# Team HHU - An Ensemble-Based Approach to Multi-Author Writing Style Analysis Combining Experts for Different Difficulty Levels

Notebook for PAN at CLEF 2025

Philipp Meier[1,*], Katarina Boland[1], Laura Kallmeyer[1] and Stefan Dietze[1]

[1]*Heinrich-Heine-University, Universitätsstr. 1, Düsseldorf, Germany*

## Abstract

In the task of Multi-Author Writing Style Analysis models must identify author changes between two subsequent sentences in a multi-author document. To solve the task, we deploy an ensemble method with three transformer-based language models acting as experts for different difficulty levels and a weighting mechanism to weigh the model predictions. Furthermore, we use data augmentation to provide a more balanced dataset and further enhance the performance of our approach. Our ensemble method outperforms a model trained on the complete dataset containing data of all difficulty levels. This shows that our model succeeds in training and selecting more useful features for the different datasets. However, when examining the datasets individually, the best performance is achieved by the respective expert which shows that there is room for improvement regarding the weighting mechanism. This work demonstrates how language models can be combined to tackle the Multi-Author Writing Style Analysis task for data that is heterogeneous in terms of difficulty and domain.

## Keywords

PAN 2025, <Multi-Author Writing Style>, <Ensemble methods, Style Change Detection, Writing Style Analysis>

## 1. Introduction

The goal of the PAN Multi-Author Writing Style Analysis [1] [2] shared task at CLEF is to identify text positions in a multi-author document at which the author changes. A model is given a sentence pair and must classify whether an author change occurs in between or not. This task is important for applications like plagiarism detection [3] [4] or machine-generated text detection [5] [6] [7].

While previous editions of this task focused on author changes at the paragraph-level within Reddit comments [8] [9], this year's challenge targets authorship changes between sentences from Reddit comments, providing substantially less context for distinguishing authorship. Data is provided across three different difficulty levels: easy, medium and hard. These difficulty levels vary with respect to the extent to which topics or syntax differ (or do not differ) between texts from different authors. Using topic information is a promising feature to detect author changes in the easy split. Since topic variation decreases with increasing difficulty, models have to rely more on stylistic cues to detect an author change for medium and hard splits.

In a realistic scenario, a differentiation in levels of difficulties for the input data is not given. Therefore, we propose a model that assumes no prior knowledge concerning the difficulty of an input sentence pair during inference but still achieves a high performance on every difficulty level by incorporating and selecting appropriate experts. The model consists of three language models, each trained on one specific difficulty (easy, medium, hard). Their predictions (logits) are combined and weighted through a multilayer perceptron which weights the predictions according to the predicted difficulty level. During training, the multilayer perceptron receives the difficulty level taken from the input data as one hot encoding for training feedback. Thus, the multilayer perceptron should learn which experts prediction

*CLEF 2025 Working Notes, 9 – 12 September 2025, Madrid, Spain*

[*] Corresponding author.

philipp.meier@hhu.de (P. Meier); katarina.boland@hhu.de (K. Boland); laura.kallmeyer@hhu.de (L. Kallmeyer); stefan.dietze@hhu.de (S. Dietze)

to prioritize when no information about the difficulty level is available during inference. These weights prioritize the prediction of the language model that was fine-tuned on the difficulty level matching the difficulty level of the input pair. Through this generalization capability, our proposed model should be able to handle real world scenarios where the difficulty level of input pairs is unknown.

## 2. Related Work

### 2.1. Writing Style

Besides of this shared task, writing style analysis has been approached from various angles. [10] and [11] incorporate lexical and syntactic features to analyze writing style. It is often the case that such features are tailored towards the dataset, lacking generalizability to other datasets [12, 13]. More general approaches cover language models like BERT [14] which are pre-trained on large text corpora and fine-tuned on writing style tasks [15]. Another approach is contrastive learning, which aims to pull instances of the same class close to each other while pushing instances from different classes away from each other in the embedding space like in [13] and [16].

#### 2.1.1. Multi-Author Writing Style Analysis

In previous iterations of the shared task, paragraphs from Reddit were used as training data. [17] deployed an ensemble of language models, combining their predictions through majority voting. For easy and medium cases, LaBSe embeddings [18] were used to measure the similarity between the input sentences. The language models were fine-tuned on all difficulty partitions. Their model yields an F1-Macro score of 0.96 for easy, 0.85 for medium and 0.86 for hard instances. However, this approach requires knowledge about the difficulty of input pairs during inference. [16] used contrastive learning and data augmentation in 2023, yielding and F1 score of 0.91 for easy, 0.82 for medium and 0.68 for hard.

### 2.2. Ensemble Methods

Ensemble mechanisms have multiple facets in related work: [19] used an ensemble mechanism for the learning with disagreement task. The authors trained a supervised classifier on the hidden state [CLS] representation of three fine-tuned language models. Mnassri et al. [20] tried different ensemble mechanisms like soft voting, maximum value and stacking for hate speech detection. [21] combined BERT logits with linguistic features through stacking for identifying propaganda.

## 3. Approach

To address the varying difficulty of author change prediction, we propose an ensemble model that combines the strengths of specialized language models through a weighting mechanism which prioritizes the prediction of the expert model according to the difficulty. Using an ensemble model was inspired from [17]. The architecture is shown in Figure 2. To increase robustness, the ensemble model includes learned weights, which capture cues in the input pair that signal which each experts prediction to prioritize. Through this, predictions from the expert language models are prioritized. Unlike [17], our model does not rely on a majority vote, since this could possibly outvote the experts opinion. Since our model relies on fine-tuned language models, extracting stylistic features is not necessary. Additionally, we also used data augmentation similar to [16] in order to mitigate the unbalanced nature of the data. (there are many more instances of no author change than of author changes among the sentence pairs).

Initial experiments indicated that language models like ERNIE [22], RoBERTa [23], BERT or DeBERTa [24] are superior to Logistic Regression or Support Vector machine using linguistic features. Linguistic

features covered reading easiness scores, capitalization ratio and spelling errors. Conducted experiments with ERNIE, RoBERTa, DeBERTa, Electra and BERT for difficulty-level-specific predictions resulted in the choice of using ERNIE as expert for easy and hard instances and RoBERTa for medium instances. ERNIE is based on the BERT-architecture but is more aware of named entities. ERNIE trained on the easy split yields a F1-Macro of 0.96, RoBERTa trained on the medium split a F1-Macro of 0.82 and ERNIE trained on the hard split a F1-Macro of 0.82. Further results on the validation set are shown in Table 2 in Section 4.
When combining experts, a majority vote would possibly outvote correct predictions. Therefore, we design the assembly mechanism as weighting mechanism. This allows the model to not require any knowledge about the difficulty level during inference, which is similar to real-world applications. Through the weighting mechanism, the ensemble model is expected to learn to prioritize the most relevant expert prediction based on the input itself.

### 3.1. Data augmentation

**Table 1**
Counts of authorship change (0) and non-authorship change in the original and augmented dataset.

| Dataset | Easy | | Medium | | Hard | |
|---|---|---|---|---|---|---|
| | 0 | 1 | 0 | 1 | 0 | 1 |
| Standard | 38,172 | 10,224 | 46,294 | 12,503 | 42,232 | 8,815 |
| Augmented | 35,435 | 29,133 | 37,027 | 34,240 | 42,012 | 27,799 |

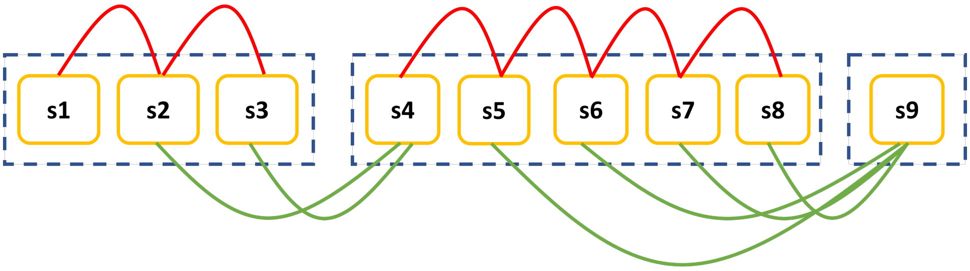

**Figure 1:** Implemented data augmentation similar to [16]. An input document is grouped according to the provided label, in this case [0,0,1,0,0,0,0,1]. This produces three groups, since two author switches are present. Green represents author changes, Red represents non-author change.

Due to the imbalance between negative (no author change) and positive (author change) pairs, the training data set was augmented similar to [16]. As illustrated in Figure 1 the document was grouped based on the labels indicating an author change for each sentence pair. As augmentation data, the whole provided training dataset of PAN 2025 [25] was used. Augmentation was performed under specific constraints: We add author change instances combining sentences belonging to two subsequent groups. The first sentence of such a new pair must not be the first of a group while the second sentence of each new pair must be the first of its group. This ensures that augmented pairs do not span authorship boundaries in a way that might introduce artificial topic shifts, which could mislead the model. Newly added author change pairs on Figure 1 are s2 to s4 and s5, s6, s7 to s9, respectively. Table 1 gives the size of the original and the augmented data. The presented augmentation method was not able to create an exactly balanced dataset. As difficulty level for the augmented data, the original label of the partition was used. For example, if the augmentation is done for the easy partition, the augmented data is also labeled as 'easy'. However, in a post analysis, we found out that this assumption does not always hold: Augmented instances for the hard partition can resemble to 'easy' or 'medium' instances. Such instances demonstrate a clear topic shift between the sentence pair and not necessarily a difference in writing style. For example, consider the following labels: [0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0]. In total, these groups contain 13 sentences. Through the augmentation method, one receives 9 non-author change instances and 5 author-change instances. This is caused by the last group which causes more no-author changes than contributing author changes by pairing the its first sentence with group 3. During data augmentation, we also filtered out duplicates resulting in a smaller number of no-author change data compared to the original dataset. This affected moderator messages, for example, which were often automated comments with the same content and mainly contained in the easy split.

### 3.2. Architecture

First, the experts were fine-tuned on the specific difficulty partition of the original dataset using the provided data-splits of PAN 2025, which contains 4200 documents per difficulty. The validation set consists of 900 documents per difficulty. Data splits are 70% for training of the whole dataset, 15% for validation and 15% for testing. The test dataset is only available during inference on TIRA [26]. Fine-tuning on the original dataset provided better results than using the augmented data. However, the ensemble model yields better F1-Macro scores using the augmented data. Learning objective of the difficulty weighting mechanism learning was to prioritize the prediction of the expert suited to the difficulty. Ernie and RoBERTa were trained using Adam Optimizer with a learning rate of 0.0005 and a weight decay of 0.05 using a batch size of 4 and trained for 10 epochs. Validation metric was F1-Macro. The best checkpoint was used for the final model.

The ensemble model is a feed-forward model, which wrapped the three fine-tuned experts and learned the difficulty weights during ensemble training. Output is the prediction of the expert matching to the difficulty. In the training procedure, the three experts in the ensemble received the input pair and outputted their logits. These logits are weighted by a weighting mechanism, which is a feed-forward network taking the last hidden state of the [CLS] token of each model as input. The input was stacked resulting in an input shape of dimension 768 $\times$ 3.

Experiments covered using a feed-forward network as classifier, which either used the concatenated [CLS] representation or the predicted logits as input. Furthermore, we experimented with using logits or using [CLS] representation as input to a feed forward network (FFN), which classified whether an author change happens or not. The latter architecture seemed too complex, yielding results that suggested overfitting. Additionally, we tried different aggregation methods for [CLS] representations like stacking, concatenating, mean pooling and weighting. Best result was achieved by using difficulty weighting and stacking the tensors of the hidden representations of the [CLS] token of each of the experts as input of the difficulty weighting mechanism.

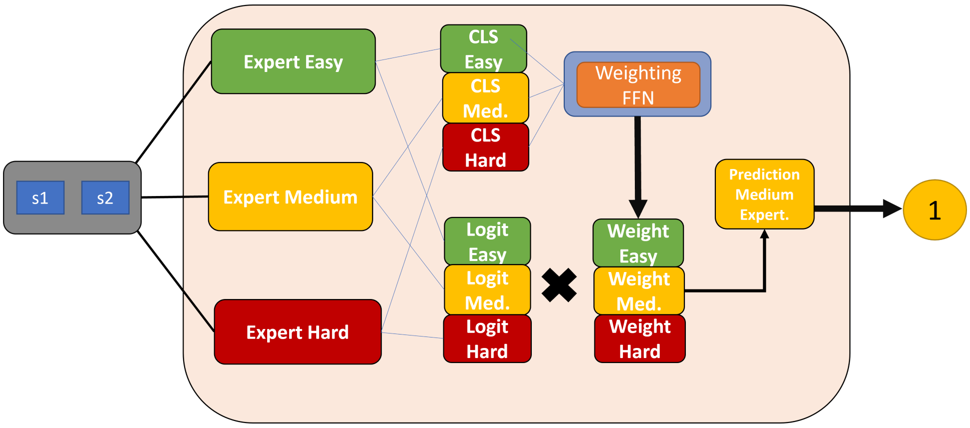

**Figure 2:** Architecture of the proposed model. The model receives an input pair, which is fed to each expert. CLS representations of each expert are stacked and fed to the difficulty weighting mechanism, which weights the stacked computed logits according to the predicted difficulty level, prioritising the prediction of the most competent expert (Expert Medium in the case shown in the figure). Colors represent the difficulty level: Green for easy, yellow for medium, red for hard.

For the weighting mechanism, a two layer neural network with ReLu activation and a hidden size of 128 was used. The FNN takes a matrix of [batch_size, 3×768] as input. The first layer $z_1$ as well as the second layer $z_2$ has a hidden size of 128. As output, a vector of three dimensions is computed, representing scores for each difficulty level: $w = [w_{easy}, w_{medium}, w_{hard}]$. Computation is shown in equation 4. As training labels for the FFN, the difficulty level of an input pair was one-hot encoded (e.g [0,0,1] for hard instances). Finally, the weights were multiplied with the combined logits $l = [l_{easy}, l_{medium}, l_{hard}]$ of the experts as shown in equation 5. Through this, the prediction (logits) of the expert suiting to the difficulty is reinforced by receiving the most weight. Afterwards, the log-softmax of the weighted prediction is calculated for the negative log-likelihood loss as shown in equation 5. Experiments have shown that the highest F1 score was reached by training on the augmented training set.

$$z_1 = W_1 x_y + b_1, \text{where } W_1 \in \mathbf{R}^{2304*128} \tag{1}$$

$$a_1 = ReLU(z_1) \tag{2}$$

$$z_2 = W_1 a_1 + b_2, , \text{where } W_1 \in \mathbf{R}^{3*128} \tag{3}$$

$$w = softmax(FNN(h)) \tag{4}$$

$$y = \text{log softmax}(\sum_{i}^{3} w_i * l_i) \tag{5}$$

The overall loss is a weighted combination of the classification loss and the difficulty weighting loss. Negative Log Likelihood (NLL) loss is used as classification loss and cross entropy (CE) loss for the difficulty weighting loss. To calculate the final loss, a weighting factor $\alpha$ was used to weigh the difficulty weighting loss. Experiments with different values of $\alpha$ ranging from 10, 5, 0.1, 0.01 and 0.001 demonstrated that a value of 0.001 yields the best overall validation performance. Formula 6 shows the loss computation, where y stands for predicted label and $\hat{y}$ for true label, $a$ for weight values and $d$ for the one-hot encoded difficulty labels.

$$\mathcal{L} = \text{NLL}(y, \hat{y}) + \alpha * \text{CE}(a, d) \tag{6}$$

## 4. Results

Table 2 shows the results of the included experts and the ensemble model on the validation set. The ensemble model yields an average of 0.76 F1-Macro over all three difficulties. Additionally, we also provide results using the sklearn [27] implementation of AdaBoost as well as a limited human evaluation carried out by one person annotating ten documents per difficulty from the validation set. This helps to contextualize the models performance by assessing how well humans can distinguish between author changes across different difficulty levels. Features for AdaBoost were lexical features like Reading Easiness Metrics, Jaccard score, capitalization rates, as well as pronoun, noun and proper noun ratio. We also provide the results of a RoBERTa model, which was trained on every difficulty partition provided by PAN [25]. During inference, this model also received no information about the difficulty level of input pairs. RoBERTa was compared to ERNIE, BERT, Electra and DeBERTa and yielded best results.

Comparing the ensemble model to the RoBERTa model, which was trained on all three difficulty levels at once, one can see that RoBERTa yields a better F1 score for the easy and hard partition. However, the ensemble model achieves a similar score for the medium split but it is not able to outperform RoBERTa. Regarding the random and majority baselines, every provided model is able to outperform the baseline.

**Table 2**
Macro F1 scores for each difficulty on the validation set.

| Model | F1-Macro Easy | F1-Macro Medium | F1-Macro Hard | Avg. across 3 difficulties |
|---|---|---|---|---|
| Ensemble Model | 0.83 | 0.80 | 0.64 | 0.76 |
| Easy Expert (ERNIE) | **0.96** | 0.56 | 0.48 | 0.67 |
| Medium Expert (RoBERTa) | 0.88 | **0.82** | 0.61 | 0.77 |
| Hard-Expert (ERNIE) | 0.44 | 0.12 | **0.82** | 0.46 |
| RoBERTa | 0.91 | 0.80 | 0.69 | **0.80** |
| Ada Boost | 0.61 | 0.5 | 0.55 | 0.55 |
| Random Baseline | 0.45 | 0.46 | 0.44 | 0.45 |
| Majority Baseline | 0.44 | 0.44 | 0.45 | 0.44 |
| Human | 0.88 | 0.64 | 0.51 | 0.68 |

The random baseline predicted a label randomly while the majority baseline predicts the most common label. The ensemble model was not able to reach the same performance as the single expert models, which indicates that the training signal of the difficulty mechanism needs improvement. However, clear benefits can be observed using a robust ensemble approach: Regarding real world scenarios, where the difficulty is unknown, the ensemble model is able to outperform almost every expert on data which is out of its domain (e.g. easy expert on hard split). Comparing the performance particularly for the expert model specialized on hard data on the easy vs. the hard data split demonstrates that different features are learned for classification. The ensemble model also yields a higher average over the 3 difficulties than the single experts for easy and hard.

The final results on the test set are shown in Table 3 below.

**Table 3**
F1 scores for the test set provided by the TIRA platform [26]

| Model | Easy | Medium | Hard |
|---|---|---|---|
| Ensemble Model (tractable-market) | 0.76 | 0.67 | 0.64 |

## 5. Conclusion

We described the motivation, architecture and results of our agnostic approach for the Multi-Author Writing Style Analysis task at PAN 2025. Our architecture combines three fine-tuned transformer models specialized on different difficulty levels. By learning difficulty weights that dynamically prioritize the most relevant expert based on inferred difficulty, our model is well-suited for real-world applications where data varies in complexity. Additionally, we also implemented data augmentation and performed a comparison to AdaBoost using linguistic features and a short human evaluation. Further work could include a refinement of the weighting mechanism and further analyses of the ensemble model under different types of writing styles and domains. The code will be available here: https://github.com/PhMeier/author_writing_25_submission.

## Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

## Acknowledgments

We would like to thank the anonymous reviewers for their useful feedback. This work was carried out in the project *NewOrder – Understanding the erosion of the traditional knowledge order in scientific online discourse and its impact in times of crisis* (project number: K490/2022) funded by the Leibniz Association.

## References

[1] E. Zangerle, M. Mayerl, M. Potthast, B. Stein, Overview of the Multi-Author Writing Style Analysis Task at PAN 2025, in: G. Faggioli, N. Ferro, P. Rosso, D. Spina (Eds.), Working Notes of CLEF 2025 – Conference and Labs of the Evaluation Forum, CEUR Workshop Proceedings, CEUR-WS.org, 2025.

[2] J. Bevendorff, D. Dementieva, M. Fröbe, B. Gipp, A. Greiner-Petter, J. Karlgren, M. Mayerl, P. Nakov, A. Panchenko, M. Potthast, A. Shelmanov, E. Stamatatos, B. Stein, Y. Wang, M. Wiegmann, E. Zangerle, Overview of PAN 2025: Voight-Kampff Generative AI Detection, Multilingual Text Detoxification, Multi-Author Writing Style Analysis, and Generative Plagiarism Detection, in: J. C. de Albornoz, J. Gonzalo, L. Plaza, A. G. S. de Herrera, J. Mothe, F. Piroi, P. Rosso, D. Spina, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Sixteenth International Conference of the CLEF Association (CLEF 2025), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2025.

[3] A. Saini, M. R. Sri, M. Thakur, Intrinsic Plagiarism Detection System Using Stylometric Features and DBSCAN, in: 2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), IEEE, 2021, pp. 13–18.

[4] V. Vysotska, Y. Burov, V. Lytvyn, A. Demchuk, Defining Author's Style for Plagiarism Detection in Academic Environment, in: 2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP), IEEE, 2018, pp. 128–133.

[5] A. Shah, P. Ranka, U. Dedhia, S. Prasad, S. Muni, K. Bhowmick, Detecting and Unmasking AI-generated Texts through Explainable Artificial Intelligence Using Stylistic Features, International Journal of Advanced Computer Science and Applications 14 (2023).

[6] R. Corizzo, S. Leal-Arenas, A Deep Fusion Model for Human $vs$. Machine-generated Essay Classification, in: 2023 International Joint Conference on Neural Networks (IJCNN), IEEE, 2023, pp. 1–10.

[7] L. Mindner, T. Schlippe, K. Schaaff, Classification of Human- and AI-Generated Texts: Investigating Features for ChatGPT, Springer Nature Singapore, 2023, p. 152–170. URL: http://dx.doi.org/10.1007/978-981-99-7947-9_12. doi:10.1007/978-981-99-7947-9_12.

[8] E. Zangerle, M. Mayerl, M. Potthast, B. Stein, Overview of the Multi-Author Writing Style Analysis Task at PAN 2023., in: CLEF (Working Notes), 2023, pp. 2513–2522.

[9] A. A. Ayele, N. Babakov, J. Bevendorff, X. B. Casals, B. Chulvi, D. Dementieva, A. Elnagar, D. Freitag, M. Fröbe, D. Korencic, M. Mayerl, D. Moskovskiy, A. Mukherjee, A. Panchenko, M. Potthast, F. Rangel, N. Rizwan, P. Rosso, F. Schneider, A. Smirnova, E. Stamatatos, E. Stakovskii, B. Stein, M. Taulé, D. Ustalov, X. Wang, M. Wiegmann, S. M. Yimam, E. Zangerle, Overview of PAN 2024: Multi-author Writing Style Analysis, Multilingual Text Detoxification, Oppositional Thinking Analysis, and Generative AI Authorship Verification Condensed Lab Overview, in: CLEF (2), 2024, pp. 231–259. URL: https://doi.org/10.1007/978-3-031-71908-0_11.

[10] F. Jafariakinabad, K. A. Hua, Style-aware Neural Model with Application in Authorship Attribution, in: 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), IEEE, 2019, pp. 325–328.

[11] G. Verma, B. V. Srinivasan, A Lexical, Syntactic, and Semantic Perspective for Understanding Style in Text, ArXiv Preprint arXiv:1909.08349 (2019).

[12] Y. Sari, M. Stevenson, A. Vlachos, Topic or Style? Exploring the Most Useful Features for Authorship Attribution, in: Proceedings of the 27th International Conference on Computational Linguistics, 2018, pp. 343–353.

[13] B. Ai, Y. Wang, Y. Tan, S. Tan, Whodunit? Learning to Contrast for Authorship Attribution, ArXiv Preprint arXiv:2209.11887 (2022).

[14] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, in: J. Burstein, C. Doran, T. Solorio (Eds.), 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), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. URL: https://aclanthology.org/N19-1423/.

doi:10.18653/v1/N19-1423.

[15] M. Fabien, E. Villatoro-Tello, P. Motlicek, S. Parida, BertAA : BERT Fine-tuning for Authorship Attribution, in: P. Bhattacharyya, D. M. Sharma, R. Sangal (Eds.), Proceedings of the 17th International Conference on Natural Language Processing (ICON), NLP Association of India (NLPAI), Indian Institute of Technology Patna, Patna, India, 2020, pp. 127–137. URL: https://aclanthology.org/2020.icon-main.16/.

[16] H. Chen, Z. Han, Z. Li, Y. Han, A Writing Style Embedding Based on Contrastive Learning for Multi-Author Writing Style Analysis., in: CLEF (Working Notes), 2023, pp. 2562–2567.

[17] T. Lin, Y. Wu, L. Lee, Team NYCU-NLP at PAN 2024: Integrating Transformers With Similarity Adjustments For Multi-Author Writing Style Analysis, Working Notes of CLEF (2024).

[18] F. Feng, Y. Yang, D. Cer, N. Arivazhagan, W. Wang, Language-agnostic BERT Sentence Embedding, ArXiv Preprint arXiv:2007.01852 (2020).

[19] A. K. Ojha, A. S. Doğruöz, G. Da San Martino, H. T. Madabushi, R. Kumar, E. Sartori, Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023), in: Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023), 2023.

[20] K. Mnassri, P. Rajapaksha, R. Farahbakhsh, N. Crespi, BERT-based Ensemble Approaches For Hate Speech Detection, in: GLOBECOM 2022-2022 IEEE Global Communications Conference, IEEE, 2022, pp. 4649–4654.

[21] A. Kaas, V. T. Thomsen, B. Plank, Team DiSaster at SemEval-2020 Task 11: Combining BERT and Hand-crafted Features For Identifying Propaganda Techniques In News, in: SemEval 2020, Association for Computational Linguistics, 2020.

[22] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, Q. Liu, ERNIE: Enhanced Language Representation With Informative Entities, ArXiv Preprint arXiv:1905.07129 (2019).

[23] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, RoBERTa: A Robustly Optimized BERT Pretraining Approach, 2019. URL: https://arxiv.org/abs/1907.11692. arXiv:1907.11692.

[24] P. He, X. Liu, J. Gao, W. Chen, Deberta: Decoding-Enhanced Bert With Disentangled Attention, ArXiv Preprint arXiv:2006.03654 (2020).

[25] J. Bevendorff, X. B. Casals, B. Chulvi, D. Dementieva, A. Elnagar, D. Freitag, M. Fröbe, D. Korenčić, M. Mayerl, A. Mukherjee, A. Panchenko, M. Potthast, F. Rangel, P. Rosso, A. Smirnova, E. Stamatatos, B. Stein, M. Taulé, D. Ustalov, M. Wiegmann, E. Zangerle, Overview of PAN 2024: Multi-Author Writing Style Analysis, Multilingual Text Detoxification, Oppositional Thinking Analysis, and Generative AI Authorship Verification, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Fourteenth International Conference of the CLEF Association (CLEF 2024), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2024.

[26] 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. URL: https://link.springer.com/chapter/10.1007/978-3-031-28241-6_20. doi:10.1007/978-3-031-28241-6_20.

[27] 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, E. Duchesnay, Scikit-learn: Machine Learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.