Contrastive Learning for Authorship Verification using BERT and Bi-LSTM in a Siamese Architecture

Notebook for PAN at CLEF 2023

Panagiotis Petropoulos

Abstract

Authorship Verification constitutes an important and essential part in the more general area of authorship analysis. In the age of digital communication, where information traverses the globe at the speed of light, determining the true authorship of a text has become a paramount challenge. From verifying the authenticity of legal documents to investigating plagiarism cases and combating online fraud, the field of authorship verification has emerged as a crucial domain in the realm of textual forensics. The PAN 2023 Authorship Verification [1] challenge focuses on determining whether two different types of texts are written by the same author, specifically when the corresponding model is consuming unseen Authors (open set setup). This paper proposes a Contrastive Learning approach with Siamese architecture, that consists of BERT [2][1] and Bi-LSTM on top of BERT. Additionally, this work proposes the creation of 2 dataset from the original one. One for written language texts and one for oral texts. In this way it would be more helpful for a model to compare-contrasts texts to extract stylistic features.

Keywords

Authorship Verification, Contrastive Learning, BERT, Bi-LSTM, Siamese Architecture, NLP, Authorship Analysis

1. Introduction

Authorship Verification is defined as the task in which an artificial intelligence model, after being trained, is able to determine the probability of two or more texts belonging to the same author or not. This can be achieved by analyzing and retrieving useful information from texts. Modern applications are aimed at retrieving writing style to solve the specific problem. There are many approaches that use traditional techniques for feature extraction to solve the task [3], [4], [5]. They use modern approaches hand-engineered features in combination with n-grams extracted from multiple CNNs [6] or pretrained word embeddings [7]. Other approaches use pre-trained language models and contrastive learning frameworks [8], [9] and [10] or a combination of one Encoder-Decoder Transformer with traditional features such as pos-tags and n-grams [11]. Finally, there are also approaches based on Cosine Similarity or dissimilarity of feature vectors for the final decision [12] and a modified or improved Impostors method, which also a method that calculated the cosine similarity between the feature vectors [13].

In this work a pretrained BERT model with a Bi-LSTM on top of that in a Siamese architecture for a contrastive learning framework is applied. The main challenge in this task is to create a model that can be able to handle stylistic information about the authors between different type of texts (email vs essay vs interview vs speech transcription). Also, the metadata information about the types of text is used and the original train dataset is reconstructed in a way that there is one train dataset that consists of written language texts (email vs essay) and another with oral language texts (interview vs speech transcription).

CLEF 2023: Conference and Labs of the Evaluation Forum, September 18-21, 2023, Thessaloniki, Greece
EMAIL: panos.petr1@gmail.com (P. Petropoulos)

2. Model Overview

The proposed model is illustrated in **Figure 1**. The BERT model is kept frozen during training.

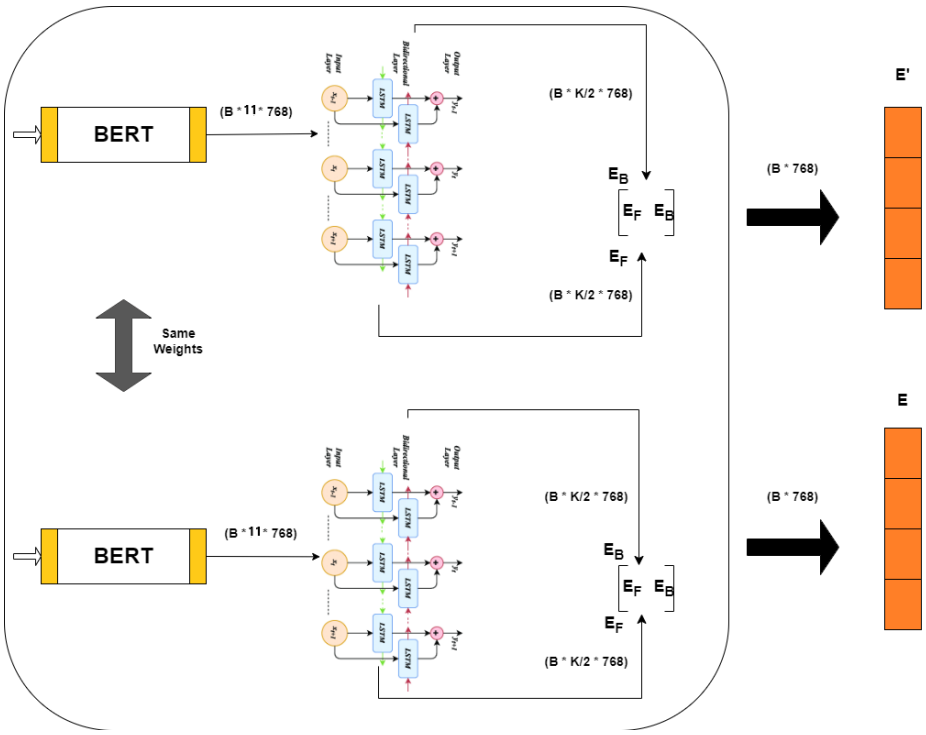

Figure 1: Siamese Architecture of the proposed Model.

2.1. Experimental setup

A Siamese architecture is used that was originally introduced by [15] and consists of 2 same instances of BERT (same weights) and on top of each instance there is a Bi-LSTM to capture the appropriate information from 2 different directions of a long sequence during Contrastive Learning procedure. Based on the experiments the best results are obtained from the concatenation of CLS token Embeddings from 2 to 12 Encode Layers. Those Embeddings are then passed through a Bi-LSTM in order to handle the stylistic information. The final embeddings (E and E') are the concatenation of the last hidden states from 2 directions of Bi-LSTM. The selected optimizer is Adam with learning rate equals to 0.002. The PAN 2023 AV dataset consists of different types of texts. Based on PAN 2022 results [5] a cross-DT task was very challenging. For that reason, a model for written texts (email vs essays) was created, another one for texts that came from speech (speech transcription vs interviews) and a general model that was trained on all types. The final predictions for those pairs that do not have a specific model for them (i.e. essay vs interview) were produced from an Ensemble procedure **Figure 2** between the 3 trained models. All models were trained for 5 epochs and the batch size was 32.

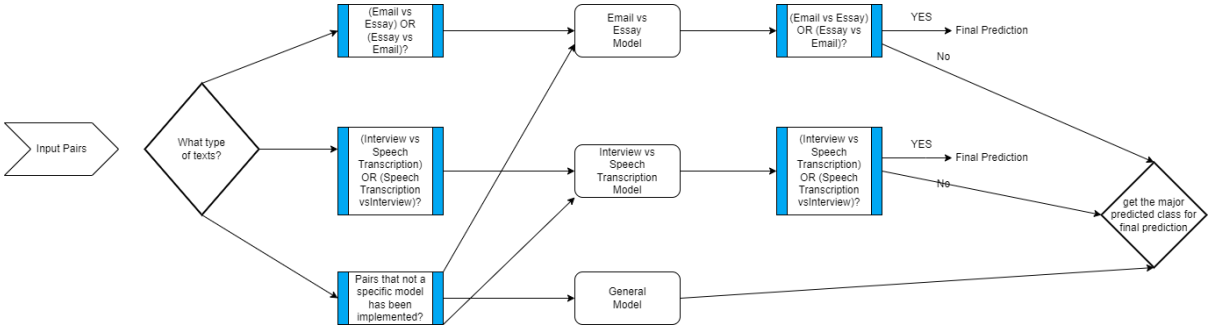

Figure 2: Final System overview - Ensemble Procedure for Final prediction.

The Final System consists of 3 sub models. The flow diagram of the end-to-end procedure for the final predictions is illustrated in **Figure 2**.

2.1.1. Data preparation and processing

The PAN 2023 Authorship Verification challenge provided a dataset that consists of 8837 problems located in FoLD [2] [14]. Each problem contains a pair of text and information about the type of each of them. This specific dataset consists of 4 main categories of text types:

1. Email
2. Essay
3. Interview
4. Speech transcription

Exactly 4418 of these pairs were written from same author and 4418 from different author.
Instead of using the predefined pairs from original dataset, All texts for each Author were collected and were appended to a pool in a way that each Author will have *N* texts. After the collection, a chunking procedure for each text was applied. The final illustration of the collection pool is shown in **Figure 3**. Based on this figure, for each author there are *N* lists of *M* chunks. Not all chunks were collected in one list because during retrieval in training/validation procedure there was no desire to select 2 chunks from the same text, to avoid handling information about the topic.

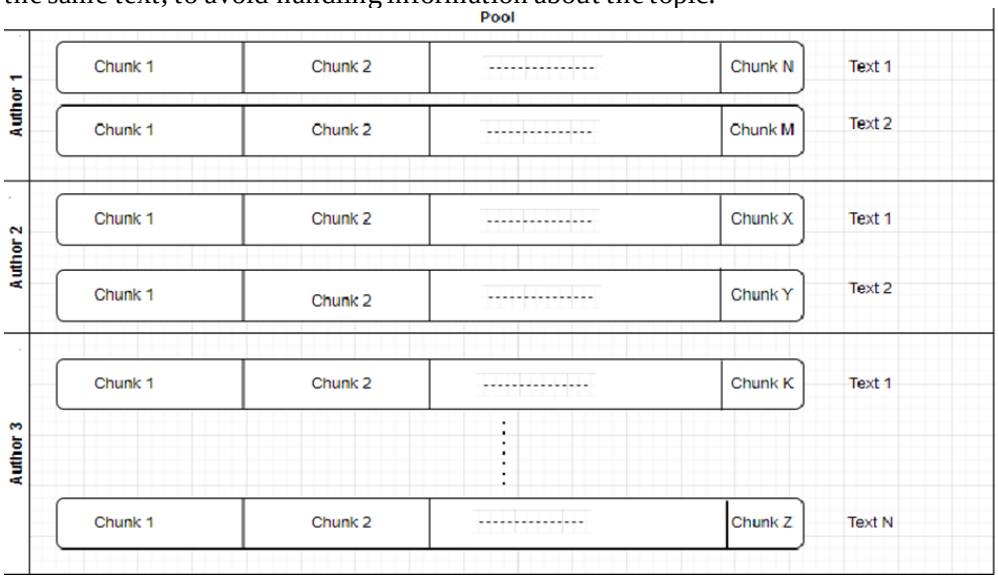

Figure 3: Collection of Chunks w.r.t. Authors and texts.

Due the fact that BERT has limitation on the length of input tokens, a chunking procedure on text is applied with max sequence length to 128. This size was selected, because in the dataset there were small

[2] https://fold.aston.ac.uk/handle/123456789/17

and bigger texts. BERT Tokenizer decided where to chunk a text. Between chunks of the same text, an overlapping approach was applied in order to avoid too long padding sequences.
All numbers in texts masked with a specific digit '1' keep the original format and length of text. Additionally, all special words such as <nl>, <new> etc were removed from the text. For the email cases the html tags were removed as well.
For a Contrastive Learning framework, we need to consider the false negatives. So, in order to create a batch, the below steps were followed:

1. Random Select Author.
2. Random Select 2 chunks from different texts, to construct a Same Author pair.

In this way a batch with different authors and same author pairs is created. The loss function is calculated for all combination in the batch. As input to the model, same Author pairs are passed (main diagonal of **Figure 5** in blue) and the negative pairs are calculated inside the loss function. This can be seen as an alternative way to increase the batch size during calculation of loss function. Due to this approach, caution is required on how to create the batch to avoid the false negatives. An illustration of how the cosine similarity is calculated is shown in **Figure 5**. Based on this figure the different author pairs are located above and below the main diagonal of dot product matrix. For example, above the main diagonal there is P = Ei x Ej and below the main diagonal there is the transpose of this product P^T (P = Ej x Ei).
The following example describes a false negative scenario.
Imagine that batch size is 64. So, 2 chunks from 32 Authors should be chosen randomly (32*2=64).
For each Author within the batch there will be 2 chunks from different texts. The implemented loss will get those chunks as input and will apply cosine similarity as metric that contributes to loss. The main diagonal of **Figure 5** will have the cosine similarity values of the same author pairs (2 chunks of Same Author). The results of cosine similarity between different author pairs will be located above and below the main diagonal of this matrix. Each chunk of the same Author will be compared with each chunk by another Author. In this way if a data loader selects twice the same author within the same batch, 4 chunks (2 + 2) by the same Author will be selected, resulting to a false negative scenario. In other words, some values above and below the main diagonal will be treated (wrongly) as different Author pairs by the loss function. The reason for the strictness of data selection is the non-existence of False Negative within Batch. If all the Authors are selected in a batch, then based on the image and the calculations of all combinations within batch, for the same Authors there will also be a pair that will be treated as Negative. In this way, within the batch there will be some False Negatives. Below is the illustration (**Figure 4**) of an incorrect Batch for better understanding of the above description.

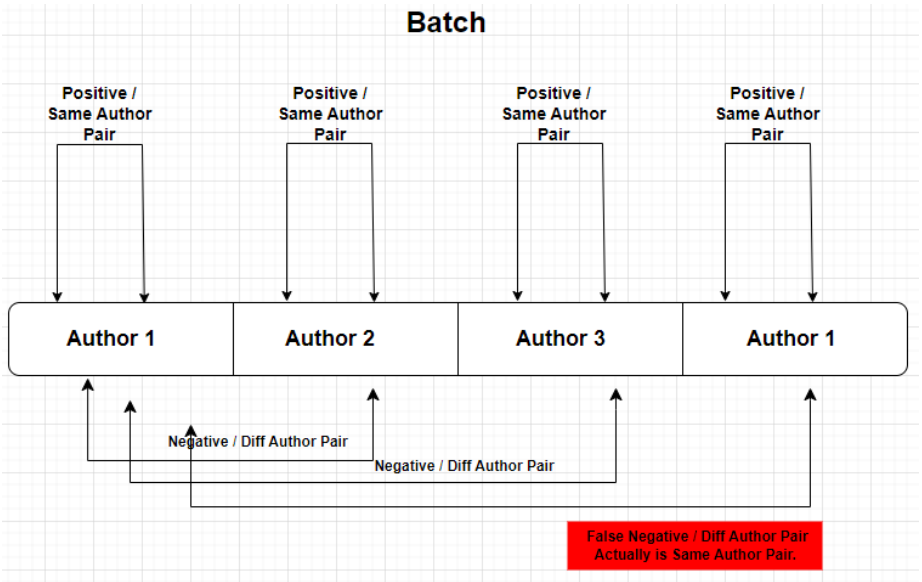

Figure 4: Example of a Batch with False Negative.

2.1.2. Loss Function

In order to train and evaluate the model a contrastive learning procedure with modified contrastive loss [16] based on Cosine similarity is followed.

$$L(E,E',Y) = \frac{1}{2}\left(Ymax\left\{\left(D(E,E') - m_p\right) \ ,0\right\}^2 + (1-Y)max\left\{\left(m_n - D(E,E')\right) \ ,0\right\}^2\right) \quad (1)$$

Where m_p is the positive margin, m_n is the negative margin. Value above positive margin and under negative margin do not contribute to loss. For training and validation those margins were kept at 1 and 0, respectively. Y is the target label as ground truth and E & E' are the Embeddings Vectors of a pair in a batch. D(E,E') are the Cosine Similarity values between the pairs of Vectors. D can also be Euclidean or Manhattan distances. In **Figure 5** the calculation of similarities between all pairs and combinations in a batch is illustrated. For the AV task the cosine similarities between same authors pairs are expected to be positive real numbers and equal to 1 (ideal scenario) and lower or negative values with minimum value equal to -1 for the different authors pairs. Cosine similarity have values between [-1,1].

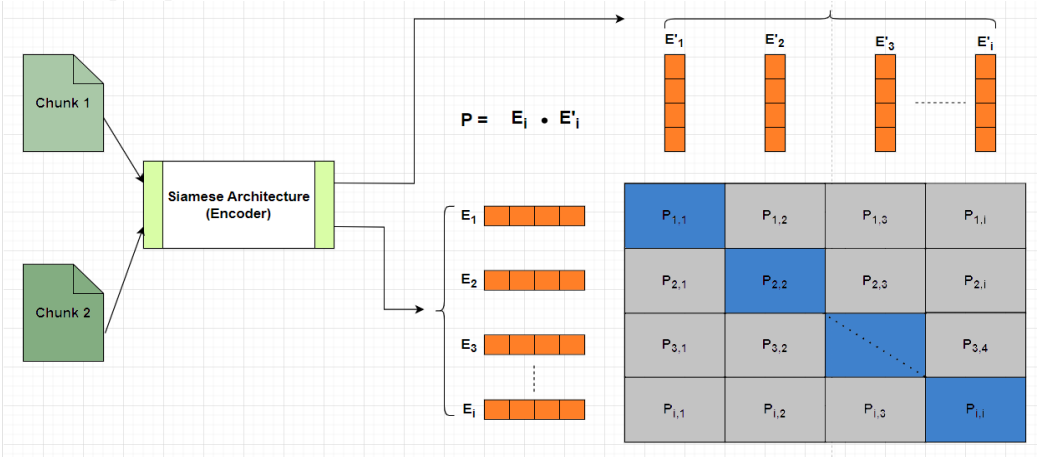

Figure 5: Dot Product on normalized Embeddings calculation for all combinations in a batch.

2.1.3. Evaluation

After splitting original data in open set setup w.r.t to authors, 70% for training 15% for validation and 15% for a holdout test set were kept. For the test set the chunking procedure for each text in a pair was performed and during inference and prediction the average predicted Embeddings from all chunks for each text in a pair was taken. After that, a cosine similarity calculation is performed. The final prediction is calculated via a threshold. Another approach would be to calculate the cosine similarities for all combinations of chunks (Embeddings) from one text with all chunks from the other text (and vice versa) and as final score calculate the average similarity. The final decision is calculated for all combination of chunks from 2 texts in a pair and not from specific chunk pairs, because the model trained with random combinations. To find an optimal threshold a grid search approach is performed, in order to calibrate the models and the threshold with the best overall score is kept. The values of those threshold per model are 0.697 for Essay vs Email model, 0.67 for Interview vs speech transcription and for the general model is 0.56. As evaluation metrics I use the metrics provided in [4].

- AUC: the conventional area-under-the-curve of the precision-recall curve
- F1-score: the harmonic mean of the precision and recall [17]
- c@1: a variant of the conventional F1-score, which rewards systems that leave difficult problems unanswered (i.e. scores of exactly 0.5) [18]
- overall: the simple average of all previous metrics

3. Results

Table 2 below shows Accuracy, F1-score AUC and C@1 scores with their corresponding std between 3 runs on my custom unseen test set. The test set was created based on open-set setup. In Tira system[3] [20] 2 approaches were submitted with different thresholds for the final decision as shown in below Table (**Table 1**). The values for the threshold are not selected randomly but based on the best Overall score in my test set. For The first run the threshold values are selected from the best (top-1) Overall score and for the second run the values are selected from the second-best score. In **Table 2** the first column of **Table 1** 0.697, 0.67, 0.56 respectively were used as thresholds.

Table 1: Threshold values for 2 different runs on test set in Tira.

Method	Threshold 1st run	Threshold 2nd run
Model Email vs Essay	0.697	0.71
Model Interview vs Speech transcription	0.67	0.65
General Model	0.56	0.54

The difference between the 2 runs (approaches) in Tira system is the values of thresholds. All the architectures, hyper-parameters and training procedure are the same.

Table 2

Metrics and std on my custom unseen test set for three proposed models (written language model, speech language model and general model).

Method	Accuracy	F1-score	AUC	C@1
Model Email vs Essay (sub-model)	0.82(+/- 0.03)	0.72(+/- 0.05)	0.80	0.58
Model Interview vs Speech transcription (sub-model)	0.87(+/- 0.05)	0.72(+/- 0.04)	0.81	0.58
General Model (sub-model)	0.55 (+/- 0.03)	0.51(+/-0.02)	0.48	0.49
Final Model (Ensemble **Figure 2**)	0.70(+/- 0.04)	0.68 (+/- 0.04)	0.62	0.51

From the results it can be seen that using the information about the type of text and creating separate models to solve the task can be very helpful. The General model in **Table 2** was not performed very well like the other models that were created for specific types of texts. The Final model is the model

[3] https://www.tira.io/

that consumes the whole custom test dataset and performs prediction including the Ensemble procedure with the sub-models. The results are lower than the other models in table, due to the fact that the test dataset contains more pairs that there is not a specific model for them such as interview vs email. For this challenging dataset the final model performs very well in this test set.

Table 3
Metrics of 2 runs on Tira system with unseen test set.

Method	AUC	C@1	F_05_u [19]	F1-Score	brier	Overall
clever-daemon (1st run)	0.525	0.516	0.55	0.624	0.743	0.591
graceful-chianti (2nd run)	0.526	0.514	0.549	0.622	0.743	0.591

It can be observed from **Table 3** that the 2 runs in Tira system returned roughly the same results. The proposed methodology did not performed very well, but makes the AV task feasible with a contrastive learning framework.

4. Conclusion

Based on results, it is observed that, handling stylistic information between different types of texts is a very challenging task. That can be also confirmed from the optimal thresholds. They are too high and that means that trained models cannot create a clean Embeddings space where negative pairs stand apart from each other and positive pairs the opposite. It can also be observed that the proposed methodology can make the cross-DT AV task feasible, with some modifications, especially when the metadata information about the type of texts is used. As future work it would be great to train a model with a triplet loss with online and fast hard negatives mining [21]. Also, additional features, apart from POS-tags, such as n-grams and complexity of texts features can be combined with contextualized word Embeddings, to create a joint embeddings space for a contrastive learning framework. Due to lack of time and resources there was no application of the same procedure with RoBERTa [22] model or another pre-trained language model like GPT [23] and it would be a good choice as feature work to perform benchmarks with other powerful pre-trained models. Finally, a more interesting and challenging approach is to re-train the BERT model with additional initial Embeddings Layer that produce embeddings from pos-tags of the original input sequence. This approach needs more data and special care on the other pre-trained layers because the new initial Layer will not be pre-trained.

5. References

[1] Efstathios Stamatatos, Krzysztof Kredens, Piotr Pezik, Annina Heini, Janek Bevendorff, Martin Potthast, and Benno Stein. Overview of the Authorship Verification Task at PAN 2023. CLEF 2023 Labs and Workshops, Notebook Papers, September 2023.

[2] Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

[3] Stamatatos, Efstathios. "Authorship verification: a review of recent advances." Research in Computing Science 123 (2016): 9-25.
[4] Kestemont, Mike, et al. "Overview of the Cross-Domain Authorship Verification Task at PAN 2020." CLEF (Working Notes). 2020.
[5] Efstathios Stamatatos, Mike Kestemont, Krzysztof Kredens, Piotr Pezik, Annina Heini, Janek Bevendorff, Benno Stein, and Martin Potthast. Overview of the Authorship Verification Task at PAN 2022. In Guglielmo Faggioli, Nicola Ferro, Allan Hanbury, and Martin Potthast, editors, CLEF 2023 Labs and Workshops, Notebook Papers, September 2023.
[6] S. Ruder, P. Ghaffari, J. G. Breslin, Character-level and multi-channel convolutional neural networks for large-scale authorship attribution, arXiv preprint arXiv:1609.06686 (2016).
[7] B. Boenninghoff, J. Rupp, R. M. Nickel, D. Kolossa, Deep bayes factor scoring for authorship verification, arXiv preprint arXiv:2008.10105 (2020).
[8] Tyo, Jacob, Bhuwan Dhingra, and Zachary C. Lipton. "Siamese Bert for Authorship Verification." CLEF (Working Notes). 2021.
[9] Boenninghoff, Benedikt, et al. "Similarity learning for authorship verification in social media." ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019.
[10] F. Jafariakinabad, K. A. Hua, A self-supervised representation learning of sentence structure for authorship attribution, arXiv preprint arXiv:2010.06786 (2020).
[11] Najafi, Maryam, and Ehsan Tavan. "Text-to-Text Transformer in Authorship Verification Via Stylistic and Semantical Analysis." Proceedings of the CLEF. 2022.
[12] Potha, Nektaria, and Efstathios Stamatatos. "A profile-based method for authorship verification." Artificial Intelligence: Methods and Applications: 8th Hellenic Conference on AI, SETN 2014, Ioannina, Greece, May 15-17, 2014. Proceedings 8. Springer International Publishing, 2014
[13] Potha, Nektaria, and Efstathios Stamatatos. "An improved impostors method for authorship verification." Experimental IR Meets Multilinguality, Multimodality, and Interaction: 8th International Conference of the CLEF Association, CLEF 2017, Dublin, Ireland, September 11–14, 2017, Proceedings 8. Springer International Publishing, 2017.
[14] Kredens, K., Heini, A. and Pezik, P. 100 Idiolects - a corpus for research on individual variation across discourse types. Aston University: FoLD Repository, 2021.
[15] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2019.
[16] B. Boenninghoff, R. M. Nickel, S. Zeiler, D. Kolossa, Similarity learning for authorship verification in social media, in: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2019, pp. 2457–2461.
[17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., Scikit-learn: Machine learning in python, the Journal of machine Learning research 12 (2011) 2825–2830.
[18] A. Peñas, A. Rodrigo, A simple measure to assess non-response (2011).
[19] J. Bevendorff, B. Stein, M. Hagen, M. Potthast, Generalizing unmasking for short texts, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 2019, pp. 654–659.
[20] Fröbe, Maik, et al. "Continuous integration for reproducible shared tasks with TIRA. io." European Conference on Information Retrieval. Cham: Springer Nature Switzerland, 2023.
[21] Gajić, Bojana, Ariel Amato, and Carlo Gatta. "Fast hard negative mining for deep metric learning." Pattern Recognition 112 (2021): 107795.
[22] Liu, Yinhan, et al. "Roberta: A robustly optimized bert pretraining approach." arXiv preprint arXiv:1907.11692 (2019).
[23] Radford, Alec, et al. "Improving language understanding by generative pre-training." (2018).