Team Liuc0757 at PAN: A Writing Style Embedding Method Based on Contrastive Learning for Multi-Author Writing Style Analysis

Notebook for PAN at CLEF 2024

Chang Liu, Zhongyuan Han*, Haoyang Chen and Qingbiao Hu

Foshan University , Foshan, China

Abstract

This paper explores the identification of authorial style shifts in multi-author documents to facilitate further author identification. The model is proposed to employ contrastive learning techniques to analyze writing styles and optimized the sentence segment embedding output from the encoder of a pre-trained model. This optimization enables the encoder to produce more similar vector representations in space for sentences with similar styles while widening the distance between the embedding representations of paragraphs with different styles. By utilizing the contrastive learning-based encoder to generate sentence embeddings through an analysis of labeled data combined with paragraph sample pairs, we classified them using a fully connected layer. Experimental results demonstrate that we achieved F1 scores of 0.696, 0.717, and 0.503 on Task 1, Task 2, and Task 3 of the official test set, respectively.

Keywords

Style Change Detection, Contrastive Learning, Sentence Embedding

1. Introduction

In the realm of text analytics, discerning writing style shifts in multi-author documents is a complex yet fascinating task. This analysis not only verifies textual integrity but also identifies potential plagiarism. Recently, following the advancement of PAN series tasks with limited topic diversity, style analysis has garnered greater attention. In this context (PAN 2024 [1]), we shift focus from merely utilizing theme information as a style change signal to the actual writing style of the article itself.

2. Related Work

In the realm of text style analysis, traditional methods customarily depend on manually extracted features such as word frequency, part-of-speech tags, and sentence length statistics, or utilize models like TF-IDF [2], N-grams [3] to capture patterns in text. Regrettably, these approaches frequently face limitations due to the complexity of feature engineering and dependency on specific domains, making their application across diverse text types or fields challenging.

With the emergence of deep learning, models such as convolutional neural networks (CNN), recurrent neural networks (RNN), long short-term memory networks (LSTM), and transformers have been extensively utilized for text style analysis tasks. These models can autonomously learn intricate patterns in text and effectively differentiate between different text styles. For instance, RNN or LSTM's ability to process sequence data can capture context information in text [4] [5], while transformers enhance the model's global information capture through self-attention mechanisms [6].

CLEF 2024: Conference and Labs of the Evaluation Forum, September 09–12, 2024, Grenoble, France

lc965024004@gmail.com (C. Liu); hanzhongyuan@gmail.com (Z. Han*); hoyo.chen.i@gmail.com (H. Chen); ezio411152084@gmail.com (Q. Hu)

0009-0000-0887-9273 (C. Liu); 0000-0001-8960-9872 (Z. Han*); 0000-0003-3223-9086 (H. Chen); 0009-0004-8237-0044 (Q. Hu)

Despite significant advancements in text style analysis by deep learning models, they still confront certain challenges. Firstly, models typically necessitate substantial labeled data for training, which may be infeasible in some domains. Secondly, deep learning models often lack interpretability [7], making it difficult to comprehend how the model makes decisions. Lastly, for the task of detecting style changes in multi-author text, deep learning models need to simultaneously capture the writing styles of different authors and distinguish between them effectively.

In recent years, contrastive learning [8], a novel learning method, has demonstrated robust performance across various fields. Contrastive learning optimizes the model by maximizing similarity between positive sample pairs and minimizing similarity between negative sample pairs, enabling the model to learn more robust and generalizable feature representations. In the field of text style analysis, contrastive learning is also employed to optimize the representation of text embeddings to boost the model's performance in style classification tasks [9].

Previously, in style change detection tasks, Chen et al. [10], utilizing a pretrained RoBERTa [11] model achieved promising results. To address the "Collapse" [12] issue of traditional BERT models in text semantic matching, a computational approach using contrastive learning fine-tuned the encoder, computing cosine distance between sentences to learn improved sentence embedding representations. This method reduces distances among similar sentences in vector space while increasing those between dissimilar ones. By continuing this study's findings, we further explored 2024's highlighted topics and text styles on new datasets.

3. Task and Datasets

PAN 24 furnishes tasks [13] with three diverse complexity tiers. It necessitates identifying all positional alterations in the stylistic conventions at the paragraphed level within an assigned text (i.e., scrutinize for stylistic discrepancies between two successive paras). The primary distinction of these tasks lies in the range of document subject matters:

- **Easy:** The paragraphs of a document cover a variety of topics, allowing methods to make use of topic information to detect authorship changes.
- **Medium:** The topical variety in a document is small (though still present) forcing the methods to focus more on style to effectively solve the detection task.
- **Hard:** All paragraphs in a document are on the same topic.

All documents are provided in English and may contain an arbitrary number of style changes. However, style changes may only occur between paragraphs (i.e., a single paragraph is always authored by a single author and contains no style changes). Each input problem is referenced by an ID (i.e., the document that detects style changes), which is then used to identify the solution submitted for the input problem. The ground truth data includes the number of authors and the binary labels of each pair of consecutive paragraphs (1 for style changes, otherwise 0), but does not provide specific paragraph author information [10].

4. Method

Within this study, the contrastive learning method based on cosine phrase is used to train an encoder [10]. The objective is to position the encoded sentences of identical authors closer in space and those derived from diverse texts further apart. Post a certain level of training completion, we generate corresponding classification labels by connecting a fully connected layer classifier.

Figure 1 displays the model's comprehensive framework, where pairs of paragraphs undergo identical pooling strategy encoders and generate vector representations in the space. We've completed the text embedding phase with writing style incorporated. Concurrently, a feature matrix (u, v, |u − v|) is used to through contrastive analysis. This enables final classification via a FCNN classifier.

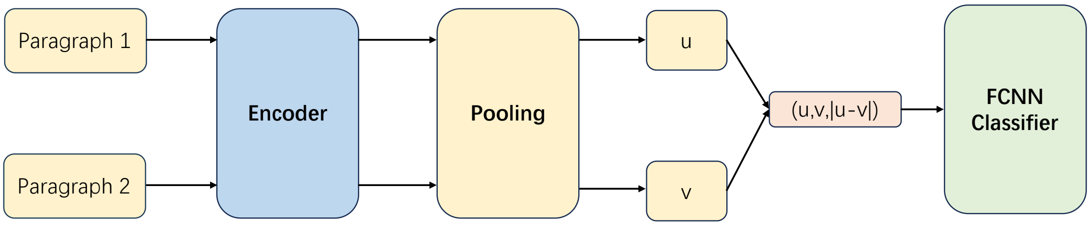


Figure 1: Model structure

4.1. Encoder Training

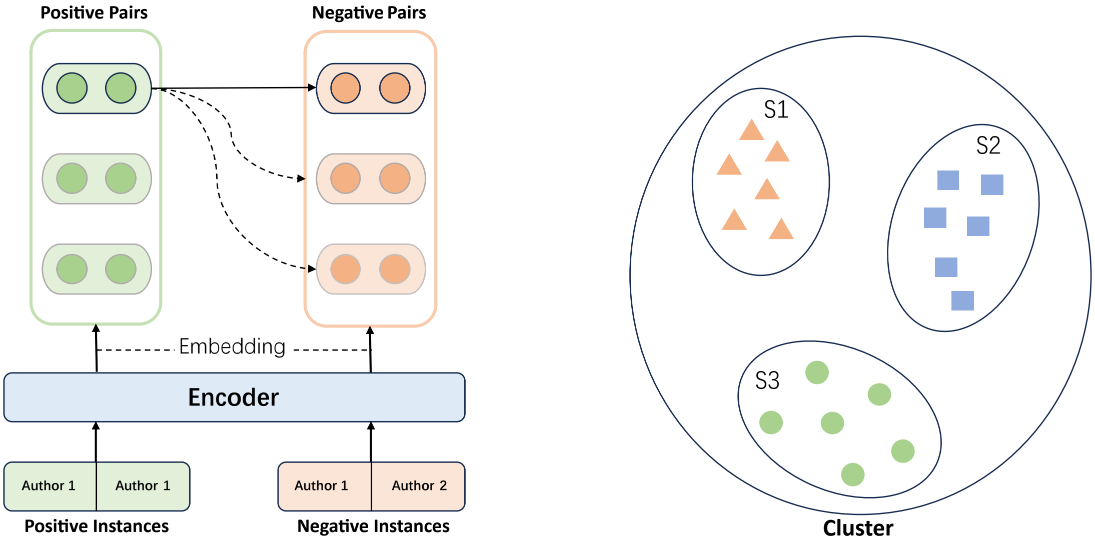
During the training phase of an encoder, each positive and negative instance pair is fed into the encoder for embedding. Ideally, the cosine distance between positive pairs should be less than that between negative pairs. That is, for any positive instances pair $(i, j) \in \Omega_{\mathrm{pos}}$ and negative instances pair $(k, l) \in \Omega_{\mathrm{neg}}$, there are:

$$\cos(u_{\mathrm{i}}, u_{\mathrm{j}}) > \cos(u_{\mathrm{k}}, u_{\mathrm{l}}) \tag{1}$$

Where u_{x} represents the embedding representation of the paragraph x. The work of Su et al., [14] [15]and Sun et al. [16] suggests an effective solution to such problems, here is the equation [10]:

$$L = \log(1 + \sum_{(i,j)\in\Omega_{\mathrm{pos}},(k,l)\in\Omega_{\mathrm{neg}}} e^{\lambda(\cos(u_{\mathrm{k}},u_{\mathrm{l}})-\cos(u_{\mathrm{i}},u_{\mathrm{j}}))}) \tag{2}$$

Where $\lambda > 0$ is a hyperparameter, which is taken as 20 in this experiment. The above equation is used to optimize the encoder, and the cosine distance of the encoder output instances is evaluated for correlation with the labels using the spearman metric, which assesses how well the relationship between two variables can be described using a monotonic function.



(a)

(b)

Figure 2: The figure on the left describes the encoder training conducted by cosine phrases base on contrastive learning, and the figure on the right shows the possible distribution of feature vectors on the plane after dimensionality reduction if three types of sentences are passed to the trained encoder.

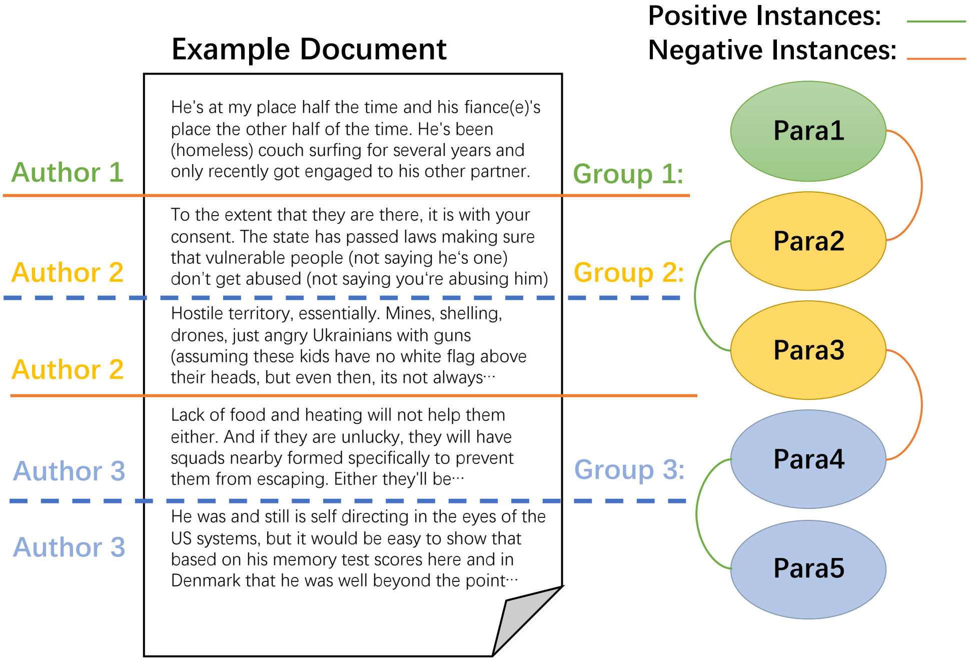
4.2. Classifier Training

Upon coding-encoder training completion, we freeze the parameters of the encoder, encode instances of the predefined paragraph pairs, extract the last layer of the model. We use the vectorial representations of these two matrices as paragraph embeddings (u, v), subtract them and take their absolute value, then concatenate them with the original matrix to form a feature matrix (u, v, |u - v|) [10]. This is fed into a linear layer activated by tanh for classification. The prediction results are optimized using cross-entropy loss and evaluated via F1-scores.

5. Experiments and Result

Firstly, it is imperative that we segment the official dataset to transform it into positive or negative instances corresponding to each paragraph. In the official data set, there are no definitive authorship information for each paragraph. This implies that if the sample transformation operation is based solely on a dataset containing binary labels, both the size and accuracy of the training set will be insufficient. (For instance, if two paragraphs written by the same author are mislabelled as negative instances while others are deemed as positive instances, that may confuses the model.) To circumvent this issue, we will employ the following strategy to generate positive and negative examples [10]:

- First, divide the paragraphs whose style has not changed into the same group, based on the labels.
- If the number of paragraphs in a group is greater than one, combine each of them in two to obtain a positive instance.
- Two-by-two combinations of negative instances between two adjacent but different groups.



He's at my place half the time and his fiance(e)'s place the other half of the time. He's been (homeless) couch surfing for several years and only recently got engaged to his other partner.

To the extent that they are there, it is with your consent. The state has passed laws making sure that vulnerable people (not saying he's one) don't get abused (not saying you're abusing him)

Hostile territory, essentially. Mines, shelling, drones, just angry Ukrainians with guns (assuming these kids have no white flag above their heads, but even then, its not always···

Lack of food and heating will not help them either. And if they are unlucky, they will have squads nearby formed specifically to prevent them from escaping. Either they'll be···

He was and still is self directing in the eyes of the US systems, but it would be easy to show that based on his memory test scores here and in Denmark that he was well beyond the point···

Figure 3: Dataset processing. Positive and negative instances are generated based on the division of the group.

Through this method shown in Figure 3, a significant volume of high-quality instance pairs can be obtained. Our method is exclusively utilized for the training set. For the validation set, we employ positive and negative instance pairs transformed from original labels only.

Secondly, we select the RoBERTa-base model as our pre-trained model and fed the prepared data to it. Our hyperparameters are set as follows: For the encoder, the batch size is set to 24, the maximum sequence length is 512, and the excess will be truncated. The initial learning rate is set to 1e-5, and trained in 10 epochs; For classifiers, the batch size is set to 64, the initial learning rate is set to 5e-5, and trained in 10 epochs.

The optimized result with the best F1 score of 0.8074 was accomplished by RoBERTa as pre-trained encoder model. The code and model were finally packaged in a docker file and submitted to Tira

Table 1
Metric on the test set.

Method	Task 1	Task 2	Task 3
F1-scores on Test set	0.696	0.717	0.503
Baseline Predict 1	0.466	0.343	0.320
Baseline Predict 0	0.112	0.323	0.346

platform for execution, leading to the final measure of the model's performance. Table 1 provides the scores obtained by our model in the official test set.

6. Acknowledgements

This work is supported by the Social Science Foundation of Guangdong Province, China (No. GD24CZY02).

References

[1] 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.

[2] Q. A. Bui, M. Visani, S. Prum, J.-M. Ogier, Writer identification using tf-idf for cursive handwritten word recognition, in: 2011 International Conference on Document Analysis and Recognition, IEEE, 2011, pp. 844–848.

[3] G. Ríos-Toledo, J. P. F. Posadas-Durán, G. Sidorov, N. A. Castro-Sánchez, Detection of changes in literary writing style using n-grams as style markers and supervised machine learning, Plos one 17 (2022) e0267590.

[4] Q. Chen, Q. Hu, J. X. Huang, L. He, Ca-rnn: using context-aligned recurrent neural networks for modeling sentence similarity, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 32, 2018.

[5] S. Santhanam, Context based text-generation using lstm networks, arXiv preprint arXiv:2005.00048 (2020).

[6] Y. Hao, L. Dong, F. Wei, K. Xu, Self-attention attribution: Interpreting information interactions inside transformer, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, 2021, pp. 12963–12971.

[7] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, L. Kagal, Explaining explanations: An overview of interpretability of machine learning, in: 2018 IEEE 5th International Conference on data science and advanced analytics (DSAA), IEEE, 2018, pp. 80–89.

[8] P. H. Le-Khac, G. Healy, A. F. Smeaton, Contrastive representation learning: A framework and review, Ieee Access 8 (2020) 193907–193934.

[9] A. Neelakantan, T. Xu, R. Puri, A. Radford, J. M. Han, J. Tworek, Q. Yuan, N. Tezak, J. W. Kim, C. Hallacy, et al., Text and code embeddings by contrastive pre-training, arXiv preprint arXiv:2201.10005 (2022).

[10] 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.

[11] P. He, X. Liu, J. Gao, W. Chen, Deberta: Decoding-enhanced bert with disentangled attention, arXiv preprint arXiv:2006.03654 (2020).

[12] Y. Yan, R. Li, S. Wang, F. Zhang, W. Wu, W. Xu, Consert: A contrastive framework for self-supervised sentence representation transfer, arXiv preprint arXiv:2105.11741 (2021).

[13] E. Zangerle, M. Mayerl, M. Potthast, B. Stein, Pan24 multi-author writing style analysis (2024). URL: https://zenodo.org/records/10677876.

[14] X. Huang, H. Peng, D. Zou, Z. Liu, J. Li, K. Liu, J. Wu, J. Su, P. S. Yu, Cosent: Consistent sentence embedding via similarity ranking, IEEE/ACM Transactions on Audio, Speech, and Language Processing 32 (2024) 2800–2813. doi:10.1109/TASLP.2024.3402087.

[15] J. Su, M. Zhu, A. Murtadha, S. Pan, B. Wen, Y. Liu, Zlpr: A novel loss for multi-label classification, arXiv preprint arXiv:2208.02955 (2022).

[16] Y. Sun, C. Cheng, Y. Zhang, C. Zhang, L. Zheng, Z. Wang, Y. Wei, Circle loss: A unified perspective of pair similarity optimization, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 6398–6407.