ENSAE Paris
Mastère Spécialisé Data Science
2024 – 2025

– MACHINE LEARNING WITH NLP –

Phoetry

Authors
Anne-Laure Kodro TANOH
Laure ZABAR

Professor
M. Christopher KERMORVANT

Due on May 4th, 2025

Abstract

This project explores how to generate poems from images by combining visual recognition and natural language generation. It consists of two main components. First, it uses OpenAI's CLIP model to identify the main object in an image from a predefined list of nature- or food-related labels. The label with the highest score is selected as the image's theme. Then, a fine-tuned GPT-2 model takes this label as input and generates a poem around it in a chosen style, such as a haiku or a free verse poem. The result is a seemingly "coherent" poem inspired by the visual content of the image.

1 Introduction

Recent advances in pre-trained large language models (PLLMs) have significantly improved the quality of machine-generated text across a variety of tasks, from question answering to text summarization. Among these tasks, poetry generation stands out due to its demand for not only fluency and coherence, but also stylistic elegance, emotional depth, and sometimes structural constraints. Models like GPT-2 and GPT-3 have demonstrated their ability to mimic poetic form, either through prompt engineering or fine-tuning on curated poetic corpora (Bena & Kalita, 2020; Beheitt & Hmida, 2022).

In parallel, the emergence of multimodal models such as CLIP (Contrastive Language-Image Pre-training) has enabled systems to bridge the gap between visual and textual modalities. CLIP learns to associate images with text-based descriptions, making it a powerful tool for image-based content generation.

This project aims to combine these two directions, multimodal understanding and poetic language modeling, into a single system capable of generating poetry from images. The proposed pipeline consists of two main components. First, OpenAI's CLIP model is used to identify the most relevant concept in an input image from a curated list of labels related to nature and food. This identified label is treated as the "theme" for the poem. Then, a GPT-2 model fine-tuned on poetic texts uses this theme as a prompt to generate a poem. The system supports different poetic forms, namely haikus and classical poems.

Our goal is not to outperform existing models on standard benchmarks, but rather to explore how a lightweight, open-source pipeline can produce seemingly coherent and stylistically relevant poetic texts from visual inputs. This work also contributes to the growing interest in creative applications of language models, especially at the intersection of vision and language.

2 State-of-the-art

Text generation, also known as natural language generation, has been one of the most important sub-fields in natural language processing (NLP), and it aims to produce plausible and readable text in a human language, from the input data in various forms including text, image, table and knowledge base (Li, Tang, ZHao, & Wen, 2022). In the last decades, text generation techniques have been extensively applied to a wide range of applications: question answering, translation, and text summarization.

Insert 1: Text generation specification

A text can be modeled as a sequence of tokens $y = \langle y_1, ..., y_j, ..., y_n \rangle$, where each token y_j is drawn from a vocabulary $\mathcal{V}$. The task of text generation aims to generate plausible and readable text in a human language. In most cases, text generation is conditioned on some input data (e.g., text, image, tabular, and knowledge base), which is denoted as x. In particular, the generated text is expected to satisfy some desired language properties such as fluency, naturalness, and coherence. We denote the desired properties for output text as a property set $\mathcal{P}$. Based on the above notations, the task of text generation can be formally described as:

$$y = f_{\mathcal{M}}(x, \mathcal{P}), \tag{1}$$

where the text generation model $f_{\mathcal{M}}$ produces the output text y given the input data x, satisfying some special properties from the property set $\mathcal{P}$. In this survey, the text generation model $f_{\mathcal{M}}$ is specially crafted based on a pre-trained large language model (PLMM).

Source : *Li et al. (2022)*

Over the last years, the field of NLP has witnessed a shift in methodologies with the advent of PLLMs (Schneider, Klettner, Simperl, & Matthes, 2024). Unlike traditional supervised learning approaches that rely on annotated datasets, PLLMs are trained in a self-supervised manner, predicting tokens within vast amounts of unlabeled data. And with the emergence of Transformers and higher computational power, the architecture of PLLMs has evolved from shallow to deeper architectures, such as BERT and OpenAI GPT.

In particular , to predict the next word, GPT-2 was trained on a large corpus (WebText) containing 40 GB of text (Radford, et al., 2019). GPT-2 architecture has proven to model the English language and has obtained state-of-art tasks such as summarization, machine translation, and question answering.

Poetry generation is arguably the toughest of the text generation subtasks, since the poem must be generated in an elegant manner and ideally following a particular structure (Beheitt & Hmida, 2022). It has been a common reasearch subject over the years.

Wang and al. (2016) apply the attention-based sequence-to sequence model to generate Chinese Song iambics. Specifically, they encode the cue sentences by a bi-directional Long-Short Term Memory (LSTM) model and then predict the entire iambic with the information provided by the encoder. They use a song iambics corpus (Songci) collected from the Internet, which consists of 15 689 Song iambics in total, of which 15 001 are used for training and 688 are used for test. A second dataset involves two corpora used to train the word embedding model, including the Gigawordcorpus (contains roughly 1.12 billion Chinese characters) and the Songci corpus (1.22 million Chinese characters roughly). The model asks users to enter a first line for the poem, and then the model generates the other lines. To learn the vector representations of the words, authors used Word2Vec. Finally, to evaluate the model, in the first phase, the authors focus on configuration selection, e.g., which corpus to use to train the word vectors and whether the word vectors should be adapted during the attention model training. In the second phase, they compare the attention model with the best configuration and the alternative approaches using two most popular tunes of Song iambics in the experiment, 'Partridge Sky' and 'Pusaman', for which there are 441 and 403 iambics respectively in the training data. The

evaluation is based on three metrics: poeticness (if the generated iambics follow the regulation on tone and rhyme), fluency (if the sentences are fluent and convey reasonable messages), and meaningfulness (if the entire generation fo cuses on a single theme and exhibits some consistent seman tic meaning). This evaluation is conducted by experts. They also use Bilingual Evaluation Understudy (BLEU-2) score as the second evaluation metric to compare the original song iambucs to the generated ones.

Bena and Kalita (2020) propose taking an approach that fine-tunes GPT-2, a pre-trained language model. They extend prior work on poetry generation by introducing creative elements. Specifically, they generate poems that express emotion and elicit the same in readers, and poems that use the language of dreams—called dream poetry. To enable the expression of emotion in generated poems, there is preliminary step of scoring a corpus of downloaded poems for emotion to produce subsets of poems that express one of eight different identified emotions. For this task, EmoLex, a word-level emotion lexicon is used, that associates English words with the eight different emotion categories. Each poem (or book of poems) in the dataset is given a score that is the total of the associated emotion scores in EmoLex for each word. This step is followed by the actual generation of poems by fine-tuning the pre-trained GPT-2 natural language model. They train eight separate models for eight different emotions, each on a sub-corpus predominantly demonstrating a particular emotion. To generate poems that use dream-like language, they create a text corpus composed of a large number of dream transcriptions created in first person by actual viewers of dreams. In this case, they apply transfer learning by fine-tuning the pre-trained GPT-2 on the dream corpus, followed by training again on poetry. To evaluate, a crowd-sourced analysis is firstly conducted: the authors present four poems from each category to ten native English speaker humans, with undergraduate level educational backgrounds. The poems presented are randomly selected from the top 20 EmoLex scored poems out of a pool of 1,000 generated poems and the reviewers are asked to rate each poem based on the emotions elicited within. The authors are able to produce poems that correctly elicit the emotions of sadness and joy 87.5 and 85 percent. The evaluation of dream-like poems is the same with four generated poems presented to ten judges, who judge on three quality conditions: the poem is generally a first-person expression, the poem's main substance is dream or vision like and the poem recounts or foretells an experience or event. Analysis of results show that machine generated poems are able to capture the first-person perspective well, achieving between 4.5 and 5 average, although scores of the poem substance containing a dream or vision are questionable. Finally, the authors use Coh Metrix which is a text evaluation software kit to judge characteristics such as text readability, word imageability and concreteness, narrativity, referential cohesion and syntactic simplicity. As a result, the generated poems are easily readable by the majority of viewers. The models are also generating text that is concrete in word choice and that paint a picture, except for the dream model and the majority of models are producing poems that are both syntactically simplistic and narrative. The models lacks however in referential cohesion.

3 Methodology

To train a language model capable of generating stylistically distinct poems, we experiment with two types of poetic data: haikus and free verse poems. Haikus follow a strict structure, typically three lines with a 5-7-5 syllable pattern; while free verse allows for more flexibility in length, rhythm, and format. By fine-tuning our model on these two distinct forms, we aim to assess its ability to adapt to different poetic constraints and styles.

3.1 Datasets

Two main datasets were used to train and fine-tune the model:

- Haiku dataset: We use the publicly available Haiku dataset from Hugging Face, which contains 49,024 haikus in English. This dataset is particularly suited for training the model on structured poetic forms.
- Classic poems : Sourced from Kaggle, this dataset contains 13,754 classic English poems published on the Poetry Foundation website. It includes a variety of poetic forms and styles, making it ideal for training on free verse and longer poetic structures.

3.2 Model

For the generation component, we use GPT-2, a transformer-based language model trained on a large corpus of internet text. It offers a good trade-off between performance and accessibility: it is open source, lightweight compared to more recent alternatives like GPT-3 or GPT-4, and does not require access to proprietary APIs.

We fine-tune GPT-2 separately for each poetic style to better align the model with the structural and stylistic expectations of each format. The full training setups are summarized later in the methodology.

3.3 Poem generation pipeline

The poem generation process follows a two-step pipeline that combines vision and language:

1. **Image recognition with CLIP:** The system first takes an image as input. Using OpenAI's CLIP model, the image is compared against a predefined list of nature- and food-related labels stored in a JSON file. CLIP returns a similarity score for each label, and the one with the highest score is selected as the thematic prompt for the poem.
2. **Poem generation with GPT-2:** The selected label is then used as a prompt for the fine-tuned GPT-2 model, which generates a poem around that theme. The user can choose between different styles: haiku or classic, and the model adapts accordingly. For haikus, the label is simply used as a minimal seed (e.g. "moon") to preserve the short format. In contrast, for classic poems, we prepend a randomly selected phrase from a small set of introductory prompts (e.g., *"For I am the", "I only I could have the", "Then we see the"*) before appending the label. This helps guide the model toward a more narrative and lyrical poem. Each style also uses its own set generation parameters to better match the expected structure and tone.

3.4 Generation configuration

Parameters	**Haiku**	**Classic**
Max length	50	100
Temperature	0.5	0.7
Top-k sampling (k)	60	60
Top-p	0.9	0.9
Repetition penalty	1.2	1.5

Table 1: Generation parameters used for each poem type

These generation parameters were manually adjusted after a few rounds of experimentation to produce outputs that match the structure and tone of each poem type.

For haikus, we chose a lower `max_length` and `temperature` to limit the generated text to shorter, more structured results. The repetition penalty was set to a moderate value to allow some stylistic recurrence while avoiding repetitive loops.

On the other hand, for classic poems, a slightly higher `temperature` was used to introduce more lexical variation and the `repetition_penalty` was also increased to reduce redundancy in longer texts. The values for `top_k` and `top_p` are the same for both styles, as they result in a good balance between diversity and coherence.

3.5 Evaluation

We measure the text readability, syntactic simplicity (average elngth of sentences) , and word imageability (how possible it is to form a mental image of the word) and concreteness (how much the word relate to a palpable object) using MRC Psycholinguistic Database (Wilson, 1988). In terms of interpretation, for the syntactic simplicity, the lowest the score the better, and for the rest the highest the score the best.

4 Results and Discussion

Below are two examples generated by the system for the same theme, `"sun"`, to illustrate the differences in output style between the haiku and classic models.

Haiku:
The shape of the moon.
Sighs me outstretched.
Dreaming of you?

Classic:
I only I could have the sun. When it was dark my name came up again for a while but no one ever called me that when they said so (so what's wrong?). So now we're going to get out of here! It will be fine as long you don't touch us or any other things.

Scores	**Haiku**	**Classic**
Readability	0.48	0.62
Imageability	0.49	0.47
Concreteness	0.42	0.44
Syntactic simplicity	0.45	0.40

Table 2: Mean scores of generated text

We apply a min-max normalization to the scores, in general the generators have a margin of improvement. The generator for classical English poems best suit is the readability and syntactic simplicity. Even though, simple specific themes have been used for the generation, the concreteness and imageability could be better for both generators. However, since poems have a part of abstraction, these scores are not alarming. The generator for haiku is better at imageability and syntactic simplicity,

even though the nature itself of this form of poetry could be explain why the readability is on the lower hand.

5 Conclusion

This project aimed to fine-tuned a language model to generate theme-based poems, using an image for the theme. Two types of models, using as baseline GPT-2, have been fine-tuned: one to generate classical English poems and one to generate haikus. In general, the models are able to produce simple and readable poems, even though they lack concreteness and imageability. The main limit of the training is that only the content of poems have been exploited, so the models are not specifically tuned for specific themes, like in the literature. The reason behind is to give free range to the user. The datasets, especially for the classical poems, could also be enriched for better results.

Another limitation of this work is the lack of structural control during both training and generation. In many cases, the generated texts contain unfinished phrases, abrupt line breaks, or loosely connected ideas, regardless of the poem type. While this can sometimes be interpreted as stylistic freedom, it often results from the absence of syntactic or structural constraints in the model. For haikus, in particular, the model is not explicitly guided to follow the traditional 5-7-5 syllable structure, and no filtering was applied to enforce this pattern during preprocessing. Future work could explore structure-aware decoding strategies or line-by-line generation to better reflect formal poetic patterns.

References

[1] Beheitt, M. E., & Hmida, M. B. (2022). *Automatic Arabic Poem Generation with GPT-2.*

[2] Li, J., Tang, T., Zhao, W. X., & Wen, J.-R. (2022, May 13). *Pre-trained Language Models for Text Generation: A Survey.*

[3] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). *Language Models are Unsupervised Multitask Learners.*

[4] Schneider, P., Klettner, M., Simperl, E., & Matthes, F. (2024, February 2). *A Comparative Analysis of Conversational Large Language Models in Knowledge-Based Text Generation.*

[5] Wang, Q., Luo, T., Wang, D., & Xing, C. (2016). *Chinese Song Iambics Generation with Neural Attention-based Model.*

[6] Wilson, M. (1988). *MRC psycholinguistic database: Machine-usable dictionary, version 2.00.* Behavior research methods, instruments, & computers, 20(1), 6-10.