TriS: A Comprehensive Model for Writing Style Change Detection by Integrating Semantic, Syntactic, and Stylistic Features

Xiangyu Lu[1], **Xiaofan Sun**[1], **Ben Yao**[1]

[1]MSc in IT and Cognition, University of Copenhagen
{zds311, dcs354, skv338}@alumni.ku.dk

Abstract

The exploration of writing style shifts within multi-authored texts is a promising field in computational linguistics. It aims to localize instances of style transition, exhibiting immense potential for real-world applications such as plagiarism detection, forensic analysis, literary study, and author attribution. In this paper, we propose the model **TriS** for multi-author writing style change detection. TriS comprises three modules, namely **S**emantic, **S**yntactic, and **S**tylistic encoders, leveraging fine-tuned BERT models and a Logistic Regression classifier to analyze stylometric patterns associated with style shifts. This study focuses on the task presented in PAN Multi-Author Writing Style Analysis 2023, which is characterized by constrained topic information. Compared to individual encoders, our fusion model demonstrates improved performance on the medium-level dataset, with an accuracy of 0.742 and an F1-score of 0.709 on the validation set.

1 Introduction

Style Change Detection (SCD) is a significant research domain in digital text analysis, dedicated to recognizing transitions in writing style within a given text or document [1]. SCD serves a critical role not only in academic domains like plagiarism detection and literary analysis but also extends to practical applications such as forensic analysis and cybersecurity. While closely related to authorship verification and attribution, which focuses on determining the author of a text and assigning an unclaimed document to an author [2, 3], SCD distinguishes itself by focusing on the identification of writing style changes within multi-authored texts. It primarily explores variations in linguistic features, including morphological, syntactic, and stylistic patterns, to uncover shifts in the text's composition [4].

The task of style change detection, which is integrally connected with authorship attribution and verification, has witnessed a significant progression in the application of computational methods. In its early phases, the domain was dominated by manual methods hinged on rule-based methodologies. Linguists utilized stylistic markers, including character-level, lexical, syntactic, discourse-level, thematic, and even rhythmic attributes, to quantitatively analyze the stylometry of questioned documents [5, 6, 7, 8, 9]. Furthermore, supervised learning models, trained with labeled datasets, have expanded substantially in the realm of style change detection, employing a range of machine learning techniques such as Logistic Regression and Support Vector Machines [7]. Additionally, deep learning models such as LSTM were also employed in prediction of authorship

alternation [10]. Moreover, pre-trained transformers [11] and ensemble pre-trained language models [12] have been used for style change detection, given that they are capable of capturing intricate stylistic characteristics of high-dimensionalitality. Diverging from supervised methods, unsupervised learning, which achieves recognition task by learning the internal traits within the document, have also been extensively adopted. Clustering, a representative unsupervised technique that does not need parameter adjustment or labeled data training, is an example of such approach for SCD [13].

The exploration of style change detection is of profound significance in computational linguistics and associated fields. Firstly, the linguistic change can be helpful in identifying collaborative writing and verifying authorship in questioned texts [14]. Secondly, it is instrumental in detecting instances of plagiarism, a crucial aspect of maintaining academic integrity [15]. Additionally, in the domain of cybersecurity, the use of style change detection is critical in investigating cybercrime, potential online fraud or other harmful activities. This tool can be really beneficial when evil-doers attempt to copy the usual communicative style but have subtle differences that can be spotted, which can also help forensic linguists to deal with skeptical files [16]. Moreover, from the perspective of authorship argument and attribution in literary works, style change detection can be leveraged as a powerful tool to infer potential writer [8].

For the current project, the specific tasks and datasets of different difficulty levels are provided by the PAN Multi-Author Writing Style Analysis 2023 [17]. PAN@CLEF has organized a series of scientific events and shared projects dedicated to the realm of computational text analysis and stylometric study. In recent years, PAN SCD competition involving intriguing tasks of diverse level, is one formidable challenge for participants. In 2021, alongside the regular verification and detection subtasks, participants were additionally asked to allocate all paragraphs within a text to specific writers [18]. At PAN 2022, a sentence-level detection task was further introduced [1]. In contrast to previous editions, this year's focus in the PAN SCD task is to tackle trickier scenarios where the theme information for paragraph-level style change detection is constrained on datasets of varying difficulty [17].

To tackle the medium-level task as required, we put forward a fusion model TriS that encompasses Semantic, Syntactic, and Stylistic Encoder for detecting paragraph-level writing style changes. The final predicted occurrence of style change is based on the fusion score of the weighted semantic, syntactic, and stylistic features.

The report is structured as follows: it begins with an introduction with the literature review subsection. Following discusses the features of the datasets. The subsequent unit is the task specifications section. The paper continues with sections dedicated to the research methods and experiments. The results which evaluate the model performance are presented in the following unit, followed by a detailed discussion. Finally, the paper ends with conclusion and outlook.

1.1 Related Works

1.1.1 Stylometric Analysis in Linguistics and Literary

Initially, some stylistic markers are introduced across different domains of linguistics and literary analysis to identify shifts in writing style through simple statistical analysis. An typical example is phonological and phonetic indicator, such as tones and rhythm feature [19, 20], which is significant in understanding musicality and underlying intention of literary works, are particularly helpful for poem-based tasks [8]. When coupled with other lexical features, this marker could better aid in attributing authorship to poems [8]. In syntax, features such as sentence complexity can be quantified to uncover changes in novels written by native English novelists [7]. Besides, phrase collocations and lexical traits such as stop-words and vocabulary variety have been utilized in style change recognition [7] , while other study also incorporated rhetorical patterns to find out the joint authorship in The Federalist papers by carrying out a statistical analysis [14].

1.1.2 Computational Models for Style Change Detection

Some of the markers mentioned above may be challenging to be computed via simple statistical methods. However, the advent of digital methods enabled authorship and stylistic analyses to be conducted on large-scale linguistic datasets with an improved precision. Sebastiani provided a general machine learning solution to text categorization tasks, emphasizing class traits learning and classifier construction [21]. [7] utilized machine learning classifiers like SVM and Logistic Regression, applying linguistic features like word choice, punctuation patterns, and phrase structures to attribute novels to their respective writing phases. Although classification performance varied across different authors, this approach demonstrated the feasibility of identifying writing periods in some novels. Similarly, another study combined stylometric analysis and machine learning approaches to investigate plagiarism in PhD theses [22]. This in-document plagiarism detection method has achieved satisfactory accuracy on current text excerpts. Deep learning models have also been explored and are anticipated to optimize performance on this task. [23] carried out a comprehensive evaluation of Long Short-Term Memory (LSTM) and compared it with n-gram language models. The findings indicated the superior performance of the LSTM-based model on certain datasets. Recently, with the popularity of large language models, researchers also started to distinguish stylistic alterations in texts produced by humans and AI, employing RoBERTa, a fine-tuned pre-trained transformer [24]. Studies of this kind contribute significantly to the detection of misinformation and AI-facilitated plagiarism. In terms of unsupervised algorithms, text clustering is the prevalent technique to identify authorship. According to [25], low-dimensional feature are preferred for unsupervised learning and therefore, the punctuation is used for the linguistic characteristic for clustering, which outperformed clustering using bi-gram.

1.1.3 Previous Approaches on PAN SCD Tasks

In recent years, the PAN SCD shared task has been committed to advancing the field of SCD by introducing several novel approaches to address SCD problem with mixed methodological focuses. For the PAN 2018 task, [26] employed statistical text features, the hashing classifier, counting classifier, and a weighted probability calculation to provide a binary answer for text authorship identification. In [27]'s work, the parallel hierarchical attention network was applied to leverage hierarchical sentence and word order-preserved information, thereby advancing style change detection. In subsequent years, more challenging tasks requiring the number of co-authors have been raised. [28] proposed a double-step model involving a binary classification and then three clustering algorithms for multiple authors recognition in documents. Another study, which also utilized clustering methods with threshold and window merge techniques to indicate the number of writers [29]. Furthermore, the pre-trained language model was adopted to produce word representations and the Random Forest classifier was selected for better performance and then trained to make final predictions [11]. For the three-fold 2021 PAN task, [10] employed different machine learning methods for tasks at varied level and finally developed a blended model for SCD prediction utilizing LSTM networks. By using Google's pre-trained transformer BERT and a classifier, the style similarity label was obtained as an indicator for style change which was measured as a binary classification [30]. In 2022 PAN SCD, more new methods are invited to tackle the challenging sentence-level detection. By incorporating a template, [31] presented a prompt-based method using pre-trained BERT and mask language model. An unconventional and uncommon method was proposed and achieved good final scores on test set by soliciting ground-truth label of the test set on search engines [32]. [12] designed an ensemble transformer-based deep learning model, which only showed a marginal increase in F-score compared to RoBERTa. Additionally, [13] utilized both paragraph- and sentence-level features and proposed a text-intrinsic authorship clustering approach to estimate the number of writers.

To conclude, the inspiring previous computational models and the significant studies in former editions of PAN SCD tasks motivate us to propose a new model aiming at enhancing the identification task and improving applicability.

2 Data

The English datasets provided by Multi-Author Writing Style Analysis 2023 task are based on users' post on the diverse subreddits of the Reddit platform [17]. In each dataset, the texts related to each input problem are assigned unique IDs. These IDs can also be linked to the submitted solutions for their respective input problems. Every text consists of paragraphs containing an indeterminate number of writing style changes. The format of its solution follows this structure:: "authors": X, "changes": [1, 0, . . .]. X refers to the number of authors, and "changes" refers to the array showing whether there is a writing style change between pairs of consecutive paragraphs. In this format, "1" suggests a style change between the first and second paragraphs in the corresponding text, while "0" shows no change between the second and third paragraphs.

There are altogether three datasets of three levels of difficulty, in order to control the topical diversity.

1. **Easy**: The topic of the paragraph varies along with authorship changes in a document, facilitating the detection of writing style changes.
2. **Medium**: The document exhibits limited but present topical diversity, which compels approaches to prioritize style as the key factor for effectively solving the detection task.
3. **Hard**: All paragraphs within a document revolve around a singular topic.

We conducted a comprehensive analysis of the training set and validation set in all the three datasets (Table 1), including the number of text files, the number of paired paragraphs, the number of positive samples (i.e., the number of occurrences where there is a change in writing style/authorship between two paragraphs), the number of negative samples (i.e., the number of occurrences where there is no change in writing style/authorship between two paragraphs), and the total number of samples (i.e., which is the sum of positive and negative samples).

The training set of all datasets consists of 4,200 files, and the validation set contains 900 files. The easy-level dataset has a relatively small number of paired paragraphs, with only 12909 in the training set and 2832 in the validation set. Nonetheless, the medium-level dataset has 28216 pairs in the training set and 7048 in the validation set. As for the hard-level dataset, it includes 19115 pairs in the training set and 4112 in the validation set. The distribution of the number of samples across the three datasets follows the same pattern observed in the distribution of paired paragraphs. The easy-level dataset has the fewest number of samples, followed by the hard-level dataset, and the medium-level dataset has the highest number. Compared to the other two datasets, the easy-level dataset exhibits extreme imbalance between the number of positive and negative samples. This disparity is observed in both the training and validation sets, where the number of positive samples significantly outweighs the number of negative samples.

Notably, we observed an inequality between the number of paired paragraphs and the number of all samples in certain datasets. For instance, in the training set of the easy-level dataset, there are 12,909 pairs of paragraphs, but the corresponding truth-problem-X.json files provide only 12,904 samples. This discrepancy highlights the presence of erroneous or incomplete data within these datasets, necessitating manual data cleansing procedures.

3 Tasks

In contrast to the previous year, the Writing Style Analysis task in 2023 focuses solely on detecting changes in writing style between pairs of consecutive paragraphs. Although the datasets have three different difficulty levels, the uniform goal is to pinpoint the locations in a multi-author document where there is changes in authorship.

Figure 1 provides examples from all the datasets to illustrate the process and outcomes of the task. Since the task remains the same across the three datasets with varying difficulty levels, we will proceed with further explanation using document B as an example.

item	Easy		Medium		Hard	
	train	**valid**	**train**	**valid**	**train**	**valid**
N of documents	4200	900	4200	900	4200	900
N of pairs	12909	2832	28216	7048	19115	4112
N of positive samples	11347	2451	13215	3029	9021	1953
N of negative samples	1557	377	15001	4013	10092	2159
N of samples	12904	2828	28216	7042	19113	4112

Table 1. Statistic results of datasets

.

Example document B in Figure 1 referring to medium-level task comprises four paragraphs, and the task required the detection of possible authorship changes between paragraphs. Two variations occurs: one between the second and third paragraphs and another between the third and fourth paragraphs. As a result, the output contains a value of 1 at the corresponding positions, indicating the presence of these variations. However, as there is no authorship change at the first transition of paragraphs, the second position of the output array shows a value of 0. It's worth noting that there are indeed subtle yet observable topic changes in align with the changes in authorship. The level of topical change, occurring alongside authorship/writing style changes, is amplified in the easy-level task. However, in the hard-level task, this thematic shift is eliminated. This controlled topical changes are viewed to be a latent assistance to the writing style changes detection task [33]. The topical change, as a controlled variable across the three datasets, can facilitate our understanding of its impact on the task of style change detection.

Certain text-derived features can be utilized to address the task of writing style detection. Stylistic attributes, such as word frequencies, average word length, and average sentence length and so on, have exhibited promising results in some types of style change detection tasks such as intrinsic plagiarism analysis and the identification of writing style inconsistencies [33]. Meanwhile, syntactic features such as part-of-speech tags have demonstrated effectiveness in tasks involving authorship identification within a given document [28], as well as in tasks aimed at detecting style changes within the document [34]. Moreover, semantic features like BERT-generated embeddings have shown exceptional performance in tasks related to writing style change detection, which is similar to the task we are required to accomplish [35]. These preceding insightful studies reinforce the potential of utilizing text-derived features to accomplish our writing style change detection task on datasets with varying difficulty.

4 Methods

In this section, we will provide a detailed introduction to the model we used for writing style analysis. The input of our model is a text file consisting of several paragraphs. Specifically, the preprocessing module of the model divides the text into separate paragraphs and combines them into pairs of paragraphs in the original order of the text file. Each text pair is then fed into the model for analysis to determine whether there is a writing style change.

The main body of our model consists of three modules. The first part is a text content analysis module that utilizes BERT as a pre-trained model (**Semantic Encoder**). This module examines the textual content and extracts semantic information for style analysis. The second part is a POS-tag encoder module that also utilizes BERT as a pre-trained model (**Syntactic Encoder**). It encodes the part-of-speech tags of the text, which can provide additional syntactic insights into the writing style. Lastly, the third part is a logistic regression model (**Stylistic Encoder**) that

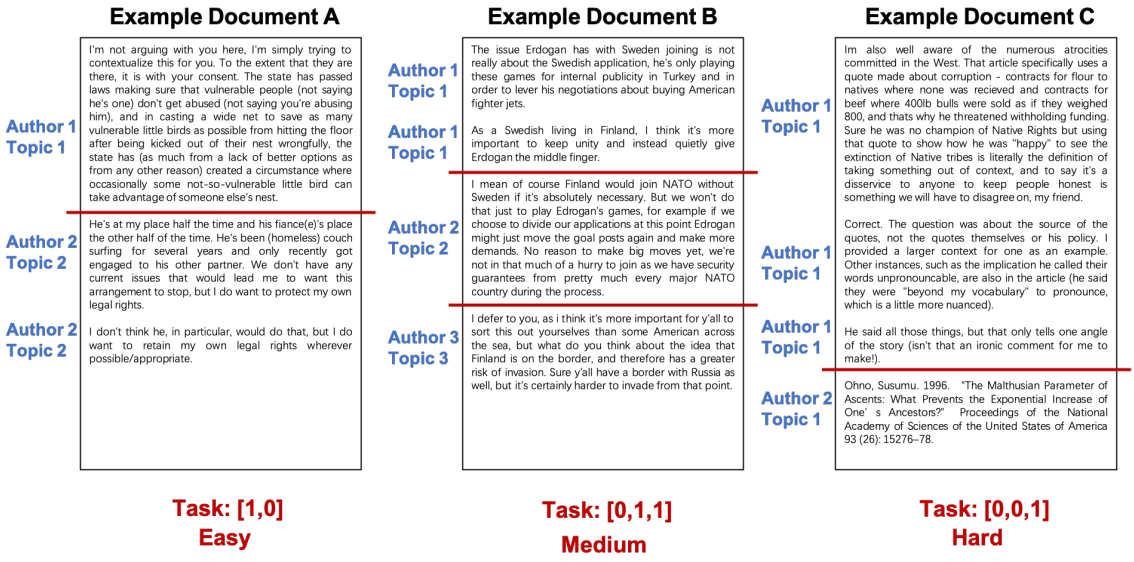

Figure 1. Example from '*Multi-Author Writing Style Analysis 2023*' task and dataset

analyzes the stylistic features of the text. This module captures specific linguistic patterns and features that contribute to writing style. The outputs of these three sub-models are then fused together in our fusion module (**Fusion Recognizer**), which combines and integrates the individual outputs to provide the final analysis result.

In addition to the aforementioned model, we also constructed a random baseline model for comparison. This baseline model randomly predicts whether there is a change in writing style for a given text pair. It assigns an equal probability of 50% for both possibilities, indicating a change or no change in writing style. It is important to note that the random baseline model lacks the sophisticated analysis and feature extraction capabilities of our developed model. Its predictions are solely based on chance, without considering any linguistic or contextual information. Therefore, the random baseline should be considered a basic and rudimentary approach, serving as a benchmark for our model's performance in writing style analysis.

Overall, our model (Figure 2) combines three encoders, namely the Semantic Encoder, Syntactic Encoder, and Stylistic Encoder, to detect and analyze writing style changes in a given text. By integrating these components, we aim to provide a comprehensive and accurate analysis of writing style variations.

4.1 Semantic Encoder

In the Semantic Encoder module (Figure 3), our goal is to extract semantic features from the text for detecting writing style. To achieve this, we have opted to fine-tune a pre-trained BERT model. For the output of the BERT model, we have selected two components for further training: the *pooler output* and the *hidden state*.

The *pooler output* represents the independent encoding feature of the special [CLS] token in BERT, which serves as a global feature extraction for the entire input text. The *hidden state*, on the other hand, represents the encoding features for each token in the input. For the *hidden state*, we first select a slice from index 1 to -1, excluding the encoding features for the [CLS] and [SEP] tokens. Then, we apply adaptive average pooling to the remaining features, aiming to obtain the average feature representation of the vocabulary used in the text. It is worth noting that after the processing steps, the dimensions of the outputs from the two parts (*pooler output* and *hidden state*) are the same.

For both the *pooler output* and the *hidden state*, they are passed through separate multi-layer perceptron (MLP) layers for fine-tuning. The outputs of these MLP layers are then concatenated,

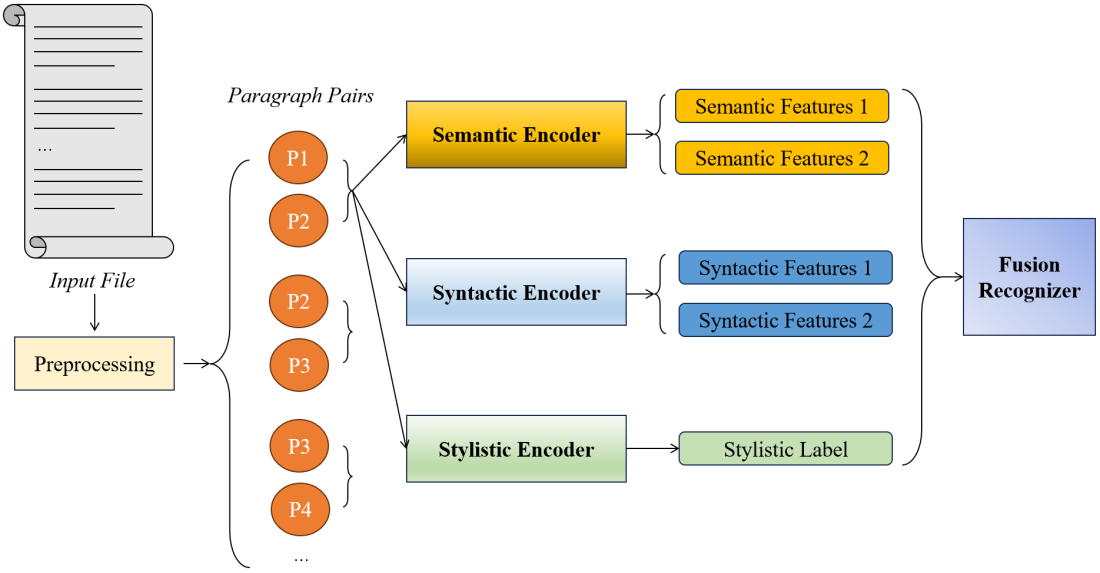

Figure 2. Our model TriS for writing style change detection

serving as the final output of the Semantic Encoder.

By utilizing the *pooler output* and the adapted *hidden state*, we capture both the global context

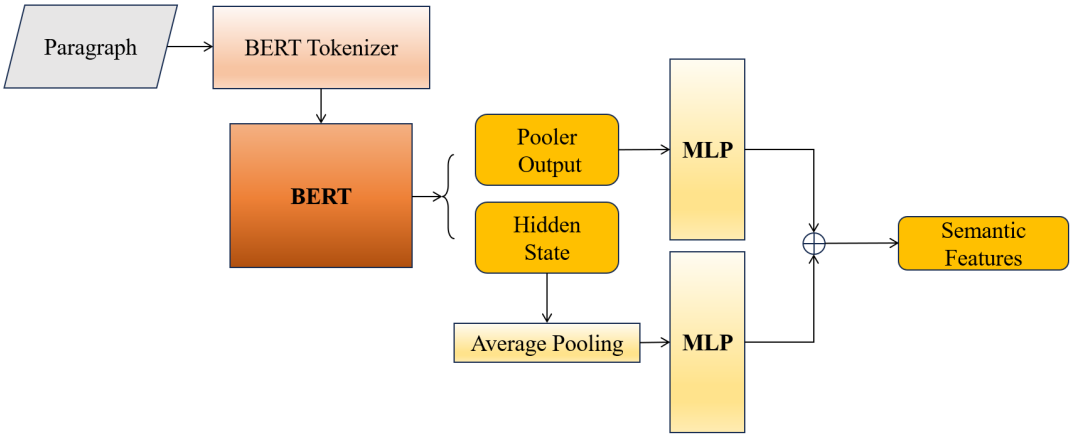

Figure 3. Semantic Encoder in TriS

information and the average token-level features, which are crucial for capturing the semantic characteristics of the text. The fine-tuning of the MLP layers further refines these features, allowing the Semantic Encoder to effectively analyze and detect writing style changes.

We use the Euclidean distance as a measure to evaluate the feature extraction performance of the Semantic Encoder. Specifically, we define the following loss function:

Given a pair of text segments, denoted as A and B, we first pass them through the Semantic Encoder to obtain their respective feature representations, denoted as F_A and F_B. The Euclidean distance between these feature representations is computed as follows:

$$D = \sqrt{\sum_{i=1}^{n} (F_A[i] - F_B[i])^2} \tag{1}$$

where n represents the dimensionality of the feature representations.

For each paragraph pair, the loss is calculated based on the label indicating whether a writing style change has occurred:

$$loss_{sem} = (1 - y) \cdot D^2 + y \cdot max(\Theta - D^2, \beta) \tag{2}$$

where y represents the label, Θ denotes the threshold, and β represents a small margin.

4.2 Syntactic Encoder

In the Syntactic Encoder module (Figure 4), our goal is to extract syntactic features from paragraphs for writing style detection. We believe that the POS tag sequences of paragraphs contain a wealth of information about word classes, phrase structures, hierarchical structures, and other syntactic features. To achieve this, we utilize a pre-trained BERT model and fine-tune it specifically for our task.

Unlike the Semantic Encoder, the Syntactic Encoder incorporates the NLTK library [36] to perform part-of-speech (POS) tagging on the input paragraphs. The resulting POS tag sequences are then encoded by the BERT encoder and fed into the BERT model. One key difference from the Semantic Encoder is that the Syntactic Encoder focuses solely on the *hidden state* output of the BERT model for fine-tuning.

Similarly to the Semantic Encoder, we select a slice from index 1 to -1, excluding the encoding features for the [CLS] and [SEP] tokens. Afterward, we apply adaptive average pooling to the remaining features. Finally, the output of the pooling layer is passed through an MLP. This MLP further refines the pooled features and serves as the final output of the Syntactic Encoder.

By utilizing the BERT model and incorporating POS tagging, the Syntactic Encoder aims

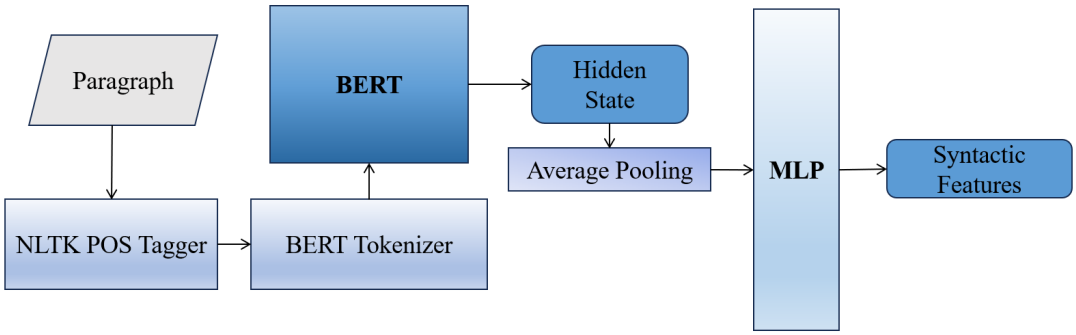

Figure 4. Syntactic Encoder in TriS

to capture and analyze the syntactic characteristics present in the POS tag sequences of the paragraphs. The output of the Syntactic Encoder, after passing through the MLP, provides valuable insights into the syntactic patterns and structures relevant to writing style.

We utilize the cosine similarity as a measure to assess the effectiveness of feature extraction in the Syntactic Encoder. Given a pair of text segments, denoted as A and B, we first pass them through the Semantic Encoder to obtain their respective feature representations,, denoted as F_A and F_B, and the loss can be defined as follows:

$$loss_{syn} = (1 - y) \times (1 - CosSim(F_A, F_B)) + y \times (1 + CosSim(F_A, F_B)) \tag{3}$$

where y represents the label, $CosSim$ is calculated using the following equation:

$$CosSim(F_A, F_B) = \frac{F_A \cdot F_B}{\|F_A\|\|F_B\|} \tag{4}$$

4.3 Stylistic Encoder

In the Stylistic Encoder module (Figure 5), our goal is to extract the stylistic features of paragraphs for writing style detection. We utilize 11 different stylistic features (as shown in Table 2) and construct their corresponding feature vectors. For each text pair, we subtract the feature vectors element-wise and take the absolute value, resulting in a stylistic difference feature vector for the text pair. Finally, we employ a logistic regression model to fit the feature vector and determine whether a writing style change has occurred.

No.	Stylistic Features
1	Average Word Length
2	Average Sentence Length
3	Paragraph Length by Characters
4	Paragraph Length by Tokens
5	Paragraph Length by Sentences
6	Type-Token Ratio
7	Average Syllables
8	Flesch-Kincaid Readability Score
9	Stopwords Count
10	Function Words Count
11	Punctuation Marks Ratio

Table 2. Features used in Stylistic Encoder.

.

These features represent different stylistic aspects of paragraphs in the following ways. For example, length-related features (Paragraph Length by Characters, Tokens, and Sentences) provide information about the overall length and density of the paragraphs. They help identify changes in the writing style based on variations in paragraph lengths. Complexity-related features, including Average Word Length, Sentence Length, Average Syllables, and Type-Token Ratio, capture the complexity and sophistication of the writing style. These features provide insights into vocabulary richness, syntactic complexity, and level of detail in the text. Readability-related features (the Flesch-Kincaid Readability Score), estimate the ease of understanding the text. They consider factors like sentence length and average syllables per word, indicating changes in the readability and comprehension level of the writing style. Language Usage-related features, such as Stopwords Count, Function Words Count, and Punctuation Marks Ratio, reflect the usage and style of language. These features provide insights into the formality, informality, syntactic structures, and emphasis in the writing style.

After extracting the stylistic features of paragraphs within a pair, we calculate the stylistic difference by subtracting the feature vectors of the two paragraphs and taking the absolute value. This difference vector represents the stylistic distinction between the pair. We then use this difference vector, along with the actual label, as input to a logistic regression model for fitting and prediction.

By categorizing the stylistic features in this manner, we gain a comprehensive understanding

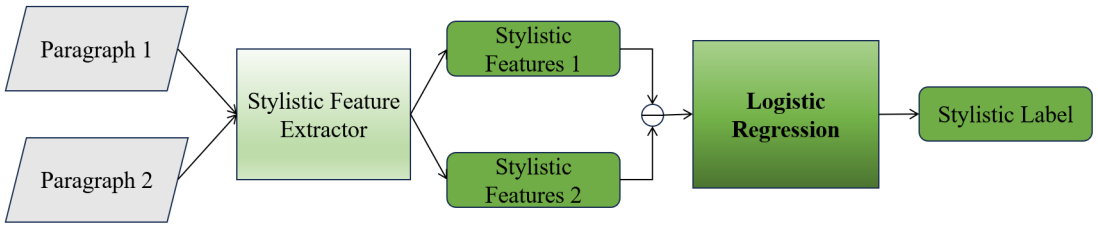

Figure 5. Stylistic Encoder in TriS

of their specific contributions to the writing style detection task. Each feature captures different aspects of the writing style, helping to identify and analyze stylistic variations in the text.

4.4 Fusion Recognizer

In the Fusion Recognizer module, our objective is to integrate the outputs of the three encoders (Semantic Encoder, Syntactic Encoder, and Stylistic Encoder) to provide the final prediction of whether a writing style change has occurred. For a given paragraph pair, the Fusion Recognizer combines the semantic, syntactic, and stylistic features as follows:

1. Semantic feature integration: The Semantic Encoder extracts and calculates the Euclidean distance $\boldsymbol{D_{sem}}$ between the semantic features of the input paragraph pair by Equation 1. This distance represents the semantic similarity or dissimilarity between the two paragraphs.
2. Syntactic feature integration: The Syntactic Encoder extracts and calculates the cosine similarity $\boldsymbol{C_{syn}}$ between the syntactic features of the input paragraph pair by Equation 4. This similarity indicates the degree of syntactic similarity between the two paragraphs.
3. Stylistic feature integration: The Stylistic Encoder predicts a label $\boldsymbol{L_{sty}}$ (0 or 1) based on the 11 stylistic features extracted from the input paragraph pair. These features capture various aspects of the writing style. The predicted label represents whether a writing style change is detected.

To obtain the final prediction label, the Fusion Recognizer first computes the fusion score using the equation:

$$FS = \lambda_{sem} \cdot (D_{sem}^2 - \gamma_{sem}) - \lambda_{syn} \cdot (C_{syn} - \gamma_{syn}) + \lambda_{sty} \cdot (L_{sty} - 0.5) \tag{5}$$

Here, λ_{sem}, λ_{syn}, and λ_{sty} represent the weights assigned to the semantic, syntactic, and stylistic components, respectively. And γ_{sem} and γ_{syn} refer to the correction terms applied to D_{sem}^2 and CS_{syn}, respectively. Then, the final prediction label is determined as follows:

$$Prediction = \frac{1 + sign(FS)}{2} \tag{6}$$

5 Experiments

Model Details

In our constructed model [1], all MLP (Multi-Layer Perceptron) hidden layers have a size of 1024,

[1] Available at https://github.com/X1angyuLu/TriS

and the output layers have a size of 512. Layer normalization is applied at the end of each MLP to aid convergence. The difference lies in the configuration of the MLPs used in each encoder.

In the Semantic Encoder, there are three fully connected layers in the MLP used to fine-tune the pooler output and hidden states. ReLU activation function is applied after each fully connected layer, and dropout is utilized to prevent overfitting. In the Syntactic Encoder, the MLP consists of two fully connected layers, and ReLU activation function is used after each layer. As for the Stylistic Encoder, we utilize the LogisticRegression model from the scikit-learn library [37]. A sufficient number of iterations is set to ensure convergence of the model.

Training Settings

During training, we utilize the Adam optimizer for all encoders. However, different training strategies are applied to each encoder.

For the Semantic Encoder, we freeze the BERT model and only train the parameters of the MLP. In the initial 5 epochs, a learning rate of 0.0001 is used to accelerate the training process. This learning rate is then reduced to 0.000001 for the remaining 15 epochs. The training loop loads a single text file at a time and calculates the loss multiple times based on the number of paragraph pairs within the file. To mitigate the potential negative impact of varying sample sizes on model optimization, we take the average loss at the text level as the loss for each batch and the threshold in Equation 2 is set to 4.

In the Syntactic Encoder, all parameters are set as trainable. The learning rate remains constant at 0.0000001 throughout the training process. A batch size of 1 is used, and the model is trained for 20 epochs on the training set.

For the Stylistic Encoder, we pre-compute the stylistic feature difference vectors for all paragraph pairs and feed them as inputs to the LogisticRegression model for fitting.

Evaluation

In our evaluation, we utilized a medium-sized validation set to assess the performance of the random baseline model, Semantic Encoder, Syntactic Encoder, Stylistic Encoder, and the full model. Considering the balanced distribution of positive and negative samples in the validation set, we selected accuracy and F1 score as our evaluation metrics. The accuracy provides a straightforward measure of the model's correct predictions, while the F1 score helps to address any potential hidden imbalances in the dataset and mitigate bias.

For the baseline model, due to its stochastic nature, we conducted 5 tests and calculated the average accuracy and F1 score as its final performance. For the Semantic Encoder and Syntactic Encoder, we employed Equation 1 and Equation 4, respectively, as the evaluation criteria. Specifically, we maintained the threshold of 4, as used during training, for the Semantic Encoder to determine whether a writing style change occurred. Regarding the Stylistic Encoder, since its predictions are already in the form of explicit labels, no further modifications were made.

In the full model, the output results are generated by the Fusion Recognizer. More specifically, in Equation 5, the weights λsem, λsyn, and λsty are set to 0.9, 0.3, and 0.35, respectively. The correction thresholds γsem and γ_{syn} are set to 3.1 and 0.9, respectively.

These weight values and correction thresholds are carefully chosen to balance the contributions of the semantic, syntactic, and stylistic components in the final prediction. By adjusting these parameters, we can fine-tune the model to emphasize certain aspects of the feature representations and improve its performance in detecting writing style changes.

6 Results

The results of our evaluation are summarized by Table 3. Overall, the results indicate that all three encoders outperform the baseline model in terms of accuracy, suggesting that they have learned underlying patterns from their respective inputs for writing style change detection. Specifically, the Semantic Encoder shows the most significant improvement over the baseline, demonstrating its ability to capture semantic features relevant to writing style. The Syntactic Encoder performs slightly lower than the Semantic Encoder but still shows promising results in capturing

syntactic patterns. On the other hand, the performance of the Stylistic Encoder is relatively lower, suggesting that the considered stylistic features alone may not be sufficient for accurate writing style detection.

However, the best performance is achieved by the full model, which combines the outputs of all three encoders. By integrating semantic, syntactic, and stylistic features, the TriS achieves the highest accuracy and F1-score of 0.742 and 0.709, respectively. This indicates the importance of leveraging multiple aspects of writing style for improved detection accuracy.

Model	Accuracy	F1-Score
Random Model (baseline)	0.504	0.471
Semantic Encoder	0.729	0.632
Syntactic Encoder	0.622	0.519
Stylistic Encoder	0.583	0.439
TriS (full model)	**0.742**	**0.709**

Table 3. Accuracy and F1 score of different models on medium task.

7 Discussion

The acquired results demonstrate that the TriS, showing an accuracy of 0.742, greatly outperforms the encoders that solely concentrate on semantic (0.729), syntactic (0.622), or stylistic (0.583) aspects. This supremacy can be credited to the fusion recognizer's multifaceted analysis of text, considering multiple significant dimensions of writing style, rather then confining the analysis to a single aspect. Previous research also demonstrated the advantage of fusion over simple concatenation by testing it on the validation set, resulting in a 10% decrease in accuracy [27]. Furthermore, the higher accuracy and F1-score also highlights the synergistic effect of individual encoders on the final fusion outcome. By complementing each other, the conspicuous weaknesses in certain encoders are mitigated, while the unique strengths of others are optimally leveraged. For instance, as the complexity of the task augments and the scope of topic information diminishes, the performance of the semantic encoder, a principal component, tends to degrade, which necessitates the supplementary support of syntactic and stylistic encoders to maintain overall performance.

The accuracies and F1 scores obtained from these three encoders and recognizer demonstrate that the different types of text-derived features contribute to effectiveness of the full model with varying degrees. The semantic encoder achieves a highest F1 score and an accuracy closest to the full model, which supports the notion that the extracted semantic features significantly contribute to the effectiveness of the writing style change detection task. It further substantiates the advantageous performance of BERT-generated embeddings in solving tasks of this nature [11]. It is noteworthy that the medium difficulty level dataset we employed exhibits subtle thematic changes associated with transitions between paragraphs of different styles, which can potentially account for the Semantic Encoder's superior capability. The accuracy and F1 scores of Syntactic Encoder which surpass the baseline indicate that the attributes captured through inputted POS tag sequences of paragraphs, such as sentence structures or usage patterns of different word types, can possibly aid in accomplishing the writing style change detection task. In contrast to the two previous elaborated encoders, Stylistic Encoder's better accuracy but lower F1 score than the Random Model manifest a unanticipated fact that the performance of the Stylistic Encoder is even inferior to the baseline, illustrating a possible intrinsic truth that features that we use in Stylistic encoder such as Average Word Length, Stopwords Count, etc. are not very beneficial in the authorship change detection task, which contradicts with previous studies using he frequency of punctuation marks [1, 38]. However, despite the underwhelming performance of the Stylistic Encoder, it is still

possible for it to have potential contributions to the full model in consideration of the optimal accuracy and F1 score shown by the comprehensive model.

We observed that the accuracy of all three encoders is considerably higher than their respective F1 scores. This imbalance persists, albeit to a lesser extent, in the full model as well. Accuracy is a common statistical indicator for evaluating the performance of classification models, measuring the proportion of samples that the model correctly classifies. On the other hand, the F1 score, which also a widely used metrics which comprehensively considers the Precision and Recall of the model. The higher accuracies and relatively lower F1 scores of these three encoders imply that all these three encoders perform poorly in certain categories and possess latent tendencies to classify samples as either positive or negative. In other words, although the model has a high proportion of correctly classified samples (i.e., high accuracy), it does not balance precision and recall well (lower F1 score). The full model leverages strengths and compensates for weaknesses of these three encoders by the Fusion Recognizer. This recognizer helps to partly balance the accuracy and F1 score, revealing a decrease in the latent tendencies present in the individual encoders.

The current project is subject to some limitations. The first constraint is the over-reliance on the semantic encoder, which is a primary contributor to the fusion model's performance. This excessive dependence leads to a dramatic improvement in semantic accuracy in tasks with extensive thematic shifts, while also resulting in a substantial decline in performance when confronted with restricted topic variations in more complex tasks. Consequently, future research should firstly aim to embrace more features at the sentence or discourse levels, such as the ones demonstrated in [38, 39]). Besides, to fully leverage the semantic information and thematic data, topic modelling techniques such as Latent Semantic Indexing can be adopted to excel the superficial textual analysis, as suggested in [40]. Secondly, our model used the PAN datasets derived from Reddit without any additional extrinsic data, leaving its applicability to diverse text genres uncertain. This underlines the need for more large-scale datasets to test its adaptability and robustness.

8 Conclusion

In this paper, we introduced the **TriS** model for multi-author writing style change detection, consisting of Semantic, Syntactic, and Stylistic encoders. Our model, trained on the PAN Multi-Author Writing Style Analysis 2023 datasets, showcased its effectiveness in capturing stylometric patterns associated with style shifts in the constrained topic setting. The fusion of these encoders resulted in superior performance, achieving an accuracy of 0.742 and an F1-score of 0.709 on the validation set. TriS demonstrates its potential as a useful tool for analyzing and detecting writing style changes in multi-author scenarios.

References

[1] E. Zangerle, M. Mayerl, M. Potthast, and B. Stein, "Overview of the Style Change Detection Task at PAN 2022," in *CLEF 2022 Labs and Workshops, Notebook Papers*, G. Faggioli, N. Ferro, A. Hanbury, and M. Potthast, Eds. CEUR-WS.org, Sep. 2022. [Online]. Available: http://ceur-ws.org/Vol-3180/paper-186.pdf

[2] E. Stamatatos, "Authorship verification: a review of recent advances," *Research in Computing Science*, vol. 123, pp. 9–25, 2016.

[3] E. Stamatatos, "A survey of modern authorship attribution methods," *Journal of the American Society for information Science and Technology*, vol. 60, no. 3, pp. 538–556, 2009.

[4] K. Lagutina, N. Lagutina, E. Boychuk, I. Vorontsova, E. Shliakhtina, O. Belyaeva, I. Paramonov, and P. Demidov, "A survey on stylometric text features," in *2019 25th Conference of Open Innovations Association (FRUCT)*. IEEE, 2019, pp. 184–195.

[5] R. Ramezani, N. Sheydaei, and M. Kahani, "Evaluating the effects of textual features on authorship attribution accuracy," in *ICCKE 2013*. IEEE, 2013, pp. 108–113.

[6] M. Cristani, G. Roffo, C. Segalin, L. Bazzani, A. Vinciarelli, and V. Murino, "Conversationally-inspired stylometric features for authorship attribution in instant messaging," in *Proceedings of the 20th ACM international conference on Multimedia*, 2012, pp. 1121–1124.

[7] H. Gómez-Adorno, J.-P. Posadas-Duran, G. Ríos-Toledo, G. Sidorov, and G. Sierra, "Stylometry-based approach for detecting writing style changes in literary texts," *Computación y Sistemas*, vol. 22, no. 1, pp. 47–53, 2018.

[8] P. Plecháč, K. Bobenhausen, and B. Hammerich, "Versification and authorship attribution. a pilot study on czech, german, spanish, and english poetry," *Studia Metrica et Poetica*, vol. 5, no. 2, pp. 29–54, 2018.

[9] E. Ferracane, S. Wang, and R. Mooney, "Leveraging discourse information effectively for authorship attribution," in *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2017, pp. 584–593.

[10] R. Deibel and D. Löfflad, "Style change detection on real-world data using an lstm-powered attribution algorithm." in *CLEF (Working Notes)*, 2021, pp. 1899–1909.

[11] A. Iyer and S. Vosoughi, "Style change detection using bert." *CLEF (Working Notes)*, vol. 93, p. 106, 2020.

[12] T.-M. Lin, C.-Y. Chen, Y.-W. Tzeng, and L.-H. Lee, "Ensemble pre-trained transformer models for writing style change detection." CLEF, 2022.

[13] S. Al-Shamasi and M. Menai, "Ensemble-based clustering for writing style change detection in multi-authored textual documents." CLEF, 2022.

[14] J. Collins, D. Kaufer, P. Vlachos, B. Butler, and S. Ishizaki, "Detecting collaborations in text comparing the authors' rhetorical language choices in the federalist papers," *Computers and the Humanities*, vol. 38, pp. 15–36, 2004.

[15] S. Meyer zu Eissen, B. Stein, and M. Kulig, "Plagiarism detection without reference collections," in *Advances in Data Analysis: Proceedings of the 30 th Annual Conference of the Gesellschaft für Klassifikation eV, Freie Universität Berlin, March 8–10, 2006*. Springer, 2007, pp. 359–366.

[16] O. De Vel, A. Anderson, M. Corney, and G. Mohay, "Mining e-mail content for author identification forensics," *ACM Sigmod Record*, vol. 30, no. 4, pp. 55–64, 2001.

[17] Eva Zangerle, M. Mayerl, M. Potthast, and B. Stein, "Pan23 multi-author writing style analysis," Mar. 2023. [Online]. Available: https://doi.org/10.5281/zenodo.7729178

[18] E. Zangerle, M. Mayerl, M. Potthast, and B. Stein, "Overview of the Style Change Detection Task at PAN 2021," in *CLEF 2021 Labs and Workshops, Notebook Papers*, G. Faggioli, N. Ferro, A. Joly, M. Maistro, and F. Piroi, Eds. CEUR-WS.org, 2021.

[19] A. Dumalus and P. Fernandez, "Authorship attribution using writer's rhythm based on lexical stress," in *11th Philippine Computing Science Congress, Naga City, Philippines*, 2011.

[20] R. Hou and C.-R. Huang, "Robust stylometric analysis and author attribution based on tones and rimes," *Natural Language Engineering*, vol. 26, no. 1, pp. 49–71, 2020.

[21] F. Sebastiani, "Machine learning in automated text categorization," *ACM computing surveys (CSUR)*, vol. 34, no. 1, pp. 1–47, 2002.

[22] H. Ramnial, S. Panchoo, and S. Pudaruth, "Authorship attribution using stylometry and machine learning techniques," in *Intelligent Systems Technologies and Applications: Volume 1.* Springer, 2016, pp. 113–125.

[23] Y. Sari, "Neural and non-neural approaches to authorship attribution," Ph.D. dissertation, University of Sheffield, 2018.

[24] T. Kumarage, J. Garland, A. Bhattacharjee, K. Trapeznikov, S. Ruston, and H. Liu, "Stylometric detection of ai-generated text in twitter timelines," *arXiv preprint arXiv:2303.03697*, 2023.

[25] M. Jin and M. Jiang, "Text clustering on authorship attribution based on the features of punctuations usage," in *2012 IEEE 11th International Conference on Signal Processing*, vol. 3. IEEE, 2012, pp. 2175–2178.

[26] K. Safin and A. Ogaltsov, "Detecting a change of style using text statistics," *Working Notes of CLEF*, 2018.

[27] M. Hosseinia and A. Mukherjee, "A parallel hierarchical attention network for style change detection: Notebook for pan at clef 2018." in *CLEF (Working Notes)*, 2018.

[28] C. Zuo, Y. Zhao, and R. Banerjee, "Style change detection with feed-forward neural networks." *CLEF (Working Notes)*, vol. 93, 2019.

[29] S. Nath, "Style change detection by threshold based and window merge clustering methods." in *CLEF (Working Notes)*, 2019.

[30] Z. Zhang, Z. Han, L. Kong, X. Miao, Z. Peng, J. Zeng, H. Cao, J. Zhang, Z. Xiao, and X. Peng, "Style change detection based on writing style similarity," *Training*, vol. 11, pp. 17–051, 2021.

[31] Z. Zhang, Z. Han, and L. Kong, "Style change detection based on prompt." CLEF, 2022.

[32] L. Graner and P. Ranly, "An unorthodox approach for style change detection," 2022.

[33] J. Bevendorff, M. Chinea-Ríos, M. Franco-Salvador, A. Heini, E. Körner, K. Kredens, M. Mayerl, P. Pęzik, M. Potthast, F. Rangel *et al.*, "Overview of pan 2023: Authorship verification, multi-author writing style analysis, profiling cryptocurrency influencers, and trigger detection," in *European Conference on Information Retrieval.* Springer, 2023, pp. 518–526.

[34] R. Singh, J. Weerasinghe, and R. Greenstadt, "Writing style change detection on multi-author documents." in *CLEF (Working Notes)*, 2021, pp. 2137–2145.

[35] E. Zangerle, M. Mayerl, G. Specht, M. Potthast, and B. Stein, "Overview of the style change detection task at pan 2020." *CLEF (Working Notes)*, vol. 93, 2020.

[36] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.", 2009.

[37] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[38] C. A. Rodríguez-Losada and D. Castro-Castro, "Three style similarity: sentence-embedding, auxiliary words, punctuation," 2022.

[39] V. Oloo, L. D. Wanzare, and C. Otieno, "An optimal feature set for stylometry-based style change detection at document and sentence level," 2022.

[40] N. Potha and E. Stamatatos, "Intrinsic author verification using topic modeling," in *Proceedings of the 10th Hellenic Conference on Artificial Intelligence*, 2018, pp. 1–7.