computers

MDPI

Article

Stylometry Analysis of Human and Machine Text for Academic Integrity

Hezam Albaqami [1,*], Muhammad Asif Ayub [2], Nasir Ahmad [3], Yaseen Ahmad [2], Mohammad M. Alqahtani [1], Abdullah M. Algamdi [1], Almoaid A. Owaidah [4] and Kashif Ahmad [3,*]

1 Department of Computer Science and Artificial Intelligence, College of Computer Science and Engineering, University of Jeddah, Jeddah 21493, Saudi Arabia; mmalqahtani@uj.edu.sa (M.M.A.); amalgamdi6@uj.edu.sa (A.M.A.)
2 Department of Computer Systems Engineering, University of Engineering and Technology, Peshawar 25120, Pakistan; asifayub836@gmail.com (M.A.A.); ahmadyaseen1628@gmail.com (Y.A.)
3 Department of Computer Science, Munster Technological University, T12 P928 Cork, Ireland; nasir.ahmad@mtu.ie
4 Department of Management Information Systems, Faculty of Economics and Administration, King Abdulaziz University, P.O. Box 80200, Jeddah 21589, Saudi Arabia; aowaidah@kau.edu.sa
* Correspondence: haalbaqamii@uj.edu.sa (H.A.); kashif.ahmad@mtu.ie (K.A.)

Abstract

This work addresses critical challenges to academic integrity, including plagiarism, fabrication, and verification of authorship of educational content, by proposing a Natural Language Processing (NLP)-based framework for authenticating students' content through author attribution and style change detection. Despite some initial efforts, several aspects of the topic are yet to be explored. In contrast to existing solutions, the paper provides a comprehensive analysis of the topic by targeting four relevant tasks, including (i) classification of human and machine text, (ii) differentiating in single and multi-authored documents, (iii) author change detection within multi-authored documents, and (iv) author recognition in collaboratively produced documents. The solutions proposed for the tasks are evaluated on two datasets generated with Gemini using two different prompts, including a normal and a strict set of instructions. During experiments, some performance reduction is observed for the proposed solutions on the dataset generated by the strict prompt, demonstrating the complexities involved in detecting machine-generated text with cleverly crafted prompts. The generated datasets, code, and other relevant materials are made publicly available on GitHub, which are expected to provide a baseline for future research in the domain.

Keywords: style change detection; generative AI; text classification; AI text classification; plagiarism detection; NLP; academic integrity

Check for updates

Academic Editor: Ming Liu

Received: 5 February 2026
Revised: 20 March 2026
Accepted: 25 March 2026
Published: 1 April 2026

Correction Statement: This article has been republished with a minor change. The change does not affect the scientific content of the article and further details are available within the backmatter of the website version of this article.

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

In recent years, there has been enormous interest in Generative Artificial Intelligence (AI) and its applications. As a result, several interesting AI-based content/text-generating tools, such as ChatGPT, have been introduced that can accurately generate grammatically correct text on demand for various purposes [1]. For instance, articles, emails, social media posts, essays, and even code in different programming languages could be generated on the fly in response to user questions. These auto-text generating tools are useful in multiple domains, particularly in the education sector, where they provide diverse knowledge sources with reduced bias compared to conventional learning approaches. However, if

used unethically, these tools could significantly undermine educational learning outcomes and objectives. For instance, students may rely on them to complete assignments without fully engaging with or understanding the underlying questions. The plagiarism and unethical use of these tools in assignments, tests, theses, and project reports have been widely discussed in academia, and great concerns over academic integrity have been reported worldwide [2]. Students' overreliance on such tools may significantly affect their critical thinking, problem-solving, communication (i.e., interaction with teachers and peers), and collaboration capabilities. Similarly, these tools have also raised concerns about the reliability and authenticity of content across various application domains, such as journalism [3], art [4], and law enforcement. For instance, such tools have significantly increased the creation and propagation of false information and fake news. These fake news and misinformation could affect the policies [5], trust in governments, and cause chaos [6], which may compromise public safety.

The encouraging point is that there are already some AI and Natural Language Processing (NLP)-based techniques that could automatically detect content (text) generated through such tools; however, such algorithms/tools can be easily bypassed by slight changes to auto-generated text (i.e., giving a human touch to the text). Interestingly, there are also AI-based tools for humanizing machine-generated text [7], making machine-generated text detection even more challenging. Moreover, academia is also facing another important challenge in terms of authorship detection, i.e., verifying that assignments, essays, project reports, and theses are written by students and are not produced by another human (e.g., peers or external experts). This challenge is equally important and harmful to academic integrity and learning outcomes. Given the buzz around AI-based content-generating tools in academia and the consequences of unethical and fraudulent practices in the education sector, a swift response from the research community is essential.

The literature already reports some initial efforts to solve these issues. However, most existing solutions rely on training ML algorithms to differentiate between machine- and human-generated text, which can be easily bypassed by humanizing machine-generated text. Moreover, the domain also lacks a large-scale benchmark dataset. To overcome these limitations, we propose a stylometry-based solution and produce a large-scale benchmark dataset. The styloetry-based solution helps tackle multiple challenges faced by academia in ensuring academic integrity; for instance, it can verify whether a document was produced by the same author or whether other authors (human or machine) contributed to it. In this work, we tackle four tasks, including (i) differentiating between machine- and human-generated text, (ii) classification of single- and multi-authored documents, (iii) identifying the parts (paragraphs) of multi-authored documents where the author changes, and (iv) identification of the author for a portion of text in a multi-authored document. These four tasks are very critical in the educational context, and with these tasks, we try to answer four research questions. Firstly, by differentiating between machine- and human-generated text, could we detect and ultimately avoid AI-based solutions in assessments? Moreover, with two different types of prompts, we want to explore whether the performance of AI-based machine detection tools is affected by the quality of the prompt. Secondly, for task 2, we want to explore whether potential collaboration (illegal or legal) cases in the production of educational content could be detected automatically. Finally, in the last two tasks, we explore the potential of AI-based tools for identifying parts produced by another author. Style analysis mainly involves the identification of distinct patterns in writing style by utilizing intrinsic features in a single-authored or collaboratively authored document. Style change detection (SCD) is one of the key applications of style analysis that could be utilized for intrinsic plagiarism detection (IPD) [8]. IPD differs from external plagiarism detection, where a document is checked against existing sources or references. On the other hand,

IPD [9] aims to detect plagiarism by examining only the input document, determining whether parts of it are not from the same author, which could be a human or an AI-based tool. IPD leverages the writing style consistency of the authors throughout the document for plagiarism detection, independent of any other reference corpus, which makes it a more effective solution for tackling these challenges. Style change analysis is not new; however, it has been mainly explored in the context of human authors only. We believe such a detailed analysis will provide a baseline for future work in the domain.

The key contributions of this work can be summarised as follows:

- The paper proposes a style change analysis-based NLP framework for tackling some of the key challenges faced by academia in the era of generative AI. These tasks include differentiating between single- and multi-authored documents, classifying machine- and human-generated text, identifying the paragraphs where author changes occur in multi-authored documents, and author recognition in multi-authored documents.
- The paper also analyzes the impact of cleverly crafted prompts on the performance of state-of-the-art NLP models-based tools for plagiarism and document authentication
- Two different sets of machine text are generated and embedded in a human-produced dataset, providing a valuable source for further research in the domain. The dataset and source code are publicly available.
- The paper highlights key challenges associated with the topic and hints at some potential solutions.

The rest of the paper is organized as follows. Section 2 provides an overview of the literature. Section 3 describes the proposed methodology. Section 4 discusses the experimental setup, conducted experiments, and the experimental results. Section 5 concludes the paper.

2. Related Work

Academic integrity has always remained one of the key topics of research and policy-making in education. Although there is no single definition of the term 'academic integrity', the literature defines it in various ways, mostly focusing on its key principles. The key principles of academic integrity include academic honesty [10], trust [11], fairness, responsibility, and respect [12,13]. The literature also guides the maintenance of academic integrity by ensuring adherence to these key principles, especially in the advent of new educational technology or policy. More recently, the emergence of Generative AI and its potential use in education has also raised worldwide concerns and debates about academic integrity [14]. The literature widely explores how the use of generative AI in education can compromise several key principles of academic integrity, such as honesty, trust, and responsibility, by producing and presenting AI content as one's own work or presenting it without verification.

One of the key challenges faced in maintaining academic integrity in the era of generative AI is the detection of AI content in academia [15]. The literature already reports several interesting solutions for efficient detection of AI content in education assessments, exploring different aspects of the task [16]. A vast majority of the literature on the topic explores the efficacy of AI content-detection tools in the educational context, such as Turnitin, GPTZero, and Copyleaks [17,18].

The literature already explores several key aspects and challenges of machine and human content classification across different application domains. In this work, we explore a different aspect of the task known as stylometry analysis [19]. The literature already reports some initial efforts in the domain. For instance, Zamir et al. [20] hinted at the potential of stylometry analysis, particularly Style Change Detection (SCD), for differentiating machine- and human-generated text. The literature already reports several interesting studies on SCD

and analysis, exploring different aspects and applications of the concept. One of the key applications of style change analysis is intrinsic plagiarism detection (IPD) [8]. In contrast to external plagiarism, where a document is checked against existing sources/references, IPD aims at discovering plagiarism by examining only the input document. Subsequently, decides whether parts of the input document are not from the same author, which could be a human or an AI-based tool. IPD leverages the writing style consistency of the authors over the length of the document for plagiarism detection, independent of any other reference corpus [21]. SCD could also be used for document authentication [20]. For instance, Zamir et al. [22] used SCD for the identification of multi-authored documents as well as the text segments where author changes occurred Several other works also used stylometry analysis for the classification of single- and collaboratively generated documents [23,24]. SCD has also been part of a benchmarking competition, PAN [25,26], and has attracted significant community interest. The literature already reports some interesting work on stylometry; however, several aspects of stylometry in general and SCD analysis in particular remain to be explored. The majority of the existing literature is based solely on human-authored documents. There are very few recent works that use style change analysis to differentiate between human- and machine-authored documents on general topics. For instance, Kumar et al. [27] employed stylometry analysis to detect machine-generated text in Twitter posts, relying on traditional stylometry features such as phraseology, punctuation, and linguistic diversity. Opara et al. [28] also relied on traditional stylometry features for the classification of human- and machine-generated text. More recently, several works have explored the feasibility of transformer-based models for stylometry analysis, providing contextual embeddings for words, sentences, or entire documents. These models have been proven to be very effective in different applications of stylometry analysis, such as author attribution [29], authorship detection [30], and SCD [22].

The existing literature has mostly explored the first task (i.e., classification of human- and machine-generated text) using machine-generated text only. The other three tasks have been explored only in the context of human authors. Moreover, the literature lacks a benchmark dataset containing human- and machine-generated text, which would enable a detailed analysis of SCD by providing annotations for various relevant tasks. Therefore, in this work, we analyze four different tasks of SCD analysis by generating corresponding machine-text datasets under two different prompts.

3. Methodology

All four challenges explored in this work are handled through text classification. The first three challenges are addressed through binary classification, while the final challenge is treated as a multi-label text classification task. We started with the creation of a dataset by generating machine-generated text through intelligent prompting, which was then embedded in a benchmark dataset composed of human-authored documents. After developing the dataset, several preprocessing techniques were applied to address various issues with the data, preparing it for text classification and analysis using state-of-the-art NLP algorithms. In the final phase, we employ various NLP algorithms for text classification. All the tasks involve these three steps, with slight changes in the classification task for the multi-label classification needed for task 4. In the next subsections, we provide a detailed overview of each of the steps.

3.1. Dataset Creation

In this phase of the methodology, we generated a large collection of machine-generated text paragraphs using the Gemini model (gemini-1.5-flash-latest) under two different prompts. The newly AI-generated text paragraphs were then inserted into the existing

human-generated dataset at random locations for training and evaluation of our models for the four tasks. Figure 1 presents the flowchart of the dataset generation process. As shown in the figure, the text was generated using multiple free API keys.

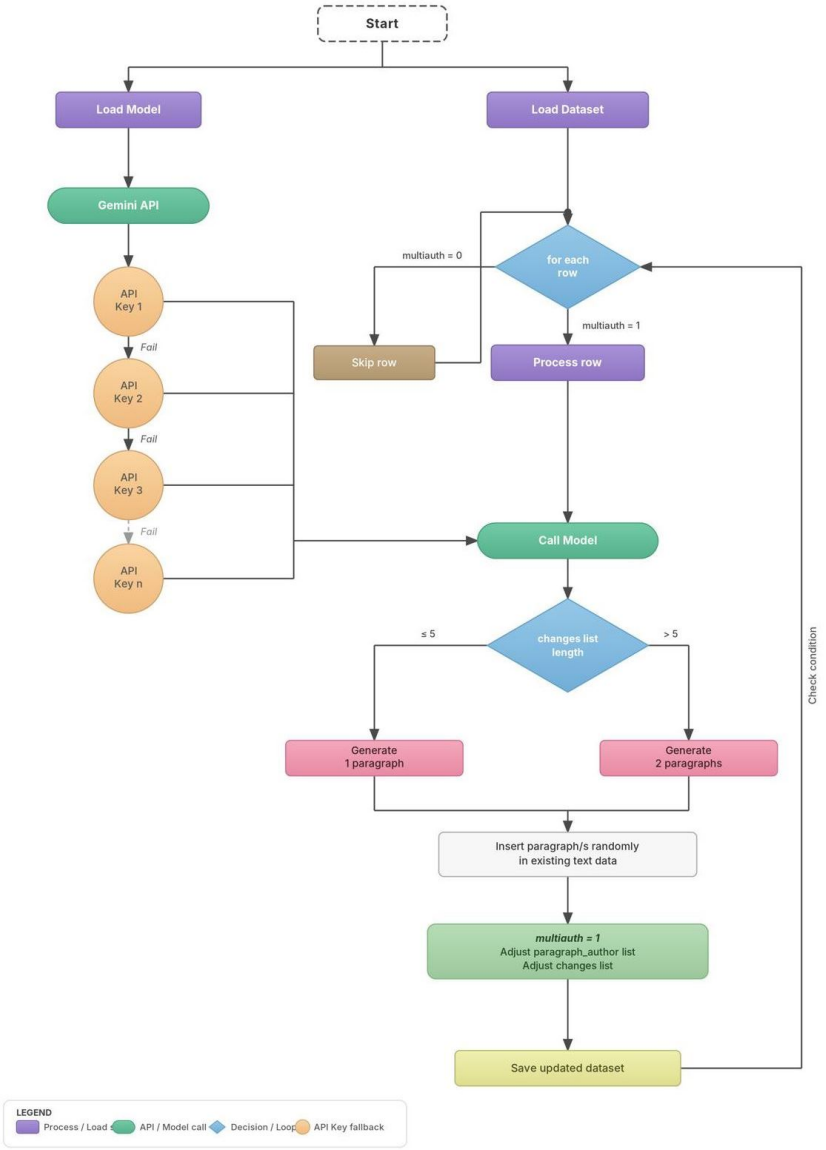

Figure 1. Flowchart of the data generation process. The same flowchart is used for both datasets, with only differences in the instruction sets.

The data generation process was designed to match the structure of the existing dataset, which contains text from human authors [23]. In the human text dataset, text samples are organized into single and multi-authored files. Moreover, the dataset contains text samples of four authors. In our case, we treat multi-authored files and generate machine-generated text, treating AI/LLMs as the fifth author. Thus, as shown in the figure, for each sample, we first determine whether it is a single-authored or multi-authored text, and proceed with the rest of the process if it is multi-authored. This approach helped avoid repetition and the labor involved in the text generation process. We note that skipping the single-authored files does not affect the machine vs human text classification task, as the machine-generated text, which is considered as the fifth author, is combined for that experiment.

In the multi-authored documents, the number of authors and the number of paragraphs varied, with some containing fewer and others more. Keeping this in view and maintaining a reasonable ratio between the machine-generated and human-generated text,

we add either a single paragraph or multiple paragraphs, depending on the number of paragraphs in each document. Thus, for each multi-authored document, we check whether the number of paragraphs (changes list-end of a paragraph in a document) is less than or equal to 5, or greater than 5; if so, we insert a single paragraph and a couple of machine-generated paragraphs, respectively. After successfully generating and inserting machine-generated text at random, the author's change list is updated accordingly, serving as ground truth for some tasks.

For the machine text, we generated two different prompts: A normal one and a stricter one. In the normal prompt, we targeted a broader range of topics covered in the human-text benchmark dataset. The topics were extracted through modeling techniques, resulting in a list of topics extracted from the human text. For topic modeling, we used three different topic modeling algorithms, namely (i) BERTopic [31], Latent Dirichlet allocation (LDA) [32], and Non-negative Matrix Factorization (NMF) [33]. These algorithms produced a list of topics extracted from the human text. We selected the top ten topics extracted by each algorithm, which were then fed into the Gemini model to generate extended topics by adding domains and subdomains of the topics. The extended topics were then fed into the Gemini model as part of the prompt for machine-generated text. Table 1 provides the list of topics used in the prompt for text generation. In the **strict** prompt, we did not use predefined topics; rather, the model provided a human sample and asked the model to generate text in the same context. A comparison of the instructions used in both prompts is provided in Table 2.

Table 1. A list of the topics used in the prompt for text generation.

File Systems & Data Management	Programming & Software Development
Networking & Cybersecurity	Cloud & Virtualization Technologies
AI, ML & Data Science	Hardware & System Performance
Software Installation & Troubleshooting	Enterprise IT & DevOps
Tech Industry Trends	User Experience & HCI

Table 2. A comparison of the set of instructions in both prompts used in this work for the generation of machine text with the model.

Instructions Used in the Normal Prompt	Instructions Used in the Strict Prompt
• Generate a single paragraph, focusing on any of the following topics within the domains: File Systems & Data Management, Programming & Software Development, Networking & Cybersecurity, Cloud & Virtualization Technologies, AI, ML & Data Science, Hardware & System Performance, Software Installation & Troubleshooting, Enterprise IT & DevOps, Tech Industry Trends, User Experience & HC • Do not include any introductory text, explanations, prefaces, or meta information such as 'Here is a new paragraph'. • Ensure variety across the five generated paragraphs. • Each paragraph should be between 40 and 80 words. • Cover features, principles, challenges, adoption trends, and pros/cons. • Use an informative and professional tone with factual correctness. • Avoid redundancy or vague speculation.	• Based on the context of the text in the single quotes, generate a new paragraph in which several words will equal the average words in the other provided paragraphs • Only output the paragraph; do not include any introductory text, explanations, prefaces, or meta information such as 'Here is a new paragraph'. • Avoid extra labels, annotations, or machine-generated markers in the output. Do not summarize or directly use ideas from the provided paragraphs. • Ensure the paragraph is relevant to the topic and adds a new perspective or insight. • Do not include introductory or explanatory text—produce only the paragraphs directly. • Focus on contributing meaningful content that complements the general agenda or subject matter. • Pay attention to the length of your generated paragraph, ensuring it matches the average word count of the other paragraphs.

3.2. Data Pre-Processing

The data was analyzed and cleaned before being fed into the models. Various tasks were carried out during data pre-processing, including data cleaning, duplicate detection and removal, and data balancing. Firstly, the dataset was checked for null values and duplicate entries by analyzing the text and authors' information. This was followed by data balancing using class weights. We note that during this phase, we explored various data balancing techniques, including oversampling, undersampling, SMOTE [34], SMOTEENN [35], ADASYN [36], NearMiss [37], Tomek, and NearMiss-Tomek. However, the final approach used class weights, assigning higher weights to the minority class to give it more attention. These weights were incorporated into the cross-entropy loss function so that minority-class errors carry a greater penalty.

3.3. Text Classification

For text classification, we utilized four state-of-the-art models: BERT (base model), ALBERT, DistilBERT, and RoBERTa. Our choice of these models is based on their proven performance on similar tasks [38]. The lower/intermediate layers of these models extract more traditional stylometry features, such as function words, punctuation, and sentence rhythm, etc. The BERT-base model consists of 12 transformer layers and attention heads, with a token length of 512. DistilBERT is a variant of the base model, providing a smaller and faster model without significantly impacting performance. RoBERTa is another variant of the base model with a different training strategy, including a larger training set and time, as well as the removal of next-sentence prediction. All the models are fine-tuned on the new dataset using different dropout and attention dropout rates to achieve better results. More details of the hyperparameters are provided in Section 4.

4. Experiments and Results

4.1. Experimental Setup

To achieve the multifaceted objectives of this work, we conducted multiple experiments to explore different aspects of the application. These experiments include:

- **Classification of machine and human-generated text**: This is a binary classification task where the models predict whether a document was produced by a human author or a machine.
- **Classification of single and multi-authored documents**: This is also a binary classification task where the models predict whether a document was produced by a single author or is multi-authored. The single-authored documents include text produced by humans and machines. In the case of multi-authored documents, machine-generated text was randomly embedded in the documents produced by multiple human authors.
- **Author change detection**: This is also a binary classification task that involves the identification of subsequent paragraphs where the author changes. Each pair of subsequent paragraphs in multi-authored documents is labeled either 0 or 1, representing "no changes" and "author changes", respectively.
- **Author recognition**: In this experiment, we need to identify the author for each document that could be either a single-authored or collaboratively written document. Our goal is to predict whether a particular author contributed to each document or not. Thus, this task is treated as a multi-label classification task with five different authors, including the machine/AI as one of the authors.

During the experiments, we used various techniques and strategies. For instance, considering our limited computational resources, we used the Gradient accumulation technique, allowing us to use a larger batch size. The batch size used in the experiments was 32;

however, gradient accumulation of two steps effectively increased it to 64. Table 3 summarizes the hyperparameter settings of the experiments. The learning rate was initialized at 0.00001 and decayed via a cosine scheduler with 200 warmup steps. Similarly, weight decay was used during training, preventing overfitting and improving the generalization of models. Moreover, we run the validation every 500 steps and save the best model checkpoint based on the weighted F1-score. We also enabled mixed-precision (fp16) training to speed up computation and reduce memory usage. Logging was employed, recording metrics every 50 steps and retaining only the five most recent checkpoints, to monitor the progress. Moreover, we used 70%, 15%, and 15% samples for training, validation, and testing. The total number of samples per class for each task is provided in Table 4.

Table 3. A summary of the hyperparameters used during the experiments.

Hyper-Parameter	Value	Hyper-Parameter	Value
Batch Size	32	Gradient Accumulation	2
Learning Rate	0.00001	Weight Decay	0.01
Epochs	5	Logging	50

Table 4. Statistics of the dataset in terms of samples per class used in each task.

Task	Number of Samples			
	Label	Training	Validation	Test
Machine vs. Human Text Classification	0	77,061	16,438	16,344
	1	12,952	2751	2748
Single- vs. Multi-authored Classification	0	2800	600	600
	1	8379	1797	1794
Author Change Detection	0	27,725	5934	5814
	1	51,138	10,887	10,886
Author Recognition	1	41,672	8985	8825
	2	20,605	4362	4354
	3	10,522	2235	2238
	4	4291	885	929
	5	12,952	2751	2748

4.2. Experimental Results

4.2.1. Machine and Human-Text Classification

Table 5 presents the experimental results for the first task, with text generated using both the normal and strict prompts, in terms of accuracy, precision, recall, and F1-score. Overall, the results are very good across both datasets, demonstrating the capabilities of transformer-based NLP models to differentiate between human- and machine-generated text. As expected, the results are significantly better on the dataset, including AI-generated text through the normal prompt, compared to the strict one. This demonstrates the complexity of differentiating human text from AI text when generated through cleverly crafted prompts. Regarding the performance of the individual models, no significant differences are observed in their results (all metrics) across both datasets. Moreover, precision and recall are within the same range, demonstrating a balanced trade-off between correctly predicting positive and negative samples. We also analyzed the performance of these models on individual classes (i.e., the classification of AI and human text). We observed a larger difference in the models' performance across the individual classes with the strict prompt than with the normal one, with scores higher for human text than for AI-generated text.

Table 5. Experimental results on the classification of machine and human text on both datasets.

Model	Normal Prompt				Strict Prompt			
	Accuracy	**Precision**	**Recall**	**F1-Score**	**Accuracy**	**Precision**	**Recall**	**F1-Score**
distilbert-base-uncased	0.9998	0.9998	0.9998	0.9998	0.9477	0.9463	0.9477	0.9463
Albert-base-v2	0.9995	0.9995	0.9995	0.9995	0.9456	0.9440	0.9456	0.9441
Roberta-base	0.9997	0.9997	0.9997	0.9997	0.9487	0.9473	0.9487	0.9474
Bert-base-uncased	0.9998	0.9998	0.9998	0.9998	0.9483	0.9469	0.9483	0.9467

To further investigate differences in model performance across both datasets, a comparative semantic analysis is conducted using Sentence-BERT embeddings to assess inter-class separation and intra-class cohesion. Such analysis, using inter-class separation and intra-class cohesion, is crucial for any classification task, particularly in the context of a machine-generated dataset, as it highlights how well each dataset preserves semantic organization. The average cosine similarity between class 0 (human text) and class 1 (machine text) of the dataset generated with the normal prompt embeddings is very low (0.047). In the embedding space, these two classes formed completely separate clusters. A similar trend is observed in the case of the dataset generated with the strict prompt, with the same inter-class similarity shown in Figure 2. However, a different trend is observed for intra-class cohesion, another critical factor affecting classification performance on a dataset. In terms of intra-class cohesion, we observed that in the dataset generated with the strict prompt, class 1 examples were moderately similar to each other, with an average intra-class cohesion of 0.14, while class 0 samples were quite diverse, with an average intra-class cohesion of 0.03. We note that class 0 is human-generated text and remained the same across both datasets. Thus, we computed intra-class cohesion only for the machine-generated class in dataset 2, which was generated using the normal prompt. The average intra-class cohesion for class 1 in dataset 2 is comparatively high (0.29), indicating greater compactness/similarity among the machine-generated samples. The low intra-class cohesion of class 1 in dataset 1 is mainly due to the prompt instructions, which require strictly similar text to be produced in the same context as the human-produced text in each document.

The high intra-class cohesion of the machine-generated data led to a better classification score for machine-generated text than for text generated with the strict prompt. This is also evident from the classification scores for the individual classes in both datasets. As a sample, we provide classification results of the Albert model in Table 6.

Considering the high scores and to better analyze the performance of the models, we conducted bootstrapping experiments for the top-performing model (Distibert) on both datasets generated through strict and normal prompts. The confidence intervals for evaluation metrics (F1-score) were estimated via stratified bootstrap resampling. For each of 5000 iterations, we sampled, with replacement, examples from each class to match the class proportions of the test set, recomputed weighted F1, and used the 2.5th and 97.5th percentiles of the bootstrap distribution to form a 95% confidence interval. The mean, variance, low-confidence, and high-confidence values for F1-score obtained on the dataset generated through the normal prompt are 0.9996, 1.61879×10^8, 0.9994, and 0.9998, respectively. On the dataset generated through the strict prompt, the mean, variance, low, and high CI are 0.9681, 1.55683×10^6, 0.9657, 0.97061, respectively. The results obtained during boostrapping are in the same range as the results reported in Table 5, with a slight increase in the mean F1-score for boostrapping, confirming the better performance of the models on the task.

Table 6. Classification scores of one of the models (Albert) on the individual classes of both datasets to demonstrate the impact of high intra-class cohesion of class 1 of dataset 2.

Class/Author	Dataset 1 (Strict Prompt)			Dataset 2 (Normal Prompt)		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score
Class 0 (Human Text)	0.9567	0.9795	0.9780	0.9995	1.00	0.9998
Class 1 (Machine Text)	0.8779	0.7691	0.8199	1.00	0.9971	0.9985

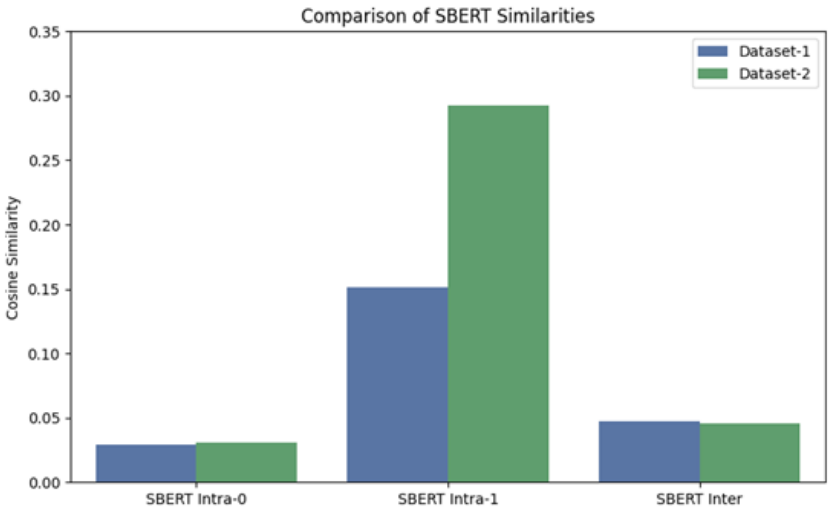

Figure 2. A comparison of SBERT similarities of both datasets. Dataset 1 contains text generated with the normal prompt, while Dataset 2 contains machine-generated text generated with the strict prompts. Moreover, class 0 and class 1 represent human and machine text, respectively.

4.2.2. Classification of Single and Multi-Authored Documents

Table 7 presents the experimental results on the classification of single and multi-authored documents. As can be seen, the models' performance is very high compared to the previous task on the dataset generated with the strict prompt. The higher scores of the models reflect the lower complexity of the task, where the models were able to distinguish between single and multi-authored documents more accurately. Moreover, to better assess the performance of the models, we conducted a bootstrapping experiment for one of the models, namely distilbert, in terms of mean F1-score, the variance in F1-score, and low and high confidence, resulting in 0.9867, 5.40898×10^6, 0.9821, and 0.9912, respectively, for the dataset generated through a strict prompt. On the dataset generated through the normal prompt, the F1-score mean, variance, low, and high confidence are 0.9773, 8.92427×10^6, 0.9713, and 0.9831, respectively. The similarity between the results on the complete dataset and those from bootstrapping experiments confirms the models' better performance on the task.

As noted earlier, in this experiment, the single documents include both human- and machine-generated text, whereas the multi-authored documents comprise text from multiple human authors, as well as machine-generated text embedded at random locations. This performance improvement may be attributed to a comparatively lower reliance on AI-generated text than in the first task, which involved differentiating between human and machine text.

Table 7. Experimental results on the single and multi-authored documents classification task using the normal and strict prompts for AI-text generation.

Model	Normal Prompt				Strict Prompt			
	Accuracy	**Precision**	**Recall**	**F1-Score**	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Distilbert-base-uncased	0.9866	0.9869	0.9866	0.9866	0.9845	0.9844	0.9845	0.9844
Albert-base-v2	0.9841	0.9850	0.9841	0.9842	0.9853	0.9854	0.9853	0.9853
Roberta-base	0.9924	0.9925	0.9924	0.9925	0.9824	0.9825	0.9824	0.9822
Bert-base-uncased	0.9857	0.9862	0.9857	0.9858	0.9774	0.9773	0.9774	0.9772

4.2.3. Author Change Detection in Multi-Authored Documents

Table 8 presents the experimental results of the author change detection task. This task involves predicting the author changes in subsequent paragraphs (i.e., whether the next paragraph is produced by the same author or not). The task involves five authors, including four human authors. The performances of the models dropped significantly compared to previous tasks. The highest F1-scores obtained on the datasets generated with normal and strict prompts are 0.680 and 0.70, respectively. The lower performance is mainly due to the complexity of the task, as also confirmed by the literature [22]. To further explore the experimental results, we analyzed the per-class scores and noticed better performances on the positive class, where the author changes were correctly predicted most of the time. However, some misclassification cases were also observed on class 0, representing the samples where the author did not change.

Table 8. Experimental results on the author change detection task in multi-authored documents using AI-text generated through the normal and strict prompts.

Model	Normal Prompt				Strict Prompt			
	Accuracy	**Precision**	**Recall**	**F1-Score**	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Albert-base-v2	0.674	0.6982	0.674	0.6806	0.6818	0.6964	0.6818	0.6867
Bert-base-uncased	0.5878	0.6795	0.5878	0.5928	0.64	0.6908	0.64	0.6484
Distilbert-base-uncased	0.6699	0.7023	0.6699	0.6774	0.6973	0.7147	0.6973	0.7026
Roberta-base	0.6667	0.6913	0.6667	0.6735	0.6948	0.7149	0.6948	0.7006

We also conducted bootstrap experiments to assess the performance of one of the models, namely distilbert, by resampling data with replacement. The mean F1-scores obtained on both datasets during bootstrapping (0.6788 on the normal prompt and 0.7021 on the strict prompt) are within the same range as reported in the complete dataset. The variance in F1-score, the low, and high confidence reported on the strict dataset are 1.21119×10^5, 0.6720, and 0.6857, respectively. For the dataset generated with the normal prompt, the mean F1-score, variance in F1-score, and the low and high confidence values obtained are 0.7026, 1.17183×10^5, 0.6960, and 0.7090, respectively.

4.2.4. Author Recognition in Multi-Authored Documents

Table 9 presents the experimental results of the final experiment, where we predict the author/s of a document. It is a multi-label classification problem in which multiple authors may have contributed to a document. The F1-scores on both datasets are very low compared to the other tasks. For the calculation of the F1-score for this task, we used optimal thresholds for each class, as these classes have different numbers of samples and different distributions. To find the optimal thresholds, we used a grid search approach by looping through and calculating the F1 score at different thresholds ranging from 0.05 to 0.95 with a 0.01 step size. The final thresholds used for classes 1 to 5 are 0.46, 0.48, 0.56, 0.51, and 0.9,

respectively. Similar to the other tasks, we also conducted a bootstrapping experiment for the DistilBert model to look for any variation in the F1-score by resampling the test samples with replacement. For the dataset generated through the normal prompt, we obtained a mean F1-score of 0.6010, which is higher than the F1-score obtained on the complete set, as reported in Table 9. One of the potential reasons for this increase in results during bootstrapping is the lower performance of the model on certain classes, such as author 3 and author 4, as a smaller number of samples during resampling could have less impact on the overall score of the model. The variance of the F1-score during the bootstrapping experiment is 1.17473×10^6, while the low and high confidence are 0.5988 and 0.6031, respectively. A similar trend is observed on the dataset generated through the strict prompt, where the mean F1-score, variance in F1-score, and the low and high confidence values are 0.5691, 1.71934×10^6, 0.5666, and 0.5716, respectively.

Another interesting observation is the significant differences in the precision and recall values produced by each model. The precision of each model is very low compared to the recall, which indicates that the models are over-predicting the positive samples. During our analysis, we noticed that the performance of the models is very low on certain classes. As a reference, per-class results are provided for the ALBERT-base model in Table 10. The scores, especially the precision for the recognition of authors 2, 3, and 4, are very poor. One of the potential reasons for lower performance on these classes is the low number of samples in these classes. We already explored several data balancing techniques; however, to further improve performance on these classes, more solutions, such as data augmentation and increasing the number of samples by collecting more samples for the authors with fewer samples, could be explored.

Table 9. Experimental results on the author recognition task in multi-authored documents, including AI-text generated through the normal and strict prompts.

Model	Normal Prompt			Strict Prompt		
	Precision	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
Albert-base-v2	0.3756	0.9106	0.5318	0.3869	0.7433	0.5089
bert-base-uncased	0.3564	0.9314	0.5155	0.3835	0.7306	0.503
distilbert-base-uncased	0.355	0.9357	0.5147	0.383	0.7094	0.4974
Roberta	0.3561	0.9369	0.516	0.3842	0.7344	0.5045

Table 10. Experimental results of the Albert model on each class of both datasets for the author recognition task.

Class/Author	Normal Prompt			Strict Prompt		
	Precision	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
Author 1	0.5396	0.999	0.70	0.6483	0.7676	0.7029
Author 2	0.2674	0.9991	0.4212	0.3005	0.7014	0.4208
Author 3	0.1542	0.7297	0.2546	0.2027	0.5624	0.2980
Author 4	0.0964	0.0086	0.0158	0.0890	0.5404	0.1528
Author 5	0.9990	0.9960	0.9978	0.8516	0.7983	0.8241

5. Conclusions and Future Work

In this paper, we analyzed and proposed solutions to key challenges associated with academic integrity in the era of generative AI, focusing on plagiarism and authorship verification. The paper explored four challenges: differentiating between machine-generated and human-generated text, identifying whether a document was produced by a single or multiple authors, identifying paragraphs where the author switches, and identifying the authors for each paragraph of a multi-authored document. Two different sets of machine-generated

text, embedded in a human-produced dataset, were used in the experiments, demonstrating the impact of the prompt (the set of instructions used to generate the machine) on AI-based plagiarism and content authentication tools. During the experiments, we observed a significant reduction in the models' performance at predicting machine-generated text, demonstrating that such tools can be bypassed with cleverly crafted prompts. The better performance of the AI-based solutions for the classification of single and multi-authored documents is very encouraging in ensuring the authenticity of educational content, for instance, an assignment is produced by a single author. Similarly, identifying author changes and author recognition in a collaboratively produced document can also ensure academic integrity. However, the lower experimental results on these two tasks demonstrate the complexity of the tasks.

Despite some initial efforts, several aspects of the domain are yet to be explored. For instance, the domain lacks a large-scale benchmark dataset in the educational context, exploring different aspects of various types of educational content, such as project reports, essays, theses, and other assignments. In this regard, multimodal solutions combining linguistic, semantic, contextual, and metadata features could be very effective, especially for identifying author switching and recognizing authorship in collaboratively produced documents. Moreover, robust solutions, by exploring adversarial frameworks, would help in coping with challenges associated with the detection of AI educational content generated through cleverly crafted prompts. We also believe that incorporating other state-of-the-art models, such as GPT-4 or Llama, would enhance the generalizability of the findings.

Author Contributions: Conceptualization, H.A., and M.A.A. and K.A.; methodology, H.A., N.A., and K.A.; software, M.A.A. and Y.A.; validation, H.A., M.M.A. and N.A.; formal analysis, A.M.A., A.A.O. and N.A.; investigation, H.A., M.A.A. and N.A.; resources, M.M.A., A.M.A., and A.A.O.; data curation, M.A.A., N.A. and K.A.; writing—original draft preparation, H.A. and K.A.; writing—review and editing, N.A., A.A.O., M.M.A. and A.M.A.; visualization, M.A.A. and Y.A.; supervision, A.M.A., A.A.O. and K.A.; project administration, H.A., M.M.A. and K.A.; funding acquisition, H.A., M.M.A. and A.M.A. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the University of Jeddah, Jeddah, Saudi Arabia, under grant No. UJ-24-DR-1290-1.

Data Availability Statement: The code and associated data are publicly available on GitHub https://github.com/MASIFAYUB/Stylometry-Analysis-of-Human-and-Machine (accessed on 5 January 2026), which can be used for research purposes.

Acknowledgments: The authors thank the University of Jeddah for its technical and financial support.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
NLP	Natural Language Processing
SMOTE	Synthetic Minority Over-sampling Technique
CNN	Convolutional Neural Network

References

1. Fariello, S.; Fenza, G.; Forte, F.; Gallo, M.; Marotta, M. Distinguishing Human from Machine: A Review of Advances and Challenges in AI-Generated Text Detection. *Int. J. Interact. Multimed. Artif. Intell.* **2025**, *9*, 6–18. [CrossRef]
2. Mulenga, R.; Shilongo, H. Academic integrity in higher education: Understanding and addressing plagiarism. *Acta Pedagog. Asiana* **2024**, *3*, 30–43. [CrossRef]

3. Jones, B.; Luger, E.; Jones, R. *Generative AI & Journalism: A Rapid Risk-Based Review*; University of Edinburgh: Edinburgh, Scotland, 2023.
4. Škiljić, A. When art meets technology or vice versa: Key challenges at the crossroads of AI-generated artworks and copyright law. *IIC-Int. Rev. Intellect. Prop. Compet. Law* **2021**, *52*, 1338–1369. [CrossRef]
5. Silva, E.C.d.M.; Vaz, J.C. How disinformation and fake news impact public policies?: A review of international literature. *arXiv* **2024**, arXiv:2406.00951. [CrossRef]
6. Balakrishnan, V.; Ng, W.Z.; Soo, M.C.; Han, G.J.; Lee, C.J. Infodemic and fake news – A comprehensive overview of its global magnitude during the COVID-19 pandemic in 2021: A scoping review. *Int. J. Disaster Risk Reduct.* **2022**, *78*, 103144. [CrossRef] [PubMed]
7. Zhou, Y.; He, B.; Sun, L. Humanizing machine-generated content: Evading AI-text detection through adversarial attack. In Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), Turin, Italy, 20–25 May 2024 , pp. 8427–8437.
8. Manzoor, M.F.; Farooq, M.S.; Abid, A. Stylometry-driven framework for Urdu intrinsic plagiarism detection: A comprehensive analysis using machine learning, deep learning, and large language models. *Neural Comput. Appl.* **2025**, 37, 6479–6513. [CrossRef]
9. Manzoor, M.F.; Farooq, M.S.; Haseeb, M.; Farooq, U.; Khalid, S.; Abid, A. Exploring the landscape of intrinsic plagiarism detection: Benchmarks, techniques, evolution, and challenges. *IEEE Access* **2023**, *11*, 140519–140545. [CrossRef]
10. Shapoval, R.V.; Nastyuk, V.Y.; Inshyn, M.I.; Posashkov, A.A. Academic honesty: Current status and ways of improvement. *Justicia* **2021**, *26*, 37–46. [CrossRef]
11. Eaton, S.E. Trust as a foundation for ethics and integrity in educational contexts. *Crit. Stud. Teach. Learn.* **2025**, *13*, 4–7. [CrossRef]
12. Fundamental, T. *Values of Academic Integrity*; The Center for Academic Integrity: Des Plaines, IL, USA, 1999.
13. Bretag, T.; Green, M. The role of virtue ethics principles in academic integrity breach decision-making. *J. Acad. Ethics* **2014**, *12*, 165–177. [CrossRef]
14. Balalle, H.; Pannilage, S. Reassessing academic integrity in the age of AI: A systematic literature review on AI and academic integrity. *Soc. Sci. Humanit. Open* **2025**, *11*, 101299. [CrossRef]
15. Baron, P. Are AI detection and plagiarism similarity scores worthwhile in the age of ChatGPT and other Generative AI? *Scholarsh. Teach. Learn. South (SOTL) South* **2024**, *8*, 151–179. [CrossRef]
16. Ardito, C.G. Generative AI detection in higher education assessments. *New Dir. Teach. Learn.* **2025**, *2025*, 11–28. [CrossRef]
17. Elkhatat, A.M.; Elsaid, K.; Almeer, S. Evaluating the efficacy of AI content detection tools in differentiating between human and AI-generated text. *Int. J. Educ. Integr.* **2023**, *19*, 17. [CrossRef]
18. Malik, M.A.; Amjad, A.I. AI vs AI: How effective are Turnitin, ZeroGPT, GPTZero, and Writer AI in detecting text generated by ChatGPT, Perplexity, and Gemini? *J. Appl. Learn. Teach.* **2025**, *8*, 91–101.
19. Michailidis, P.D. A scientometric study of the stylometric research field. *Informatics* **2022**, *9*, 60. [CrossRef]
20. Zamir, M.T.; Ayub, M.A.; Khan, J.; Ikram, M.J.; Ahmad, N.; Ahmad, K. Document provenance and authentication through authorship classification. In *Proceedings of the 2023 1st International Conference on Advanced Innovations in Smart Cities (ICAISC)*; IEEE: Jeddah, Saudi Arabia, 2023; pp. 1–6.
21. Oberreuter, G.; Velásquez, J.D. Text mining applied to plagiarism detection: The use of words for detecting deviations in the writing style. *Expert Syst. Appl.* **2013**, *40*, 3756–3763. [CrossRef]
22. Zamir, M.T.; Ayub, M.A.; Gul, A.; Ahmad, N.; Ahmad, K. Stylometry analysis of multi-authored documents for authorship and author style change detection. *arXiv* **2024**, arXiv:2401.06752.
23. Zangerle, E.; Mayerl, M.; Tschuggnall, M.; Potthast, M.; Stein, B. Pan21 Authorship analysis: Style Change Detection. In Proceedings of the CEUR Workshop Proceedings, Bucharest, Romania, 21–24 September 2021.
24. Strøm, E. Multi-label Style Change Detection by Solving a Binary Classification Problem. In Proceedings of the CLEF 2021—Conference and Labs of the Evaluation Forum, Bucharest, Romania, 21–24 September 2021; pp. 2146–2157.
25. Kestemont, M.; Tschuggnall, M.; Stamatatos, E.; Daelemans, W.; Specht, G.; Stein, B.; Potthast, M. Overview of the author identification task at PAN-2018: Cross-domain authorship attribution and style change detection. In Proceedings of the Working Notes Papers of the CLEF 2018 Evaluation Labs, Avignon, France, 10–14 September 2018; pp. 1–25.
26. Zangerle, E.; Mayerl, M.; Potthast, M.; Stein, B. Overview of the Style Change Detection Task at PAN 2020. *CLEF (Work. Notes)* **2020**, *93*.
27. Kumarage, T.; Garland, J.; Bhattacharjee, A.; Trapeznikov, K.; Ruston, S.; Liu, H. Stylometric detection of ai-generated text in twitter timelines. *arXiv* **2023**, arXiv:2303.03697. [CrossRef]
28. Opara, C. StyloAI: Distinguishing AI-generated content with stylometric analysis. In *Proceedings of the International Conference on Artificial Intelligence in Education*; Springer: Berlin/Heidelberg, Germany, 2024; pp. 105–114.
29. Fabien, M.; Villatoro-Tello, E.; Motlicek, P.; Parida, S. BertAA: BERT fine-tuning for Authorship Attribution. In Proceedings of the 17th International Conference on Natural Language Processing (ICON), Patna, India, 18–21 December 2020; pp. 127–137.

30. Almutairi, A.; Kang, B.; Al Hashimy, N. Bibert-av: Enhancing authorship verification through siamese networks with pre-trained bert and bi-lstm. In *Proceedings of the International Conference on Ubiquitous Security*; Springer: Berlin/Heidelberg, Germany, 2023; pp. 17–30.
31. Grootendorst, M. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. *arXiv* **2022**, arXiv:2203.05794.
32. Chauhan, U.; Shah, A. Topic modeling using latent Dirichlet allocation: A survey. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 1–35. [CrossRef]
33. Lee, D.; Seung, H.S. Algorithms for non-negative matrix factorization. *Adv. Neural Inf. Process. Syst.* **2000**, *13*, 535–541.
34. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]
35. Batista, G.E.; Prati, R.C.; Monard, M.C. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explor. Newsl.* **2004**, *6*, 20–29. [CrossRef]
36. He, H.; Bai, Y.; Garcia, E.A.; Li, S. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *Proceedings of the 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*; IEEE: Hong Kong, China, 2008; pp. 1322–1328.
37. Nayan, N.M.; Islam, A.; Islam, M.U.; Ahmed, E.; Hossain, M.M.; Alam, M.Z. Smote oversampling and near miss undersampling based diabetes diagnosis from imbalanced dataset with xai visualization. In *Proceedings of the 2023 IEEE Symposium on Computers and Communications (ISCC)*; IEEE: Gammarth, Tunisia, 2023; pp. 1–6.
38. Konuma, R.; Huaping, Z.; Gao, C.; Wang, J. Japanese Author Attribution Using BERT Finetuning with Stylometric. In *Intelligent Multilingual Information Processing: First International Conference, IMLIP 2024, Beijing, China, 16–17 November 2024, Proceedings*; Springer Nature: Berlin/Heidelberg, Germany, 2025; Volume 2395, p. 293.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.