Author Authentication of Generative AI Based on Bert by Regularization Method

Notebook for PAN at CLEF 2024

Baijian Huang, Changle Zhong*, Kai Yan and Yong Han

Foshan University, Foshan, China

Abstract

Rapid development and widespread use of large language models (LLMs) have made distinguishing between human-written and machine-generated text increasingly challenging. This paper proposes a novel author verification method for generative AI based on a BERT model to address this issue. The model is trained by concatenating input text. By using the Regularized Dropout(R-Drop) method to constraint output and analyze similarities and differences between various samples, the model learns internal data characteristics and identifies unique features of different authors' writing styles. Experimental results of a bert model trained using regularization techniques show that the model has an ROC-AUC measure of 0.942, ranking 13th out of 30 in the submission ranking and beating all baselines. The experimental results show that the proposed method can effectively distinguish between human-generated text and machine-generated text in different test scenarios, providing a robust solution for author verification in the context of generative AI.

Keywords

Authorship Verification, BERT, Regularized Dropout

1. Introduction

The author verification task aims to determine which of two texts was generated by a human by analyzing various linguistic features and stylistic elements of the text. Effective author authentication techniques enable the identification of human-generated articles in fields such as literature, law, and journalism. This process is crucial for ensuring texts' authenticity, enhancing textual content's quality and credibility, combating plagiarism, and maintaining academic integrity, among other important objectives.

PAN at CLEF 2024 [1] introduces a task where participants are given pairs of texts on the same topic—one authored by a human and the other by a machine—and are challenged to identify the human-written text. The Generative AI Authorship Verification Task [2], in collaboration with the Voight-Kampff Task ELOQUENT Lab, adopts a builder-breaker format. PAN participants will develop systems to differentiate between human-written and AI-generated texts, while ELOQUENT participants will explore new text generation and obfuscation techniques to evade detection [3]. Our approach consists of two key steps: First, we describe the same topic for two texts in the PAN 2024 text dataset, which we form into a pair of texts where one human-generated text is linked to each of the other 13 machine-generated texts.Each combined text (text1 and text2) is truncated to ensure that its total length is within 512 token. In addition to competing datasets, we also introduced an external dataset Kaggle dataset: LLM-Detect AI Generated Text Train [1] to enhance our generalization of the model. The training set of the Kaggle competition dataset consists of essays written for two of the seven prompts. Almost all of the essays in the training set are written by students, and only a few examples of generated essays are included. It does not distinguish between topics and treats them with random machine-generated

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

*Corresponding Author

huangbaijianh@gmail.com (B. Huang); 103409103@qq.com (C. Zhong); yankai@fosu.edu.cn (K. Yan); hanyong2005@fosu.edu.cn (Y. Han)

0009-0001-3792-6910 (B. Huang); 0009-0008-3044-2383 (C. Zhong); 0000-0002-4960-7108 (K. Yan); 0009-0001-4094-006X (Y. Han)

[1] This data set can be found in https://www.kaggle.com/competitions/llm-detect-ai-generated-text/data

text. Finally, the model is trained with (*t1*,*t2*) as the sample. After fine-tuning the BERT model under the transformers library as our base model, we verify the authorship of the generated AI through the ability to embed a powerful pre-trained context.

To solve and mitigate the overfitting problem, we also implemented BERT model by using the R-Drop method [4]. The R-Drop method is a regularization technique designed to reduce variance and improve the robustness of BERT models by ensuring that the model's predictions remain consistent across multiple dropout masks and enhance classification ability, Kullback-Leibler (KL) divergence is used to ensure output consistency, and dropout is used to make slightly different model paths.

2. Related Work

In the realm of AI text detection, numerous methodologies have been proposed. Prominent examples include GROVER[5] and GLTR[6]. GROVER generates news snippets and incorporates a fine-tuned generation model for snippet detection. Conversely, GLTR renders machine-generated text discernible by computing each word's histogram of logarithmic likelihood values. Furthermore, DetectGPT[7] generates text variants by perturbing the model and evaluating the probabilistic curvature of these variants. Recently, Fast-DetectGPT[8] has enhanced this approach by estimating conditional probability curvature via a single model forward pass, thus substantially enhancing detection efficiency and accuracy. These investigations underscore the significance and potential of statistical features and language models in the realm of machine-generated text detection for PAN's generative AI author verification task. Conducting detailed comparative experiments requires a lot of computing resources and time. My research is limited in resources and time, so I focus on my methods and results. Inspired by these methods, we combined regularization and processed the data to improve the author's verification task. The subsequent experimental results also demonstrated the effectiveness of our method. The accuracy of the test results in multiple datasets is 1, and the average accuracy of the results is 0.942.

2.1. Data Imbalance

In the field of AI text detection, data imbalance is a common and important challenge. Data imbalance means that the number of samples in some categories is much smaller than that in other categories, which causes the model to favor majority-class samples and ignore minority-class samples during training, thus affecting the detection effect. There are many ways to deal with data imbalance. Traditional methods, such as oversampling and cost-sensitive learning, are still effective, and new methods, such as deep learning and transfer learning, show better prospects. Combining and optimizing these methods can help improve the detection performance of the model on imbalanced datasets and promote the further development of text detection tasks.

2.2. R-Drop manner

The concept of R-Drop[4] entails each data sample traversing the same model with dropout repeatedly. Subsequently, KL divergence is utilized twice to constrain the output, maximizing its consistency. Owing to the stochastic nature of dropout, the path network traversed by input x can be approximated as two subtly different models. This process is illustrated in Figure 1.

One solution to balance the detection of semantic and non-semantic transfer is to calculate the distance score in the feature space of both the pre-trained and fine-tuned models. This approach accounts for the detection of both types of transfer cases.

3. Model

We improve the performance of the model by training a BERT model fine-tuned by the R-Drop method.

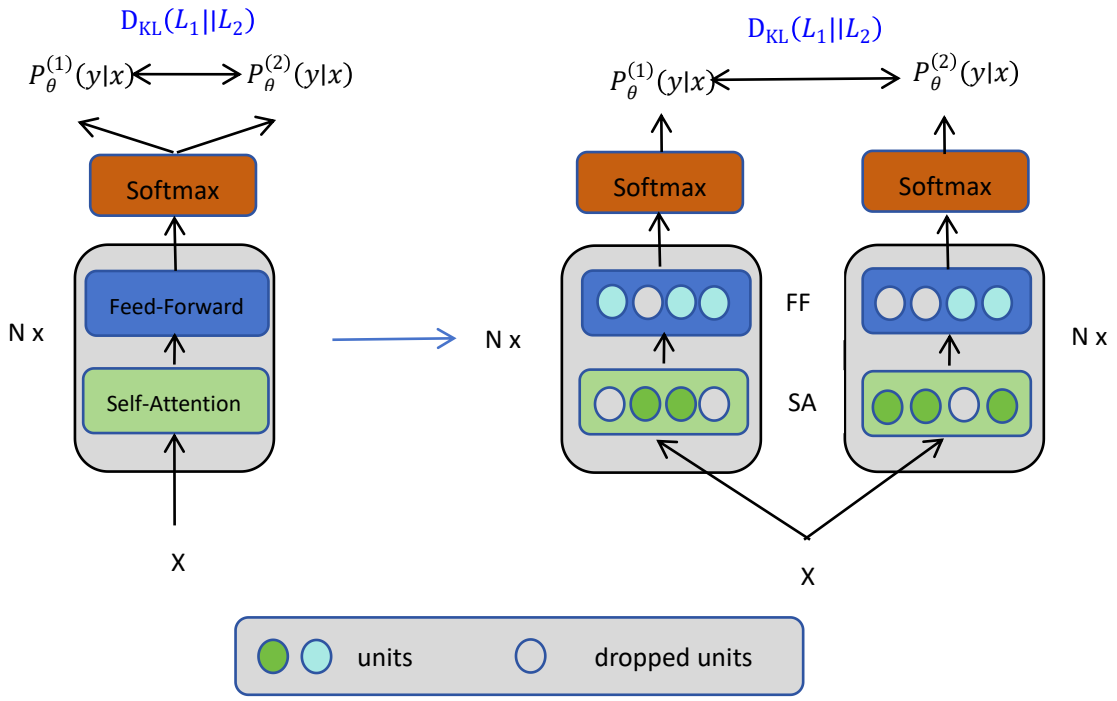

Figure 1: R-Drop manner[4] overview

3.1. Model architecture

We introduce the R-Drop method into the training process. Firstly, the input text is encoded through the BERT coding layer to generate the representation vector. These vectors are processed through the Dropout layer to prevent overfitting. The vectors are then mapped through a linear classification layer to the classification label space, generating multiple outputs. The regularization term is increased by calculating the KL divergence between each pair of outputs. Finally, the loss function combines KL divergence and classification loss, and updates the model parameters through backpropagation, thus completing the whole model training process. The overall architecture of the model is shown in Figure 2.

R-Drop imposes consistency constraints on the output layer of the model, which may indirectly affect the weight update and learning process of each layer within the model (including the dropout layer). Specifically, since R-Drop requires the model to produce similar outputs under different dropout settings, this may prompt the model to learn more robust and universal feature representations during training, thereby affecting the weights and outputs of the input embedding, multi-head attention, feed-forward, and other layers. The R-Drop method mainly acts on the BERT model's output layer and improves the model's generalization ability and robustness by imposing consistency constraints. It may indirectly affect the weight update and learning process of the dropout layer of the input embedding, multi-head attention, or feed-forward layer.

3.2. Loss function

The loss function is as follows: The training data is (x_i, y_i) Where x_i represents the input feature, y_i represents the label or target value corresponding to x_i, and the model uses (x_i, y_i) in the training data to learn how to predict the target value y_i from the input feature x_i. , the model is $P_\theta(x_i|y_i)$, and the cross entropy of each sample is:

$$L_i = -\log P_\theta(y_i|x_i) \tag{1}$$

In the case of "dropout twice," we can assume that the sample has passed through two slightly different

models, denoted as:

$$L_1 = -\log P_\theta^{(1)}(y_i|x_i) - \log P_\theta^{(2)}(y_i|x_i) \tag{2}$$

The other part of the loss function is the KL divergence, which aims to make the two outputs as consistent as possible:

$$L_2 = \frac{1}{2}\left[\mathrm{KL}(P_\theta^{(1)}(y_i|x_i) \parallel P_\theta^{(2)}(y_i|x_i)) + \mathrm{KL}(P_\theta^{(2)}(y_i|x_i) \parallel P_\theta^{(1)}(y_i|x_i))\right] \tag{3}$$

The total loss function is the weighted sum of the cross entropy loss and the KL divergence loss, where α is a hyperparameter that balances the importance of the two parts of the loss. The formula of the total loss function is as follows:

$$L = L_1 + \alpha L_2 \tag{4}$$

We add the R-drop method before the softmax function of the BERT model to limit the output constraints. The experimental results also show that our method is effective.

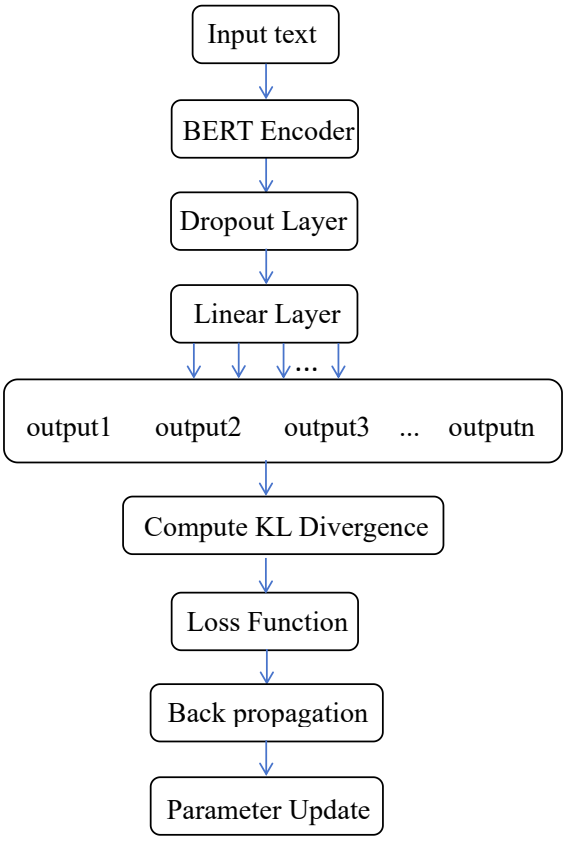

Figure 2: Model overall frame diagram

4. Experiments and Results

4.1. Dataset and Preprocessing

The training dataset provided by PAN@CLEF 2024 consists of 13 machine-generated datasets and one human-generated dataset, each containing 1,087 pieces of data. These datasets include news articles,

Wikipedia introductory texts, and fan fiction. Additionally, PAN@CLEF 2024 offers a guide dataset comprising multiple real and fake news articles that made headlines in the United States in 2021. This guide dataset is presented in a JSONL file format, with each file separated by line breaks. Each file contains a list of articles, all maintaining the same article ID and line order, except for the model-specific prefix before the first file. This ensures that the same line always corresponds to the same topic but from a different "author".

We extracted additional human-generated data from Kaggle to balance the training dataset, creating a set of 1,681 human data points in the same format as the provided training data.

The format of the test data is different from that of the training data set. Each line in the test data contains a pair of texts, and the ID is garbled. Our task is to predict which text in each pair is manually generated. For each test case in the input file, the ID of each text pair and a confidence score are output, which indicates the possibility of human generation versus machine generation. A score of 1 indicates a strong belief that the text is manually generated, while a score of 0 indicates that the text is machine-generated. If the prediction score is less than 0.5, text1 is considered manually generated; if it is greater than 0.5, text2 is manually generated. The basic dataset composition is shown in Table1. In

Table 1
Data sets for different models

Data Name	Data numbers
human.jsonl	1087
search_human.jsonl	1681
alpaca-7b.jsonl	1087
bigscience-bloomz-7b1.jsonl	1087
chavinlo-alpaca-13b.jsonl	1087
gemini-pro.jsonl	1087
gpt-3.5-turbo-0125.jsonl	1087
gpt-4-turbo-preview.jsonl	1087
meta-llama-llama-2-7b-chat-hf.jsonl	1087
meta-llama-llama-2-70b-chat-hf.jsonl	1087
mistralai-mistral-7b-instruct-v0.2.jsonl	1087
mistralai-mixtral-8x7b-instruct-v0.1.jsonl	1087
qwen-qwen1.5-72b-chat-8bit.jsonl	1087
text-bison-002.jsonl	1087
vicgalle-gpt2-open-instruct-v1.jsonl	1087

deep learning, preprocessing is important, and more text data means more text features a model can learn. The original training dataset has 11 files with 1,087 entries each. Unlike the training set, the test set lacks labels for human-machine-generated text. We processed each entry to match the test dataset format.

The ratio of the human-generated dataset to the machine-generated dataset is 1:13, leading to the problem of the unbalanced dataset. Every line in all files corresponds to the same topic, we combine one artificial text and one machine-generated text for the same topic, one ID of the article corresponds to two text entries, and the label distinguishes which text is manually generated.

Hope that these data can better capture text features and avoid creating imbalanced training datasets, which would result in poor generalization for minority class samples. By processing the data this way, we can enhance the model's ability to learn the characteristics of minority samples more effectively.

Finally, the dataset is preprocessed, and the collected artificial data is randomly paired with machine-generated data to form text pairs, thereby expanding the dataset for model training. Finally, 14,135 training data are obtained, and the positive and negative sample ratio (text1 and text2 are manually set) is maintained at about 1:1. The dataset details are shown in Table2. The training, validation, and test sets are randomly divided into 6:3:1.

Table 2
Preprocessed data set partitioning

Dataset	Text1	Text2	sum
train	7135	7000	14135

4.2. Evaluation Metrics

To evaluate the proposed models, we used the evaluation tool officially proposed by tira[9] to evaluate our model. The evaluation indicators are as follows:

- ROC-AUC: The area under the ROC (Receiver Operating Characteristic) curve.
- Brier: The complement of the Brier score (mean squared loss).
- C@1: A modified accuracy score that assigns non-answers (score = 0.5) the average accuracy of the remaining cases.
- F1: The harmonic mean of precision and recall.
- F0.5: A modified F0.5 measure (precision-weighted F measure) that treats non-answers (score = 0.5) as false negatives.
- Mean: The arithmetic mean of all the metrics above.

4.3. Experiment setting

In this paper, the bert-base model is selected, and the parameters are as follows: L=12: there are 12 layers of transformer encoder. H=768 indicates that the number of hidden units in each layer is 768. A=12 indicates that each layer has 12 self-attention heads. Total parameters=110M: indicates that the model has about 110 million parameters.

The BERT-base model is selected as the pre-trained model, and the BERT and fully connected network classification models are constructed using PyTorch. Our hyperparameters are set as follows: the batch size is 32, the maximum sequence length is 512, the initial learning rate is set to 2e-5, and the model is trained for 10 epochs. Optimization is performed using AdamW, and the best model weights are saved based on the accuracy of the test set at each epoch.

After setting the above hyperparameters, we also trained a single text. Specifically, we added a label "label" to the official dataset, using "0" or "1" to indicate whether the text was artificially generated. Then we used this method to train the model. However, the verification effect was not ideal.

Late-bit is a training method without adding R-Drop.Bitter-metaphor is a method where we apply the R-Drop regularization method on top of the previous approach.Nutty-combat integrates four training models based on text training. These four trained models predict our test data, and we select the most frequent prediction results as the final value.

4.4. Results

We reassembled the training dataset proposed by PAN24 and evaluated the model's performance on this reassembled dataset. Additionally, we tested our model on the PAN24 generative AI author authentication test dataset. The test results are presented in Table3.

By comparing the late-bit and nutty-combat methods, table3 shows that the model fusion method is not as good as the single model in all evaluation indicators. This may be due to using a 0.5 threshold for predictions, which led to performance degradation. Specifically, when the predicted results are the same, a score of 0.5 is assigned, resulting in diminished performance on the test data.

When comparing bitter-metaphor and late-bit, it is evident that bitter-metaphor achieves better test results across all metrics. This demonstrates the effectiveness of the R-Drop method for the AI author verification task.

It can be seen from Tables 3 and 4 that the scores of our model for most data sets are higher than the baseline. The Median is greater than 0.936, which generally exceeds the baseline median. Through the

Table 3
Overview of the mean accuracy over 9 variants of the test set. We report the minumum, median, the maximum, the 25-th, and the 75-th quantile, of the mean per the 9 datasets.

Approach	Minimum	25-th Quantile	Median	75-th Quantile	Max
bitter-metaphor	0.570	0.857	0.946	0.978	1.000
late-bit	0.310	0.785	0.941	0.980	1.000
nutty-combat	0.328	0.781	0.936	0.975	1.000
baseline Binoculars	0.342	0.818	0.844	0.965	0.996
baseline Fast-DetectGPT (Mistral)	0.095	0.793	0.842	0.931	0.958
baseline PPMd	0.270	0.546	0.750	0.770	0.863
baseline Unmasking	0.250	0.662	0.696	0.697	0.762
baseline Fast-DetectGPT	0.159	0.579	0.704	0.719	0.982
Overall 25-th quantile	0.353	0.496	0.658	0.675	0.711
Overall Median	0.605	0.645	0.875	0.889	0.936
Overall 75-th quantile	0.758	0.865	0.933	0.959	0.991
Overall 95-th quantile	0.863	0.971	0.978	0.990	1.000
Overall Min	0.015	0.038	0.231	0.244	0.252

result analysis, the results for the data sets "gpt-4-turbo-preview-german" and "text-bison-002-german" are unsatisfactory. The scores of other data sets are close to 1. This may be due to the different language structures leading to model judgment errors.

Table 4
Detailed description of all dataset variants

Model	ROC AUC	Brier	$c@1$	F1	F0.5U	Mean
bitter-metaphor	-	-	-	-	-	1.0
gemini-pro-high-temperature	0.997	0.996	0.996	0.997	0.999	0.997
gpt-3.5-turbo-0125	1.0	1.0	1.0	1.0	1.0	1.0
gpt-4-turbo-preview	1.0	1.0	1.0	1.0	1.0	1.0
gpt-4-turbo-preview-german	0.557	0.612	0.553	0.52	0.523	0.553
meta-llama-llama-2-70b-chat-hf	1.0	1.0	1.0	1.0	1.0	1.0
meta-llama-llama-2-7b-chat-hf	1.0	1.0	1.0	1.0	1.0	1.0
mistralai-mistral-7b-instruct-v0.2	1.0	0.998	0.996	0.996	0.994	0.997
mistralai-mixtral-8x7b-instruct-v0.1	1.0	0.998	0.996	0.996	0.994	0.997
qwen-qwen1.5-72b-chat-8bit	1.0	1.0	1.0	1.0	1.0	1.0

5. Conclusion

In this article, we describe our approach to author authentication for PAN 2024. We reorganize the training dataset to match the format of the test dataset, thereby creating a new, larger training dataset. By augmenting the data in this way, we aim to improve our model's ability to distinguish between text generated by human authors and text generated by machines. Additionally, we introduce a regularization method to learn the intrinsic characteristics of the data by comparing the similarities and differences between different samples. This approach helps the model learn the unique characteristics of different authors' writing styles.

We also hope to explore various model fusion techniques to combine models with strong predictive performance with good generalization ability. Of course, there are other methods to improve generalization ability that are also worth studying, and we hope to train better models to identify authors in the future.

Acknowledgments

This work is supported by the Natural Science Foundation of Guangdong Province, China (No.2022A1515011544)

References

[1] J. Bevendorff, M. Wiegmann, J. Karlgren, et al., Overview of the "Voight-Kampff" Generative AI Authorship Verification Task at PAN and ELOQUENT 2024, in: G. Faggioli, N. Ferro, P. Galuščáková, A. G. S. de Herrera (Eds.), Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum, CEUR Workshop Proceedings, CEUR-WS.org, 2024.

[2] J. Bevendorff, X. B. Casals, B. a. a. Chulvi, Overview of the Voight-Kampff Generative AI Authorship Verification Task at PAN 2024, in: Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum, CEUR-WS.org, 2024.

[3] M. Fröbe, M. Wiegmann, N. Kolyada, et al., 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. doi:10.1007/978-3-031-28241-6_20.

[4] L. Wu, J. Li, Y. Wang, Q. Meng, T. Qin, W. Chen, M. Zhang, T.-Y. Liu, et al., R-drop: Regularized dropout for neural networks, Advances in Neural Information Processing Systems 34 (2021) 10890–10905.

[5] O. Nikula, Linguistic feature analysis of real and fake news: Human-written vs. grover-written (2023).

[6] S. Gehrmann, H. Strobelt, A. M. Rush, Gltr: Statistical detection and visualization of generated text, arXiv preprint arXiv:1906.04043 (2019).

[7] E. Mitchell, Y. Lee, A. Khazatsky, C. D. Manning, C. Finn, Detectgpt: Zero-shot machine-generated text detection using probability curvature, in: International Conference on Machine Learning, PMLR, 2023, pp. 24950–24962.

[8] G. Bao, Y. Zhao, Z. Teng, L. Yang, Y. Zhang, Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature, arXiv preprint arXiv:2310.05130 (2023).

[9] 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. doi:10.1007/978-3-031-28241-6_20.