Style+Control: Disentangling Literary Style from Semantic Content in Vector Space

Alex Tulip

Department of Computer Science, Università degli Studi di Milano,
Milan, Italy.

Abstract

This study investigates how literary style is encoded in the vector space of Large Language Models (LLMs). I specifically look at whether I can separate the distinct writing styles of Jane Austen and Herman Melville. Using a dataset of non-parallel texts and the BGE embedding model, I show that authors form distinct clusters in high-dimensional space (99% classification accuracy). However, I find that "style" is not a single point but a broad cloud. I also performed a style transfer experiment using Gemini-2.5-Flash. My results show a conflict between content and style: the final semantic layers of the model focus on the original content and fail to detect the style change. In contrast, the lower layers (Layer 2) successfully identify the new style. This suggests that literary style is encoded in the early structural layers of the model, separate from the high-level meaning found in later layers.

1 Introduction

Natural Language Processing (NLP) has become very good at understanding meaning. Modern models like BERT or RoBERTa are trained to see sentences like "The dog ate" and "The canine fed" as almost identical because they mean the same thing. However, in literature, the *difference* between those two sentences—the style—is just as important as the meaning. As noted in recent surveys on Neural Style Transfer (NST) [1], defining "style" is much harder and more subjective than defining content.

In this project, I address a specific question: **Is literary style stored in a separate region of the vector space in modern LLMs, and can I move text from one style to another without losing the original meaning?**

To test this, I chose two authors with very different voices: Jane Austen (known for polite society and domestic realism) and Herman Melville (known for nautical terms and complex, archaic language). I used text from *Emma* and *Moby Dick* to see if their styles could be separated and if an LLM could rewrite Austen's text in Melville's voice.

2 Methodology

My methodology follows three main steps: data preparation, embedding analysis, and style transfer generation.

2.1 Dataset Curation

Since there is no existing dataset where Melville rewrote Austen, I created a non-parallel dataset using the Project Gutenberg versions of *Emma* and *Moby Dick*. I split the raw text into 600 chunks (300 per author), each about 128 tokens long. This length ensures the model has enough context to detect the style.

2.2 Models

- **Embedding:** I used `BAAI/bge-small-en-v1.5`. While Pan et al. [3] used RoBERTa, I chose BGE because it is optimized for semantic retrieval. Using a model designed to focus on *meaning* is a strong test: can style differences survive in a vector space designed to ignore them?
- **Generation:** For the style transfer task, I used `Google Gemini-2.5-Flash`. I chose this model for its speed and ability to follow complex instructions.

2.3 Evaluation Metrics

Following the standards for unsupervised style transfer [3], I used three metrics:

1. **Geometric Distance:** I calculated the distance between the text vectors and the center of each author's cluster to measure the shift in style.
2. **Style Probability (ACC):** I trained a Logistic Regression classifier to predict the author based on the embeddings.
3. **Fluency (PPL):** I calculated Perplexity using GPT-2 to ensure the generated text was still valid English.

2.4 Layer-Wise Analysis

Inspired by findings in image processing where style (texture) is found in lower layers and content is found in higher layers [2], I analyzed the internal states of the Transformer. I extracted embeddings from both **Layer 2** (early layer, focuses on syntax) and the **Final Layer** (late layer, focuses on meaning) to see where the style information is stored.

3 Experimental Results

3.1 Vector Space Topology

First, I projected the 600 text chunks into the embedding space. As shown in Fig. 1, the t-SNE visualization shows two completely separate islands for the two authors.

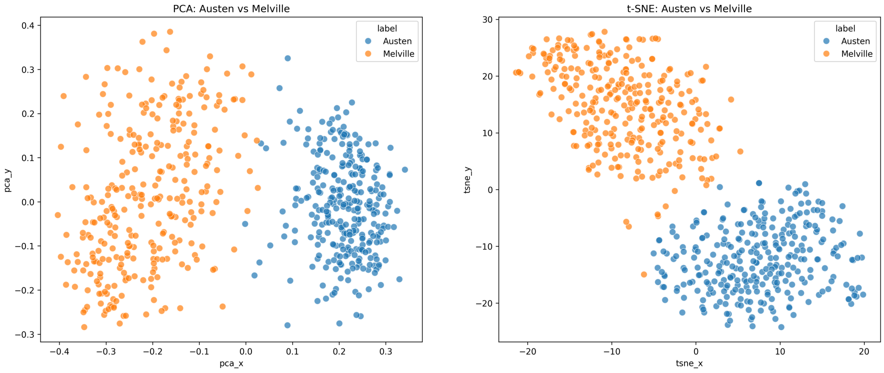

Fig. 1 t-SNE visualization showing clear separation between Austen (Cluster 0) and Melville (Cluster 1).

To measure this, I used K-Means clustering ($k = 2$). The algorithm achieved near-perfect separation, matching the real labels with **99.1% accuracy** (Table 1).

Table 1 Confusion Matrix of Unsupervised Clustering

Label	Cluster 0 (Austen)	Cluster 1 (Melville)
Austen	300	0
Melville	6	294

However, I found the **Silhouette Score** was low at **0.119**. This does not mean the clustering is bad. In this context, it means that while the authors are separate, their clusters are not tight points. They are broad clouds. This makes sense because the books cover many different topics; Austen talking about a wedding is mathematically far from Austen talking about a picnic, even if the style is the same.

3.2 Style Transfer Experiment

I selected a text from Austen describing a carriage ride with Mr. Elton. I then asked Gemini-2.5-Flash to rewrite it in the style of Melville, providing a real Melville paragraph as an example.

Qualitative Result: The generation was successful.

> *Original (Austen):* ”her heroism reached only to silence.”
> *Rewritten (Melville):* ”Her valor, a fragile, land-locked thing, sought its only refuge in the mute.”

The model successfully added archaic words (”valor,” ”mute”) and nautical metaphors (”land-locked”), changing the tone significantly.

Quantitative Analysis:

- **Fluency:** The Perplexity (PPL) increased from **68.24** (Original) to **101.67** (Rewritten). This increase does not mean the text is bad. Instead, it quantifies the higher complexity of Melville’s style compared to Austen’s simpler prose.
- **Vector Shift:** In the semantic embedding space (Final Layer), the vector moved from the Austen center (Distance: 0.099) to the Melville center (Distance: 0.271). However, it remained closer to Austen (0.259), meaning it failed to fully cross the boundary.
- **Classification:** The style classifier probability shifted from 91.4% Austen to 54.9% Melville. While this is technically a ”success” (crossing 50%), the margin is very slim.

3.3 Layer-Wise Disentanglement

To understand why the semantic vector struggled to classify the text as Melville, I analyzed the embeddings at **Layer 2**.

Table 2 Comparison of Distances to Centroids by Layer

Layer	Dist. to Austen	Dist. to Melville	Result
Semantic (Final)	**0.2596**	0.2716	Fail (Closer to Source)
Syntactic (Layer 2)	0.1325	**0.0934**	Success (Closer to Target)

As shown in Table 2, I found that the early layer successfully identified the text as Melville. Visualizing the attention weights (Fig. 2 and Fig. 3) explains why. Layer 2 shows a strong ”diagonal” pattern, meaning words pay attention to their immediate neighbors (local syntax). Layer 12, on the other hand, spreads attention globally to find named entities.

4 Concluding Remarks

This study confirms that literary style is a measurable phenomenon in vector space, but it is deeply mixed with content.

My results highlight a key limitation in current embeddings: the ”Content Anchor.” Because the rewritten text kept Austen’s characters (”Mr. Elton,” ”Mrs. Goddard”), the final layer of the model—which looks for entities—refused to classify the text as Melville, despite the obvious style change.

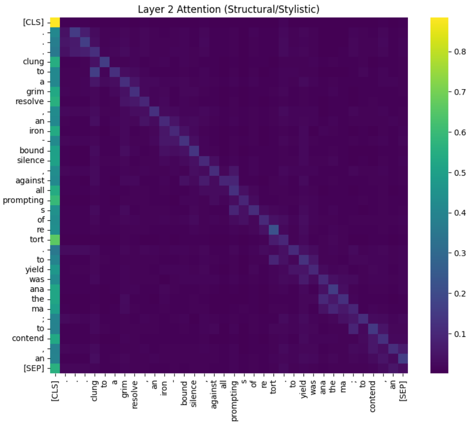

Fig. 2 Layer 2 Attention Map showing diagonal focus (local syntax).

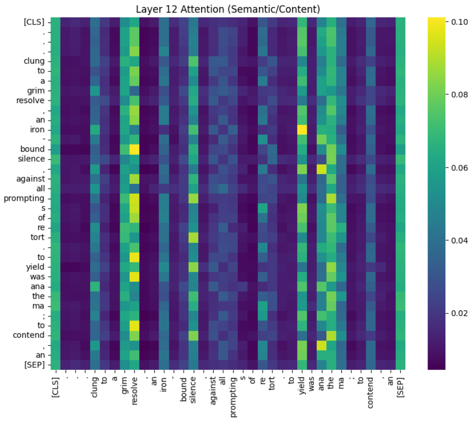

Fig. 3 Layer 12 Attention Map showing diffused focus (global semantics)

However, by looking at the lower layers of the model (Layer 2), I showed that the style shift *was* successfully encoded. The syntax-focused attention in the early layers correctly identified the Melvillean structure. This aligns with the findings of Cai et al. [2] regarding the hierarchical nature of feature extraction. Future work in text style transfer should use these lower-layer embeddings to better separate "how" something is written from "what" is written.

AI Usage Disclaimer

Parts of this project code and the conceptual outlining were developed with the assistance of **Google Gemini**. The AI was used to support the drafting of boilerplate code (matplotlib/sklearn), debugging dimension mismatch errors, and refining the choices in the project. All outputs have been modified, verified, and integrated into the final workflow by me.

References

[1] Jing, Y., Yang, Y., Feng, Z., Ye, J., Yu, Y., & Song, M. (2019). Neural style transfer: A review. *IEEE transactions on visualization and computer graphics*, 26(11), 3365-3385.

[2] Cai, Q., Ma, M., Wang, C., & Li, H. (2023). Image neural style transfer: A review. *Computers and Electrical Engineering*, 108, 108723.

[3] Pan, L., Lan, Y., Li, Y., & Qian, W. (2024). Unsupervised Text Style Transfer via LLMs and Attention Masking with Multi-way Interactions. *arXiv preprint arXiv:2402.13647*.