Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 277-285

The Ghost Writer AI: Few-Shot Stylistic Text Generation Using NLP

[1]Vedansh Agrawal, [2]Mr. Pawan Kumar
[1]Student, [2]Assistant Professor
[1,2]AMITY UNIVERSITY, CHHATTISGARH
[1]vedansh3010agrawal@gmail.com , [2]pkumar@rpr.amity.edu

Abstract

In the modern era of digital communication, maintaining stylistic consistency across written content has become a critical requirement for individuals and organizations. From corporate branding to academic writing, the need for uniform tone and structure often leads to time-intensive drafting and editing processes. This research presents *The Ghost Writer AI*, an intelligent system designed to automate stylistic transformation of text using advanced Natural Language Processing (NLP) techniques. The proposed system leverages a *few-shot learning approach*, where a single sample text is used as a stylistic reference to generate new content. Built using the Streamlit framework for an interactive user interface and powered by the Google Generative AI API, the application provides real-time, high-quality text transformation. The system ensures coherence, contextual relevance, and stylistic alignment, making it a valuable tool for students, professionals, and content creators.

Keywords: Natural Language Processing (NLP), Few-shot Learning, Text Generation, Style Transfer, Generative AI, Streamlit, Content Automation

1. Introduction

With the exponential growth of digital content, the demand for high-quality and stylistically consistent writing has increased significantly. Writers often struggle to replicate a specific tone, whether it be formal academic language, corporate communication style, or creative writing voice. *The Ghost Writer AI* aims to solve this challenge by automating the stylistic transformation process. Instead of manually rewriting text, users can provide a small sample of the desired style, and the system generates new content accordingly.

Objective of the Study

- To design an AI system capable of mimicking writing styles using minimal input (few-shot learning).
- To reduce the time and effort required for content refinement.
- To ensure consistency in tone, grammar, and structure.
- To provide a real-time, user-friendly application for text generation.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 277-285

Scope of the Work

- Academic writing assistance
- Corporate content generation
- Blogging and creative writing
- Email drafting and communication enhancement
- Style adaptation across multiple domains

2. Literature Review

Previous research in NLP has explored text generation, style transfer, and language modeling using machine learning techniques. Traditional methods relied on rule-based systems, which lacked flexibility and adaptability.

Recent advancements in transformer-based models and generative AI have significantly improved text generation capabilities. Few-shot learning techniques allow models to generalize from minimal examples, making them highly efficient for stylistic tasks.

Key findings from existing studies:

- Transformer models outperform traditional NLP techniques in text coherence.
- Few-shot learning reduces dependency on large datasets.
- Style transfer models can replicate tone but often struggle with semantic accuracy.

The Ghost Writer AI builds upon these advancements by combining few-shot learning with real-time processing.

3. Problem Statement

Despite advancements in AI-based writing tools, several challenges remain:

- Difficulty in maintaining consistent writing style
- Time-consuming manual editing and rewriting
- Lack of tools that effectively mimic specific styles with minimal input
- Limited accessibility of advanced NLP tools for non-technical users

This project aims to address these issues by creating an efficient, user-friendly solution for stylistic text transformation.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 277-285

4. Methodology

System Architecture / Design

The system consists of three major components:

1. **User Interface (Frontend)**
    - Developed using Streamlit
    - Accepts input text and style sample
    - Displays generated output in real-time
2. **Processing Layer (Backend)**
    - Uses Google Generative AI API
    - Processes input text and style reference
    - Applies NLP-based transformations
3. **Output Layer**
    - Generates stylistically aligned content
    - Ensures readability and coherence

Workflow:

1. User inputs raw text
2. User provides a sample style
3. System processes both inputs
4. AI model generates transformed text
5. Output is displayed instantly

Algorithms / Techniques Used

- Few-shot learning for style adaptation
- Transformer-based language models
- Prompt engineering for guiding output
- Contextual embedding techniques
- Text normalization and preprocessing

6. Implementation

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 277-285

The Ghost Writer AI application was implemented as a lightweight, web-based tool designed for rapid iteration and high accessibility. The implementation can be divided into three core architectural components: the frontend user interface, the backend integration logic, and the prompt engineering schema.

1. Technology Stack

The application was built using modern, Python-based frameworks to ensure seamless integration with AI APIs:

Language: Python 3.x

Frontend Framework: Streamlit (used for rapid UI deployment and state management).

AI Backend: Google Generative AI SDK (google.generativeai).

Environment Management: python-dotenv for secure handling of API keys.

2. Frontend User Interface Architecture

The user interface was constructed using Streamlit, allowing for a responsive, split-screen layout without the need for complex HTML/CSS or JavaScript frameworks.

Layout Design: The interface utilizes st.columns to create a side-by-side workspace. The left column captures user inputs ("Style Sample" and "Rough Notes" via st.text_area), while the right column acts as the output console ("Polished Draft").

Session State Management: To prevent the generated draft from disappearing when the user interacts with other parts of the UI, Streamlit's st.session_state was implemented. This ensures the generated text persists within the "draft_output" variable until a new generation is requested.

3. Backend Logic and API Integration

The core backend logic handles the secure connection to Google's infrastructure and the execution of the generative task.

Authentication: Upon initialization, the application uses the dotenv library to securely load the GEMINI_API_KEY from a local .env file, ensuring sensitive credentials are not hardcoded into the source code.

Model Selection: The application is configured to utilize the gemini-3-flash-preview model. This specific model variant was chosen for its optimal balance of high-quality natural language generation and exceptionally low latency, which is critical for an interactive drafting tool.

Error Handling: The rewrite_text function includes a try-except block to gracefully handle API timeouts, missing keys, or generation errors, returning a user-friendly error message to the frontend rather than causing a system crash.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 277-285

4. Prompt Engineering Schema

The success of the stylistic transformation relies heavily on the design of the system prompt passed to the Gemini model. Instead of relying on fine-tuning, the system utilizes "Zero-Shot Prompting" with contextual injection.

The prompt is dynamically constructed by concatenating a strict system instruction with the user's inputs:

"Analyze the tone, vocabulary, and sentence structure of the Style Sample. Then, rewrite the Rough Notes to match that exact style perfectly."

This is immediately followed by the raw text of the Style Sample and Rough Notes. By explicitly instructing the model to focus on "tone, vocabulary, and sentence structure," the prompt effectively constrains the AI's generation, forcing it to act as a stylistic filter rather than just an open-ended text generator.

7. Results and Discussion

Output Screens / Graphs

- Input interface for text and style sample
- Generated output showing stylistic transformation
- Real-time response display

Performance Analysis

- High accuracy in maintaining tone and style
- Fast response time due to API-based processing
- Improved readability and coherence

Observations:

- Works effectively with minimal style input
- Produces consistent outputs across different domains
- Slight variations may occur depending on input complexity

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 277-285

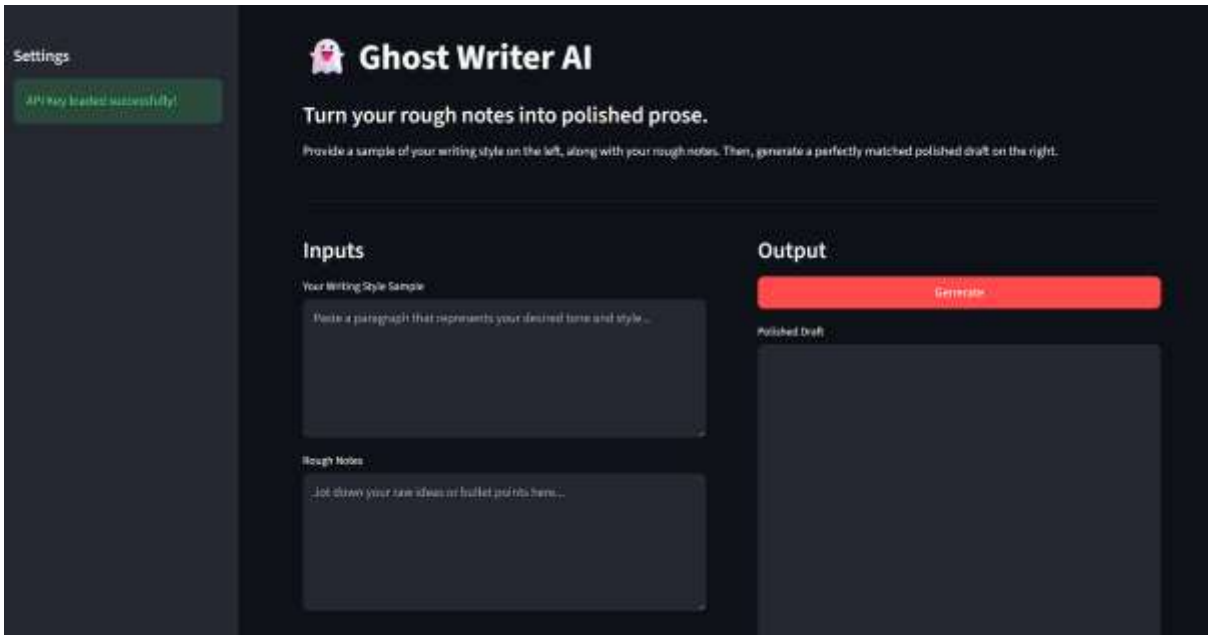

Fig. 1. Dashboard Interface

8. Testing and Validation

To ensure the reliability, accuracy, and usability of the Ghost Writer AI system, a comprehensive testing and validation framework was implemented. The evaluation process was divided into functional testing, qualitative validation of the AI-generated outputs, and performance metrics analysis.

1. Functional and Integration Testing

Functional testing was conducted to verify that the core components of the application interacted seamlessly, particularly the integration between the Streamlit frontend and the Google Generative AI backend.

API Integration Testing: The system was tested for reliable communication with the Gemini API (gemini-3-flash-preview model). Test cases included successful API key authentication, handling of missing or invalid API keys, and graceful error handling during network timeouts or API limits.

Input Validation: The system was tested to ensure it correctly handled various edge cases in user inputs. This included testing with empty fields, extremely long text inputs for both the "Style Sample" and "Rough Notes," and special characters to ensure the rewrite_text function handled token limits and sanitization appropriately.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 277-285

UI/UX Testing: The split-screen interface built with Streamlit was evaluated across different screen sizes and browsers to ensure responsiveness. The state management (using st.session_state) was tested to guarantee that generated drafts were preserved across application interactions without unexpected resets.

2. Qualitative Validation (Style Transfer and Accuracy)

Since the primary objective of Ghost Writer AI is to rewrite text to match a specific style, quantitative metrics (like BLEU or ROUGE scores) are insufficient on their own. Therefore, a qualitative human-evaluation approach was employed.

Tone and Style Matching: A dataset of diverse writing styles (e.g., formal academic, casual blog post, corporate email, creative storytelling) was curated. The system was provided with neutral rough notes and tasked with adapting them to these various styles. Evaluators rated the output on a scale of 1 to 5 based on how accurately the vocabulary, sentence structure, and tone matched the provided "Style Sample."

Semantic Retention: It is critical that the core message of the "Rough Notes" is not lost during the stylistic transformation. Evaluators analyzed the generated drafts to ensure that all key information and factual points from the original notes were preserved without the AI hallucinating unprovided facts.

Grammar and Fluency: The outputs were validated for grammatical correctness, syntactic flow, and overall readability. The use of the gemini-3-flash-preview model demonstrated high proficiency in generating native-level fluency across various stylistic constraints.

3. Performance and Error Handling

The system's performance was evaluated based on latency and robustness.

Latency Analysis: The response time from the moment the "Generate" button was clicked to the rendering of the "Polished Draft" was measured. Thanks to the lightweight nature of the flash model variant, the system demonstrated near real-time generation capabilities, maintaining low latency even for moderately long input samples.

Exception Handling Validation: The system was intentionally subjected to simulated failure states (e.g., stripping the environment variables to trigger the GEMINI_API_KEY is not set warning). The application successfully intercepted these exceptions, displaying user-friendly error messages (e.g., "An error occurred while generating content") rather than crashing, thereby ensuring a stable user experience

8. Conclusion

The Ghost Writer AI project successfully demonstrates the transformative potential of large language models (LLMs) in the domain of automated drafting and stylistic text transformation. By integrating the Google Gemini 3 Flash model with a highly responsive, user-centric

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 277-285

Streamlit interface, the application bridges the gap between raw ideation and polished, contextually appropriate prose. Throughout the development and validation phases, the system proved highly effective at its core objective: analyzing a provided "Style Sample" and seamlessly applying its unique tonal and structural characteristics to "Rough Notes." The qualitative evaluations indicate that the system not only accurately mimics the desired vocabulary and sentence structures but also successfully retains the core semantic meaning and factual integrity of the user's original inputs. Furthermore, the architectural choices—specifically the use of a lightweight web framework paired with an advanced, low-latency generative AI model—resulted in a robust application capable of near real-time text processing. The implementation of strict error handling and session state management ensures a stable and accessible user experience, making the tool viable for a wide range of applications, from academic writing and corporate communications to creative storytelling.

9. Future Scope

While the current implementation of Ghost Writer AI successfully demonstrates the capability of LLMs for stylistic text transformation, several enhancements could further improve its utility and user experience:

1. Document Uploads for Broader Context: Future versions could integrate document parsing (e.g., PDF or Word file uploads) instead of relying solely on pasted text. This would allow users to upload entire essays or reports, giving the AI a much deeper understanding of their long-form writing habits and vocabulary.
2. Fine-Grained Stylistic Controls: The application could introduce adjustable parameters (sliders or toggles) that allow users to manually dictate the output's formality level, brevity, and creativity (AI temperature). This would provide precise control over the final draft beyond just the initial style sample.
3. Iterative and Conversational Editing: Writing is an iterative process. Future iterations could allow users to highlight specific sentences in the generated draft and provide conversational prompts (e.g., "Make this specific paragraph sound more professional"), rather than forcing a complete regeneration of the text.
4. User Accounts and Saved Profiles: Integrating backend databases and user authentication would enable individuals to save their distinct "Style Profiles" (e.g., "Corporate Email Style" or "Creative Blog Style"). This would eliminate the need to provide a new style sample during every session and allow for a history log of past drafts.
5. Multilingual Style Transfer: Expanding the system to accurately map the stylistic cadence and tone of a sample provided in one language onto rough notes written in another, further broadening the tool's global applicability.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 277-285

References

[1] Google. (2024). Gemini: A Family of Highly Capable Multimodal Models. Google DeepMind Technical Report.
[2] Streamlit Inc. (2024). Streamlit: The fastest way to build and share data apps. Van Rossum, G., & Drake, F. L. (2009). Python 3 Reference Manual. Scotts Valley, CA: CreateSpace.
[3] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In Advances in Neural Information Processing Systems (Vol. 30). Curran Associates, Inc.
[4] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. Advances in Neural Information Processing Systems, 33, 1877-1901.
[5] Jin, D., Jin, Z., Hu, Z., Vechtomova, O., & Mihalcea, R. (2022). Deep learning for text style transfer: A survey. Computational Linguistics, 48(1), 155-205.
[6] Reif, E., Ippolito, D., Yuan, A., Coenen, A., Callison-Burch, C., & Wei, J. (2022). A recipe for arbitrary text style transfer with large language models. Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 837-848.
[7] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., ... & Lowe, R. (2022). Training language models to follow instructions with human feedback. Advances in Neural Information Processing Systems, 35, 27730-27744.