Voight-Kampff Generative AI Authorship Verification Based on T5

Notebook for the PAN Lab at CLEF 2024

Zhaojian Lin*, Fanzhi Zeng, Yan Zhou, Xiangyu Liu and Yuexia Zhou

Foshan University, Foshan, China

Abstract

This paper proposes a method for fine-tuning the pre-trained language model Text-to-Text Transfer Transformer (T5) for Generative AI Authorship Verification. During the training phase, the input sequence consists of explicit instructions and training samples, while the output sequence represents the classification results in the form of "positive </s>" or "negative </s>". During inference, the model's vocabulary is restricted to "positive" and "negative", selecting the word with the highest probability as the classification result. Finally, on the test set, our performance metrics scored 0.138, 0.529, 0.744, 0.874, and 0.877 for the minimum, 25th percentile, median, 75th percentile, and maximum values, respectively.

Keywords

Generative AI Authorship Verification, Pre-trained Language Model, Classification

1. Introduction

Text classification is a fundamental research direction in NLP tasks. The aim of this direction is to determine whether two texts are written by the same person. AI Authorship Verification can be widely applied in environments where the authenticity of information needs to be verified, such as in legal proceedings and news reporting.

In the PAN 2024 AI Authorship Verification Task [1, 2], our challenge is to differentiate between human-authored texts and machine-generated texts from two texts with the same topic. In this paper, we focus on using fine-tuning methods to address this problem. We first conducted thorough data preprocessing on the training set provided by the organizers, including data cleaning and character conversion steps, to ensure the quality and consistency of the data. Then, we fine-tuned a T5 model [3, 4] on the preprocessed dataset to perform the text classification task. Finally, we submitted our results on TIRA.io [5] to evaluate the performance of our method in real-world applications.

2. Related Work

Text detection in machine-generated text is an active research area, primarily employing three distinct approaches to differentiate between human-written text and machine-generated text.

The first approach is traditional statistical methods, which identify anomalies by analyzing statistical characteristics of text samples. For example, the statistical method called the Giant Language Model Test Room (GLTR), designed by Gehrmann et al. [6] This method comprises three testing steps: Tests 1 and 2 examine whether generated words are sampled from the top of the distribution, while Test 3 verifies if the system is overly confident in its next prediction due to familiarity with previously generated contexts. Through a study involving human subjects, GLTR successfully increased the accuracy of identifying fake text from 54% to 72% without any pretraining, significantly enhancing human discernment of the genuineness of generated text.

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

2112203036@stu.fosu.edu.cn (Z. Lin*); coolhead@126.com (F. Zeng); zhouyan791266@fosu.edu.cn (Y. Zhou); xyliu1805@fosu.edu.cn (X. Liu); fs_zyx@fosu.edu.cn (Y. Zhou)

0009-0004-0426-748X (Z. Lin*); 0009-0006-4760-6837 (X. Liu)

The second approach is unsupervised learning methods, particularly zero-shot classification. This method utilizes pre-trained large language models (LLMs) to detect their own generated text or text generated by similar models. Solaiman et al. [7] proposed a baseline method that makes classification decisions by evaluating log-probabilities and corresponding thresholds. However, compared to statistical methods, the performance of zero-shot classification methods is typically inferior.

The third approach is supervised learning methods, which involve fine-tuning existing language models to create a text detector. For instance, Zeller et al. [8] utilized fine-tuning linear layers on the hidden states of the GROVER encoder to distinguish whether input text originates from the GROVER model or from human hands. However, compared to the first two methods, supervised learning methods require a significant amount of labeled data for model training, making the training process more time-consuming.

3. System Overview

3.1. Data Source

The training data set of the Generative AI Authorship Verification task is a bootstrap dataset of real and fake news articles spanning multiple 2021 U.S. news headlines. It consists of JSON files written by 13 different machine authors and 1 human author. Each file contains a list of articles, and each file contains articles of the same topic. In all files, the ID and row order of the article are the same, so the same row always corresponds to the same topic, but from different “authors”. Each document contains 24 topics and 1087 articles. Considering the token length limitations of large language models, we have conducted a token count on these data. The total number of articles and the token length range in the dataset written by different authors are shown in Table 1.

Table 1
Token distribution by different authors

Author	Token Range	Quantity
Human	[25,7989]	1087
alpaca-7b	[0,3141]	1087
bigscience-bloomz-7b1	[96,3557]	1087
chavinlo-alpaca-13b	[0,5505]	1087
gemini-pro	[1205,5881]	1087
gpt-3.5-turbo-0125	[75,5961]	1087
gpt-4-turbo-preview	[1189,6931]	1087
meta-llama-llama-2-70b-chat-hf	[1209,5957]	1087
meta-llama-llama-2-7b-chat-hf	[367,5865]	1087
mistralai-mistral-7b-instruct-v0.2	[1446,6274]	1087
mistralai-mixtral-8x7b-instruct-v0.1	[811,6928]	1087
qwen-qwen1.5-72b-chat-8bit	[1404,3917]	1087
text-bison-002	[0,5613]	1087
vicgalle-gpt2-open-instruct-v1	[52,3653]	1087

3.2. Dataset Preprocessing

For the provided training set, we initially preprocess texts authored by machines and humans. We remove all empty texts, replace all full-width characters with half-width characters, and remove spaces within the texts. Subsequently, we merge texts authored by machines and humans. Specifically, we create a text tuple ("pair": ["text", "label"]), where "text" represents the content of the article, and "label" indicates whether it is authored by a human (1 for positive, 0 for negative). We use 80% of the training dataset for training and 20% for validation, with 11,393 samples for training and 3,797 samples for validation.

3.3. Method

Our method fine-tunes the T5 model for the text classification task. This approach transforms the text classification problem into a sequence-to-sequence (seq-to-seq) problem, enabling the model to handle and understand text data more flexibly.

During the training phase, we utilized the training dataset provided by PAN 2024 and conducted necessary preprocessing to ensure the data was suitable for model training. Since the T5 model was exposed to numerous tasks with explicit instructions or prompts during its pre-training phase, adding a prompt can fully leverage the model's pre-training knowledge, thereby improving its performance on specific tasks. Consequently, the input sequence consists of two parts: one is explicit instructions (for example: "Distinguish whether the following text is written by a human"), and the other is the text sample to be classified. The output sequence represents the classification result in the form of "positive </s>" or "negative </s>", where "</s>" is a sequence-ending token. To adapt the model to this text classification task, we replaced the head of the T5 model with a randomly initialized head.

During the inference phase, we restricted the model's vocabulary to only include the words "positive" and "negative". The model predicts whether the input text is human-written or machine-generated based on the probability distribution of these two words. For two disputed texts in the test dataset, the model first predicts each text individually, outputting either "positive" or "negative". Subsequently, we compare the predicted labels of these two texts. If the predictions differ, a clear conclusion can be drawn. If the predictions are the same, we compare the probability values of the predictions and choose the result with the higher confidence. Additionally, we introduced a special <PAD> token in the T5 model's decoder to help maintain consistent output formats when handling input sequences of different lengths. The detailed design of the entire network architecture is illustrated in Figure 1.

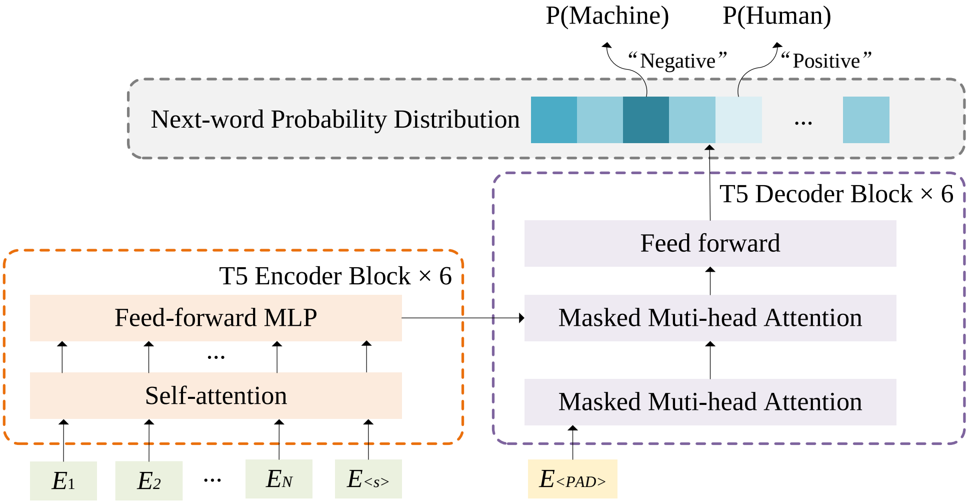

Figure 1: Architecture for T5

4. Experiments and Results

4.1. Experiment setup

In this work, we adopt T5 as the pretrained model, and we use Pytorch to implement T5. Our hyperparameters are set as follows: the batch size is 512, the loss function is cross entropy, the initial learning rate is set to 5e-4, and 5 epochs are trained. Each training is optimized with AdamW. Our experiment was conducted on the A800 server.

4.2. Evaluation

To assess the effectiveness of our proposed model, we utilized the evaluation tool provided by PAN, which includes the following metrics:

ROC-AUC: ROC-AUC is a comprehensive evaluation of the balance between the true positive rate and the false positive rate.

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: F1 score is the harmonic mean of precision and recall, combining both metrics into a single value.

F0.5u: 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. Results

We evaluated the performance of our model and baselines(Binoculars [9], Fast-DetectGPT (Mistral) [10], PPMd [11, 12], Unmasking [13, 14] and Fast-DetectGPT [10]) on the official test set provided in PAN 2024.

Table 2 shows the performance of our method across various metrics. Our method surpasses Unmasking and Fast-DetectGPT in ROC-AUC, C@1, F0.5u and mean respectively, but there is still a certain gap with Binoculars.

Table 3 further shows the average accuracy of our model on different dataset variants, particularly on the test sets of 9 variants. Our method surpasses Fast-DetectGPT (Mistral) on the minimum, surpasses Unmasking and Fast-DetectGPT on the median, surpasses PPMd, Unmasking, and Fast-DetectGPT on the 75-th quantile, and surpasses PPMd and Unmasking on the max.

Compared with the quantile results of other participants, our model is close to or exceeds the 25-th quantile model on most indicators, and exceeds the Min model on all indicators, indicating that our method performed poorly on the test set and still has a significant gap compared to the current state-of-the-art methods.

Table 2
Overview of the accuracy in detecting if a text is written by an human in task 4 on PAN 2024 (Voight-Kampff Generative AI Authorship Verification)

Approach	**ROC-AUC**	**Brier**	**C@1**	**F_1**	**$F_{0.5u}$**	**Mean**
Ours	0.739	0.739	0.739	0.686	0.782	0.737
Baseline Binoculars	0.972	0.957	0.966	0.964	0.965	0.965
Baseline Fast-DetectGPT (Mistral)	0.876	0.8	0.886	0.883	0.883	0.866
Baseline PPMd	0.795	0.798	0.754	0.753	0.749	0.77
Baseline Unmasking	0.697	0.774	0.691	0.658	0.666	0.697
Baseline Fast-DetectGPT	0.668	0.776	0.695	0.69	0.691	0.704
95-th quantile	0.994	0.987	0.989	0.989	0.989	0.990
75-th quantile	0.969	0.925	0.950	0.933	0.939	0.941
Median	0.909	0.890	0.887	0.871	0.867	0.889
25-th quantile	0.701	0.768	0.683	0.657	0.670	0.689
Min	0.131	0.265	0.005	0.006	0.007	0.224

Table 3
Overview of the mean accuracy over 9 variants of the test set.

Approach	Minimum	25-th Quantile	Median	75-th Quantile	Max
Ours	0.138	0.529	0.744	0.874	0.877
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
95-th quantile	0.863	0.971	0.978	0.990	1.000
75-th quantile	0.758	0.865	0.933	0.959	0.991
Median	0.605	0.645	0.875	0.889	0.936
25-th quantile	0.353	0.496	0.658	0.675	0.711
Min	0.015	0.038	0.231	0.244	0.252

5. Conclusion

The article comprehensively elaborates on our research progress in the field of Voight-Kampff Generative AI Authorship Verification in 2024. In this study, we fine-tuned a T5 pre-trained model to enhance the detection capability of AI text generation. The experimental results demonstrate that this method effectively enhances text detection capability, but there still exists a certain gap compared to the state-of-the-art methods. In the future, we plan to continue optimizing and refining this method to achieve higher levels of precision and efficiency. Additionally, we will explore the potential applications of this method in a wider range of natural language processing tasks, aiming to expand its scope of application and further increase its practical value.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 61972091), Natural Science Foundation of Guangdong Province of China (No. 2022A1515010101, No. 2021A1515012639).

References

[1] A. A. Ayele, N. Babakov, J. Bevendorff, X. B. Casals, B. Chulvi, D. Dementieva, A. Elnagar, D. Freitag, M. Fröbe, D. Korenčić, 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, in: L. Goeuriot, P. Mulhem, G. Quénot, D. Schwab, L. Soulier, G. M. D. Nunzio, P. Galuščáková, A. G. S. de Herrera, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Fifteenth International Conference of the CLEF Association (CLEF 2024), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2024.

[2] J. Bevendorff, M. Wiegmann, E. Stamatatos, M. Potthast, B. Stein, Overview of the Voight-Kampff Generative AI Authorship Verification Task at PAN 2024, in: G. F. N. Ferro, P. Galuščáková, A. G. S. de Herrera (Eds.), Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum, CEUR-WS.org, 2024.

[3] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring

the limits of transfer learning with a unified text-to-text transformer, Journal of machine learning research 21 (2020) 1–67.

[4] Y. Chen, H. Kang, V. Zhai, L. Li, R. Singh, B. Raj, Gpt-sentinel: Distinguishing human and chatgpt generated content, arXiv preprint arXiv:2305.07969 (2023).

[5] 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.

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

[7] I. Solaiman, M. Brundage, J. Clark, A. Askell, A. Herbert-Voss, J. Wu, A. Radford, G. Krueger, J. W. Kim, S. Kreps, et al., Release strategies and the social impacts of language models, arXiv preprint arXiv:1908.09203 (2019).

[8] R. Zellers, A. Holtzman, H. Rashkin, Y. Bisk, A. Farhadi, F. Roesner, Y. Choi, Defending against neural fake news, Advances in neural information processing systems 32 (2019).

[9] A. Hans, A. Schwarzschild, V. Cherepanova, H. Kazemi, A. Saha, M. Goldblum, J. Geiping, T. Goldstein, Spotting llms with binoculars: Zero-shot detection of machine-generated text, arXiv preprint arXiv:2401.12070 (2024).

[10] 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).

[11] O. Halvani, C. Winter, L. Graner, On the usefulness of compression models for authorship verification, in: Proceedings of the 12th international conference on availability, reliability and security, 2017, pp. 1–10.

[12] D. Sculley, C. E. Brodley, Compression and machine learning: A new perspective on feature space vectors, in: Data Compression Conference (DCC'06), IEEE, 2006, pp. 332–341.

[13] M. Koppel, J. Schler, Authorship verification as a one-class classification problem, in: Proceedings of the twenty-first international conference on Machine learning, 2004, p. 62.

[14] J. Bevendorff, B. Stein, M. Hagen, M. Potthast, Generalizing unmasking for short texts, in: 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), 2019, pp. 654–659.