An Ensemble Method Based on the Combination of Transformers with Convolutional Neural Networks to Detect Artificially Generated Text

Vijini Liyanage and **Davide Buscaldi**
LIPN , Université Sorbonne Paris Nord, CNRS UMR 7030
99 av. Jean-Baptiste Clément, 93430 Villetaneuse, France
{liyanage,davide.buscaldi}@lipn.univ-paris13.fr

Abstract

Thanks to the state-of-the-art Large Language Models (LLMs), language generation has reached outstanding levels. These models are capable of generating high quality content, thus making it a challenging task to detect generated text from human-written content. Despite the advantages provided by Natural Language Generation, the inability to distinguish automatically generated text can raise ethical concerns in terms of authenticity. Consequently, it is important to design and develop methodologies to detect artificial content. In our work, we present some classification models constructed by ensembling transformer models such as SciBERT, DeBERTa and XLNet, with Convolutional Neural Networks (CNNs). Our experiments demonstrate that the considered ensemble architectures surpass the performance of the individual transformer models for classification. Furthermore, the proposed SciBERT-CNN ensemble model produced an F1-score of 98.36% on the ALTA shared task 2023 data.

1 Introduction

Nowadays, people have access to state-of-the-art LLMs which help them simplify some of their daily activities. One of the most notable breakthroughs in recent years is the evolution of OpenAI's GPT models which are capable of generating text that looks as if they are written by a human. Especially, the latest models such as ChatGPT and GPT4 (OpenAI, 2023) have won global attention for providing solutions to any kind of question or concern that humans possess. Moreover, these models produce outputs that appear to be written by a human.

Thus there is a potential risk in determining the authenticity of textual content that mankind refers to. Especially, in a domain such as academia, leveraging generation models in composing articles might raise an ethical concern. For example in ICML 2023, they have included a note under the "Ethics" section prohibiting the use of text generated by ChatGPT and other LLMs, unless "presented as part of the paper's experiential analysis." [1]. Accordingly, it is essential to have mechanisms for detecting artificially composed text from human written text.

Currently, a substantial amount of research has focused on the detection of automatically generated text. Recent research ((Zellers et al., 2019), (Glazkova and Glazkov, 2022) and Liyanage and Buscaldi (2023)) mostly consider detection as a binary classification task and leverage SOTA classification models to distinguish machine-generated text from original text. Besides, some employ statistical detection tools such as GLTR (Gehrmann et al., 2019) or latest deep learning based tools such as GPT2 output detector[2], DetectGPT (Mitchell et al., 2023) or GPTZero [3]. Moreover, several researchers (Liyanage et al. (2022), (Kashnitsky et al., 2022)) have published corpora composed of machine-generated content, which can be utilized by future research on detection.

Our work is based on the participation of our team in the ALTA shared task 2023 (Molla et al., 2023) The objective of the task is to build automatic detection systems that can discriminate between human-authored and synthetic text generated by Large Language Models (LLMs). Their corpus is composed of artificial contents that belong to a variety of domains (such as law, medical) and are generated by models such as T5 (Raffel et al., 2020) and GPT-X.

This paper is organized as follows. We provide the corpus and task description in Section 2. In Section 3, we describe our methodology and Section 4, deliver the experimental setup and the official results. Section 5 concludes this paper.

[1] https://icml.cc/Conferences/2023/llm-policy
[2] https://openai-openai-detector--5smxg.hf.space
[3] https://gptzero.me/

2 Task Overview

2.1 Task Definition

The task at hand revolves around distinguishing between automatically generated and human-written texts. In essence, it involves a binary classification challenge where the goal is to categorize provided texts into two distinct and exclusive groups. To outline this formally:

- Input: We are presented with text segments.
- Output: The objective is to assign one of two possible labels to each text segment: either "human-written" or "machine-generated".

This undertaking aims to establish a clear boundary between texts created through automated processes and those crafted by human authors. The primary aim is to develop a model that can effectively differentiate between these two categories based on the characteristics of the given excerpts.

2.2 Corpus

The dataset published for the ALTA shared task is a balanced one composed of 9000 original (human written) excerpts and 9000 fake (artificially generated) excerpts. On average, the excerpts consist of 35 words each. To gain a deeper comprehension of the corpus, category-wise (original vs generated) statistics with respective example excerpts are provided in Table 1.

3 Methodology

Given that the shared task frames detection as a binary classification challenge, we utilized a range of classification models to address this objective. In the subsequent subsections, in-depth explanations are provided pertaining to the examined statistical, recurrent and transformer models, and the corresponding ensemble architectures.

3.1 Statistical Models and their Respective Ensemble Architectures

In our work, we primarily employed Naive Bayes, Passive Aggressive and Support Vector Machine (SVM), which are classification algorithms used in machine learning to categorize data points into different classes (Bishop and Nasrabadi, 2006). Naive Bayes is a probabilistic classification algorithm based on Bayes' theorem and it is widely used for tasks such as spam detection. It assumes that the features are conditionally independent given the class label. Passive Aggressive is a type of algorithm that aims to make aggressive updates when it encounters a misclassified point and passive updates when the point is correctly classified. SVM is a powerful supervised machine learning algorithm used for classification and regression tasks. It is a popular algorithm in text classification tasks. These algorithms were employed in conjunction with the two text encoding methodologies, namely Bag of Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF).

Furthermore, we harnessed the capabilities of ensembles comprising the aforementioned statistical models, applying various ensemble methodologies such as voting, stacking, bagging, and boosting. By amalgamating the predictions of multiple models, ensemble techniques aim to enhance the overall predictive power of our system. Voting combines the outputs through a majority or weighted decision, stacking involves training a meta-model on the predictions of base models, bagging leverages bootstrapped subsets of data for training individual models, and boosting iteratively adjusts model weights to prioritize difficult-to-classify instances. Through these ensemble strategies, we sought to extract richer insights from our data and attain improved classification performance.

3.2 Recurrent Models and their Respective Ensemble Architectures

Recurrent models, a subset of neural network architectures, are models designed to capture temporal dependencies and patterns within sequences. We conducted experiments with LSTM and Bi-LSTM models, which are a type of RNN architecture specifically designed to address the vanishing gradient problem that can occur in traditional RNNs. To further improve classification accuracies of these models, we ensembled them with a Convolutional Neural Networks (CNNs) architecture. The proposed hybrid RNN-CNN approach helps in enhancing the predictive capabilities overall model by capitalizing on their respective strengths in capturing temporal dependencies and spatial features. We trained the entire ensemble end-to-end, allowing the network to learn how to best combine the features extracted by both LSTM and CNN components.

	Original	Generated
Min. word count	10	1
Max. word count	96	192
Avg. word count	25	45
Example excerpt	This is the data I collected so far (motorcycle standing on central stand, back wheel revolving, velocity comes from the back wheel, ABS LED blinking).	In this sense, she emphasized that it was a mistake to tie development aid to times of economic booms, as it is a "permanent commitment".

Table 1: Statistics of the ALTA shared task corpus (The avg. figures are rounded off to the nearest whole number)

3.3 Transformer Models and their Respective Ensemble Architectures

For our classification experiments, we leveraged cutting-edge transformer models, namely BERT, SciBERT, DeBERTa, and XLNet. These state-of-the-art architectures have demonstrated exceptional proficiency in a wide spectrum of natural language processing tasks, including classification. BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2018) introduces bidirectional context by pretraining on a massive corpus and then fine-tuning on task-specific data. SciBERT (Beltagy et al., 2019) is specialized for scientific text, adapting BERT's embeddings to domain-specific language. DeBERTa (Decoding-enhanced BERT with Disentangled Attention) (He et al., 2020) enhances attention mechanisms, capturing dependencies among words more effectively. XLNet (Yang et al., 2019) employs a permutation-based training approach to capture bidirectional context and alleviate BERT's limitations.

Initially, we created ensembles by combining the capabilities of SciBERT and DeBERTa models with the foundational BERT model. This process involves channeling the data through each base model, which comprises the transformer block along with a subsequent max pooling layer. Subsequently, the outcomes derived from these individual models are concatenated to generate a unified representation, which is then channeled into a linear classification layer for making refined predictions.

Furthermore, we combined the transformer model with Convolutional Neural Networks (CNNs) to build ensemble architectures that exhibit enhanced performance. As depicted in the architectural diagram 1, the embeddings produced by the transformer model are used as input for a CNN. This network includes three stacked convolutional layers to cover a large enough part of the input. The output of the three stacked layers is then passed through a dropout, a max pooling and another dropout layer before being passed to a dense layer for the classification. In our approach, we don't need to embed the output using nn.Embedding layers, as there is no need for a lookup table.

4 Experiments and Results

The text underwent preliminary processing, involving the elimination of stopwords and stemming, before being supplied to either statistical or neural network architectures. The processed data was then transformed into numerical vectors using Bag of Words (BoW) or tf-idf encoding techniques, which were subsequently utilized as inputs for the statistical models. All of the employed statistical models, as well as their corresponding ensemble methods, were imported from the Scikit-learn library. For constructing LSTM and CNN models, the relevant layers were imported from TensorFlow's Keras module. Training these recurrent models, including those combined with CNN ensembles, involved running 10 epochs. The LSTM and Bi-LSTM architectures were trained using batch sizes of 64 and 128, respectively.

Concerning transformer architectures and their associated ensembles, pre-trained models from Hugging Face (Wolf et al., 2020) were imported and subsequently fine-tuned through the utilization of Simple Transformers [4]. The BERT tokenizer was consistently employed across all models. The fine-tuning process involved 3 epochs, a batch size of 16, and a maximum sequence length of 128. Leveraging the T4 GPU Hardware accelerator, the average training time for models was approximately 30 minutes. For standalone models, the input consisted of unprocessed text, while ensembles underwent pre-processing involving punctuation removal and conversion to lowercase. As represented

[4] https://simpletransformers.ai

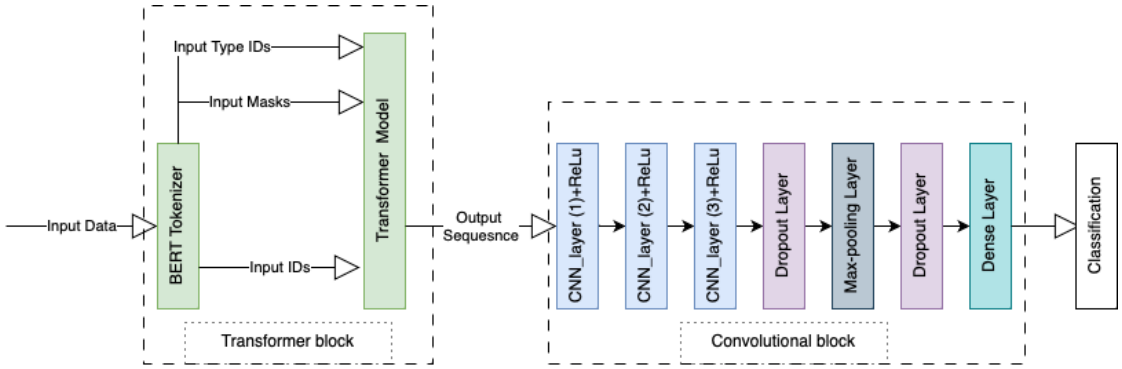

Figure 1: Architecture of Transformer-CNN Ensemble (Here, the “input type ids," “input masks," and “input ids" are the components used to prepare and encode the input data for the transformer model.)

in Figure 1, the CNN block of the ensembles was composed of three convolutional layers.

The dataset was split in 80:20 ratio for training and testing. To assess the classification performance of the models under consideration, the F1 score was employed. This score, being a balanced combination of precision and recall, offers a comprehensive evaluation. Each model underwent a total of five experimental iterations, and the resultant average F1 scores are presented in Table 2.

In general, the ensemble architectures have exhibited superior performance compared to their corresponding original models. Our best-performing solution is the combination of DeBERTa$_{large}$ with CNN, achieving an F1 score of **98.36%**.

Considering that baseline models such as Naïve Bayes and tf.idf weighting obtain scores close to 90%, it is clear that the dataset is not well balanced. In fact, looking at the Multinomial Naïve Bayes and the log probabilities differences for all features, we observed a thematic bias. Specifically, the top most probable words in the negative category (human-generated) are law-oriented: “plaintiff", “defendant", and “judgment". On the other hand, LLM-generated text contains words like “round", “league", “players", etc. Therefore, it is not clear whether these results are generalizable to the general task of detecting artificial text.

5 Conclusion

In this work, we have explored the application of different SOTA classification models on the detection of automatically generated text from human written text. Moreover, we have created various ensemble methods with the aforementioned models and examined their performance on the detection

Model	F1
Statistical Models	
NB + BoW	89.04
PA + BoW	84.07
SVM + BoW	87.51
NB + tf-idf	89.02
NB + tf-idf	91.00
NB + tf-idf	91.42
Ensembles of Statistical Models	
Voting (NB + PA + SVM) + BoW	90.29
Stacking (NB + PA + SVM) + BoW	88.23
Bagging (NB + PA + SVM) + BoW	91.56
Boosting (NB + PA + SVM) + BoW	90.28
Recurrent Models	
LSTM	49.08
Bi-LSTM	90.58
Ensembles of RNNs	
LSTM + CNN	49.08
Bi-LSTM + CNN	90.02
Transformer Models	
BERT$_{base}$	90.81
SciBERT	94.89
DeBERTa$_{large}$	96.67
XLNet$_{large}$	93.62
Ensembles of BERT models	
BERT$_{base}$ + SciBERT	97.80
BERT$_{base}$ + DeBERTa$_{large}$	97.47
Ensembles of transformers with CNN	
BERT$_{base}$ + CNN	97.42
SciBERT + CNN	97.56
DeBERTa$_{large}$ + CNN	**98.36**
XLNet$_{base}$ + CNN	97.44

Table 2: Classification Scores

task. Our results on the test data showed that generally the ensemble architectures outperform the considered original models. However, an analysis of the dataset raises some doubts about the generalizability of these results as it looks like the data are thematically biased. Therefore, these results should be considered only within the scope of the ALTA 2023 shared task.

As future work, we plan to examine the applicability of our ensemble architectures in detecting artificially generated text in multilingual corpora. Another potential research direction involves assessing the effectiveness of knowledge-based approaches for detecting artificial text.

References

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*.

Christopher M Bishop and Nasser M Nasrabadi. 2006. *Pattern recognition and machine learning*, volume 4. Springer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M Rush. 2019. Gltr: Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 111–116.

Anna Glazkova and Maksim Glazkov. 2022. Detecting generated scientific papers using an ensemble of transformer models. In *Proceedings of the Third Workshop on Scholarly Document Processing*, pages 223–228.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Yury Kashnitsky, Drahomira Herrmannova, Anita de Waard, Georgios Tsatsaronis, Catriona Fennell, and Cyril Labbé. 2022. Overview of the dagpap22 shared task on detecting automatically generated scientific papers. In *Third Workshop on Scholarly Document Processing*.

Vijini Liyanage and Davide Buscaldi. 2023. Detecting artificially generated academic text: The importance of mimicking human utilization of large language models. In *International Conference on Applications of Natural Language to Information Systems*, pages 558–565.

Vijini Liyanage, Davide Buscaldi, and Adeline Nazarenko. 2022. A benchmark corpus for the detection of automatically generated text in academic publications. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4692–4700.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. *arXiv preprint arXiv:2301.11305*.

Diego Molla, Haolan Zhan, Xuanli He, and Qiongkai Xu. 2023. Overview of the 2023 alta shared task: Discriminate between human-written and machine-generated text. In *Proceedings of the 21st Annual Workshop of the Australasian Language Technology Association (ALTA 2023)*.

OpenAI. 2023. Gpt-4 technical report.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. *Advances in neural information processing systems*, 32.