Style Change Detection: Method Based On Pre-trained Model And Similarity Recognition

Notebook for PAN at CLEF 2022

Xinyin Jiang, Haoliang Qi*, Zhijie Zhang, and Mingjie Huang

Foshan University, Foshan, China

Abstract

For the Style Change Detection task, given a document, the purpose is to determine the number of authors and where authors change. We treat it as a classification problem. For Task1, given a text written by two authors that contains a single style change only, find the position of this change. Task2 is to find all locations where the writing style has changed on given text written by two or more authors. Finally, Task3 is to determine the position of all writing style changes for a text written by two or more authors. Now the style changes occur not only between paragraphs but also at the sentence level. This paper proposes a method of Writing Style Similarity. We treat Task1 and Task3 as binary classification and Task2 as multi-classification. We use ELECTRA, which is a better model than BERT in discriminating tasks, and we choose different versions of models for various tasks. Our approach offers several opportunities for further research. The F1 scores of Task1, Task2, and Task3 are 0. 7346, 0. 4687 and 0. 6720 respectively. Among them, Task2 added two evaluation indicators this year: Diarization Error Rate (DER) and Jaccard Error Rate (JER). Our scores are 0. 2380 and 0. 3138.

Keywords

Style Change Detection, Pre-trained Model, Style Similarity, Paragraph-authors

1. Introduction

Style Change Detection is an essential task at present. Writing style tests allow us to determine whether the author of a document has plagiarized and from which parts. Style Change Detection task aims to identify the text location where the author switches in a given multi-author document. If multiple authors write an essay together, is there any evidence of that fact? Is there any way to detect changes in writing style? This problem is among the most difficult and interesting challenges in authorship identification. If no comparative text is given, Style Change Detection is the only way to detect plagiarism in a document. Similarly, Style Change Detection can help discover authorship, verify authorship, or develop new techniques for authorship verification [4].

A pre-trained language model has been shown to improve many natural language processing tasks [5][6]. These include sentence-level tasks such as natural language inference [7][8] and paraphrasing [9], which aim to predict the relationships between sentences by analyzing them holistically [10]. We use ELECTRA [11], a better model than BERT [10], in discriminating tasks. We have proposed replacing token detection, a new self-supervised task for language representation learning. The key idea is training a text encoder to distinguish input tokens from high-quality negative samples produced by a small generator network. Compared to masked language modeling, our pre-training objective is more compute-efficient and results in better performance on downstream tasks. It works well even when using relatively small amounts of computing [11]. ELECTRA's approach perfectly solves the problem of low expected utilization of MLM tasks, ELECTRA [11] uses adversarial

CLEF 2022 – Conference and Labs of the Evaluation Forum, September 5-9, 2022, Bologna,Italy
EMAIL: singinjiang@gmail.com (A.1); qihaoliang@fosu.edu.cn (A.2)(*corresponding author); zhangzhijie5454@gmail.com (A.3)
ORCID: 0000-0002-7926-8581 (A.1); 0000-0003-1321-5820 (A.2); 0000-0002-4636-3507 (A.3)


training and uses a small version of BERT [10] as a generator to predict the words to be masked. After analyzing the objectives of different tasks, we conceive of the Style Change Detection as discovering the similarity of writing styles between different text paragraphs. The tasks of Style Change Detection are commonly recognized as separate tasks, and various models are implemented to solve the respective issues of each task [4].

2. Data

The Style Change Detection provided three datasets. To develop and then test algorithms, three datasets, including ground truth information, are provided [12]:

Training set: This contains 70% of the whole data set and includes ground truth data.

Validation set: This contains 15% of the whole data set and includes ground truth data.

Test set: This contains 15% of the whole data set. No ground truth data is given.

The following analysis was performed on the three data sets-statistics of the data set as shown in Table 1.

Table 1
Statistics of data set:

Number of texts	Task1	Task2	Task3
training	1400	7000	7000
validation	300	1500	1500
test	300	1500	1500

3. Method

After analyzing the datasets, the datasets of Task1 and Task2 are multiple paragraphs of text, in which each paragraph consists of multiple sentences. Task1 requires distinguishing the author category by paragraph and output binary. Task2 is to output 1, 2, 3, and 4 for authors who need to identify each paragraph uniquely. Another difference is the amount of data. The data of Task2 is five times that of Task1, which greatly increases the cost of training. Task3 is a one-sentence paragraph, making it even more expensive to train.

Through reference [4], we learned that in this paper, paragraph features are extracted with the popular pre-training model BERT [10] for estimating the similarity of such writing styles. A model was built to accomplish three tasks simultaneously. But for this task, we found that BERT [10] doesn' t perform well and is costly.

3.1. Identify authors based on similar writing styles

For a given token, its input representation is constructed by summing the corresponding token, segment, and position embeddings [10]. We believe that the two paragraphs can be used for similarity measurement. We take the similarity of the paragraphs as the criterion for judging whether the two paragraphs belong to the same author. If there is a high similarity between two paragraphs, then we regard them as belonging to the same author, and if the similarity is low, then the opposite. Therefore, it turns into a binary classification. There are many paragraphs in a document. Separate them first. Then judge whether the style of each paragraph and its preceding paragraph has changed. We choose to label each paragraph. For the first paragraph, set the first paragraph as 1; if there is a change in the next paragraph, the label is 1. Otherwise, the label is 0, and each next paragraph should be compared with the previous paragraph. If there is any change, the new label is 1. Otherwise, the new label will be 0. In Task2, the paragraph author tags include 1,2,3,4. It turns into a multi-classification. Here we do not perform the binary conversion.

3.2. Model

What we use is a pre-training model ELECTRA [11] and a Fully Connected Neural Network Classifier. ELECTRA [11] is a method for self-supervised language representation learning. It can be used to pre-train transformer networks using relatively little computing. After pre-training, they throw out the generator and only fine-tune the discriminator (the ELECTRA [11] model) on downstream tasks. ELECTRA [11] achieves higher downstream accuracy than BERT [10] when fully trained. BERT [10] masks 15% of all WordPress tokens in each sequence at random. Although this allows us to obtain a bidirectional pre-trained model, a downside is that it creates a mismatch between pre-training and fine-tuning since the [MASK] token does not appear during fine-tuning.

Different from the previous "two-way" model, it is from the level of semantic understanding. The model needs to "understand" the context meaning to make a correct prediction. But MLM missions have their pitfalls. ELECTRA's approach perfectly solves the problem of low expected utilization of MLM tasks, ELECTRA [11] uses adversarial training and uses a small version of BERT [10] as a generator to predict the words to be masked. The predicted sentence is fed to a discriminator, which classifies each character and determines whether each word has been replaced. The structure of the discriminator is almost the same as BERT's, except that it distinguishes the lexical dimension from the hidden dimension, just like ALBERT [14]. This operation greatly improves training efficiency and allows the lightweight $\text{ELECTRA}_{\text{Base}}$ to replace $\text{BERT}_{\text{Base}}$. Also, ELECTRA [11] doesn't share layer-to-layer parameters like ALBERT [14] does, which makes it a better fit. It can be seen that ELECTRA's comprehensive performance in time, space, and effect is very advantageous and can almost perfectly replace BERT [10].

ELECTRA [11] is more parameter-efficient than BERT [10] because it does not have to model the full distribution of possible tokens at each position [11]. It changes the generated Masked Language Model (MLM) pre-trained task into the discriminative Replaced Token Detection (RTD) task to judge whether the language model has replaced the current token, uses an MLM-based generator to replace some tokens in the example, and then throws it to the discriminator for discrimination, And through the method of parameter sharing to reduce parameters, to improve the learning efficiency of parameters. The most important thing is that the computing power is reduced and the efficiency is improved. It is selected in this experiment.

For these three missions, we used two ELECTRA [11] of different sizes, $\text{ELECTRA}_{\text{Base}}$ and $\text{ELECTRA}_{\text{Large}}$. Where, $\text{ELECTRA}_{\text{Base}}$ configuration parameters:layers=12, hidden size=768, total parameters=110M, and $\text{ELECTRA}_{\text{Large}}$:layers=24, hidden size=1024, total parameters=335M. According to the size of the task data set, Task1 chooses to use $\text{ELECTRA}_{\text{Large}}$, while Task2 and Task3 use $\text{ELECTRA}_{\text{Large}}$ to complete training.

As shown in Figure 1, we input the paragraphs of the document into the model and classify them, and then output the similarity label.

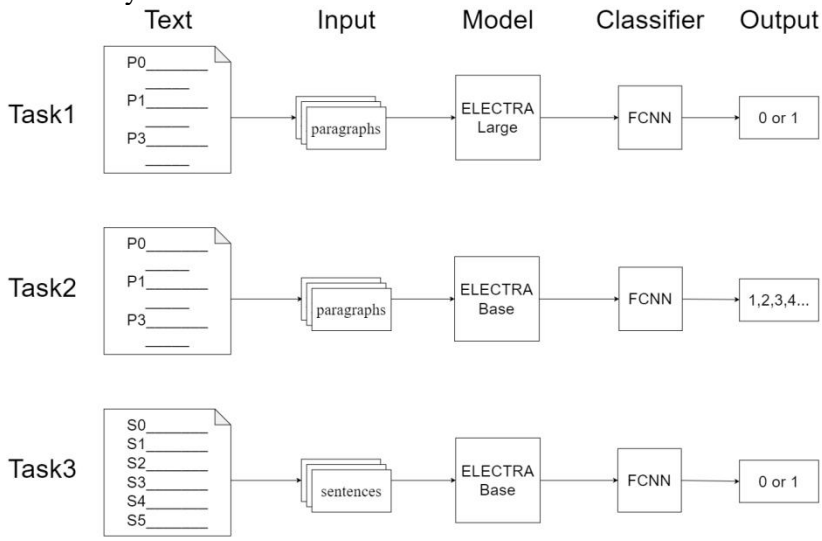

Figure 1: Architecture of the whole model

4. Experiments

In our method, we mainly use a popular pre-trained model ELECTRA [11]. Different versions are adopted according to the amount of data. Before entering the paragraph into the model, we set the maximum length of the paragraph to 256 and 128 to analyze the effect. The maximum length is the sum of the two lengths. Each paragraph can be divided into at least one sentence. Intuitively, when the maximum length is 256, we should retain as much information as possible to achieve a good classification effect. However, when we compare the results, they are similar. It is suggested that the paragraph 2 classification task may not need too much information, and the length of truncated paragraphs should not be too long. In this section, we did some experiments. We evaluated this part using validation sets. The metric used is accuracy; the results are shown in Table 2. Because the results are similar, we choose 128 on Task1 and 64 on Task2 and Task3 because it saves running space and time. After training two epochs, we can get good results.

Table 2
The result of the validation set:

measure	maxlen	batch_size	model	val_accuracy
Task1	256	4	BERT	0.79028
Task1	128	64	BERT	0.82345
Task1	64	32	BERT	0.81878
Task1	64	32	ELECTRA	**0.83372**
Task1	128	64	ELECTRA	0.82812
Task2	256	4	BERT	0.76334
Task2	64	32	BERT	0.73786
Task2	64	32	ELECTRA	0.75076
Task2	128	64	BERT	0.77834
Task2	128	64	ELECTRA	**0.78410**
Task3	64	32	BERT	0.66967
Task3	64	64	BERT	0.68803
Task3	64	32	ELECTRA	0.54567
Task3	128	64	ELECTRA	**0.70703**

5. Results

The trained model is used for evaluation with the officially provided test set, and the results are shown in Table 3.

Table 3
The result of the test set:

measure	score
Task1. F1-score	0. 7346
Task2. F1-score	0. 4686
Task2. DER	0. 2380
Task2. JER	0. 3138
Task3. F1-score	0. 6720

6. Conclusion

This paper proposes a method based on a pre-trained model and similarity recognition. Style change and authorship verification tags are regarded as multi-classification tasks based on writing style similarity, and the pre-trained model is used to estimate writing style similarity. With the method proposed in this paper, three tasks of Style Change Detection can be realized. Finally, we get the F1 scores of Task1, Task2 and Task3 are 0. 7346, 0. 4686 and 0. 6720 respectively. Among them, Task2 added two evaluation indicators this year: Diarization Error Rate (DER) and Jaccard Error Rate (JER). Our scores here are 0. 2380 and 0. 3138 respectively.

7. Acknowledgments

This work is supported by the Social Science Foundation of Guangdong Province (No. GD20CTS02).

8. References

[1] J. Bevendorff, B. Chulvi, E. Fersini, A. Heini, M. Kestemont, K. Kredens, M. Mayerl, R. Ortega-Bueno, P. Pezik, M. Potthast, F. Rangel, P. Rosso, E. Stamatatos, B. Stein, M. Wiegmann, M. Wolska, E. Zangerle: Overview of PAN 2022: Authorship Verification, Profiling Irony and Stereotype Spreaders, and Style Change Detection, in: A. B. Cedeno, G. D. S. Martino, M. D. Esposti, F. Sebastiani, C. Macdonald, G. Pasi, A. Hanbury, M. Potthast, G. Faggioli, N. Ferro (Eds.), Proceedings of the Thirteenth International Conference of the CLEF Association (CLEF 2022), Springer, 2022.

[2] E. Zangerle, M. Mayerl, M. Potthast, B. Stein: Overview of the Style Change Detection Task at PAN 2022. Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, CEUR-WS.org (2022)

[3] M. Potthast, T. Gollub, M, Wiegmann, B. Stein: TIRA Integrated Research Architecture, in: Information Retrieval Evaluation in a Changing World, ser. The Information Retrieval Series, N. Ferro, C. Peters, Berlin Heidelberg New York: Springer, Sep. 2019.

[4] Zhijie Zhang, Zhongyuan Han, Leilei Kong, et al. Style Change Detection Based On Writing Style Similarity. Notebook for PAN at CLEF 2021

[5] Andrew M Dai and Quoc V Le. 2015. Semi-supervised sequence learning. In Advances in neural information processing systems, pages 3079–3087.

[6] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018a. Deep contextualized word representations. In NAACL.

[7] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In EMNLP. Association for Computational Linguistics.

[8] Adina Williams, Nikita Nangia, and Samuel R. Bow-man. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In NAACL.

[9] William B Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In Proceedings of the Third International Workshop on Paraphrasing (IWP2005). I. Editor (Ed.), The title of book one, volume 9 of The name of the series one, 1st. ed. , University of Chicago Press, Chicago, 2007. doi:10. 1007 3-540-09237-4.

[10] Devlin J. , Chang M. W. , Lee K. , et al. Bert: Pre-trained of deep bidirectional transformers for language understanding[C]//Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2019, 1: 4171-4186

[11] Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. Electra: Pre-trained text encoders as discriminators rather than generators. In Proceedings of ICLR, 2019.

[12] E. Zangerle, M. Mayerl, M. Tschuggnall, M. Potthast, and B. Stein, “ Pan22 authorship analysis: Style change detection, ” 2022. https://zenodo.org/record/6334245
[13] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and V eselin Stoyanov. RoBERTa: A robustly optimized BERT pre-training approach. arXiv preprint arXiv:1907.11692, 2019.
[14] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A lite bert for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942, 2019.