Text-Segment Interaction for Authorship Verification using BERT-based Classification

Notebook for PAN at CLEF 2023

Xurong Liu, Leilei Kong*, Mingjie Huang

Foshan University, Foshan, Index, China

Abstract

In this paper, we used a method based on BERT for text segmentation and combination; we divided long texts into segments and encoded them using BERT. By learning the semantic and contextual information from the original text, we calculated the similarity between the two segments. Then, we conducted a comprehensive evaluation of the overall similarity between the two texts to determine whether they were written by the same author. This model achieved good scores on the PAN2023 Authorship Verification dataset.

Keywords

Authorship verification, Text classification, Pre-training model

1. Introduction

In the field of NLP (Natural Language Processing), Authorship Verification refers to the task of identifying and attributing the author of a text based on its features and style[1]. Authorship Verification has significant applications in various domains, such as literature, law, and computational social science. It helps address issues like text fraud detection, copyright protection, literary research, and authorial style analysis[2-4]. For example, Authorship Verification can help identify the identity of anonymous writers or document authors and can also be used to detect plagiarism in academic papers[5].

This task aims to determine whether two texts are written by the same author. We extract features from these samples and train a model that can recognize unknown authors to perform Authorship Verification. However, there are challenges involved, including diverse text styles, imbalanced samples, cross-domain generalization, and so on. These are complex problems to address in the task of Authorship Verification[6-8].

In this study, the long texts are segmented, and BERT (Bidirectional Encoder Representations from Transformers) encodes the concatenated texts. By learning the semantics and contextual information from the original texts, we calculate the similarity and provide an overall evaluation of the two texts. This evaluation helps us determine the similarity between the two texts and ultimately make a conclusion about the result.

2. Datasets

Authorship verification task datasets consist of many English text pairs and pairs of texts from the Aston 100 Idiolects Corpus in English covering DTs (Discourse Types) of both written and spoken language: essays, emails, interviews, and speech transcriptions. Each sentence pair is assigned a unique identifier to distinguish between the same author pair and different author pairs. Additionally, metadata on the discourse type for each text in the pair is offered. Many tags are used within the text

[1]CLEF 2023: Conference and Labs of the Evaluation Forum, September 18--21, 2023, Thessaloniki, Greece
EMAIL: Liu_xurongx@163.com (A. 1); kongleilei@fosu.edu.cn (A. 2); mingjiehuang007@163.com (A. 3)
ORCID: 0009-0000-4386-5336 (A. 1); 0000-0002-4636-3507 (A. 2); 00000-0002-0889-5027 (A. 3)

to identify the original message's boundaries, new lines, and author-specific and topic-specific information, such as named entities. There are a total of 56 authors and 8,836 text pairs in the data text.

Table 1
Data information of Authorship verification datasets

Discourse Types	number
essays	2,549
emails	7,054
interviews	6,090
speech transcriptions	1,934

Table 1 shows the number of texts that appear in the dataset for four different text types. We performed simple text processing, including handling labels, emojis, and extra spaces. We replaced the labels with "<MASK>"[9], converted emoticons into corresponding words, and removed unnecessary spaces.

3. Method

3.1. Data Processing

In order to determine whether two variable-length texts, $text_1$, and $text_2$, are written by the same author, we divide each text into fixed-length segments: $text_1$: $\{text_{1\_1}, text_{1\_2}, text_{1\_3}, \ldots, text_{1\_n-1}, text_{1\_n}\}$ and $text_2$: $\{text_{2\_1}, text_{2\_2}, text_{2\_3}, \ldots, text_{2\_m-1}, text_{2\_m}\}$. Then, we combine and concatenate the corresponding segments of the two texts. If there are remaining text segments after the combination, we repeat the process by starting from the first segment of the other text and continue combining until all the text segments are joined. This results in pairs of combined text segments:
$\{[text_{1\_1}, text_{2\_1}], [text_{1\_2}, text_{2\_2}], [text_{1\_3}, text_{2\_3}], \ldots, [text_{1\_n-1}, text_{2\_m}], [text_{1\_n}, text_{2\_1}]\}$.

During the combination and concatenation process, we simultaneously label the pairs with a value indicating whether they are written by the same author or not: {0,1,0, ..., 1,1}. This allows us to extract features from these samples and train a model to recognize unknown authors. The model is built to determine whether the authors are the same based on the extracted features and training.

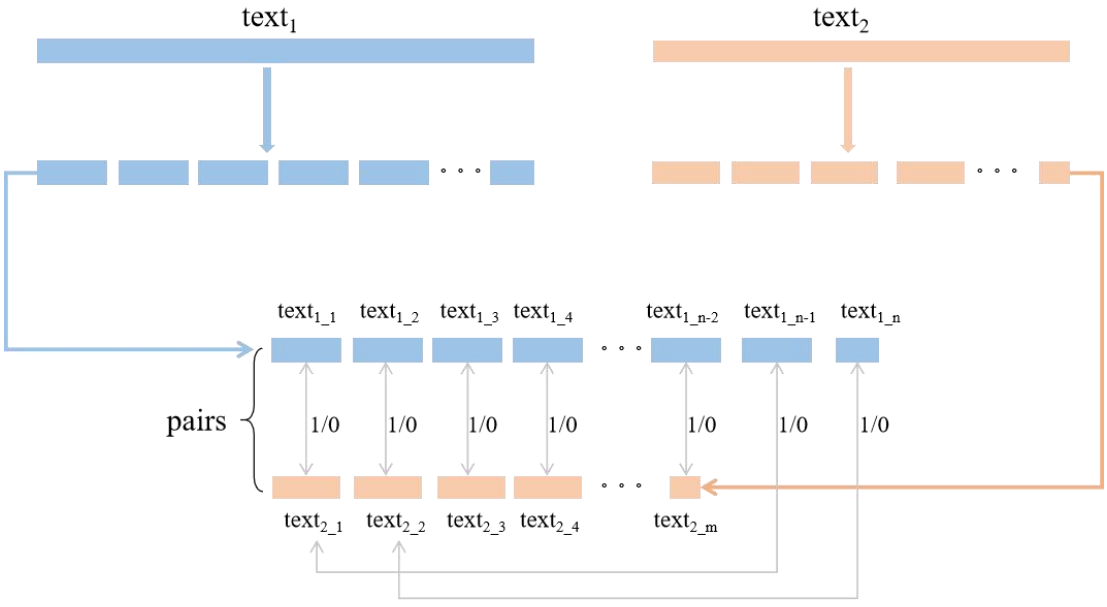

Figure 1: Data Process

The text preprocessing method corresponds to the content in the Figure 1; we added tags such as "[CLS]" and "[SEP]" to the text to differentiate between sentence pairs. Except for the last segment that is split, the length of each segmented text is kept as close to 256 tokens as possible[10]; this ensures that the maximum sentence length is achieved when feeding the data into the model, allowing for maximum feature extraction from these samples. The extracted features are then used to train the model. If one of the text segments has more splits than the other, the combination and concatenation process starts from the beginning segment of the other text, aiming to maximize the amount of training data for the model and ensure sufficient training.

3.2. Model

This paper utilizes BERT, a pre-trained language model based on the Transformer architecture, to fine-tune the model for the task of Authorship Verification in the field of natural language processing. BERT is trained in an unsupervised manner on a large-scale dataset, allowing it to learn rich semantic and contextual information. BERT introduces bidirectional context modeling, unlike traditional language models such as GPT (Generative Pretrained Transformer), that only consider the left or right context. By using self-attention mechanisms, BERT can consider the contextual information of all positions in a sentence simultaneously, capturing the semantic relationships more effectively[11].

In BERT, certain word fragments in the input text are randomly masked, and the model attempts to predict these masked fragments. This task encourages the model to learn the correlations between contexts, enabling it to correctly predict the masked fragments[12]. BERT takes a pair of consecutive sentences as input and determines whether they are continuous in the original text. This task helps the model learn the relationships and coherence between sentences[13].We input each pair of concatenated sentences into the model to obtain a result, and then take the average of each result as the final answer.

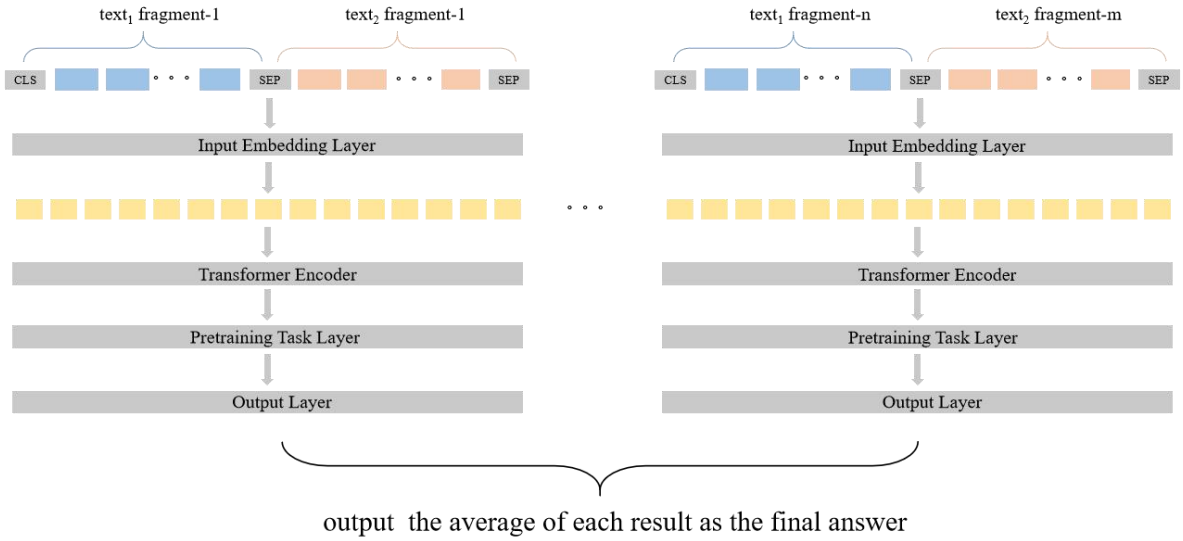

Figure 2: Model training process

4. Experiments and Results

4.1. Experimental setting

In this work, $BERT_{base}$ is used with 12 layers, 768 hidden units, and 12 attention heads, allowing the model to attend to the input with different attention weights. BERT and a fully connected network classification model are built using Keras. Firstly, we employ a method to find the most suitable

learning rate, which utilizes a 1cycle learning rate policy[14]. Then, we train the model using one of the three learning rates recommended in the BERT paper: 2e-5[11].

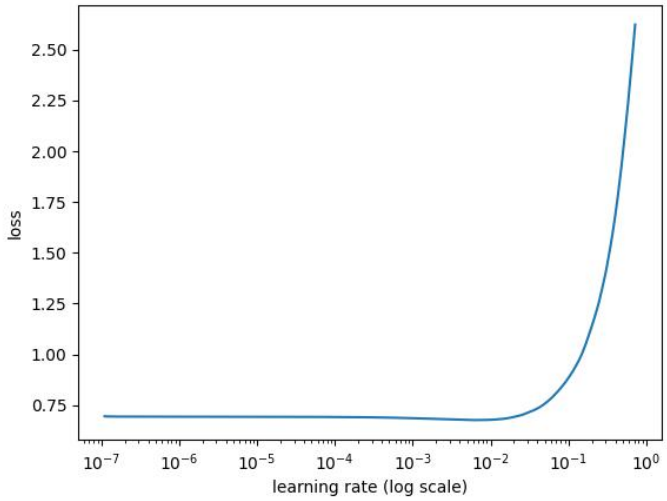

Figure 3: Find suitable learning rate

A batch size of 32 is set, and sparse categorical cross-entropy is used as the loss function. The optimization method employed is Adam. The concatenated text is inputted into the model for training. The model predicts and evaluates the individual segments of the combined text. Once predictions are obtained for all text segments, an overall evaluation is performed to determine the final answer regarding whether the texts are written by the same author.

4.2. Results

We randomly extract data from the dataset according to an 8:2 ratio, with 80% of the data being used as the training set and 20% as the test set.We saved the fine-tuned model with high accuracy and proceeded to validate it using our self-organized dataset. Our objective was to identify the model that would yield the best results. Here are the top five overall validation experiment results:

Table 2
Better training results on the training set

epoch	AUC	C@1	f_05_u	F1	brier	overall
24	0.941	0.941	0.928	0.943	0.941	0.939
19	0.940	0.940	0.936	0.941	0.940	0.940
26	0.944	0.945	0.932	0.946	0.945	0.942
20	0.945	0.945	0.932	0.947	0.945	0.943
23	0.952	0.952	0.943	0.953	0.952	0.950
25	0.950	0.950	0.947	0.951	0.950	0.950

Table 3
Better validation results on the validation set

epoch	AUC	C@1	f_05_u	F1	brier	overall
12	0.538	0.538	0.429	0.292	0.538	0.467
3	0.540	0.540	0.429	0.289	0.540	0.467
9	0.538	0.538	0.430	0.292	0.538	0.467
5	0.540	0.540	0.430	0.291	0.540	0.468
7	0.539	0.539	0.431	0.292	0.539	0.468
8	0.540	0.540	0.438	0.301	0.540	0.472

It can be observed that the performance of the saved model during training differs from its performance on the validation data. The model at epoch 8 did not yield the best results during training, but it demonstrated excellent performance across various evaluation metrics during validation.

We have uploaded three submitted runs on TIRA[15]: 'coincident-sound', 'foggy-raster', and 'perpendicular-field'. 'coincident-sound' calculates the length of sentences by counting the number of tokens using BERT tokenizer and submits the best overall validation result. 'foggy-raster' and 'perpendicular-field' calculate the length of sentences by counting the number of words separated by spaces and submit the best results for validation AUC and F1. The results on the official training set and test set are shown in the following table:

Table 4
The results on the official training set

Run Name	AUC	C@1	f_05_u	F1	brier	overall
coincident-sound	0.930	0.930	0.908	0.933	0.930	0.926
foggy-raster	0.958	0.958	0.965	0.958	0.958	0.959
perpendicular-field	0.977	0.977	0.978	0.977	0.977	0.978

Table 5
The results on the official test set

Run Name	AUC	C@1	f_05_u	F1	brier	overall
coincident-sound	0.548	0.548	0.547	0.544	0.548	0.547
foggy-raster	0.533	0.533	0.493	0.424	0.533	0.503
perpendicular-field	0.534	0.534	0.493	0.421	0.534	0.503

5. Conclusion

In this study, a BERT-based method for text segmentation and combination was employed to determine whether two texts were written by the same author. As shown in Tables 2, 3, and 4, good performance results were obtained during training. However, there was a significant discrepancy between the results obtained during validation and training, indicating that the experiments may not have fully considered issues such as diverse text styles, imbalanced samples, and cross-domain generalization.

6. Acknowledgments

This work is supported by the National Social Science Foundation of China (No. 22BTQ101).

7. References

[1] Stamatatos, E. (2009). A survey of modern authorship attribution methods. Journal of the American Society for information Science and Technology, 60(3), 538-556.

[2] Maurer, H. A., Kappe, F., & Zaka, B. (2006). Plagiarism-A survey. J. Univers. Comput. Sci., 12(8), 1050-1084.

[3] Stamatatos, E., Kredens, K., Pęzik, P., Heini, A., Bevendorff, J., Potthast, M., & Stein, B. (2023). Overview of the Authorship Verification Task at PAN 2023. In CLEF 2023 Labs and Workshops, Notebook Papers. Conference and Labs of the Evaluation Forum (CLEF 2022). CEUR-WS.org.

[4] Bevendorff, J., Borrego-Obrador, I., Chinea-Ríos, M., Franco-Salvador, M., Fröbe, M., Heini, A., Kredens, K., Mayerl, M., Pęzik, P., Potthast, M., Rangel, F., Rosso, P., Stamatatos, E.,

Stein, B., Wiegmann, M., Wolska, M., & Zangerle, E. (2023). Overview of PAN 2023: Authorship Verification, Multi-Author Writing Style Analysis, Profiling Cryptocurrency Influencers, and Trigger Detection. In A. Arampatzis, E. Kanoulas, T. Tsikrika, S. Vrochidis, A. Giachanou, 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). Thessalonikki, Greece: Springer.

[5] Zheng, R., Li, J., Chen, H., & Huang, Z. (2006). A framework for authorship identification of online messages: Writing - style features and classification techniques. Journal of the American society for information science and technology, 57(3), 378-393.

[6] L'heureux, A., Grolinger, K., Elyamany, H. F., & Capretz, M. A. (2017). Machine learning with big data: Challenges and approaches. Ieee Access, 5, 7776-7797.

[7] Mozafari, M., Farahbakhsh, R., & Crespi, N. (2020). Hate speech detection and racial bias mitigation in social media based on BERT model. PloS one, 15(8), e0237861.

[8] Crawford, M., Khoshgoftaar, T. M., Prusa, J. D., Richter, A. N., & Al Najada, H. (2015). Survey of review spam detection using machine learning techniques. Journal of Big Data, 2(1), 1-24.

[9] Wei, C., Fan, H., Xie, S., Wu, C. Y., Yuille, A., & Feichtenhofer, C. (2022). Masked feature prediction for self-supervised visual pre-training. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 14668-14678).

[10] Aggarwal, A., Chauhan, A., Kumar, D., Verma, S., & Mittal, M. (2020). Classification of fake news by fine-tuning deep bidirectional transformers based language model. EAI Endorsed Transactions on Scalable Information Systems, 7(27), e10-e10.

[11] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[12] Joshi, M., Chen, D., Liu, Y., Weld, D. S., Zettlemoyer, L., & Levy, O. (2020). Spanbert: Improving pre-training by representing and predicting spans. Transactions of the Association for Computational Linguistics, 8, 64-77.

[13] Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao, J. (2021). Deep learning--based text classification: a comprehensive review. ACM computing surveys (CSUR), 54(3), 1-40.

[14] Maiya, A. S. (2022). ktrain: A low-code library for augmented machine learning. The Journal of Machine Learning Research, 23(1), 7070-7075.

[15] Fröbe, M., Wiegmann, M., Kolyada, N., Grahm, B., Elstner, T., Loebe, F., Hagen, M., Stein, B., & Potthast, M. (2023). 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) (pp. 236-241). Berlin Heidelberg New York: Springer.