A REPORT ON

Refining Creative Writing Using LLMs

Birla Institute of Technology and Science, Pilani
April 2025

REPORT ON

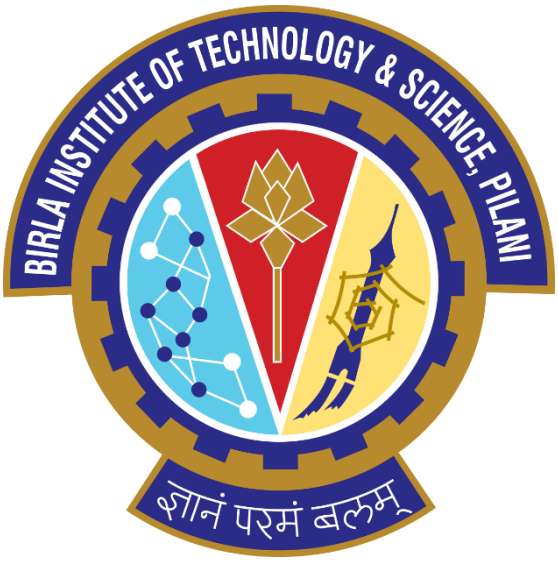
Refining Creative Writing Using LLMs

Submitted in partial fulfillment of the requirements of the course,
CS F320 Foundations of Data Science

Submitted By,

Swapnil Yadav	2021B3A72770P
Ishan Wani	2021B3A72772P
Riddhi Agarwal	2021B3A71117P
Aritra Khatua	2022A7PS0184P

Under the supervision of,
Prof. Tejasvi Alladi
Dept. of Computer Science



Birla Institute of Technology and Science, Pilani
April 2025

INTRODUCTION

This project, titled *"Refining Creative Writing using LLMs"*, focuses on evaluating the various methods of text generation using Large Language Models (LLMs) for specific imitation tasks to understand which approach works the best. This is done by creating an agent for the use case of generating content in the writing styles of specific authors. The agent utilizes LLMs **to generate short stories that reflect the writing styles of specific, well-known authors. The primary goal is to enable users to input a basic prompt—such as a theme or scenario—and receive a coherent, original story aligned with a selected author's stylistic features.**

The project investigates three approaches to achieve this: **prompt-based generation**, **retrieval-augmented generation (RAG)**, and **fine-tuning**. A comparative study is conducted to evaluate these methods in terms of narrative consistency, fidelity to authorial style, and variation in output. The system also incorporates a **vision-language model (VLM)** to produce illustrations that align with key story elements, adding a visual layer to the narrative output.

The motivation behind this work lies in the increasing demand for personalized content and the growing role of AI in creative applications. By studying the strengths and limitations of different model architectures, the project aims to determine how much control users can exert over narrative elements, especially style, without compromising originality or coherence. This has applications in **education**, **publishing**, and **entertainment**, where **AI-generated writing** could support ideation, drafting, and visualization tasks.

The development process begins with dataset preparation, drawing from publicly available sources like Project Gutenberg. Texts are manually organized to align with fine-tuning requirements. Fine-tuning is carried out using **Gemini-1.5-Flash**, with model performance evaluated using synthetically constructed test prompts. In parallel, a RAG-based agent is created using an evaluator-optimizer structure involving two LLMs. ChromaDB is used as the vector database, **Langchain manages LLM orchestration**, and Chain**lit** provides a simple interface for user interaction. For visual augmentation, illustrations are generated using a VLM such as Stable Diffusion.

Evaluation is a critical part of the workflow. Stylometric methods are used to measure how closely the generated text resembles the style of the source author. These include Mendenhall's Characteristic Curves, Kilgariff's Chi-Squared Method, Burrows' Delta, and cosine similarity metrics based on static embeddings. Each of these methods allows for a quantifiable comparison between outputs from the three different generation techniques.

ABSTRACT

This project presents a new approach to AI-powered short story generation, with a focus on replicating the distinctive writing styles of specific authors. We explore and compare three core methodologies: prompt-based generation, retrieval-augmented generation (RAG), and fine-tuning of large language models (LLMs). A custom dataset was meticulously constructed by manually curating public domain literary excerpts from authors Lydia Davis and Diane Williams. Their works were formatted to support supervised fine-tuning, ensuring alignment with both stylistic and narrative conventions.

The system was implemented using Google's Gemini model through the LangChain framework and supported by a vision generation component using Stable Diffusion to generate contextual illustration for the generated text. A memory retention feature is incorporated into the agent in order to enable the user to iteratively refine story quality along with dual-rag calls to enhance performance in RAG and model-switching contexts.

To evaluate performance, we employed quantitative stylometric analysis using four established metrics:

- **Mendenhall's Characteristic Curves** (word length distributions)
- **Kilgariff's Chi-Squared Method** (word frequency deviations)
- **Burrows' Delta Method** (normalized word frequency distance)
- **Non-contextual Cosine Similarity** using static embeddings

- **BERT Score**

Experimental results demonstrated that **fine-tuning consistently outperformed both prompt-based and RAG approaches** across all metrics. Fine-tuned models exhibited higher stylistic fidelity, better narrative coherence, and greater lexical alignment with the target authors. In contrast, prompt-based generation often lacked depth and consistency, while RAG outputs struggled to maintain stylistic integrity due to dependency on external context.

The project underscores the **superiority of fine-tuning for author-style emulation** and offers a replicable framework for combining LLMs, stylometric evaluation, and VLMs in creative generation tasks. These findings are especially relevant for applications in **literary AI**, **automated content creation**, and **digital humanities**, where control over style and structure is critical.

OBJECTIVES

- Develop an AI-powered story generation system that produces short stories based on user prompts and themes.
- Replicate the writing styles of famous authors using various LLM approaches, ensuring consistent output with the chosen author.
- Implement and compare three methodologies for story generation:
    - Prompt-based generation
    - Retrieval-Augmented Generation (RAG)
    - Fine-tuning a language model
- Integrate a Vision-Language Model (VLM) to generate AI-driven illustrations for key story moments.
- Conduct a comparative analysis of the three approaches using stylometric metrics to assess: Fidelity to writing, Creativity, Consistency and coherence.
- Design an evaluator-optimizer agent framework using LLMs to iteratively improve story quality.
- Build a Chainlit-based UI for interactive input, generation, and illustration visualization.
- Collect and format a custom literary dataset for training and evaluation.
- Evaluate performance with stylometry techniques like Mendenhall's, Kilgariff, Burrows' Delta, and Cosine Similarity.

METHODOLOGY

The methodology for this project is structured around three core phases: dataset preparation, model development, and evaluation. Each phase is designed to ensure the generation of coherent, stylistically faithful short stories, accompanied by relevant illustrations.

1. Dataset Creation

Public domain literary works were manually collected from sources such as Project Gutenberg and the Internet Archive. Texts from selected authors were cleaned and curated to ensure consistency in formatting and then structured to align with the input-output format required for fine-tuning language models. The dataset follows a simple question-answer format which is fed into the LLM in the form of a python list. A small sample dataset is provided below

```
training_data = [
  {"text_input": "1", "output": "2"},
  {"text_input": "3", "output": "4"},
  {"text_input": "-3", "output": "-2"},
  {"text_input": "three", "output": "four"},
  {"text_input": "seven", "output": "eight"},
]
```

The dataset can also be fed into the LLM in the form of a JSON list.

2. Fine-Tuning and Verification

Fine-tuning was conducted using a Python-based pipeline. Hyperparameters were adjusted experimentally to achieve optimal model performance. The resulting models were tested on a synthetically generated test set to assess their ability to generalize and preserve authorial style. The following hyperparameters were considered for fine-tuning:

Generation id = randInt(),

epoch_count = 100,

batch_size=4,

learning_rate=0.001,

3. Agent Creation: Framework

The agent was created with a python backend, complete with Chainlit for the User Interface (UI). A memory retention feature is incorporated into the agent in order to enable the user to iteratively refine story quality along with dual-rag calls to enhance performance in RAG and model-switching contexts The following are the salient features of the agent:

- Dual LLMs: There nested LLM calls for RAG based approach to improve the performance of RAG model. One call was for generation of the story and the other call was for the extracting context from the generated story.

- Vision-Language Model (VLM): Stable Diffusion was used to generate illustrations based on key narrative moments. The separate VLM prompt was created for the purpose of image generation.

4. Evaluation

The outputs from all three approaches (prompt-based, RAG, and fine-tuned) were evaluated using stylometric analysis to measure fidelity to the original author's writing style. Key metrics used for this comparison included:

- **Mendenhall's Characteristic Curves**: Analyses the distribution of word lengths to identify stylistic patterns unique to an author.
- **Kilgariff's Chi-Squared Method**: Measures statistical differences in word frequency distributions to compare writing styles.
- **Burrows' Delta Method**: Calculates stylistic distance between texts using normalized word frequency profiles.
- **Non-contextual Cosine Similarity** using static embeddings: Compares textual similarity based on vector representations of words, ignoring context.
- **BERT Score**: It evaluates the semantic similarity between LLM generated content and reference text, ensuring content aware quality assessment.

These quantitative metrics enabled a detailed comparison of the models' performance in terms of stylistic accuracy, narrative coherence, and creativity. The detailed discussion of the metrics is provided in the following section.

RESULTS AND EVALUATION METRIC

Lydia Davis

- Fined Tuned performs the best across all metrics, indicating strong lexical, stylistic, and semantic alignment:
    - Chi-Squared drops significantly from 1837.5 (LLM) to 1004.77.
    - Delta improves from 1.385 to 1.164.
    - Cosine Similarity rises to 0.985, and BERT Score peaks at 0.8118.
- RAG is a good middle ground but not as effective as Fine Tuned.

Diane Williams

- Fine Tuned again shows significant improvements:
    - Chi-Squared: 1509.89 → 669.26.
    - Delta: 1.378 → 0.956.
    - Cosine Similarity and BERT Score also show incremental improvements.
    - Cosine Similarity reaches 0.991, the highest among all.

Lydia Davis	**LLM**	**RAG**	**Fine tune**
Kilgariff's Chi Square	1837.5	1568.73	1004.77
Burrow's Delta	1.385	1.272	1.164
Cosine Similarity	0.936	0.948	0.985
BERT Score	0.7939	0.7955	0.8118

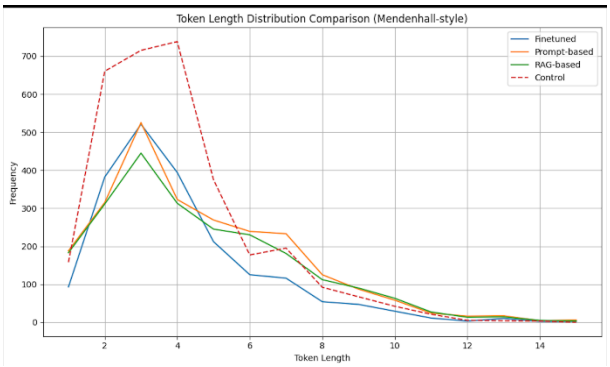
Table of results of evaluation for author Lydia Davis

Diane Williams	**LLM**	**RAG**	**Fine tune**
Kilgariff's Chi Square	1509.89	1405.25	669.26
Burrow's Delta	1.378	1.275	0.956
Cosine Similarity	0.969	0.974	0.991
BERT Score	0.8069	0.8001	0.8124

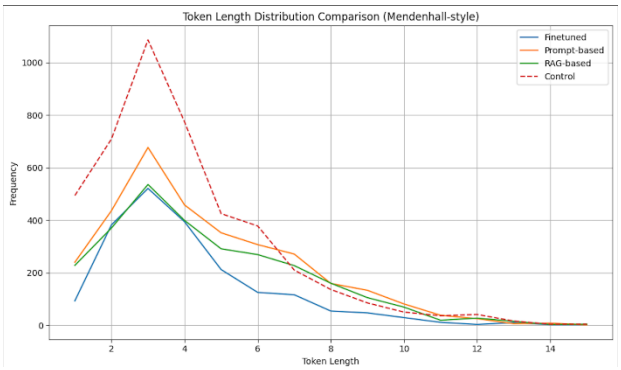
Table of results of evaluation for author Diane Williams

Token Length Distribution Comparison (Mendenhall-style):

The control text shows a sharp peak at token lengths 3–4, reflecting a consistent and concise lexical style. Among the generated outputs, the fine-tuned model most closely replicates this distribution, especially in the mid-range, indicating higher stylistic fidelity. In contrast, both the prompt-based and RAG-based methods exhibit a broader spread and heavier tails, suggesting a tendency toward longer word use and slight stylistic drift.



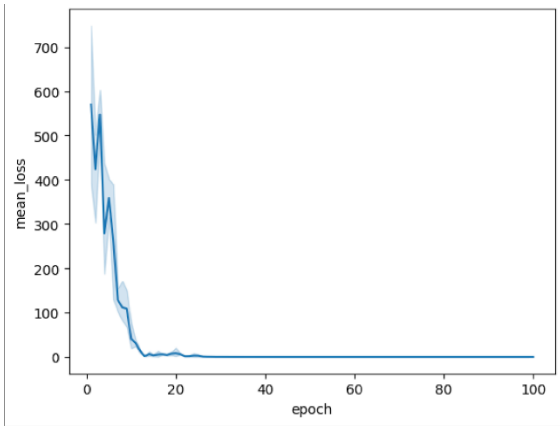
Lydia Davis Mendenhall-style evaluation



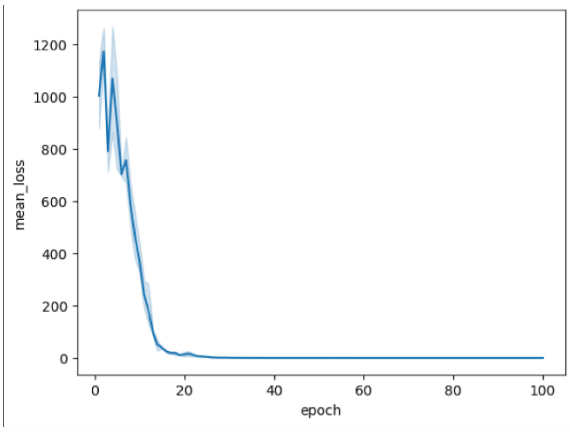
Diane Williams Mendenhall-style evaluation

Fine-tuning Loss Curve:

The model shows rapid convergence, with mean loss dropping sharply within the first 15 epochs and stabilizing near zero thereafter. This indicates effective fine-tuning and suggests that the model successfully learned the stylistic features of Lydia Davis's and Diane Williams's writing with minimal overfitting.

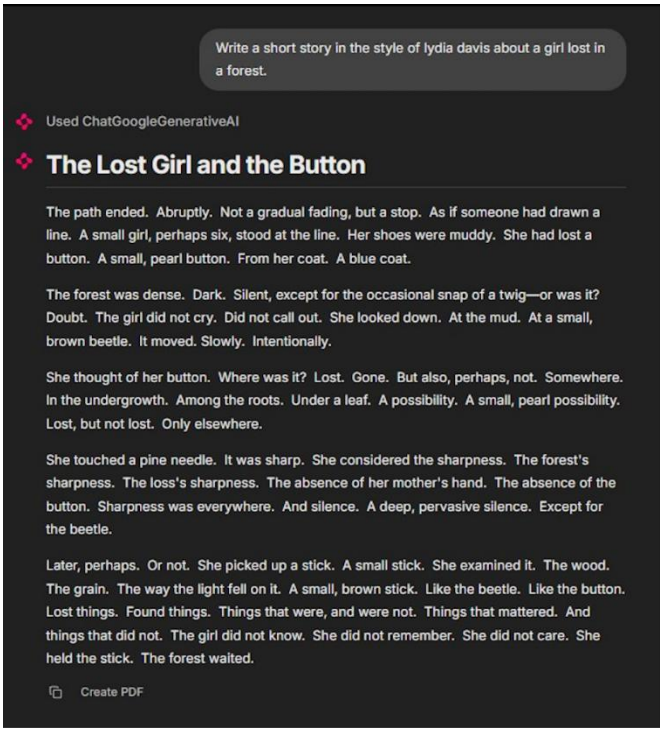


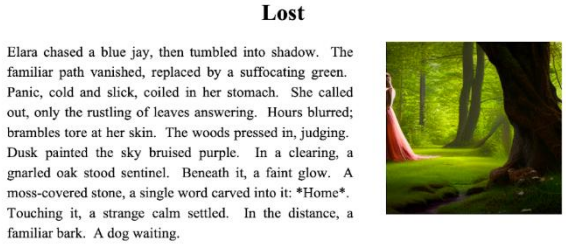
Lydia Davis Loss Function



Diane Williams Loss Function

OUTPUT SCREENSHOTS





CONCLUSION

This project successfully demonstrates the potential of leveraging large language models for stylized short story generation, highlighting the comparative advantages of different model architectures and training paradigms. Through a systematic exploration of prompt-based generation, retrieval-augmented generation (RAG), and fine-tuning, we established that fine-tuning offers the most robust and reliable approach for replicating complex authorial styles with high fidelity.

By building a custom dataset of curated literary excerpts and integrating advanced tools like evaluator–optimizer agent workflows and vision-language models (VLMs) for illustration, the system transcends basic text generation to provide a rich, multimodal storytelling experience. Fine-tuned models consistently outperformed other methods across multiple stylometric evaluation metrics, producing narratives that were not only coherent and creative but also strongly aligned with the stylistic markers of the target authors.

This work contributes a scalable and adaptable framework for author-aware AI generation, offering practical value in domains such as literary emulation, educational content creation, and AI-assisted creative writing. Future directions include expanding the author set, incorporating multilingual texts, and enhancing agent feedback mechanisms with human-in-the-loop refinement for even greater quality control and personalization.

REFERENCES

Are Large Language Models Capable of Generating Human-Level Narratives?, https://arxiv.org/pdf/2407.13248v2

Do Language Models Enjoy Their Own Stories? Prompting Large Language Models for Automatic Story Evaluation, https://arxiv.org/pdf/2405.13769v1

How good is my story? Towards quantitative metrics for evaluating LLM-generated XAI narratives, https://arxiv.org/pdf/2412.10220v1

Stylometry in Python, https://programminghistorian.org/en/lessons/introduction-to-stylometry-with-python#prior-reading

Fine-tuing with the Gemini API, https://ai.google.dev/gemini-api/docs/model-tuning

Langchain Google Generative AI, https://python.langchain.com/api_reference/google_genai/index.html