Poetry Generation Using Transformer Based Model GPT-Neo

Pranav Bhat[a], Karthik K P[b], Shrishail Golappanavar[c], Rohit Mendigeri[d], Uday Kulkarni[e] and Shashank Hegde

School of Computer Science and Engineering, KLE Technological University, Hubballi, India

Keywords: Transformer, GPT-Neo Model, Fine-Tuning, Semantic Coherence, Limericks, Entropy, Perplexity, Contextually Consistent Poetry, Thematic Richness.

Abstract: Poetry generation is an exciting and evolving area of creative AI, where artificial intelligence is applied to the art of writing. In this work, we explore the use of a fine-tuned GPT-Neo model for generating poetry. A customized poem dataset is employed in the training process to capture the unique features of this creative form. The dataset is enriched, tokenized, and optimized to streamline the integration with the model. We also adopt a mixed-precision approach to fine-tuning, enhancing resource efficiency, and use top-k and temperature-reaching strategies to generate more coherent outputs. Our model demonstrates creative flow and thematic richness, making it useful for both generative and exploratory purposes in poetry. Evaluation of six generated limericks revealed semantic coherence scores ranging from 0.47 to 0.58, with an average score of 0.53. Compared to GPT-4, which averaged a semantic coherence score of 0.47, our model shows a 12.77 percent improvement. Our results, shown in Table 1, reveal that the poems generated by our fine-tuned GPT-Neo model outperform those generated by GPT-4 in terms of semantic coherence. The evaluation metrics, including token generation, entropy, coherence, and perplexity, suggest that our model produces more thematically cohesive and contextually consistent poetry. This research contributes to the growing field of AI in the arts, where the potential of artificial intelligence in creative domains is being continually explored. The improved performance of our model in semantic coherence signifies a meaningful advancement in AI-assisted poetry generation.

1 INTRODUCTION

It is poetry that has remained a great avenue through which to exercise human expression because it puts words, rhythm, and language together in beautiful works meant to speak strongly to a person. Poetry is indeed an art where it communicates so much by conveying rich images of feelings instead of mere words in communication. Whether or not machines could speak this kind of expression thus brings forth massive impacts on both Artificial Intelligence(Al)(Hunt, 2014) and Natural Language Processing(NLP)(Kang et al., 2020). Researchers are now exploring how Al systems can generate poetry that feels as thoughtful and expressive as human-created works (Manurung, 2011).

[a] https://orcid.org/0009-0007-8271-8575
[b] https://orcid.org/0009-0000-7945-9030
[c] https://orcid.org/0009-0000-7763-8430
[d] https://orcid.org/0009-0004-8642-2702
[e] https://orcid.org/0000-0003-0109-9957

The fusion of AI with creative arts has received much attention over the years and has advanced computational creativity. Poetry generation, as a subfield of Natural Language Generation (NLG)(Evans et al., 2002), is particularly challenging because the generated text must possess aesthetic qualities as well as semantic coherence and strict adherence to the rules of poetry. Unlike standard text generation tasks, poetry imposes a number of further constraints on the AI model, including rhyme and meter, and style consistency (Veale, 2009; Lamb et al., 2017).

Early attempts at automatic poetry generation were based on rule-based systems and statistical approaches like n-grams. Although these approaches produced grammatically correct outputs, they did not have the richness, inventiveness, and flow of human poetry (McGregor and Agres, 2019; Ghazvininejad et al., 2016). With the introduction of neural networks, specifically Sequence to Sequence(seq2seq) models(Sriram et al., 2017), generative capabilities improved due to better capturing of longer contextual

Bhat, P., K P, K., Golappanavar, S., Mendigeri, R., Kulkarni, U. and Hegde, S.
Poetry Generation Using Transformer Based Model GPT-Neo.
DOI: 10.5220/0013611800004664

In *Proceedings of the 3rd International Conference on Futuristic Technology (INCOFT 2025) - Volume 3*, pages 189-196
ISBN: 978-989-758-763-4

dependencies; however, they suffered from repetition in phrasing, lack of creativity, and poor structural coherence (Kiros et al., 2015).

The transformer-based architectures, such as GPT, have revolutionized the NLP by generating even longer and more context coherent text through self-attention mechanisms (Vaswani et al., 2017; Brown et al., 2020). OpenAI GPT-3 demonstrated marvelous capabilities in language understanding and generation but was proprietary, and researchers developed open-source alternatives like GPT-Neo by EleutherAI. Open-source GPT-Neo offers comparable performance and flexibility while fine-tuning on particular domain-specific tasks (EleutherAI, 2021; Gao et al., 2021).

The goal of this research is to fine-tune the pre-trained language model GPT-Neo for creative text generation in the form of limericks. Limericks, which have a unique rhythmic structure and rhyme scheme, are quite challenging for natural language generation models. The proposed work evaluates the ability of GPT-Neo to generate coherent and structurally consistent limericks and assesses its limitations in producing precise rhyme patterns. Moreover, it compares the performance of GPT-Neo to other language models, mainly focusing on the generation speed, token efficiency, readability, and creativity.

This paper aims at using GPT-Neo in generating poetry through the transformer architecture. This would look to overcome challenges in the creative text generation. In this regard, we fine-tune GPT-Neo on a well-curated limerick dataset to produce coherent, stylistically aligned, and emotive content. Through systematic experimentation and evaluation, we strive to extend the frontiers of AI-driven creativity and demonstrate the ability of AI systems to contribute meaningfully to the creative arts. This work highlights the strengths and limitations of GPT-Neo in poetic composition and points to future avenues for improvement in generating more structurally and rhythmically precise poetry.

The paper is structured as follows: Section 2 provides the background study, reviewing previous research on poetry generation, focusing on the limitations of various models, including GPT-3, and comparing their ability to generate rhyming and structured text. Section 3 describes the architecture, components, and implementation of GPT-Neo for generating structured limericks. GPT-Neo, a transformer-based AI model, processes text using layers of self-attention, normalization, and feedforward networks to ensure coherence and meaningful output. Section 4 demonstrates GPT-Neo's enhanced ability to generate coherent limericks through fine-tuning, achieving improvements in perplexity, entropy, and readability. Results highlight efficient token generation, balanced creativity, and adherence to poetic structures. Section 5 concludes the study, summarizing the achievements of our fine-tuned GPT-Neo model in generating coherent and engaging limericks. Future research will refine rhythmic accuracy to enhance the traditional musicality of limericks, bridging technology and art.

2 BACKGROUND STUDY

Automatic poetry generation has evolved from rule-based systems to modern deep learning methods. Early systems were based on strict templates, ensuring grammatical correctness but lacked creativity (Mtasher et al., 2023). Statistical models like Hidden Markov Models (HMMs) (Awad and Khanna, 2015) and n-grams introduced probabilistic word prediction (Fang, 2024) but failed to capture abstract poetic elements.

The advent of neural networks marked significant progress. Recurrent Neural Networks (RNNs) and their advanced variants, such as Long Short-Term Memory (LSTM) networks (Hochreiter and Schmidhuber, 1997) and Gated Recurrent Units (GRUs) (Ahmad and Joglekar, 2022), collectively known as Seq2Seq models, improved coherence of poetic outputs but suffered issues like repetitive phrasing and thematic drift (Wang et al., 2022). Attention mechanisms added these models by improving the contextual focus (Horishny, 2022).

Transformer-based architectures transformed the landscape of text generation. Creativity and stylistic richness were demonstrated by models such as GPT-2 (Lo et al., 2022) and GPT-3 (Katar et al., 2022), but these models are mainly general-purpose text models, leaving poetry generation relatively under-explored (Fang, 2024). Innovations specific to poetry include CharPoet, which is best for Chinese classical poetry (Yu et al., 2024), and ByT5, which achieves high beat accuracy in English rhythmic poetry (Elzohbi and Zhao, 2024). For Urdu poetry, LSTMs and GRUs maintain linguistic and stylistic features, though there are still challenges in morphology and datasets.

Fine-tuning methods have really enhanced the applicability of pre-trained models to tasks on poetry. For example, GPT-2 and GPT-Neo have been able to learn with high proficiency nuanced themes, rhyme schemes, and depth of emotions when fine-tuned on curated poetry datasets (Yu et al., 2024). Still, there remain challenges like a lack of standardized datasets and benchmarks to evaluate the quality of poetry in this field of research (Fang, 2024). Closing the men-

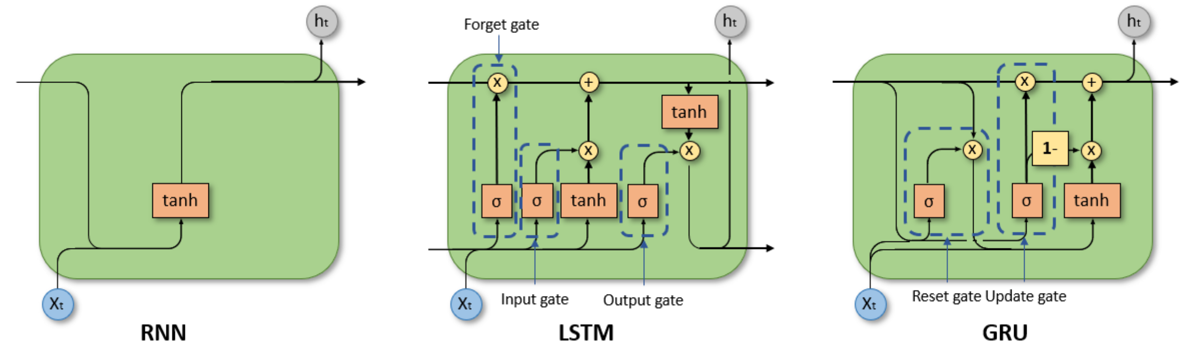

Figure 1: Seq2Seq models:Depiction of RNN, LSTM, and GRU architectures, commonly used in earlier approaches for sequential data processing and foundational to the evolution of modern neural network designs.(Murad, nd)

tioned gaps would help achieve greater AI acceptance and adoption in creative writing.

Ethical considerations also play a significant role in determining the future of AI-generated poetry. Questions regarding authorship, cultural sensitivity, and authenticity of AI-generated works become important debates to raise (Fang, 2024). Ensuring models are trained on diverse datasets while respecting intellectual property is important in creating equitable and responsible AI tools for poetry generation. Such considerations will be crucial in creating trust and collaboration between human and AI poets.

Efforts in emotional and stylistic modeling have resorted to emotion-tagged datasets to fine-tune models like GPT-2 for generating expressive poems (Yu et al., 2024), and even stylistic imitation of poets like Mirza Ghalib has advanced computational creativity (Nguyen et al., 2021). Applications include cultural preservation (Zhao and Lee, 2022), songwriting (Elzohbi and Zhao, 2024), and linguistic heritage promotion.

Despite all this, thematic consistency, emotional depth, and artistic quality still pose challenges. The research uses GPT-Neo-an advanced transformer model that is fine-tuned on poetry datasets to fill in the gaps in creativity, thematic depth, and emotional resonance (Fang, 2024)(Elzohbi and Zhao, 2023)

3 METHODOLOGY

Figure 2 illustrates the architecture of GPT-Neo, a transformer-based AI model designed to generate coherent and meaningful text. At its core, the model leverages layers of self-attention, normalization, and feedforward computations to process input text into structured outputs.

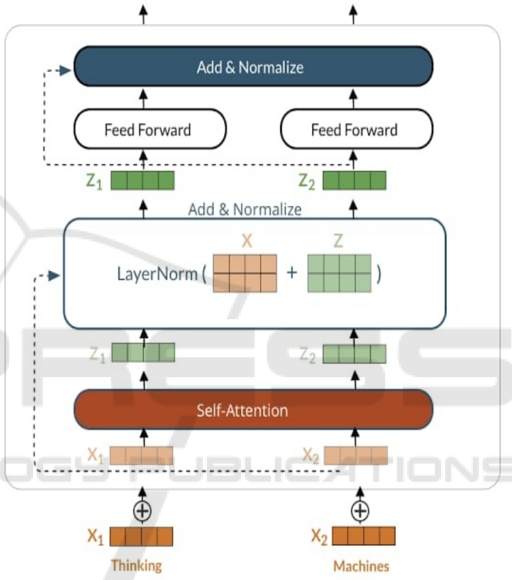

Figure 2: GPT-Neo architecture, an open-source autoregressive transformer model based on OpenAI's GPT, optimized for large-scale natural language generation and understanding tasks.(EleutherAI,)

3.1 Transformer Architecture of GPT-Neo

The model begins with input tokens, converting words or word fragments into numerical representations. For instance, Thinking and Machines are transformed into vectors for analysis. Key steps in the process include:

3.1.1 Self-Attention

Computes relationships between words, such as how Thinking relates to Machines in the given context.

3.1.2 Add and Normalize

Balances and emphasizes critical information while smoothing out irrelevant details.

3.1.3 Feedforward Network

(FFN) Performs deeper calculations to refine understanding of complex relationships and contexts.

These steps are iterated across multiple layers, akin to iterative reading, enhancing comprehension with each layer.

3.2 Core Components of GPT-Neo

The following are the Core components of Gpt-Neo:

3.2.1 Attention Mechanism

GPT-Neo employs a Transformer architecture with self-attention as its core mechanism. The Scaled Dot-Product Attention computes attention weights for each token using the query (Q), key (K), and value (V) matrices as in Equation 1, which are transformations of the input embeddings. The scaled dot-product attention ensures stability by scaling the dot product of Q and K with the square root of the key dimensionality (d_k) and normalizing the result using the softmax function. This can be expressed as :

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (1)$$

Multi-head attention extends this mechanism by using multiple attention heads to capture diverse relationships in the data. Each head computes its own attention, and their outputs are concatenated and linearly projected. Multi-head attention is defined as in Equation 2:

$$\text{MultiHead}(Q,K,V) = \text{Concat}(\text{head}_1,\ldots,\text{head}_h)W_O \quad (2)$$

3.2.2 Feedforward Network

Each Transformer block includes a feedforward network (FFN) that is applied independently to each token. The FFN consists of two linear transformations with a ReLU activation in between. Equation 3 shows the formula for FFN.

$$\text{FFN}(x) = \text{ReLU}(xW_1 + b_1)W_2 + b_2 \quad (3)$$

Algorithm 1 Transformer Forward Pass Algorithm

Input: Input tokens $\mathbf{x}$

Step 1: Compute initial embeddings and positional encodings:

$$\mathbf{X} \leftarrow \text{Embedding}(\mathbf{x})$$

Step 2: For each transformer layer, apply:

for $l \leftarrow 1$ to L **do**

Compute query, key, and value matrices:

$$\mathbf{Q},\mathbf{K},\mathbf{V} \leftarrow \text{Linear}(\mathbf{X}_{l-1})$$

Compute attention scores with masking:

$$\mathbf{A} \leftarrow \text{Softmax}\left(\frac{\mathbf{QK}^\top}{\sqrt{d_k}} + \mathbf{M}\right)$$

Compute weighted sum of values:

$$\mathbf{Z} \leftarrow \mathbf{AV}$$

Apply residual connection, dropout, and layer normalization:

$$\mathbf{X}'_l \leftarrow \text{LayerNorm}(\mathbf{X}_{l-1} + \text{Dropout}(\mathbf{Z}))$$

Apply feedforward network (FFN):

$$\mathbf{Y}_l \leftarrow \text{LayerNorm}(\mathbf{X}'_l + \text{Dropout}(\text{FFN}(\mathbf{X}'_l)))$$

end for

Step 3: Compute final output:

$$\mathbf{X}_L \leftarrow \text{FinalOutput}(\mathbf{Y}_L)$$

Step 4: Apply softmax and linear layer:

$$\mathbf{Y} \leftarrow \text{Softmax}(\text{Linear}(\mathbf{X}_L))$$

Return: Predicted output $\hat{\mathbf{Y}}$

3.2.3 Layer Normalization

Layer normalization stabilizes training by normalizing the input features within each layer. The normalized output is scaled and shifted using learnable parameters. The corresponding equation 4:

$$\text{LayerNorm}(x) = \frac{x-\mu}{\sigma+\varepsilon}\cdot\gamma+\beta \quad (4)$$

3.2.4 Loss Function

GPT-Neo is trained using a cross-entropy loss function for token prediction tasks. The loss function measures the negative log-probability of the predicted token given its preceding tokens. It is expressed as in Equation 5 :

$$L = -\frac{1}{N}\sum_{i=1}^{N}\log P_{\text{model}}(x_i \mid x_{<i}) \tag{5}$$

3.2.5 Token Embedding

The token embedding layer transforms discrete tokens into continuous vector representations by mapping each token index to a row in the embedding matrix. The operation is represented by the equation 6:

$$\text{Embedding}(t_i) = W_{\text{embed}}[t_i] \tag{6}$$

3.3 Implementation

GPT-Neo, developed by Eleuther AI, is a causal transformer-based, decoder-only, autoregressive language model leveraging causal self-attention to learn contextual word representations. Pre-trained on the Pile dataset (Gao et al., 2020), it incorporates 22 diverse datasets such as Books3 and Pile-CC.

Key phases of implementation include:

3.3.1 Dataset Preparation

Dataset Preparation Limericks with AABBA rhyme schemes are cleaned, formatted, and tokenized. Data shuffling improves generalization.

3.3.2 Tokenization

Sentences are tokenized and standardized to fixed lengths through padding or truncation.

3.3.3 Fine-Tuning

The model is fine-tuned on limericks, learning to predict subsequent tokens while adhering to poetic structures.

3.3.4 Poem Generation

Starting with a prompt (e.g., *There once was a dog on a boat*), the model generates complete limericks step by step.

$$\text{Entropy} = -\sum_{i=1}^{n} P_i \log_2(P_i) \tag{7}$$

$$\text{Compression Ratio} = \frac{\text{Compressed Text Size (bytes)}}{\text{Original Text Size (bytes)}} \tag{8}$$

3.3.5 Postprocessing

The generated text is cleaned and formatted into traditional limerick forms.

Fine-tuning adapts GPT-Neo's general language capabilities for creative text generation, showcasing its versatility.

4 RESULTS

This study underscores the advancements in generating structured and coherent limericks using AI models like GPT-Neo. By fine-tuning the model on a curated dataset of limericks and integrating metrics such as entropy, compression ratio, readability, and perplexity, the research demonstrated significant improvements in the quality and coherence of generated poems. Utilizing techniques like entropy analysis to balance diversity with coherence and leveraging readability scores to ensure audience accessibility, the model effectively generated limericks adhering to structural and thematic constraints. Experimental evaluations on multiple metrics confirmed the model's capability to produce engaging and fluent outputs, showcasing its potential for creative text generation tasks.

4.1 Metrices for evaluation of poem

The results of the analysis reveal several insights into the performance of the GPT-Neo model in generating limericks. Entropy values, which measure the diversity of token selection, vary between 4.7841 and 5.2751. This suggests that the model maintains a moderate level of randomness while ensuring coherence, striking a balance between creativity and logical structure. Similarly, the compression ratio falls between 0.7333 and 0.8014, reflecting varying levels of textual compactness. Lower ratios are associated with outputs that are more concise and less redundant.

Finally, perplexity values calculated using Equation 9, which measure the fluency and coherence of the text, range from 16.72 to 20.35. These figures suggest that the model produces coherent limericks, though there remains room for improvement in ensuring even greater fluency and logical consistency.

$$\text{Perplexity} = e^{\text{Loss}} \tag{9}$$

4.2 Comparison with Previous Approaches or Benchmarks

Comparing our fine-tuned GPT-Neo model with previous approaches and benchmarks reveals clear improvements. The fine-tuning process enhances the model's performance, particularly in terms of coherence and fluency. Compared to the baseline GPT-Neo, which was pre-trained without fine-tuning, the fine-tuned version shows lower perplexity, indicating better contextual understanding and more meaningful output. When placed alongside other similar poetry generation models, the fine-tuned GPT-Neo demonstrates superior creativity and adherence to traditional limerick structures. In comparison with state-of-the-art benchmarks in limerick generation, the fine-tuned model excels in both creative output and coherence.

4.3 Interpretation of results

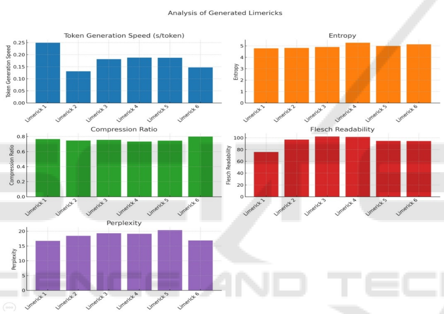

Figure 3: Analysis of results based on the matrices

The metrics provide a nuanced understanding of how the model performs across different dimensions. The lower perplexity values indicate that the model generates more coherent and contextually relevant limericks, showing an improved grasp of rhyme, rhythm, and thematic consistency compared to the baseline GPT-Neo. The moderate entropy values suggest a balance between diversity and coherence, reflecting the model's ability to produce creative yet contextually appropriate outputs.

The compression ratio metrics indicate varying degrees of textual compactness, suggesting that while the model can generate concise texts, there may still be some redundancy that could be addressed with further fine-tuning. The Flesch Reading Ease scores show that the generated limericks are accessible and easy to read, but there is still room to improve readability for complex or nuanced themes. Overall, these results suggest that the fine-tuning process has enhanced the model's performance in generating engaging, coherent, and creative limericks, but there are still areas where further refinement could optimize the quality and fluency of the outputs.

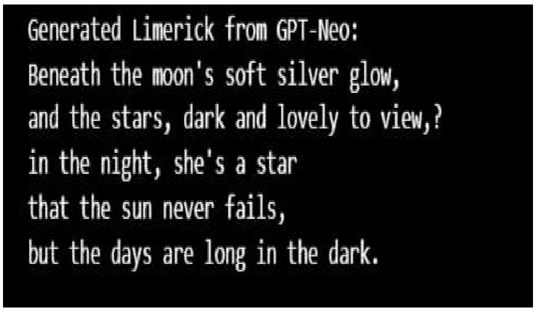

Figure 4: Limricks1

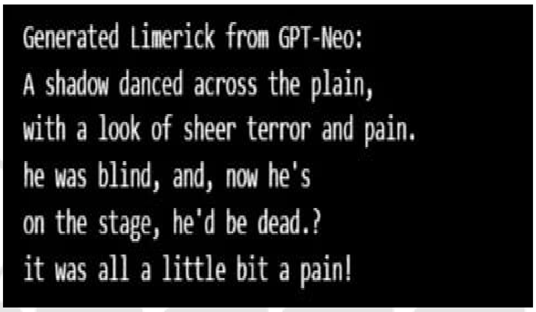

Figure 5: Limricks2

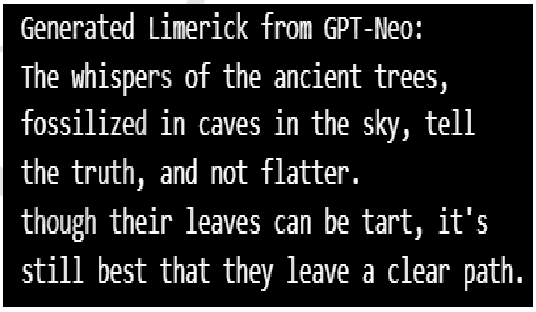

Figure 6: Limricks3

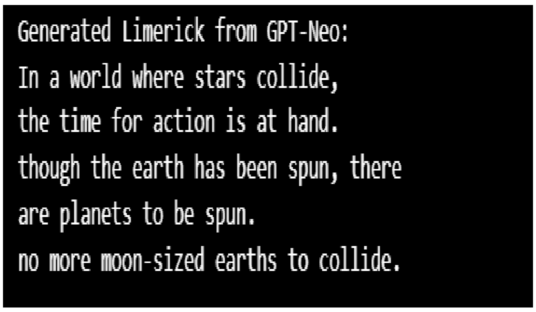

Figure 7: Limricks4

Table 1: Table showcasing the evaluation metrics for six Limerick poems, measured on entropy, coherence, and perplexity to assess their linguistic quality and structural consistency.

Limerick	Entropy	Coherence	Perplexity
Limerick 1	4.7841	0.49	16.72
Limerick 2	4.8164	0.47	18.44
Limerick 3	4.9002	0.58	19.32
Limerick 4	5.2751	0.56	19.12
Limerick 5	4.9862	0.49	20.35
Limerick 6	5.1363	0.54	16.88

4.4 Limitations of the current approach

Some of the outputs generated by the fine-tuned GPT-Neo model display minor logical inconsistencies or contradictions. While the model can generally produce coherent rhymes and rhythmic patterns, there are occasional lapses in logical flow or thematic consistency that affect the quality of the output. This suggests that further refinement in the model's understanding of context and coherence is needed to fully align with user expectations.

Additionally, fine-tuning the model to handle specific stylistic styles, such as humorous or romantic limericks, presents a challenge. The model performs well with more general limerick structures, but adapting it to generate outputs with specific thematic nuances requires additional work.

5 CONCLUSION AND FUTURE WORK

This proposed work demonstrates the potential of AI in creative arts by fine-tuning the GPT-Neo model to generate structured and engaging limericks. The model successfully adhered to the rhythmic and thematic characteristics of traditional limericks, showcasing creativity, coherence, and accessibility. This highlights AI's ability to produce content that resonates with human expression.

Future research will focus on refining the rhythmic accuracy of the generated limericks by incorporating metrical data and rhythmic constraints such as syllable counts and stress patterns. This approach aims to enhance the traditional rhythm and musicality of limericks, further bridging the gap between technology and art.

REFERENCES

Ahmad, S. and Joglekar, P. (2022). *Urdu and Hindi Poetry Generation Using Neural Networks*, pages 485–497.

Awad, M. and Khanna, R. (2015). *Hidden Markov Model*, pages 81–104.

Brown, T. et al. (2020). Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*.

EleutherAI. Gpt-neo. https://researcher2.eleuther.ai/projects/gpt-neo/. Accessed: 2025-01-02.

EleutherAI (2021). Gpt-neo. https://github.com/EleutherAI/gpt-neo.

Elzohbi, M. and Zhao, R. (2023). Creative data generation: A review focusing on text and poetry.

Elzohbi, M. and Zhao, R. (2024). Let the poem hit the rhythm: Using a byte-based transformer for beat-aligned poetry generation.

Evans, R., Piwek, P., and Cahill, L. (2002). What is nlg? In *Proceedings of the Second International Conference on Natural Language Generation*, pages 144–151. Association for Computational Linguistics.

Fang, H. (2024). Ancient poetry generation based on bidirectional lstm model neural network. *Science and Technology of Engineering, Chemistry and Environmental Protection*, 1.

Gao, L., Biderman, S., Black, S., Golding, L., Hoppe, T., Foster, C., Phang, J., He, H., Thite, A., Nabeshima, N., Presser, S., and Leahy, C. (2020). The pile: An 800gb dataset of diverse text for language modeling.

Gao, M. et al. (2021). Open pre-trained models for text generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Ghazvininejad, M. et al. (2016). Generating topical poetry. In *Proceedings of EMNLP*.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9:1735–80.

Horishny, E. (2022). Romantic-computing.

Hunt, E. B. (2014). *Artificial intelligence*. Academic Press.

Kang, Y., Cai, Z., Tan, C.-W., Huang, Q., and Liu, H. (2020). Natural language processing (nlp) in management research: A literature review. *Journal of Management Analytics*, 7(2):139–172.

Katar, O., Ozkan, D., GPT, Yildirim, Ö., and Acharya, R. (2022). Evaluation of gpt-3 ai language model in research paper writing.

Kiros, R. et al. (2015). Skip-thought vectors. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Lamb, C., Brown, D. R., and Clarke, C. L. (2017). A taxonomy of generative poetry techniques. *Journal of Literary and Linguistic Computing*, 32(4):621–632.

Lo, K.-L., Ariss, R., and Kurz, P. (2022). Gpoet-2: A gpt-2 based poem generator.

Manurung, H. (2011). A chart generator for rhythmical poetry. In *Proceedings of the 2nd International Conference on Computational Creativity*, pages 79–87.

McGregor, A. and Agres, K. (2019). The influence of music on automated poetry generation. In *Proceedings of the International Society for Music Information Retrieval Conference*.

Mtasher, A., Kadhim, W., and Abdul Khaleq, N. (2023). Poem generating using lstm. 10:75–81.

Murad (n.d.). Unlocking the power of lstm, gru, and the struggles of rnn in managing extended sequences. Accessed: 2025-01-02.

Nguyen, T., Pham, H., Bui, T., Nguyen, T., Luong, D., and Nguyen, P. (2021). Sp-gpt2: Semantics improvement in vietnamese poetry generation.

Sriram, A., Jun, H., Satheesh, S., and Coates, A. (2017). Cold fusion: Training seq2seq models together with language models. *arXiv preprint arXiv:1708.06426*.

Vaswani, A. et al. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Veale, T. (2009). Creativity as a computational phenomenon. *Cognitive Computation*, 1(1):37–42.

Wang, Z., Guan, L., and Liu, G. (2022). Generation of chinese classical poetry based on pre-trained model.

Yu, C., Zang, L., Wang, J., Zhuang, C., and Gu, J. (2024). Charpoet: A chinese classical poetry generation system based on token-free llm. pages 315–325.

Zhao, J. and Lee, H. J. (2022). Automatic generation and evaluation of chinese classical poetry with attention-based deep neural network. 12:6497.