likely because the OPT-125M is much smaller than GPT-3, so its performance is worse at the baseline level. The OPT-125M models by far had the most improvement after fine-tuning when compared to the GPT-3 models; the fine-tuned OPT-125M models ended up being comparable to and sometimes outperformed the fine-tuned GPT-3 models. This makes sense since OPT-125M is a much smaller model than GPT-3, so we expect that fine-tuning has more of a positive effect on the task. In terms of overall performance, our models greatly struggled at producing valid haikus. The majority of the models generated valid haikus less than half of the time. When inspecting the ‘form’ scores, however, we see that the majority of models were able to consistently produce poems that were three lines and considered *close* to a haiku. Therefore, this shows that syllabic structure proves to be very difficult for a model to learn, even when fine-tuned on a very large, syllabically correct dataset.

Example Outputs. Here are outputs generated by each of the tested models on the prompt, *"Write a haiku about dancing:"*

GPT-3 Curie Baseline Zero-Shot
I am dancing,
I am dancing,
I am dancing,

GPT-3 Curie Baseline Few-Shot
the dance of the stars
in the night sky
the dance of the moon

GPT-3 Curie Fine-tune Zero-Shot
I’m dancing in my
living room to the new I
Love It album now

GPT-3 Curie Fine-tune Few-Shot
dancing in the rain
the water’s reflection
is a mirror

OPT-125M Baseline Zero-Shot
I’m a dancer, I

OPT-125M Baseline Few-Shot
dancing of the heart
in multiple dimensions
whirl across the sky

OPT-125M Fine-tune Zero-Shot
I’m dancing with
my boyfriend and I’m so happy
I’m so happy

OPT-125M Fine-tune Zero-Shot
with Beam Search
ive always wanted
to be a dancer but i
don’t know what to do

OPT-125M Fine-tune Few-Shot
dancing in the snow
the snow is falling down on
me like a dance floor

From these outputs, we see that most of the models were able to successfully generate 3-lined poems. Out of GPT-3 models, only fine-tuned model with zero-shot learning produced a valid haiku, whereas for the OPT-125M models, the fine-tuned model with zero-shot learning and beam search, the fine-tuned model with few-learning, and the baseline model with few-shot learning were all able to generate valid, sensical haikus. In all cases, all models were able to include the topic word, ‘dancing’, which further demonstrates that the models struggle much more with the structure of haikus rather than their content.

5 Conclusion

The results of our project demonstrate how fine-tuning Large Language Models is a powerful tool that can increase their performance on specified tasks. For both the GPT-3 and OPT-125M models, the fine-tuned versions outperformed their baselines. Balancing creativity with a rigid structure and form proved challenging, but nevertheless the GPT-3 fine-tuned model with zero-shot prompting and the OPT-125M fine-tuned models with zero-shot prompting (both with and without beam search) were able to produce many fluent and artistic haikus. The often similar performance of these three models suggest larger models are not always better, and that smaller models such as OPT-125M may exhibit superior performance given a goal of achieving a singular specified task. This project also demonstrates how prompting techniques, instruction-tuning, and sampling choices such as temperature and beam search can greatly affect the generated output, both positively and negatively. Our findings indicate that these techniques, when utilized thoughtfully, can significantly enhance the caliber of the generated haikus, with a higher degree of fluency, quality, and form. Given the primary limitation of our models was their struggle to produce haikus of valid structure, future work would focus on improving this ability, perhaps through more fine-tuning data or better instruction-tuning. Overall, the results of this research have promising implications for the development of language models capable of producing high-quality creative writing in various genres, outside of haikus and poetry. The methods and techniques explored in this paper can serve as a foundation for future work in the field of creative writing generation and can be further built upon to create even better models.

References

Brendan Bena and Jugal Kalita. 2020. Introducing aspects of creativity in automatic poetry generation.

bfbarry. 2021. Kaggle haiku dataset. `https://www.kaggle.com/datasets/bfbarry/haiku-dataset`. Accessed: 2023-03-13.

Edward Loper Bird, Steven and Ewan Klein. 2009. Natural language processing with python. lm package. `https://www.nltk.org/api/nltk.lm.html`.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom

Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *CoRR*, abs/2005.14165.

Tuhin Chakrabarty, Vishakh Padmakumar, and He He. 2022. Help me write a poem: Instruction tuning as a vehicle for collaborative poetry writing.

Hugo Gonçalo Oliveira. 2017. A survey on intelligent poetry generation: Languages, features, techniques, reutilisation and evaluation. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 11–20, Santiago de Compostela, Spain. Association for Computational Linguistics.

Harshit Jhalani. 2021. Kaggle haiku dataset. `https://www.kaggle.com/datasets/hjhalani30/haiku-dataset?resource=download`. Accessed: 2023-02-13.

G. Miceli. 2021. Haiku generation, a transformer based approach, with lots of control.

OpenAI. Openai documentation. `https://platform.openai.com/docs/introduction`. Accessed on 02/01/2023.

statworx. 2022. Hugging face haiku dataset. `https://huggingface.co/datasets/statworx/haiku`. Accessed: 2023-02-13.

Xianchao Wu, Momo Klyen, Kazushige Ito, and Zhan Chen. 2017. Haiku generation using deep neural networks.

Xiaoyuan Yi, Maosong Sun, Ruoyu Li, and Wenhao Li. 2018. Automatic poetry generation with mutual reinforcement learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3143–3153, Brussels, Belgium. Association for Computational Linguistics.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pre-trained transformer language models.

A Appendix (optional)

If you wish, you can include an appendix, which should be part of the main PDF, and does not count towards the 6-8 page limit. Appendices can be useful to supply extra details, examples, figures, results, visualizations, etc., that you couldn't fit into the main paper. However, your grader does not have to read your appendix, and you should assume that you will be graded based on the content of the main part of your paper only.

A.1 Test Prompts

The list of key words included in our test prompts is as follows: 'wind', 'nature', 'fairies', 'gardens', 'space', 'children', 'beach picnic', 'vacation', 'valentines day', 'salad', 'cactus', 'california', 'wine', 'cities', 'love', 'the moon', 'guitar', 'daydreams', 'earthquakes', 'dancing', 'echoes', 'the sea', 'trees'.

A.2 Perplexity Plots

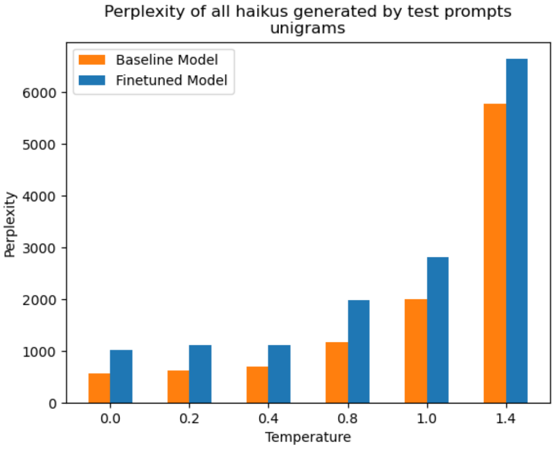

Figure 4: GPT-3 Few-Shot Perplexities with Temperature