Enhancing Authorship Verification using Sentence-Transformers

Notebook for PAN at CLEF 2023

Momen Ibrahim[1], Ahmed Akram[1], Mohammed Radwan[1], Rana Ayman[1], Mustafa Abd-El-Hameed[1], Nagwa El-Makky[1] and Marwan Torki[1]

[1]*Computer and Systems Engineering Department, Faculty of Engineering, Alexandria University, Alexandria, Egypt*

Abstract

Authorship verification is a growing field of research that aims to determine whether two texts were written by the same author or different authors. In this paper, we describe our system for the PAN@CLEF 2023 Authorship Verification challenge [1] which requires solving the task on a cross-Discourse Types and open-set collection of essays (written discourse), emails (written discourse), Interviews (spoken discourse), and Speech transcriptions (spoken discourse). We use Sentence-Transformers which is a popular framework for generating sentence embedding based on the idea of using pre-trained language models, such as BERT (Bidirectional Encoder Representations from Transformers) to capture the semantic features of different authors' documents. As a result of studying multiple sentence-transformers, we selected **all-MiniLM-L12-v2** model and achieved the first rank in PAN 23 with an overall score of **62.3%** on the **testing set**.

Keywords

Authorship Verification, pre-trained language models, Sentence-Transformers, SBERT

1. Introduction

Authorship Verification (AV) is a specialized area in the field of digital texts that focuses on analyzing and comparing the unique stylistic and linguistic characteristics present in multiple texts. The primary objective of AV is to ascertain whether these texts were authored by the same individual. Its significance has been magnified in recent times due to the abundance of digital content accessible through digital libraries, online journalism platforms, and social media networks.

CLEF 2023 – Conference and Labs of the Evaluation Forum, September 18–21, 2023, Thessaloniki, Greece

es-momen.ibrahim2019@alexu.edu.eg (M. Ibrahim); eng-ahmed.akram2018@alexu.edu.eg (A. Akram); es-mohamed.radwan2000@alexu.edu.eg (M. Radwan); es-rana.hussein2023@alexu.edu.eg (R. Ayman); es-mostafahmed22@alexu.edu.eg (M. Abd-El-Hameed); nagwamakky@alexu.edu.eg (N. El-Makky); mtorki@alexu.edu.eg (M. Torki)

https://github.com/Momen-Ibrahim-Fawzy/ (M. Ibrahim); https://github.com/Akramovic1/ (A. Akram); https://github.com/MohamedRadwan2000/ (M. Radwan); https://github.com/RanaAyman/ (R. Ayman); https://github.com/MostafaAhmed660/ (M. Abd-El-Hameed)

0009-0001-7287-0705 (M. Ibrahim); 0009-0007-8424-2240 (A. Akram); 0009-0005-1661-9144 (M. Radwan); 0009-0004-8853-656X (R. Ayman); 0009-0007-5179-9632 (M. Abd-El-Hameed); 0000-0002-9571-7543 (N. El-Makky); 0000-0002-6149-1718 (M. Torki)

AV serves a crucial purpose in various domains where automatic verification of document authorship is essential. With the exponential growth of digital texts available online, the need to accurately determine the authorship of documents has become increasingly prominent. This is particularly relevant in contexts where the evaluation of researchers relies on the impact and quantity of their publications. Similarly, public figures often face scrutiny based on their social media posts. In such scenarios, AV offers a valuable tool to establish the authenticity and authorship of documents.

The digital landscape, encompassing digital libraries, online journalism, and social media platforms, has witnessed a surge in textual data. However, along with this surge, there has also been an upswing in online crimes. AV plays a pivotal role in addressing these challenges by identifying instances of fraudulent activities, such as phishing emails, and detecting cases of plagiarism. By analyzing stylistic features embedded in texts, including writing style, genre, temperament, sentiment, native language, and gender, AV provides insights into the unique attributes and traits of authors.

In summary, AV is a specialized field that utilizes advanced techniques to analyze and compare the distinctive patterns and features present in multiple texts, with the aim of determining whether they share a common author. The increasing availability of digital texts and the need for accurate document authorship verification in various domains underscore the significance of AV in today's digital landscape.

In the authorship verification task at PAN@CLEF-2023 [2] they introduced more challenging scenarios where each author verification case considers texts from different discourse types. This edition focuses on cross-discourse type authorship verification, including both written (essays and emails) and spoken language (interviews and speech transcriptions). The corpus consists of texts from around 100 individuals, all native English speakers aged 18-22. The topics of the texts are unrestricted, and the level of formality may vary within each discourse type.

Most of the documents in the training set of PAN23 are emails, interviews and speech-transcriptions which are relatively short documents. This inspired us to use Sentence-Transformers specially SBERT [3] due to their power in semantic similarity tasks in addition to their efficiency. We evaluated different pre-trained models for their quality to embedded sentences.

As, the used models are trained for short sentences, this directed us to chunk the documents in the pairs to fit in the max-sequence length of the used pre-trained models. We tried different chuncking techniques. These chunking techniques will be explained in section 4.2.

This paper is structured as follows: Section 2 provides a background and related work on authorship verification. Section 3 describes the dataset used in the PAN@CLEF 2023 challenge, including the different discourse types and the number of texts in each category. Section 4 explains the preprocesing step on the dataset and the chunking technique for handling long texts. Section 5.1 presents the methodology used in our system, including the use of Sentence-Transformers for generating sentence embeddings. Section 6 discusses experiments on other models, including a model that uses only stylometric features. Section 7 discusses the results and performance of these models on the validation set. Finally, Section 8 concludes the paper.

2. Background and Related work

Reimers and Gurevych (2019) [3] propose a new method for generating sentence embeddings, which are vector representations of sentences that capture their semantic meaning. The authors use a Siamese neural network architecture, based on the BERT model, to generate embeddings that are optimized for use in tasks such as semantic similarity and paraphrase detection. The model is trained on a large corpus of text data, and the resulting embeddings are shown to outperform previous state-of-the-art methods on several benchmark datasets. The authors also introduce a new technique for fine-tuning the model on smaller datasets, which further improves its performance on specific tasks. In PAN@CLEF 2021, Peng et al. (2021) [4] propose a method that utilizes a pre-trained model to encode text information to solve the authorship verification in the PAN@CLEF 2021. To resolve the problem of long text encoding, the method proposed is to split long texts into short texts that a pre-trained model, BERT, can encode. The classification model achieved the highest c@1 and F1-score on the small dataset of PAN Authorship Verification datasets. Accordingly, the approach described can encode long text information efficiently in long text pairs.

Stylometric feature extraction is another approach proposed by, Weerasinghe et al. (2021) [5] The authors demonstrate the impact of different feature representations on the performance of the proposed method of computing the absolute difference between the feature vectors as input to the logistic regression classifier for each document pair, they evaluate several feature sets, including word and character n-grams, punctuation usage, and syntactic features.

BERT-like transformers for authorship verification also perform well in this task. Manolache et al., (2021) [6] explore the use of them for authorship verification, The authors evaluate several transformers on the PAN-2020 dataset and find that they achieve high performance, but also they show through Integrated Gradients XAI technique that, they rely on topical clues rather than stylistic features. To address this issue, the authors propose new splits for the dataset that ensure topic and author diversity and show that they improve the models' domain generalization ability. The authors also introduce a new dataset, DarkReddit, that contains texts from different domains and genres, and use it to test the models in a low-data regime.

Galicia et al.,(2022)[7] proposed a Graph-based Siamese Network approach for the authorship verification task at PAN 2022 [8]. They also introduced a way to split PAN22 dataset to ensure that the training and validation sets are both balanced and author-disjoint. This is important to avoid misleading results that may arise from data leakage or overfitting.

3. Data Description

The evolution of authorship verification tasks in previous editions of PAN@CLEF competitions is as follows. Initially, the task focused on comparing writing styles in different languages and genres. Later, cross-domain authorship verification using fan-fiction texts was explored and found to be feasible across different fandom. In the 2022 edition, more challenging scenarios were introduced, involving cross-DT authorship verification, where texts belonged to different discourse types (DTs), all within the realm of written language. The current edition focuses on cross-discourse type authorship verification, where both written language (essays and emails) and spoken language (interviews and speech transcriptions) are included as discourse types.

The dataset provided by PAN is a collection of English texts from different Discourse Types (DT). The goal of the task is to determine if two texts from different DTs are written by the same author. The dataset has a tag for each pair of texts indicating the authorship, the author, and the discourse type of each text. Since we augment the training set of PAN 23 by the training set of PAN 22. The next subsections describe both of them.

3.1. Authorship Verification Dataset at PAN 2022

In that dataset the organizers provide cross-DT authorship verification cases using DTs corresponding to written language.
The discourse types and their corresponding number of texts in the train dataset are:

- `Essays (written discourse)`: 2986
- `Emails (written discourse)`: 10116
- `Text messages (written discourse)`: 9446
- `Business memos (written discourse)`: 1980

The corpus consists of texts from approximately 100 individuals who are native English speakers and share a similar age range of 18-22. The text samples cover a wide range of topics, without any restrictions, and within each discourse type, there can be variations in the level of formality.

Table 1
Combinations of Different Discourse Types for Each Pair in PAN 22

Pair Discourse Type Combination	Number of Pairs
('Email', 'Text message')	7484
('Essay', 'Email')	1618
('Business memo', 'Text message')	780
('Essay', 'Text message')	1182
('Email', 'Business memo')	1014
('Essay', 'Business memo')	186

3.2. Authorship Verification Dataset at PAN 2023

In that dataset the organizers provide cross-DT authorship verification cases using DTs not only corresponding to written language but also to spoken language.
The discourse types and their corresponding number of texts in the train dataset are:

- `Essays (written discourse)`: 2594
- `Emails (written discourse)`: 7054
- `Interviews (spoken discourse)`: 6090
- `Speech transcriptions (spoken discourse)`: 1934

And also, the corpus consists of texts from approximately 100 individuals who are native English speakers and share a similar age range of 18-22. The text samples cover a wide range of topics, without any restrictions, and within each discourse type, there can be variations in the level of formality.

Table 2
Combinations of Different Discourse Types for Each Pair in PAN 23

Pair Discourse Type Combination	Number of Pairs
('Speech transcription', 'Email')	1036
('Email', 'Interview')	4564
('Essay', 'Email')	1454
('Essay', 'Interview')	884
('Speech transcription', 'Interview')	642
('Essay', 'Speech transcription')	256

3.3. Data Augmentation

Based on the strong similarity between the two tasks of PAN 22 and PAN 23, and because the number of pairs in the new task of PAN 23 is too small, we decided to combine both train datasets into one train dataset.

Table 3
Number of problems, texts, authors and topics in the PAN 22 and PAN 23 train datasets.

	Problems	Unique Texts	Authors	Discourse Types
PAN 22 Train Dataset	12264	1046	56	Essays, Emails, Text messages and Business memos
PAN 23 Train Dataset	8836	886	56	Essays, Emails, Interviews and Speech transcriptions
Combined Dataset	21100	1932	112	Essays, Emails, Text messages, Business memos, Interviews and Speech transcriptions

That new dataset has the following number of texts for each discourse type:

- `Essays (written discourse)`: 5580
- `Emails (written discourse)`: 17170
- `Text messages (written discourse)`: 9446
- `Business memos (written discourse)`: 1980
- `Interviews (spoken discourse)`: 6090
- `Speech transcriptions (spoken discourse)`: 1934

And the new number of combinations in that combined dataset are shown in Table 4.

Table 4
Combinations of Different Discourse Types for Each Pair in the Combined Dataset

Pair Discourse Type Combination	Number of Pairs
('Email', 'Text message')	7484
('Essay', 'Email')	3072
('Business memo', 'Text message')	780
('Essay', 'Text message')	1182
('Email', 'Business memo')	1014
('Essay', 'Business memo')	186
('Speech transcription', 'Email')	1036
('Email', 'Interview')	4564
('Essay', 'Interview')	884
('Speech transcription', 'Interview')	642
('Essay', 'Speech transcription')	256

4. Data Preprocessing

This section describes the pre-processing operations we did on the combined dataset.

4.1. Data Splitting

We used an author-disjoint method similar to the one introduced in [7] to split the dataset into training and validation sets for our model evaluation. This means that no text from one set has the same author as any text from another set. Since the dataset had 21,100 problems and only 112 authors, this method resulted in unbalanced sets, with more positive problems than negative ones. To address this issue, we generated new negative instances by pairing texts from different authors. We followed these steps: Let A and B be the subsets of texts from each set grouped by author. We obtained positive and negative pairs by computing the Cartesian products $P = A \times A$ and $N = A \times B$ respectively. Then, we removed pairs that had the same DT, and finally randomly sampled positive and negative pairs from P and N sets to balance the training and validation sets. The final dataset had an equal number of true and false problems. The total number of problems and the number of problems in the positive class, the number of texts and authors on each partition are shown in Table 5.

Table 5
Total number of problems, number of problems in the positive class, number of texts and number of authors on our splits.

Split	Total	Positive	Unique Texts	Authors
Train	18968	9484	1740	101
Val	2132	1066	192	11

4.2. Texts Chunking

The sentence transformer is based on a Siamese Network that takes two texts (documents) as input. The length of document 1 and/or document 2 can be larger than the max-sequence-length,K, allowed by the sentence -transformer model. So, for training, we chunk each document into a set of chunks , each of size K. We combine the first 2 chunks of the 2 documents, the second 2 chunks, etc. This creates for each original pair, M pairs, where M is the number of chucks of the larger document. The chunking at evaluation time is described in section 5.2.

5. Model Configuration

5.1. Model Architecture

We chose all-MiniLM-L12-v2 as our model. It is a Siamese network, based on BERT-like pre-trained model. It belongs to the family of sentence transformers. It uses a contrastive learning objective and has a max sequence length of 256 tokens. It is a lightweight model with a size of 120 MB. However, it provides high-quality semantic similarity in addition to its efficiency. Figure 1 shows the model architecture at training and at inference times.

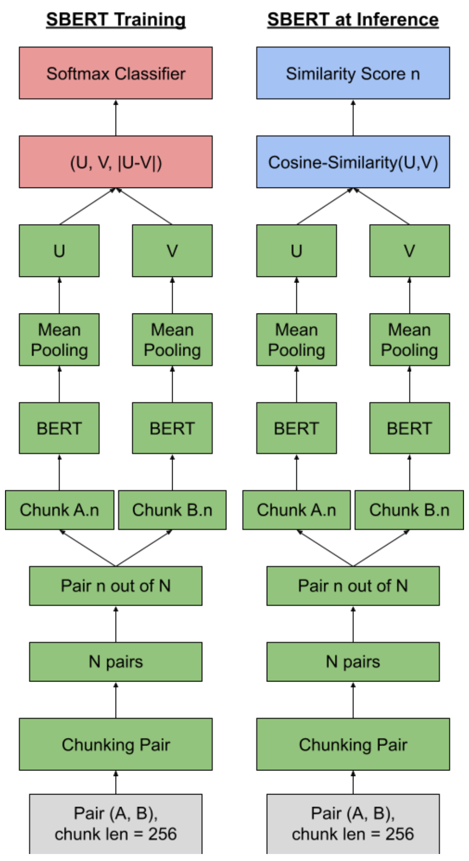

Figure 1: Model Architecture

5.1.1. Embedding Layer

It is the first layer of the model, responsible for converting input tokens into dense vector representations (embeddings). BERT-like models typically use a combination of token embeddings, segment embeddings, and position embeddings. The token embeddings are learned representations specific to each token in the model's vocabulary. Segment embeddings are used to distinguish different sentences, and position embeddings give the model a sense of the order of words in a sentence since transformers don't have a built-in understanding of sequence order.

5.1.2. Transformer Encoder Layers

These are the core of the model. Each transformer encoder layer consists of two sub-layers:

- **Self-Attention Mechanism (Multi-Head Attention)**: The purpose of the self-attention mechanism is to compute a representation of each word that takes into account the influence of other words in the sentence.
  In each attention head, for every word, the model calculates a score (attention score) that signifies how much focus should be placed on other words. These scores determine how much each word will contribute to the final representation of the current word.
  The mechanism performs the following steps:
  - **Query, Key, and Value Vectors**: Each input word is transformed into Query, Key, and Value vectors using learned linear transformations.
  - **Attention Score**: An attention score is computed for each pair of words. This score is calculated as the dot product of the Query vector of the current word and the Key vector of the other word, followed by a scaling operation (dividing by the square root of the dimension of the key vector).
  - **Softmax Normalization**: The attention scores are then passed through a softmax function to obtain the weights, which ensures they are positive and sum to 1.
  - **Weighted Sum**: Finally, a weighted sum of the Value vectors of all words is computed, where the weights are the softmax-normalized attention scores.

  This process is done in parallel in each attention head. Each head may potentially learn to pay attention to different types of connections between words.
  The output from each head is concatenated and linearly transformed to produce the final output of the multi-head attention layer.
- **Feed-Forward Neural Network**: The output from the self-attention mechanism for each position passes through this layer. It's a simple feed-forward neural network that consists of two fully connected layers with a ReLU activation function in between. This network doesn't have recurrent or convolutional connections, and it operates independently on each position, treating each position identically. This is where most of the model's parameters reside.

Apart from these, there are also residual connections around both the self-attention and FFNN components, followed by layer normalization. The residual connections help in preventing the vanishing gradient problem and enable the model to be deeper. Layer normalization is used to stabilize the network's learning process and reduce training time.

These components work together to create a powerful model that can capture complex patterns in the input data. The stacking of multiple layers allows the model to learn more abstract and high-level features as the information propagates up the layers.

5.1.3. Pooler Layer

Once the transformer layers have processed the input, the final step in the model architecture is to convert the output into a fixed-size sentence embedding. This is achieved by applying a pooling operation to the output of the transformer layers. The output is a single vector that represents the entire input sentence.

5.2. Evaluation

As can be seen from the model architecture (Figure 1) the output at evaluation time is the cosine similarity of two chunks from the two documents. At the evaluation, the first chunk of K tokens of each document is compared, the second grouping of K tokens from each document is compared, etc. Finally, we take the average of the M scores as the final cosine similarity. Since the cosine similarity range is from -1 to 1, we re-scale the score to be from 0 to 1 using the approach proposed in the baseline model of PAN 22. During training, we optimize two threshold values p1 and p2. All evaluation scores lower than p1 correspond to negative answers, and all evaluation scores greater than p2 are scaled to positive answers. The remaining scores (between p1 and p2) are set to 0.5 and are left unanswered.

5.3. Hyper-parameter Tuning

Hyperparameter tuning plays a crucial role in optimizing the performance of the model. In this section, we outline the hyperparameters used in our experiments and describe the process of tuning them.

5.3.1. Model Configuration

We employed the "all-MiniLM-L12-v2" model architecture (with 256 tokens as max-seq-length) for our experiments. This model has shown promising results in various natural language processing tasks and is pre-trained on a large corpus of text data.

5.3.2. Hyper-parameters

The list below includes the hyper-parameters considered, together with the values selected after fine tuning, using the validation set.

- **`Chunk Length`**: We set the chunk length to 256, which determines the maximum length of text chunks used during training and evaluation.
- **`Batch Size`**: The batch size was set to 32, which determines the number of training samples processed in parallel during each iteration.
- **`Distance Metric`**: We used the cosine distance metric to measure the similarity between text embeddings.

- `Loss Function`: We utilized the ContrastiveLoss function as the loss function for training the model.
- `Epochs`: We trained the model for 189 epochs, representing the number of times the entire dataset was iterated during training.
- `Scheduler`: The learning rate scheduler was set to 'warmuplinear', which gradually increases the learning rate during the warm-up phase and then linearly decays it.
- `Warmup Steps`: The warm-up steps were set to 0, indicating that no warm-up phase was employed.
- `Learning Rate`: We set the initial learning rate to 2e-05, which determines the step size during gradient descent optimization.
- `eps`: The epsilon value was set to 1e-06, which prevents division by zero in certain calculations.
- `Thresholds P1&P2`: The thresholds values that were used to leave difficult cases unanswered as explained in section 5.2 as, they were set to 0.45 and 0.54 respectively.

5.3.3. Hyper-parameter Tuning Process

The process of hyper-parameter tuning involved a combination of manual exploration and iterative experimentation. We started with initial values based on previous studies and gradually refined them through empirical evaluation.

We performed a grid search on the validation set to explore different combinations of hyper-parameters. For each combination, we trained the model and evaluated its performance using the evaluation metrics in section 5.2, such as area under the curve (AUC), C@1 score, F1 score, Brier score, and overall performance.

Based on the evaluation results, we iteratively adjusted the hyper-parameters to improve the model's performance. This process involved experimenting with different values for specific hyper-parameters while keeping others constant. We repeated this iterative process until we achieved satisfactory performance on the validation set.

5.3.4. Performance Comparison

To assess the impact of hyper-parameter tuning, we compared the performance of the model before and after tuning. The evaluation metrics mentioned in section 5.2 were used to measure and compare the performance across different hyper-parameter configurations.

6. Experiments on Other Models

We performed experiments on two other models. The first is nli-distilroberta-base-v2 , which is a sentence-transformer model based on distilroberta. It has a max-sequence-length of 512 tokens. The second model is the Stylometric model proposed in [5]. The results of these models on PAN 23 validation set are given in section 7.

7. Results

Table 6 shows our results on the Authorship verification PAN 23 validation dataset using all-MiniLM-L12-v2, nli-distilroberta-base-v2 and the Stylometric model.

Table 6
Results of Authorship verification PAN23 Validation dataset

Model	AUC	C@1	F0.5u	F1-score	Brier	Overall
all-MiniLM-L12-v2	0.684	0.632	0.615	0.711	0.761	0.681
nli-distilroberta-base-v2	0.572	0.509	0.553	0.667	0.692	0.598
Stylometric Model	0.505	0.512	0.433	0.343	0.521	0.463

So, the best one is the **all-MiniLM-L12-v2** model, which was selected for the submissions. The performance of our three submitted runs on the PAN23 dataset is presented in Table 7, according to the TIRA [9] website.

Table 7
Results of the performance of our three submissions on the PAN23 testing dataset (from TIRA)

Submission name	Model	AUC	C@1	F0.5u	F1-score	Brier	Overall
resolving-globe	all-MiniLM-L12-v2	0.616	0.572	0.562	0.617	0.746	0.623
reduced-graph	all-MiniLM-L12-v2	0.616	0.572	0.562	0.617	0.746	0.623
golden-ottoman	all-MiniLM-L12-v2	0.598	0.546	0.55	0.622	0.744	0.612

We submitted three runs on TIRA website: 'resolving-globe','reduced-graph', and 'golden-ottoman'. All of them use same version of the trained model (**all-MiniLM-L12-v2**). The first two runs are the same but, the second run has additional option to take p1 and p2 [described in section 5.2] as input so that we can modify them. And both of them have the same default values and they use the chunking technique described in section 4.2. The third run was to see the effect of soft-chunking on the input data so that there is no sentence split between two chunks unless it has a number of tokens larger than the max sequence length of the model [256 in our case].

8. Conclusion

In this work, we presented the effectiveness of sentence transformers in the authorship verification problem. Based on the result we found that the **all-MiniLM-L12-v2** model achieved the best performance on the validation dataset compared to other models like **nli-distilroberta-base-v2** and **Stylometric** model. These results highlight the effectiveness of sentence transformers in tackling authorship verification, showcasing the potential of the **all-MiniLM-L12-v2** model for improving the accuracy of such tasks.

References

[1] E. Stamatatos, K. Kredens, P. Pezik, A. Heini, J. Bevendorff, M. Potthast, B. Stein, Overview of the Authorship Verification Task at PAN 2023, in: CLEF 2023 Labs and Workshops, Notebook Papers, CEUR-WS.org, 2023.

[2] J. Bevendorff, I. Borrego-Obrador, M. Chinea-Ríos, M. Franco-Salvador, M. Fröbe, A. Heini, K. Kredens, M. Mayerl, P. Pęzik, M. Potthast, F. Rangel, P. Rosso, E. Stamatatos, B. Stein, M. Wiegmann, M. Wolska, , E. Zangerle, Overview of PAN 2023: Authorship Verification, Multi-Author Writing Style Analysis, Profiling Cryptocurrency Influencers, and Trigger Detection, in: A. Arampatzis, E. Kanoulas, T. Tsikrika, A. G. Stefanos Vrochidis, D. Li, M. Aliannejadi, M. Vlachos, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Fourteenth International Conference of the CLEF Association (CLEF 2023), Lecture Notes in Computer Science, Springer, 2023.

[3] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, arXiv preprint arXiv:1908.10084 (2019). URL: https://arxiv.org/abs/1908.10084.

[4] Z. Peng, L. Kong, Z. Zhang, Z. Han, X. Sun, Encoding text information by pre-trained model for authorship verification., in: CLEF (Working Notes), 2021, pp. 2103–2107.

[5] J. Weerasinghe, R. Singh, R. Greenstadt, Feature vector difference based authorship verification for open-world settings., in: CLEF (Working Notes), 2021, pp. 2201–2207.

[6] A. Manolache, F. Brad, E. Burceanu, A. Barbalau, R. Ionescu, M. Popescu, Transferring bert-like transformers' knowledge for authorship verification, arXiv preprint arXiv:2112.05125 (2021).

[7] J. A. Martinez-Galicia, D. Embarcadero-Ruiz, A. Ríos-Orduña, H. Gómez-Adorno, Graph-based siamese network for authorship verification (2022).

[8] 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: Advances in Information Retrieval: 45th European Conference on Information Retrieval, ECIR 2023, Dublin, Ireland, April 2–6, 2023, Proceedings, Part III, Springer, 2023, pp. 518–526.

[9] M. Fröbe, M. Wiegmann, N. Kolyada, B. Grahm, T. Elstner, F. Loebe, M. Hagen, B. Stein, M. Potthast, Continuous Integration for Reproducible Shared Tasks with TIRA.io, in: J. Kamps, L. Goeuriot, F. Crestani, M. Maistro, H. Joho, B. Davis, C. Gurrin, U. Kruschwitz, A. Caputo (Eds.), Advances in Information Retrieval. 45th European Conference on IR Research (ECIR 2023), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2023, pp. 236–241.