RoBERTa and Bi-LSTM for Human vs AI Generated Text Detection

Notebook for PAN at CLEF 2024

Panagiotis Petropoulos[1,†], Vasilis Petropoulos[2,†]

Abstract

We are living in a new era of rapidly evolving AI, where new and increasingly powerful versions of existing models, or even new models, are constantly being produced. Humanity and industry are increasingly investing in these advancements. There are many real-life instances where these models are used by humans either to facilitate themselves or to deceive. Students and scholars who are ceasing to delve deep into knowledge and the production of fake news are two of main occurrences that happen frequently. Hence, there is a need to create a classifier capable of detecting and distinguishing AI-generated text from human-authored text. Several and very good approaches have been done, but they must continue to evolve as LLMs evolve. In this year shared task of PAN at CLEF [1] [2]sheds light on the aforementioned need. In this work an architecture of a combination of RoBERTa[3] and Bi-LSTM on top is proposed in order to solve the task.

Keywords

RoBERTa, Bi-LSTM, NLP, AI generated Texts Detection, Authorship Analysis

1. Introduction

In the new era of LLMs there are several approaches that resolve the issue regarding AI generated text detection. Some approaches use LLMs to distinguish AI generated text from human's texts. Recent approach of DetectGPT [4] uses a pretrained T5 encoder-decoder to produce alternations or variations (perturbations) of a given input and then compares the log probability of the produced samples with the original samples, in order to determine if the original text is AI generated or not. Another prior work using RoBERTa transformer model, approaches the problem by modeling the task as a partial Positive-Unlabeled (PU) [5] problem and formulating a Multiscale Positive Unlabeled (MPU) training framework, in order to overcome the issue of wrong predictions when input is a short text [6]. In addition, other methodologies propose a zero-shot setting for the Human vs AI text detection, by computing log perplexity using an "observer" LLM and cross-perplexity using a "performer" LLM for a given text [7]. The performer tries to predict the next token of a sequence and the observer tries to evaluate the prediction. The ratio of these two metrics, called Binoculars score. Fast-DetectGPT [8] is also an efficient method for detecting machine generated texts. Based on this approach the proposed model calculates conditional probability curvature between text passages and language models in a zero-shot setting.

CLEF 2024: Conference and Labs of the Evaluation Forum, September 9-12, 2024, Grenoble, France

† These authors contributed equally.

panos.petr1@gmail.com (P.Petropoulos); petropoulos.95@gmail.com (V. Petropoulos);

2. Experimental setup

To solve the task, a RoBERTa [3] base version model as a backbone is used. As of the previous year of AV task [9] a model architecture with stacked BERT like transformer and Bi-LSTM on top [10] made the solution of the task feasible. On top of RoBERTa model, a Bi-LSTM is used to capture information from both directions of the RoBERTa output embeddings. Only the last 4 encode Layers of RoBERTa model were unfreezed during 3-epoch training on GPU with 12 GB vRAM. The input Embeddings Sequence for Bi-LSTM are calculated from the sum of the last 4 encode layers output of RoBERTa. After Bi-LSTM a dropout Layer with 0.3 probability of an element (Neural Network unit) to be zeroed is used, followed by a Fully Connected Layer, as classification head, to classify the input text into Human or machine generated. The Dropout and Fully Connected Layers take as input the concatenation of both directions from last hidden states of Bi-LSTM output. The Learning rate was 5e-5 with AdamW optimizer, and the loss function was categorical crossentropy. The batch size was 32. The architecture of the proposed model is illustrated in **Figure 1.**

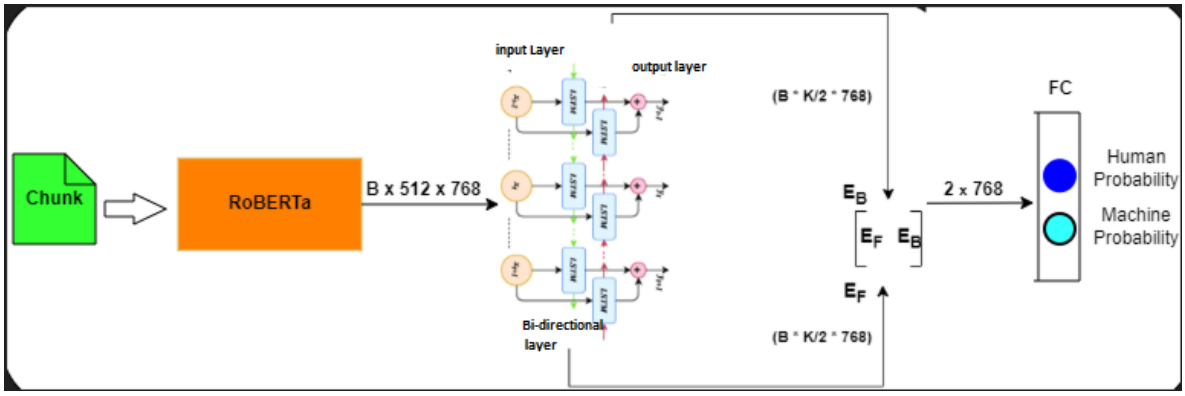

Figure 1: Architecture of the proposed Model, where k is the sequence length and B the Batch size.

2.1. Data preparation and processing

The Voight-Kampff Generative AI Authorship Verification 2024 PAN Challenge [1] provided a train dataset that consists of multiple jsonl files, that requested via Zenodo [2]. One file that consists of news articles written by humans and 13 jsonl files with texts generated by a known LLMs like GPT-4 , gemini pro, Llama or Mistral. Both Human and machine generated texts in train dataset contain information about the same topic. In order to have a ready-to-use dataset and train the proposed model 3 basic steps are followed:

1. Parse texts from humans and machines.
2. Shuffle them, keeping the original sample id.
3. Perform a basic processing procedure, that includes only the replacement of digits (numbers) with constant value of '1', keeping the original format.
4. Chunking due to the limitation of the RoBERTa base Model in input sequence length (max seq. length 512 tokens).

RoBERTa base model has a limitation on the length of input tokens. For that reason, a chunking procedure is applied with max sequence length of 512 (including Special Tokens). Between chunks of the same text an overlapping approach was applied to avoid extremely long padding sequences.

[2] https://zenodo.org/records/10718757

2.1.1. Evaluation

For the experiments 2 basic approaches are followed, following a random shuffling between human and machine generated texts. Consider the LLMs to be treated as "Authors".

- open-set setup: Validation and Test datasets do not contain the same text and LLMs as Train dataset. Also, the same approach between Validation set and Test set is followed.
- close-set setup: Train, Validation and Test datasets contain the same machines but different texts.

The Train-validation splitting ratio was 70% of chunks for train, 20% for Validation and 10% for hold-out test set. To determine if a text is AI generated or not, the proposed approach of this work uses all the chunks from input text of Test set by averaging the posterior probabilities from each chunk. The decision is calculated by the max average probability of binary output. As for the evaluation metrics, we use the metrics provided in [12]

- AUC: the conventional area-under-the-curve of the precision-recall curve
- F1-score: the harmonic mean of the precision and recall [13]
- c@1: a variant of the conventional F1-score, which rewards systems that leave difficult problems unanswered (i.e. scores of exactly 0.5) [14]
- overall: the simple average of all previous metrics

3. Results

Table 1 below shows Accuracy, F1-score AUC, and C@1 scores after predictions on dataset with name pan24-generative-authorship-smoke-test available for testing on Tira[3] [11]. **Table 2** shows us the scores after running the prediction and evaluation procedures on custom unseen test set, for both evaluation methods (open-set and close-set setup). In this table of results the std for Accuracy and F1-score is also reported. To calculate the std of those metrics 3 evaluation runs are performed.

Table 1: Metrics for run on provided dataset on Tira (pan24-generative-authorship-smoke-test).

Method	AUC	C@1	F_05_u	F1-Score	brier	Overall
RoBERTa + Bi-LSTM	0.902	0.925	0.874	0.918	0.909	0.906

On **Table 1** We can see that the proposed model is performed very well on the dataset provided on Tira and makes the task feasible.

Table 2
Metrics and std on custom unseen test set for 2 evaluation methods (open-set and close-set).

Method	Accuracy	F1-score	AUC	C@1
RoBERTa + Bi-LSTM (open-set setup)	0.973(+/- 0.02)	0.956(+/- 0.02)	0.981	0.942
RoBERTa + Bi-LSTM (close-set setup)	**0.996(+/- 0.02)**	**0.974(+/- 0.01)**	**0.993**	**0.953**

From the results it can been seen that for both evaluation methods the scores are almost the same and too high. In close-set setup there are better scores due to the fact, that the proposed model trained

[3] https://www.tira.io/

and evaluated with samples from all LLMs. In contrast, for open-set setup the model trained with texts from different LLMs.

4. Conclusion

Based on the results, it is observed that, the proposed architecture and methodology is able to detect human vs AI generated texts with high accuracy, achieving above 90% on accuracy and F1-score. The high AUC and C@1 scores indicate that the model is able to reliably distinguish between human and AI generated texts. In the future, it would be preferable to train a Siamese model architecture within a contrastive learning framework either with simple contrastive loss [10] or with a triplet loss with online and fast hard negatives mining[15] in order to train a model to produce different Vectors of Embeddings for human and LLM texts in a Vector space. This could help improve the model's generalization ability to detect AI texts from unseen and future text generators. Also, additional features, such as POS-tags, can be combined with contextualized word Embeddings, producing Vectors that can be treated as features of a Classifier. Overall, the proposed approach shows promising results on this task, but continual research is needed to keep up with the rapid advances in large language models.

5. 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 ThinkingAnalysis, 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 theFifteenth International Conference of the CLEF Association (CLEF 2024), Lecture Notes in ComputerScience, Springer, Berlin Heidelberg New York, 2024.

[2] J. Bevendorff, M. Wiegmann, J. Karlgren, L. Dürlich, E. Gogoulou, A. Talman, E. Stamatatos,M. Potthast, B. Stein, Overview of the "Voight-Kampff" Generative AI Authorship Verification Taskat 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 WorkshopProceedings, CEUR-WS.org, 2024.

[3] Liu, Yinhan, et al. "Roberta: A robustly optimized bert pretraining approach." arXiv preprint arXiv:1907.11692 (2019).

[4] Mitchell, Eric, et al. "Detectgpt: Zero-shot machine-generated text detection using probability curvature." International Conference on Machine Learning. PMLR, 2023.

[5] Bekker, Jessa, and Jesse Davis. "Learning from positive and unlabeled data: A survey." Machine Learning 109.4 (2020): 719-760.

[6] Tian, Yuchuan, et al. "Multiscale positive-unlabeled detection of ai-generated texts." arXiv preprint arXiv:2305.18149 (2023).

[7] Hans, Abhimanyu, et al. "Spotting LLMs With Binoculars: Zero-Shot Detection of Machine-Generated Text." arXiv preprint arXiv:2401.12070 (2024).

[8] Bao, Guangsheng, et al. "Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature." arXiv preprint arXiv:2310.05130 (2023).

[9] Efstathios Stamatatos, Krzysztof Kredens, Piotr Pezik, Annina Heini, Janek Bevendorff, Martin Potthast, and Benno Stein. Overview of the Authorship Verification Task at PAN 2023. CLEF 2023 Labs and Workshops, Notebook Papers, September 2023.

[10] Petropoulos, Panagiotis. "Contrastive learning for authorship verification using BERT and bi-LSTM in a Siamese architecture." Working Notes of CLEF (2023).

[11] Fröbe, Maik, et al. "Continuous integration for reproducible shared tasks with TIRA. io." European Conference on Information Retrieval. Cham: Springer Nature Switzerland, 2023.
[12] Kestemont, Mike, et al. "Overview of the cross-domain authorship verification task at PAN 2020." Working notes of CLEF 2020-Conference and Labs of the Evaluation Forum, 22-25 September, Thessaloniki, Greece. 2020.
[13] Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." the Journal of machine Learning research 12 (2011): 2825-2830.
[14] A. Peñas, A. Rodrigo, A simple measure to assess non-response (2011).
[15] Gajić, Bojana, Ariel Amato, and Carlo Gatta. "Fast hard negative mining for deep metric learning." Pattern Recognition 112 (2021): 107795.