BLGAV：Generative AI Author Verification Model Based on BERT and BiLSTM

Notebook for PAN at CLEF 2024

Linjiu Guo[1], Wenyin Yang[2,*] , Li Ma[3] , Jinli Ruan[4]

[1]*Foshan University, Foshan, China*
[2]*Foshan University, Foshan, China*
[3]*Foshan University, Foshan, China*
[4]*Foshan University, Foshan, China*

Abstract

In recent years, large language models like GPT-3, BERT, and GPT-4 have made significant advancements in the field of natural language processing, enhancing tasks such as document summarization, language translation, and question answering. Despite these benefits, the authenticity and credibility of texts generated by these models have raised societal concerns, including misinformation and plagiarism. To address these issues, the PAN organization has initiated a series of tasks to differentiate between machine-generated and human-written texts. This paper proposes a Generative AI Authorship Verification model based on BERT and BiLSTM, which enhances text discrimination capabilities by combining Transformer encoders with multi-text feature techniques. The model leverages a pretrained BERT for deep feature extraction and incorporates additional text features calculated by the spaCy , further processed by BiLSTM and Transformer encoders for classification. Experimental results show that the model achieved a mean score of 0.971 on the PAN validation dataset, surpassing all baseline models. This approach not only improves detection accuracy but also enhances adaptability to various text types, making it significant for maintaining the authenticity and reliability of information in the era of automatic content generation.

Keywords

PAN 2024, Generative AI Authorship Verification, Pre-training BERT, SpaCy, Multi-Text Features

1. Introduction

In recent years, large language models such as GPT-3, BERT, ChatGPT, Llama2, PaLM2, and GPT-4 have demonstrated exceptional performance in the field of natural language processing. They are widely utilized in tasks such as document summarization, language translation, and question answering [1,2,3,4]. These models not only facilitate automated content creation and dialogue systems but also enhance efficiency across various industries including customer service, education, law, and healthcare, through intelligent solutions [5,6]. However, with the widespread adoption of these technologies, issues related to the authenticity and credibility of texts generated by these models have increasingly attracted public attention. Key concerns include the spread of misinformation, generation of nonsensical or misleading content, and plagiarism of intellectual property and original content, which are considered significant societal issues .

In this context, the PAN[7] organization has launched a series of tasks to differentiate between machine-generated and human-written texts. This initiative not only aids in identifying and verifying the authenticity of texts but also effectively curbs the spread of misleading information and copyright infringement, thereby protecting the rights of information recipients. Generative

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

1837247142@qq.com(L .Guo); cswyyang@163.com (W. Yang)(*corresponding author); molly_917@fosu.edu.cn (L. Ma) ; 13104922330@163.com (J. Ruan)

0009-0008-7868-8972 (L. Guo); 0000-0003-4842-9060 (W. Yang); 0000-0002-5013-052X (L. Ma) ; 0009-0007-0411-8312 (J. Ruan)

AI Author Verification is typically viewed as a binary classification problem, which involves distinguishing whether a text is written by a human or generated by a machine. Some approaches, based on statistical features, classify texts by comparing the statistical characteristics of texts written by humans and those generated by machines, such as word frequency, syntactic features, and semantic similarity [8]. For instance, Wang et al. proposed a detection method based on word frequency and n-gram features [9]. While initially effective, its performance significantly decreased when faced with more complex generation models. Gehrmann et al. introduced manually designed statistical features [10], which have also shown some effectiveness in assisting humans in detecting machine-generated texts. Another approach involves the use of fine-tuned pretrained language models. These methods fine-tune large-scale pretrained models, such as BERT and GPT-3, on extensive text data, enabling them to better capture subtle differences in texts . Methods based on pretrained language models typically exhibit higher detection accuracy and generalizability, and can adapt to different types of generative models and texts. However, these methods also face challenges, such as the need for substantial computational resources and data for training, and high complexity and computational costs associated with the models.

This paper proposes a BERT and Bidirectional Long Short-Term Memory Network (BiLSTM) based Generative AI Authorship Verification model (BLGAV), which enhances the ability to discriminate between machine-generated and human-written texts by combining Transformer encoders with multi-feature fusion techniques. The model initially uses a pretrained BERT to extract deep textual features, and integrates additional text features computed by the spaCy model, such as lexical diversity and average sentence length, to enhance its discriminative ability. It then processes these features further using BiLSTM[11] and Transformer encoders, and finally, classification is performed through a fully connected layer:

- Multi-text feature fusion method:This model not only relies on deep language feature extraction by the BERT model but also enhances its discriminative ability by calculating multiple text features such as lexical diversity and average sentence length. This multi-feature fusion method improves the model's accuracy in recognizing generated text.
- Experimental results show that the model achieved a Mean score of 0.971 on the official validation dataset provided by the PAN laboratory, surpassing all five benchmark models provided by the official sources.

2. Related Work

With the rapid development and application of large language models, detecting machine-generated text and verifying authorship have become significant research topics. Existing work mainly focuses on the following areas.

2.1. Unsupervised Methods Based on Statistical Features

To overcome the overfitting problem in supervised learning models, researchers have begun exploring unsupervised methods based on statistical features. These methods use statistical anomalies in the text to distinguish between machine-generated and human-written texts. For example, Lavergne et al. studied statistical anomalies in entropy[12], while Badaskar et al. used n-gram frequencies as detection features[13]. Gehrmann et al. introduced manually designed statistical features to assist humans in detecting machine-generated texts [10]. Solaiman et al. proposed a simple zero-shot method to detect machine-generated text by evaluating the log probability of each word and using a threshold for segmentation [14]. Mitchell et al. observed that machine-generated texts often lie within the local curvature of log probabilities and introduced DetectGPT. Although this method performs exceptionally well, it requires substantial computational resources [15].

2.2. Methods Using Pretrained Models

In recent years, pretrained models like BERT and RoBERTa have made significant progress in natural language processing tasks and have been applied to the task of detecting machine-generated texts. For example, Solaiman et al. introduced a GPT-2 detector by fine-tuning the RoBERTa model on outputs from GPT-2[16]. Similarly, Guo et al. developed a ChatGPT detector by fine-tuning the RoBERTa model on the HC3 dataset to distinguish between human-written texts and texts generated by ChatGPT[17]. These methods demonstrate the effectiveness of fine-tuning pretrained models for specific tasks but also expose potential overfitting issues when the training data distribution differs from the actual application data distribution[18,19].

3. Methodology

In this paper, we first convert the data provided by PAN into a format suitable for model training through cleaning and formatting to improve data quality. Then, the model uses a pre-trained BERT to extract deep semantic features of the text and integrates additional text features calculated by the spaCy model, such as lexical diversity and average sentence length, which enhances the model's discriminative capability. Subsequently, these features are deeply processed by combining BiLSTM and Transformer encoders to capture complex text structures. Finally, the model classifies through a fully connected layer, effectively distinguishing between human and machine-generated texts.

3.1. Dataset Preprocessing

The dataset provided by PAN consists of two types of files: one written by human authors and the other generated by machines. Generative AI Authorship Verification is typically viewed as a binary classification problem, that is, distinguishing whether the text is generated by humans or machines. We classify the texts and match labels for each text, where human-written texts are marked as 0 and machine-generated texts as 1. The original data format is transformed from {"id": "...", "text": "..."} to {"text": "...", "label": "0 or 1"}. This process can be described as follows:

$$\{(id, text)\} \rightarrow \{(text, label)\} \tag{1}$$

The conversion rules are as follows:

$$label \begin{cases} 0 \ \ if\ author = human \\ 1\ if\ author = machin \end{cases} \tag{2}$$

This indicates that we transform the original data format, which includes text ID, text content, and author type (human or machine), into a format that only includes text content and the corresponding label (0 or 1). Moreover, to enhance model training effectiveness, the following data cleaning steps were carried out:

- Removing irrelevant information: Clearing numbers, punctuation, and other distracting characters.
- Unifying text format: Converting all text to lowercase and removing stop words.
- Improving feature quality: The above cleaning steps help more accurately reflect the language structure and features, facilitating effective feature extraction by the model.

3.2. Network Architecture

Traditional unsupervised methods based on statistical features detect patterns by calculating word frequency, character frequency, word length, and sentence length. Although simple and easy to implement, these methods fail to capture deep semantic information and rely heavily on manually designed features. Their effectiveness is limited, making it difficult to handle the diversity and complexity of texts. Pre-trained models (such as word2vec and GloVe) represent the semantic information of sentences by averaging or summing word vectors. This approach lacks contextual interaction, ignores the order and dependencies between words, and cannot effectively handle polysemy and synonyms. Additionally, it fails to capture deep structural and contextual information within sentences, leading to shortcomings in text detection tasks.

To overcome the limitations of traditional methods, we designed a Generative AI Authorship Verification based on BERT and BiLSTM. As shown in Figure 1,The model first utilizes the pre-trained BERT model to extract deep semantic features from the text. These features capture the complex contextual relationships between words. The BERT model processes the input text sequence using the following formula:

$$H_L = \left[h_L[CLS], h_L^1, \ldots, h_L^T, h_L[SEP]\right] \tag{3}$$

where H_L represents the output feature sequence of the BERT model at the L layer, including the special tokens [CLS] and [SEP]. In this study, we used the contextual embeddings from the last layer of the BERT model. Specifically, we obtained the embeddings for each token (last_hidden_state) from the last layer of BERT. These contextual embeddings were concatenated with features calculated by spaCy. The concatenation was done at the token level, not at the layer or CLS token level. That is, the embeddings for each token from the last layer of BERT were concatenated with the extended spaCy feature representations and then used as input to the LSTM.

After extracting deep semantic features, this study used the en_core_web_sm model from the spaCy library to calculate additional text features. These features include lexical diversity, average sentence length, average word length, number of grammatical errors, sentiment tendency, repetition rate, and stop word ratio. en_core_web_sm is a small English language model provided by spaCy, suitable for various natural language processing tasks such as tokenization, part-of-speech tagging, dependency parsing, and named entity recognition. The formulas for the additional text features are as follows:

$$V_{features} = SpaCy[f_1, f_2, f_3, f_4, f_5, f_6, f_7] \tag{4}$$

where f_1 represents lexical diversity, f_2 represents average sentence length, f_3 represents average word length, f_4 represents the number of grammatical errors, f_5 represents sentiment tendency, $f_6$represents repetition rate, and f_7 represents stop word ratio. These features provide the model with the ability to analyze the text from different perspectives, enhancing its capability to distinguish between human-written and machine-generated texts.

The contextual embeddings from the last layer of BERT (the embeddings for each token) have a dimension of (batch_size, sequence_length, 768). The additional text features extracted by spaCy are 7-dimensional vectors. These 7-dimensional spaCy features are expanded to 21 dimensions through a fully connected layer. Then, the expanded spaCy features are repeated at each time step to match the sequence length of the BERT output, resulting in a dimension of (batch_size, sequence_length, 21). Finally, the contextual embeddings from BERT and the expanded spaCy features are concatenated at the token level, resulting in a concatenated feature dimension of (batch_size, sequence_length, 789). The fused features are then fed into the BiLSTM for processing. The formulas are as follows:

$$F_{fuscd} = H_L \oplus V_{features} \tag{5}$$

$$\overrightarrow{\mathrm{h_t}} = \mathrm{LSTM}\left(\overrightarrow{\mathrm{h_{t-1}}}, h_t^L\right) \tag{6}$$

$$\overleftarrow{h_t} = \mathrm{LSTM}\left(\overleftarrow{\mathrm{h_{t+1}}}, h_t^L\right) \tag{7}$$

Where $\overrightarrow{\mathrm{h_t}}$ represents the hidden state obtained from the forward LSTM, $\overleftarrow{h_t}$ represents the hidden state obtained from the backward LSTM, and h_t^L represents the representation of the fused features at time step t.The features processed by the BiLSTM are then fed into the Transformer encoder, followed by a fully connected layer for binary classification, producing the classification results.

$$H_{out} = TransformerEncoder\left(\overrightarrow{\mathrm{h_t}} \oplus \overleftarrow{h_t}\right) \tag{8}$$

$$p(y_c|H_{out}) = softmax(WH_{out}) \tag{9}$$

where H_{out} represents the optimized feature representation output by the Transformer encoder, $p(y_c|H_{out})$ represents the probability of the output being class c, and W represents the weight matrix of the fully connected layer.

Through this hybrid feature approach, the BLGAV model not only improves recognition accuracy but also enhances its adaptability to different types and styles of text. This makes it a powerful tool for automatically detecting and classifying machine-generated content. This is of great significance in the current environment of information explosion and increasing automatic content generation, helping to maintain the authenticity and reliability of information.

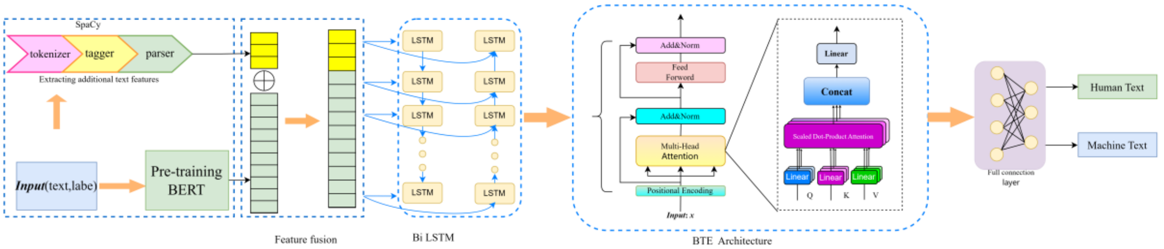

Figure 1: Model Structure Diagram

4. Experiments and Results

4.1. Experimental Setting

For dataset partitioning, the dataset was first preprocessed and then split into training and validation sets in a 7:3 ratio. The study used the pre-trained BERT version bert-base-uncased to extract deep semantic features. The batch size was set to 8, the maximum length of the BERT encoder to 512, the learning rate to 2e-5, and the random seed to 42. The BiLSTM model had a hidden layer dimension of 256 and 2 layers. During the training phase, an RTX 4070 GPU was used for training, with a total of 50 epochs, and the Adam optimizer was employed to update the model weights.

In the testing phase, the format of the test data is {"id": "iixcWBmKWQqLAwVXxXGBGg", "text1": "...", "text2": "..."}. The model predicts each pair of input texts ("text1" and "text2") separately to determine which text is more likely human-written. The process is as follows.

First, both texts are cleaned and formatted separately, and then encoded using the BERT tokenizer. Next, additional text features such as lexical diversity, average sentence length, and average word length are extracted using the spaCy model. The processed texts and extracted features are input into the model, which outputs the probability that each text is "machine-generated." The "human-written" probability for each text is calculated as 1 minus the machine-generated probability. By comparing these probabilities, the text with the higher "human-written" confidence is selected as the human-written text, and this confidence score is output as the result.

4.2. Results

To evaluate the performance of our proposed model, we used the evaluation platform provided by PAN, which includes the following metrics:

- ROC-AUC: Measures the model's ability to distinguish between positive and negative samples, with higher values being better.
- c@1: Assesses the classifier's ability to handle uncertainty while maintaining high accuracy, with higher values being better.
- F0.5u: A variant of the F-score that places more emphasis on precision, suitable for reducing false positives, with higher values being better.
- F1: The harmonic mean of precision and recall, used to evaluate the overall performance of a classification model, with higher values being better.
- Brier: Evaluates the error between predicted probabilities and actual outcomes, with lower values being better.
- Mean: The average of various evaluation metrics, used to comprehensively assess the overall performance of the model.

We uploaded a software named "merciless-lease" to TIRA[20], which evaluates the detection method proposed in this paper. Table 1 shows the comparison results of our method with five baseline methods on various evaluation metrics.

As can be seen from the table, the model proposed in this paper performs excellently on multiple evaluation metrics, especially in ROC-AUC, Brier, F_1, $F_{0.5u}$, and Mean metrics. Notably, the ROC-AUC reaches 0.994, significantly surpassing other models, indicating its outstanding ability to distinguish between positive and negative samples. The Mean metric also demonstrates the high efficiency and reliability of our model, reflecting its excellent overall performance across different evaluation dimensions. However, in the C@1 metric, our model is slightly lower than Binoculars, which may be due to Binoculars being more refined in handling high-probability samples.

These evaluation results reflect the advantages of our model in feature extraction and processing. Our method significantly improves the accuracy and overall performance of detecting machine-generated text by combining the deep semantic features of BERT, additional feature analysis of spaCy, and the feature processing of BiLSTM and Transformer. In contrast, the five baseline models are relatively simple in feature extraction and processing, lacking the capture of complex semantics and contextual relationships, resulting in poor performance when detecting complex texts.

Table 1

Comparison of Different Methods on Various Evaluation Metrics

Approach	ROC-AUC	Brier	C@1	F_1	$F_{0.5u}$	Mean
BLGAV (our)	0.994	0.975	0.963	0.963	0.962	0.971
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

5. Conclusion

This paper proposes a BERT and BiLSTM based Generative AI Authorship Verification model, which significantly enhances the ability to distinguish between machine-generated and human-written texts by combining Transformer encoders and multi-feature fusion techniques. Specifically, the model first uses a pretrained BERT to extract deep features of the text, and integrates additional text features calculated by the spaCy model, such as lexical diversity and average sentence length, to enhance its discriminative ability. Subsequently, these features are further processed using LSTM and Transformer encoders, and finally, classification is performed through a fully connected layer. Experimental results show that the model achieved a Mean score of 0.971 on the official validation dataset provided by the PAN laboratory, surpassing all benchmark models.

In future work, we will further optimize the model by introducing more effective features, compressing long texts, and exploring other methods to improve system performance and detection accuracy. We believe that with continuous improvement, this model will play a greater role in the field of machine-generated text detection.

Acknowledgements

This work was supported by grants from the Guangdong-Foshan Joint Fund Project (No. 2022A1515140096) and Open Fund for Key Laboratory of Food Intelligent Manufacturing in Guangdong Province (No. GPKLIFM-KF-202305).

References

[1] Brown T, Mann B, Ryder N, et al. Language models are few-shot learners[J]. Advances in neural information processing systems, 2020, 33: 1877-1901.

[2] Radford A, Wu J, Child R, et al. Language models are unsupervised multitask learners[J]. OpenAI blog, 2019, 1(8): 9-32.

[3] Touvron H, Martin L, Stone K, et al. Llama 2: Open foundation and fine-tuned chat models[J]. arXiv preprint arXiv:2307.09288, 2023.

[4] Lao Q, Ma L, Yang W, et al. Style Change Detection Based On Bert And Conv1d[C]//CLEF (Working Notes). 2022: 2554-2559.

[5] Kolasani S. Optimizing natural language processing, large language models (LLMs) for efficient customer service, and hyper-personalization to enable sustainable growth and revenue[J]. Transactions on Latest Trends in Artificial Intelligence, 2023, 4(4).

[6] Baidoo-Anu D, Ansah L O. Education in the era of generative artificial intelligence (AI): Understanding the potential benefits of ChatGPT in promoting teaching and learning[J]. Journal of AI, 2023, 7(1): 52-62.

[7] 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.

[8] Mutlu B, Sezer E A. Enhanced sentence representation for extractive text summarization: Investigating the syntactic and semantic features and their contribution to sentence scoring[J]. Expert Systems with Applications, 2023, 227: 120302.

[9] Wang T, Chen L C, Genc Y. A dictionary-based method for detecting machine-generated domains[J]. Information Security Journal: A Global Perspective, 2021, 30(4): 205-218.

[10] Gehrmann, S., Strobelt, H., & Rush, A. M. (2019). GLTR: Statistical Detection and Visualization of Generated Text. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1112-1123.

[11] Yang Z, Ma L, Yang W, et al. A Intelligent Detection Method for Irony and Stereotype Based on Hybird Neural Networks[C]//CLEF (Working Notes). 2022: 2708-2713.

[12] Lavergne T, Urvoy T, Yvon F. Detecting Fake Content with Relative Entropy Scoring[J]. Pan, 2008, 8(27-31): 4.

[13] Badaskar S, Agarwal S, Arora S. Identifying real or fake articles: Towards better language modeling[C]//Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II. 2008.

[14] Solaiman I, Brundage M, Clark J, et al. Release strategies and the social impacts of language models[J]. arXiv preprint arXiv:1908.09203, 2019.

[15] Mitchell E, Lee Y, Khazatsky A, et al. Detectgpt: Zero-shot machine-generated text detection using probability curvature[C]//International Conference on Machine Learning. PMLR, 2023: 24950-24962.

[16] Solaiman I, Clark J, Brundage M. GPT-2: 1.5 B Release[J]. OpenAI. Available online at https://openai. com/blog/gpt-2-1-5b-release/, checked on, 2019, 11(13): 2019.

[17] Guo B, Zhang X, Wang Z, et al. How close is chatgpt to human experts? comparison corpus, evaluation, and detection[J]. arXiv preprint arXiv:2301.07597, 2023.

[18] Bakhtin A, Gross S, Ott M, et al. Real or fake? learning to discriminate machine from human generated text[J]. arXiv preprint arXiv:1906.03351, 2019.

[19] Uchendu A, Le T, Shu K, et al. Authorship attribution for neural text generation[C]//Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2020: 8384-8395.

[20] 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.