Multi-Loss Fusion: Angular and Contrastive Integration for Machine-Generated Text Detection

Anonymous ACL submission

Abstract

Modern natural language generation (NLG) systems have led to the development of synthetic human-like open-ended texts, posing concerns as to who the original author of a text is. To address such concerns, we introduce DeB-Ang: the utilisation of a custom DeBERTa model (He et al., 2021) with angular loss and contrastive loss functions for effective class separation in neural text classification tasks. We expand the application of this model on binary machine-generated text detection and multi-class neural authorship attribution. We demonstrate improved performance on many benchmark datasets whereby the accuracy for machine-generated text detection was increased by as much as 38.04% across all datasets.

1 Introduction

There has been considerable activity in the field of detecting machine-generated text. Driven by the significant growth and the increasing prevalence of large language models (LLMs) and natural language generation (NLG) models. This has led to the production of high-quality human-like texts that have brought about useful applications in many domains such as machine translation, text summarisation and data generation (Kieuvongngam et al., 2020; Goyal et al., 2022; Iyer et al., 2023; Uchendu et al., 2021). Irrespective of the many useful applications, the deployment of NLG models has concurrently given rise to serious concerns such as plagiarism, and spreading misinformation and hate speech (Pu et al., 2022; Hu et al., 2023; Qadir, 2022; Solaiman et al., 2019). Therefore, the need to discriminate between human and machine-generated text becomes paramount, especially in light of the growing sophistication and rapid updates of these models.

Given the diverse applications of NLG models, authorship attribution (AA) methods have been increasingly employed to detect the original author of synthetic data generated by machines (Ai et al., 2022; Uchendu et al., 2020; Jawahar et al., 2020). The main concern with traditional AA methods is that, typically, they are feature-based systems and consist of largely document-specific features. Therefore, the application of this traditional model is often author, dataset and model-specific (Sari, 2018; Ai et al., 2022). Previous research addressed the need for generalisable detection systems to identify machine-generated text (Fagni et al., 2021; Jakesch et al., 2023; He et al., 2024; Jawahar et al., 2020). Research involving the use of LLMs in authorship attribution has demonstrated that the simple fine-tuning of pre-trained language models can surpass the accuracy of traditional methods significantly (Fabien et al., 2020; Mitrović et al., 2023; Fagni et al., 2021).

In particular, we introduce DeB-Ang, a pre-trained DeBERTa model with a specialised angular loss and contrastive loss integration. Additionally, we demonstrate improved classification when applying DeB-Ang to several well-known machine-generated text and authorship attribution datasets. Contrastive learning is an unsupervised representation learning technique, aiming to learn a representation of data such that similar instances are close in the representation space whereas dissimilar instances are far apart (Aljundi et al., 2022). Loss functions are crucial in contrastive learning as they quantify the similarity and dissimilarity between pairs, guiding the model to learn meaningful representations for class discrimination (Hadsell et al., 2006; Gao et al., 2022; Wang et al., 2017). However, recent studies suggest that various loss functions, including cross-entropy loss, contrastive loss and triplet loss, fail to consider the intrinsic angular distribution exhibited by the low-level and high-level feature representations (Choi et al., 2020), which contributes to our choice of using angular loss in DeB-Ang. Angular loss is a scale-invariant loss function designed to improve the learning similarity metrics by considering the angle between vectors (Wang et al., 2017).

In summary, the contributions of this work are four-fold:

1. We propose a novel customisable contrastive learning framework that combines a custom fine-tuned DeBERTa model (He et al., 2021) with contrastive and angular loss functions. We assess the difference in classification performance when utilising various combinations of the aforementioned loss functions for the proposed task.
2. We assess the application of the proposed model on multi-class authorship attribution and binary machine-generated text detection.
3. We introduce three new large-scale datasets for evaluating text classification models. These datasets were constructed by leveraging state-of-the-art language models, including Gemma-7b (Team et al., 2024), GPT4-Turbo (OpenAI, 2023) and Flan-T5-Large (Chung et al., 2022).
4. We conduct linguistic error analysis of incorrectly and correctly classified examples.

2 Related Work

2.1 Machine-generated text detection

Studies have demonstrated that human participants were unable to distinguish between machine-generated texts and human written texts (Jakesch et al., 2023; Islam et al., 2023; Ippolito et al., 2020; Dugan et al., 2020, 2022). Previous work highlighted that disambiguating between human and LLM-generated texts is increasingly difficult (Pu and Demberg, 2023; Jakesch et al., 2023; Cox, 2005). Automatic detection of machine-generated text has thus gained popularity, and can be categorised according to their underlying method (Solaiman et al., 2019; Uchendu et al., 2020; Fagni et al., 2021; Bakhtin et al., 2019; Ippolito et al., 2020). Simple classifiers often involve linguistic feature analysis (Dugan et al., 2020, 2022). Other methods include zero-shot detection (Solaiman et al., 2019), and fine-tuned model detection (Uchendu et al., 2020; Ippolito et al., 2020; Fagni et al., 2021; Adelani et al., 2019; Tay et al., 2020; Zellers et al., 2021). Irrespective of the large number of approaches to identifying machine-generated texts, detection remains a challenge (Crothers et al., 2023; Ai et al., 2022).

2.2 Authorship Attribution

Traditional attribution approaches utilise linguistic features in a univariate (utilising a single linguistic feature, e.g. function words) (Martindale and McKenzie, 1995) or multivariate (utilising multiple linguistic features, e.g Writeprints) approach (Abbasi and Chen, 2008; Sari, 2018). As aforementioned, feature-based linguistic identification requires dataset-specific engineering, displaying limited scalability (Sari, 2018; Ai et al., 2022). More recently, the use of learning-based approaches has grown with the use of pre-trained LLMs (Fabien et al., 2020). These approaches have demonstrated the power of LLMs in significantly surpassing the accuracy of traditional approaches with little analysis required beforehand (Ai et al., 2022).

2.3 Research gaps

Existing approaches in detecting synthetic texts created by LLMs have many limitations. For example, these detection tools are now outdated due to rapid technological advancements, e.g., DetectGPT classifies texts only generated by GPT2 (Mitchell et al., 2023). This necessitates classifier retraining which could negatively affect the accuracy of these models (OpenAI, 2023). Additionally, the increased advancements of NLG models have led to more human-like texts. Further, these models are LLM-specific and therefore, do not detect synthetic texts generated by other language models. Also, these methods have a black-box nature, making it difficult for humans to understand their output for correctly and incorrectly classified texts. Given the increasing prevalence of machine-generated texts, it is vital that we are able to distinguish which NLG model was used to generate a given text. We extend this to being able to detect the exact model version.

3 Data

3.1 Data collection

There is a strong consensus that datasets must be diverse and representative (Tang et al., 2023). To this end, we chose to utilise datasets with original, human-written texts; different versions of each text are then generated with the aid of LLMs which were given carefully designed prompts. Datasets were taken from Kaggle and the Turing Test benchmark known as TuringBench (Uchendu et al., 2021). The TuringBench dataset consists of articles generated by 20 authors. There are a total of 20 datasets from 19 different NLG model versions and one human author. The DAIGT-V2 dataset consists of 37 authors (36 NLG models and one human author, with 60K texts). Further dataset details can be seen in Appendix A in Table 7. All datasets utilised were generated for the proposed tasks.

Specifically, we utilised 5 randomly selected datasets from TuringBench datasets. We opted to use one dataset per model. For example, there are two datasets generated by XLNET. The exact model verions are XLNET_base and XLNET_large. Therefore, we employ only one of these model versions. We randomly sampled the dataset due to limited computational resources. Details of the specific processing steps and size of the data taken for each of the different datasets are provided in Section 5. Table 7 in Appendix A presents—for each dataset that we utilised—the dataset name, source and the models that were used to generate the texts contained in each dataset.

3.2 Data Generation

We also generated our own datasets by using GPT4-Turbo (OpenAI, 2023), Gemma-7b (Team et al., 2024) and Flan-T5-large (Chung et al., 2022). The TuringBench dataset set consists of 19 different NLG model versions however, these models are no longer considered state-of-the-art models. We decided to generate three additional datasets from more recent models which are considered to be the current state-of-the-art and were not included in the original TuringBench dataset. This dataset set is referred to as TuringExtended. This enables the examination of MGT and AA within the context of newer NLG models, underpinning the exploration as to whether newer NLG models are more challenging to identify as machine-generated.

Specifically, additional datasets were generated as an extension of TuringBench (Uchendu et al., 2021). Considering only the human-written texts from the original AA dataset from TuringBench, we extracted only a total of 7678 (non-duplicated) rows of text. The models that we employed are GPT4 Turbo[1] (He et al., 2021), Gemma-7b[2] (Team et al., 2024) and Flan-T5-large[3] (Chung et al., 2022), resulting in the creation of three new datasets. The models were given the prompt *"generate a similar article"*, which is slightly similar to what was used in TuringBench (*"generate an article similar to the human-written one"*).

[1]https://platform.openai.com/docs/models
[2]https://huggingface.co/google/Gemma-7b
[3]https://huggingface.co/google/flan-t5-large

	GPT4-Turbo	Gemma-7b	Flan-T5-Large
BERTScore P	83.30	92.32	90.51
BERTScore R	84.11	95.55	83.98
BERTScore F1	83.70	93.88	87.08
IAA	88.64	89.30	84.64

Table 1: Averaged BERTScore Precision (P), Recall (R) and F1-score (F1) for the datasets generated by the specified models. Inter-annotator agreement (IAA) is also provided.

The three models for generation (GPT4-Turbo, Gemma-7b and Flan-T5-large) were chosen on the basis that they were either the current state-of-the-art models, or that they were not previously employed in creating the TuringBench datasets.

3.3 Data Evaluation

We evaluated the quality of the generated data using a combination of the automated metric BERTScore (Zhang et al., 2020) and human evaluators. BERTScore calculates token similarity using contextual embeddings to calculate the similarity between tokens in the candidate and reference text. This metric has demonstrated an advanced performance by correlating strongly with human judgement in various evaluative tasks (Zhang et al., 2020). In parallel, four human annotators were trained on evaluating generated text and were provided with some background information on text generation. Each annotator assessed 250 rows from each dataset and was asked to label the data as coherent (0) or incoherent (1). For a data sample to be labelled as coherent it had to meet two criteria: texts should be semantically and grammatically sound. Inter-annotator agreement (IAA) was then measured between all annotators for each dataset. The averaged BERTScore precision, recall and F1-scores, and IAA results are presented in Table 1.

4 Methodology

4.1 Loss Functions

Previous studies have focussed on increasing similarity between representations by using varying loss functions (Ai et al., 2022; Vygon and Mikhaylovskiy, 2021). However, many approaches focus on the utilisation of a single loss function. In this paper, we propose a multi-loss fusion by using the weighted sum of a combination of various loss functions: angular loss, cross-entropy loss and contrastive loss. Cross-entropy loss measures probability distributions; the objective is to minimise the error between the predicted probability and true distribution (Mao et al., 2023). This is used in updating the model weights during optimisation. Angular loss, often used in deep learning tasks (Wang et al., 2017; Kim et al., 2023, 2021; Choi et al., 2020), considers the angle between vectors to enhance learning for an improved similarity metric. The utilisation of angular loss in text classification often leads to more adaptable and robust models capable of handling linguistic diversity (Gao, 2022; Hui et al., 2019; Wang et al., 2017; Deng et al., 2019). Contrastive learning focusses on learning representations of data so that similar instances are closer in the embedding space and dissimilar instances are apart (Tan et al., 2024).

4.2 Problem Statement

The goal of our approach is to capture nuanced semantic representations and to effectively discriminate learned embeddings. We propose leveraging the DeBERTa model with angular and contrastive loss integration (DeB-Ang). This process aims to enhance the discriminative capabilities and quality of embeddings to improve the model's performance on downstream classification tasks.

4.3 Implementation

Our textual datasets underwent cleaning and preprocessing procedures. A 70:10:10 split was applied to partition the data into a training, validation and test sets for model evaluation.

Building upon the DeBERTa base model (`microsoft/deberta-base`), we implemented a new model, DeB-Ang, that integrates the angular and contrastic loss into the training step. The model was implemented using PyTorch (Paszke et al., 2019) and Simple Transformers[4] and was configured with specific hyperparameters (See Appendix B); additionally, early stopping criteria were set to improve training efficiency.[5]

4.4 Angular Loss Computation

The angular loss function begins by computing the cosine similarity between all pairs of extracted embeddings. Positive and negative pairs are then generated to ensure the model can distinguish between embeddings with the same and different labels. Subsequently, we compute the loss for positive and negative pairs in order to optimise embeddings to have lower similarity for pairs with different labels and higher similarity for those with the same. This ultimately decides their position in the embedding space. The sum of positive and negative loss creates a complete loss function. This function, given below, guides the optimisation to achieve embeddings that have properties of similarity and dissimilarity.

$$L_{Angular} = \sum_{i=1}^{n} \log\left(\sum_{j \neq i} e^{s_{ij}}\right) - \log\left(\sum_{j \neq i} e^{s_{ji}}\right) \quad (322)$$

where:

- n is the number of embeddings;
- s_{ji} is the cosine similarity between embeddings i and j;
- The first term encourages embeddings from different classes (negative pairs) to have lower cosine similarity;
- The second term encourages embeddings from the same class (positive pairs) to have higher cosine similarity.

4.5 Contrastive Loss Computation

Generating positive pairs facilitates the learning of intra-class relationships by allowing embeddings with the same labels but different indices to be considered for optimisation. Generating negative pairs enhances the discrimination capability of these embeddings. We compute the loss for positive and negative pairs. This allows embeddings of similar instances to be pushed closer to each other in the embedding space, whereas negative embeddings push them apart thus improving intra-class clustering and inter-class separation. Combining the positive and negative loss, as shown below, guides the model to learn embeddings that capture both intra-class relationships and inter-class distinctions.

$$L_{Contrastive} = \sum_{i,j} y_{ij} d_{ij} + (1 - y_{ij}) \max(0, m - d_{ij})$$

where:

- y_{ij} is a binary label indicating whether embeddings i and j belong to the same class (1) or different classes (0);
- d_{ij} is the distance between embeddings i and j;

[4]SimpleTransformers: https://simpletransformers.ai/docs/classification-specifics/

[5]Our datasets and code will be made publicly available upon paper acceptance.

- m is a margin hyperparameter;
- For positive pairs ($y_{ij} = 1$), the loss is d_{ij}, encouraging embeddings to be closer together;
- For negative pairs ($y_{ij} = 0$), the loss is $\max(0, m - d_{ij})$, encouraging embeddings to be apart by at least a distance of m.

4.6 Our DeB-Ang Model

In the DeB-Ang model, we utilise three loss functions, as shown in the equation below: angular loss, cross-entropy loss and contrastive loss. Angular loss is used to facilitate intra-class compactness and inter-class separation. Within our model, we utilise cross-entropy loss to penalise the models' misclassification by computing the difference between predicted and actual labels. Cross-entropy is the standard loss function that was incorporated into the DeBERTa model. Constrastive loss enhances the embeddings' discriminative abilities by encouraging similarity for positive pairs and dissimilarity for negative pairs.

$$
\begin{aligned}
L_{Total} = {} & w_{CE} L_{CE} \\
& + w_{Angular} L_{Angular} \\
& + w_{Contrastive} L_{Contrastive}
\end{aligned}
$$

where:

- L_{Total} is the total loss function used for training the DeBERTa model;
- L_{CE} is the standard cross-entropy loss for classification tasks, calculated as $L_{CE} = -\sum_{i=1}^{n} \log P(y_i|X)$, where X is the input sequence and y_i is the true label for the i-th example;
- $L_{Angular}$ is the angular loss based on cosine similarity;
- $L_{Contrastive}$ is the contrastive loss;
- w_{CE}, $w_{Angular}$, and $w_{Contrastive}$ are the corresponding weights for each loss component, allowing for fine-tuning the contribution of each loss value during training.

This combined loss function incorporates three learning objectives:

1. The cross-entropy loss which ensures that the model learns to correctly classify the input sequences based on the true labels.
2. The angular loss which encourages the model to learn more separated representations for different classes, based on the cosine similarity between the embeddings.
3. The contrastive loss further enforces the separation between inter-class embeddings, while bringing intra-class embeddings closer together, based on the similarity calculations and a specified margin.

By combining these three loss components, the DeBERTa model can potentially learn more robust and discriminative representations, leading to improved classification performance on various natural language processing tasks.

4.7 Evaluation and Error Analysis

Considering the scale of the datasets, some accuracy values, when taken at face value, may not demonstrate any meaningful improvement in performance. Therefore, we utilise McNemar's test (Sundjaja et al., 2023) to demonstrate the statistical significance of our results. McNemar's test is a non-parametric test that can be used in comparing the performance of two classification models.

For error analysis, we extracted both incorrectly classified and correctly classified data samples and performed an in-depth linguistic analysis of the outputs. We also computed the semantic similarity between correctly and incorrectly classified data by measuring the cosine similarity between the embeddings of the text pairs. We extracted contextual embeddings using the same DeBERTa model.

5 Results and Discussions

5.1 Machine-generated Text Detection

In this section, we investigate binary machine-generated text detection, whereby the task is focussed on differentiating between human and machine-written texts. Table 2 presents the results for this task on a variety of datasets from TuringBench. From the table, it is evident that the proposed model outperforms both Contra-X (Ai et al., 2022) and a baseline DeBERTa model with a minimum improvement of 0.23% and maximum improvement of 38.04% in accuracy. Statistical significance was computed by comparing DeB-Ang with the baseline DeBERTa model, as they exhibited the closest performance. The results for machine-generated text detection for the TuringExtended data is presented in Table 3. This demonstrates that the DeB-Ang model can differentiate between human and machine-generated texts even if the latter were generated by the newer NLG models, displaying detection accuracy over 96% for texts generated by Flan-T5-Large, GPT-4Turbo and Gemma-7b.

Accuracy and F1-score									
	Contra-X		Baseline DeBERTa		DeB-Ang		SS (Y/N)	Min-Max Improvement	
PPLM-gpt2	99.34	99.32	99.66	99.66	99.98	100	Y	0.32	0.64
FAIR-wmt20	61.95	60.94	99.39	99.36	99.99	99.98	Y	0.60	38.04
GPT-3	97.85	97.85	98.8	98.8	99.73	99.73	Y	0.93	1.88
Grover-large	99.26	99.26	99.76	99.77	99.99	99.99	Y	0.23	0.73
transfo-xl	97.85	97.85	99.69	99.69	99.99	99.99	Y	0.30	2.14

Table 2: Accuracy and F1-score for the baseline Contra-X, DeBERTa, and the proposed DeB-Ang model on various TuringBench datasets containing texts generated by different NLG models (rows). Min-Max refers to the minimum and maximum classification accuracy that DeB-Ang obtained for each dataset. Statistical significance (SS) between baseline DeBERTa and DeB-Ang is either yes (Y) or no (N) according to McNemar's test.

From our initial experimentation, we noted that the baseline DeBERTa model outperforms other approaches in binary machine-generated text detection; therefore, for the remaining experiments we proceed with .

Accuracy and F1-score						
	Baseline DeBERTa		DeB-Ang		SS (Y/N)	Min-Max Improvement
Flan-T5-Large	92.14	92.14	96.99	96.99	Y	4.85
Gemma-7b	99.96	99.96	99.99	99.99	N	0.03
GPT4-Turbo	72.14	72.14	99.96	100	Y	27.82

Table 3: Accuracy and F1-score for the baseline DeBERTa and the proposed DeB-Ang model for TuringExtended. Min-Max refers to the minimum and maximum classification accuracy that DeB-Ang obtained for each dataset. Statistical significance (SS) between baseline DeBERTa and DeB-Ang is either yes (Y) or no (N) according to McNemar's test.

The results for the DAIGT-V2 dataset can be seen in Table 5. This improvement demonstrates the models' generalisability across various NLG datasets, for both older and newer models. Uchendu et al. (2021) comments *"No one size fits all"* in their study as they used several models on these datasets and found that different models obtain different levels of performance, depending on the dataset. However, as presented in Table 2, it is clear that the model consistently outperforms our baseline models on all datasets.

5.2 Authorship Attribution

In assessing the generalisability of the DeB-Ang approach on various text classification settings, we present the following authorship attribution tasks:

1. Authorship attribution for human and machine-generated text detection.
2. Authorship attribution for model variation detection, e.g. differentiating between GPT-3.5 and GPT-4.

	Accuracy	F1	SS (Y/N)
Syntax-CNN	66.13	64.8	Y
BERT-AA	78.12	77.58	Y
Contra-X	80.73	80.54	Y
TopRoBERTa	82.83	82.00	Y
Baseline DeBERTa	77.71	77.56	Y
DeB-Ang	83.61	82.68	Y

Table 4: Accuracy and F1 for the authorship attribution (AA) dataset from TuringBench (Uchendu et al., 2021) comparing various AA approaches. McNemar's test was conducted to see if the result between DeB-Ang and all other models is statistically significant (SS) or not.

3. Authorship attribution for model developer detection, e.g. OpenAI for GPT-4 and GPT-3.5.

The results for each task is presented in Table 5 under authorship attribution, model detection and developer detection, respectively.

From Table 4, it is evident that our approach also surpasses prior attempts on the TuringBench dataset. As previously mentioned, this dataset consists of texts generated by 20 different authors (a total of 200K texts from 19 NLG models and 1 human author) with high topical dissimilarity between each model. This dissimilarity is expected as the dataset was generated in certain topic set (Ai et al., 2022; Uchendu et al., 2021). For the DAIGT-V2 dataset, we downsized the data to approximately 10K rows per model. This reduction was necessitated due to the dataset's size, which demanded significant computational resources. In Table 5, it can observed that the DeB-Ang model outperforms the baseline DeBERTa model with an accuracy improvement of 1.80% in authorship attribution and 8.66% in machine-generated text detection. To delve deeper into the machine-generated text results from the baseline DeBERTa, we conducted

		Baseline DeBERTa		DeB-Ang		Accuracy
Task	Model	Accuracy	F1	Accuracy	F1	Improvement
Machine-generated text detection		82.69	90.53	91.36	92.31	8.66
Authorship attribution [37]		86.00	85.96	87.80	87.79	1.80
Model detection	Open AI [10]	88.64	88.64	91.75	91.75	3.11
	Meta [13]	42.27	42.29	47.95	47.51	5.68
	Google [7]	56.96	56.55	57.60	57.78	0.64
	Anthropic [2]	95.63	95.58	99.03	99.03	3.40
	Mistral [4]	93.15	93.39	93.96	95.12	0.81
Developer detection	All [5]	89.78	89.78	92.98	92.98	3.20

Table 5: Table presenting evaluation results on the DAIGT-V2 dataset, including authorship attribution scores for all NLG models, machine-generated text detection (human vs. machine), model detection (distinguishing between different model variations), and authorship attribution for model developers. The numbers in brackets (e.g., "Open AI [10]") indicate the number of classes (i.e., the number of models).

an analysis focussing on the disparity between the accuracy and F1-score. This involved computing the Area Under the Receiver Operating Characteristic (AUROC) score and assessing misclassification. Our analysis revealed that the model exhibited a considerable number of false positives, incorrectly predicting a majority of human-written texts. The AUROC score was determined to be 50.12 whereas the AUROC score for DeB-Ang was 88.14 indicating DeB-Ang's superior discrimination capabilities. We also address the previously mentioned limitation regarding the scarcity of research in classifying models from a single developer; our results are provided in Table 5. We investigate a range of developers and models varying from older to newer model versions. We were able to improve results from baseline DeBERTa for this task by 0.64% to 5.68%. The low accuracy observed for Meta and Google models can be attributed to the high similarity between the model variations used, e.g., Llama-2-7b and Llama-2-13b. This makes distinguishing between these version challenging, leading to misclassification. Further investigation is necessary to comprehensively understand the reasons for misclassification. We were also able to classify generated texts according to model developer with an accuracy as high as 92.98%.

As mentioned in prior research, classifying a range of outputs, e.g. texts with high topic variation, is an increasingly difficult classification task (Uchendu et al., 2021; Juola, 2008). Furthermore, it is important to note that TuringBench consists of texts from multiple sources. Also, some models are being used repeatedly to generate texts; this can decrease performance as there can be semantics and stylistic overlap between generated texts.

5.3 Assessing loss functions

To assess the significance of the loss functions used, we investigated various combinations of loss functions on the multi-class authorship attribution and binary machine-generated text detection tasks. The results are presented in Table 6. We provide the accuracy, F1-score and AUROC scores for these tasks obtained by the DeB-Ang model. We ran each experiment for one epoch for initial benchmarking to assess each models performance. This allowed us to identify the which approaches we would use for further investigations. We identified the optimal parameter combination for the loss functions for each task and re-ran the experiment for 8 epochs. The aim of this investigation is to assess the performance improvement resulting from the various loss functions. This also highlights the customisability of the model. We extend the metrics by adding the AUROC score as this metric considers the trade-off between precision and recall (McDermott et al., 2024).

We found that a certain loss function combination may ascertain significant results at one epoch given a simple model. However, once the model or dataset complexity increases then a different loss combination would be more appropriate. Angular loss has the advantage of learning embeddings such that similar samples have a smaller angular separation. It is vital to understand that angular loss focusses on learning embeddings (Wang et al., 2017) whereas cross-entropy focusses on measuring the dissimilarity between predicted and true probability distribution of classes (Teahan, 2000). This difference may account for the accuracy difference.

Task	Loss function	Accuracy	F1	AUROC
Authorship Attribution	ANG + CE	86.51	86.54	98.12
	CL + CE	86.72	86.78	98.07
	CL + ANG	87.02	87.08	98.12
	ANG	86.37	86.42	97.88
	CL	86.72	86.63	98.11
	CE	86.87	86.71	98.00
	ANG + CE + CL (1.0; 1.0; 1.0)	86.35	86.41	97.97
	ANG + CE + CL (1.0; 0.25; 1.0)	87.13	8722	98.15
8 epoch	ANG + CE + CL (1.0; 0.25; 1.0)	88.8	88.06	98.5
Binary machine-generated text detection	ANG + CE	90.87	91.89	77.00
	CL + CE	91.6	92.47	90.89
	CL + ANG	94.18	94.61	95.61
	ANG	94.49	94.86	95.61
	CL	88.15	89.19	72.58
	CE	91.69	92.33	92.83
	ANG + CE + CL (1.0; 1.0; 1.0)	90.76	91.92	87.57
	ANG + CE + CL (1.0; 0.75; 0.75)	82.69	90.53	78.71
8 epoch	ANG + CE + CL (1.0; 0.75; 0.75)	93.76	88.23	90.94

Table 6: Comparison of single and combined loss functions for AA and machine-generated text detection using the DeB-Ang model with varying number of epochs. Parameter values for all loss functions were set to 1.0 and all experiments were run for 1 epoch unless otherwise specified. Key: AUROC = area under the receiver operating characteristic, CE = cross-entropy loss, CL = contrastive loss and ANG = angular loss. The values in brackets refer to the parameter values.

5.4 Analysing the misclassified data

For our error analysis, 100 instances of incorrect and correct classifications were extracted for the binary classification task. We found that texts were being labelled as machine-generated more frequently than human data; this could be due to the class imbalance or due to the NLG model's ability to create human-like text.

Based on the manual analysis, there was no specific linguistic category which would clearly lead to the misclassification. Therefore, we extracted features from varying categories (see Table 9 in Appendix C). A total of 250 features were extracted. 100 random features were sampled and the raw counts and mean for each feature was plotted (see Figure 1 in Appendix C). From this plot, it is evident that there is a clear discrepancy in feature usage. The correctly classified data points exhibit lower feature counts and an overall lower mean whereas the incorrectly classified data is slightly more sporadic and exhibits an overall higher mean. The statistical significance for these differences for all features was computed using the Mann-Whitney U test (Nachar, 2008) as the data was not normally distributed (as affirmed by the Shapiro-Wilk test) (Aryadoust and Raquel, 2020). The statistical significance was less than 0.05 thus rejecting the null hypothesis and confirming the difference between the feature counts and mean for the correctly classified and incorrectly classified data is significant.

We then measured the semantic similarity between correctly classified and incorrectly classified instances using contextual embeddings obtained using DeBERTa. The mean similarity score for all data points is 85.54 (minimum score of 63.22 and maximum score of 92.64). Figure 2 in Appendix C presents a correlation coefficient of -0.03 indicating a very weak negative linear relationship almost suggesting no linear relationship between the data points. This indicates that any observed differences or similarities in the similarity score are likely due to random variation and not a meaningful underlying relationship. We conclude that the similarity scores do not provide useful information to distinguish between correctly and incorrectly classified instances.

6 Conclusion and Future Work

In this research, we have created a custom DeBERTa model integrating contrastive and angular loss. To our knowledge, this is one of the first attempts at this integration and we have demonstrated the success of the proposed DeB-Ang model on several datasets. We investigated more fine-grained machine-generated text detection by classifying model variations and developers. We were able to outperform prior approaches in machine-generated text detection with a minimum improvement of 0.23 % and a maximum improvement of 38.04% across all datasets. We were able to classify model variations with accuracy scores ranging from 0.64% to 5.68%, and to identify developers with an accuracy improvement of 3.20%. For authorship attribution, we were able to improve classification with a maximum accuracy of 17.48% on the extensive TuringBench dataset which is characterised by high topical dissimilarity. Future work will involve identifying texts in which multiple NLG models or humans have been used to intentionally mask the writing style of a text. Additionally, a more extensive examination of linguistic features of synthetic data across generations of LLMs can provide insights into language evolution these models.

7 Limitations

Guerrero and Alsmadi (2022) lists several research gaps in the field of machine-generated text detection e.g. domain-specific text detection. It would be interesting to investigate texts that are cross-domain, genre or multimodal. Further, we investigated misclassified instances but did not use this information to improve the model due to time constraints. The limitations associated with data generation are model-related. Data generation is a time-consuming process and requires many computational resources; we were only able to extend our evaluation data with three datasets.

Ethics Statement

The materials used for this study did not require human participation and the data does not contain any harmful or sensitive information. The datasets used in this study were acquired from prior research. The dataset generated using NLG models (Open AI's GPT-4 model, Gemma-7b and Flan-T5-large) was evaluated to ensure that there is no overtly harmful text. Data was annotated and evaluted by PhD students, the task was explained in regards to how data will be used and proposed tasks. Nevertheless, the potential negative use of this research should not be ignored. The insights provided by this work have the potential to be exploited for malicious purposes, potentially undermining the effectiveness of these detectors. However, we hope that this research will be used to support the efforts in detecting neural machine-generated used in applications with malicious intent.

References

Ahmed Abbasi and Hsinchun Chen. 2008. Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace. *ACM Transactions on Information Systems*, 26(2):1–29.

David Ifeoluwa Adelani, Haotian Mai, Fuming Fang, Huy H. Nguyen, Junichi Yamagishi, and Isao Echizen. 2019. Generating Sentiment-Preserving Fake Online Reviews Using Neural Language Models and Their Human- and Machine-based Detection.

Bo Ai, Yuchen Wang, Yugin Tan, and Samson Tan. 2022. Whodunit? Learning to Contrast for Authorship Attribution.

Rahaf Aljundi, Yash Patel, Milan Sulc, Daniel Olmeda, and Nikolay Chumerin. 2022. Contrastive Classification and Representation Learning with Probabilistic Interpretation.

Vahid Aryadoust and Michelle Raquel. 2020. *Quantitative data analysis for language assessment. Volume II, Advanced methods*. Routledge research in language education. Routledge, Abingdon, Oxon ;.

Anton Bakhtin, Sam Gross, Myle Ott, Yuntian Deng, MarcAurelio Ranzato, and Arthur Szlam. 2019. Real or Fake? Learning to Discriminate Machine from Human Generated Text. *arXiv*.

Hongjun Choi, Anirudh Som, and Pavan Turaga. 2020. AMC-Loss: Angular Margin Contrastive Loss for Improved Explainability in Image Classification.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling Instruction-Finetuned Language Models.

Michael T. Cox. 2005. Metacognition in computation: A selected research review. *Artificial Intelligence*, 169(2):104–141. Special Review Issue.

Evan N. Crothers, Nathalie Japkowicz, and Herna L. Viktor. 2023. Machine-Generated Text: A Comprehensive Survey of Threat Models and Detection Methods. *IEEE Access*, 11:70977–71002.

Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. 2019. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699.

Liam Dugan, Daphne Ippolito, Arun Kirubarajan, and Chris Callison-Burch. 2020. RoFT: A tool for evaluating human detection of machine-generated text. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 189–196, Online. Association for Computational Linguistics.

Liam Dugan, Eleni Miltsakaki, Shriyash Upadhyay, Etan Ginsberg, Hannah Gonzalez, DaHyeon Choi, Chuning Yuan, and Chris Callison-Burch. 2022. A feasibility study of answer-agnostic question generation for education. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1919–1926, Dublin, Ireland. Association for Computational Linguistics.

Maël Fabien, Esau Villatoro-Tello, Petr Motlicek, and Shantipriya Parida. 2020. BertAA : BERT fine-tuning for authorship attribution. In *Proceedings of the 17th International Conference on Natural Language Processing (ICON)*, pages 127–137, Indian Institute of Technology Patna, Patna, India. NLP Association of India (NLPAI).

Tiziano Fagni, Fabrizio Falchi, Margherita Gambini, Antonio Martella, and Maurizio Tesconi. 2021. TweepFake: About detecting deepfake tweets. *PLOS ONE*, 16(5):e0251415.

Shuai Gao. 2022. Système de traduction automatique neuronale français-mongol (historique, mise en place et évaluations) (French-Mongolian neural machine translation system (history, implementation, and evaluations) machine translation (hereafter abbreviated MT) is currently undergoing rapid development, during which less-resourced languages nevertheless seem to be less developed). In *Actes de la 29e Conférence sur le Traitement Automatique des Langues Naturelles. Volume 2 : 24e Rencontres Etudiants Chercheurs en Informatique pour le TAL (RECITAL)*, pages 97–110, Avignon, France. ATALA.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2022. SimCSE: Simple Contrastive Learning of Sentence Embeddings.

Tanya Goyal, Junyi Jessy Li, and Greg Durrett. 2022. News summarization and evaluation in the era of gpt-3. *arXiv preprint arXiv:2209.12356*.

Jesus Guerrero and Izzat Alsmadi. 2022. Synthetic Text Detection: Systemic Literature Review.

R. Hadsell, S. Chopra, and Y. LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR06)*, volume 2, pages 1735–1742.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTa: Decoding-enhanced BERT with Disentangled Attention.

Xinlei He, Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. 2024. MGTBench: Benchmarking Machine-Generated Text Detection.

Beizhe Hu, Qiang Sheng, Juan Cao, Yuhui Shi, Yang Li, Danding Wang, and Peng Qi. 2023. Bad actor, good advisor: Exploring the role of large language models in fake news detection. *arXiv preprint arXiv:2309.12247*.

Le Hui, Xiang Li, Chen Gong, Meng Fang, Joey Tianyi Zhou, and Jian Yang. 2019. Inter-class angular loss for convolutional neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3894–3901.

D. Ippolito, D. Duckworth, C. Callison-Burch, and D. Eck. 2020. Automatic detection of generated text is easiest when humans are fooled. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1808–1822. Association for Computational Linguistics.

Niful Islam, Debopom Sutradhar, Humaira Noor, Jarin Tasnim Raya, Monowara Tabassum Maisha, and Dewan Md Farid. 2023. Distinguishing Human Generated Text From ChatGPT Generated Text Using Machine Learning.

Vivek Iyer, Pinzhen Chen, and Alexandra Birch. 2023. Towards Effective Disambiguation for Machine Translation with Large Language Models. *arXiv preprint arXiv:2309.11668*.

Maurice Jakesch, Jeffrey T. Hancock, and Mor Naaman. 2023. Human heuristics for AI-generated language are flawed. *Proceedings of the National Academy of Sciences*, 120(11):e2208839120.

Ganesh Jawahar, Muhammad Abdul-Mageed, and Laks V. S. Lakshmanan. 2020. Automatic Detection of Machine Generated Text: A Critical Survey.

Patrick Juola. 2008. Authorship attribution. *Foundations and Trends® in Information Retrieval*, 1:233–334.

Virapat Kieuvongngam, Bowen Tan, and Yiming Niu. 2020. Automatic text summarization of covid-19 medical research articles using bert and gpt-2. *arXiv preprint arXiv:2006.01997*.

Taehyeon Kim, Eungi Hong, and Yoonsik Choe. 2021. Deep Morphological Anomaly Detection Based on Angular Margin Loss. *Applied Sciences*, 11(14):6545.

Taehyeon Kim, Seho Park, and Kyoungtaek Lee. 2023. Traffic Sign Recognition Based on Bayesian Angular Margin Loss for an Autonomous Vehicle. *Electronics*, 12(14):3073.

Anqi Mao, Mehryar Mohri, and Yutao Zhong. 2023. Cross-Entropy Loss Functions: Theoretical Analysis and Applications.

Colin Martindale and Dean McKenzie. 1995. On the utility of content analysis in author attribution: The Federalist. *Computers and the Humanities*, 29:259–270.

Matthew B. A. McDermott, Lasse Hyldig Hansen, Haoran Zhang, Giovanni Angelotti, and Jack Gallifant. 2024. A Closer Look at AUROC and AUPRC under Class Imbalance.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. DetectGPT: Zero-Shot Machine-Generated Text Detection using Probability Curvature.

Sandra Mitrović, Davide Andreoletti, and Omran Ayoub. 2023. ChatGPT or Human? Detect and Explain. Explaining Decisions of Machine Learning Model for Detecting Short ChatGPT-generated Text. *ArXiv*.

Nadim Nachar. 2008. The Mann-Whitney U: A Test for Assessing Whether Two Independent Samples Come from the Same Distribution. *Tutorials in Quantitative Methods for Psychology*, 4(1):13–20.

OpenAI. 2023. GPT-4 Technical Report.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library.

Dongqi Pu and Vera Demberg. 2023. ChatGPT vs Human-authored Text: Insights into Controllable Text Summarization and Sentence Style Transfer.

Jiameng Pu, Zain Sarwar, Sifat Muhammad Abdullah, Abdullah Rehman, Yoonjin Kim, Parantapa Bhattacharya, Mobin Javed, and Bimal Viswanath. 2022. Deepfake Text Detection: Limitations and Opportunities.

Junaid Qadir. 2022. Engineering Education in the Era of ChatGPT: Promise and Pitfalls of Generative AI for Education. *TechRxiv*.

Yunita Sari. 2018. *Neural and non-neural approaches to authorship attribution.* Ph.D. thesis, University of Sheffield, UK. British Library, EThOS.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, Miles McCain, Alex Newhouse, Jason Blazakis, Kris McGuffie, and Jasmine Wang. 2019. Release Strategies and the Social Impacts of Language Models.

JH Sundjaja, R Shrestha, and K Krishan. 2023. McNemar And Mann-Whitney U Tests. *StatPearls [Internet]*. Updated 2023 Jul 17.

Zhiquan Tan, Yifan Zhang, Jingqin Yang, and Yang Yuan. 2024. Contrastive Learning Is Spectral Clustering On Similarity Graph.

Ruixiang Tang, Yu-Neng Chuang, and Xia Hu. 2023. The Science of Detecting LLM-Generated Texts.

Yi Tay, Dara Bahri, Che Zheng, Clifford Brunk, Donald Metzler, and Andrew Tomkins. 2020. Reverse Engineering Configurations of Neural Text Generation Models.

William John Teahan. 2000. Text classification and segmentation using minimum cross-entropy. In *Content-Based Multimedia Information Access-Volume 2*, pages 943–961.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikuła, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Pier Giuseppe Sessa, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. 2024. Gemma: Open Models Based on Gemini Research and Technology.

Adaku Uchendu, Thai Le, Kai Shu, and Dongwon Lee. 2020. Authorship attribution for neural text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8384–8395, Online. Association for Computational Linguistics.

Adaku Uchendu, Zeyu Ma, Thai Le, Rui Zhang, and Dongwon Lee. 2021. TURINGBENCH: A benchmark environment for Turing test in the age of neural text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2001–2016, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Roman Vygon and Nikolay Mikhaylovskiy. 2021. Learning efficient representations for keyword spotting with triplet loss. In *Speech and Computer: 23rd International Conference, SPECOM 2021, St. Petersburg, Russia, September 27–30, 2021, Proceedings 23*, pages 773–785. Springer.

Jian Wang, Feng Zhou, Shilei Wen, Xiao Liu, and Yuanqing Lin. 2017. Deep Metric Learning with Angular Loss.

Rowan Zellers, Ari Holtzman, Matthew Peters, Roozbeh Mottaghi, Aniruddha Kembhavi, Ali Farhadi, and Yejin Choi. 2021. PIGLeT: Language grounding through neuro-symbolic interaction in a 3D world. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2040–2050, Online. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT.

Appendix A Dataset model breakdown

Dataset	Source	Language Models	Link
TuringBench	TuringBench	GPT-1, GPT-2, GPT-3, GROVER, CTRL, XLM, FAIR, Transformer_XL, XLNET, PPLM	TuringBench
DAIGT-V2	Kaggle	LlaMa2, Darragh_Claude, Mistral7binstruct, Gemma, opt	DAIGT-V2
TuringExtended	Github	Gemma, Flan-T5, GPT	To be added

Table 7: Overview of datasets utilized in the study, detailing dataset name, source, and the language models used to generate text. Note: While not exhaustive, datasets may encompass various iterations of a single model (e.g., LlaMa-7b and Llama-13b).

Appendix B Hyperparameter settings for the DeBERTa model

Hyperparameter	Amended value
`num_train_epochs`	1 - 8
`train_batch_size`	16
`eval_batch_size`	16
`gradient_accumulation_steps`	4
`n_gpu`	1
`max_seq_length`	512
`class_weight`	Equal weighting specified
`early_stopping_patience`	2
`early_stopping_delta`	0.01
`contrastive_loss_weight`	[0.05 - 0.25 - 0.50 -0.75 - 1.00]
`angular_loss_weight`	[0.05 - 0.25 - 0.50 -0.75 - 1.00]
`crossentropy_loss_weight`	[0.05 - 0.25 - 0.50 -0.75 - 1.00]

Table 8: The hyperparameters used in training the DeB-Ang model. Parameter values for the epochs and loss functions varied and the specific values used are detailed in Section 5.

Appendix C Error analysis: linguistic analysis

C.1 Linguistic features extracted

Linguistic category	Feature
Character-level features	uni, bi and tri-grams, word length distribution,
	number counts, text length, average word
	sentence length, casing, type-token ratio
Syntactic features	Part-of-speech tags, dependency tags
Word-level features	Function words

Table 9: Linguistic features extracted from the correctly and incorrectly classified texts for the task of binary machine-generated text detection.

C.2 Linguistic feature groups

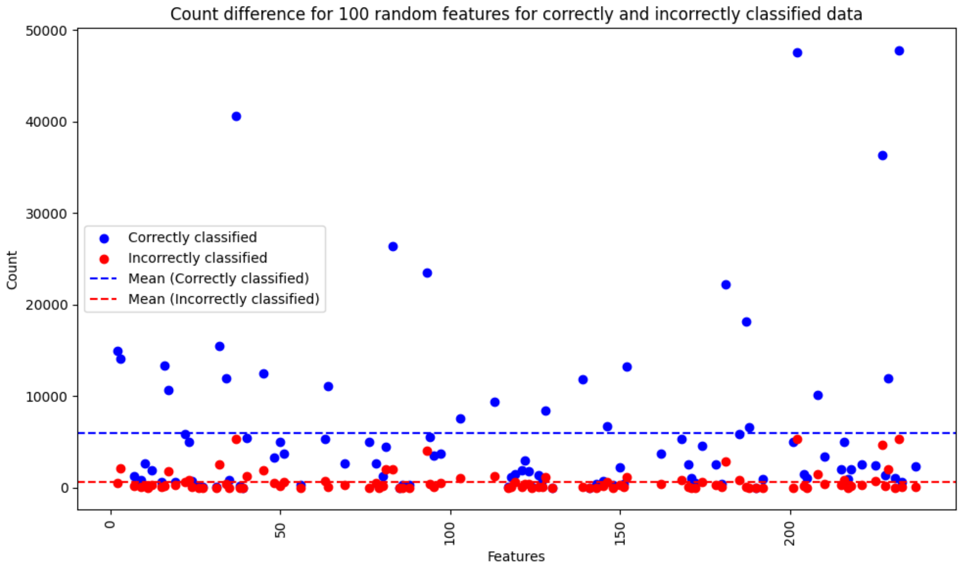

Figure 1: Scatterplot displaying the raw counts and mean feature usage of incorrectly and correctly classified samples.

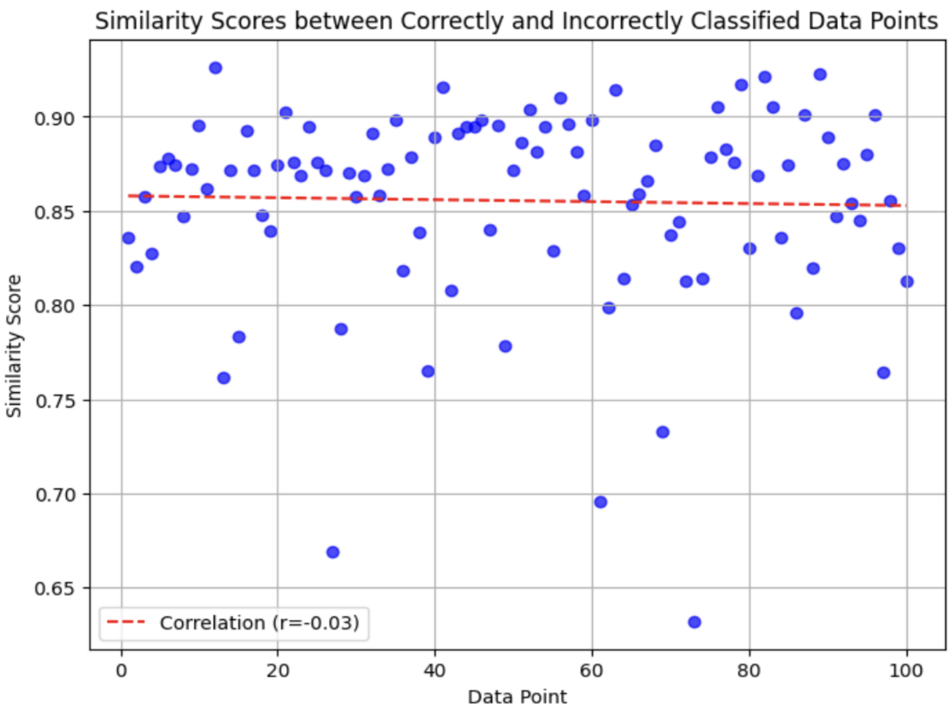

Figure 2: Scatterplot displaying the similarity scores between each correctly and incorrectly classified data samples.