Recursive Journal of Informatics
Vol. 3, No. 2, September 2025
p-ISSN 2963-5551 https://journal.unnes.ac.id/journals/index.php/rji e-ISSN 2986-6588

Improving Pantun Generator Performance with Fine Tuning Generative Pre-Trained Transformers

Achmat Sodikkun[1], Kholiq Budiman[2]
[1,2]Computer Science Department, Faculty of Mathematics and Natural Sciences,
Universitas Negeri Semarang, Indonesia

Abstract.
Purpose: The study aims to address the challenges in generating high-quality pantun, an important element of Indonesian cultural heritage. Traditional methods struggle with limited vocabulary, variation, and consistency in rhyme patterns. This research seeks to enhance the performance of a pantun generator by applying fine-tuning techniques to the Generative Pre-trained Transformers (GPT) model, coupled with post-processing, and validated by linguistic experts.
Methods/Study design/approach: The research involves fine-tuning the GPT model using a dataset of Indonesian pantun. The methodology includes dataset collection, data pre-processing for cleaning and adjustment, and hyperparameter optimization. The effectiveness of the model is evaluated using perplexity and rhyme accuracy metrics. The study also incorporates post-processing to refine the generated pantun further.
Result/Findings: The study achieved a best perplexity value of 14.64, indicating a strong predictive performance by the model. Post-processing significantly improved the rhyme accuracy of the generated pantun to 89%, a substantial improvement over previous studies by Siallagan and Alfina, which only achieved 50%. These results demonstrate that fine-tuning the GPT model, supported by appropriate hyperparameter settings and post-processing techniques, effectively enhances the quality of generated pantun.
Novelty/Originality/Value: This research contributes to the development of generative applications in Indonesian, particularly in the context of cultural preservation. The findings highlight the potential of fine-tuning GPT models to improve language generation tasks and provide valuable insights for creative and educational applications. The validation by experts ensures that the generated pantun adheres to established writing standards.

Keywords: Pantun, Generative, Pre-trained, Transformers, fine-tuning

Received Month 20xx / **Revised** Month 20xx / **Accepted** Month 20xx

INTRODUCTION

One of the cultural heritages that needs to be preserved and maintained is pantun. Pantun is an old form of poetry consisting of four lines with an a-b-a-b or a-a-a-a rhyme scheme. The first two lines are called sampiran, while the third and fourth lines are called isi [1]. The sampiran usually serves as a rhyming introduction and is unrelated to the isi, which conveys the main message of the pantun. According to Wahyuni [2], pantun has three characteristics: it consists of four lines with an a-b-a-b pattern; each line comprises 8-12 syllables; and the first two lines are sampiran, while the last two lines are isi. Originally, pantun was an oral literary form, but it is now also found in written literature [3]. Pantun is often used in traditional ceremonies and various cultural events as a reflection of local wisdom. Additionally, it is utilized as a teaching material for students from elementary to high school in Indonesian language subjects.

Despite its cultural significance, there are several challenges in creating pantun, particularly in educational settings. According to Mr. Arifin, one major challenge is the difficulty in finding appropriate rhyme combinations between sampiran and isi to adhere to traditional rules. As discussed by Mr. Anthoni, rhyme refers to the sound pattern at the end of the lines, while the rhyme scheme encompasses the overall structure of the pantun. Other challenges include limited vocabulary knowledge, lack of variety, and difficulty maintaining consistent patterns and rhymes [4]. These issues affect the quality of the pantun produced, often resulting in deviations from traditional writing rules, less engaging content, and less appealing

[1]*Corresponding author.
Email addresses: shodikkunachmat21@students.unnes.ac.id (Sodikkun)
DOI: 10.15294/sji.v8i1.25356

language style [1]. To address these issues, there is a need for an automated tool for generating pantun that complies with traditional rules, utilizing artificial intelligence (AI) technologies like Generative Pre-trained Transformers (GPT).

GPT is an AI-based natural language model architecture capable of generating human-like text with high accuracy [5]. GPT has improved text generation capabilities through modifications in training objectives and more efficient sampling techniques, as seen in GPT-2. According to Shree [6], GPT can be used for various Natural Language Processing (NLP) tasks, such as answering questions and summarizing texts. It is one of the latest examples of NLP models using transformer architectures, which can learn from large text datasets and produce text similar to human writing [7]. Therefore, GPT can be used to assist in generating pantun and other NLP tasks. Despite significant advancements in GPT development for NLP tasks, such as text processing and translation, optimization is necessary for specific tasks like generating pantun according to traditional rules.

Fine-tuning allows the model to adapt to specific contexts and styles [8], thereby enhancing its ability to generate pantun that adhere to desired rules and aesthetics. By refining the training data and selecting optimal hyperparameters, fine-tuning can produce a more robust model for these specific tasks [9]. In addition to performance improvements, fine-tuning enables the model to gain a deeper understanding of the creative and aesthetic elements involved in generating pantun [10]. This allows the model to produce more unique and creatively appealing works.

Thus, fine-tuning the GPT model for a pantun generator is not only about improving the quality of the final output but also about enhancing the model's efficiency, contextual sensitivity, and creative ability to create pantun that meets traditional rules. With this pantun generator, it is hoped that it can help address the problems in creating pantun and preserve Indonesian culture, especially in the field of literature and culture. Additionally, this application can be used as a learning medium to assist students in creating pantun.

Previous research has applied fine-tuning to optimize GPT models for text generation tasks, specifically pantun creation. In research by [10], the performance of two generative models, Sequence Generative Adversarial Nets with Policy Gradient (SeqGAN) and GPT-2, was compared using a pantun dataset of 7.8 thousand entries. The study showed that GPT-2 outperformed SeqGAN across all metrics, including structure, rhyme, and lexical richness. However, both models failed in rhyme accuracy, with GPT-2 achieving only 50% accuracy. Another study successfully addressed rhyme errors through post-processing to improve the rhyme to match desired patterns. The study by [11] employed fine-tuning of the GPT-2 model to generate limericks through a two-stage generation process utilizing forward and backward language modeling, achieving a highest perplexity of 18.3541 and successfully evaluating the rhyme scheme as a-a-b-b-a-a through post-processing. This study inspired the author to enhance pantun rhyme accuracy using a similar method, involving post-processing with pantun decomposition.

Another study by [6] focused on a pantun generator using the GPT-2 model with fine-tuning. The study used a dataset of 636 Indonesian pantun obtained through manual web scraping, resulting in a perplexity of 16.941 and successfully creating a pantun generator application using Streamlit. However, the application did not use a fine-tuned model but instead searched for pantun from a provided dataset based on input of two lines of sampiran or isi.

From these studies, perplexity scores ranged from 16 to 18, with the best rhyme accuracy at 50%. These perplexity and accuracy scores still need improvement to produce pantun that adhere to traditional writing rules. The author intends to combine methods previously used by other researchers, including using the GPT-2 model with fine-tuning and post-processing with pantun decomposition. These methods will be trained using datasets used in previous studies, which consist of Indonesian pantun.

METHODS

The methodology employed in this study aims to enhance the performance of a pantun generator through the fine-tuning of the Generative Pre-trained Transformer (GPT) model and the application of advanced postprocessing techniques. The research process is structured into several critical stages, including data preparation, model fine-tuning, and postprocessing. Each stage involves specific modifications designed to improve the quality of the generated pantun. The detailed procedures and data analysis techniques used in this study are systematically illustrated in Figure 1.

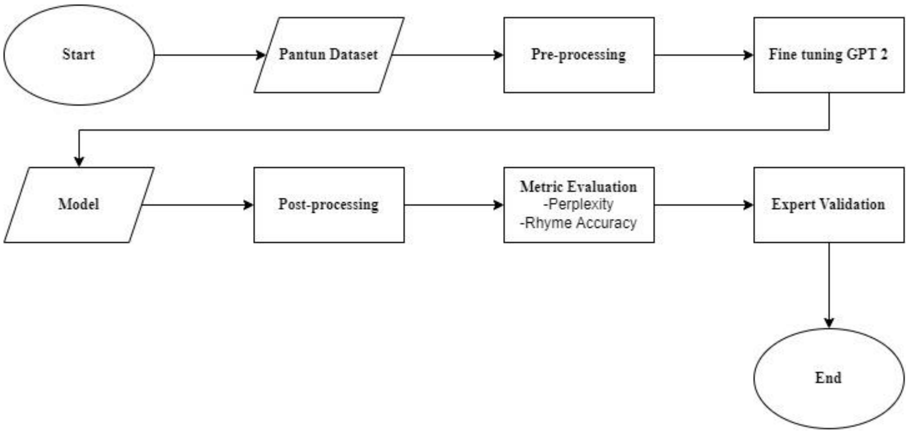

Figure 1. Research Design Flowchart

Dataset

The first step involves acquiring datasets from two sources: the Siallagan dataset from GitHub and the Khairul dataset from Google Drive [6]. The Siallagan dataset, uploaded by ir-nlp-csui in 2022 [10], contains 7,808 Indonesian pantuns. Examples of pantuns from the Siallagan dataset are shown in Table 1.

Table 1. Sample Dataset Siallagan.

Text
<BOS> karena terlalu lama berdiri <CLS> para tamu pada ngeliatin <CONTENT> selamat hari raya idul fitri <CLS> mohon maaf lahir dan batin <EOS>
<BOS> tanaman diserang hama <CLS> kata adik sepupu ibuku <CONTENT> waktu jadi terasa sangat lama <CLS> bila dirimu tak ada disampingku <EOS>
<BOS> di bawah pohon yang mati <CLS> mereka asyik membaca buku <CONTENT> akan datang waktunya nanti <CLS> kau akan menyesal tlah pergi dariku <EOS>
<BOS> dimanamana jalan macet <CLS> terpantau dipadati para pemudik <CONTENT> lagi kangen kamu banget <CLS> kangen ngajak kamu piknik <EOS>
<BOS> lebah sembunyi didalam sarang <CLS> jangan dekatdekat berbahaya <CONTENT> ramadhan ini belum punya ayang <CLS> lebaran nanti moga udah punya <EOS>

Additionally, the research uses the Khairul dataset, which was uploaded in 2023 on Google Drive and contains 636 Indonesian pantuns. Examples from the Khairul dataset are shown in Table 2.

Table 2. Sample Dataset Khairul.

Text
Anak teruna tiba di darat \n Dari Makasar langsung ke Deli \n Hidup di dunia biar beradat \n Bahasa tidak dijual beli \n,Pantun Adat
Lebat daun bunga tanjung \n Berbau harum bunga cempaka \n Adat dijaga pusaka dijunjung \n Baru terpelihara adat pusaka \n,Pantun Adat
Laksamana berbaju besi \n Masuk ke hutan melanda-landa \n Hidup berdiri dengan saksi \n Adat berdiri dengan tanda \n ,Pantun Adat
Gadis Aceh berhati gundah \n Menanti teruna menghulur tepak \n Gula manis sirih menyembah \n Adat dijunjung dipinggir tidak \n ,Pantun Adat
Bukan lebah sembarang lebah \n Lebah bersarang di buku buluh \n Bukan sembah sembarang sembah \n Sembah bersarang jari sepuluh \n ,Pantun Adat

The Siallagan and Khairul datasets form the foundation for fine-tuning the GPT-2 model to generate pantuns that adhere to traditional structure and rhyme patterns. These preprocessed data sets are then used to train the model, aiming to enhance the accuracy and quality of the generated pantuns.

Preprocessing

The preprocessing phase is a crucial step in preparing the datasets for fine-tuning the GPT-2 model to generate traditional pantuns with accurate rhyme schemes. The preprocessing involves several key tasks, including:

Data Cleaning
The datasets, Siallagan and Khairul, underwent thorough cleaning to remove any irrelevant or noisy data. This includes eliminating unnecessary punctuation and standardizing text formatting to ensure consistency across the data. The results of the data cleaning can be seen in the Figure 2 and Figure 3.

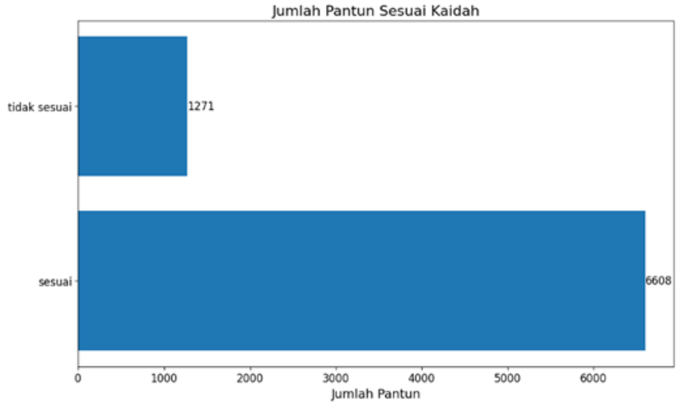

Figure 2. Data Cleaning Siallagan Dataset

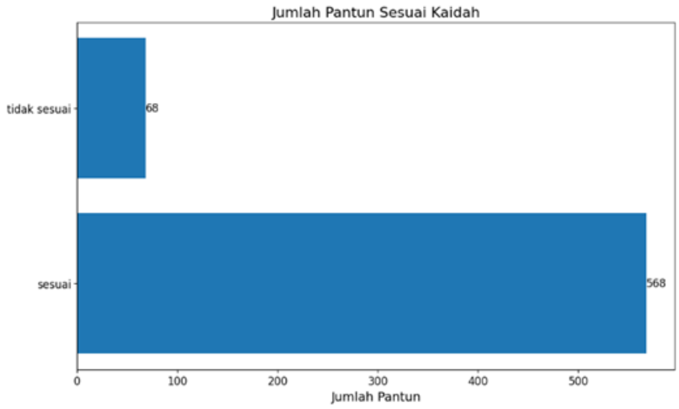

Figure 3. Data Cleaning Khairul Dataset

Tokenization.
Each pantun was tokenized into meaningful units to facilitate the model's understanding of the pantun structure. The tokenization process also involved identifying key segments of the pantun, such as the introduction, content, and rhyme pairs, which were marked with special tokens (e.g., <BOS>, <CONTENT>, <CLS>, <EOS>). The dataset will go through two tokenization processes, for the sampiran and the isi which can be seen in Figure 4.

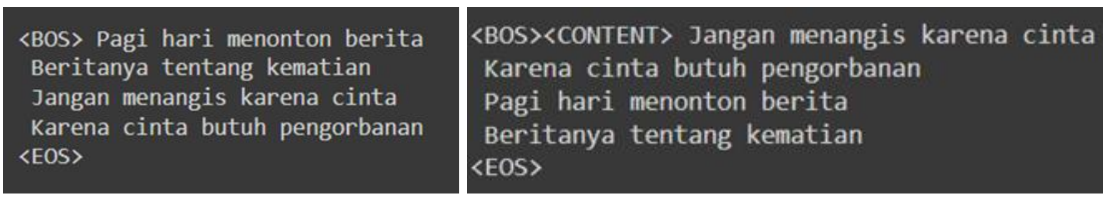

Figure 4. Tokenization Dataset

To maintain the traditional four-line structure of a pantun, sequence labeling was applied to ensure that each pantun was segmented properly into its two couplets (e.g., a-b-a-b rhyme scheme). This labeling is crucial for the model to learn and reproduce the pantun's structure accurately. In addition, the difference in <CONTENT> tokens and changes in pantun structure are expected for the model to be able to distinguish between isi and sampiran.

Splitting the Dataset

Only pantuns that adhered to the traditional structure were selected for the training process. From the Siallagan dataset, 6,608 pantuns were deemed suitable after preprocessing, while 568 pantuns from the Khairul dataset met the criteria. This selection was based on the number of syllables, rhyme patterns, and thematic relevance, ensuring that the final dataset was of high quality for model training. The dataset will then be divided as can be seen in Table 3.

Table 3. Splitting Dataset

Dataset	Total	Percent
Data Train Isi Siallagan	5286	80%
Data Train Sampiran Siallagan	5286	80%
Data Test Isi Siallagan	1322	20%
Data Test Sampiran Siallagan	1322	20%
Data Train Isi Khairul	539	95%
Data Train Sampiran Khairul	539	95%
Data Test Isi Khairul	29	5%
Data Test Sampiran Khairul	29	5%

The preprocessed datasets were then divided into training and testing sets. For the Siallagan dataset, an 80-20 split was used, while the Khairul dataset was split 95-5. This step was vital to create a balanced dataset that could be effectively used for both model training and evaluation. The Sampiran data and isi data will be combined into one as shown in Table 4.

Table 4. Dataset Merging

Dataset	Total
Data Train Siallagan	10572
Data Test Siallagan	2644
Data Train Khairul	1078
Data Test Khairul	58

Finetuning GPT 2

The fine-tuning process in this research was crucial for adapting the GPT-2 model to generate pantuns, a traditional Indonesia poetic form, using the Siallagan and Khairul datasets. The Siallagan dataset, containing 7,808 pantuns, and the Khairul dataset, with 636 pantuns, were selected for their cultural relevance and subjected to extensive preprocessing. This preprocessing involved cleaning, tokenization, and sequence labeling to ensure that each pantun adhered to traditional structures and rhyme patterns (a-b-a-b or a-a-a-a). The fine-tuning was conducted using Causal Language Modeling (CLM) on the gpt2-medium-indonesian model from Huggingface, within a Python environment on Google Colab with a T4 GPU.

Table 5. Hyperparameters that been used on the experiments.

Hyperparameter	Value(s)
Batch Size	32, 64, 128
Learning Rate	5e-5 to 1-e5
Max Sequence Length	128, 256, 512
Warmup Steps	0 to 500
Epoch	3, 5, 10
Gradient Accumulation	1, 2, 4

During the fine-tuning process with hyperparameter in Table 5, hyperparameters such as learning rate, batch size, max sequence length, and epochs were carefully optimized to enhance the model's performance. The model was trained through multiple iterations to recognize and generate pantuns with accurate rhyme schemes and coherent thematic content. Throughout the training, perplexity was monitored as a key metric to evaluate the model's ability to predict the next word and generate natural, coherent text. The fine-tuned model was subsequently evaluated on its ability to produce pantuns that met traditional standards, ensuring both structural integrity and thematic relevance. This approach successfully enabled the GPT-2 model to generate high-quality pantuns, contributing to the preservation and generation of traditional Indonesia poetry through advanced AI techniques.

Fine-tuning is a crucial process in adapting a pre-trained model, such as GPT-2, to a specific task by further training it on a smaller, domain-specific dataset [12]. This process involves adjusting the internal representations learned during the initial pre-training to better align with the new data, which, in this study, is focused on generating pantuns a traditional form of Indonesia poetry.

The fine-tuning process for GPT-2 involves several steps, starting with the preparation of the dataset. In this case, pantuns are tokenized and converted into sequences that the model can process. The model then undergoes training with hyperparameters tailored to optimize performance on the specific task of pantun generation. Key hyperparameters include the learning rate, max sequence length, number of epochs [13], warmup steps, and gradient accumulation [14].

The learning rate is a critical hyperparameter that controls the size of the steps taken towards the minimum of the loss function during training [brown]. Choosing the correct learning rate can significantly affect the model's convergence speed and overall accuracy. Similarly, the max sequence length determines how much of the input context the model can utilize, with longer sequences capturing more context but also increasing computational complexity [15].

After fine-tuning, the model's performance is evaluated using a validation dataset to ensure that it has effectively adapted to the new task. This process ensures that the fine-tuned model can generate pantuns that adhere to traditional structures and rhyme schemes. The entire fine-tuning workflow is depicted in Figure 4.

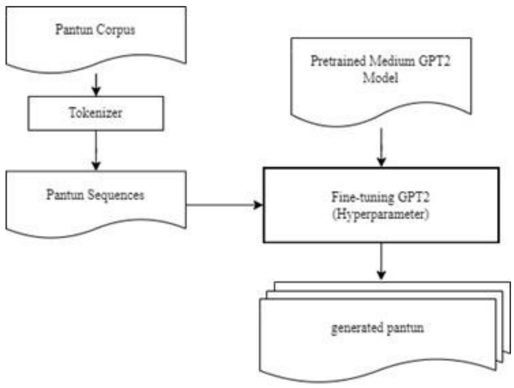

Figure 4. Fine-tuning Workflow [10]

Model

The model utilized in this research is GPT-2, specifically the GPT-2-medium-Indonesian, a language model based on the transformer architecture developed by OpenAI. The GPT-2 architecture is built upon a transformer decoder, enabling the model to generate coherent and high-quality text through a robust self-attention mechanism [16]. This model leverages a pre-training process on large-scale data, followed by fine-tuning on specific datasets to adapt to particular tasks, such as generating pantuns [15].

The GPT architecture employs the decoder portion of the transformer architecture, where input tokens are first passed through an embedding layer. This process is followed by a stack of decoder blocks, each comprising a linear layer, self-attention, feed-forward layer, and finally, a classifier layer, as illustrated in Figure 5.

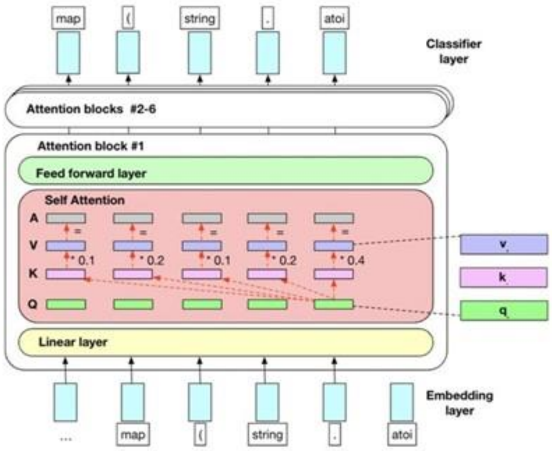

Figure 5. GPT Architecture [17]

To facilitate the fine-tuning process, this research employed the Huggingface platform. Huggingface provides access to a variety of pre-trained models and tools such as Transformers, Datasets, and Tokenizers, which simplify the implementation of transformer models in various NLP tasks. The platform integrates with frameworks like PyTorch and TensorFlow, allowing users to efficiently configure models and perform fine-tuning with ease.

The fine-tuning was conducted using Huggingface's Trainer API, which automates the configuration of hyperparameters such as learning rate, batch size, and the number of epochs [18]. In this study, the pantun dataset was converted into sequences using a tokenizer, and the model was trained to recognize and replicate the patterns present in pantuns. During the fine-tuning process, previous layers were "frozen," and only the layers relevant to the new task were retrained, as illustrated in Figure 6.

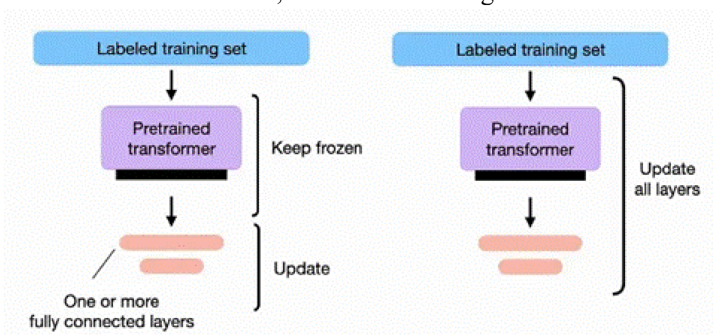

Figure 6. Fine-tuning Model [18]

The fine-tuning process is depicted in Figure 6, which shows how Huggingface's Trainer was used to train the model on the prepared dataset. After fine-tuning, the model could generate pantuns with structure and rhyme that adhere to traditional rules, as shown in Figure 7.

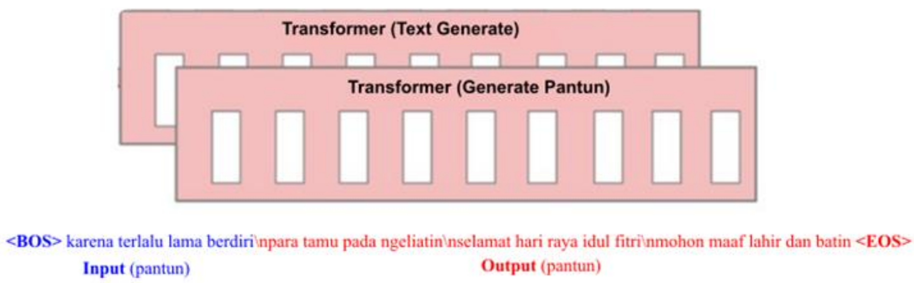

Figure 7. Model Generate Pantun [19]

Postprocessing

Post-processing is a critical component in automated text generation, particularly in ensuring the quality and accuracy of the output produced by AI models. When generating pantuns, post-processing is essential to refine the rhyme scheme, as pantuns adhere to a strict and consistent rhyme structure across lines. While models like GPT-2 can produce text with high linguistic quality, they often encounter challenges in maintaining consistent rhyme patterns, necessitating post-processing to address these issues.

The techniques employed in this research to refine the rhyme in pantuns encompass a variety of approaches, ranging from rule-based methods to machine learning algorithms. A notable method used is the parse-limerick approach, which corrects the generated rhymes by analyzing the final syllables and aligning them with the desired pattern. This approach, referenced in the work of [11], forms the foundation for the rhyme correction implemented in this study.

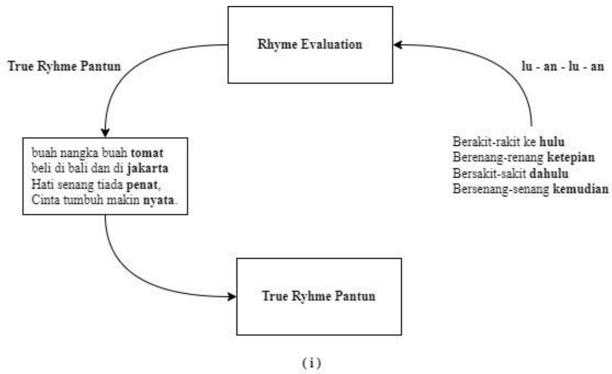

Figure 8. Post Processing Rhyme (i)

The post-processing procedure for refining rhyme begins with an initial evaluation of the generated pantuns, as depicted in Figure 8. During this phase, each pantun is examined to determine whether it conforms to the traditional rhyme pattern. If the rhyme structure is correct, the pantun is categorized as having a correct rhyme and progresses to the next evaluation stage. If the rhyme does not meet the criteria, further corrections are applied to align it with the expected structure.

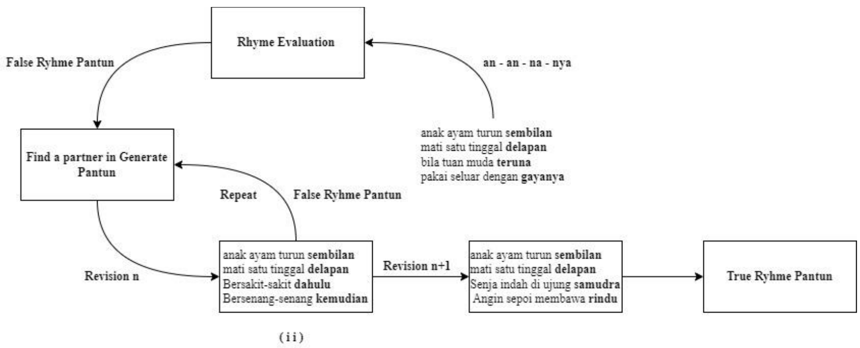

Figure 9. Post Processing Rhyme (ii)

Figure 9 illustrates the subsequent stage, where the system attempts to find a suitable rhyme pair by iterating through other generated pantuns. This iterative process continues until a correct rhyme is identified, at which point the pantun is deemed to have a correct rhyme and is accepted into the final output. However, as shown in Figure 10, if no suitable rhyme pair is found after several iterations, the pantun is categorized as having an incorrect rhyme and is set aside.

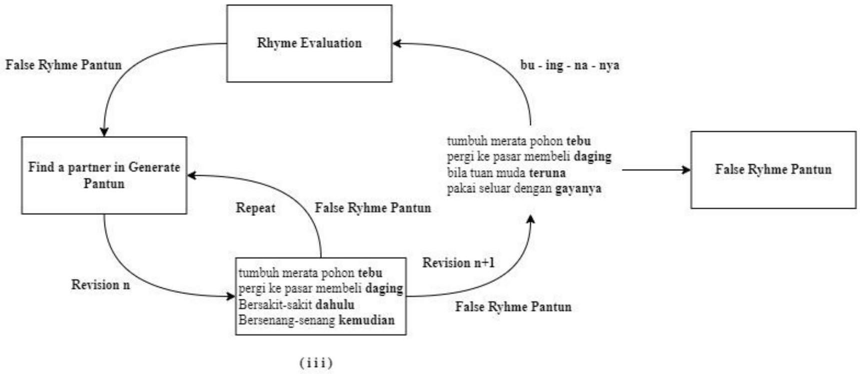

Figure 10. Post Processing Rhyme (iii)

These post-processing steps are integral to ensuring that the generated pantuns adhere to traditional structures, significantly enhancing the overall quality of the text. By refining the rhyme scheme, the post-processing phase plays a vital role in the text generation pipeline, ensuring that the output meets both linguistic and cultural expectations.

Metric Evaluation

In this study, the evaluation metrics for the performance of the pantun generator model include perplexity and rhyme accuracy.

Perplexity measures the model's ability to predict the next word in a sequence, with lower perplexity values indicating a more confident and accurate model [20]. Perplexity is calculated using the following formula 1.

$$Perplexity(\mathrm{P}) = 2^{-\frac{1}{N}\sum_{i=1}^{N}\log_2 P(w_i)} \quad (1)$$

where N is the total number of tokens in the text, $P(w_i)$ is the model's predicted probability for token w_i, and $\log_2 P(w_i)$ is the base-2 logarithm of that predicted probability. A lower perplexity value suggests that the model is better at predicting the sequence of words, reflecting a better understanding of the language's structure.

Rhyme accuracy evaluates the model's ability to produce pantuns that adhere to the traditional a-b-a-b rhyme scheme. It is calculated as the percentage of generated pantuns that correctly follow the expected rhyme pattern using the following formula 2.

$$\text{Rhyme Accuracy} = \frac{\text{Total Pantun with Correct Rhyme}}{\text{Total Generate Pantun}} \times 100\% \quad (2)$$

This metric ensures that the generated pantuns maintain the cultural and structural integrity of traditional pantun poetry [10].

Expert Validation

To complement the automated metrics, an Expert Validation was conducted, involving literary experts who assessed the generated pantuns for structural correctness, rhyme accuracy, and thematic consistency. Their evaluation ensured that the pantuns adhered to traditional rules and captured the intended cultural nuances.

This qualitative feedback was crucial in identifying areas where the model could be refined, offering insights into aspects that automated metrics might overlook, such as the creative expression and cultural relevance of the pantuns. The expert input helped ensure that the model not only performed well technically but also produced pantuns that resonated with traditional aesthetics.

RESULT AND DISCUSSION

This section presents the findings of the research and their implications, focusing on the performance of the GPT-2 model in generating pantuns and the impact of post-processing techniques. The discussion also includes a comparison with previous studies and expert validation of the results.

The fine-tuning of the GPT-2 model was conducted using the Siallagan and Khairul datasets. The training process aimed to optimize the model's ability to generate pantuns by adjusting key hyperparameters such as the learning rate, batch size, and the number of epochs. The training performance across different epochs is summarized in Table 6 and Table 7.

Table 6. Khairul Dataset Fine-tuning Results.

epoch	train loss	train steps per second	train samples per second	train runtime
2	3.8324555	0.936	15.064	00:02:32
3	2.000101	1.298	20.891	00:03:23
5	1.165142	25.16	1.564	00:03:34
10	0.350958	1.494	24.04	00:07:28

Table 7. Siallagan Dataset Fine-tuning Results.

epoch	train loss	train steps per second	train samples per second	train runtime
2	3.30160	1.62	25.911	00:13:36
3	3.094461	1.558	24.921	00:21:12
5	2.829141	1.62	25.913	00:33:59
10	2.396227	1.71	27.356	01:04:24

The training results demonstrate a clear improvement in the model's ability to generate pantuns as the number of epochs increased, particularly for the Siallagan dataset. The train loss decreased progressively with each epoch, indicating that the model was learning effectively from the dataset. The Khairul dataset, despite having fewer data points, also showed a decrease in train loss, albeit with a longer runtime and lower efficiency due to the smaller dataset size.

After fine-tuning, the model was evaluated using perplexity, a common metric in natural language processing that measures how well a model predicts a sequence of words. A lower perplexity value indicates better model performance. Table 8 and Table 9 presents the evaluation results across different epochs.

Table 8. Khairul Dataset Model Evaluation Results.

epoch	eval loss	eval runtime	Perplexity
2	3.014186143875122	00:00:05	20.372503904652483
3	3.1960973739624023	00:00:05	24.43697548442755
5	4.096014499664307	00:00:05	60.100279953853025
10	4.52395486831665	00:00:05	92.19951482220479

Table 9. Sialagan Dataset Model Evaluation Results.

epoch	eval loss	eval runtime	Perplexity
2	3.0452001094818115	00:00:25	21.01423593002522
3	2.9353342056274414	00:00:26	18.827794438685288
5	2.802616834640503	00:00:23	16.48773604658388
10	2.6837868690490723	00:00:24	14.64042984426307

The evaluation results indicate a decreasing trend in perplexity for the Siallagan dataset as the number of epochs increased, suggesting improved model performance. The lowest perplexity was observed at epoch 10, confirming it as the most effective training period. Conversely, the Khairul dataset exhibited an increase in perplexity after epoch 2, which may indicate overfitting, where the model performs well on training data but poorly on unseen data. Post-processing techniques were applied to the generated pantuns to improve their rhyme accuracy, which is crucial for adhering to the traditional structure of pantuns. The impact of post-processing is detailed in Table 10.

Table 10. Results of Rhyme Accuracy of Pantun

Dataset	Initial Generate	Post-processing1	Post-processing2
Khairul	1%	19%	52%
Siallagan	5%	45%	89%

The initial rhyme accuracy was notably low, with only 5% of the pantuns from the Siallagan dataset and 1% from the Khairul dataset meeting the correct rhyme structure. However, after applying the first round of post-processing, the rhyme accuracy improved significantly, particularly for the Siallagan dataset. A second round of post-processing further enhanced accuracy, reaching 89% for the Siallagan dataset and 52% for the Khairul dataset. These results highlight the effectiveness of post-processing in refining the model's output.

The performance of the model in this study was compared to previous research that also focused on generating pantuns using similar methodologies but without post-processing techniques. Table 11 and Table 12 summarize the comparison in terms of perplexity and rhyme accuracy.

Table 11. Perplexity Comparison.

Author	Method	Dataset	Perplexity
Khairul Faza & Putra	Fine-Tuning GPT-2	Khairul	16.941
This Study	Fine-Tuning GPT-2 + Post-processing	Khairul	20.372

Table 12. Rhyme Accuracy Comparison.

Author	Method	Dataset	Rhyme Accuracy
Siallagan & Alfina	Fine-Tuning GPT-2 + SeqGAN	Siallagan	49.70%
This Study	Fine-Tuning GPT-2 + Post-processing	Siallagan	89%

The comparison indicates that while the perplexity of the model trained in this study using the Khairul dataset was slightly higher than in the previous study by [6], the inclusion of post-processing significantly improved the rhyme accuracy, which was the primary focus of this research. The rhyme accuracy achieved with the Siallagan dataset was notably higher at 89%, compared to 49.70% reported by [10].

This improvement is attributed to the comprehensive pre-processing, careful hyperparameter tuning, and the implementation of post-processing techniques, which effectively corrected the rhyme patterns in the generated pantuns. These steps ensured that the model not only produced pantuns with correct structure and rhyme but also adhered to the cultural nuances expected in traditional pantun writing.

In addition to automated metrics, the generated pantuns were subjected to expert validation to assess their adherence to traditional pantun rules, creativity, and thematic consistency. Two experienced Indonesian language teachers evaluated the pantuns before and after post-processing. With the validation results shown in Table 13.

Table 13. Expert Validation Scores.

Aspect	Before Postprocessing	After Postprocessing
Adherence to Pantun Rules	1.5	5
Creativity in Composition	1.5	5
Beauty of Expression	1.5	5
Clarity in Message Delivery	1.5	5
Ease of Using the Pantun Generator	5	5

Expert validation further confirmed these improvements. Initially, the pantuns without post-processing were rated poorly in terms of rhyme, creativity, and adherence to traditional rules. However, after post-processing, the pantuns were rated highly across all evaluation criteria, achieving perfect scores in most aspects. This validates the importance of post-processing in producing high-quality, culturally accurate pantuns.

The significant increase in rhyme accuracy and the high validation scores from experts highlight that the methodology applied in this study effectively enhances the quality of generated pantuns. The improvements suggest that future research on pantun generation should consider incorporating comprehensive pre-processing, optimal hyperparameter tuning, and post-processing techniques to achieve similar or even better results.

CONCLUSION

The conclusion of this research demonstrates the effectiveness of fine-tuning the GPT-2 model for generating pantuns, particularly when using datasets such as Siallagan and Khairul. The study successfully utilized a structured approach, involving data preprocessing, strategic hyperparameter tuning, and post-processing techniques, to enhance the model's performance. The fine-tuned model achieved a perplexity of 14.640 and a rhyme accuracy of 89% with the Siallagan dataset, showcasing its capability to generate high-quality pantuns that adhere to traditional structures. Expert validation further confirmed the significant improvement in the quality of the generated pantuns, with validation scores increasing from 1.5 to 5 after post-processing. This research not only underscores the potential of GPT-2 in cultural text generation but also highlights the importance of combining fine-tuning with targeted post-processing to achieve optimal results. The impact of this study lies in its contribution to advancing natural language generation techniques, particularly in preserving and enhancing the traditional literary form of pantuns through modern AI tools.

REFERENCES

[1] H. Lebu, D. Wardiah, and M. Indasari, "ANALISIS KEMAMPUAN MENULIS PANTUN SISWA KELAS V SD," *Indonesian Journal of Elementary Education (IJOEE)*, vol. 1, no. 2, Aug. 2020, doi: 10.31000/ijoee.v1i2.2934.

[2] Multafifin, "Kemampuan Menulis Pantun Siswa Kelas VII SMP Negeri 52 Konawe Selatan," *Jurnal Hamanika*, vol. 3, no. 15, Dec. 2015.

[3] D. E. Maulina, "Keanekaragaman pantun di Indonesia," *Semantik*, vol. 1, no. 1, 2012.

[4] A. Pebrianto and M. H. Prof. Dr. Markhamah, "Analisis Kesulitan Siswa Dalam Menulis Pantun Sesuai Dengan Syarat Pantun Di SMP Negeri 16 Surakarta," 2017.

[5] T. B. Brown *et al.*, "Language Models are Few-Shot Learners," *Advances in neural information processing systems 33*, pp. 1877–1901, 2020, [Online]. Available: https://commoncrawl.org/the-data/

[6] R. Khairul Faza and D. H. Putra, "IMPLEMENTASI GENERATIVE PRE-TRAINED TRANSFORMERS SEBAGAI PANTUN GENERATOR," 2023.

[7] Misnawati Misnawati, "ChatGPT: Keuntungan, Risiko, Dan Penggunaan Bijak Dalam Era Kecerdasan Buatan," *PROSIDING SEMINAR NASIONAL PENDIDIKAN, BAHASA, SASTRA, SENI, DAN BUDAYA*, vol. 2, no. 1, pp. 54–67, Apr. 2023, doi: 10.55606/mateandrau.v2i1.221.

[8] E. Latif and X. Zhai, "Fine-tuning ChatGPT for automatic scoring," *Computers and Education: Artificial Intelligence*, vol. 6, p. 100210, Jun. 2024, doi: 10.1016/j.caeai.2024.100210.

[9] G. Vrbancic and V. Podgorelec, "Transfer Learning With Adaptive Fine-Tuning," *IEEE Access*, vol. 8, pp. 196197–196211, 2020, doi: 10.1109/ACCESS.2020.3034343.

[10] E. A. Siallagan and I. Alfina, "Poetry Generation for Indonesian Pantun Using SeqGAN and GPT-2," *Jurnal Ilmu Komputer dan Informasi*, vol. 16, no. 1, pp. 59–67, Mar. 2023, doi: 10.21609/jiki.v16i1.1113.

[11] K.-L. Lo, R. Ariss, and P. Kurz, "GPoeT-2: A GPT-2 Based Poem Generator," May 2022, [Online]. Available: http://arxiv.org/abs/2205.08847

[12] L. Zhang, X. Hu, B. Wang, D. Zhou, Q.-W. Zhang, and Y. Cao, "Pre-training and Fine-tuning Neural Topic Model: A Simple yet Effective Approach to Incorporating External Knowledge," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2022, pp. 5980–5989. doi: 10.18653/v1/2022.acl-long.413.

[13] C. Raffel *et al.*, "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020, [Online]. Available: http://jmlr.org/papers/v21/20-074.html

[14] M. Ott, S. Edunov, D. Grangier, and M. Auli, "Scaling Neural Machine Translation," Jun. 2018.

[15] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language Models are Unsupervised Multitask Learners," 2019. [Online]. Available: https://github.com/codelucas/newspaper

[16] A. Vaswani *et al.*, "Attention Is All You Need," Jun. 2017.

[17] G. A. Aye, S. Kim, and H. Li, "Learning Autocompletion from Real-World Datasets," in *2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, IEEE, May 2021, pp. 131–139. doi: 10.1109/ICSE-SEIP52600.2021.00022.

[18] B. Banjara, “A Comprehensive Guide to Fine-Tuning Large Language Models,” Aug. 2024. [Online]. Available: https://www.analyticsvidhya.com/blog/2023/08/fine-tuning-large-language-models/

[19] Z. Guo, P. Wang, Y. Wang, and S. Yu, “Improving Small Language Models on PubMedQA via Generative Data Augmentation,” May 2023.

[20] Y. Li, “Estimating Contamination via Perplexity: Quantifying Memorisation in Language Model Evaluation,” Sep. 2023.