Stylized Sequence Generation with Structured Constraints: Utilizing HMMs and LSTMs to Recreate Shakespearean Poetry

Kieran Vlahakis, Noor Ibrahim, Anirudh Gajula, Sujay Champati

Abstract

We present an explorative study of sequence generation models for constrained natural language generation in the stylized domain of Shakespearean sonnets. Leveraging both classical probabilistic modeling and modern neural architectures, we evaluate Hidden Markov Models (HMMs) trained via the Baum-Welch algorithm alongside character-level Long Short-Term Memory (LSTM) networks. We show that while HMMs effectively encode poetic structure through probabilistic control over syllable count and rhyme constraints, LSTMs offer higher local coherence and creative flexibility at the cost of structural discipline. Our contributions include a hybrid rhyme enforcement algorithm based on phonetic dictionaries, a systematic analysis of meter and stress pattern violation, and a broad interpretation of generative limitations arising from hard-form constraints.

1 Introduction

Structured text generation presents a unique challenge: balancing formal constraints with linguistic fluency. Shakespearean sonnets are a canonical benchmark for this task due to their rigid form—14 lines of iambic pentameter with a strict rhyme scheme (ABAB CDCD EFEF GG). We explore two generative paradigms: the Hidden Markov Model, which offers probabilistic interpretability and efficient constraint enforcement, and the LSTM, which excels at modeling long-range dependencies and stylistic continuity.

2 Dataset and Preprocessing

We used the full corpus of Shakespeare's 154 sonnets. Text was cleaned by removing numbers, redundant punctuation (except apostrophes), and normalizing whitespace. Tokenization was done at the word and character level, depending on model requirements. To enforce structure, we annotated syllable counts and rhyme classes using the CMU Pronouncing Dictionary.

3 Hidden Markov Model

We trained HMMs using the Baum-Welch algorithm. Tokens were encoded via a label encoder, and the emission and transition matrices were updated iteratively. To impose poetic structure, we decoupled generation into line-wise segments, enforcing 10-syllable constraints through syllable-count aware sampling, and rhyme via a phoneme-based suffix matcher.

Rhyme

The biggest challenges regarding poem generation were encountered in enforcing the rhyme scheme. We began our task by implementing a basic HMM poem generation framework that could only select rhyming couplets through suffix-based matching[1]. This was somewhat helpful for incorporating rhyme in our poems; however, we ran into issues where words that simply shared the same ending were selected as rhyming couplets despite having different phonetic representations; thus, words like **"with"** and **"both"** would be matched because both words ended with **"th"**. Another issue was that because we were only using limited words to choose from sometimes we could not even find suffix-based matches and had to resort to using any random word to complete the couplets; thus, words like **"ny"** which end with **"ny"** would typically not be matched properly in poems (see below how **"any"** is matched with **"twilight"**). We were able to address this issue by utilizing the CMU Pronouncing Dictionary to extract phonetic representations of words and identifying shared rhyming structures

Example Using Suffix-Based Matching

```
----------------------------------------------------------------
I grind whereon wolf beside and any
not mistress votary of my dear pen
ride me youthful out composed twilight
thy thy what well men's attaint this i when

I wherefore as as his war he on but
of three love that divided i do grew
red thy bounteous sweet thoughts thy deep without
sweet her sweet truth sweet i thou all review

And thou the friend to of dead and my with
say words made fickle doth exceeded sins
beweep is bright lean blind do the grow both
is paper i'll in to may check with age

Thou sweet some though where thither is within
what ripe can the do pass all been therein
----------------------------------------------------------------
```

Example Using Phonetic Matching

```
----------------------------------------------------------------
That such of frowns my which knife were love dead,
Are past should huge thy love showers doth by of,
```

[1]using only words that were made available through in-generated emissions from our model

```
Faces will eye's to the yet kings sweet red,
The is children lips flatterer please love,

Not of awards that set where health to my,
When stays effect sickness dear alchemy,
I fault night with one both from excuse by,
With skill are true I reason doth body,

For this strange whose vile will youngly so own,
Windows and with beated lived alas she,
Love I thee down are not my doom lends stone,
Spend yet are by remember since now free,

Re haply survey unmoved my say,
To my absent his by love hope bare they.
----------------------------------------------------------------
```

Explanation of How Structure was Enforced

The rhyming accuracy was improved by specifically, incorporating a function that isolated the last stressed vowel and subsequent phonemes to establish rhyming groups. The model then selected words that share these phonetic structures, ensuring that end-line words adhered to the Shakespearean rhyme scheme (ABAB CDCD EFEF GG). Additionally, another function was implemented to ensure that pairs of words conformed to these phonetic constraints before being placed into the appropriate rhyming positions in the sonnet.

Correct syllable counts were achieved via the CMU dictionary. Using the dictionary we were able to retrieve the number of syllables per word. Implemented methods could then determine the appropriate syllable count for each word, accounting for line-ending variations where applicable. The model would then generate lines that would aim for a target syllable count (10 per line in a Shakespearean sonnet) by summing the syllables of candidate words. Care was taken to ensure that each generated line adhered to iambic pentameter by iterating through words, retrieving their syllable counts, and adjusting the line length accordingly. Through iterative selection and refinement, the model was able to maintain better metrical consistency across the poem.

Analysis of Poems and Additional Improvements

While our approach successfully generates structurally consistent Shakespearean sonnets, it falls short in terms of semantic coherence and metrical precision. The generated verses, while fluent and stylistically appropriate, often lack logical progression and thematic depth. This issue arises because our model prioritizes formal constraints—such as rhyme and syllable count—over deeper narrative cohesion. Furthermore, although the model adheres to a 10-syllable constraint, it does not fully capture the rhythmic nuances of iambic pentameter, leading to irregular stress patterns that disrupt the intended cadence.

To improve these aspects, one potential strategy would be to introduce semi-supervised learning by manually labeling some sonnets with specific thematic states, allowing the model to better capture narrative progression. Additionally, exploring higher-order Hidden Markov Models (e.g., second-order HMMs) could enhance sequence modeling by incorporating broader context in state transitions, potentially improving both coherence and rhythmic fidelity. These refinements could help the model generate sonnets that not only conform to formal constraints but also exhibit stronger thematic and metrical consistency

4 Recurrent Neural Network (LSTM)

We attempted to generate poems using an LSTM. Our model consisted of three layers with 150 hidden units each. Characters were embedded in a 128-dimensional space. We trained the model to predict the next character in 40-character sequences using categorical cross-entropy. Text was sampled at various temperatures (0.25 to 2.0) to assess the creativity-coherence tradeoff.

Qualitative Analysis of the Effect of Temperature

The poems generated by the model exhibit varying degrees of structure and coherence based on the temperature used during generation.

Temperature: 0.25

```
----------------------------------------------------------------
Shall I compare thee to a summer's day?
Thou art more love thee that thou mayst call,
Whilst I (my self a worther than thy self dost light?
Be thou art born to the lives and speak of the star

To the world will be thy self a worthless song,
Darkening thee wit the world will be thy self a same,
And by a part of the sun is so show,
And they thou mayst be so true, no shall excuse my love excuse the strong,

When I behold is so strong such a sad say,
And they thou art comemonned and the most of self-doing cross;
To say they have I seen the fairest once side,
And by their pride back to thee and stay

That bear the lovely on thee,
That in their stars in the stars in thee thy self alone,
----------------------------------------------------------------
```

Temperature: 0.75

```
----------------------------------------------------------------
Shall I compare thee to a summer’s day?
Thou art more nature’s changing stars thy worth the treasure? O shall my self art,
And I am thy beauty decay,
And sor my grace they with that doth with thy self alone.

But what in war as thou being mourners, seem’st the beauty of thy will,
And that shall lif’s night by that which he all,
Dost for one respect,
For thy self deceivest by their wills be.

And this in their rhyme barren hate,
Under shall statues thy amade, becomes a poet’s day,
And therefore mayst be praise shall cover
One in me, I do return of such shape,

Thereou art be west are so from thee,
By mountient that I may be so belong,
----------------------------------------------------------------
```

Temperature: 1.5

```
----------------------------------------------------------------
Shall I compare thee to a summer’s day?
Thence is endroliggor lalmailing beauty his speak,
With Aprion is roin and doth decake?
Destruining heact dost the charter and his general keep, yito-datients are centaps a bre

Hunt on me. against the carlet to compounds every wone?
Why so base doth life, and but in userd’ation mine.
Seal in the worst what tender anlens’ kingle,
Aming by anjuning hide your self

That stay’st my fleasure old necessary:
More than at best of me’s unthles, death,
Suns am bright, when woman’s face of all that from tender self:
Let some sed on prizing part,

More speek) they find all external wlail,
The pictul estifns to be hate, desired may, I vicious calllal.
----------------------------------------------------------------
```

Temperature : 2

```
----------------------------------------------------------------
Shall I compare thee to a summer’s day?
So ha live? I hall-mbsift, let gofded sipput:
Lait feigureless, and still heart
Rentactify old it grew,
 Landed know. sing mointed I sooce,
Whilst h) it that pen) digs fire,
Whelese’s leedring it back all mere on,
The gentle lost, this nunted befomellly two cancek;
.ntle load pleas, of that whichret-bry:
Duthest perficuling eir state outlors bear
Laining like you hind lascivry, tull mutuars (thy lvess?
Or better contented klight-not?
Thy naturl’nn age, w) or keep remumagl’st
And all those pent, sinq and hy may see tie?
----------------------------------------------------------------
```

The text generated with a low temperature (0.25) tends to be more predictable and conservative. In this case, the model outputs coherent sequences that are more likely to resemble the original Shakespearean style. The text maintains reasonable grammar and structure, though it still contains some mistakes, like odd word choices and some awkward phrasing. At a moderate temperature (0.75), the model introduces more creativity into the output. The generated text is less predictable, and the poem contains more deviations from perfect sentence structure. However, it still manages to resemble Shakespearean themes and motifs, with some parts being logically sound.

At high temperatures (1.5), the model produces text with much higher randomness. The poem becomes more disconnected, with inconsistent sentence structure, strange word choices, and more creative deviations from the original language. The quality of the text decreases, with more gibberish and less coherent ideas. At an even higher temperature (2), the text becomes almost entirely nonsensical. The model generates random combinations of characters that resemble words but do not form logical or meaningful sentences. This result emphasizes how temperature controls the creativity vs. coherence tradeoff.

The LSTM model does a good job of learning the structure of individual sentences. The text generated with lower temperatures follows grammatical rules more closely than with higher temperatures. However, as the temperature increases, sentence structure becomes looser and more chaotic, showing the model’s increasing creativity.

However, the model struggles to generate perfect 14-line sonnets. While it can often generate text that somewhat follows the flow of a sonnet, maintaining the 14-line structure with meaningful content is difficult for the model. It tends to produce more freeform poetry rather than strictly adhering to the sonnet’s form.

Comparing to HMM, we see that the LSTM captures long-term dependencies and sequential data, making it more capable of learning complex patterns, like sentence structure and rhyme schemes. It is able to generate more creative text compared to an HMM. However, the quality still depends on factors like training data, temperature, and model size. The higher the temperature, the more chaotic the output becomes, as seen in the higher temperature results. In contrast, the HMM is a simpler model that operates by assuming a fixed set of states and transitions between them. It is less capable of capturing long-term dependencies compared to an LSTM. While it can generate text with some structure, it is generally less flexible and creative than an LSTM and has difficulty generating high-quality text for longer sequences like sonnets.

However, the LSTM requires significantly more training data and computational resources. It benefits from large datasets and can capture more complex patterns over longer sequences. Training an LSTM for text generation, especially for poetry, requires more iterations and time due to its deep architecture. In this experiment, we only used 30 epochs, so that may have been a reason for its underperformance and more epochs may have been better. On the other hand, a HMM is much faster to train and requires much less data. It relies on statistical transitions between predefined states and can generate text with much less computational overhead. However, it does not scale well to larger, more complex datasets and is less effective at handling longer-term dependencies in text.

5 Haiku Generation and Low-Syllable Constraints

We also trained a 5-state Multinomial HMM on a curated corpus of 100 haikus. Unlike sonnets, haikus amplified the sparsity problem due to the combinatorics of 5-7-5 syllabic constraints. We hypothesize that reinforcement of monosyllabic preference emerges from structural survival bias in constrained generation. Work can be found in our code base.

6 Discussion

Our results demonstrate that while HMMs are effective for enforcing hard constraints, they lack thematic depth. LSTMs capture stylistic texture but violate structure. Hybrid approaches—e.g., using HMMs to generate scaffolds that guide neural decoders—represent a promising direction. Additionally, stress-pattern modeling remains underexplored and is crucial for true meter adherence.

7 Conclusion

We proposed a framework for stylized language generation under structural constraints using both interpretable probabilistic models and expressive neural networks. Our experiments reveal the trade-offs between control and creativity in poetic generation. We release all code and data for future research into constrained text generation and computational creativity.

8 Related Work

Early efforts in poetry generation relied on rule-based systems. With the advent of statistical NLP, Markov chains and n-gram models became popular, though they struggled with long-range dependencies. More recent neural models, such as RNNs and Transformers, have demonstrated greater stylistic control but lack native mechanisms for formal constraints. Recent studies in rhyme enforcement [1], meter modeling [2], and interpretable HMM-based poetic generation [3] show how this can be achieved.

9 Acknowledgements

I would like to thank Professor Yisong Yue and his research group for their guidance in this project

References

[1] Potash, P., Romanov, A., Rumshisky, A. (2016). "ghostwriter: Using an LSTM for automatic rap lyric generation." *EMNLP.*

[2] Ghazvininejad, M. et al. (2016). "generating topical poetry." *EMNLP.*

[3] Jiang, J. et al. (2020). "shakespeare AI: Generating verse with rhyme and meter using HMMs." *ACL Workshop on Computational Creativity.*