

The Mechanical Bard: An Interpretable Machine Learning Approach to Shakespearean Sonnet Generation

Edwin Agnew*, Michelle Qiu*, Lily Zhu*, Sam Wiseman, Cynthia Rudin

Duke University, Durham, NC

edwin.agnew@duke.edu, michelle.qiu@duke.edu, lily.zhu@duke.edu
swiseman@cs.duke.edu, cynthia@cs.duke.edu

Abstract

We consider the automated generation of sonnets, a poetic form constrained according to meter, rhyme scheme, and length. Sonnets generally also use rhetorical figures, expressive language, and a consistent theme or narrative. Our *constrained decoding* approach allows for the generation of sonnets within preset poetic constraints, while using a relatively modest neural backbone. Human evaluation confirms that our approach produces Shakespearean sonnets that resemble human-authored sonnets, and which adhere to the genre’s defined constraints and contain lyrical language and literary devices.

1 Introduction

We consider the task of automatically generating Shakespearean sonnets, a popular poetic form with highly specific rhyme and meter constraints¹. Each sonnet consists of three quatrains followed by a single couplet according to the rhyme scheme ABAB BCBC CDCD EE, and each line contains ten syllables with a stress pattern of iambic pentameter.

Rather than train a model to obey these constraints implicitly (which leads to enormous models that still do not obey the constraints), we opt to enforce them explicitly using a simple but novel approach to generation.

In particular, we use part-of-speech (POS) templates selected and edited from individual lines in Shakespeare’s sonnets, with each template intended to offer a different combination of parts of speech and narrative directions. Associated thematic words are then selected and placed at the end of each template, and their rhyming pairs are chosen dynamically by a language model (e.g., GPT-2, Radford et al., 2019) and placed at the end of the corresponding lines according to the rhyme scheme.

*denotes equal contribution

¹Our code is available at https://github.com/edwinagnew/Poetix_Sonnets

When all the lovers of this world are dead,
The sun of heaven on a golden day
To burn the earth’s fire by the flame and spread
Where all the flowers of your fair days lay.
These are the blossoms that you take care of.
Why do you linger such a long delay?
Forgive the fluttered flower of meek love
Or who you have so long to love the day?
The joys of love, the beauty on the face,
Shall be your fate and be your own delight.
You have the beauty of your own embrace.
You cannot reminiscence. Cannot write.
Between the living and the deadening breath.
You go the way of your beloved death.

Figure 1: A sonnet generated with the theme “death”.

The rest of the line is filled with related words that fit the specified POS and meter, leading to the end rhyme word. Figure 1 shows sample output.

Our use of these templates ensures sophisticated-seeming language and syntax that competing systems do not capture. Our approach provides excellent grammatical structure comparable to that of human-written poetry, all while using a relatively simple model and generation procedure.

We extensively evaluate the ability of our approach to generate *whole* sonnets (a setting often ignored by recent work in poetry generation) and find that our approach is preferred over strong baselines by both expert annotators (recruited from an academic English department) and by crowdworkers. As this research was conducted before the release of ChatGPT, we were not able to robustly compare our model’s performance against this language model. However, we make several observations about the poetic quality of sonnets generated by ChatGPT.

2 Related Work

Early attempts at poetry generation relied mainly on rule-based methods (Gervás, 2000; Oliveira,

2012; Manurung et al., 2000; Veale, 2013). More recent automated poetry generation techniques, especially for sonnet generation, have relied on combinations of task-specific language models and rules. For instance, Ghazvininejad et al. (2016)’s Hafez uses a finite state acceptor to generate a large number of possible lines, the best of which are then selected with an RNN trained on song lyrics. Like our approach, they use rhyming dictionaries to find rhyming words and word embeddings to find topical words. Similarly, Benhardt et al. (2018) preselects rhyming words and generates lines backwards with a recurrent neural network (RNN). Also in this vein are Lau et al. (2018)’s Deepspare, which consists of an LSTM language model, an iambic model, and a rhyming model, and the recent work of Van de Cruys (2020) and Wang et al. (2021).

Our approach distinguishes itself in using a general-purpose pretrained language model, but more importantly in its use of human-curated constraints and templates. These allow for generating high-quality poems with a very simple approach.

3 Methodology

The general idea of our approach is to take a pretrained language model (in this case GPT-2) and apply hard constraints to the generation procedure so that it can only output text satisfying various poetic constraints. These constraints can be broadly divided into *hard* constraints (e.g., number of syllables) and *soft* constraints (e.g., sounding poetic), and our methodology can be separated similarly. Our generation process is in Figure 2.

3.1 POS Templates

The most important part of our method is the use of handcrafted grammar templates. Taking inspiration from existing sonnets, we created a list of about 120 templates that encode the part-of-speech structure of a line of poetry. Each template can generate an unbounded number of possible poetic lines. For example, the line “The beauty of life on a lonely sea” is represented by the template “THE NN OF NN ON A JJ NN.” More sample templates are in Section A.1. Since the templates allow for considerable flexibility, obeying the templates does not alone suffice for poetry. For example, the same template could be used to write poetic lines with distinct meanings such as “The tree of anguish on a stormy night” or a nonsensical line like “The fork of ant on an unpacked transfer.” A subset of these

templates is also chosen for starting a stanza.

3.2 Strict Sonnet Constraints

The two most critical features of sonnets distinguishing them from other poetry forms are that they are written in iambic pentameter (i.e., each line has 10 syllables of alternating stress pattern), and they follow an ABAB CDCD EFEF GG rhyme scheme. To detect iambic pentameter, we use the CMU Pronouncing Dictionary (CMU, 2019), which reveals how many syllables a word contains and the stress of each syllable. An unstressed syllable is represented as ‘0’ and a stressed syllable as ‘1’, and so the line “The beauty of life on a lonely sea” is represented as ‘0 10 1 0 1 0 10 1’. For simplicity, 1-syllable words can be designated as either 0 or 1.

Given a POS-tag for every word in our dictionary, we create a tree-like data structure that represents every possible meter for a given template. Continuing the example, the first word could only be ‘the’, but the second word could be filled with a 1-syllable noun like ‘tree’, a 2-syllable noun like ‘chaos’ (10), or a 3-syllable noun like ‘audio’ (101), and so on. Each choice affects the possible pronunciations of the next word as well as the number of remaining words in order to reach 10 syllables. The pronunciation dictionary ensures the last syllable of the last word on each line matches its partner.

3.3 Language Model

We use a language model to generate individual sonnet lines, subject to the formal constraints outlined above. In particular, we first fine-tune GPT-2 (Radford et al., 2019) on a large corpus of over 15000 poems² and a smaller corpus of sonnets³. We then use a constrained beam-search to generate each line, where only legal tokens (under the aforementioned constraints) can be generated at each step; this generation approach resembles previous constrained decoding techniques used in sonnet generation (Ghazvininejad et al., 2016), although our approach differs in the choice of model and direct enforcement of constraints. For a comparison of generation quality using a GPT-2 model that has not been fine-tuned, see Section 4.1.

3.4 Thematic Word Choice

To ensure the content of the poem fits the theme specified by the user, we provide an excerpt of a

²<https://www.kaggle.com/datasets/johnhallman/complete-poetryfoundationorg-dataset>

³<https://www.kaggle.com/datasets/michelleqiu/sonnets>

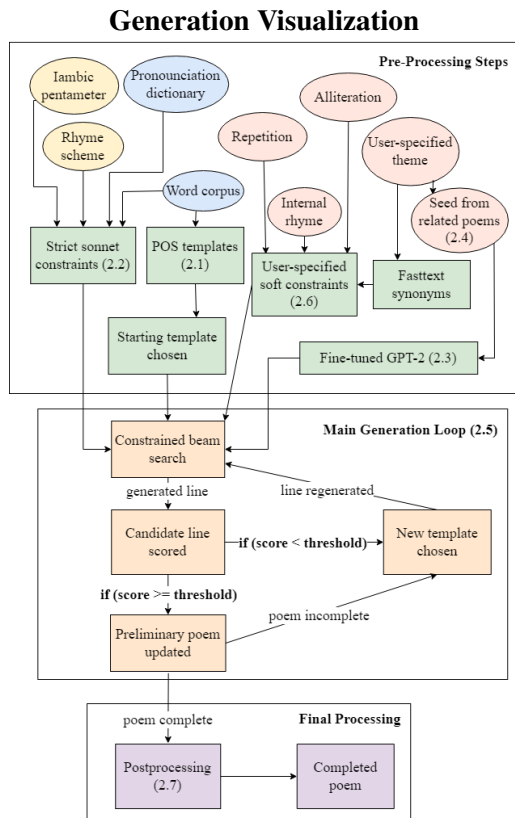


Figure 2: Numbers in parentheses denote subsections in Section 3.

theme-appropriate poem as additional context to GPT-2 during generation. This additional poem is selected by finding a list of synonyms to the theme word using the WordNet synonym database (Miller, 1998) and then choosing lines from a poem corpus that contain at least one synonym. We also remove words from the vocabulary if they have less than 0.5 cosine similarity with the theme word, based on the corresponding FastText word embeddings (Bojanowski et al., 2017). This avoids having words like “algebra” in poems with themes like “forest.”

3.5 Generation Procedure

Having introduced our method’s components, we now describe the generation procedure. A user inputs a theme word, a beam search parameter, b , and the number of templates sampled per line, k . A seed is chosen with the above method. Then for each line, we sample k random templates. For each template, we generate the line using a modified beam search. Specifically, the beam search maintains b different hypotheses per line at all times. For each hypothesis, we first mask out any tokens that violate our hard POS, meter, or rhyme constraints and select the b best next-tokens for each

of the k templates. These b^2 new candidates are re-ranked according to our custom scoring function, and the top $k \times b$ proceed to the next stage. The constraint-filtering at each stage guarantees that the generated line will match the input template, while the beam search allows more flexible word choice than greedy word-filling for each POS. If none of the $k \times b$ generated lines score better than a specific threshold, then a new template is chosen and the line is generated again. Otherwise, line generation continues until the poem is completed.

3.6 Poetic Devices

To make the poems more poetic, we adjust our scoring function to weight lines with alliteration, penalties for repetition, and/or internal rhyme. Alliteration occurs when a line contains words starting with the same letter, repetition occurs when a word is present several times throughout a poem, and internal rhyme occurs when two words rhyme within the same line. To weight alliteration, when the first token of a new word is being generated, a list $\vec{A} = [a_1, a_2, \dots, a_n]$ is generated where a_i is the number of occurrences of the first letter of the i th token in the current line. To weight and discourage repetition, a list $\vec{T} = [t_1, t_2, \dots, t_n]$ is generated where t_i is the number of occurrences of the i th token in the poem, negated. To weight internal rhyme, a list $\vec{R} = [r_1, r_2, \dots, r_n]$ is generated where $r_i = 1$ if the i th token is part of a word that rhymes with any of the words in the current line generated so far, and $r_i = 0$ otherwise. The final token distribution is then proportional to $\tilde{P} + \alpha_A \times \vec{A} + \alpha_T \times \vec{T} + \alpha_R \times \vec{R}$, where \tilde{P} is the language model’s next-token distribution, and α_A , α_T , and α_R are user-specified non-negative parameters, which represent the degree to which alliteration, repetition, and internal rhyme should be favored during generation.

3.7 Postprocessing

After a poem is completed and all 14 lines score above a fixed threshold, a small number of adjustments are made. These include fixing common mistakes made by GPT-2 like not capitalizing the word ‘I’ and not capitalizing following punctuation.

4 Experiments

We used human input to test our sonnets against both model-generated and human-written sonnets. To test adherence to a theme throughout a son-

| Expert Evaluation | | |
|---------------------|-------|---|
| Category | Mean | p-value |
| PoeTryMe | | |
| Grammar | 4.50* | <u>1.71×10^{-4}</u> |
| Emotion | 4.30* | <u>3.13×10^{-3}</u> |
| Poetic | 4.30* | <u>3.13×10^{-3}</u> |
| Human | 4.10* | <u>5.77×10^{-3}</u> |
| Theme | 2.60 | 0.211286 |
| Benhardt et al. | | |
| Grammar | 3.83* | <u>0.03</u> |
| Emotion | 3.67* | <u>0.05</u> |
| Poetic | 3.75* | <u>0.04</u> |
| Human | 3.75* | <u>0.02</u> |
| Theme | 2.42 | 0.06 |
| Human-written poems | | |
| Grammar | 1.36 | 1.00×10^{-6} |
| Emotion | 1.4 | 5.00×10^{-6} |
| Poetic | 1.64 | 5.40×10^{-5} |
| Human | 1.36 | 1.00×10^{-6} |
| Theme | 1.57 | 7.70×10^{-5} |

Figure 3: Starred figures indicate average scores of > 3 , and underlined figures indicate that the p-value is low enough (< 0.05) to claim that this higher average is statistically significant.

net, we desired baselines that generate whole sonnets with user-provided themes. This limited our competitors, as some generate discrete quatrains or generate without input themes (e.g., DeepSpear), leaving only Benhardt et al. (2018) and PoeTryMe (Oliveira, 2012); see Section A.2.

Furthermore, an evaluation of poetry quality is incomplete without human-written sonnets, selected from sonnets.org. Though these poems do not have an explicit theme, we selected poems that followed our five themes.

To optimally test our model, we conducted an internal analysis and selected k values sampled from 3, 5, or 7, b values sampled from 3, 5, or 7, and repetition penalty values sampled from 1.4, 1.6, or 1.8 that we concluded produced the highest quality sonnets. To evaluate adherence to theme, we generated poems with themes “death,” “darkness,” “forest,” “love,” and “wisdom.”

In each test, respondents compared six randomly selected pairs of sonnets, with each of our sonnets displayed with a competing model/human-written sonnet generated with the same theme word. Respondents indicated which of the two sonnets performed better in categories of theme, poeticness, grammar, emotion, and likelihood of being human-written. Detailed instructions are in A.3.

| Amazon MTurk Evaluation | | |
|-------------------------|-------|---|
| Category | Mean | p-value |
| PoeTryMe | | |
| Grammar | 3.66* | <u>2.00×10^{-6}</u> |
| Emotion | 3.54* | <u>1.16×10^{-4}</u> |
| Poetic | 3.55* | <u>3.70×10^{-5}</u> |
| Human | 3.59* | <u>1.60×10^{-5}</u> |
| Theme | 2.86 | 0.19 |
| Benhardt et al. | | |
| Grammar | 3.34* | <u>6.57×10^{-3}</u> |
| Emotion | 3.16* | 0.12 |
| Poetic | 3.11* | 0.19 |
| Human | 3.06* | 0.33 |
| Theme | 2.77 | 0.06 |
| Human-written poems | | |
| Grammar | 3.13* | 0.14 |
| Emotion | 2.86 | 0.14 |
| Poetic | 2.91 | 0.24 |
| Human | 2.92 | 0.27 |
| Theme | 2.67 | 0.02 |

Figure 4: Starred figures indicate average scores of > 3 , and underlined figures indicate that the p-value is low enough (< 0.05) to claim that this higher average is statistically significant.

4.1 Expert Evaluation

For an expert evaluation, we recruited six faculty members and students from an academic English department. Figures 3 and 5 show that we strongly outperform PoeTryMe in all categories but theme with high statistical significance ($p < 0.006$), and we outperform Benhardt et al. in all poetic categories but theme and emotion with statistical significance ($p < 0.05$). Notably, while we outperform other computer-generated poems, respondents could still distinguish between our poems and human-written sonnets quite easily. See more in A.4.

4.2 Amazon MTurk Evaluation

Along with expert evaluation, we used Amazon MTurk services to assess poems on a larger scale. Figures 4 and 6 show our superior performance against competitors in several categories. As expected of most computer-generated work, our poems failed to outperform human-written poems. However, we can only strongly conclude that the human-written poems are better in one category, theme. Our poems even outperformed human-written poems in grammar (albeit with low statistical significance), showing that our strictly constrained beam search generates high quality grammar. See more in A.5.

Expert Evaluation

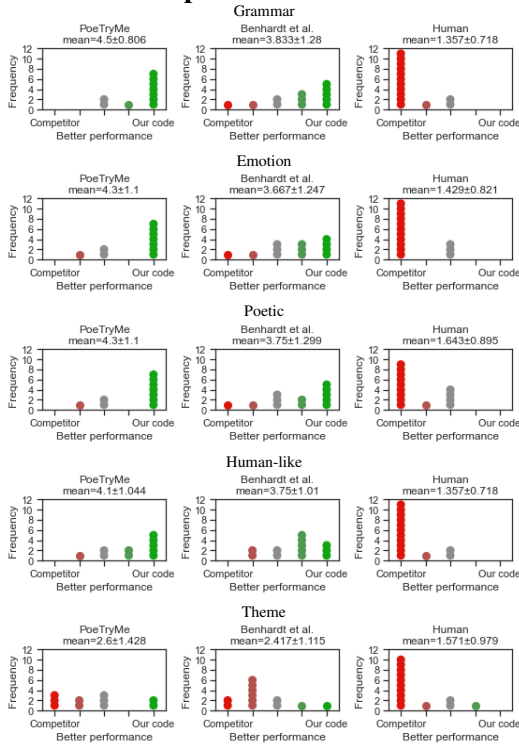


Figure 5: Values > 3 (green), < 3 (red), and = 3 (gray) denote that our poetry model performs better, the competitor performs better, and the poems performed similarly, respectively.

4.3 Ablative Evaluation

We also conducted ablative studies showing the efficacy of two key elements of our method: line templates and the fine-tuned GPT-2 language model. We generated two sets of ablative poems: one with the fine-tuned GPT-2 and no templating, and one using the untrained GPT-2 model and templating. We then used Amazon MTurk services to test each set against poems generated with both factors under the same criteria as previous experiments. From Figure 11, it is the combination of the fine-tuned model and templating that ensures higher quality sonnets than if only one factor is implemented. Our poems with both factors outperform both sets of ablative poems with varying statistical significance. Specifically, providing templates is clearly the critical piece to generate poems of a high caliber. See more in A.6.

5 Conclusion

We propose a novel method for generating high-quality poems that uses POS templating to determine a logical syntactical structure and rigorously

Amazon MTurk Evaluation

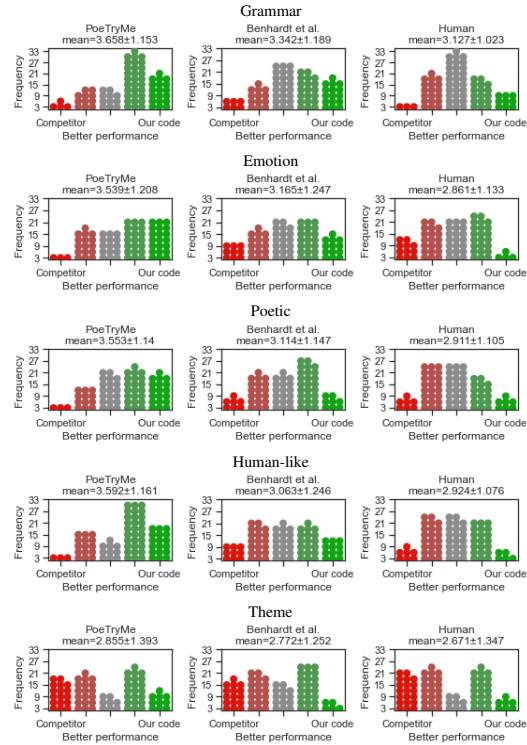


Figure 6: Values > 3 (green), < 3 (red), and = 3 (gray) denote that our poetry model performs better, the competitor performs better, and the poems performed similarly, respectively.

maintains constraints necessary for any sonnet. Our method is highly versatile, with poetic factors like alliteration, internal rhyme, repetition, and theme adjustable to ensure creative output. After extensive surveys conducted with expert evaluators and MTurk participants, our model’s success over similar competitors is evident, though our model’s poems, like those of most computer poetry generators, remain distinguishable from human written poems.

While we were unable to compare our model’s performance to that of ChatGPT, our finetuned GPT-2 requires far less computing power than subsequent GPT models. Additionally, while we commenced this project’s evaluation prior to the release of ChatGPT, after a preliminary qualitative evaluation, ChatGPT seems to produce very generic poetry (see A.7). Thus, for this particular application, our model may be a viable method that is more cost-effective and produces relatively high-quality sonnets.

Limitations

Though our method produces full sonnets that are more impressive than all previous approaches, it

is still not at the level of human-generated poetry. It is not clear how to achieve this level, whether it would be using massive large language models, or through our general approach, which is to bend those models around an interpretable framework that knows the rules that sonnets obey. Certainly our approach requires a lot less data – even if one used all the sonnets that have ever been written to train a language model, it is unclear that the language model would learn the very specific rules required of sonnets. However, there may be other ways to obtain these constraints that have not yet been developed.

Ethics Statement

As with all neural generation, there are concerns about misinformation and generating toxic text. These concerns apply to some degree to poetry generation, although our rigidly constrained approach and limited vocabulary should mitigate this.

References

- John Benhardt, Peter Hase, Liuyi Zhu, and Cynthia Rudin. 2018. [Shall I compare thee to a machine-written sonnet? An approach to algorithmic sonnet generation.](#)
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146.
- Carnegie Mellon University CMU. 2019. [The CMU pronouncing dictionary.](http://www.speech.cs.cmu.edu/cgi-bin/cmudict) <http://www.speech.cs.cmu.edu/cgi-bin/cmudict>, Internet.
- Pablo Gervás. 2000. Wasp: Evaluation of different strategies for the automatic generation of spanish verse. In *Proceedings of the AISB-00 Symposium on Creative & Cultural Aspects of AI*, pages 93–100.
- Marjan Ghazvininejad, Xing Shi, Yejin Choi, and Kevin Knight. 2016. Generating topical poetry. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1183–1191.
- Jey Han Lau, Trevor Cohn, Timothy Baldwin, Julian Brooke, and Adam Hammond. 2018. [Deep-speare: A joint neural model of poetic language, meter and rhyme.](#) In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1948–1958, Melbourne, Australia. Association for Computational Linguistics.
- Ruli Manurung, Graeme Ritchie, and Henry Thompson. 2000. Towards a computational model of poetry generation. <https://era.ed.ac.uk/handle/1842/3460>.
- George A Miller. 1998. *WordNet: An electronic lexical database.* MIT press.
- Hugo Gonçalo Oliveira. 2012. Poetryme: a versatile platform for poetry generation. *Computational Creativity, Concept Invention, and General Intelligence*, 1:21.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. <https://github.com/openai/gpt-2>.
- Tim Van de Cruys. 2020. [Automatic poetry generation from prosaic text.](#) In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2471–2480, Online. Association for Computational Linguistics.
- Tony Veale. 2013. [Less rhyme, more reason: Knowledge-based poetry generation with feeling, insight and wit.](#) In *Proceedings of the Fourth International Conference on Computational Creativity, ICCO 2013, Sidney, Australia, June 12-14, 2013*, pages 152–159. computationalcreativity.net.
- Jianyou Wang, Xiaoxuan Zhang, Yuren Zhou, Christopher Suh, and Cynthia Rudin. 2021. [There once was a really bad poet, it was automated but you didn't know it.](#)

A Appendix

A.1 Templating Mechanism

Figure 8 presents more examples of our templating mechanism. We combine an adapted version of the Penn Treebank Project’s part of speech tags along with articles, conjunctions, prepositions, and other filler words to construct these templates. Additionally, we provide the stress pattern of the syllables to ensure that the constraint of iambic pentameter is met. However, outside of the pre-determined filler words, POS do not have to directly adhere to the given stress pattern in splitting up words. For instance, in the first template, the provided syllable stress indicates that the JJ tag (adjective) should have two syllables, while the final VB tag (verb) should have only one syllable. However, the generated line ends with a monosyllabic adjective and a bisyllabic verb. As long as the stressing of the syllables aligns properly, each word can vary in its number of syllables. This is also visible in the fourth template example in Figure 8.

A.2 Elaboration on Experimental Competitors

[Benhardt et al. \(2018\)](#), referred to as Benhardt et al., uses a RNN to preselect rhyming words and

restrict different parts of speech to fit within the sonnet format. Oliveira (2012), referred to as CoPoetryMe, is a versatile platform using semantic and grammar templates to alter the type of poem, input words, and “surprise” factor generated.

A.3 Experimental Procedure

For each pair of sonnets, respondents were asked to indicate whether Sonnet A or Sonnet B performed better based on factors such as adherence to the inputted theme, poeticness, grammatical correctness, ability to convey emotion, and likelihood of being written by a human. Available answer choices and their corresponding numeric scores from 1 to 5 were “Definitely A” (5), “Probably A” (4), “The same” (3), “Probably B” (2), and “Definitely B” (1). Both our sonnet and the competing model-human-sonnet had equal probability of being either sonnet A or sonnet B in each pair. To analyze this data, user inputs were translated into numeric scoring values corresponding to our model’s sonnet being Sonnet A (i.e. if our sonnet is presented as B to the user, a response of “Definitely B” corresponds to a score of 5, “Probably B” corresponds to 4, “Probably A” corresponds to 2, and “Definitely A” corresponds to 1). Additionally, respondents were asked to answer sanity check questions to filter out respondents who answer illogically or who do not have a sufficient grasp of English grammar. This setup remained the same across all experiments, and an additional space was allocated for expert evaluators to leave qualitative comments on sonnet quality. Sample sonnet evaluation questions are visible in Figure 9.

After calculating the mean and standard deviation for scores across sonnets, we can immediately see whether our model performed better (an average score of > 3) or worse (an average score of < 3) than the competitor in each respective category. We then performed a series of t-tests to establish these results’ statistical significance. For factors that indicated our model performed better, we performed a right-tailed t-test (with the null-hypothesis as our model performed worse than the baseline), and we performed a left-tailed t-test for the remaining factors (with the null-hypothesis as our model performed better than the baseline).

A.4 Expert Evaluation Analysis

In the expert evaluation, we emailed faculty at an American academic English department to recruit six faculty members and students to take our survey

without payment. While we showed strong performance against the other computer-generated poems, we are consistently outperformed by human-written poems in all categories. Weaker performance on theme in experimental results may be explained by competitors’ more frequent inclusion of the user-inputted theme word. For instance, in the expert evaluation, between two poems generated with the theme word “forest” (see Figure 10), one survey respondent states, “Sonnet B repeats forest too much for my taste,” subsequently giving our model a 5 in each of poeticness, grammar, emotion, and humanness, yet a 2 in theme.

A.5 Amazon MTurk Analysis

In our evaluation using Amazon MTurk Services, we requested survey respondents from primarily English-speaking countries and with an approval rate of $\geq 95\%$. Crowdworkers were paid through the Amazon MTurk platform for this survey that on average took less than 30 minutes to complete. The questions and formatting remained the same as the expert evaluation, except no space was provided for qualitative feedback.

Based on Figure 4 there is enough statistical significance to conclude that our sonnets outperform PoeTryMe in poetic, grammar, emotion, and human categories ($p < 0.001$). Against Benhardt et al., there is enough statistical significance to conclude that our sonnets perform better in grammar ($p < 0.001$), and perform slightly better with weak statistical significance in emotion ($p < 0.15$). Against human-written sonnets, the p-values for poetic, emotion, and even human categories are too large to strongly reject the null hypothesis that our model performed better than the baseline. Additionally, while the p-value indicates that this value is not statistically significant, it is interesting to note that our poems on average scored better in the grammar category.

A.6 Ablation Analysis

In our ablation analysis, we replicate the Amazon MTurk analysis yet replace the competitor/human-written sonnets with poems generated with either the fine-tuned GPT-2 model without templating or the GPT-2 model without fine-tuning and with templating. This lets us test the individual efficacy of each factor (templating and fine-tuning GPT-2) against our method implementing both. Against poems generated with the fine-tuned GPT-2 and no templating, our sonnets performed better across

all categories, and we can strongly reject the null hypothesis that our model performed worse than the baseline ($p < 0.0001$). Against the poems generated with the GPT-2 model without fine-tuning and with templates, we can conclude with high statistical significance ($p < 0.01$) that we performed better in emotion, and conclude with weak statistical significance ($p < 0.10$) that we performed better in grammar and theme. These results indicate that our method is successful due to its usage of both the fine-tuned GPT-2 model and templating.

A.7 ChatGPT Qualitative Analysis

While we did not have time to extensively evaluate the quality of our sonnets against those of ChatGPT, after generating several sonnets to test ChatGPT's sonnet quality, it seems as though this language model generates relatively generic, non-cohesive sonnets even with different parameters. For instance, in Figure 7, both of the sonnets are unable to cohesively connect these three topics along a reasonable storyline. Additionally, Sonnet A in particular seems to dedicate a single stanza to each of the three topics passed in, hardly attempting to connect them. Of course, with more intensive prompt engineering, it is possible to generate a sonnet more tailored to one's preference. However, even this short analysis demonstrates there are clearly still strides to be made in the field of automatic poetry generation even with the advent of ChatGPT.

| Sonnet A |
|---|
| <p>In realms of words, where novels find their reign, My friend, a skeptic, shuns the written tales. No pages turned, no fictional campaigns, Averse to narratives that time unveils. Yet in his hand, a pear's sweet succulence, A juicy feast, a taste of nature's grace. Each bite, a moment of pure indulgence, A palate's joy, a respite from life's race. But beyond fruit's delight, his spirit soars, Within the bounds of a duelist's domain. With sword in hand, he dances, parries, scores, A noble art where honor's stance remains. Though novels may elude his curious gaze, Through pears and fencing, passions find their ways.</p> |
| Sonnet B |
| <p>In realms of love, where rom-coms dance their waltz, My friend, a rebel, strays from silver screens. With flannel-clad allure, he stands, exalts, A soul untouched by scripted love's routines. No tears shed for the tales of starry-eyed, His heart resists the charms of love's façade. Yet stitches formed with yarn, his hands provide, Creating warmth, affection neatly clad. Through threads of patience, vibrant patterns bloom, His nimble fingers weave a tapestry. Each crochet stitch, a tale of friendship's loom, A testament to artistry set free. Though rom-coms fall short of his embrace, In flannel and crochet, love finds its place.</p> |

Figure 7: Comparison of two sonnets generated with ChatGPT. Sonnet A was generated with the prompt "generate a sonnet about my friend who hates novels, eats pears, and fences" and Sonnet B was generated with the prompt "generate a sonnet about my friend who hates rom-coms, wears flannels, and crochets."

| Template | Syllable Stress | Example Line |
|--|---------------------|--|
| Where all the NNS of PRPD\$ JJ NNS VB. | 0 1 0 1 0 1 0 1 0 1 | “Where all the gods of their past lives dictate” |
| And it VBD ABNN to the NN | 0 1 0 10 1 0 10 1 | “And it seemed evil to the enterprise” |
| Between the VBG and the VBG NN | 0 1 0 10 1 0 10 1 | “Between the glistening and the dying muse” |
| A JJ NN from the JJ NN | 0 10 10 1 0 1 0 1 | “A little lightness from the earthy sky” |
| Upon PRPO, PRPD\$ NN POS NN | 0 1 0 1 0 10 10 1 | “Upon you, your life’s possibility” |
| Why VBC PRPS VBG such a JJ NN? | 0 1 0 10 1 0 10 1 0 | “Why do you squander such a precious thing?” |
| The NNS of ABNN, the NN on the NN | 0 1 0 1 0 10 1 0 1 | “The ghosts of death, the spirit on the earth” |

Figure 8: Template examples, their corresponding syllable stress in order to adhere to iambic pentameter, and a sample line generated using the template.

The key word for both of these poems is "wisdom." Which poem best adheres to this theme?

Definitely A Probably A Same Probably B Definitely B

Which poem sounds more poetic?

Definitely A Probably A Same Probably B Definitely B

Which poem is more grammatically correct?

Definitely A Probably A Same Probably B Definitely B

Which poem conveys emotions more effectively?

Definitely A Probably A Same Probably B Definitely B

Which poem is more likely to be written by a human?

Definitely A Probably A Same Probably B Definitely B

Figure 9: Survey questions presented for each pair of sonnets.

Sonnnet A: Our Code

I was aghast to see the fireflies
Inflamed soothed toads, where there the dead boughs lay
And it seemed evil to the enterprise
The hag I had, the hag, the hog, the gray.
But I knew to my painless fireflies
And beauty was a kind and loving thing.
My life’s light isle so longed on otherwise
So too my fireflies bloomed to my king.
Those eagles that with auburn hair flew oaks,
Beauty and beauty beamed within the air
Which made oasis overcomes to coax?
So too my hogs beheaded to my lair.
The windy night was in the mistletoe
And wept soiled toads in my dream’s studio.

Sonnnet B: PoetryMe

forest some more and reforest a trip!
in deserts where heavenly woodlands clink
many, many, many clustered before
come: not in establishments of the floor
the fields of agony, the endless circumstance
findings to lies to interrupt your earth
with summation and set, triumph and agony
floors of horror forest before my eyes
those that study clustered plant are psychologists
taking over my ness a second forest
an’ you’ve got to forest them reforest
on every forest, indeed, that rainforests
and grounds of forest coming to accord
floor of establishments and lilt of sing

Figure 10: Comparison of two sonnets generated with theme word “forest”. Sonnet A was generated with our code, and Sonnet B was generated using PoeTryMe.

| Ablation Evaluation | | | | |
|----------------------------|-------|---|-------|---|
| Category | Mean | p-value | Mean | p-value |
| Grammar | 3.51* | <u>5.10×10^{-5}</u> | 3.21* | 0.06 |
| Emotion | 3.61* | <u>9.00×10^{-6}</u> | 3.40* | <u>3.89×10^{-3}</u> |
| Poetic | 3.61* | <u>4.00×10^{-6}</u> | 3.09* | 0.29 |
| Human | 3.66* | <u>1.00×10^{-6}</u> | 3.01* | 0.46 |
| Theme | 3.50* | <u>8.00×10^{-5}</u> | 3.20* | 0.06 |

Figure 11: Left: fine-tuned GPT-2 with no templates. Right: GPT-2 without fine-tuning, but with templates. Starred figures indicate average scores of > 3 , and underlined figures indicate that the p-value is low enough (< 0.05) to claim that this higher average is statistically significant.

ACL 2023 Responsible NLP Checklist

A For every submission:

- A1. Did you describe the limitations of your work?
Limitations
- A2. Did you discuss any potential risks of your work?
Ethics
- A3. Do the abstract and introduction summarize the paper's main claims?
Abstract, 1
- A4. Have you used AI writing assistants when working on this paper?
Left blank.

B Did you use or create scientific artifacts?

3,4

- B1. Did you cite the creators of artifacts you used?
3.2,3.3,References
- B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.
- B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.
- B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Data used from publicly available sonnets/poems were assumed to be not subject to dispute.
- B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.
- B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
3.3

C Did you run computational experiments?

Left blank.

- C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
No response.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

- C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?

No response.

- C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?

No response.

- C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?

No response.

D Did you use human annotators (e.g., crowdworkers) or research with human participants?

4.4.1,4.2,4.3

- D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?

Appendix

- D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?

A.5,A.6

- D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?

We do not believe having data on poetry evaluation raises any ethical issues.

- D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?

We do not believe having crowdworkers evaluate the same poems that were given to English professors raises any ethical issues.

- D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?

A.6