Machine Learning with Applications 22 (2025) 100732

Contents lists available at ScienceDirect

Machine Learning with Applications

journal homepage: www.elsevier.com/locate/mlwa

Combining style and semantics for robust authorship verification

Britt van Leeuwen [a,b,*], Sandjai Bhulai [b], Rob van der Mei [a,b]

[a] Centrum Wiskunde en Informatica (Stochastics group), Science Park 123, Amsterdam, The Netherlands
[b] Vrije Universiteit Amsterdam, Boelelaan 1111, Amsterdam, The Netherlands

ARTICLE INFO

Keywords:
Authorship verification
RoBERTa
Style features
Feature Interaction Network
Pairwise Concatenation Network
Siamese Network

ABSTRACT

Authorship Verification is a key task in Natural Language Processing, essential for applications like plagiarism detection and content authentication. This paper analyzes the use of deep learning models for Authorship Verification, focusing on combining semantic and style features to enhance model performance. We propose three models: the Feature Interaction Network, Pairwise Concatenation Network, and Siamese Network, which aim to determine if two texts are written by the same author. Each model uses RoBERTa embeddings to capture semantic content and incorporates style features such as sentence length, word frequency, and punctuation to differentiate authors based on writing style.

Our results confirm that incorporating style features consistently improves model performance, with the extent of improvement varying by architecture. This demonstrates the value of combining semantic and stylistic information for Authorship Verification. While limitations such as RoBERTa's fixed input length and the use of predefined style features exist, they do not hinder model effectiveness and point to clear opportunities for future enhancement through extended input handling and dynamic style feature extraction.

In contrast to prior studies such as Bevendorff et al., (2020) and Kestemont, et al., (2022), which relied on balanced and homogeneous datasets with consistent topics and well-formed language, our work evaluates models on a more challenging, imbalanced, and stylistically diverse dataset, better reflecting real-world Authorship Verification conditions. Despite the increased difficulty, our models achieve competitive results, underscoring their robustness and practical applicability.

These findings support the value of combining semantic and style features for real-world Authorship Verification.

1. Introduction

1.1. Motivation

Authorship Verification (AV) is a critical task within the field of authorship analysis. Contrary to Authorship Classification (AC), where a model is trained on a predefined set of authors, AV aims to determine whether two texts were written by the same author, even without prior knowledge of that author's writing. Verifying authorship has applications in various fields, including digital forensics (Juola, 2021), literary analysis (Lagutina et al., 2021), and copyright infringement detection (Stein, Lipka, & Prettenhofer, 2011). Beyond these, AV has also been applied in more specialized domains, such as verifying authorship in song lyrics (Yılmaz & Scheffler, 2023) and conducting cross-lingual AV (Misini, Kadriu, & Canhasi, 2024), where stylistic and linguistic variations introduce additional complexity. However, this task remains particularly challenging due to the need for robust generalization beyond known examples, making it an active area of research.

Extensive research has been conducted on AC, as discussed in Section 1.3. The task is framed as a closed-set classification problem, meaning the model is trained and evaluated on a fixed set of known authors. Based on the text, the model recognizes distinct patterns associated with a specific author and uses these style features to learn to differentiate one author from another.

Contrarily, AV involves an open-set problem, where the model encounters authors it has never seen during training. Here lies a considerable challenge: instead of classifying texts into predefined categories, the model must generalize to identify patterns between two texts without having prior knowledge about the authors. This forces AV models to focus on identifying patterns of similarity and divergence between texts, relying heavily on abstract style features. The challenge is magnified when dealing with highly variable writing styles or texts written in different contexts, genres, or time periods.

* Corresponding author at: Centrum Wiskunde en Informatica (Stochastics group), Science Park 123, Amsterdam, The Netherlands.
E-mail addresses: b.e.van.leeuwen@cwi.nl (B. van Leeuwen), s.bhulai@cwi.nl (S. Bhulai), rob.van.der.mei@cwi.nl (R. van der Mei).

https://doi.org/10.1016/j.mlwa.2025.100732
Received 14 May 2025; Received in revised form 18 August 2025; Accepted 8 September 2025
Available online 23 September 2025

Despite these hurdles, significant progress has been made in AV through the integration of advanced Natural Language Processing (NLP) techniques. One promising approach is to combine deep learning models like RoBERTa (Liu et al., 2019) with traditional style features, which capture aspects of an author's writing style such as sentence structure, word choice, and punctuation usage. By embedding both semantic content and style features, models can better identify patterns indicative of authorship, even in the absence of direct author-specific training data. This hybrid approach has the potential to bridge the gap between generalizable pattern recognition and fine-grained stylistic analysis, offering a more robust solution to AV.

In this paper, we propose the combination of RoBERTa-based embeddings with style features to improve the accuracy of AV models. Our goal is to leverage the strengths of deep learning for semantic understanding while incorporating stylistic information to capture author-specific writing traits. By doing so, we aim to address the core challenge of AV: verifying authorship in a way that generalizes across both known and unseen authors. Through careful experimentation, we demonstrate that combining semantic and style features leads to consistently strong performance across multiple neural architectures. This hybrid approach especially improves recall and F1 scores, showing its effectiveness in accurately identifying same-author pairs, particularly in challenging and imbalanced AV scenarios

1.2. Background

Authorship analysis concerns a range of methodologies aimed at understanding and identifying the characteristics of written texts in relation to their authors. This field can be broadly categorized into several sub-disciplines, each regarding different facets of authorship and its implications. Some of these disciplines that are concerned with our task are discussed in this section.

1.2.1. Authorship classification

AC, in machine learning occasionally also referred to as Authorship Attribution (AA), involves a defined set of authors. The task is to classify a text into one of these known categories. This *closed-set classification* is often more straightforward, as it relies on established patterns and features learned during training. However, it lacks the flexibility required for real-world applications where unknown authors may be encountered.

1.2.2. Authorship verification

AV focuses on determining whether two texts are written by the same author. Unlike classification, AV deals with an *open-set* scenario, wherein the model must generalize to assess authorship without prior knowledge of the author's specific writing patterns. This task is critical in various fields such as forensic linguistics, where confirming the authorship of a document can have legal implications.

1.2.3. Author profiling

Author profiling seeks to deduce information about an author based on their writing style and language usage. This can include demographic information such as age, gender, or education level, inferred from textual features. This analysis can be particularly useful in marketing and social-media analysis, where understanding the audience's preferences can drive targeted content creation.

1.2.4. Other

Additionally, the field also touches on phenomena such as obfuscation, plagiarism, multi-author detection (Zamir et al., 2024), and style imitation. These topics delve into how texts can be manipulated or altered, intentionally or otherwise, to obscure their original authorship or to mimic the style of another author. Each of these areas presents unique challenges and necessitates specific analytical approaches.

These various forms of authorship analysis are not mutually exclusive; they often intersect and inform one another. For instance, advancements in AC can enhance AV methodologies, and vice versa. The rapid evolution of computational methods, particularly in NLP and machine learning, has significantly propelled the effectiveness and accuracy of these analyses.

Given the diverse applications and implications of authorship analysis, it is important to understand its different areas. As we direct our approach in AV, we must take advantage of new technologies while also tackling the complexities of language and authorship in today's fast-changing digital world.

1.3. Related work

AV faces several key challenges. AV models focus on consistencies between texts without prior knowledge of the authors. In addition, sufficient writing samples are crucial as writing styles can vary significantly across different contexts, genres, and time periods (Stamatatos et al., 2023). Moreover, to prevent overfitting in AV, it is necessary to focus on the author's style and not fixate solely on the genres and topics of the texts.

Recent research has focused on applying NLP techniques to address these challenges, leading to notable advancements in model accuracy, robustness, and generalization across diverse AV tasks

1.3.1. Feature based approaches

Traditional AV methods rely on handcrafted linguistic features, such as lexical choices, syntactic structures, and stylistic markers. These features capture distinctive writing habits and have been commonly used in classical machine learning models like Support Vector Machines (SVMs) and Random Forests. However, feature engineering can be time-consuming and may not generalize well across different datasets.

Style techniques have been widely used for AV by analyzing linguistic styles and writing characteristics. While these methods perform well on long texts, they face significant challenges with short documents, especially in cases involving a large number of authors (Brocardo et al., 2013a).

Character n-grams are commonly used in both AV and AC (Alsanoosy, Shalbi, & Noor, 2024; Brocardo et al., 2013b; Castro et al., 2015; Potha & Stamatatos, 2017, 2020). However, such methods can be questionable for AV, because overlapping character sequences are associated with content words. As a result, these methods may focus more on the topic and genre of the text rather than the authors' style, increasing the risk of trained models misclassifying texts with a similar topic to being written by the same author, even though the authors are different.

In AC, a study by Alsanoosy et al. (2024) investigated various feature extraction techniques, such as Bag-of-Words (BoW), Term Frequency–Inverse Document Frequency (TF–IDF), and n-grams, applied to tweets. Like other studies (Abbasi et al., 2022; Abedzadeh, Ramezani, & Fatemi, 2021), the TF–IDF method yielded the most desirable performance. However, TF–IDF treats words as independent entities and does not capture the context in which they appear. This is a major drawback for AV, where writing style and contextual patterns are crucial. Moreover, TF–IDF works best with longer texts, whereas AV usually concerns shorter texts. Nonetheless, their findings highlight the value of combining lexical features with contextual representations, supporting the use of deep embeddings and style features in AV.

1.3.2. Deep learning and representation learning

Recent work has increasingly shifted towards deep learning models, particularly those using neural embeddings to capture textual similarities. Transformer-based models such as BERT, RoBERTa, and their variations have been employed to encode texts into dense vector representations (Jones, Nurse, & Li, 2022; Manolache et al., 2021; Tyo, Dhingra, & Lipton, 2021, 2022). These representations can then be

compared using similarity measures or processed through architectures like Siamese Networks (SN). These models leverage contextualized word embeddings to capture subtle stylistic patterns without the need for extensive manual feature extraction, making them well-suited for AV tasks.

Tyo et al. (2021) adopt a SN initialized with pretrained BERT encoders, with the learning objective designed to map texts by the same author closer in the embedding space and texts by different authors farther apart. This methodology demonstrates the power of pretrained language models in AV, as well as the potential for distance-based loss functions to guide the network towards meaningful representations.

Recent work has explored the combination of deep learning-based embeddings with style-based features for authorship-related tasks. One approach (Wang & Iwaihara, 2021) integrates RoBERTa with a Convolutional Neural Network (CNN) to extract contextualized text representations, while also incorporating a writing style module that captures stylistic patterns. These representations are fused and used for AA, demonstrating the benefits of combining content-based and style-based features. Additionally, a similarity-based approach is employed through cosine distance computations, highlighting the effectiveness of measuring stylistic consistency across texts. These findings suggest that leveraging both linguistic representations and style features can enhance authorship analysis, motivating further exploration in this direction.

1.3.3. Hybrid and contrastive learning approaches

Some studies combine traditional feature-based methods with neural embeddings to enhance model performance. Hybrid models integrate style features with deep learning representations to leverage the strengths of both approaches. Additionally, contrastive learning techniques, which train models to differentiate between similar and dissimilar text pairs, have been explored to improve generalization across unseen authors.

Recent work by Zamir et al. (2024) proposed style features and RoBERTa for detecting author changes in multi-authored documents, focusing on tasks like single and multiple author-switching detection. While their work aims at document provenance and authentication, our approach leverages similar techniques for AV, demonstrating their versatility in analyzing writing style.

Another study by Najafi and Tavan (2022) applied a deep neural network approach to AV, leveraging the T5 language model along with CNNs and an attention mechanism to capture stylistic and semantic features. While their model achieved high accuracy on a manually created test set, its performance on the official PAN dataset highlights the challenge of generalization in AV—an issue our work also aims to address.

PAN shared tasks

At PAN 2020, a shared task of AV was tackled (Bevendorff et al., 2020). Given two fanfiction texts, the goal was to determine if they were written by the same author. While research in AV with PAN 2020 has achieved high performance metrics (Boenninghoff, Nickel, & Kolossa, 2021; Kipnis, 2020; Weerasinghe & Greenstadt, 2020), these studies rely on well-structured datasets where texts are long, grammatically correct, and often centered around a consistent topic per author. Such datasets can make the AV task easier, as models may partially rely on content-related features such as topic and genre rather than purely on stylistic elements.

Additionally, these studies use artificially balanced datasets with an equal number of positive and negative pairs, not accounting for the real-world challenge where generally positive pairs occur substantially less than negative pairs.

At PAN 2022, another shared task of AV was tackled (Stamatatos et al., 2022). Given two texts from different discourse types, the goal was to determine if they were written by the same author. The dataset is well-structured, consisting only of native English speakers from the same age group, with an equal 50–50 split between positive and negative pairs. PAN 2022 inspired multiple research efforts (Crespo-Sanchez et al., 2022; Huang et al., 2022; Konstantinou, Li, & Zinonos, 2022; Lei et al., 2022; Martinez-Galicia et al., 2022; Najafi & Tavan, 2022; Ye et al., 2022) to tackle the task, with the highest overall performance reaching 0.600 with an F1 score of 0.669 and an AUROC of 0.546. In this shared task, the graph-based approach proved to perform better for longer texts like essays, whereas for shorter texts like emails, business memos, and text messages, the pre-trained language model approaches performed better. The shared task that PAN 2022 provided was challenging, as it involved texts written on different platforms. However, the dataset is well-structured, providing a suitable base to achieve good performance. Still, the overall results of the different methods leave considerable room for improvement. Notably, the results showed that pre-trained language models performed better on shorter texts, suggesting that the assumptions and methods used in PAN 2020 may not generalize well to more diverse and real-world data.

Overall, the PAN 2020 and PAN 2022 shared tasks provide a strong foundation, but show to be not ready for real-life applications yet. A logical next step towards improving the generalizability of AV for real-life scenarios is to combine traditional style models with pre-trained language models.

1.4. Our contribution

The contribution of this paper is three-fold. First, in contrast to the current state-of-the-art research, this study utilizes a more diverse and noisy dataset of blog texts, where authors write across multiple topics and genres. This setup ensures that our models primarily capture style features rather than topic-driven patterns, making them more robust for real-world applications where genre consistency cannot be assumed.

Second, unlike studies that use artificially balanced datasets with an equal number of positive and negative pairs, we adopt a more realistic 20–80 distribution, where same-author pairs are significantly less frequent. This imbalance better reflects real-world AV scenarios, where the vast majority of comparisons involve texts from different authors. While this setting is inherently more challenging and may lead to lower absolute performance scores, it provides a more reliable evaluation of a model's effectiveness in practical applications.

Third, our results demonstrate that incorporating style features significantly improves model performance, reinforcing the importance of linguistic style in authorship analysis beyond content-based cues.

2. Data

2.1. Data set

2.1.1. Data extraction

The dataset used in this study originates from a collection of blogs gathered from Blogger.com in August 2004 by Schler et al. (2006). It contains 681,284 text posts written by 19,320 authors, offering a large-scale and diverse corpus for AV research. Each blog includes all posts from its inception up to the time of collection. The original dataset comprised over 71,000 blogs, filtered to retain only those with at least 200 common English words and available author metadata such as gender and, in some cases, age. To support demographic analyses, a balanced sub-corpus of 37,478 blogs was created, with an equal number of male and female authors across different age groups.

The analysis of text lengths across the dataset, as shown in Figs. 1 and 2, both in terms of words and characters, reveals a highly skewed distribution, with most texts falling within a relatively narrow range and a long tail of extreme outliers. While the majority of documents are under 250 words or 3000 characters, there are numerous outliers that significantly exceed these lengths, with the longest text reaching almost 120,000 words. To visualize the distribution of text lengths more effectively, we capped the y-axis at 2000 words and 50,000 characters, as the presence of extreme outliers distorted the interpretability of the original boxplots.

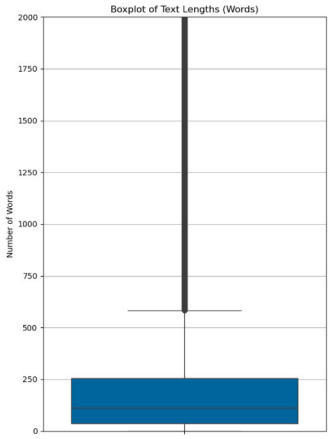

Fig. 1. Boxplot of text length in number of words used.

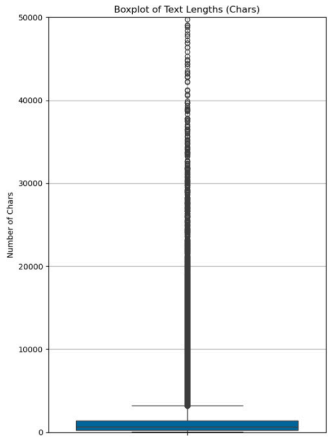

Fig. 2. Boxplot of text length in number of characters used.

2.1.2. Challenges

A key challenge in this study arises from the use of RoBERTa, which has a maximum input length of 512 tokens. This constraint can lead to truncation of longer texts, potentially resulting in information loss. While newer transformer models such as Longformer and BigBird address this limitation by supporting longer input sequences, RoBERTa was chosen for its proven performance and compatibility with our architecture. Another significant challenge is the risk of overfitting and data leakage. Since the dataset consists of a limited number of authors, the model could inadvertently learn patterns specific to the training data rather than generalizable style features, reducing its effectiveness on unseen data.

2.1.3. Comparison to other data

The BlogAuthorshipCorpus (Schler et al., 2006) is significantly noisier than state-of-the-art AV datasets such as those from the PAN shared tasks. Unlike PAN datasets, which are carefully curated with balanced author distributions, consistent text lengths, and controlled topical variation, the BlogAuthorshipCorpus consists of informal, user-generated blog posts with high variability in style, topic, and quality. This includes frequent spelling errors, inconsistent punctuation, and wide intra-author stylistic differences. Additionally, the dataset lacks topic control, leading to strong topical signals that may confound stylistic analysis, and includes imbalanced author contributions, further complicating model training and evaluation.

2.2. Dataset preprocessing

To ensure a fair evaluation, we constructed pairs from the blog texts while strictly maintaining author separation between the training and test sets. This guarantees that no author appears in both sets, preventing information leakage and ensuring that the model is evaluated on truly unseen authors. Additionally, to mitigate overfitting, each text post is included in at most one pair, reducing the likelihood that the model memorizes specific examples rather than learning broader stylistic patterns.

3. Model description

3.1. Proposed models

This study employs three different deep neural networks to address the AV problem: a SN, a Feature Interaction Network (FIN), and a Pairwise Concatenation Network (PCN). Each network is designed to implement different mechanisms for comparing text pairs, emphasizing varying aspects of similarity and feature integration. Below, we provide an overview of each architecture.

Siamese network

The SN architecture consists of two identical subnetworks f_θ, parameterized by shared weights θ, which process the two input texts $\mathbf{x}_1$ and $\mathbf{x}_2$ independently. The network produces embeddings $\mathbf{h}_1 = f_\theta(\mathbf{x}_1)$ and $\mathbf{h}_2 = f_\theta(\mathbf{x}_2)$, where $\mathbf{h}_1, \mathbf{h}_2 \in \mathbb{R}^d$, and d represents the dimensionality of the output embeddings, indicating the number of features each embedding vector contains.

The similarity between the embeddings is then computed using a distance metric D, such as cosine similarity or Euclidean distance:

$$D(\mathbf{h}_1, \mathbf{h}_2) = \|\mathbf{h}_1 - \mathbf{h}_2\|_2 \quad \text{or} \quad \cos(\mathbf{h}_1, \mathbf{h}_2) = \frac{\mathbf{h}_1 \cdot \mathbf{h}_2}{\|\mathbf{h}_1\|\|\mathbf{h}_2\|}.$$

The output of D is passed through a decision layer to predict whether the texts share the same author. The shared weights θ ensure that both inputs are projected into the same feature space, enhancing the model's ability to generalize patterns of similarity.

Feature Interaction Network

The Feature Interaction Network (FIN) explicitly models pairwise interactions between features extracted from the two texts. Let $\mathbf{h}_1, \mathbf{h}_2 \in \mathbb{R}^d$ represent the embeddings of $\mathbf{x}_1$ and $\mathbf{x}_2$, respectively, obtained from a shared encoder f_θ. The interaction is modeled using a pairwise interaction mechanism, such as the outer product or attention mechanisms.

For example, an element-wise interaction can be expressed as:

$$\mathbf{z} = \phi([\mathbf{h}_1; \mathbf{h}_2; \mathbf{h}_1 \odot \mathbf{h}_2]),$$

where $[\cdot;\cdot]$ denotes vector concatenation, $\odot$ represents element-wise multiplication, and ϕ is a transformation (e.g., a feedforward neural network).

Alternatively, an attention-based mechanism may compute the interaction matrix $\mathbf{I} \in \mathbb{R}^{d \times d}$:

$$\mathbf{I} = \text{softmax}\left(\frac{\mathbf{h}_1\mathbf{W}_q \cdot (\mathbf{h}_2\mathbf{W}_k)^\top}{\sqrt{d}}\right),$$

where $\mathbf{W}_q$ and $\mathbf{W}_k$ are learnable projection matrices. These interactions allow the network to capture detailed feature relationships, leading to richer representations.

Pairwise concatenation network

In the PCN, the embeddings of the two texts $\mathbf{h}_1$ and $\mathbf{h}_2$, obtained from a shared or independent encoder, are concatenated into a single vector:

$$\mathbf{z} = [\mathbf{h}_1; \mathbf{h}_2],$$

where $\mathbf{z} \in \mathbb{R}^{2d}$.

The concatenated vector is then passed through fully connected layers:

$$\mathbf{o} = \sigma(\mathbf{W}_2\,\mathrm{ReLU}(\mathbf{W}_1\mathbf{z} + \mathbf{b}_1) + \mathbf{b}_2),$$

where $\mathbf{W}_1$ and $\mathbf{W}_2$ are weight matrices, $\mathbf{b}_1$ and $\mathbf{b}_2$ are biases, σ is the activation function (e.g., sigmoid), and $\mathbf{o}$ is the output. The model directly learns the relationship between the two texts from their joint representation.

By incorporating these mathematical formulations, the networks emphasize different aspects of feature extraction, interaction, and relationship modeling, enabling a comprehensive evaluation of their suitability for the AV task.

3.2. Style features

To complement the semantic representations, we extracted a set of style features designed to reflect writing style independently of topic. We extracted the Flesch reading ease score (Flesch, 1948) to measure how readable a text is. Average sentence length was used to capture syntactic complexity, while counts of nouns and verbs reflected grammatical tendencies. We also calculated the stopword ratio to account for function word usage and the average word length as a proxy for lexical sophistication. All texts were processed using SpaCy for accurate tokenization and sentence segmentation. The resulting features were normalized using a standard scaler to allow comparison across features. While we did not conduct formal feature importance analyses such as SHAP in this study, our primary aim was not to optimize a handcrafted feature set, but to evaluate whether even simple, interpretable style features can provide complementary information when combined with deep semantic embeddings. This choice of handcrafted features also reflects a deliberate preference for interpretability and computational efficiency, which is especially important in resource-constrained or transparency-critical contexts such as forensic linguistics. To integrate stylistic and semantic information, the style features were concatenated with the RoBERTa embeddings at the input level, forming a unified representation for each text. This allowed for clear isolation of the contribution of traditional style features when combined with semantic representations.

3.3. RoBERTa

Adding semantic information helps the model capture meaningful differences in how authors express themselves. While style features are valuable for capturing consistent stylistic patterns, they do not fully account for semantic content. Although most sentence embedding models, including RoBERTa, are designed to focus on the semantic meaning of sentences and generally de-emphasize function words such as stopwords, semantic embeddings can still indirectly reflect stylistic consistencies—especially when combined with style features or fine-tuned for the authorship verification task. It is important to clarify that RoBERTa primarily captures what is said rather than how it is said. Therefore, the combination of semantic embeddings with explicit style features allows the model to better leverage both the content and style dimensions necessary for robust authorship verification across different topics.

RoBERTa (Robustly Optimized BERT Pretraining Approach) is an advanced transformer-based language model introduced by Liu et al. (2019). It builds upon BERT by optimizing its training process through dynamic masking, removal of the next sentence prediction objective, and training on larger datasets with increased batch sizes. These improvements enhance RoBERTa's ability to capture contextual and semantic relationships in text, making it particularly effective for various NLP tasks, including AV. By leveraging pre-trained representations from RoBERTa, AV models can benefit from deep contextualized embeddings, allowing for more robust comparisons of writing style and linguistic patterns.

RoBERTa can only be used on a maximum number of characters/words. By taking only that number of words from the texts, the model generalizes on any length of text, short and long. However, a bit of information might get lost from the long texts.

3.4. Loss function

The Weighted Binary Cross-Entropy (WBCE) is an adaptation of the standard Binary Cross-Entropy (BCE) loss function, designed to address issues arising from class imbalance in binary classification tasks. In many real-world scenarios, the distribution of classes is skewed, meaning that one class may be significantly more frequent than the other. This can lead to models that perform well on the majority class but poorly on the minority class. WBCE seeks to mitigate this issue by assigning different weights to the two classes. Consequently, the introduction of weights enhances model performance. WBCE allows the model to achieve better recall and precision for the underrepresented class, improving overall classification performance in imbalanced scenarios. The WBCE loss function ultimately leads to better generalization and predictive accuracy concerning our task.

The standard BCE loss function is defined as:

$$\mathrm{BCE}(y, \hat{y}) = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)\right], \tag{1}$$

where:

- y_i is the true label (0 or 1).
- $\hat{y}_i$ is the predicted probability of the positive class (1).
- N is the total number of samples.

This loss function effectively measures the performance of a classification model whose output is a probability value between 0 and 1. To employ WBCE, we introduce weights w_0 and w_1 to penalize the model differently for misclassifying the two classes. The WBCE loss is then defined as:

$$\begin{aligned}\mathrm{WBCE}(y, \hat{y}) = -\frac{1}{N}\sum_{i=1}^{N}\Big[&w_1 \cdot y_i \log(\hat{y}_i) \\ &+ w_0 \cdot (1 - y_i)\log(1 - \hat{y}_i)\Big],\end{aligned} \tag{2}$$

where:

- w_1 is the weight for the positive class (1).
- w_0 is the weight for the negative class (0).

4. Experimental design

In this section, we outline the experimental setup for the networks evaluated in this study, including choices for hyperparameters, loss functions, and evaluation metrics. We use three neural network architectures: FIN, PCN, and SN. To ensure a fair and accurate comparison between the models, we conducted a separate grid search for each model. This helped us find the best hyperparameter settings, such as learning rate, dropout rate, batch size and number of epochs, for each model individually. This individualized tuning accounts for the distinct optimization dynamics of each architecture, ensuring stable training and fair performance comparison. Models were also tuned separately for configurations with and without style features, as the inclusion of style-based information alters input dimensionality and learning behavior. This approach ensures that all models are evaluated under conditions that reflect their true potential, avoiding bias due to suboptimal training settings. The final architectural and training choices for all six configurations are summarized in Table 1.

Table 1
Optimal hyperparameters per model configuration, selected through grid search.

Model	Learning rate	Dropout	Batch size	Epochs
Feature Interaction (No style)	exp(−4)	0.01	16	15
Feature Interaction (Style)	exp(−3)	0.01	32	15
Pairwise Concatenation (No style)	exp(−4)	0.1	16	10
Pairwise Concatenation (Style)	exp(−4)	0.1	32	15
Siamese Network (No style)	exp(−3)	0.01	16	10
Siamese Network (Style)	exp(−3)	0.01	16	15

4.1. Siamese network

The SN is a deep learning model designed for comparing two input texts. It utilizes a shared base network that processes both inputs in parallel. The base network consists of dense layers with ReLU activations, followed by batch normalization and dropout to mitigate overfitting. The output of both inputs is a fixed-size representation that is compared using cosine similarity, calculated as the dot product between the processed outputs of the two inputs. The similarity score is then scaled, using a Lambda layer (Chollet et al., 2015). The final output layer uses a sigmoid activation function to classify whether the two inputs are from the same author.

Hyperparameters: The SN is trained using the Adam optimizer with a learning rate of 1×10^{-3} for both the model with and without style features. The model with style features is trained for 15 epochs, while the model without style features is trained for 10 epochs. Both models use a batch size of 16. To address class imbalance, class weights are calculated and applied during training. Dropout is applied with a rate of 0.01 in both models' base networks to prevent overfitting. Additionally, L2 regularization with a coefficient of 0.0001 is used to encourage weight sparsity and mitigate overfitting.

Model Architecture: The Siamese Network (SN) consists of a base network $f(\mathbf{x})$, which maps the input text embeddings to a latent space. The base network is composed of three fully connected layers:

- A dense layer with 256 units, ReLU activation, batch normalization, dropout, and L2 regularization.
- A dense layer with 128 units, ReLU activation, batch normalization, dropout, and L2 regularization.
- A dense layer with 64 units, ReLU activation, batch normalization, dropout, and L2 regularization.

The two input embeddings $\mathbf{x}_1$ and $\mathbf{x}_2$ are processed independently through the base network, yielding embeddings $f(\mathbf{x}_1)$ and $f(\mathbf{x}_2)$. The network then computes the cosine similarity between these embeddings using the dot product:

$$\text{Cosine Similarity} = \frac{f(\mathbf{x}_1) \cdot f(\mathbf{x}_2)}{\|f(\mathbf{x}_1)\|_2 \|f(\mathbf{x}_2)\|_2}.$$

The output is a similarity score, which is transformed to a range of [0, 1] using a Lambda layer and is used for classification.

4.2. Feature interaction network

The network operates by taking two input texts, transforming them into embeddings, and performing several element-wise operations (subtraction, absolute difference, and multiplication). These interactions are concatenated and passed through a series of fully connected layers with ReLU activations. Dropout is applied to the hidden layers to prevent overfitting. The output layer uses a sigmoid activation function to predict whether the two texts come from the same author. The network is optimized using a custom weighted BCE loss to handle class imbalance effectively.

Hyperparameters: The FIN is trained with a learning rate of 1×10^{-4} (0.0001) using the Adam optimizer, which adapts the learning rate during training and is suitable for minimizing complex, non-convex loss functions. The model is trained for a total of 15 epochs, with a batch size of 16, balancing training time and memory usage. Dropout is applied at a rate of 0.01 after each dense layer to prevent overfitting. The model is trained without style features. For the version of the FIN with style features, the learning rate is set to 1×10^{-3} (0.001). The model is trained with a batch size of 32 for 15 epochs, using a dropout rate of 0.01 after each dense layer to avoid overfitting.

Model Architecture: The model consists of two inputs, each representing one text in the pair, which are processed using simple arithmetic operations to model their interaction. Specifically, we compute the element-wise difference, absolute difference, and multiplication of the input vectors. These interaction features are then concatenated and passed through three fully connected layers with 4096, 256, and 64 units, respectively. Each layer uses the ReLU activation function, followed by dropout to regularize the model. The final output layer has a sigmoid activation to produce a binary classification prediction. No attention mechanisms are used; the model relies entirely on these arithmetic operations to capture both symmetric and asymmetric relationships between the paired inputs.

4.3. Pairwise concatenation network

The PCN operates by taking two input texts, transforming them into embeddings, and concatenating these embeddings along the feature dimension. The concatenated representation is then passed through three fully connected layers, each with ReLU activation. Dropout is applied after each layer to mitigate overfitting. The final output layer uses a sigmoid activation function for binary classification, predicting whether the two texts come from the same author.

Hyperparameters: The model is trained using the Adam optimizer with a learning rate of 1×10^{-4}, both with and without style features. The model with style features is trained for 15 epochs with a batch size of 32, while the model without style features is trained for 10 epochs with a batch size of 16. To address class imbalance, class weights are calculated and applied during training. Dropout is applied with a rate of 0.1 in both models.

Model Architecture: The network begins with two input layers, $input_a$ and $input_b$, each of shape (input_shape). These embeddings are concatenated along the last axis, forming a tensor of shape ($2 \times$ input_shape). The concatenated vector is passed through three fully connected layers with 4096, 256, and 64 units, respectively. Each of these layers uses the ReLU activation function, and dropout is applied after each layer to prevent overfitting. The final output layer uses a sigmoid activation function to perform binary classification.

The model is compiled using the Adam optimizer, and a custom WBCE loss function addresses class imbalance during training.

4.4. Training and validation

All networks are trained and evaluated on the same training and test datasets. The training dataset is used to fit the models, while the test dataset is used to assess generalization. The models are evaluated at the end of each epoch on the test set, and the best performing model based on validation accuracy is selected.

4.5. Loss function

To address class imbalance, the network is trained with a custom weighted BCE loss function, designed to optimize for the pairwise similarity task while addressing class imbalance. The WBCE loss is shown in Eq. (2).

We apply a fixed weighting scheme where misclassifying the minority class is penalized more heavily than the majority class. We apply the WBCE Loss where false negatives receive a weight of 10, while false positives receive a weight of 1. This encourages the model to focus more on correctly identifying the minority class while still considering the majority class.

Table 2
Final test set performance across all models with and without style features.

Metric	Feature interaction (No style)	Feature interaction (Style)	Siamese network (No style)	Siamese network (Style)	Pairwise concatenation (No style)	Pairwise concatenation (Style)
Loss	1.4631	**1.4409**	1.5475	1.4745	1.4521	1.4526
Accuracy	0.8263	0.8281	0.8383	**0.8472**	0.8311	0.8292
Precision	0.7183	0.7058	0.7326	0.7402	**0.7621**	0.7049
Recall	0.6681	0.6961	0.7107	0.7265	0.6121	**0.7052**
F1 score	0.6791	0.6906	0.7085	0.7228	0.6632	**0.6942**
AUC	0.7995	0.8062	0.8114	0.8260	0.7879	**0.8015**

4.6. Evaluation metrics

The model is evaluated using a variety of metrics to provide a comprehensive performance evaluation:

- **Accuracy:** Measures the overall correctness of the model's predictions.
- **Precision:** The proportion of true positives among all predicted positives.
- **Recall:** The proportion of true positives among all actual positives.
- **F1 Score:** The harmonic mean of precision and recall.
- **AUC-PR:** The area under the precision–recall curve, which provides a more informative evaluation in imbalanced classification problems.

5. Results

We evaluated three models for the AV task: FIN, SN, and PCN. Each model was tested with and without the inclusion of style features, which capture authorial writing patterns beyond the semantic content encoded in transformer embeddings. Below, we report and analyze the final test results for all model variants.

5.1. Performance measures

We report six evaluation metrics for each model variant: loss, accuracy, precision, recall, F1 score, and area under the ROC curve (AUC). These metrics are presented in Table 2, allowing for a detailed comparison of performance between versions with and without style features. Each column corresponds to one of the three model architectures — FIN, SN, and PCN — evaluated with and without style information. The results provide a comprehensive view of the models' ability to distinguish between same-author and different-author text pairs, and highlight the effects of including surface-level writing features alongside contextual embeddings.

In addition to summary statistics, we visualize classifier performance using receiver operating characteristic (ROC) curves for all model variants in Fig. 3. The ROC curves provide a comparative view of true positive rate versus false positive rate across different decision thresholds. As shown, all models outperform the random baseline, with the SN with style model achieving the highest overall performance (AUC = 0.89), followed by the SN and FIN models (both AUC = 0.88). Incorporating style features consistently improves or maintains performance across all architectures.

To monitor model learning behavior, we tracked accuracy, recall, precision, F1 score, and AUC across training epochs. Training and validation curves for each metric are provided in Appendix D, illustrating convergence patterns and the effects of style features. These plots serve to complement the final test results in Table 2.

For a detailed view of the classification outcomes, the confusion matrices for all models — both with and without style features — are included in Appendix B. These matrices illustrate differences in false positive and false negative rates across models and show how the inclusion of style features affects prediction balance.

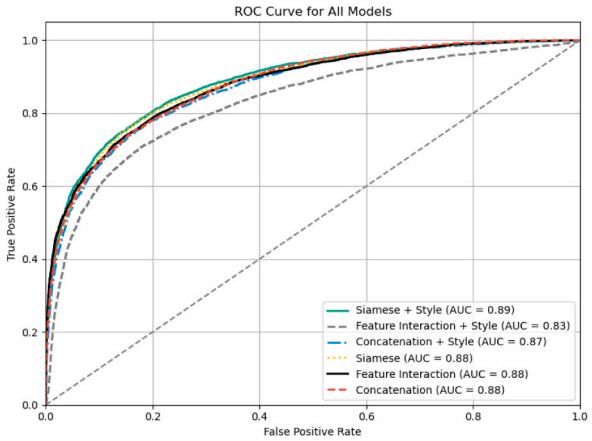

Fig. 3. ROC curves comparing all model variants, with and without style features.

5.2. Comparison to (PAN) baselines

Our models were evaluated in a more challenging setting than the ones used in PAN 2020 and PAN 2022, where datasets were artificially balanced, texts were well-structured, and authors often wrote about consistent topics in grammatically correct English. In contrast, our dataset contains platform-agnostic, stylistically diverse texts with realistic class imbalance, closely resembling practical AV scenarios.

The PAN 2020 shared task (Bevendorff et al., 2020) featured fanfiction texts and offered both a large and small dataset for training. The top-performing model (Boenninghoff et al., 2021) achieved an AUC of 0.969 and an F1-score of 0.936. However, this setting favors high performance due to homogeneous text structure, topic consistency, and ample training data. In our case, the best model — PCN with style features — achieved an AUC-PR of 0.7953 and an F1-score of 0.6782, reflecting solid performance in a much less constrained and noisier environment.

Compared to PAN 2022 (Stamatatos et al., 2022), which introduced discourse variation across platforms but still maintained balance and demographic uniformity, our models also outperform the top results. The best F1-score at PAN 2022 was 0.669, with an AUROC of 0.546. Our models, trained on heterogeneous data, achieved a higher F1-score and precision–recall AUC, emphasizing improved real-world applicability.

5.3. Performance vs. text length

To investigate whether the models exhibit any bias related to input text lengths, we computed Pearson correlation coefficients between three variables: average input length (`avg_len`), absolute length difference between the two texts (`length_diff`), and whether the prediction was correct (`correct`). The tables in Fig. A.1 present these correlation matrices for the six different models. Across all models, the correlations between the correctness of a prediction and both length-based metrics are consistently close to zero, indicating no practical dependence on either input size or imbalance in length.

5.4. Topic similarity analysis across models

Specifically, we investigated the relationship between topic similarity scores and the models' mean prediction probabilities. Topic similarity was computed using two approaches: (1) LDA-based similarity, where documents were represented by their topic distributions obtained from a Latent Dirichlet Allocation model, and similarity between document pairs was measured using cosine similarity on these distributions; and (2) RoBERTa-based similarity, where sentence embeddings from a pre-trained RoBERTa model were used, and similarity between document pairs was computed via cosine similarity on these embeddings.

Across all six model variants, LDA-based topic similarity revealed only weak associations with model outputs, as shown in Table Table C.1. The highest Spearman correlation was observed for the Siamese network without style features ($\rho \approx 0.425$, $R^2 \approx 0.164$), while the lowest correlations occurred for pairwise concatenation and feature interaction models with style features ($\rho \approx 0.256$–0.275, $R^2 \approx 0.099$–0.102). In general, including style features slightly reduced the correlation with topic similarity, suggesting that stylistic cues contribute to predictions beyond topical content.

RoBERTa-based similarity showed consistently low correlations for all models ($\rho \leq 0.058$, $R^2 \leq 0.115$), reflecting saturation near high similarity values. This indicates that semantic embeddings in this context do not meaningfully distinguish between model prediction scores.

Overall, these results indicate that topical similarity contributes only modestly to model predictions, and that higher predictions are not simply a result of overlapping topics. The weak correlation across models implies that the models, particularly when incorporating style features, rely more heavily on other textual characteristics — likely stylistic or deeper semantic patterns — rather than surface-level topical overlap. This provides confidence that the models are capturing nuanced signals relevant to authorship verification rather than relying solely on topic content.

For further clarity, detailed plots of the topic similarity distributions, along with a table summarizing the corresponding values, are provided in Appendix C.

6. Discussion

This study set out to address the challenges of AV in real-world scenarios by evaluating multiple model architectures and exploring the value of incorporating style features alongside contextual embeddings. Unlike previous AV benchmarks, such as those demonstrated at PAN 2020 and PAN 2022, which focused on well-curated datasets with controlled topics, consistent grammar, and balanced class distributions, our approach dealt with a more complex and realistic setting. Texts originated from multiple platforms and exhibited considerable variation in discourse type, writing style, and structure. Moreover, the class distribution was intentionally imbalanced to reflect the rarity of positive (same-author) cases in real-life applications.

Our findings highlight a couple of main contributions. First, all three evaluated models — FIN, SNs, and PCN — achieved solid performance in a cross-platform AV setting. Among these, the SN model yielded the highest accuracy (0.8472), indicating strong overall predictive performance. However, the FIN model with style features achieved the best results in F1 score (0.7228), recall (0.6961), and AUC (0.8260), highlighting its strength in correctly identifying positive matches. These findings suggest that while the SN architecture excels at general classification, the FIN model, when enhanced with style cues, is particularly effective at capturing author-specific patterns crucial for accurate AV.

Second, the integration of style features consistently improved recall across all models, which is particularly relevant in operational scenarios where minimizing false negatives is important. Surface level features, such as sentence length, lexical variety, and punctuation use, provide complementary information that enhances the discriminative power of contextual embeddings, supporting the use of hybrid models that combine deep language representations with explicit stylistic indicators. Overall, models incorporating style features performed better across most metrics, achieving the highest accuracy, recall, F1 score, and AUC. This suggests that style features enhance the model's ability to capture author-specific characteristics, which are essential for AV tasks. While precision was slightly better in non-style models, recall, being more critical for identifying true matches, was significantly higher in style-based models. The improved recall and F1 score in these models make them more effective for AV, as they are better at correctly identifying matches and minimizing false negatives.

Third, while benchmark datasets like PAN 2020 and PAN 2022 report higher results in more controlled settings, our models are evaluated in a more challenging and realistic scenario. The best-performing model achieves an F1 score of 0.7228 and an AUC score of 0.8260, demonstrating improved robustness compared to PAN 2022 and a stronger trade-off between precision and recall. These results underscore the practical applicability of the proposed methods to real-world AV tasks.

Finally, this study outlines a flexible framework for building AV systems that generalize across platforms, domains, and unknown authors. By reframing the task as one of measuring stylistic similarity, rather than closed-set classification, this work contributes to the development of more interpretable and adaptable solutions for forensic and large-scale AV applications.

6.1. Major findings

The results demonstrate that incorporating style features consistently improves the performance of all three model architectures across most evaluation metrics. Notably:

- The SN with style features achieved the highest accuracy (0.8472) and F1 score (0.7228) among all models, indicating that this combination is particularly effective at identifying authorship similarity when stylistic nuances are considered.
- Although the PCN model (without style features) had the highest precision (0.7621), it showed significantly lower recall (0.6121), suggesting that it was more conservative in making positive predictions. Once style features were added, recall improved to 0.7052, resulting in a more balanced performance across precision and recall.
- The FIN model with style features showed consistent improvements over its no-style counterpart in recall (from 0.6681 to 0.6961), F1 score (from 0.6791 to 0.6906), and AUC (from 0.7995 to 0.8062), although precision slightly decreased. This suggests that while the model became marginally less precise, it became more effective at identifying true positives, making it more suitable for real-world AV tasks where recall is crucial.

Overall, adding style features benefits recall and F1 score the most, indicating that style information helps models better generalize over variations in writing, particularly in subtle cases where surface-level content similarity may be insufficient. Importantly, in AV, recall is a critical metric, as it reflects the model's ability to correctly identify true same-author pairs. A higher recall means fewer false negatives, which is essential in applications where missing a true authorship match could undermine trust in the system or lead to incorrect conclusions in forensic or security-related contexts.

The FIN generally has higher complexity than SN due to the explicit interaction layers that compute pairwise combinations between features of both inputs. This added complexity results in longer training times for FIN, larger model size due to more parameters in the interaction layers and higher memory usage during both training and inference. However, this added cost is minimal and leads to a notable gain in recall. Across all variants tested (with and without style), FIN

consistently outperformed SN in recall. We consider that the increased recall from FIN's added complexity is a worthwhile trade-off, especially in high-stakes scenarios such as authorship verification or forensic analysis, where missing a true positive can have serious consequences. While FIN does increase computational cost, it remains feasible for near-real-time use on modern hardware, and further optimizations, such as model pruning or batching, can help reduce latency.

6.2. Limitations

The proposed model, while effective, has certain limitations that should be acknowledged. The process of extracting style features from texts is based on a predetermined set of features, which may not capture the full complexity of an author's writing style. This reliance on predefined features can limit the model's ability to fully understand subtle or complex stylistic nuances. As a result, the model may miss important aspects of authorship that are not represented by the extracted features.

RoBERTa, like other transformer-based models, processes only a limited number of tokens at a time, typically constrained by the model's input size (e.g., 512 tokens). This means that each datapoint has a fixed maximum length, and texts longer than this limit are trimmed. Consequently, important contextual or stylistic information may be lost if the input text exceeds this length, potentially affecting the accuracy of the AV task.

6.3. Future research

Future research could focus on several directions to enhance the current work and address the limitations identified in this study. One potential avenue for improvement is to capture more comprehensive information from the entire text. Currently, due to RoBERTa's token length limitation, only a portion of the text (e.g., the first 512 tokens) is processed. By extending the model's ability to handle longer texts, we could better capture the full context and style features of a document, leading to more accurate AV.

We could also explore the application of more advanced feature extraction techniques for short texts, such as those using deep learning models like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), which have demonstrated effectiveness in authorship attribution tasks for informal texts like tweets (Shrestha et al., 2017).

Another area for future research is the expansion of the style features extracted from the texts. At present, only a minimal set of features is used to represent an author's style. By incorporating a broader range of style features, such as measures of lexical diversity, discourse structure, or punctuation usage, the model could achieve a more nuanced understanding of an author's writing patterns, which might improve performance, particularly in more challenging cases. We also recognize the value of neural style embeddings (e.g., from BERT or similar models), and incorporating these richer representations of style is a natural next step. Our current findings already show that even basic stylometric features can enhance performance, suggesting that more sophisticated style modeling could improve results further.

Moreover, while our architecture allows the networks to implicitly learn interactions between style and semantic signals, no explicit analysis was performed to isolate or visualize the contribution of individual style features in relation to semantic components. Future work may investigate such interactions using attention mechanisms or feature attribution methods to better understand the interplay between writing style and meaning, which could enhance interpretability and model robustness.

A further valuable extension would be to develop a model whose architecture or loss function dynamically adapts based on the degree of class imbalance. This could involve incorporating imbalance-aware sampling strategies, cost-sensitive learning, or designing auxiliary components that assess and adjust decision thresholds according to the observed class ratio. By making the model's behavior a function of class distribution, rather than assuming uniformity, we may improve its robustness and generalization to real-world deployment settings where true positive cases are scarce and verification is critical.

In future work, we also aim to evaluate our approach on the PAN authorship verification datasets to enable direct comparison with existing state-of-the-art methods. This would provide a more standardized benchmark and facilitate a clearer assessment of our model's generalizability.

Given the generalizability of the models developed in this study, there is potential for their application across different domains. With small adjustments, we believe that the methods could be adapted for AV tasks in other contexts, such as academic writing, social-media posts, or literary works. Exploring cross-domain generalizability would provide valuable insights into the robustness and adaptability of the proposed methods in various settings.

7. Conclusion

In this study, we proposed and evaluated several models for AV and authorship attribution, including the FIN, PCN, and SN. We analyzed the incorporation of style features into the models and the impact of these features on model performance. Our experiments demonstrated the potential for improving AV by combining semantic embeddings with style features, especially when addressing class imbalance through weighted loss functions.

Although our models demonstrated strong performance, several limitations were identified, including the constraint of using predefined style features and the limited text length due to the nature of RoBERTa embeddings. These limitations suggest areas for future improvements, such as extending the input text length or incorporating additional style features to enrich the models further.

Overall, this research highlights the importance of combining linguistic features with deep learning models to improve the accuracy of AV. The approach is adaptable across domains, and with further adjustments, it can be generalized to tackle other text classification tasks. Our hybrid models mark a significant advancement towards developing robust AV systems suitable for practical deployment.

Future work could focus on refining the models' generalization capabilities, exploring more advanced feature extraction techniques, and extending the approach to longer and more complex text corpora.

CRediT authorship contribution statement

Britt van Leeuwen: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Project administration, Resources, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Sandjai Bhulai:** Conceptualization, Supervision, Writing – review & editing. **Rob van der Mei:** Conceptualization, Supervision, Writing – review & editing.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work the authors used ChatGPT (OpenAI) in order to improve the clarity and coherence of the writing. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Britt van Leeuwen reports financial support was provided by Research Institute for Mathematics and Computer Science in the Netherlands. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Length-related bias in models with style features

This section presents correlation matrices examining length-related biases in all models, both with and without style features. The matrices show relationships between average text length, length difference between text pairs, and prediction correctness, providing insight into how length influences model performance (see Fig. A.1).

Correlation matrix for Pairwise Concatenation (without style)

	avg_len	length_diff	correct
avg_len	1.000000	0.878600	0.023731
length_diff	0.878600	1.000000	-0.005317
correct	0.023731	-0.005317	1.000000

Correlation matrix for Feature Interaction (without style)

	avg_len	length_diff	correct
avg_len	1.000000	0.878600	0.008540
length_diff	0.878600	1.000000	-0.016155
correct	0.008540	-0.016155	1.000000

Correlation matrix for Siamese (without style)

	avg_len	length_diff	correct
avg_len	1.000000	0.878600	0.010822
length_diff	0.878600	1.000000	-0.012645
correct	0.010822	-0.012645	1.000000

Correlation matrix for Pairwise Concatenation (with style)

	avg_len	length_diff	correct
avg_len	1.000000	0.903202	0.013779
length_diff	0.903202	1.000000	-0.008542
correct	0.013779	-0.008542	1.000000

Correlation matrix for Feature Interaction (with style)

	avg_len	length_diff	correct
avg_len	1.000000	0.903202	0.008439
length_diff	0.903202	1.000000	-0.000517
correct	0.008439	-0.000517	1.000000

Correlation matrix for Siamese Network (with style)

	avg_len	length_diff	correct
avg_len	1.000000	0.903202	0.017153
length_diff	0.903202	1.000000	0.001718
correct	0.017153	0.001718	1.000000

Fig. A.1. Correlation matrices for all models with and without style features.

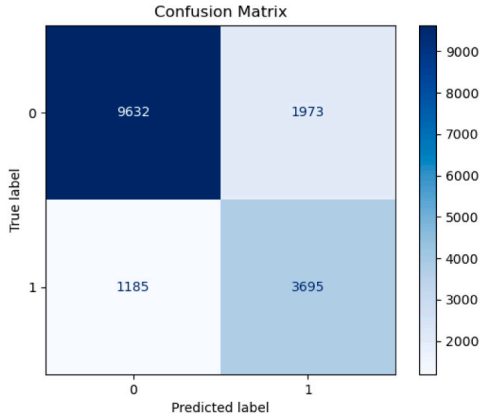

Fig. B.1. Confusion matrix for the Feature Interaction model without style features.

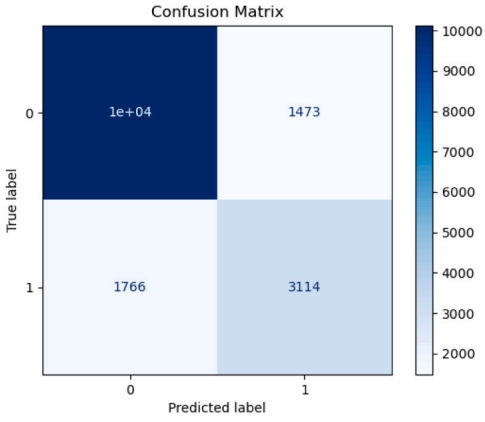

Fig. B.2. Confusion matrix for the Feature Interaction model with style features.

Appendix B. Confusion matrices

Figs. B.1–B.6 present confusion matrices for the three evaluated models — Feature Interaction, Pairwise Concatenation, and Siamese — both with and without the inclusion of style features. These matrices provide detailed insights into the classification performance and error distribution for each configuration.

Appendix C. Topic bias plots

Figs. C.1–C.12 displays topic bias plots for all three models, both with and without style features. These plots illustrate the relationship between topic similarity and the models' mean predictions, highlighting how topic influence varies across different model configurations (see Table C.1).

Appendix D. Training curves

To supplement the main evaluation metrics, Figs. D.1–D.5 display the training and validation performance for all models over the course of training.

Table C.1
Correlation between model predictions and topic similarity across models. R^2 indicates the proportion of variance explained, and ρ is Spearman's rank correlation.

Model	Style features	R^2	ρ (Spearman)
Pairwise Concatenation (PCN)	No	0.125	0.327
Pairwise Concatenation (PCN)	Yes	0.102	0.256
Feature Interaction (FIN)	No	0.141	0.305
Feature Interaction (FIN)	Yes	0.099	0.275
Siamese Network (SN)	No	0.164	0.425
Siamese Network (SN)	Yes	0.148	0.394
RoBERTa SN	No	0.115	0.058
RoBERTa SN	Yes	0.069	0.056
RoBERTa FIN	No	0.052	0.030
RoBERTa FIN	Yes	0.052	0.042
RoBERTa PCN	No	0.035	0.027
RoBERTa PCN	Yes	0.045	0.036

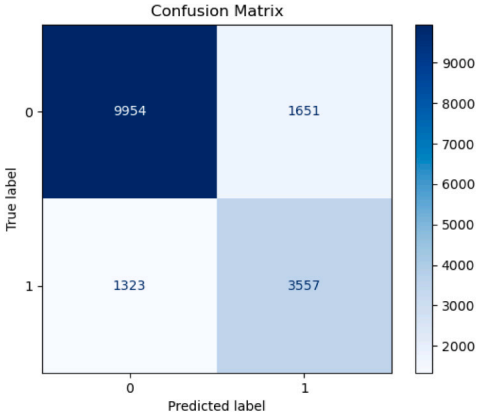

Fig. B.3. Confusion matrix for the pairwise concatenation model without style features.

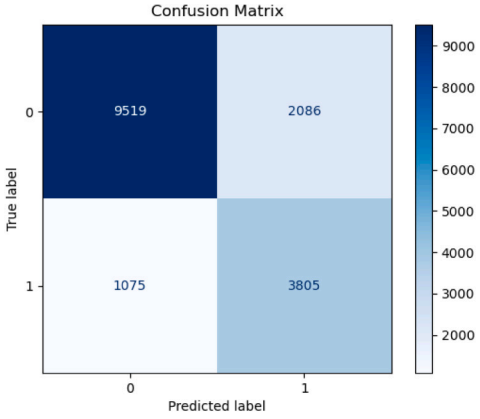

Fig. B.5. Confusion matrix for the Siamese model without style features.

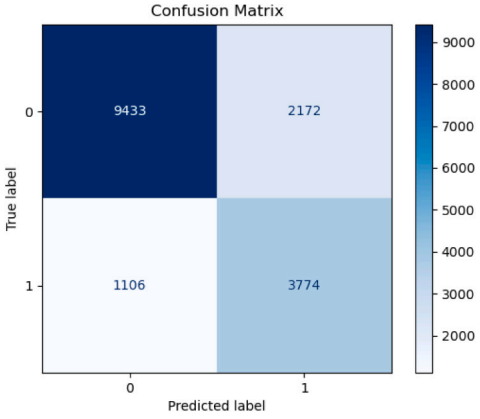

Fig. B.4. Confusion matrix for the pairwise concatenation model with style features.

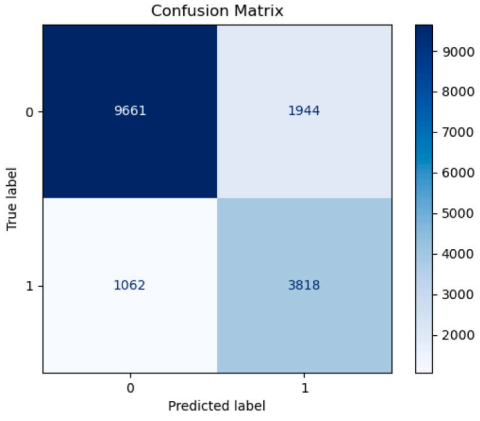

Fig. B.6. Confusion matrix for the Siamese model with style features.

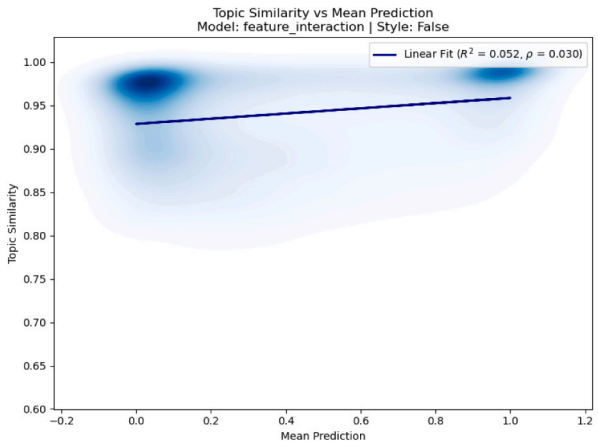

Fig. C.1. Feature Interaction model without style features (RoBERTa-based topic similarity). A weak positive relationship between topic similarity and mean prediction is observed.

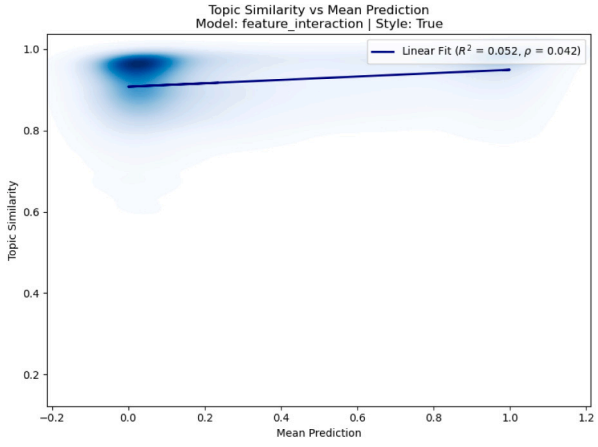

Fig. C.2. Feature Interaction model with style features (RoBERTa-based topic similarity). Topic influence remains weak.

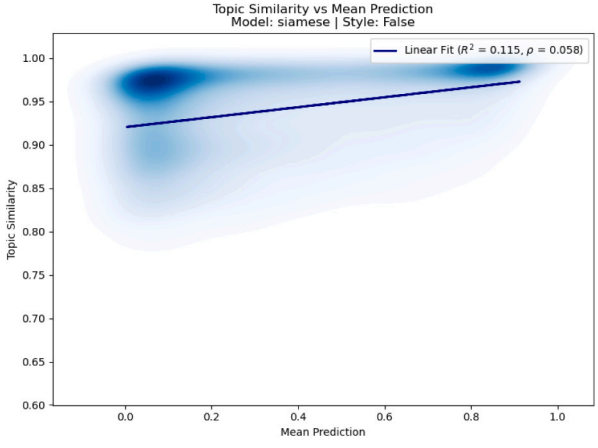

Fig. C.3. Siamese model without style features (RoBERTa-based topic similarity). Shows the strongest topic influence among models.

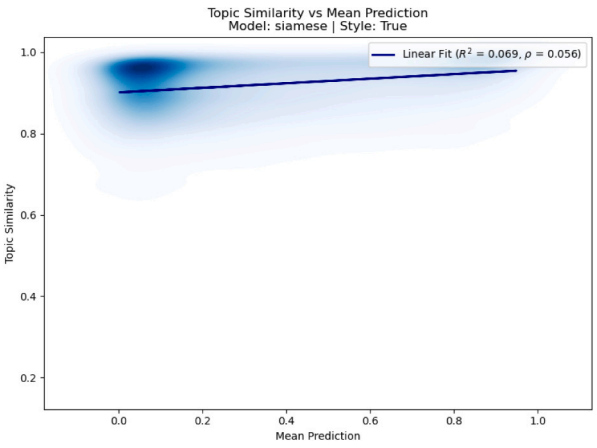

Fig. C.4. Siamese model with style features (RoBERTa-based topic similarity). Topic influence is reduced compared to the version without style.

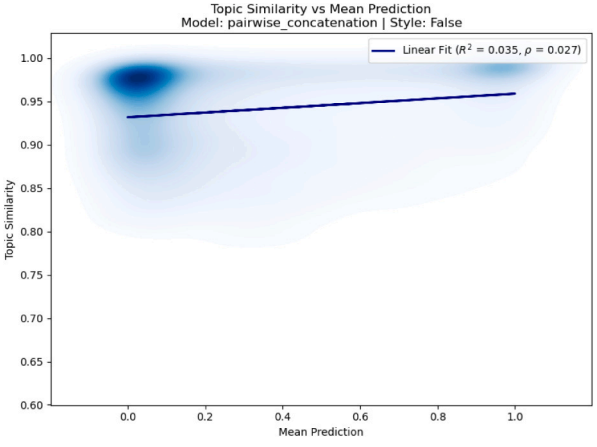

Fig. C.5. Pairwise concatenation model without style features (RoBERTa-based topic similarity). Exhibits the weakest topic correlation.

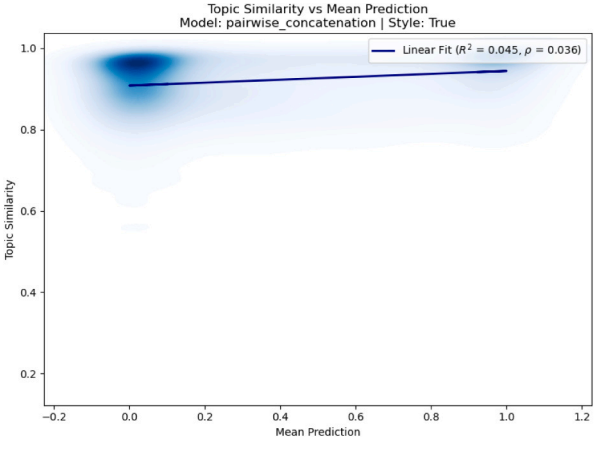

Fig. C.6. Pairwise concatenation model with style features (RoBERTa-based topic similarity). Slightly increased topic dependence.

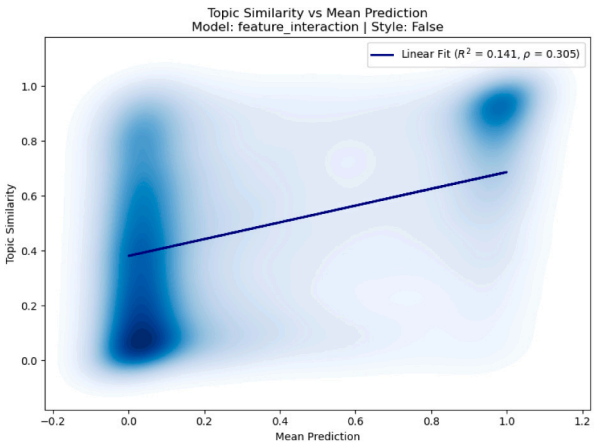

Fig. C.7. Feature Interaction model without style features (LDA-based topic similarity). Shows slightly more spread and a weak positive trend.

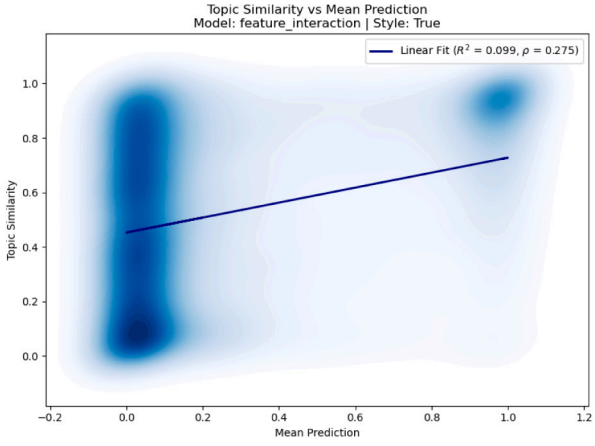

Fig. C.8. Feature Interaction model with style features (LDA-based topic similarity). Slightly increased topic influence compared to RoBERTa-based similarity.

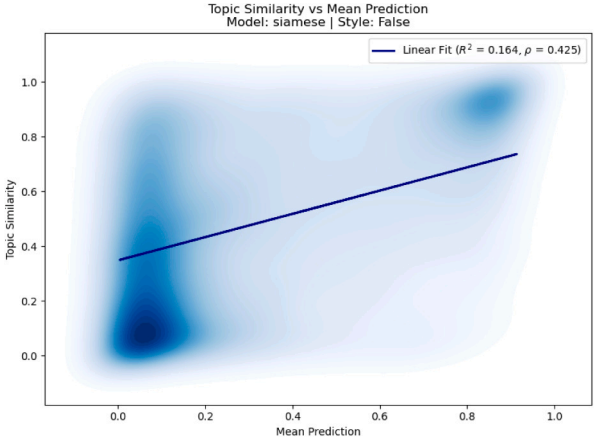

Fig. C.9. Siamese model without style features (LDA-based topic similarity). Shows a moderate positive trend.

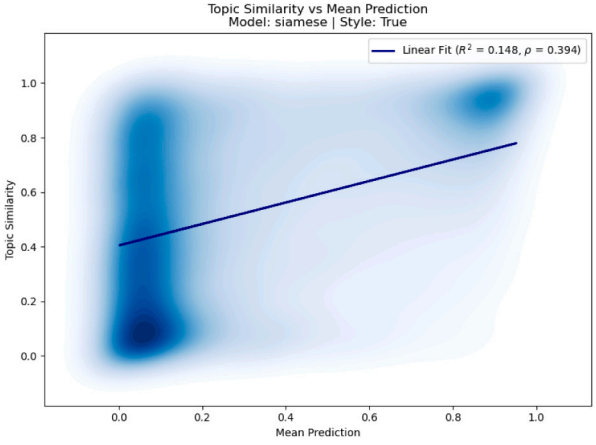

Fig. C.10. Siamese model with style features (LDA-based topic similarity). Topic influence is weaker than without style, similar to RoBERTa-based results.

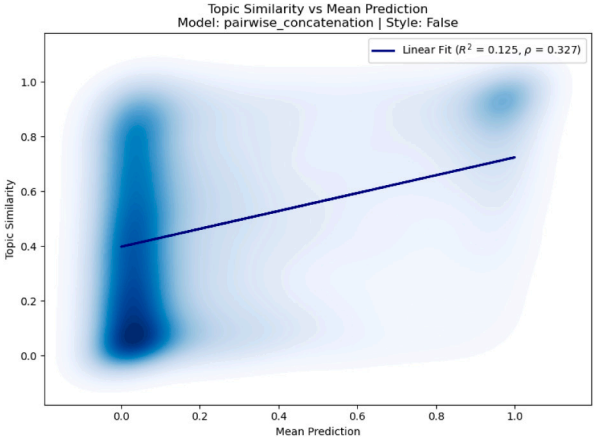

Fig. C.11. Pairwise concatenation model without style features (LDA-based topic similarity). Weak topic dependence.

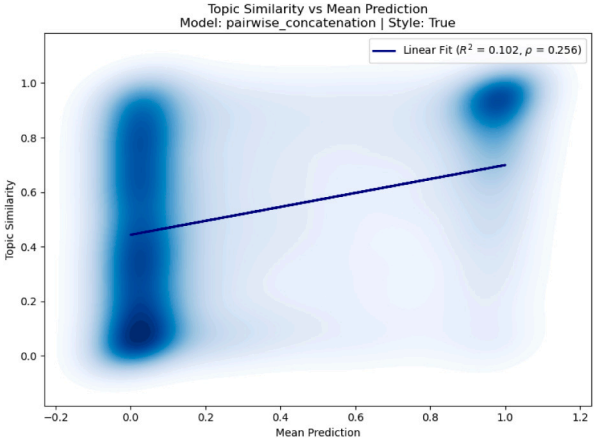

Fig. C.12. Pairwise concatenation model with style features (LDA-based topic similarity). Slightly increased topic correlation.

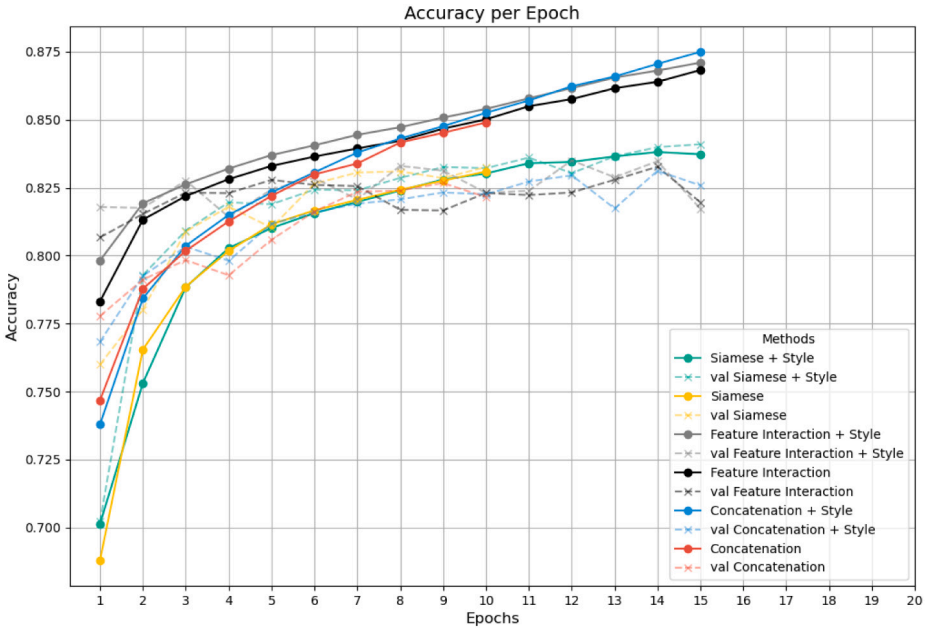

Fig. D.1. Accuracy.

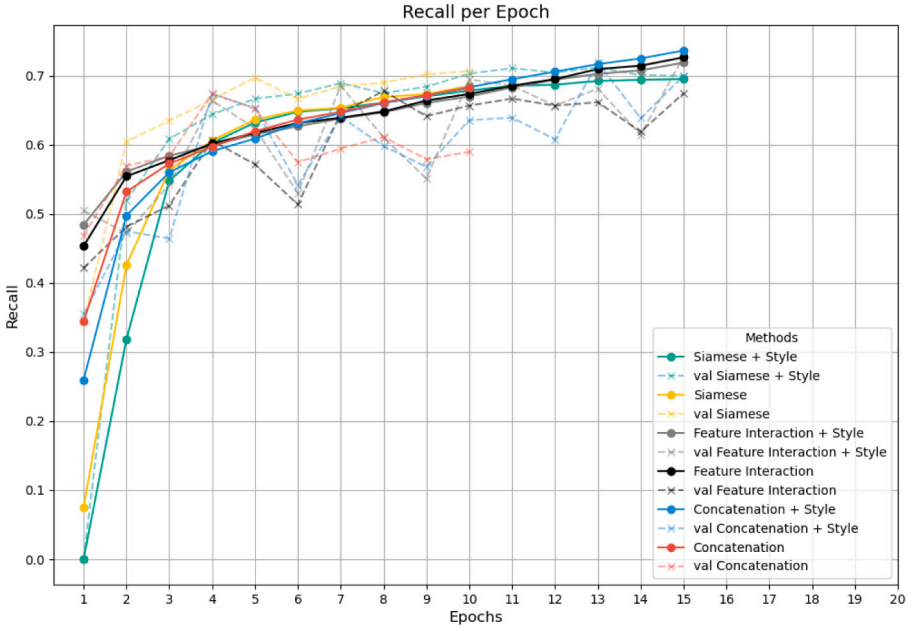

Fig. D.2. Recall.

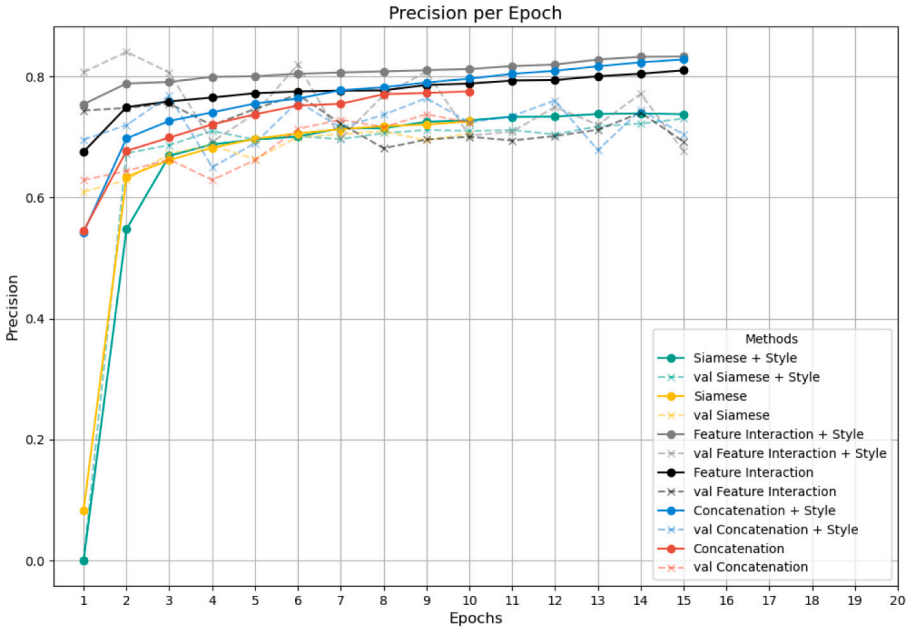

Fig. D.3. Precision.

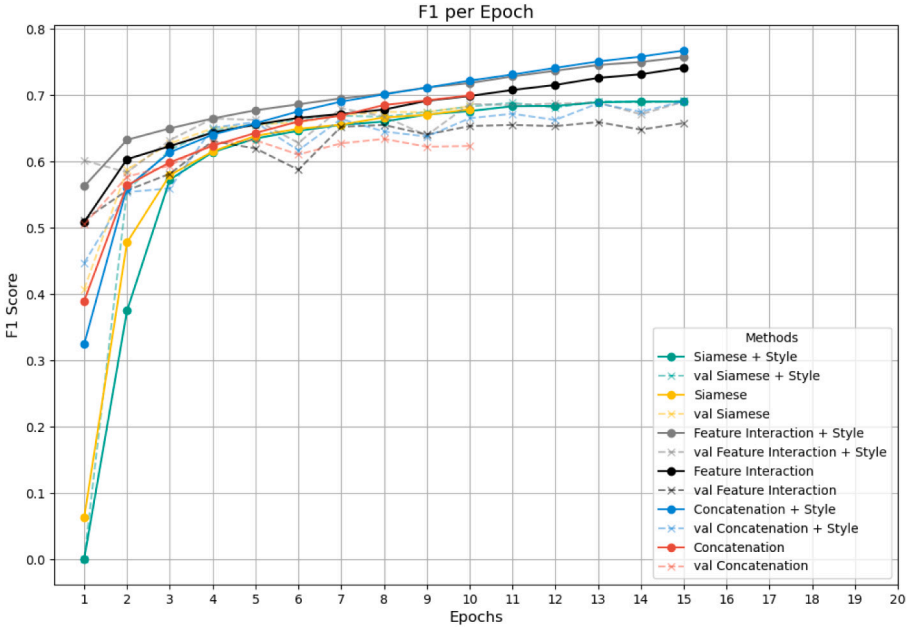

Fig. D.4. F1 score.

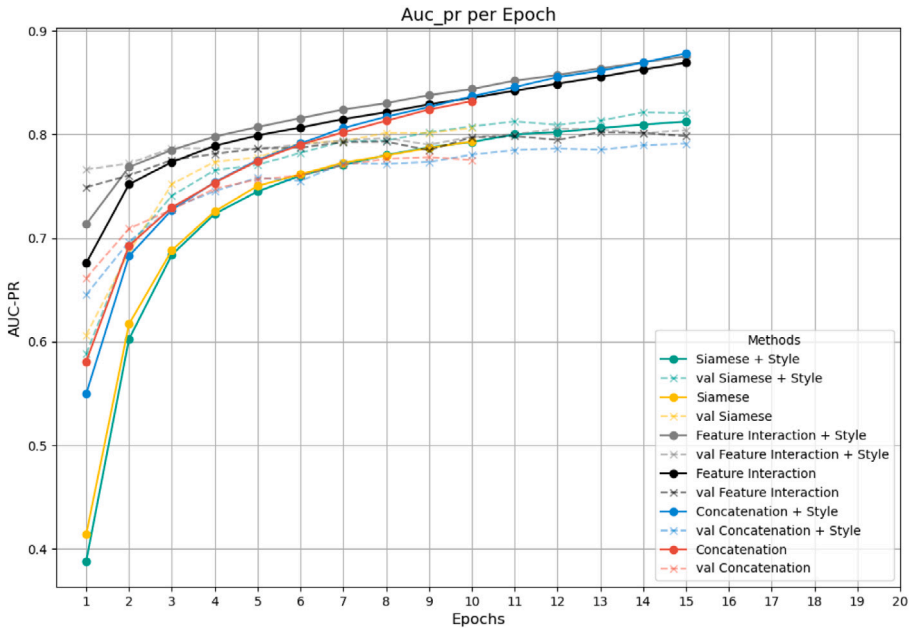

Fig. D.5. AUC-PR.

Data availability

The data is mentioned in the paper. It is open source.

References

Abbasi, Ahmed, et al. (2022). Authorship identification using ensemble learning. *Scientific Reports*, *12*(1), 9537.

Abedzadeh, Ali, Ramezani, Reza, & Fatemi, Afsaneh (2021). A weighted TF-IDF-based approach for authorship attribution. In *2021 11th international conference on computer engineering and knowledge* (pp. 188–193). IEEE.

Alsanoosy, Tawfeeq, Shalbi, Bodor, & Noor, Ayman (2024). Authorship attribution for english short texts. *Engineering, Technology & Applied Science Research*, *14*(5), 16419–16426.

Bevendorff, Janek, et al. (2020). Overview of pan 2020: Authorship verification, celebrity profiling, profiling fake news spreaders on twitter, and style change detection. In *Experimental IR meets multilinguality, multimodality, and interaction: 11th international conference of the CLEF association, cLEF 2020, thessaloniki, Greece, September 22–25, 2020, proceedings 11* (pp. 372–383). Springer.

Boenninghoff, Benedikt, Nickel, Robert M., & Kolossa, Dorothea (2021). O2D2: Out-Of-Distribution Detector to Capture Undecidable Trials in Authorship Verification—Notebook for PAN at CLEF 2021. In Guglielmo Faggioli, Nicola Ferro, Alexis Joly, Maria Maistro, & Florina Piroi (Eds.), *CLEF 2021 labs and workshops, notebook papers*. CEUR-WS.org, URL: http://ceur-ws.org/Vol-2936/paper-158.pdf.

Brocardo, Marcelo Luiz, et al. (2013a). Authorship verification for short messages using stylometry. In *2013 international conference on computer, information and telecommunication systems* (pp. 1–6). http://dx.doi.org/10.1109/CITS.2013.6705711.

Brocardo, Marcelo Luiz, et al. (2013b). Authorship verification for short messages using stylometry. In *2013 international conference on computer, information and telecommunication systems* (pp. 1–6). IEEE.

Castro, Daniel Castro, et al. (2015). Authorship verification, average similarity analysis. In *Proceedings of the international conference recent advances in natural language processing* (pp. 84–90).

Chollet, François, et al. (2015). Keras. https://keras.io.

Crespo-Sanchez, Melesio, et al. (2022). A content spectral-based analysis for authorship verification.. In *CLEF (working notes)* (pp. 2416–2425).

Flesch, Rudolf (1948). A new readability yardstick. *Journal of Applied Psychology*, *32*(3), 221–233. http://dx.doi.org/10.1037/h0057532.

Huang, Mingjie, et al. (2022). Authorship verification based on fully interacted text segments.. In *CLEF (working notes)* (pp. 2491–2495).

Jones, Keenan, Nurse, Jason R. C., & Li, Shujun (2022). Are you robert or roberta? deceiving online authorship attribution models using neural text generators. *16*, In *Proceedings of the international AAAI conference on web and social media* (pp. 429–440).

Juola, Patrick (2021). Verifying authorship for forensic purposes: A computational protocol and its validation. *Forensic Science International*, *325*, Article 110824.

Kipnis, Alon (2020). Higher criticism as an unsupervised authorship discriminator.. In *CLEF (working notes)*.

Konstantinou, Stefanos, Li, Jinqiao, & Zinonos, Angelos (2022). Different encoding approaches for authorship verification. In Guglielmo Faggioli, Nicola Ferro, Allan Hanbury, & Martin Potthast (Eds.), *CEUR workshop proceedings*: *vol. 3180*, *Proceedings of the working notes of CLEF 2022 - conference and labs of the evaluation forum, bologna, Italy, September 5th - to - 8th, 2022* (pp. 2532–2540). CEUR-WS.org, URL: https://ceur-ws.org/Vol-3180/paper-206.pdf.

Lagutina, Ksenia, et al. (2021). Authorship verification of literary texts with rhythm features. In *2021 28th conference of open innovations association* (pp. 240–251). IEEE.

Lei, Ziwang, et al. (2022). Application of BERT in author verification task.. In *CLEF (working notes)* (pp. 2560–2564).

Liu, Yinhan, et al. (2019). Roberta: A robustly optimized BERT pretraining approach. arXiv:1907.11692 URL: https://arxiv.org/abs/1907.11692.

Manolache, Andrei, et al. (2021). Transferring bert-like transformers' knowledge for authorship verification. arXiv preprint arXiv:2112.05125.

Martinez-Galicia, Jorge Alfonso, et al. (2022). Graph-based siamese network for authorship verification. In *CLEF 2022 labs and workshops, notebook papers* (pp. 2594–2606).

Misini, Arta, Kadriu, Arbana, & Canhasi, Ercan (2024). Authorship classification techniques: Bridging textual domains and languages. *International Journal on Information Technologies and Security*, *16*(1), 27–38.

Najafi, Maryam, & Tavan, Ehsan (2022). Text-to-text transformer in authorship verification via stylistic and semantical analysis.. In *CLEF (working notes)* (pp. 2607–2616).

Potha, Nektaria, & Stamatatos, Efstathios (2017). An improved impostors method for authorship verification. In *International conference of the cross-language evaluation forum for European languages* (pp. 138–144). Springer.

Potha, Nektaria, & Stamatatos, Efstathios (2020). Improved algorithms for extrinsic author verification. *Knowledge and Information Systems*, *62*(5), 1903–1921.

Schler, Jonathan, et al. (2006). Effects of age and gender on blogging.. (pp. 199–205).

Shrestha, Prasha, et al. (2017). Convolutional neural networks for authorship attribution of short texts. In Mirella Lapata, Phil Blunsom, Alexander Koller (Eds.), *Proceedings of the 15th conference of the European chapter of the association for computational linguistics: volume 2, short papers* (pp. 669–674). Valencia, Spain: Association for Computational Linguistics, URL: https://aclanthology.org/E17-2106/.

Stamatatos, Efstathios, Kestemont, Mike, et al. (2022). Overview of the authorship verification task at PAN 2022. *CEUR Workshop Proceedings*, [ISSN: 1613-0073] *3180*, 2301–2313, URL: https://ceur-ws.org/Vol-3180/.

Stamatatos, Efstathios, Kredens, Krzysztof, et al. (2023). Overview of the authorship verification task at PAN 2023. In *Conference and labs of the evaluation forum*. URL: https://api.semanticscholar.org/CorpusID:264441636.

Stein, Benno, Lipka, Nedim, & Prettenhofer, Peter (2011). Intrinsic plagiarism analysis. *Language Resources and Evaluation*, *45*, 63–82.

Tyo, Jacob, Dhingra, Bhuwan, & Lipton, Zachary C. (2021). Siamese bert for authorship verification.. In *CLEF (working notes)* (pp. 2169–2177).

Tyo, Jacob, Dhingra, Bhuwan, & Lipton, Zachary C. (2022). On the state of the art in authorship attribution and authorship verification. arXiv preprint arXiv:2209.06869.

Wang, Xiangyu, & Iwaihara, Mizuho (2021). Integrating RoBERTa fine-tuning and user writing styles for authorship attribution of short texts. In Leong Hou U, Marc Spaniol, Yasushi Sakurai, & Junying Chen (Eds.), *Web and big data* (pp. 413–421). Cham: Springer International Publishing.

Weerasinghe, Janith, & Greenstadt, Rachel (2020). Feature Vector Difference based Neural Network and Logistic Regression Models for Authorship Verification—Notebook for PAN at CLEF 2020. In Linda Cappellato, Carsten Eickhoff, Nicola Ferro, & Aurélie Névéol (Eds.), *CLEF 2020 labs and workshops, notebook papers*. CEUR-WS.org.

Ye, Yihui, et al. (2022). Authorship verification using convolutional neural network. In *CLEF (working notes)*.

Yılmaz, Tunç, & Scheffler, Tatjana (2023). Song authorship attribution: a lyrics and rhyme based approach. *International Journal of Digital Humanities*, *5*(1), 29–44.

Zamir, Muhammad Tayyab, et al. (2024). Stylometry analysis of multi-authored documents for authorship and author style change detection. arXiv:2401.06752 URL: https://arxiv.org/abs/2401.06752.