DTU Compute
Department of Applied Mathematics and Computer Science

Stylometric Analysis for Authorship Attribution

Evaluating Human and LLM-Generated Texts

Master Thesis

Stylometric Analysis for Authorship Attribution
Evaluating Human and LLM-Generated Texts

Master Thesis
February, 2025

By
Piotr Kalota

Cover photo: Vibeke Hempler, 2012
Published by: DTU, Department of Applied Mathematics and Computer Science,
Richard Petersens Plads, Build. 324, 2800 Kgs. Lyngby Denmark
www.compute.dtu.dk

Approval

This thesis has been prepared over six months at the Department of Applied Mathematics and Computer Science, at the Technical University of Denmark, DTU, in fulfilment for the degree Master of Science in Engineering, MSc Eng.

It is assumed that the reader has a basic knowledge in the areas of statistics and machine learning.

Piotr Kalota - s222914

....

Signature

....

Date

Abstract

The recent development of large language models (LLMs) has applications in various domains, enhancing daily human tasks in fields like software engineering and medical care. Nevertheless, similar models enable the creation of harmful solutions, like generating fake news. It revitalized the research on authorship attribution, a potential preventive mechanism against such misuse. While multiple methodologies, ranging from LLM output probabilities analysis to linguistic feature examination, have been developed to address the classification challenge, fewer efforts have concentrated on exploring writing style analysis. This thesis investigates the question: Do large language models exhibit novel writing styles? To achieve this, we employ stylometric analysis within the framework of the authorship attribution problem, leveraging our own dataset comprising texts authored by ten human novelists and texts generated by five LLMs prompted to mimic these authors.

Our findings show that human and LLM-generated texts exhibit distinct stylistic patterns, with human authors displaying higher variability and originality. LLM outputs rely on averaged writing styles and predictable linguistic patterns. While classification within the novel domain is effective, the approach struggles to generalize to other text domains, highlighting domain dependence. These results underscore the creative limitations of LLMs and the challenges of cross-domain stylometric attribution.

Acknowledgements

I would like to express my deepest gratitude to Prof. Sune Lehman and Assistant Prof. Jonas L. Juul for their invaluable guidance, support, and insights throughout the development of this thesis. Their expertise and encouragement were instrumental in shaping this work. I also extend my sincere thanks to DTU for funding the creation of the custom dataset used in this thesis, which was pivotal for the research and its outcomes.

I acknowledge the use of ChatGPT [1], according to DTU's AI referencing rules [2], to edit my master thesis in order to improve its readability. I affirm that all facts, concepts, and numbers have been delivered by me and have been verified for accuracy after the editing process. Each initial paragraph was provided to ChatGPT with the prompt, ”Edit it in the way that it fits the jargon of a master thesis, use active voice 'we'.” The resulting edits have been utilized either in full or in part.

[1] OpenAI, https://chat.openai.com

[2] https://www.bibliotek.dtu.dk/en/publishing/reference-management/kunstig-intelligens

Contents

1 Introduction

The advent of text generation capabilities through Large Language Models (LLMs) has revolutionized the way artificial intelligence interacts with natural language. These models, powered by advanced architectures such as transformers [1], can generate coherent, contextually relevant text across diverse applications. LLM-generated text finds utility in domains such as personalized education [2], legal documentation [3], and software development [4], showcasing significant potential to enhance human productivity and innovation. However, alongside these benefits, the proliferation of LLM-generated content raises concerns regarding its misuse [5][6] . From propagating misinformation and biases to obscuring the line between authentic and synthetic text, the societal and ethical implications of LLMs necessitate a nuanced understanding of their strengths and vulnerabilities. This duality underscores the need for rigorous study, particularly in areas like authorship attribution, to better manage the risks associated with LLM-generated text while leveraging its transformative capabilities.

The current state of research demonstrates significant progress in the domains of generated text detection and authorship attribution. These approaches aim to identify, with minimal or no prior knowledge about the text, the specific author or source. One prominent solution involves fine-tuned detectors [7][8][9], where large language models are further trained to classify text as either human-written or model-generated based on the source material. Alternative methodologies leverage statistical analysis of output probabilities generated by models [10][11] or use linguistic features of the text as input for various classification algorithms [12] [13][14]. While these methods achieve varying degrees of success, they often address authorship attribution within a constrained scope, either limited to a specific set of authors or lacking the adaptability required to keep pace with rapid advancements in artificial intelligence.

Despite their contributions, these solutions generally approach writing style analysis as a secondary concern rather than a primary focus. However, early works [15][16] analyzed the writing styles of different authors, showing the effectiveness of function words for authorship attribution tasks. These foundational insights can be further extended through the analysis of texts generated by large language models. An emerging, though relatively underexplored, area of research examines the ability of large language models to emulate requested writing styles. This line of inquiry is particularly relevant as it probes whether LLMs possess a human-like understanding of writing styles, enabling them to convincingly imitate human authors and potentially deceive human evaluators. Studies in this domain [17][18][19] applied different forms of prompting to generate the texts of very specific genres, like poems. The works indicate that large language models exhibit proficiency in replicating the structural and linguistic features of requested styles. However, they also emphasize that such models lack the originality inherent to human authors, highlighting an essential distinction between human and machine-generated texts.

Despite progress in generated text detection and stylistic analysis, current solutions often fall short in cross-domain generalizability and the nuanced understanding of writing styles [20]. This thesis addresses these gaps by investigating whether large language models exhibit unique writing styles. By exploring these distinctions, we aim to enhance authorship attribution methodologies and describe the stylistic tendencies of LLMs.

Through a combination of later discussed methods, this work has implications for improving detection algorithms and explainability of its decisions in sensitive applications, such as regulatory compliance in academic institutions, countering disinformation in media, and safeguarding intellectual property in publishing. Furthermore, the findings underscore the importance of interdisciplinary approaches that balance technological innovation with ethical considerations.

In this thesis, we aim to explore the novelty of the investigated texts, focusing on their uniqueness and unpredictability in writing style. In subsequent chapters we propose statistical definition of this term. Our primary goal is to answer the question: **Do large language models exhibit novelty in their writing styles?** Previous studies, although focusing on other objectives, also explore this question to some extent. These studies serve as a baseline and a source of inspiration for the experiments we conduct. To address this, we created a dataset called the Authors' Writing Style dataset, which includes texts from ten different novelists alongside texts generated by five large language models. We generated these texts by prompting the models to write in the style of one of the ten human authors. This dataset serves as the central resource for examining whether large language models are capable of comprehending and reproducing human writing styles, or whether their outputs are inherently characterized by distinct, model-specific stylistic tendencies.

To enrich our analysis and evaluate the generalizability of our findings, we incorporated additional datasets, including legal [21], social media [22] and news [23] domains. These supplementary datasets allowed us to validate our methods across different contexts and ensure robustness in our conclusions.

To address the research question, we analyzed linguistic features and visualized the results to find out how various texts cluster together. We also examined metrics such as the Information Content of linguistic attributes and other measures commonly used in natural language processing to uncover stylistic differences between human-authored and machine-generated texts.

Next, we used the most informative features from these analyses in classification experiments to test their ability to reliably distinguish between the various text sources. To evaluate the domain-independence of our approach, we applied these methods to an additional dataset, DAIGT-V4 [24], which was created specially for classification tasks. In the final chapter, we interpret our experiments' results, and present the conclusions, and key insights gained from this research.

2 Related Work

This chapter explores the foundational concepts of Large Language Models (LLMs), their societal and scientific impacts, and the specific areas of Authorship Attribution and Writing Style Analysis relevant to this thesis. The chapter concludes by highlighting research gaps that this work seeks to address.

2.1 Large Language Models

Large Language Models (LLMs), a subdomain of artificial intelligence within Natural Language Processing (NLP), generates natural language text. The advent of transformer-based architectures, as proposed by Vaswani et al. [1], marked a paradigm shift in NLP. Models such as GPT [25] and BERT [26] adopt this architecture at their core, achieving unprecedented performance on a wide range of NLP tasks.

The effectiveness of LLMs stems from their ability to process vast corpora of text, enabling unsupervised learning of linguistic features. Once trained, these models function as systems that accept user queries in natural language and provide coherent, contextually relevant responses. Their potential to revolutionize various domains underscores their significance [27].

2.2 Impacts of Large Language Models

2.2.1 Benefits of Employing Large Language Models

LLMs have catalyzed advancements across diverse fields by efficiently handling tasks otherwise infeasible for human labor. Singhal et al. [28] demonstrated this by fine-tuning the PaLM model to develop Med-PaLM, which addresses medical queries and aids in report generation and diagnosis. Similarly, Peng [4], highlighted the effectiveness of GitHub Copilot in enhancing the productivity of software engineers.

Other applications include legal assistance [3], personalized education systems [2], and beyond. Current research exposes multiple fields, which are or might be positively affected by introduction of LLMs. Collectively, these contributions underscore the transformative societal benefits of LLMs.

2.2.2 Risks Associated with Large Language Models

However, LLMs pose significant risks due to their accessibility and ability to generate content at scale. For instance, Sun et al. [5] revealed that LLMs could easily produce fake news indistinguishable from authentic content, creating a potential for societal harm. Similarly, Mosca et al. [6] showcased how LLMs could author scientific articles, which, while credible on in-domain tasks, risk spreading false information and undermining trust in scientific publications. On the other hand, Huang et al. [29] highlight that LLM hallucinations, can share knowledge that is not based on facts, reflecting an undesirable aspect of these models.

Efforts to mitigate these risks include developing robust LLM-generated text detection, advancing authorship attribution techniques, and improving writing style analysis, all of which are discussed in the subsequent sections.

2.3 Author Attribution and Writing Style analysis

Authorship Attribution and Writing Style Analysis are closely related fields that address key challenges associated with LLM-generated text. While text detection systems aim to distinguish human-written content from machine-generated outputs, authorship attribution identifies the specific author of a text. Writing Style Analysis delves deeper into linguistic patterns, providing insights into the stylistic features that characterize individual authors.

2.3.1 Historical Foundations

Authorship attribution has a long history predating the advent of LLMs. A seminal study by Mosteller and Wallace [15] employed Bayesian inference and function word analysis to attribute the authorship of 12 disputed Federalist Papers to Alexander Hamilton or James Madison. Their methodology was validated on undisputed texts, establishing a strong foundation for stylometric analysis.

Binongo [16] extended this approach to resolve authorship disputes surrounding books by Ruth Plumly Thompson and L. Frank Baum. Using Principal Component Analysis (PCA), Binongo visualized function word distributions, producing clear separations between authors in a two-dimensional plot. These foundational works laid the groundwork for modern stylometry, now revitalized by the emergence of LLMs.

2.3.2 Fine Tuned Detectors

The detection of LLM-generated texts has garnered significant attention in recent years, with numerous studies leveraging fine-tuning techniques on transformer-based models to address this challenge. Z. Lai et al. [7] developed detectors based on an ensemble of fine-tuned transformers. Their approach achieved high accuracy across both in-distribution and out-of-distribution test sets, showcasing the adaptability of transformers in this domain. The ensemble method combined predictions from multiple models, mitigating overfitting and enhancing generalization to unseen text distributions.

Similarly, S. F. Ebrahimi [8] and T. Marchitan et al. [9] adopted comparable approaches, utilizing state-of-the-art transformers fine-tuned for binary classification tasks to identify machine-generated content. Both studies reported impressive results, underscoring the efficacy of fine-tuning strategies for this purpose. Marchitan et al. also experimented with hybrid architectures, integrating features from traditional machine learning and deep learning methods, further refining the detection process.

While these methods offer robust tools for addressing the problem of LLM-generated text detection, they primarily focus on binary classification rather than delving into the stylistic characteristics that differentiate human and machine-generated text. The lack of an analysis explaining why and how these texts differ stylistically leaves a gap in understanding the broader implications of LLM outputs and their alignment with human-written norms.

2.3.3 LLM Output Probabilities Methods

An alternative approach to detecting machine-generated text involves analyzing the log probabilities assigned by language models. GLTR (Giant Language Model Test Room) [10] exemplifies this methodology, utilizing pretrained language models to identify statistical anomalies in token distributions. By highlighting improbable or unexpected tokens, GLTR can effectively flag text that is likely generated by a machine. This token-level analysis provides a detailed view of the distributional differences between human-written and LLM-generated text.

A more recent innovation, DetectGPT [11], introduces a zero-shot detection approach that does not require any prior fine-tuning or additional labeled examples. It focuses on the curvature of log probabilities rather than individual token probabilities. DetectGPT relies on the observation that small perturbations in machine-generated text typically reduce its overall log probability—a behavior not commonly observed in human-authored text. This method eliminates the need for additional fine-tuning, making it computationally efficient and versatile across different LLMs.

While both GLTR and DetectGPT are effective for identifying machine-generated text, they are limited in scope. These methods generally analyze individual LLMs and focus on statistical patterns, offering little insight into the stylistic nuances that distinguish texts across diverse authors or models. This limitation restricts their applicability in tasks that require a deeper understanding of stylistic variation.

2.3.4 Linguistic Features Classifiers

A complementary line of research investigates the linguistic differences between LLM-generated and human-authored texts, emphasizing lexical, syntactic, and structural features. T. Kumurage and H. Liu [30] explored such features, leveraging them as inputs for various classifiers. Their study compared open-source and proprietary LLMs, using SHapley Additive exPlanations (SHAP) [12] to identify the most impactful features. The results highlighted lexical diversity and structural features as key discriminators, not only between human and machine texts but also among different LLMs.

A. Shah et al. [13] further expanded this line of inquiry, incorporating Local Interpretable Model-Agnostic Explanations (LIME) [31] alongside SHAP to analyze feature importance. Their findings emphasized the role of readability scores and vocabulary richness in distinguishing human from machine texts. This study also underscored the potential for interpretability techniques like SHAP and LIME to provide insights into model behavior, making the classification process more transparent.

A more focused analysis by A. Muñoz-Ortiz et al. [14] examined the stylistic features of news articles, comparing original human-written content with LLM-generated outputs. Despite the linguistic fluency of machine-generated texts, their findings revealed distinct patterns, such as reduced vocabulary diversity and a lack of emotional realism. These characteristics not only distinguish LLM outputs from human writing but also expose stylistic differences across various LLMs, providing a more granular understanding of their limitations and capabilities.

2.3.5 Emulating Styles Investigation

An emerging research direction involves prompting LLMs to emulate specific writing styles, which offers valuable insights into their ability to replicate human creativity and stylistic nuance. Bhandarkar et al. [17] explored this topic by employing diverse prompting techniques, ranging from simple instructions with brief author excerpts to sophisticated prompts incorporating detailed linguistic attributes. Their findings highlighted significant challenges for LLMs in accurately replicating writing styles, particularly in maintaining consistent punctuation, lexical diversity, and coherence over longer texts.

Other studies have examined style emulation in poetic contexts, such as the work by Walsh [18] and the GPT Poetry project [19]. These studies employed structured prompts specifying both the topic and desired poetic style, with human evaluators assessing the quality of the outputs. While LLMs demonstrated an ability to mimic structural elements of poetry, they often fell short in producing the novelty and creativity characteristic of human poets. These findings suggest that LLMs, while adept at structural mimicry, continue to struggle with the subtleties of stylistic originality.

2.4 Summary

While current approaches enable the detect and analysis of LLM-generated texts, they still leave several gaps. Fine-tuned models excel in classification tasks but provide limited insights into stylistic variations. Log-probability approaches are computationally efficient yet lack nuance, while linguistic feature-based methods focus predominantly on binary distinctions rather than cross-author or cross-model comparisons. Finally, studies on style emulation highlight LLMs' challenges in replicating human creativity and originality. Addressing these gaps is crucial to advancing the field and ensuring responsible deployment of LLM technologies.

3 Datasets

In our research, we use multiple datasets, with our own Authors' Writing Style dataset serving as the cornerstone of our analysis. To enhance and broaden our investigation, we supplement this primary dataset with four additional datasets: the Twitter and Reddit dataset [22], the Global News dataset [23], the Legal Contracts dataset [21], and the DAIGT-V4 dataset [24]. These supplementary datasets provide diverse perspectives and information beyond our core dataset. In particular, we leverage the DAIGT-V4 dataset [24], specifically designed for human versus model text classification, to incorporate an additional domain dimension into our classification tasks.

3.1 Authors' Writing Style Dataset

We created the Authors' Writing Style dataset using two corpora: a collection of books written by human authors and corresponding texts generated by large language models, instructed to mimic the writing style of different authors. For the human-authored texts, we used books from the open-source Gutenberg Project, a well-known repository of classic literary works. Given the expansive nature of this collection, we selected a subset of authors to ensure manageability and relevance to our research goals. To generate the machine-authored texts, we employed five large language models: *gpt-3.5-turbo-0125 (OpenAI)*, *gpt-4o (OpenAI)*, *gemini-1.5-flash (Google)*, *open-mixtral-8x7b (MistralAI)*, and *claude-3-haiku-20240307 (Anthropic)*.

3.1.1 Project Guthenberg Books

The Gutenberg Project is a digital library offering free access to thousands of eBooks, primarily classic literary works whose copyrights have expired. It provides an extensive source of human-written text, making it highly suitable for our study. For this research, we used the curated version provided by L. Shibamouli [32], which includes 3,036 English books written by 142 authors. The text volumes for individual authors range from approximately 6,000 words to 9.5 million words, offering a diverse and substantial corpus.

To extract text, we employed a method of splitting the content by whitespaces, under the assumption that the minor inaccuracies of this approach would not impact the selection process. We carefully cleaned the downloaded texts by removing metadata, licensing information, and transcribers' notes wherever possible, ensuring that the resulting corpus is well-suited for stylistic analysis.

The "Inference in an Authorship Problem" paper [15] employs around 200,000 words for its stylometric analysis, which serves as the minimum threshold for author selection, narrowing the pool to 114 candidates. The analysis was planned to be conducted on a single domain, which led as to choose only the novels books, which further limited the number of considered authors. Generation of such corpora comes with a cost of requests to the LLM provider's API and the processing time of our analysis. Due to our resources availability we limited the analysis to 10 authors. Since explained criteria yielded more than ten authors, the rest of authors where excluded arbitrarily. Table 3.1 presents selected authors and corresponding number of words from all the available books. All the writings has been saved in the the form of files named `[author]___[title].txt`.

3.1.2 LLM-generated Corpora

For the purpose of this analysis we selected five models - *gpt-3.5-turbo-0125 (OpenAI)*, *gpt-4o (OpenAI)*, *gemini-1.5-flash (Google)*, *open-mixtral-8x7b (MistralAI)*, and *claude-3-*

Author	Words Count
Mark Twain	3M
Zane Grey	2M
Joseph Conrad	2M
George Eliot	2M
Benjamin Disraeli	2M
Lucy Maud Montgomery	1M
William Henry Hudson	1M
Howard Pyle	755k
Virginia Woolf	375k
Lewis Carroll	299k

Table 3.1: Word counts for selected authors from Authors' Writing Style dataset. Numbers are rounded to either the nearest millions (M) or thousands (k).

haiku-20240307 (Anthropic). These models were among the most popular, at the moment of data collection process, LLMs developed by various providers, ensuring a comprehensive analysis. To generate the data, which includes diverse sequences of text, 400 one-sentence queries were prepared. These queries are abstract and universal to avoid the use of references from other sources. The first five queries are listed below:

```
Describe the most peculiar meal you've ever encountered.
Explain the finer points of etiquette for a barnyard gathering.
Convince a grumpy mule to pull a rickety carriage.
Debate the merits of a mustache versus a beard.
Write a love letter from a lovesick cactus to a blooming rose.
```

Each query is accompanied by additional instructions on how the final answer should be formatted. The following prompt has been used to generate text for the authors:

```
Come up with the answer in [author]'s writing style.
Don't use direct references and citations of [author].
Answer in plain text format. Use 3000 words.
[query]
```

In this prompt, the placeholder *[author]* is replaced by the name of the specific author, and *[query]* is replaced by one of the queries mentioned above. The prompt aims to achieve several key objectives:

1. Ensure the text follows the writing style of the author, making the analysis more challenging. It is assumed that language models are familiar with the books of requested authors, due to the training processes and open-sourced availability of these texts.

2. Prevent direct citations from the author's writings, as these LLMs are likely trained on these books among other sources.

3. Require the LLM to return the text in a simple, plain format for technical reasons.

4. Instruct the LLM to use only 3000 words to maintain context, acknowledging that LLMs often do not strictly adhere to this constraint but generally aim to stay within it.

Despite of single deviations explained at the end of this sub-section such a prompting strategy resulted in comprehensive stories generated by large language models. Answers have followed the requested topics and the 400 queries were sufficient to reach the desired threshold of 200k words for each of the collections.

For each combination of model, author, and query (5 models * 10 authors * 400 queries), requests were made, and the responses were saved in a defined JSON format, as displayed below:

```
{
    "id": ...,
    "requested_number_of_words": ...,
    "response_length": ...,
    "model": ...,
    "created_at": ...,
    "author": ...,
    "prompt_template": ...,
    "query": ...,
    "response": ...
}
```

During generation the default settings of providers' API [33][34][35][36] were used. The *temperature* parameter, which controls the randomness of the output, was set to its default value (ranging from 0.7 to 1 depending on the provider). Lower temperature values make the model's responses more deterministic and focused, while higher values increase the diversity and creativity of the responses. To allow the LLMs maximum flexibility, the *top p* parameter was set to 1, meaning the model considers all possible tokens for generation, thus not restricting token selection based on cumulative probability. The *max tokens* parameter, which controls the maximum number of tokens generated, was left unset, as there were no strict constraints on the length of the answers. Other parameters, deemed less relevant to the project's objectives, were also set to their default values.

Tuning these parameters could affect the response length or the choice of tokens, which are integral to the writing style of LLMs. Additionally, it could influence the content of the generated answers. To remain objective and account for uncertainty about the optimal configuration, we used the default parameter values, as this is likely the most common setup chosen by researchers.

For each model-author pair, approximately 280,000 words were generated, surpassing the minimum threshold of 200,000 words to accommodate potential word losses during future cleaning processes. The exception is the *claude-3-haiku-20240307* model, which generated around 600,000 words for each author across 400 queries, showing the biggest lack of discipline in the number of generated words. As depicted in table 3.2, the final word counts vary due to the irregular lengths of responses from the Language Model (LLM). Similar to authors selection, text was split into words using whitespace, as we consider this simple method sufficient to present the overview of the text generation.

All generated responses are stored within directories labeled by `{model_name}/{author_name}`. To share an understanding of the structure of responses, presented below are two responses [1] generated by gpt-4o for the query:

```
    Describe the most peculiar meal you've ever encountered.
```

[1] Due to visual reasons responses are truncated.

Author	Model 1	Model 2	Model 3	Model 4	Model 5
Mark Twain	267k	322k	296k	251k	689k
Zane Grey	291k	340k	311k	251k	630k
Joseph Conrad	310k	342k	318k	289k	608k
George Eliot	303k	302k	280k	260k	559k
Benjamin Disraeli	273k	253k	252k	252k	558k
Lucy Maud Montgomery	290k	276k	278k	266k	547k
William Henry Hudson	299k	269k	286k	252k	589k
Howard Pyle	294k	260k	279k	251k	481k
Virginia Woolf	290k	251k	278k	251k	470k
Lewis Carroll	300k	262k	263k	251k	608k

Table 3.2: Number of words generated by various models for each author, rounded to the nearest thousand and expressed in thousands (k). Models are enumerated in the following order: *gpt-3.5-turbo-0125*, *gpt-4o*, *gemini-1.5-flash*, *open-mixtral-8x7b*, and *claude-3-haiku-20240307*

Response [2] in Mark Twain's style:

```
Well now, let me tell you about the most peculiar meal
I ever did encounter. It happened on a hot summer's day,
the kind where the sun seems to hang lazily...
```

Response [3] in George Eliot's style:

```
It was in a modest village, tucked away amidst rolling hills
and meandering streams, that I stumbled upon the most
peculiar meal I have ever encountered...
```

Even though it was not a big obstacle, it is worth highlighting that LLMs do not adhere to the request regarding the output length. For instance, around 200 requests out of 3660 *gemini-1.5-flash* resulted in an empty string, with no apparent cause discerned from the investigation, whereas other LLMs did not exhibit this issue. A particular query `Write a newspaper article announcing the invention of the internet.` appeared to be confusing for many of the LLMs. Instead of writing the article models were returning only the headline of it. Despite these edge cases, the models generally performed as expected, providing comprehensive responses. Although prompt engineering suggests tailoring prompts for each model, the quality of generated stories demonstrated that a single, universal prompt suffices for such a task. Notably, no instances of response refusal due to perceived harmful content were encountered during the process.

3.2 DAIGT-V4 Dataset

The DAIGT-V4 dataset [24] is a carefully constructed and comprehensive dataset designed for the ”Detect AI-Generated Text” competition hosted on Kaggle [37]. Created in January 2024, this dataset serves as a benchmark for developing classifiers capable of distinguishing between human-written and LLM-generated text. The dataset comprises 73,573 essays, generated either by humans or by models from the following families: *mistral*, *llama*, *gpt*, *claude*, *falcon*, *palm*, *cohere*, *ada*, *babbage*, *curie*, and *davinci*.

[2] File: `res/datasets/writing-style/raw/models/gpt-4o/Mark Twain/1717948495.json`

[3] File: `res/datasets/writing-style/raw/models/gpt-4o/George Eliot/1717948549.json`

Collection	Words count
human	10M
mistral	7M
llama	6M
gpt	1M
claude	690k
falcon	2M
palm	770k
cohere	130k
ada	150k
babbage	150k
curie	170k
davinci	620k

Table 3.3: Number of words in various DAIGT-V4 dataset collections, rounded to the nearest thousand and expressed in millions (M) or thousands (k).

As shown in Table 3.3, the word count for each collection ranges from approximately 120,000 to 10 million. Each row in the dataset includes the following attributes:

1. *text*: The content of the essay.
2. *label*: A binary indicator where 0 denotes human-generated text and 1 indicates LLM-generated text.
3. *prompt name*: The topic of the essay, injected into the prompt template.
4. *RDizzl3 seven*: A boolean flag indicating whether the essay is based on prompts included in the hidden test set of the competition.
5. *source*: Specifies the exact name of the model used to generate the text or stores *persuade corpus* for human-written essays.
6. *model*: A string representing the family of the model identified in the *source* column.

The DAIGT-V4 dataset uniquely combines a diverse set of texts sourced from multiple open-source and proprietary large language models (LLMs) alongside a substantial collection of labeled human-written and AI-generated content. This rich diversity makes it an exceptional resource for evaluating domain-independence in classification tasks. Beyond its primary role in investigating classification performance, as described in later chapters, the dataset also contributes significantly to the analysis of global word distribution, a concept that will be thoroughly explained in subsequent sections.

3.3 Global News Dataset

The Global News dataset [23] comprises news articles collected over several months and shared on Kaggle platform. The primary motivation behind curating this dataset was to develop and experiment with various natural language processing (NLP) models. The data is sourced from the NewsAPI [4], a comprehensive and up-to-date news aggregation service. The API provides access to a wide range of news articles from various reputable sources, making it a valuable resource for constructing a diverse and informative dataset.

The dataset consists of 100k articles, each containing the following fields:

[4] https://newsapi.org/

1. *article id*: Unique article id
2. *source id*: Source title
3. *source name*: Source name
4. *author*: The author of the article
5. *title*: The headline or title of the article.
6. *description*: A description or snippet from the article.
7. *url*: The direct URL to the article.
8. *url to image*: The URL to a relevant image for the article.
9. *published at*: The date and time that the article was published, in UTC.
10. *content*: The unformatted content of the article, where available. Truncated to 200 characters.
11. *category*: The category of the article
12. *full content*: The unformatted full content of the article, where available

The dataset contains over **53 million words** in the *full content* field. It serves as a rich resource for capturing domain-specific vocabulary and linguistic patterns pertinent to the news domain. These characteristics play a critical role in modeling and defining global word distribution.

3.4 Legal Contracts Dataset

The Legal Contracts dataset [21], also hosted on Kaggle, is a specialized corpus consisting of over 13,000 labeled spans across 510 commercial legal contracts. These annotations were curated manually by The Atticus Project, identifying 41 categories of key legal clauses commonly reviewed by professionals in contract analysis. The dataset is provided in various formats to accommodate diverse research applications:

1. *full contract txt*: Plain-text files containing the raw content of all 510 contracts, useful for textual reference and analysis.
2. *full contract pdf*: PDF versions of the contracts, mirroring the content in *full contract txt*.
3. *master clauses.csv*: A CSV file with 83 columns and 510 rows, where each row represents a contract. This file provides detailed metadata and manually labeled text spans corresponding to 41 predefined clause categories, including contract name, annotated clause text, and human-labeled data for clause categorization.
4. *SQuAD style JSON*: A JSON file formatted akin to the SQuAD 2.0 dataset, designed for training and evaluating question-answering models. This file maps contract paragraphs to clause categories, enabling clause prediction tasks.

This dataset encompasses **3 million words** across its plain-text contract files and is a significant resource for defining global word distribution. Its detailed annotations provide valuable insights into legal language usage, supporting both domain-specific and cross-domain analyses.

3.5 Twitter and Reddit Dataset

The Twitter and Reddit dataset was developed as part of a university project on *Sentiment Analysis Across Multi-Source Social Media Platforms*. It consists of two subsets: tweets from Twitter and comments from Reddit, both annotated with sentiment labels. These datasets focus on public discourse around Indian political leaders and general public opinions about the 2019 Indian General Elections. The data was sourced using the Tweepy [5] and PRAW [6] APIs, which enable the extraction of tweets and comments, respectively.

Each tweet and comment is labeled with a sentiment score based on the following schema:

1. 1: Positive sentiment.
2. 0: Neutral sentiment.
3. -1: Negative sentiment.

The dataset is composed of two sub-datasets:

1. Twitter dataset: Contains 163,000 tweets, comprising approximately 3 million words.
2. Reddit dataset: Includes 37,000 comments, with a total of 1 million words.

Before analysis, the textual data was preprocessed using Python's `re` library and NLP techniques to remove noise. These datasets contribute an additional **4 million words** to the global word distribution definition. By encapsulating sentiment-driven language from diverse social media platforms, this dataset enriches the linguistic variety necessary for our analysis.

[5] https://www.tweepy.org/

[6] https://praw.readthedocs.io/en/stable/

4 Methods and Results

In this chapter, we present the experiments conducted and the results obtained. Our analysis is divided into three key sections: Principal Components Analysis, Classification and, Information Content analysis. Each section provides detailed insights into the methodologies and outcomes specific to its focus. Before delving into the analysis, it is essential to define certain terms that may be interpreted differently across various academic works:

1. *Word:* A sequence of characters containing Latin letters, such as *father* or *headdress*. This definition aligns with the conventional concept of a word in English.
2. *Token:* Any segment resulting from the tokenization process. Tokens may include words, punctuation marks, numbers, or other meaningful units, such as *father*, *!*, or *23*.
3. *Complex Word:* A word consisting of at least three syllables, indicating greater linguistic or phonetic complexity.

As outlined in the introduction, our primary objective is to address the overarching question: **Do large language models exhibit novelty in their writing styles?**. We break this question down into smaller, more manageable sub-questions, systematically answering each to build toward a comprehensive conclusion. In the following sections, we introduce these sub-problems and describe our approach to tackling them.

4.1 Principal Components Analysis

We aimed to determine whether it is possible to distinguish human-written texts from model-generated ones. This fundamental hypothesis can be investigated through various methods. Drawing inspiration from previous studies, like the one of Binongo [16], we explored the potential of leveraging linguistic features combined with Principal Components Analysis (PCA) to uncover patterns through dimensionality-reduced plots.

We adopted this approach as the initial step in our investigation. It provided us with first insights and allowed us to validate our assumptions, which we further explored through subsequent methods. The process, which uses the Authors' Writing Style dataset, began with rigorous data cleaning, preprocessing and metrics extraction, detailed in the following subsections. These preparatory steps ensured that our analysis was robust and yielded meaningful insights.

4.1.1 Data Cleaning

First, we loaded the Authors' Writing Style dataset, as described in the previous chapter. For each of the ten selected authors and six collections (human-written and five model-generated corpora), we loaded the corresponding texts. To ensure the reliability of our analysis, we performed a cleaning process on the dataset. Recognizing the unique sources of noise present in books and LLM-generated content, we applied distinct cleaning procedures tailored to each type of text.

For the books, we identified and removed the following sources of noise:

1. *Italic formatting:* Words surrounded by $*$ characters, which indicate italic formatting, were stripped of these characters.
2. *Dividers:* Section or chapter separators, often represented as sequences of $*$ characters, were removed to maintain textual continuity.

3. *Illustration annotations:* Images within the books, replaced by annotations such as `[Illustration: ...]` in the text format, were removed as they do not contribute to the narrative content.

4. *Note annotations:* Footnotes, represented by superscript numbers in the original text and rendered as `{}`-enclosed annotations in the digital version, were removed. Since these notes are typically added by publishers rather than the authors, we excluded them to preserve the authenticity of the text.

For LLM-generated texts, we identified several forms of noise and applied the following cleaning steps:

1. *Small chunks* Texts shorter than 100 words were excluded, as these were likely generated due to the language model failing to comprehend the task, resulting in irrelevant responses.

2. *Repeated substrings* We removed texts that ended with repeated substrings, such as:

   ```
   This is my story about the apple the apple the apple...
   ```

   Specifically, texts with at least three repeated substrings, each consisting of a minimum of two words, were excluded to avoid introducing biases.

3. *Emojis:* All emoji encodings were removed from the texts to maintain consistency in formatting and focus on semantic content.

4. *@ Signs:* Texts containing the `@` symbol were cleaned to remove these symbols

5. *HTML Tags:* In some cases, generated texts contained HTML tags used for formatting. We removed all such tags to standardize the text structure.

6. *Bold formatting:* Words or sequences formatted using bold syntax (`**...**`) were stripped of the formatting characters.

The cleaning process removed approximately 1% of the raw dataset. These procedures were essential for eliminating irrelevant or redundant content, which removed outliers encountered during the development.

4.1.2 Data Preprocessing

We transformed the texts as a preparatory step for subsequent analysis. For the books, we defined a ”text” as an entire book, whereas for the models, we defined a ”text” as a single response generated for a prompt. To segment these texts into sentences, we used the `sent_tokenize` function from the `nltk.tokenize` module, specifically employing the `PunktSentenceTokenizer`. To prepare the data for analysis, we first divided the texts into raw segments, which served as intermediate units for further processing:

1. *Model-generated texts:* Each response, being a distinct and self-contained logical unit, was treated as a single raw segment.

2. *Books:* To ensure contextual integrity, we divided books into raw segments of approximately 25,000 characters (equivalent to roughly 5,000 words), carefully avoiding arbitrary sentence truncation. By adopting a character-based segmentation approach, we minimized the computational burden associated with tokenizing entire books at this stage. While this method introduces slight imprecision, it remains acceptable for subsequent tasks that accept unequal segmentation of the texts.

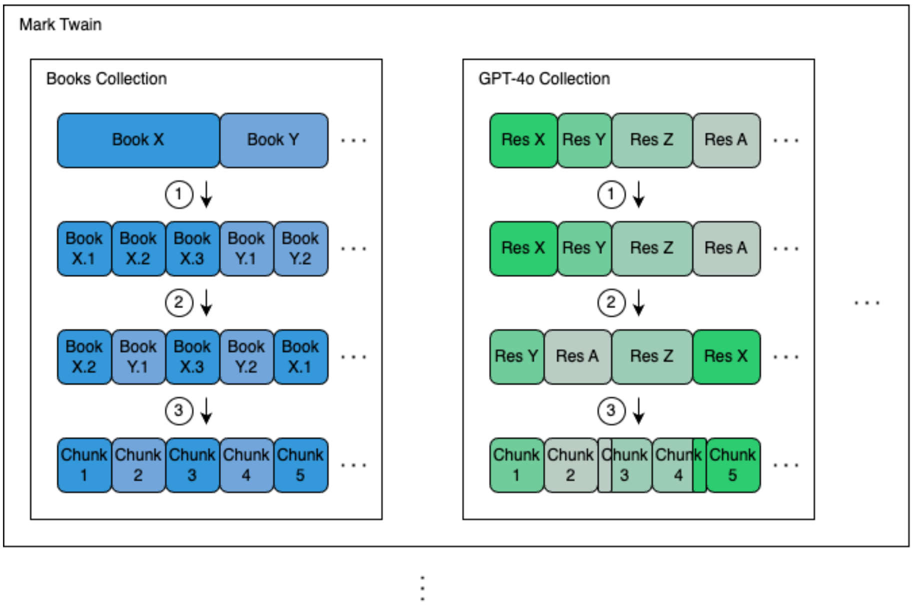


Figure 4.1: The figure illustrates the chunking pipeline for author-collection pairs. For books, the process involves: (1) segmentation into raw segments, (2) shuffling the raw segments, and (3) chunking into uniform sizes of approximately 5,000 tokens. For model collections, step (1) is skipped, proceeding directly to steps (2) and (3).

Next, we shuffled these raw segments within each author-collection pair to ensure balanced representation. For model-generated texts, shuffling had minimal impact because the raw segments were already independent. However, for books, shuffling was critical to avoid sampling only from a single book when selecting raw segments for an author. This approach ensured that token blocks were distributed proportionally across all books in the author's collection, reducing potential biases introduced by focusing on a single storyline.

After shuffling, we processed the raw segments to create the final token blocks required for our analysis:

1. Each sentence within a raw segment was tokenized into words using the `nltk.word_tokenize` function and the `PunktWordTokenizer`.
2. The tokenized sentences were subsequently merged into sequential blocks of approximately 5,000 tokens each. Minor deviations from the target size were allowed to avoid splitting sentences and preserve their integrity.
3. We repeated this process until each collection-author pair reached exactly 40 blocks, totaling around 200,000 tokens.

This pipeline produced a standardized text segmentation, with each collection-author pair containing approximately **200,000 tokens**, divided into uniform **5,000-token blocks**, hereafter referred to as **chunks**. It leaves as with 2400 chunks totaling around 480M tokens for our investigation. Analyzing these chunks, rather than entire texts, enabled us to explore within-text variability more effectively.

In the next step each chunk was transformed to the following structure:

1. *Text:* Raw chunk's text
2. *Sentences:* List of all chunk's sentences

3. *Tokens:* List of all chunk's tokens
4. *Words:* List of all chunk's words
5. *Syllables number:* Number of syllables in the chunk
6. *Complex words number:* Number of complex words in the chunk

The first three objects—text, sentences, and tokens—were preprocessed in the earlier steps. Subsequently, we filtered words from the tokens using Python's regex function with the pattern `[a-zA-Z]`. On average, this filtering retained 87.5% of the token set. To extract syllables for each word, we used the `nltk.corpus.CMUDictCorpusReader`. For words without syllable matches in this resource, we employed `pyphen.Pyphen` configured for the English language as a backup solution. The extracted syllables were counted and stored in the respective variables of the data structure. Simultaneously, we counted the number of complex words—defined as words containing at least three syllables.

Author	Collection	Sentences	Tokens	Words	Syllables	Complex Words
Mark Twain	books	8342	200708	173078	237421	14378
Mark Twain	gemini-1.5-flash	9861	200452	169394	231999	13291
Joseph Conrad	books	11007	200555	171924	240832	15488
George Eliot	gpt-4o	8306	200529	174380	271276	24182
George Eliot	open-mixtral-8x7b	6044	200623	177382	273435	24562
Zane Grey	gpt-3.5-turbo-0125	7865	200521	178652	264370	19744

Table 4.1: Preprocessing statistics for six randomly selected chunks from a total of 2400.

Table 4.1 presents the preprocessing statistics for six randomly selected chunks, illustrating the general distribution of specific language measures. As visible on the table, our pipeline results in the chunks of roughly 200k tokens, without the lost of any information. Instead, the texts are sliced into smaller, independent analysis pieces that help as check the variability within a single text. This concludes the preprocessing stage and provides an overview of the data structure that will be used in the subsequent analysis.

4.1.3 Metrics and Features Extraction

To perform PCA, we first needed to determine which linguistic metrics to use as input features. To make this decision, we reviewed relevant literature addressing similar challenges. Many studies have used metrics related to word, sentence, or paragraph lengths [14][13][30]. Other approaches have analyzed distributions of parts of speech or employed various text scores that reflect either vocabulary richness or text readability. Additionally, researchers have examined structural aspects of texts, such as stopword usage or punctuation patterns.

In selecting the metrics for our study, we aimed to balance their utility and the computational resources required. We chose to focus on fundamental metrics, such as average word length and syllables per word, which provide a numerical overview of a text chunk's structure. Furthermore, we incorporated established readability and richness scores to gain deeper insights into text characteristics. Given the potential influence of punctuation on our task, we also included punctuation counts as a feature. Lastly, we considered the frequency of function words, as prior studies indicate that the usage patterns of these context-independent words (e.g., ”of,” ”a,” ”the”) can reveal stylistic tendencies. Below, we provide a detailed list of all metrics extracted for each text chunk:

1. *Unique Word Number:* The number of unique words within the text chunk.

2. *Average Word Length:* The mean length of words in the text chunk, measured in characters.

3. *Average Sentence Length:* The mean length of sentences in the text chunk, measured in words.

4. *Average Syllables Per Word:* The mean number of syllables per word in the text chunk.

5. *Function Words Counts:* A dictionary where keys represent specific function words (e.g., ”of,” ”the”) and values indicate their frequency within the text chunk.

6. *Puncutations Counts:* A dictionary where keys represent specific punctuation marks (e.g., ”.”, ”,”) and values indicate their frequency within the text chunk.

7. *Flesch Reading Ease:* A readability metric that evaluates the ease of comprehension of the text. Scores range from 0 to 100, with higher scores indicating easier readability.

   $$206.835-1.015(\frac{N}{S})-84.6(\frac{SL}{N})$$

   Where N is the total number of words, S is the number of sentences and SL is the total number of syllables.

8. *Flesch Kincaid Grade Level:* A readability metric estimating the U.S. school grade level required to understand the text. Lower scores indicate simpler text.

   $$0.39(\frac{N}{S})-11.8(\frac{SL}{N})-15.59$$

   Where N is the total number of words, S is the number of sentences and SL is the total number of syllables.

9. *Gunning Fog Index:* A readability metric assessing the number of years of formal education needed to comprehend the text. Scores range from 0 to 20, with lower scores indicating simpler text.

   $$0.4[(\frac{N}{S})+100(\frac{C}{N})]$$

   Where N is the total number of words, S is the number of sentences and C is the total number of complex words.

10. *Yules Characteristic K:* A measure of text ”disorderliness” based on word frequency distribution. Lower values indicate greater vocabulary diversity, while higher values suggest more repetition.

    $$\frac{10^4}{N^2}\sum_{i=1}^{V}(n_i-c)^2$$

    Where N is the total number of words, V is the vocabulary size, n_i is the frequency of the $i-th$ word, and c is the mean frequency of all words.

11. *Herdan's C:* A measure quantifying word frequency distribution and vocabulary richness.

$$\frac{logV}{logN}$$

Where N is the total number of words, V is the vocabulary size.

12. *Maas:* A measure of vocabulary richness, assessing the proportional expansion of unique words relative to the total number of words. Lower scores indicate greater richness.

$$\frac{logN - logV}{log(N^2)}$$

Where N is the total number of words, V is the vocabulary size.

13. *Simpsons index:* A measure quantifying the likelihood of two randomly selected words being identical. Values range from 0 (high diversity) to 1 (low diversity).

$$1 - \sum_{i=1}^{V}(\frac{n_i}{N})^2$$

Where, V, n_i and N are as defined above.

To define the set of function words, we used the Python `functionwords.FunctionWords` instance for the English language, which provides a total of 512 function words. Similarly, we extracted punctuation characters using the `punctuation` from Python's `string` , yielding 32 unique punctuation marks. By considering the counts of each function word and punctuation mark as individual metrics, along with the other metrics outlined in the previous section, we initially defined a model with 555 metrics.

To address the potential noise introduced by rarely used punctuation marks, we implemented a filtering strategy. For each text chunk, we identified the five most frequently occurring punctuation marks and merged these sets across all chunks. This resulted in a final selection of seven punctuation marks: *.* (period) *,* (comma) *?* (question mark) *;* (semicolon) - (hyphen) *"* (double apostrophe) and *'* (apostrophe).

Function words have gone under similar strategy, but this time we chose ten function words with the highest count number in each chunk. It resulted in the following 23 function words after the merge: "you", "that", "it", "as", "a", "had", "she", "said", "and", "of", "her", "their", "its", "our", "was", "he", "my", "in", "his", "the", "to", "is", "with".

Through this process, we reduced the dimensionality of the dataset from 555 to 45 metrics, which we now refer to as **features**. These features form the basis for the Principal Component Analysis discussed in the following section. They are further used in other, later presented methods, which results show that this set does not translates well to other text domains. The reasons and potential solutions for that are presented in *Discussion* and *Future Work* chapters.

4.1.4 Results

We employed Principal Component Analysis (PCA) to gain initial insights into the distinctions between human-written and model-generated texts, utilizing the features extracted in the preceding steps. We used an implementation of PCA from the `sklearn.decomposition` package, coupled with `StandardScaler` from `sklearn.discriminant_analysis` for feature scaling. In each experiment, we reduced the 45 features to two principal components, which we subsequently visualized on a scatter plot.

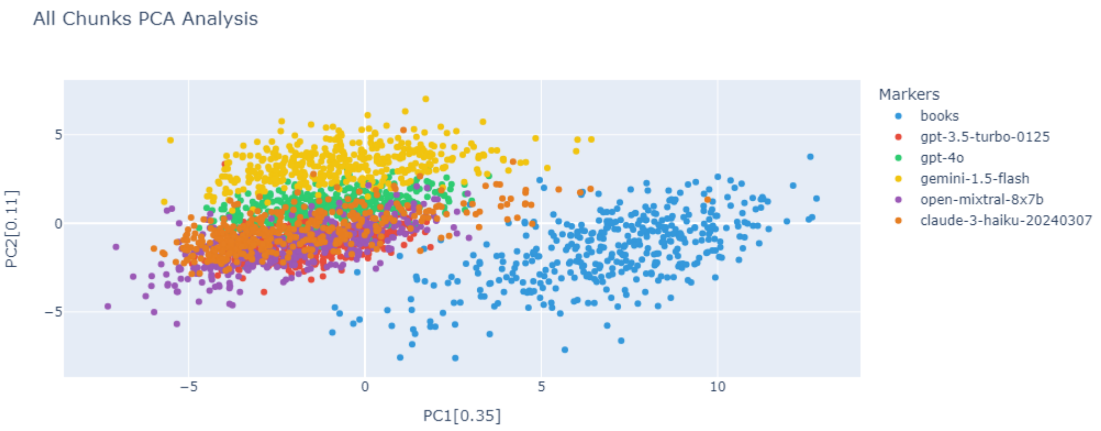


Figure 4.2: Scatter plot visualizes each text chunk as a single point, derived from a Principal Component Analysis (PCA) performed on all available chunks in the Authors' Writing Style dataset. Points are colored according to their collection of origin.

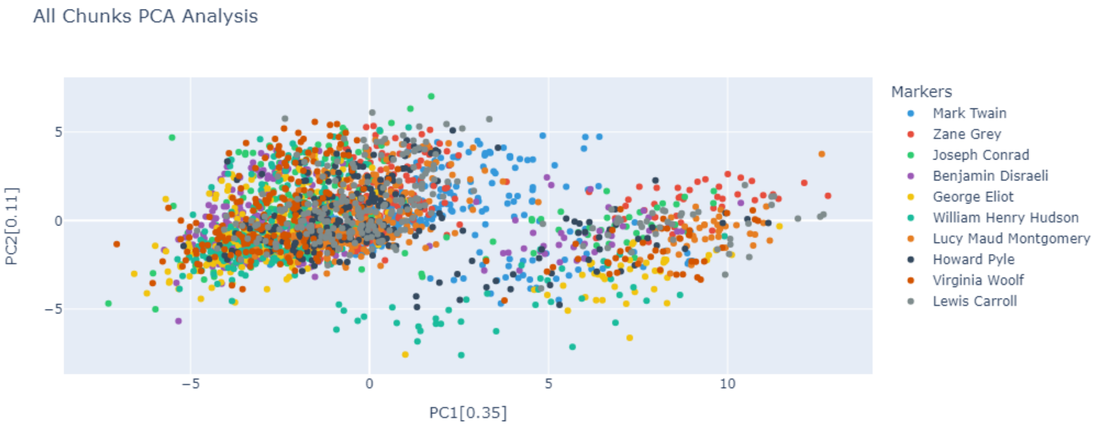


Figure 4.3: Scatter plot visualizes each text chunk as a single point, derived from a Principal Component Analysis (PCA) performed on all available chunks in the Authors' Writing Style dataset. Points are colored according to their author of origin.

Our initial PCA experiment encompassed all 2400 available text chunks from all authors and collections. Figures 4.2 and 4.3 depict each chunk as a single point on a two-dimensional plot. While the underlying data remains consistent, we marked the points by their corresponding collection in Figure 4.2 and by their author in Figure 4.3. The first two principal components explained 46% of the variance in the data—35% by PC1 and 11% by PC2—suggesting their potential efficacy in representing the full feature set.

The plot readily distinguishes a cluster of human-written texts (books) from the model-generated texts. The models clustered together, forming a larger group within which further sub-clustering by individual model could be attempted, however, presenting a more challenging task. Conversely, the plot colored by author revealed no discernible patterns of points clustering around specific authors. We observed that Mark Twain's writing style, and the style mimicking him, exhibited greater similarity to other authors in the dataset compared to the remaining authors within the dataset.

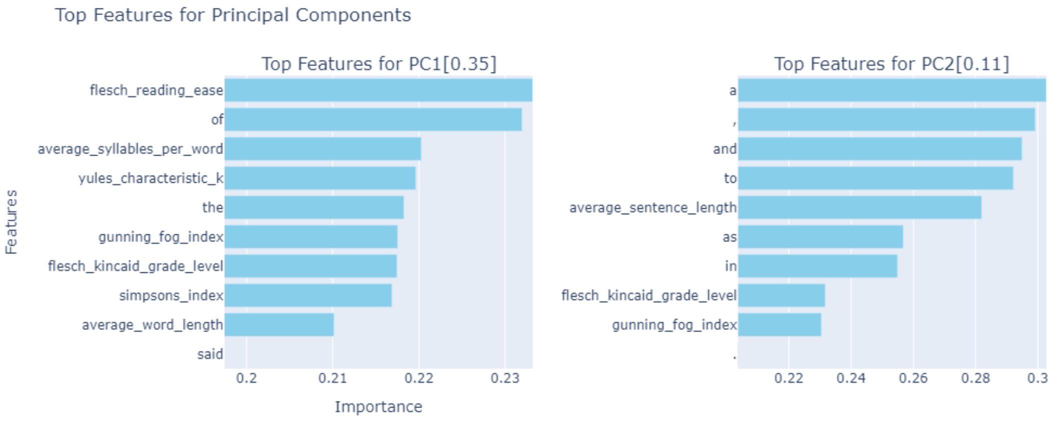


Figure 4.4: Bar plots display top ten feature importances of each principal component from the PCA performed on all text chunks from Authors' Writing Style dataset.

Furthermore, we investigated the contribution of each feature to the PCA results. Figure 4.4 presents the top ten features for both principal components. We observed that various scores applied to the text chunks and the frequency of function words played a significant role in determining the position of each point.

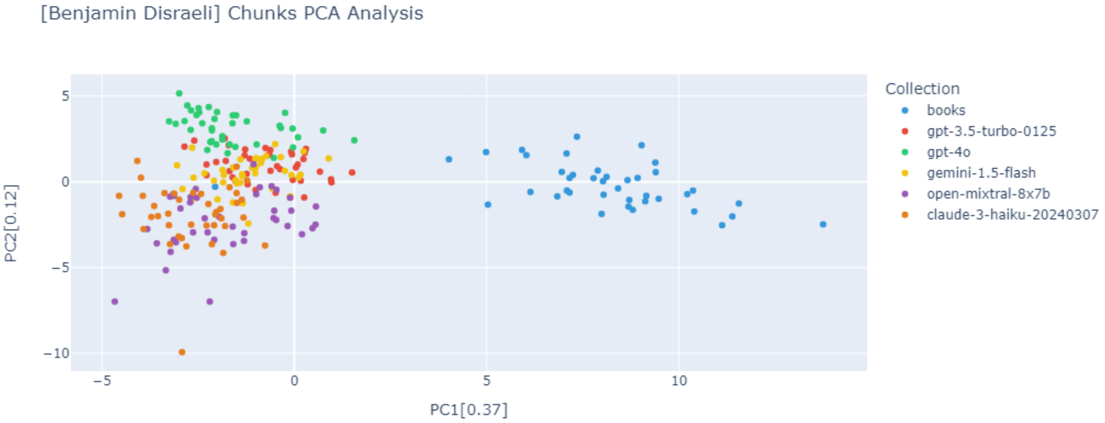


Figure 4.5: Scatter plot displays the results of PCA applied to text chunks of Benjamin Disraeli from Authors' Writing Style dataset, with each point representing a single chunk.

In a subsequent experiment, we applied a similar PCA process, but confined the analysis to individual authors. This involved training a separate PCA model on the text chunks of each author (240 chunks per author) and visualizing the results on a scatter plot. Figure

4.5, depicting the results for Benjamin Disraeli, serves as a representative example of the observed pattern across all authors. We found that within the narrowed scope of a single author, the distinction between human-written and model-generated texts becomes even more pronounced, facilitating clearer clustering.

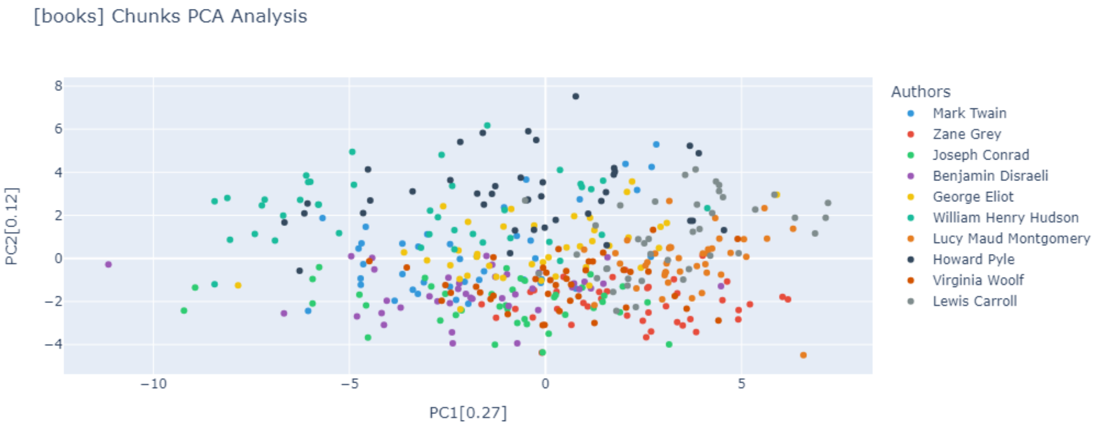


Figure 4.6: Scatter plot displays the results of PCA applied to text chunks of the books collection from Authors' Writing Style dataset, with each point representing a single chunk.

Our final set of PCA experiments focused on analyzing text chunks within the scope of individual collections. Figure 4.6 presents the results of this analysis for the book collection. We confirmed the observations from the plot of all chunks marked by author, namely the absence of clear clustering and the overlap of features from different authors. We observed similar behavior across the remaining collections under this focused analysis.

4.2 Classification

In the following experiments, we focus on classifying the chunks from the Authors' Writing Style dataset, as previously defined. Our objective is to employ various classification algorithms with different input features and target labels. It helps us validate our insights from *Principal Component Analysis*, which uncover the differences between human-written and llm-generated texts. This analysis aims to further investigate the writing styles of different sources and assess whether our approach is applicable across diverse domains. To examine its generalizability, we applied our methodology to the DAIGT-V4 dataset [24], which comprises essays authored by humans and various language models. Moreover, we will investigate the differences between the writing style of various authors rather than collections.

4.2.1 Authors' Writing Style Dataset - Core Classification

The PCA results revealed clear separations between human-written and model-generated texts, particularly when analyzed at the collection level. However, these findings primarily provide a visual and exploratory understanding. To quantitatively validate these distinctions, we next employ classification algorithms to assess how effectively these patterns can be used to predict text origins.

We applied the principal components obtained earlier to three classification algorithms: Logistic Regression, Support Vector Machines (SVM), and Decision Tree. Each of these algorithms was chosen to provide different perspectives on the classification task, ensuring the robustness of our findings. Logistic Regression is a statistical method often used for binary classification tasks. It models the probability of a sample belonging to a specific class by fitting a logistic function to the data. This algorithm is particularly effective when the relationship between the features and the target variable is linear.

Support Vector Machines (SVM) is a powerful supervised learning algorithm designed to find the hyperplane that best separates data points from different classes in a high-dimensional space. It aims to maximize the margin between the classes, which makes it robust to outliers and effective for non-linear classification tasks when used with kernel functions.

Decision Trees, on the other hand, are intuitive and interpretable models that split the data into subsets based on feature thresholds. They create a tree-like structure where each node represents a decision based on a feature, and each leaf represents a class label. Decision Trees are particularly suited for capturing non-linear patterns and interactions between features.

By using these diverse algorithms, we aimed to validate previously gathered insights regarding the principal components' ability to capture meaningful patterns in the data. Furthermore, comparing their performance allowed us to assess the extent to which different modeling approaches can leverage the same feature space.

The classification pipeline was consistent across all models. First, we extracted two, previously computed principal components from all text chunks, which served as inputs. The `collection` column was mapped to a binary label, assigning `human` for book collections and `llm` for texts generated by language models. The dataset was then split into training (80%) and test (20%) subsets using `sklearn.model_selection.train_test_split` with standard arguments. The `sklearn` implementations of the selected algorithms were trained on the training set, and their accuracy was evaluated on the test set. The trained models were saved for future use.

In the next experiment, we analyzed all chunks using the full set of 45 features defined in Section 4.1.3. As in the previous classification experiment, we labeled the chunks as either `human` or `llm`, with numerical labels 0 and 1 assigned respectively. Unlike the earlier approach, we used all 45 features instead of relying on principal components. We applied the same data split strategy, dividing the data into training and test sets.

For this experiment, we employed the XGBoost algorithm from `sklearn`, configured with the Logistic Regression loss function, which is designed for binary classification tasks. XGBoost (Extreme Gradient Boosting) is a highly efficient and scalable implementation of gradient-boosted decision trees. It builds an ensemble of weak learners (decision trees) sequentially, where each tree corrects the errors of the previous one. This approach often results in high predictive accuracy and is particularly effective for structured data. XGBoost is also well-known for its ability to handle missing data, regularization to prevent overfitting, and parallel processing capabilities, making it a powerful tool for this classification task.

To interpret the model's predictions and assess the importance of individual features, we used the `shap.TreeExplainer` implementation. SHAP (SHapley Additive exPlanations) is a game-theoretic approach to explain the output of machine learning models. It assigns each feature a Shapley value, quantifying its contribution to the model's prediction. SHAP values allow us to understand how each feature influences the classification of a particular instance. The `shap.TreeExplainer` is specifically optimized for tree-based models, such as XGBoost, providing efficient and interpretable visualizations of feature importance. By analyzing SHAP values, we gained insights into how specific features impacted the model's decision-making process, allowing us to identify patterns and validate the relevance of the selected features. These experiments encompass the core classification tasks applied to the Authors' Writing Style dataset and, as detailed in the next section, extend to the DAIGT-V4 dataset.

4.2.2 DAIGT-V4 - Core Classification

The classification results on the custom dataset might highlight the efficacy of our features and algorithms within the novel domain. However, these findings raise an important question: are these patterns domain-specific, or do they generalize to other text types? To address this, we extend our analysis to the DAIGT-V4 dataset, which includes essays from diverse sources.

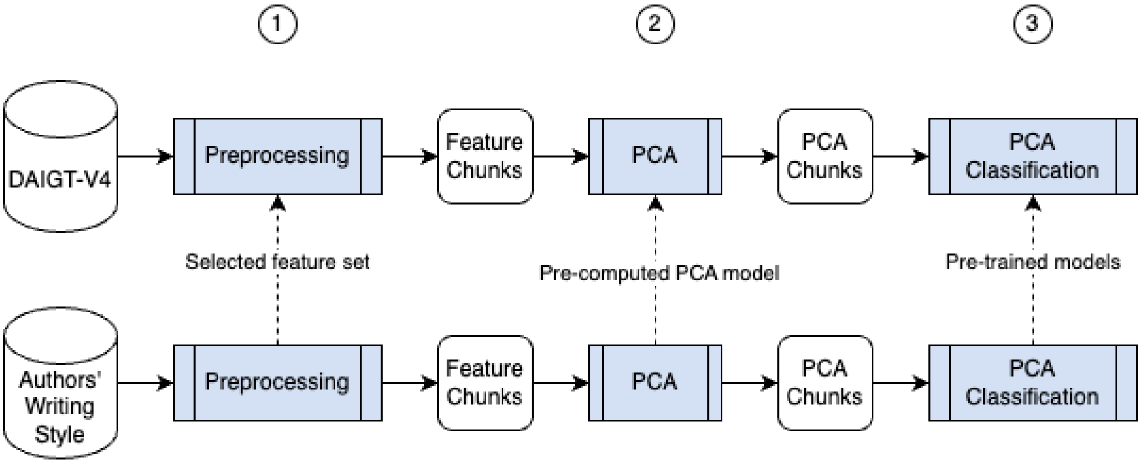


Figure 4.7: Schema of PCA Classification for the DAIGT-V4 dataset, illustrating dependencies on the Authors' Writing Style (AWS) pipeline: 1) Feature set defined during AWS preprocessing is reused, 2) AWS PCA models are applied for dimensionality reduction, and 3) Pre-trained classification models (Logistic Regression, SVM and, Decision Tree) from AWS are used for DAIGT-V4 classification.

We followed the same preprocessing pipeline as described for the Authors' Writing Style dataset in Section 4.1. After loading the data, we cleaned the model-generated texts using the same process as outlined previously. Texts written by humans were left untouched, as the initial investigation did not indicate a need for modification. The cleaning process removed 4,490 rows out of 73,573 (6.1%) identified as model-generated texts. To define our collections, we used the `model` column, which included categories such as *human*, *mistral*, *llama*, *gpt*, *claude*, *falcon*, *palm*, *cohere*, *ada*, *babbage*, *curie*, and *davinci*. In this experiment, models are grouped into families rather than specific versions; for instance, *gpt* may include *gpt3.5-turbo-01250* or other GPT variants. While this dataset does not provide specific author information, such as the human authors included in the Authors' Writing Style dataset, this limitation does not affect our experiment since the classification focuses on the original collection. The chunking preprocessing followed the same procedure as that described for model-generated texts in Section 4.1.3.

We reused the same features as in the previous experiment. Specifically, we applied the previously calculated top punctuation and function words without recalculating them. These features served as inputs to the principal component space and scaler from the earlier analysis, resulting in a two-dimensional input alongside the collection column. For the classification task, we transformed the collection column to `llm` for all values except `human`. Additionally, we reused the three pre-trained models—Logistic Regression, SVM, and Decision Tree—without retraining them to classify all chunks obtained during the preprocessing stage. These steps aimed to evaluate whether our analysis generalizes to other domains or overfits to a specific domain.

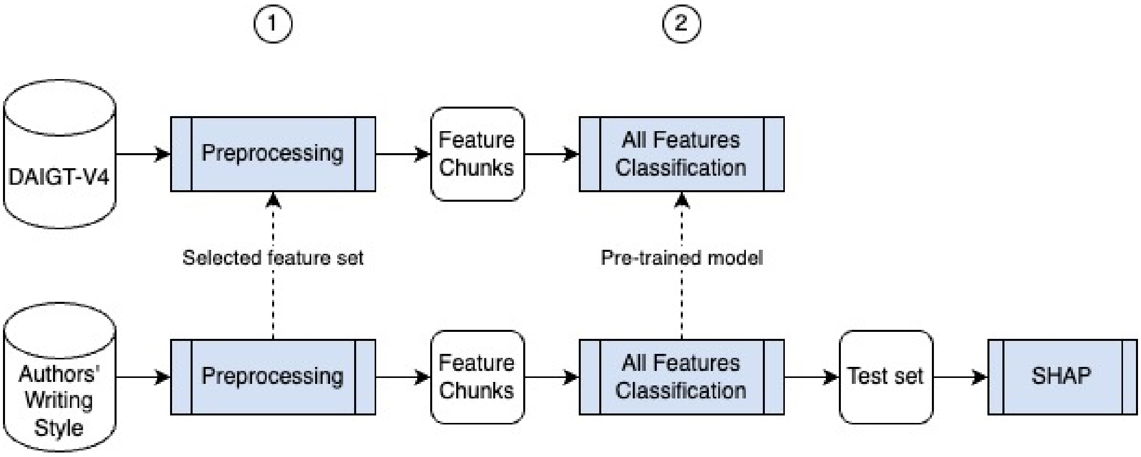


Figure 4.8: Schema of All Features Classification for the DAIGT-V4 dataset, illustrating dependencies on the Authors' Writing Style (AWS) pipeline: 1) Feature set defined during AWS preprocessing is reused, 2) Pre-trained XGBoost classification model from AWS are used for DAIGT-V4 classification.

We followed the same approach regarding the XGBoost algorithm with all features as input. The set of features was the same as defined in Section 4.1.3, and we reused the model trained on the Authors' Writing Style dataset.

The final experiment, extending beyond the scope of *Authors' Writing Style - Core Classification*, involved training and predicting the label of each DAIGT-V4 chunk utilizing all available features. In this task, we adopted a similar methodology to the previous experiment; however, we trained the model on the essays domain instead of reusing the model trained on the novel domain. Following a consistent procedure, we incorporated all chunks and the complete set of 45 features, divided the data into training and test sets using an 80:20 split strategy, assessed the accuracy of the predictions on the test set, and analyzed feature impacts through SHAP values. This experiment aimed to evaluate the classification performance on this dataset under a straightforward scenario.

4.2.3 Authors' Writing Style Dataset - Additional Classification Experiments

While the core classification experiments could demonstrate high accuracy in distinguishing text origins, they primarily operated at the collection level. To further investigate the granularity of writing styles, we explore classification at the author level. This deeper analysis seeks to understand whether distinct authorial styles remain discernible within individual collections.

Initially, we focused on a scenario where PCA was applied separately to chunks from each collection. The goal was to classify chunks into one of ten authors within the constraints of a single collection's scope. Following the standard methodology, we split the chunks into training and test sets and applied three previously described classification models - Logistic Regression, SVM and, Decision Tree. This experiment aimed to evaluate the feasibility and efficiency of author classification under controlled conditions.

In the subsequent step, we further simplified the classification problem by selecting chunks from only two specific authors within a single collection, reducing the task to binary classification. By focusing on this more relaxed scenario, we sought to understand how easily specific authors could be distinguished within the same collection. These additional experiments were designed to provide deeper insights into the granularity of author-level distinctions, complementing our broader analyses.

4.2.4 Results

Our initial experiments on the custom Authors' Writing Style dataset confirmed that distinguishing between human-generated and model-generated texts is feasible. Using the principal components of each chunk as input, enabled us to successfully learn various models distinguish human and model texts. Evaluated the classification performance of the three models, the results were as follows:

1. *Logistic Regression:* 99.6%
2. *SVM:* 99.3%
3. *Decision Tree:* 98.3%

The consistently high accuracy (around 99%) across all algorithms indicates that the extracted features, reduced via PCA, effectively abstract the text chunks and allow accurate classification of their origins. These results validate the suitability of our feature extraction and reduction approach for in-domain data.

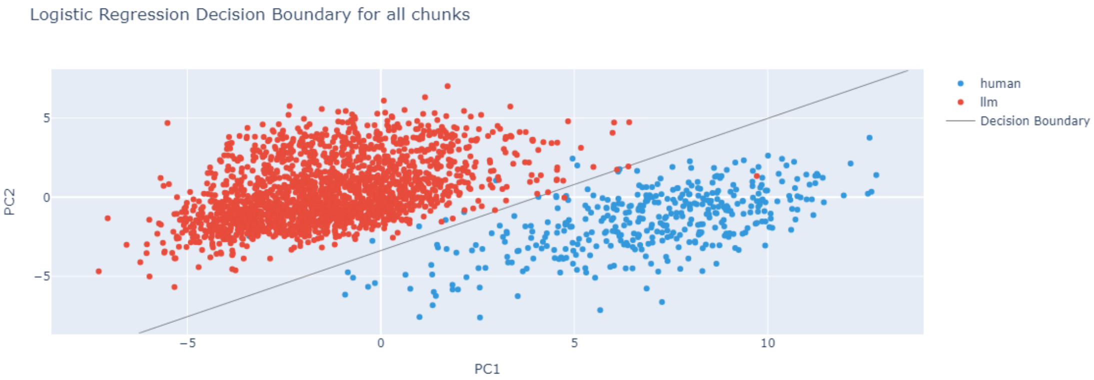


Figure 4.9: Scatter plot visualizes each text chunk as a single point, derived from a Principal Component Analysis (PCA) performed on all available chunks from Authors' Writing Style dataset. Points are colored blue for the books collection and red otherwise. Additionally, the decision boundary of trained logistic regression model is displayed.

The XGBoost model achieved an accuracy of 100% on the classification task, demonstrating that a more comprehensive feature set combined with a robust algorithm results in superior performance. Additionally, the use of SHAP allowed us to interpret the influence of individual features, as shown in Figure 4.10. The results indicate that decisions were primarily influenced by punctuation, function words, and textual score metrics. Specifically, high frequencies of punctuation marks, such as the period (.), dash (-), and semicolon (;), are characteristic of human-generated texts. However, function words presented a more nuanced pattern: while certain function words, such as ”was,” ”said,” and ”he,” were more typical of human writing, others, like ”our,” were more frequently associated with model-generated texts.

Further analysis of Yule's Characteristic K and Simpson's Index revealed that human-written texts exhibit higher lexical richness, characterized by less repetition and more varied word usage. It also tells us that human writing styles also tend to align with a more even distribution of word frequencies, demonstrating greater diversity in word choice compared to model-generated texts.

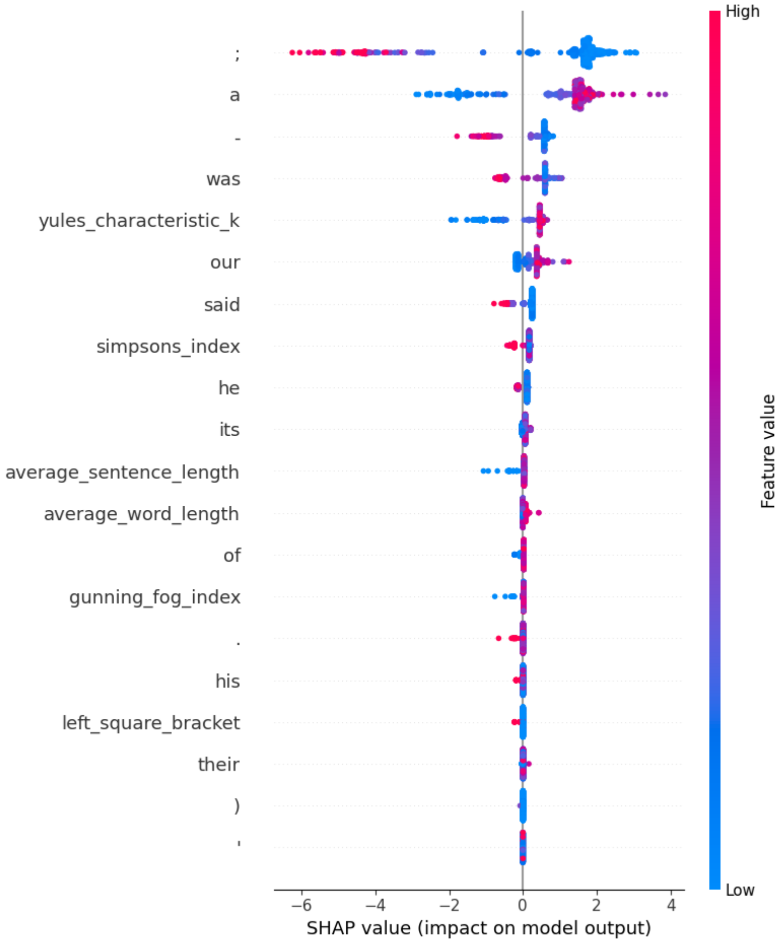


Figure 4.10: Beeswarm plot illustrating the impact of the ten most influential features on the XGBoost model's decisions trained and tested on Authors' Writing Style dataset. The visualization, generated using SHAP, is based on the test set of the full dataset comprising all chunks.

An important observation from the SHAP plot is the distribution of feature values. Features that pushed predictions toward model-generated texts (right-hand side of the plot) often exhibited low variance, as seen in "-", "was," "simpson_index," and "said." Conversely, human-generated texts displayed much greater variability in feature values, particularly for the top-ranked features. In general, features with higher variance had the most significant impact on the model's predictions. This highlights the diversity inherent in human-written texts, as opposed to the more uniform and averaged style of model-generated outputs. These findings align with our observations from the PCA visualization of all chunks by collection (4.2), further supporting the notion that human writing styles are more diverse than those generated by language models.

In the next experiment, we classified the principal components of DAIGT-V4 chunks using models pre-trained on our custom dataset. The results revealed a lack of flexibility in our approach. The final accuracies for each model were as follows:

1. *Logistic Regression:* 59.7%

2. *SVM:* 60.3%
3. *Decision Tree:* 60%

We observed that the current setup achieved an accuracy of only around 60% for out-of-domain data, which can be considered very low compared to our in-domain results. Moreover, the pre-trained XGBoost model with all features as input achieved only 66% accuracy, further highlighting the failure of this approach. However, the XGBoost model trained on the DAIGT-V4 full-featured data and tested on the same dataset achieved a high accuracy of 93%, demonstrating the potential for strong performance on this data.

Top	Authors' Writing Style	DAIGT-V4
1	;	averages_syllables_per_word
2	a	gunning_fox_index
3	-	,
4	was	unique_word_count
5	yules_characteristic_k	.

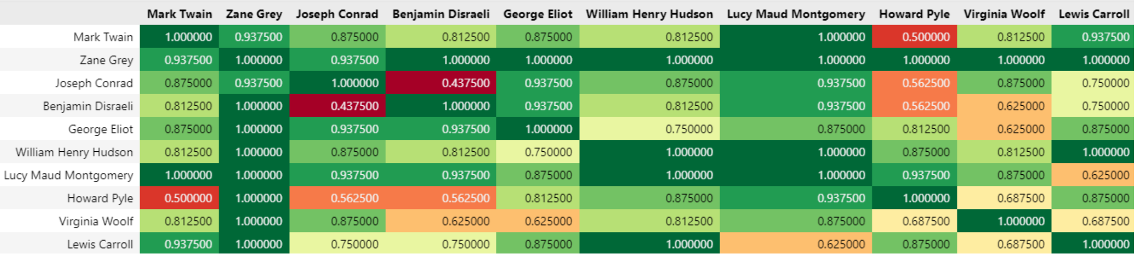
Table 4.2: Top five features according to SHAP analysis for two datasets trained and tested on in-domain data — Authors' Writing Style dataset and DAIGT-V4 dataset.

These results indicate that our approach overfits to a specific domain. One reason for this could be the choice of features. Table 4.2 highlights the top five features according to SHAP analysis for both datasets. It underscores the importance of different features, emphasizing numerical metrics of text structure such as ”averages_syllables_per_word” and ”unique_word_count for DAIGT-V4 dataset”. SHAP analysis reveals that human texts exhibit less sophisticated features, such as a low ”gunning_fox_index,” which aligns with the human writers being 6th-12th grade students. Moreover, punctuation signs also play a significant role, however, a different set of these characters is considered.

Collection	Logistic Regression	SVM	Decision Tree
books	0.4	0.4	0.375
gpt-3.5-turbo-0125	0.325	0.2375	0.1875
gpt-4o	0.275	0.3	0.15
gemini-1.5-flash	0.3125	0.325	0.2875
open-mixtral-8x7b	0.2	0.225	0.1875
claude-3-haiku-20240307	0.3	0.325	0.2625

Table 4.3: The accuracy of classifying one of ten authors in the scope of each collection listed on the left. The same experiment has been repeated for three models listed on the top row.

To analyze the granularity of writing styles, we conducted author-level classification experiments within individual collections. These experiments aimed to evaluate whether distinct authorial styles are discernible, using PCA for dimensionality reduction and three classifiers (Logistic Regression, SVM, and Decision Tree). By testing both multi-class classification across ten authors and binary classification for two authors, we gained deeper insights into the feasibility and precision of distinguishing authors within the same collection. In this final experiments, we observed that distinguishing authors' writing styles within the scope of a single collection was infeasible. The average accuracies across collections were as follows: 30.2% for Logistic Regression, 30.2% for SVM, and 24.2% for Decision Tree. The detailed accuracies are presented in Table 4.3. These results fall significantly below a satisfactory performance threshold, indicating that our current setup cannot effectively differentiate between authors' texts. However, we observed that the accuracies for the books collection were slightly higher compared to the others. This trend may support the finding about greater variability in human writing styles, potentially making it easier for our architecture to distinguish between authors in this context.



	Mark Twain	Zane Grey	Joseph Conrad	Benjamin Disraeli	George Eliot	William Henry Hudson	Lucy Maud Montgomery	Howard Pyle	Virginia Woolf	Lewis Carroll
Mark Twain	1.000000	0.937500	0.875000	0.812500	0.875000	0.812500	1.000000	0.500000	0.812500	0.937500
Zane Grey	0.937500	1.000000	0.937500	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
Joseph Conrad	0.875000	0.937500	1.000000	0.437500	0.937500	0.875000	0.937500	0.562500	0.875000	0.750000
Benjamin Disraeli	0.812500	1.000000	0.437500	1.000000	0.937500	0.812500	0.937500	0.562500	0.625000	0.750000
George Eliot	0.875000	1.000000	0.937500	0.937500	1.000000	0.750000	0.875000	0.812500	0.625000	0.875000
William Henry Hudson	0.812500	1.000000	0.875000	0.812500	0.750000	1.000000	1.000000	0.875000	0.812500	1.000000
Lucy Maud Montgomery	1.000000	1.000000	0.937500	0.937500	0.875000	1.000000	1.000000	0.937500	0.875000	0.625000
Howard Pyle	0.500000	1.000000	0.562500	0.562500	0.812500	0.875000	0.937500	1.000000	0.687500	0.875000
Virginia Woolf	0.812500	1.000000	0.875000	0.625000	0.625000	0.812500	0.875000	0.687500	1.000000	0.687500
Lewis Carroll	0.937500	1.000000	0.750000	0.750000	0.875000	1.000000	0.625000	0.875000	0.687500	1.000000

Figure 4.11: Heatmap representing the accuracy of classifying book chunks from Authors' Writing Style dataset between two authors. Values are the result of Logistic Regression model.

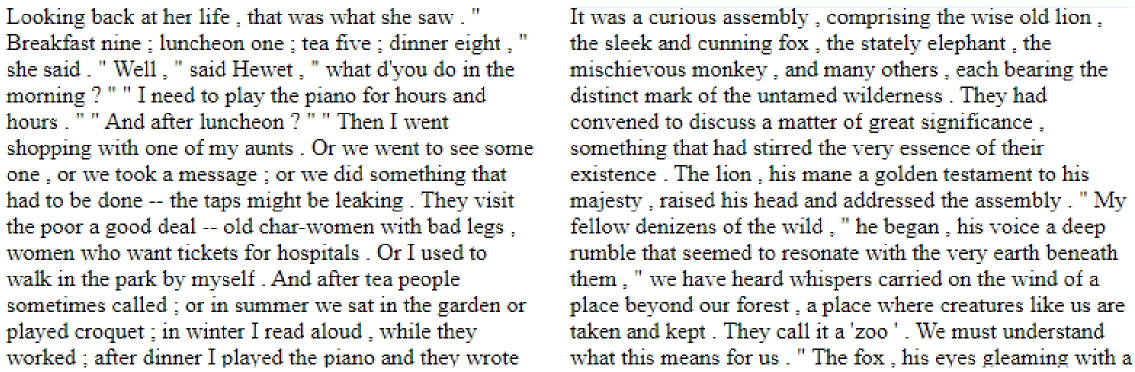
In contrast, when we focused on a simpler scenario—classifying chunks from a single collection between only two authors—we achieved significantly better results. However, as shown in Figure 4.11, classifying certain author pairs remains challenging. These results were consistent across models. Interestingly, regardless of the model provider, the generated texts exhibited a similar level of difficulty in binary author classification, and in some cases, performance was even lower.

The findings from these classification experiments either corroborate previous results or provide new insights. These outcomes serve as a foundation for the third part of our study, which we elaborate on in the next section.

4.3 Information Content Analysis

In the third part of our analysis, we investigate the novelty present in human-written and model-generated texts within the Authors' Writing Style dataset. To begin, we present two chunks in Figure 4.12, carefully selected to represent central characteristics of their respective collections. We invite the reader to examine these chunks closely and try to guess which one is human-written and why. These specific chunks were chosen because they highlight key stylistic and structural differences between human and model-generated texts, making them ideal for this exploration. Throughout this section, we analyze these differences in detail, ultimately revealing the origins of the chunks. Specifically, we quantify the diversity and unpredictability of the text, which provides a measure of its uniqueness. To achieve this, we apply Information Content Theory, as introduced by James V. Stone [38]. This theoretical framework enables us to mathematically quantify the amount of information within an entity, measured in bits, making it particularly suited for analyzing textual features.

We employ Shannon Information Content (SIC) [38] to estimate the information encoded in linguistic features within chunks of text. SIC provides a precise way to quantify how



Looking back at her life , that was what she saw . " Breakfast nine ; luncheon one ; tea five ; dinner eight , " she said . " Well , " said Hewet , " what d'you do in the morning ? " " I need to play the piano for hours and hours . " " And after luncheon ? " " Then I went shopping with one of my aunts . Or we went to see some one , or we took a message ; or we did something that had to be done -- the taps might be leaking . They visit the poor a good deal -- old char-women with bad legs , women who want tickets for hospitals . Or I used to walk in the park by myself . And after tea people sometimes called ; or in summer we sat in the garden or played croquet ; in winter I read aloud , while they worked ; after dinner I played the piano and they wrote

It was a curious assembly , comprising the wise old lion , the sleek and cunning fox , the stately elephant , the mischievous monkey , and many others , each bearing the distinct mark of the untamed wilderness . They had convened to discuss a matter of great significance , something that had stirred the very essence of their existence . The lion , his mane a golden testament to his majesty , raised his head and addressed the assembly . " My fellow denizens of the wild , " he began , his voice a deep rumble that seemed to resonate with the very earth beneath them , " we have heard whispers carried on the wind of a place beyond our forest , a place where creatures like us are taken and kept . They call it a 'zoo ' . We must understand what this means for us . " The fox , his eyes gleaming with a

Figure 4.12: Truncated beginning of two chunks. One originates from model-generated text and the other from human-written text (not necessarily in the display order). Due to the post-merge operation, the original spacing may be altered.

much information is conveyed by an event, such as the occurrence of specific linguistic attributes. In our context, it allows us to determine how unexpected or novel a text chunk is based on the probability of its linguistic properties.

The Shannon Information Content is defined mathematically as:

$$\text{bits} = -log_2(p(x)) \tag{4.1}$$

where $p(x)$ represents the probability of an event, which in our analysis corresponds to the likelihood of linguistic features appearing within a chunk of text. This approach enables us to evaluate text novelty by capturing the unpredictability of linguistic elements, providing valuable insights into the differences between human-authored and model-generated texts.

4.3.1 Text features

We focus on various linguistic attributes, starting with the chunk features introduced in Section 4.1.3. Specifically, we used all 45 features extracted from the 2400 text chunks in our dataset. To facilitate further analysis, we discretize the feature values by mapping them into bins. For each feature, we determine its minimum and maximum values across the 2400 chunks and divide this range into 100 equally spaced bins. This approach allows us to construct probability distributions for each feature based on the entire dataset.

Next, we compute the Shannon Information Content (SIC) for each chunk, processing the features individually. For each feature, we identify the corresponding bin for the feature value and extract its probability from the precomputed distribution. The extracted probability is then substituted into Equation 4.1, yielding a numerical SIC value. This systematic approach ensures that we quantify the information content for each chunk in a consistent and reproducible manner. By doing so, we assess how unique the features of a specific chunk are, thereby highlighting its level of novelty within the dataset.

4.3.2 Words distributions

The next step involves investigating word usage within the gathered texts. To achieve this, we construct two probability distributions: a local word probability distribution based on the Authors' Writing Style dataset and a global word probability distribution derived from all the datasets listed in Section 3. These distributions enable us to analyze the uniqueness and rarity of word usage across different corpora.

The local word probability distribution is computed using all the words present in the chunks, each with approximately 200,000 words, produced by the preprocessing pipeline described in Section 4.1.2. We aggregate repeated words and calculate their probabilities of occurrence based on the total number of words in the dataset. This process results in a distribution of approximately 67,000 unique words, which have been lowered before the process. The purpose of this distribution is to examine how specific sources—whether collections of texts or individual authors—exhibit distinct word usage patterns, thereby highlighting their uniqueness within the corpus.

The global word probability distribution is constructed using all the words from the combined datasets: Authors' Writing Style, DAIGT-V4, Global News, Legal Contracts, Twitter, and Reddit. This diverse collection of datasets spans various domains, ensuring that the resulting distribution captures a wide range of language use and enhances the robustness of the analysis. To tokenize the text, we employ a simple whitespace-based split instead of a dedicated tokenizer. While this method may introduce minor errors, these are negligible compared to the computational efficiency it provides, allowing us to handle the large-scale data effectively. In total, the global corpus includes approximately 641 million words, all lowered before the further steps, offering a comprehensive basis for constructing the probability distribution. As with the local distribution, repeated words are aggregated, and probabilities are calculated based on their frequencies within the corpus. This distribution provides insights into the rarity and commonality of words in the analyzed texts, enabling a deeper understanding of linguistic patterns.

At this stage, we extend the set of features' SIC scores by incorporating two additional attributes: the Local Word Probability Distribution and the Global Word Probability Distribution. For each chunk, we iterate through its lowered words, summing the probabilities of each word based on the respective local and global distributions. We then divide the total probability by the number of words in the chunk, yielding a normalized probability for each attribute. Finally, we apply Equation 4.1 to these normalized probabilities, extracting SIC scores for the local and global word probability distribution attributes. This enhanced feature set allows us to evaluate the information content of each chunk, considering both local uniqueness and global rarity in word usage. Resulting SIC scores of these distribution are presented in the later subsections.

4.3.3 Words sequence

The final attribute we incorporate is the sequence entropy, which quantifies the number of bits required to encode a sequence of words. This measure is based on the work of James P. Bagrow [39], who analyzed tweets to study information flow in social networks. One of the key metrics used in his research is the sequence entropy rate (h), defined as the average number of bits needed to encode the next word in a sequence, given the preceding words. We adopt this metric as an additional information content (IC) attribute for our analysis.

The sequence entropy rate is calculated using the following formula:

$$h = \frac{NlogN}{\sum_{i=1}^{N} \Lambda_i} \tag{4.2}$$

where N is the length of the sequence of words and Λ_i is the match length of the prefix at position i, that is, it is the length of the shortest subsequence (of words) starting at i that has not previously appeared. All words are lowered for the matching process.

To illustrate, consider the example word sequence:

```
Example sequence presenting example sequence entropy.
```

The corresponding match lengths (Λ) are as follows:

```
[1, 1, 1, 3, 2, 1]
```

We apply this method to the list of words in each chunk, calculating the sequence entropy for each. This results in a new attribute, referred to as sequence IC, which provides additional insights into the degree of repetition within the analyzed texts. By capturing patterns of word recurrence, this attribute complements the other characteristics and enhances our understanding of the information dynamics of the text.

4.3.4 Average Samples

Previously, we introduced several attributes—Feature SIC, Local and Global Word Distribution SIC, and Sequence IC—that collectively form the set of information content (IC) attributes used in our analysis. Once the IC values for each chunk have been calculated, we identify the average sample for each collection by computing the mean and standard deviation for each attribute across all chunks within the collection.

To calculate the mean for an attribute, we use the following formula:

$$m = \frac{1}{n} \sum_{j=1}^{n} y_j$$

where n represents the total number of chunks in the collection, and y_j is the IC attribute value for the j^{th} chunk.

The standard deviation is calculated as:

$$S = \sqrt{\frac{s^2}{n}}$$

where s, the sample standard deviation, is defined as:

$$s = \sqrt{\frac{\sum_{j=1}^{n} (y_j - m)^2}{n - 1}}$$

is the standard deviation for the sample.

With the mean (m) and standard deviation (S) for each IC attribute computed, we proceed to calculate the weighted distance for each chunk. This distance quantifies how far a chunk's IC attributes values deviate from the average sample of its collection. The weighted distance is defined as:

$$d = \sum_{a \in A} \frac{1}{S_a} * (x_a - m_a)^2$$

where A is the set of IC attributes, S_a is the standard error for attribute a, x_a is the IC value of attribute a for a specific chunk, and m_a is the mean IC value of attribute a.

The weighting by $\frac{1}{S_a}$ mitigates the impact of attributes with inherently higher variance, ensuring that the contributions of all attributes are balanced. This prevents the results from being skewed due to differences in attribute variability.

We compute the weighted distance d for every chunk within a collection. The average sample for each collection is then identified as the chunk with the smallest distance d from the mean.

This representative sample provides a central point for each collection and serves as the foundation for visualizations in the results section. Using this approach, we ensure that the visualized chunks accurately reflect the typical information content attributes of their respective collections. First, we present the Sequence IC Visualization, where we highlight each token based on its match length (Λ). Tokens with a match length of 2 or higher are shaded in yellow, with darker shades indicating higher match lengths. Next, we introduce the Displayable SIC Features Visualization, focusing on function words and punctuation. From these features, we select the top ten based on their SIC values, representing the features most unique to each chunk. It is important to note that displayable SIC features do not include other metrics such as average word length or the Simpson index, as these cannot be visualized on the token level.

4.3.5 Results

In the preceding analysis, we explored several information content (IC) attributes, including feature SIC, Local and Global Word Distribution SIC, and Sequence IC, across various collections. Here, we aim to deepen the insights by examining the distributions of these IC attributes across collections and highlighting key stylistic differences between human-written and model-generated texts.

Figure 4.13 and 4.14 presents the IC value distributions for two representative collections: books and *gpt-4o*. The distributions for *gpt-4o* align largely with those of other model-generated collections, showcasing relatively consistent patterns. However, the books collection stands out with noticeably higher variance across most attributes. This higher variance indicates that the linguistic attributes in the books collection are more diverse and less predictable compared to those in model-generated texts.

Particularly noteworthy are the last three attributes—Sequence IC, Local Word Distribution, and Global Word Distribution—which exhibit minimal variance across all collections. This observation suggests a lack of repetitive patterns in the texts and a comparable distribution of word usage. Such consistency in these attributes further underscores the similarities in the linguistic styles of the model-generated collections. On the other hand, similar variance observed in the books collection indicates that human-written texts do not stand out in this dimension.

Figure 4.15 provides a heatmap of IC values for the average chunk of each collection. The average chunk, as described earlier, serves as a representative sample for the entire collection. By analyzing these values, we can infer the stylistic characteristics of each collection. The heatmap confirms the observations from the variance analysis, showing that the last three attributes are consistently similar across collections. Notably, the books collection stands out with higher IC values across multiple attributes, emphasizing its uniqueness.

Interestingly, certain function words and punctuation marks play a significant role in distinguishing the books collection. Words such as ”of,” ”she,” ”her,” and ”he,” along with punctuation marks like the semicolon (;) and double quotation marks (”), are more prevalent in the books collection compared to the model-generated collections. These findings align with the trends observed in the earlier analyses. While there are some exceptions—for example, the elevated IC value of the word ”she” in the *claude-3-haiku-20240307* and *gpt-3.5-turbo-0125* collections, or ”the” in *claude-3-haiku-20240307*—the overall distribu-

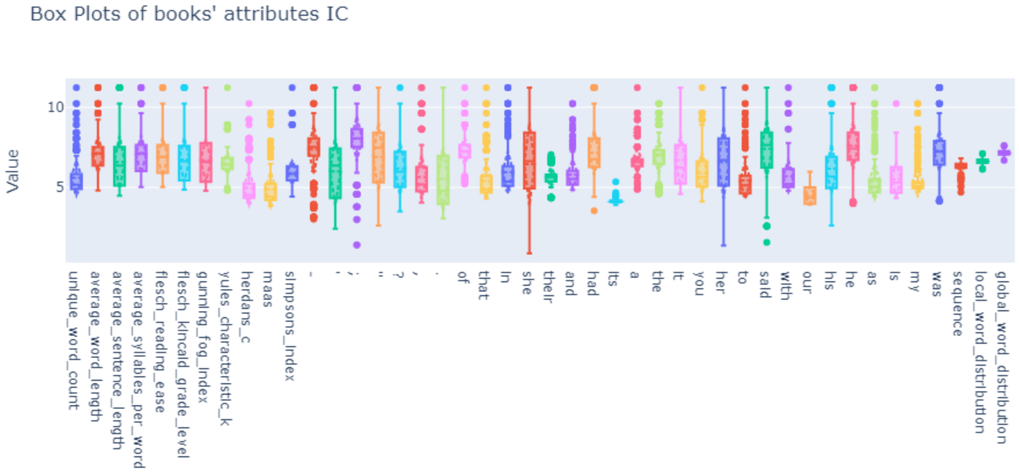


Figure 4.13: Books collection's IC values distribution.

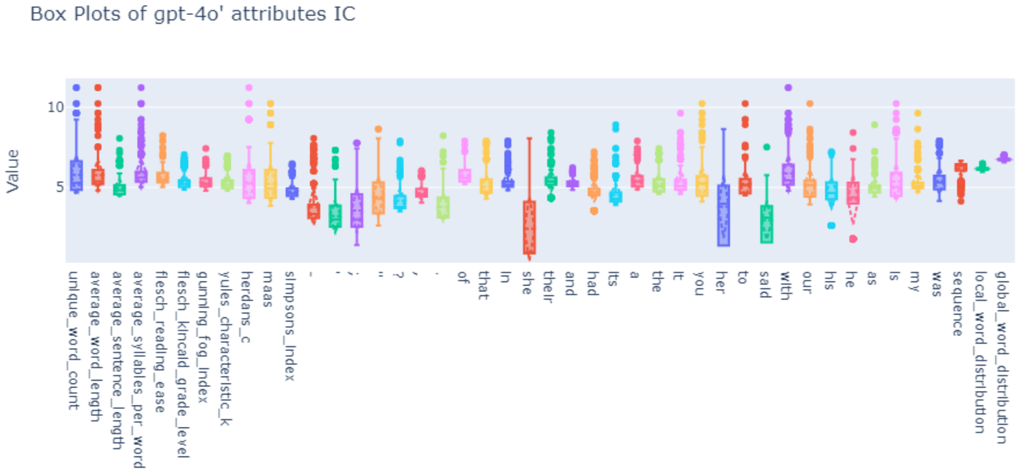


Figure 4.14: Gpt-4o collection's IC values distribution.

tion of IC values across model-generated collections is relatively uniform.

Figure 4.16 explores further the sequence IC attribute by presenting truncated samples of the average chunks from the books and *gpt-4o* collections. We observe that neither chunk demonstrates significant token repetition, apart from the natural repetition expected in language. The sequence IC values for these chunks, 6.43 for books and 6.25 for *gpt-4o*, confirm the similarity in token repetition patterns across collections. This result suggests that the model-generated texts are constructed in a way that avoids excessive token repetition, reflecting a well-engineered balance in their generative processes, simulating human style in this regard.

Figure 4.17 further examines the linguistic attributes by highlighting the top ten displayable features (function words and punctuation marks) in the average chunks of books and *gpt-4o*. A key distinction is the higher frequency of semicolon usage (;) in the books collection, compared to the frequent use of the function word "of" in the *gpt-4o* collection. Additionally, while not immediately visible in the figure, the books collection also demonstrates

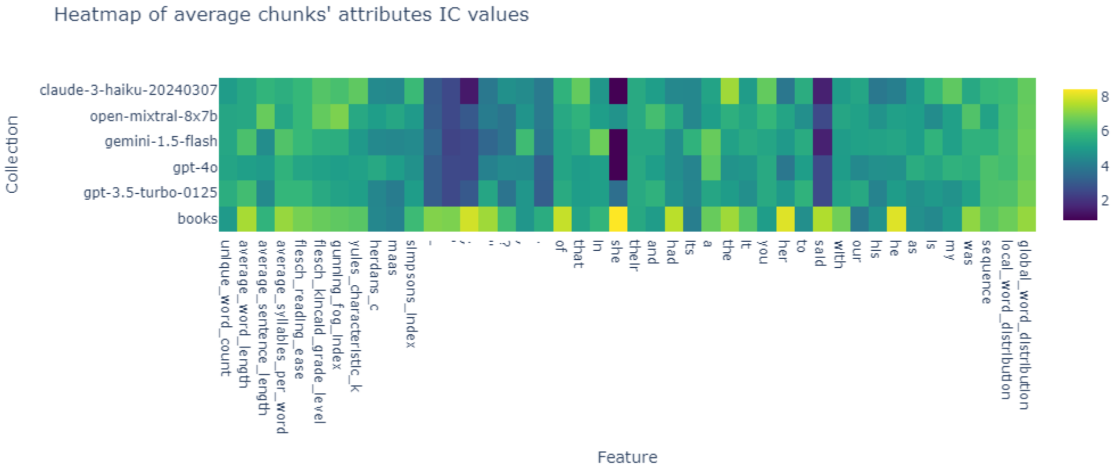
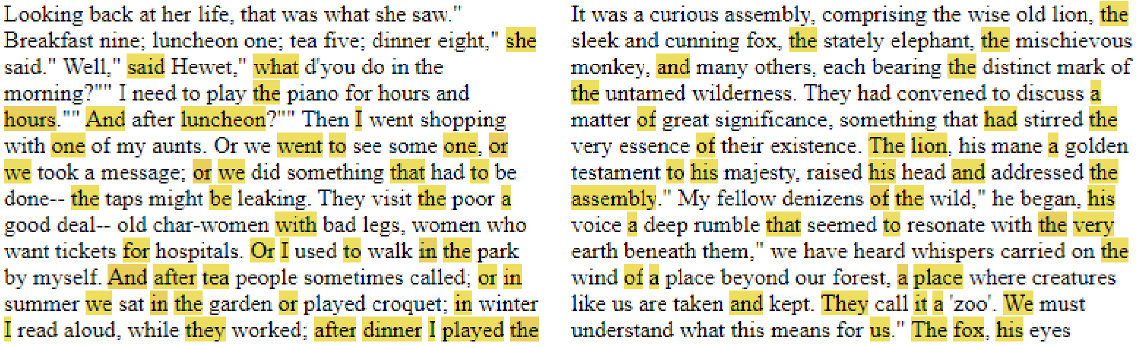


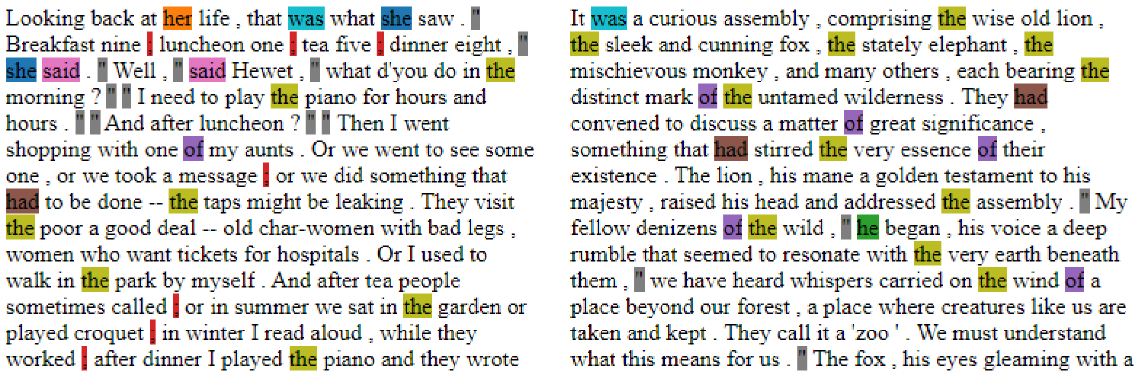
Figure 4.15: Heatmap representing IC value for each attribute for each collection's average chunk.



Looking back at her life, that was what she saw." Breakfast nine; luncheon one; tea five; dinner eight," she said." Well," said Hewet," what d'you do in the morning?"" I need to play the piano for hours and hours."" And after luncheon?"" Then I went shopping with one of my aunts. Or we went to see some one, or we took a message; or we did something that had to be done-- the taps might be leaking. They visit the poor a good deal-- old char-women with bad legs, women who want tickets for hospitals. Or I used to walk in the park by myself. And after tea people sometimes called; or in summer we sat in the garden or played croquet; in winter I read aloud, while they worked; after dinner I played the

It was a curious assembly, comprising the wise old lion, the sleek and cunning fox, the stately elephant, the mischievous monkey, and many others, each bearing the distinct mark of the untamed wilderness. They had convened to discuss a matter of great significance, something that had stirred the very essence of their existence. The lion, his mane a golden testament to his majesty, raised his head and addressed the assembly." My fellow denizens of the wild," he began, his voice a deep rumble that seemed to resonate with the very earth beneath them," we have heard whispers carried on the wind of a place beyond our forest, a place where creatures like us are taken and kept. They call it a 'zoo'. We must understand what this means for us." The fox, his eyes

Figure 4.16: Truncated beginning of two average chunks of books and gpt-4o collections respectively. Visualization is marked by the values of match lengths (Λ) of each token.



Looking back at her life , that was what she saw . " Breakfast nine ; luncheon one ; tea five ; dinner eight , " she said . " Well , " said Hewet , " what d'you do in the morning ? " " I need to play the piano for hours and hours . " " And after luncheon ? " " Then I went shopping with one of my aunts . Or we went to see some one , or we took a message ; or we did something that had to be done -- the taps might be leaking . They visit the poor a good deal -- old char-women with bad legs , women who want tickets for hospitals . Or I used to walk in the park by myself . And after tea people sometimes called ; or in summer we sat in the garden or played croquet ; in winter I read aloud , while they worked ; after dinner I played the piano and they wrote

It was a curious assembly , comprising the wise old lion , the sleek and cunning fox , the stately elephant , the mischievous monkey , and many others , each bearing the distinct mark of the untamed wilderness . They had convened to discuss a matter of great significance , something that had stirred the very essence of their existence . The lion , his mane a golden testament to his majesty , raised his head and addressed the assembly . " My fellow denizens of the wild , " he began , his voice a deep rumble that seemed to resonate with the very earth beneath them , " we have heard whispers carried on the wind of a place beyond our forest , a place where creatures like us are taken and kept . They call it a 'zoo ' . We must understand what this means for us . " The fox , his eyes gleaming with a

Figure 4.17: Truncated beginning of two average chunks of books and gpt-4o collections respectively. Visualization is marked by the top ten displayable features (function words and punctuation characters) IC values, which tokens are highlighted.

a higher frequency of feminine words such as ”her” and ”she,” further aligning with the trends observed in the heatmap (Figure 4.15).

In summary, our analysis reveals significant differences between human-written and model-generated texts. Human-written texts, as represented by the books collection, exhibit greater variability, uniqueness, and unpredictability across IC attributes. In contrast, model-generated texts display a more uniform style, characterized by lower variance in linguistic features. These findings highlight the nuanced stylistic differences between human authorship and machine-generated texts, providing valuable insights into the linguistic characteristics of these two distinct text sources.

5 Discussion

In this chapter, we delve into the key findings of our study, analyzing the outcomes of various experiments and their implications for understanding the stylistic differences between human and LLM-generated texts. By leveraging techniques such as Principal Component Analysis, Classification Models, and Information Content analysis, we explore the distinctive patterns in writing styles and assess the capabilities and limitations of language models in emulating human authorship, and moreover, the novelty of generated texts. This discussion contextualizes the results within existing literature, highlights the challenges faced, and underscores the broader significance of our findings for authorship attribution and stylometric analysis.

5.1 Principal Component Analysis Results

Through our application of PCA, we sought to address the question: Can we distinguish human-written texts from model-generated ones? Using our custom dataset, the Authors' Writing Style dataset, we observed clear visual separation between model-generated and human-written texts, as illustrated in Figure 4.2. This result confirms our hypothesis, highlighting distinct differences between these two groups. Interestingly, we identified two prominent clusters—one for human-written texts and another for model-generated texts—rather than six distinct clusters representing individual models, analogical to the human authors clusters in the research of Binongo [16]. This finding suggests that generated texts exhibit notable stylistic similarities, irrespective of their source. This uniformity indicates that different large language models (LLMs) share common stylistic habits, which is a critical observation for addressing subsequent research questions.

We observed that the texts generated by *gemini-1.5-flash* were slightly separated from the other model-generated texts, forming a unique model-generated group that partially overlaps with the main cluster. This divergence, along with the observed similarities among other models, may be influenced by differences in the underlying architectures or the corpora used for training. Models trained on overlapping open-source datasets may exhibit similar stylistic patterns, while models with access to unique or proprietary datasets might develop distinctive characteristics. Additionally, shared architectural designs could contribute to the uniformity seen in some models, whereas variations in design principles or implementation, as in the case of *gemini-1.5-flash*, may account for the observed stylistic differences. However, since most of these models remain proprietary, we are unable to definitively determine the specific factors driving these variations.

We further examined the clustering of chunks based on authorship, as shown in Figure 4.3, but found no clear clusters corresponding to specific authors. This result suggests that the writing styles of different human authors overlap significantly, making it challenging to distinguish individual authors' chunks. This overlap may indicate limitations in our feature selection, which might not sufficiently capture the nuanced characteristics of individual writing styles. Alternatively, it could reflect the influence of the single-domain nature of our dataset, where the shared characteristics of novels obscure distinctions between authors.

Additionally, we observed that requesting a specific author's writing style had no discernible effect on clustering within the larger group of model-generated texts. It appears that this aspect of the prompt was either ignored or did not meaningfully influence the models' outputs. The lack of clear distinctions between human authors' styles might also contribute to this phenomenon, as it limits the models' ability to mimic specific authors effectively within the constraints of the dataset.

We visualized the importance of the top ten features for the first and second principal components, which provided initial insights into the key elements defining a chunk's writing style. Our analysis revealed that scores derived from metrics such as the number of words, syllables, sentences, and complex words play a crucial role in distinguishing writing styles. We also identified the significant contribution of function words, emphasizing their importance in our classification approach. These findings underscore the relevance of linguistic and structural features in differentiating between human and model-generated texts.

5.2 Classification

Through our classification experiments, we aimed to validate and extend the insights obtained from PCA analysis, while shedding light on the distinguishing writing features between human-written and LLM-generated texts. The first phase of our experiments provided quantitative evidence that the visual clusters observed in PCA translate into high classification accuracy. This confirmed the feasibility of distinguishing human-written texts from model-generated ones. Moreover, through SHAP analysis, we observed that human writers tend to use punctuation more frequently and demonstrate higher lexical richness, which aligns with findings of Bhandarkar et al [17]. These factors emerged as the primary features separating the two types of text. Additionally, SHAP analysis revealed greater variance in feature values for human-generated texts, reflecting their diversity and lack of adherence to a single writing style. In contrast, model-generated texts displayed a more averaged and consistent writing pattern, shared across models.

Subsequent experiments, however, highlighted the limitations of our approach when applied to other domains. The low accuracies observed on the DAIGT-V4 dataset exposed the domain-specificity of our solution, which is tailored to novelistic texts. We found that a different feature set is more effective for the essay domain. This discrepancy likely arises from the differing backgrounds of the writers—professional novelists in one case versus lower-grade students in the other—or from the genres themselves. Novels, for instance, frequently include dialogue and punctuation like hyphens, alongside past-tense language, as reflected in the importance of the function word "was." These features are less prominent in essays, making our chosen features less applicable. Moreover, the dynamic nature of languages and its writing might also be the factor of domain definition. Authors' Writing Style dataset uses mostly classical literature from 20th century, which language might significantly differ from tweets or Reddit posts.

Prompt engineering within the datasets may also enforce specific writing styles, introducing new dimensionality to the domain itself. Additionally, the DAIGT-V4 dataset's use of various language models likely introduces further variability. Overall, these experiments demonstrated that our chosen features lack generalizability and are not transferable across domains.

In the final phase of our classification experiments, we explored the challenge of author classification. Here, we attempted to distinguish both human and model authors based on their writing styles. These experiments revealed that distinguishing authors—whether human or model—is a challenging task. While the solution performed satisfactorily in simplified binary classification scenarios, real-world applications involving numerous authors from diverse sources proved far more difficult. Our analysis did not uncover features that could effectively differentiate the writing styles of ten authors. Interestingly, we also found no evidence that models consistently adhered to specific requested writing styles. PCA plots, such as Figure 4.8, indicated that model-generated texts do not overlap with human texts and followed model's averaged writing tendencies.

In summary, our experiments confirmed key differences between human and LLM-generated texts, such as greater lexical diversity in human writing, but revealed the domain-specific limitations of our features. Author classification proved challenging, with LLMs exhibiting averaged styles rather than distinct emulation of human authors. These findings highlight the need for more adaptable features or different approach.

5.3 Information Content Analysis

Through the lens of Information Content (IC) analysis, we investigate the level of novelty exhibited by each collection. Our findings indicate that the books collection demonstrates significantly higher unpredictability across a wide range of linguistic attributes. This increased unpredictability highlights the uniqueness and diversity inherent in human-written texts. The most striking contrasts are observed in the use of punctuation characters, similar to findings of T. Kumurage and H. Liu [30] and feminine function words, such as ”she” and ”her.” These observations may partly reflect the nature of the selected novels, which could feature plots that include a significant number of female characters.

Conversely, the comparatively low occurrence of these words in the model-generated texts may suggest a bias in the training data or a reflection of the global distribution of written content, where masculine language—frequently encountered in domains such as scientific literature, or other formal writings—tends to dominate. This finding aligns with broader discussions on the limitations and biases inherent in large language models, particularly regarding their training corpora.

Despite these distinctions, certain attributes, such as Sequence IC and Local and Global Word Distribution IC, reveal that model-generated texts align closely with the books collection in these aspects. This alignment underscores the effectiveness of the underlying generative architectures in emulating human-like linguistic patterns, which opposes A. MuñozOrtiz et al. [14] statement of model's reduced vocabulary diversity. Similarly, other linguistic attributes, including Herdan's C score, exhibit comparable trends, suggesting that language models are well-trained to reproduce human-like text characteristics in some dimensions.

When we turn our attention specifically to the model-generated texts, a striking pattern emerges: these texts display a high degree of similarity to one another. This consistency confirms insights from earlier analyses and reinforces the notion that model-generated texts are governed by a probabilistic framework. The low variance observed across IC attributes for these texts indicates a lack of significant unpredictability, reflecting the averaged, generalized patterns typical of language models. Importantly, this suggests that while individual language models might incorporate subtle stylistic differences, their overall writing style converges toward a shared, predictable norm.

Overall, our analysis demonstrates that model-generated texts are both highly predictable and relatively homogeneous in style. In contrast, human-written texts exhibit far greater variability and uniqueness, characterized by the rarity and unpredictability of certain linguistic features. These findings highlight the novelty inherent in human authorship, which distinguishes it from the systematic, probabilistic outputs of machine-generated content.

5.4 Do large language models exhibit novelty in their writing styles?

The concept of novelty in text is inherently subjective, shaped by the reader's perception like in the works of Walsh [18] and GPT Poetry project [19]. However, throughout our research, we aimed to identify measurable linguistic attributes that could provide an objective framework for analyzing this phenomenon. To achieve this, we concentrated on the set of attributes defined in Section 4.3, including features introduced in Section 4.1.3.

Our analysis focused on the chunks extracted from the Authors' Writing Style dataset, where we tasked language models with generating texts that emulate the writing styles of various human authors. This inquiry had two primary objectives: first, to assess whether language models are capable of replicating specific human authors' stylistic nuances, and second, to determine the extent to which model-generated texts resemble one another. Using methods such as Principal Component Analysis, classification, and Information Content (IC) analysis, we observed that language models fail to effectively capture and reproduce the stylistic complexity of individual human authors. Furthermore, the texts generated by five different language models clustered closely together across multiple dimensions, with only minor deviations. This clustering suggests that the generated texts lack novelty and instead adhere to an averaged, generalized writing style.

Our findings also highlight the challenges inherent in statistically defining and analyzing human writing styles. Not only did language models struggle to emulate these styles, but our analytical approach encountered similar difficulties in isolating definitive stylistic markers. This observation underscores the inherent complexity and elusiveness of human writing style and novelty, which likely contribute to the limitations of language models in this domain.

In summary, human-authored texts exhibit significantly greater unpredictability, characterized by the uniqueness and rarity of their stylistic elements—qualities that are closely tied to the notion of novelty. By contrast, large language models produce outputs that conform to predictable patterns, lacking the distinctiveness and variability that define human writing. This reinforces the conclusion that while language models excel at generating coherent and plausible text, they fall short in replicating the nuanced creativity and novelty inherent in human-authored content.

5.5 Ethical Implications and Societal Impact

The findings of this research carry significant ethical and societal implications, particularly in the context of the increasing prevalence of AI-generated content. As demonstrated, large language models exhibit limited creativity and originality compared to human authors, and their stylistic patterns can often be predicted. This opens opportunities for robust detection algorithms and authorship attribution methods that can enhance trust and accountability in digital content.

Authorship attribution plays a vital role in addressing the challenges posed by AI-generated texts. By distinguishing human-authored content from machine-generated outputs, these methods empower fact-checking organizations, news outlets, and regulatory bodies to combat the proliferation of misinformation and ensure the integrity of published material.

Lastly, the cross-domain limitations observed in this study highlight challenges in applying detection methods across diverse contexts. While our approach shows promise within specific domains, its ability to generalize to others remains constrained. This finding emphasizes the need for continued refinement of detection methodologies to ensure their applicability in varied and high-stakes environments.

By addressing these ethical considerations, this research underscores the importance of advancing authorship attribution techniques and highlights the need for interdisciplinary collaboration to guide the responsible development and deployment of language models, balancing innovation with societal values.

6 Conclusions

Throughout our research, based on the Authors' Writing Style dataset, we demonstrate that our approach, which incorporates multiple methods and relies on features such as linguistic scores, punctuation characters, and function words, is sufficient to classify human-written texts and model-generated ones. These classifications highlight the distinct writing styles exhibited by the two groups. Moreover, our findings indicate that all five language models analyzed display similar linguistic patterns, leading to the clustering of their texts. Specifically, we observe that punctuation characters such as semicolons (;) and hyphens (-) occur significantly more often in human-written texts. This pattern may be influenced by the novel-focused domain of our dataset, while language models exhibit an averaged writing style that is less reflective of typical novel conventions. Furthermore, we identify that specific function words, such as ”she” and ”her,” are more prevalent in human texts, likely due to the dataset's inclusion of novels featuring female protagonists. In contrast, language models tend to favor masculine expressions, reflecting the dominance of masculine language in internet text sources.

We acknowledge, however, that our approach does not generalize well to other domains, such as essay datasets introduced via DAIGT-V4 in our classification experiments. Our analysis in these domains yielded low performance in distinguishing human-written from model-generated texts. This limitation appears to stem from the feature set's overfitting to the novel domain, where these features do not hold the same significance in other genres.

Additionally, we evaluated whether our approach could distinguish texts authored by multiple human authors. While related studies suggest that methods similar to ours perform well with a limited number of authors, our findings reveal challenges in correctly assigning texts to the appropriate human author in cases involving a larger number of authors. Similarly, for model-generated texts, even when we prompted models to emulate specific authors' writing styles, their outputs retained the generic, averaged writing style characteristic of language models. This suggests that the task of producing text reflective of a specific author's linguistic features remains ambiguous for language models, possibly due to misdefinitions of ”style” in our framework and language model architectures itself, or limitations in prompt engineering that failed to enforce stylistic specificity.

Lastly, we explored the novelty of model-generated texts, an area that remains under-researched. Defining novelty as the presence of rare and unpredictable linguistic attributes, we employed Information Content Theory to evaluate this dimension. Our analysis reveals that model-generated texts lack novelty compared to human-authored texts. Human authors consistently demonstrate higher originality in their writing, whereas language models produce texts characterized by highly probable and predictable patterns. These findings underscore that while language models excel in human-machine textual interaction and effectively convey information to some extent, they fall short in producing original and unique content—a hallmark of human writing.

This research emphasizes both the potential and limitations of language models in stylistic and creative contexts, providing a foundation for future studies aimed at enhancing the authenticity and diversity of model-generated texts.

7 Future Work

In this thesis, we presented and concluded our research; however, some ideas could not be pursued due to limited resources or arose during the development process but were set aside due to time constraints. One such idea involves extending the Authors' Writing Style dataset by incorporating text generated from prompts that do not explicitly request a specific author's style. This approach would use identical prompts while allowing the language model (LLM) to generate content without stylistic constraints. By analyzing outputs from these two types of prompts, we could investigate whether explicitly requesting a writing style influences the output and if this adjustment aligns the generated text closer to the original author's style. Such an analysis would help us determine whether LLMs, despite not perfectly replicating a style, exhibit an initial understanding of stylistic features.

Another avenue for future work involves refining the feature set used in our analysis. Some features, such as the average number of syllables per word and the Flesch Reading Ease score, are highly correlated since the latter depends on metrics like syllables and word counts. Identifying and addressing such correlations would allow us to reduce the feature set to its most essential components, improving efficiency and interpretability. Additionally, incorporating features from a broader range of domains could result in a more generalized cross-domain feature set applicable to diverse text types. Finally, exploring new features, such as part-of-speech (POS) tags commonly used in related works, could further enhance our analysis.

The size of text chunks used in our analysis presents another potential area for improvement. Our research used 5,000-token chunks; however, evaluating the approach's robustness with smaller chunks, such as the size of a tweet, would provide valuable insights. This investigation could reveal whether chunk size should be treated as a hyperparameter significantly influencing the overall performance of the architecture.

Lastly, we recommend applying cross-validation in certain classification tasks to ensure more robust results. For example, in the classification task involving two authors from the same collection, the limited number of chunks may result in biased outcomes. Cross-validation would offer a more rigorous evaluation framework, yielding fairer and more reliable results.

By addressing these areas, future research can build upon our findings to develop a more comprehensive understanding of writing style analysis and further refine the methodologies employed.

Bibliography

[1] Ashish Vaswani et al. *Attention Is All You Need*. 2023. arXiv: 1706.03762 [cs.CL]. URL: https://arxiv.org/abs/1706.03762.

[2] Shen Wang et al. *Large Language Models for Education: A Survey and Outlook*. 2024. arXiv: 2403.18105 [cs.CL]. URL: https://arxiv.org/abs/2403.18105.

[3] Jinqi Lai et al. “Large language models in law: A survey”. In: *AI Open* 5 (2024), pp. 181–196. ISSN: 2666-6510. DOI: https://doi.org/10.1016/j.aiopen.2024.09.002. URL: https://www.sciencedirect.com/science/article/pii/S2666651024000172.

[4] Sida Peng et al. *The Impact of AI on Developer Productivity: Evidence from GitHub Copilot*. 2023. arXiv: 2302.06590 [cs.SE]. URL: https://arxiv.org/abs/2302.06590.

[5] Yanshen Sun et al. *Exploring the Deceptive Power of LLM-Generated Fake News: A Study of Real-World Detection Challenges*. 2024. arXiv: 2403.18249 [cs.CL]. URL: https://arxiv.org/abs/2403.18249.

[6] Edoardo Mosca et al. “Distinguishing Fact from Fiction: A Benchmark Dataset for Identifying Machine-Generated Scientific Papers in the LLM Era.” In: *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*. Ed. by Anaelia Ovalle et al. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 190–207. DOI: 10.18653/v1/2023.trustnlp-1.17. URL: https://aclanthology.org/2023.trustnlp-1.17.

[7] Zhixin Lai, Xuesheng Zhang, and Suiyao Chen. *Adaptive Ensembles of Fine-Tuned Transformers for LLM-Generated Text Detection*. 2024. arXiv: 2403.13335 [cs.LG]. URL: https://arxiv.org/abs/2403.13335.

[8] Seyedeh Fatemeh Ebrahimi et al. *Sharif-MGTD at SemEval-2024 Task 8: A Transformer-Based Approach to Detect Machine Generated Text*. 2024. arXiv: 2407.11774 [cs.CL]. URL: https://arxiv.org/abs/2407.11774.

[9] Teodor-George Marchitan, Claudiu Creanga, and Liviu P. Dinu. *Transformer and Hybrid Deep Learning Based Models for Machine-Generated Text Detection*. 2024. arXiv: 2405.17964 [cs.CL]. URL: https://arxiv.org/abs/2405.17964.

[10] Sebastian Gehrmann, Hendrik Strobelt, and Alexander M. Rush. *GLTR: Statistical Detection and Visualization of Generated Text*. 2019. arXiv: 1906.04043 [cs.CL]. URL: https://arxiv.org/abs/1906.04043.

[11] Eric Mitchell et al. “DetectGPT: Zero-Shot Machine-Generated Text Detection using Probability Curvature”. In: *Proceedings of the 40th International Conference on Machine Learning*. Ed. by Andreas Krause et al. Vol. 202. Proceedings of Machine Learning Research. PMLR, July 2023, pp. 24950–24962. URL: https://proceedings.mlr.press/v202/mitchell23a.html.

[12] Scott Lundberg and Su-In Lee. *A Unified Approach to Interpreting Model Predictions*. 2017. arXiv: 1705.07874 [cs.AI]. URL: https://arxiv.org/abs/1705.07874.

[13] Aditya Shah et al. “Detecting and Unmasking AI-Generated Texts through Explainable Artificial Intelligence using Stylistic Features”. In: *International Journal of Advanced Computer Science and Applications* 14.10 (2023). DOI: 10.14569/IJACSA.2023.01410110. URL: http://dx.doi.org/10.14569/IJACSA.2023.01410110.

[14] Alberto Muñoz-Ortiz, Carlos Gómez-Rodríguez, and David Vilares. “Contrasting Linguistic Patterns in Human and LLM-Generated News Text”. In: *Artificial Intelligence Review* 57.10 (2024), p. 265. ISSN: 1573-7462. DOI: 10.1007/s10462-024-10903-2. URL: https://doi.org/10.1007/s10462-024-10903-2.

[15] Frederick Mosteller and David L. Wallace. “Inference in an Authorship Problem”. In: *Journal of the American Statistical Association* 58.302 (1963), pp. 275–309. ISSN: 01621459, 1537274X. URL: http://www.jstor.org/stable/2283270 (visited on 11/28/2024).

[16] José Nilo G. Binongo. “Who Wrote the 15th Book of Oz? An Application of Multivariate Analysis to Authorship Attribution”. In: *CHANCE* 16.2 (2003), pp. 9–17. DOI: 10.1080/09332480.2003.10554843. eprint: https://doi.org/10.1080/09332480.2003.10554843. URL: https://doi.org/10.1080/09332480.2003.10554843.

[17] Avanti Bhandarkar et al. “Emulating Author Style: A Feasibility Study of Prompt-enabled Text Stylization with Off-the-Shelf LLMs”. In: *Proceedings of the 1st Workshop on Personalization of Generative AI Systems (PERSONALIZE 2024)*. Ed. by Ameet Deshpande et al. St. Julians, Malta: Association for Computational Linguistics, Mar. 2024, pp. 76–82. URL: https://aclanthology.org/2024.personalize-1.6.

[18] Melanie Walsh, Anna Preus, and Elizabeth Gronski. *Does ChatGPT Have a Poetic Style?* 2024. arXiv: 2410.15299 [cs.CL]. URL: https://arxiv.org/abs/2410.15299.

[19] Ernest Davis. *ChatGPT’s Poetry is Incompetent and Banal: A Discussion of (Porter and Machery, 2024)*. Nov. 2024. URL: https://cs.nyu.edu/~davise/papers/GPT-Poetry.pdf.

[20] Jad Doughman et al. *Exploring the Limitations of Detecting Machine-Generated Text*. 2024. arXiv: 2406.11073 [cs.CL]. URL: https://arxiv.org/abs/2406.11073.

[21] Konrad Banachewicz. *Atticus Open Contract Dataset*. 2023. URL: https://www.kaggle.com/datasets/konradb/atticus-open-contract-dataset-aok-beta.

[22] Charan Gowda et al. *Twitter and Reddit Sentimental analysis Dataset*. 2019. DOI: 10.34740/KAGGLE/DS/429085. URL: https://www.kaggle.com/ds/429085.

[23] Kumar Saksham. *Global News Dataset*. 2023. DOI: 10.34740/KAGGLE/DSV/7105651. URL: https://www.kaggle.com/dsv/7105651.

[24] Darek Kłeczek. *DAIGT-v4 Train Dataset*. 2022. URL: https://www.kaggle.com/datasets/thedrcat/daigt-v4-train-dataset.

[25] Alec Radford et al. “Improving language understanding by generative pre-training”. In: (2018).

[26] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL]. URL: https://arxiv.org/abs/1810.04805.

[27] Tyna Eloundou et al. *GPTs are GPTs: An Early Look at the Labor Market Impact Potential of Large Language Models*. 2023. arXiv: 2303.10130 [econ.GN]. URL: https://arxiv.org/abs/2303.10130.

[28] Karan Singhal et al. *Large Language Models Encode Clinical Knowledge*. 2022. arXiv: 2212.13138 [cs.CL]. URL: https://arxiv.org/abs/2212.13138.

[29] Lei Huang et al. “A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions”. In: *ACM Trans. Inf. Syst.* (Nov. 2024). Just Accepted. ISSN: 1046-8188. DOI: 10.1145/3703155. URL: https://doi.org/10.1145/3703155.

[30] Tharindu Kumarage and Huan Liu. “Neural Authorship Attribution: Stylometric Analysis on Large Language Models”. In: *2023 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*. 2023, pp. 51–54. DOI: 10.1109/CyberC58899.2023.00019.

[31] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. “”Why Should I Trust You?”: Explaining the Predictions of Any Classifier”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’16. San Francisco, California, USA: Association for Computing Machinery,

2016, pp. 1135–1144. ISBN: 9781450342322. DOI: 10.1145/2939672.2939778. URL: https://doi.org/10.1145/2939672.2939778.

[32] Shibamouli Lahiri. “Complexity of Word Collocation Networks: A Preliminary Structural Analysis”. In: *Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics*. Gothenburg, Sweden: Association for Computational Linguistics, Apr. 2014, pp. 96–105. URL: http://www.aclweb.org/anthology/E14-3011.

[33] OpenAI. *OpenAI API Reference*. https://platform.openai.com/docs/api-reference/chat. June 2024.

[34] Google DeepMind. *Gemini API Reference*. https://cloud.google.com/vertex-ai/generative-ai/docs/model-reference/gemini. June 2024.

[35] Mistral AI. *Mistral AI API Reference*. https://docs.mistral.ai/api. June 2024.

[36] Anthropic. *Anthropic API Reference*. https://docs.anthropic.com/en/api/messages. June 2024.

[37] Kaggle. *LLM-Detect: AI-Generated Text Detection*. https://www.kaggle.com/competitions/llm-detect-ai-generated-text. Kaggle Competition. 2024. URL: https://www.kaggle.com/competitions/llm-detect-ai-generated-text.

[38] James V Stone. *Information Theory: A Tutorial Introduction*. 2019. arXiv: 1802.05968 [cs.IT]. URL: https://arxiv.org/abs/1802.05968.

[39] James P. Bagrow, Xipei Liu, and Lewis Mitchell. “Information flow reveals prediction limits in online social activity”. In: *Nature Human Behaviour* 3.2 (Jan. 2019), pp. 122–128. ISSN: 2397-3374. DOI: 10.1038/s41562-018-0510-5. URL: http://dx.doi.org/10.1038/s41562-018-0510-5.

Technical
University of
Denmark

Richard Petersens Plads, Build. 324
2800 Kgs. Lyngby
Tlf. 4525 1700

www.compute.dtu.dk