Journal of Computers in Education
https://doi.org/10.1007/s40692-024-00325-y

Unleashing the transformers: NLP models detect AI writing in education

José Campino[1]

Received: 14 September 2023 / Revised: 29 April 2024 / Accepted: 7 May 2024

Abstract
Artificial Intelligence (AI) has witnessed widespread application across diverse domains, with education being a prominent focus for enhancing learning outcomes and tailoring educational approaches. Transformer models, exemplified by BERT, have demonstrated remarkable efficacy in Natural Language Processing (NLP) tasks. This research scrutinizes the current landscape of AI in education, emphasizing the utilization of transformer models. Specifically, the research delves into the influence of AI tools facilitating text generation through input prompts, with a notable instance being the GPT-4 model developed by OpenAI. The study employs pre-trained transformer models to discern whether a given text originates from AI or human sources. Notably, BERT emerges as the most effective model, fine-tuned using a dataset comprising abstracts authored by humans and those generated by AI. The outcomes reveal a heightened accuracy in distinguishing AI-generated text. These findings bear significance for the educational realm, suggesting that while endorsing the use of such tools for learning, vigilance is warranted to identify potential misuse or instances where students should independently develop their reasoning skills. Nevertheless, ethical considerations must be paramount when employing such methodologies. We have highlighted vulnerabilities concerning the potential bias of AI models towards non-native English speakers, stemming from possible deficiencies in vocabulary and grammatical structure. Additionally, users must ensure that there is no complete reliance on these systems to ascertain students' performance. Further research is imperative to unleash the full potential of AI in education and address ethical considerations tied to its application.

Keywords Transformer models · Artificial intelligence · Natural language processing · ChatGPT · BERT · Education

✉ José Campino
josepedrocampino@gmail.com

[1] Nova School of Business and Economics, Carcavelos, Portugal

Published online: 13 June 2024

Introduction

The interest on Artificial Intelligence (AI) and its possible use in education is not new (Devedžić, 2004). However, the recent developments in this field, the improvement of computational resources and their availability, allowed the development of new technologies and further applications, inciting changes in the education system (Lund & Wang, 2023). In this field, the interest has been increasing and in recent years the number of papers published on the topic of AI are increasing (Chen et al., 2020). The application of AI in education has covered a wide range of activities according to Chen et al., (2020): (i) assessment of students and schools; (ii) grading and evaluation of papers and exams; (iii) personalized teaching; (iv) smart school; and (v) online and remote education. Its application in education has gained momentum, as an increasing number of institutions are exploring the use of AI for improving learning outcomes. AI has the potential to revolutionize the way of teaching and learning, posing advantages and dangers as any other technology.

Concerning the advantages of integrating AI in education, several key benefits have been recognized. Firstly, AI facilitates personalized learning by tailoring educational experiences to individual students based on their strengths, weaknesses, and preferred learning styles. This approach enables students to progress at their own pace, fostering heightened engagement, motivation, and ultimately yielding improved learning outcomes (Dimitriadou & Lanitis, 2023; Xu & Ouyang, 2022). Secondly, intelligent tutoring systems powered by AI offer instantaneous feedback to students, aiding in the identification and correction of mistakes. This not only alleviates the workload on teachers but also enhances the overall effectiveness of instructional methods (Xu & Ouyang, 2022). Thirdly, AI's capability for data analysis proves instrumental in scrutinizing extensive datasets. This analytical prowess assists teachers in discerning students' learning patterns, allowing for the customization of teaching strategies to enhance education quality and elevate student performance. Lastly, the integration of AI fosters accessibility in education, particularly benefiting students with disabilities. Notably, technologies such as speech recognition enable students with hearing impairments to actively participate in classroom discussions, thereby promoting inclusivity (Xu & Ouyang, 2022).

The incorporation of AI in education also presents inherent risks encapsulated. Firstly, the issue of bias arises, as AI algorithms may exhibit biases leading to discriminatory outcomes (Akgun & Greenhow, 2022). Liang et al. (2021)unveil a notable bias in GPT detectors against non-native English writers, evident in the high misclassification rate of non-native-authored TOEFL essays compared to the near-zero misclassification rate of presumed native-authored college essays. The discrepancy is attributed to limited linguistic variability and word choices by non-native authors, resulting in lower perplexity text. The study raises concerns about the reliability of current detection methods underscoring the need for more robust detection techniques that account for nuances introduced by prompt design. Secondly, the integration of AI in education raises legitimate privacy

concerns. The collection of personal data by AI systems, including biometric information and browsing history, raises the specter of misuse or theft, posing potential harm to students and educators alike (Akgun & Greenhow, 2022). Furthermore, there is the risk of dependence, wherein an over-reliance on AI has the potential to diminish critical thinking and problem-solving skills among students, leading to a scenario where students become overly dependent on AI systems for learning, thereby compromising their ability to think independently. Moreover, the introduction of such disruptive technology necessitates adaptation by both students and lecturers, encompassing understanding, potential applications, and correct administration of the technology. Lastly, there is a tangible risk of exclusion, wherein the imperative for simultaneous adaptation to such technologies may not be universally feasible due to various factors such as disparate levels of knowledge, economic disparities, or accessibility challenges, potentially resulting in exclusion from the benefits of AI in education.

The integration of AI technologies into education presents a noteworthy challenge related to plagiarism, necessitating the incorporation of effective control mechanisms to identify texts generated by AI (Abd-Elaal et al., 2022). While previous solutions have been put forth (Tien & Labbé, 2017; Shahmohammadi et al., 2020), they were not able to adopt recent advancements, particularly in Natural Language Processing (NLP). The emergence of NLP models based on the Transformer methodology, exemplified by Large Language Models (LLMs) like BERT and GPT (Generative Pre-trained Transformer), requires novel approaches. In this context, our contribution lies in leveraging such models to detect AI-generated content in essay-type texts within the context of university-level education. To achieve this goal, pre-existing models were fine-tuned using a database comprising both human and AI-generated academic abstracts. Notably, the BERT model exhibited superior performance, boasting high accuracy levels. The application of the fine-tuned model to an experimental database, comprising 200 human-generated texts and those produced by various AI models, provided compelling evidence of the practicality and efficacy of this approach. In essence, this paper aims to contribute to the ongoing discourse surrounding the integration of AI in education by proposing a novel methodology to address critical challenges associated with essay grading systems.

From a theoretical perspective, our research proposes a new approach to essay grading contributing to advancing our understanding of the intersection between AI, education, and ethics. By examining the challenges and implications of AI-generated content in academic settings, we shed light on broader questions of algorithmic fairness and educational equity. Moreover, our study provides insights into the evolving role of educators in navigating the complexities of AI-enhanced teaching and learning, highlighting the importance of pedagogical adaptation and ethical awareness in the digital age. In addition to its theoretical implications, our research has practical significance for a range of stakeholders in the education ecosystem. For educators, our findings offer valuable insights into the potential risks and opportunities associated with AI technologies in the classroom, empowering them to make informed decisions about assessment practices. Similarly, educational administrators can use our research to inform the development of guidelines governing the responsible use of AI in education, ensuring that ethical considerations are prioritized in

the adoption and implementation of AI-driven educational tools. For students, our findings hold the promise of enhanced academic integrity awareness through personalized mechanisms powered by AI, which have the potential to allow a faster and more accurate grading process using the methods proposed in this research together with future developments.

Literature review

Natural language processing (NLP) models

NLP models are machine learning models that are designed to process and understand human language. These models are used for a wide range of tasks, including machine translation, sentiment analysis or text classification. Previously, NLP models relied on hand-crafted rules and heuristics to analyze and process language (François et al., 2012). These models were limited by the complexity and ambiguity of natural language, and they often struggled to perform well on real-world tasks. More recently, however, there has been a shift towards using machine learning techniques to build NLP models. These models are trained on large datasets of human language, and they learn to recognize patterns and make predictions based on that data.

One of the key innovations in NLP models has been the development of pre-trained language models. These models are trained on massive amounts of text data and learn to represent the meaning of words and sentences in a high-dimensional vector space. These representations can then be fine-tuned on specific downstream tasks, such as sentiment analysis or machine translation, to improve their performance. There are many different types of NLP models, including rule-based models, statistical models, and deep learning models. Deep learning models, such as convolutional neural networks (CNNs) (O'Shea & Nash, 2015) and recurrent neural networks (RNNs) (Salehinejad et al., 2017), have become increasingly popular in recent years due to their ability to capture complex patterns in language data (Cho, et al., 2014; Peters, Neumann et al., 2018).

Recently, the Transformer architecture introduced by Vaswani et al., 2017, has become the standard for pre-training large-scale language models, with a different approach from RNN models (Raffel, et al., 2020). RNNs process input data sequentially and maintain a hidden state that captures information from the previous inputs. This hidden state is passed to the next time step and updated based on the current input, allowing the model to capture temporal dependencies in the input sequence. Transformers, on the other hand, use an attention mechanism to directly model the relationships between all input positions, rather than processing them sequentially (Dai, et al., 2019). The innovation brought by the Transformer architecture has led to a new era of pre-trained language models, where large-scale models are pre-trained on massive amounts of text data and then fine-tuned on specific downstream tasks. This has led to significant improvements in many NLP tasks, including machine translation, sentiment analysis and question answering. The recently developed critical models are summarized in Table 1 and can be categorized as follows: (i) BERT

Table 1 Summary of the pre-trained models used

	Transformer layers	Hidden size	Attention heads	Parameters	Processing	Length of training
BERT base	12	768	12	110 million	4 TPUs	4 days
ALBERT base	12	768	12	11 million	64 TPUs	1 day
RoBERTa base	12	768	12	125 million	32 TPUs	4 days
ELECTRA base	12	768	12	110 million	1 TPU	3 days
XLNet base	12	768	12	110 million	32 TPUs	3.5 days

(Bidirectional Encoder Representations from Transformers) introduced by Devlin et al., (2019); ALBERT (A Lite BERT) which is a modification of BERT introduced by Lan et al., (2020); RoBERTa (Robustly Optimized BERT Pretraining Approach) introduced by Liu et al., (2019); ELECTRA (Efficiently Learning an Encoder that Classifies Token Replacements Accurately) introduced by Clark et al., (2020) and; XLNet (eXtreme MultiLabelNet) introduced by Yang et al., (2019).

Pre-training a transformer model involves training the model on a large corpus of unlabeled data before fine-tuning it on a downstream task. The objective of pre-training is to learn general representations of language that can be reused for multiple downstream tasks, rather than optimizing the model for a specific task. Previosuly, training was done using specific databases such as news articles (Jozefowicz et al., 2016), Wikipedia articles (Merity et al., 2016) or books (Kiros, et al., 2015). The current approach is to train the transformer model in the most diverse database possible (Radford, et al., 2019), this will allow a posterior fine-tuning for specific tasks with a better performance. The most common pre-training objective for transformer models is masked language modeling (MLM) (Sinha, et al., 2021). In MLM, a certain percentage of the input tokens are randomly masked, and the model is trained to predict the masked tokens based on the context provided by the unmasked tokens. This objective encourages the model to capture the context-dependent relationships between different tokens in the input sequence. Another pre-training objective used in transformer models is next sentence prediction (NSP) (Shi & Demberg, 2019). In NSP, the model is trained to predict whether two sentences are contiguous or not, based on a given pair of sentences. This objective encourages the model to capture the relationships between different sentences in a document. Pre-trained models have several advantages over traditional NLP models. First, pre-trained models require less labeled data for fine-tuning. This is because the pre-training process allows the model to learn general language structures, which can be applied to new, specific tasks. This reduces the amount of labeled data required to achieve high performance on new tasks. Second, pre-trained models are transferable across languages and domains. This means that a model pre-trained on one language can be fine-tuned for a different language, without the need for large amounts of labeled data. Additionally, a pre-trained model can be fine-tuned on specific domains, such as legal or medical language, without requiring large amounts of labeled data in those domains.

AI development and the link with education

The use of AI in education has large impacts and consensually positive outcomes, particularly in administration, instruction and learning tasks. For the purpose of this research, it is particularly interesting the positive impacts the use of AI has in the engagement and improved learning of students and the needed control mechanisms (Chen et al., 2020). A remarkable advance in accessible and user-friendly AI is the wide publicly available ChatGPT by OpenAI in 2018.

The technology is an AI-powered chatbot or conversational agent developed by OpenAI, a leading artificial intelligence research organization. ChatGPT is based on the GPT (Generative Pre-trained Transformer) architecture, which is a type of deep learning neural network that has been trained on massive amounts of text data to generate natural language responses to user queries. ChatGPT is designed to engage in human-like conversation on a wide range of topics, from answering questions and providing information to engaging in casual chat. It is capable of understanding and responding to natural language inputs, using its knowledge of language and context to generate responses that are relevant, informative, and engaging. ChatGPT is accessible through various interfaces, such as web browsers, messaging platforms, or applications that integrate with the model's API. It can be used for a variety of purposes, such as customer service, education, entertainment, and more. Since the first generation lauch in 2018, OpenAI has developed the technology which had a large improvement with the 3.5 version. The latest version, to the date of the writing, is the GTP 4 which builds on the previous version and provides impressive solutions mainly concerning video, image recognition and code writing.

ChatGPT has tremendous impacts in academia, libraries and consequently in education (Lund & Wang, 2023). However, there are fragilities in the technology (Mathew, 2023) and the model still needs improvement has it still provides biased and wrong answers which can compromise its overall accuracy (Johnson, et al., 2023). This should be improved in the future as new generations will be released. The use of technologies such as ChatGPT will disrupt the current academic environment and will likely become as common as current tools such as a calculator or a computer (McMurtrie, 2023). The writing has become automated, and the grading of essays might be biased due to the use of this technology. This might have impacts at the level of simple essays but it may also affect the academic thesis writing. Arguing the possible end of essays does not seem to be a solution (Rudolph et al., 2023). Essays play an important role in the development of critical thinking, development of ideas and writing and speaking capabilities (Taghizadeh et al., 2020) from young ages until the higher education. Therefore, although AI tools usage should be supported due to the tremendous advantages it poses, this implies an extra step when evaluating essays which should be written individually and should be used to improve students' capabilities.

The necessity of assessing if an essay is written by a computer program was already raised in academia. Abd-Elaal et al., (2022) highlight the growing concern of computer-generated writing tools and their potential to undermine academic integrity and standards. It underscores the need for raising awareness among academics about these tools, developing ways to identify its characteristics, and

implementing clear policies to regulate its usage. Such mechanisms of computer writing detection were already developed, for instance, in the paper by Tien & Labbé, (2017). This work addresses the need for automatic detection of automatically generated texts to uphold the quality of bibliometric services. The study explores various text generation methods, demonstrating that documents produced by these methods can be reasonably well-classified. The paper introduces the Grammatical Structure Similarity (GSS) system, demonstrating an 80% positive detection rate and less than 1% false detection rate for sentences from known Probabilistic Context Free Grammar (PCFG) generators. The system's efficacy is highlighted against other machine learning techniques, though practicality diminishes when applied to generators using different techniques. Although the authors use machine learning techniques, the paper was an early research and thus not capable of capturing recent developments in this field. Hence, Shahmohammadi et al., (2020) discuss paraphrase detection, which is a fundamental task in the area of natural language processing. Paraphrase refers to sentences or phrases that convey the same meaning but use different wording. In this research, the authors propose a new deep-learning based model which can generalize well despite the lack of training data for deep models. The evaluation results show that the proposed model outperforms almost all the previous works in terms of F-measure and accuracy. Notably, the authors suggest that future research incorporates additional word embeddings, including ELMo, and leveraging state-of-the-art models like BERT and attention techniques that have gained recent attention in the field. However, as noticed by Weber-Wulff et al., (2023), the current AI writing detection tools are not as accurate as expected showing poor performance. The authors state that all detection tools scored below 80% of accuracy and only 5 over 70%.

Methodology

The main objective of this research is to use Transformer models to predict if a text is either written by AI or by a human. This is a difficult task because the NLP models are trained on large datasets to perfectly mimic human writing. However, there are patterns used by these models which can be captured by another trained Transformer model. The pre-trained Transformer model should then be fine-tuned to a specific task to perform well on text classification or any other assignment. By preprocessing the data and employing the AdamW optimizer, the model undergoes iterative refinement to enhance its performance. Unlike conventional techniques that often rely on simpler word frequency-based representations and lack the ability to capture semantic such as bag of words, TF-IDF, or n-grams, this method exploits the power of transformer models, enabling the capture of intricate patterns and contextual nuances. This is possible by leveraging the contextual understanding and self-attention mechanisms inherent in transformer architectures. The incorporation of performance measures ensures rigorous evaluation, providing a robust validation of the model's efficacy in distinguishing AI-generated content.

Table 2 Characteristics of the computer used to fine-tune and run the model

Characteristics	
OperatingFsystem	MacOS
Memory	16FGB
CPU	AppleFM1
GPU	AppleFM1F(no FCUDAFavailable)

Table 3 Descriptive statistics of the database used

	Small train	Small test	Large train	Large test
Nb.FOfFtexts	80	70	120	100
Min.FTokens	94	39	82	39
Max.FTokens	464	555	464	555
Avg.FTokens	200	184	202	178

Computer used

The task of text classification will most likely be performed at individual or at small groups' level. This is probably so due to the need of fine-tuning to a specific task and due to individual or small groups' needs. Therefore, the computational resources are likely to be limited. The estimations were performed on an individual computer with average characteristics that can be confirmed in Table 2. Since fine-tuning does not require much computational resources, the task can be well performed using a computer with the characteristics mentioned. However, when replicating the results, the user should expect slow outputs, particularly when running several epochs, and some limitations in terms of data usage. In other words, the number of observations per database should be smaller in such machines.

Databases

Following the previous reasoning, small databases have been used to perform the fine-tuning, which characteristics can be confirmed on Table 3. The main reason is to test if the task of fine-tuning can easily be performed at individual level in a daily use.

Two databases were created: (i) training and (ii) testing. The test database is particularly large compared with the training database with the objective of accurately show the performance of the models. The databases are composed of three columns: (i) text to be analyzed, (ii) label identifying if a text is written by AI (binary variable); (iii) source of the text for future identification.

For this research, the text used for classification was composed of abstracts from academic research papers. These abstracts were collected from a search in Scimago using the simple keyword "leadership". Then, a random selection of the abstracts from several research fields and from the year of 2023 was performed. Thereafter,

the title of the research papers selected was used as a prompt to ask a text generation model to write an abstract for a paper with that title. Hence, it is possible to approximate the themes of the human written abstracts with the ones written by AI, improving the quality of the performance results. The model used to generate the texts was the GPT-3.5 developed and trained by OpenAI. This can be done via a simple code in Python with the respective API or simply by using the available tools such as ChatGPT which is constantly being updated and currently is based on the GPT-4 version.

Fine-tuning

As already mentioned, the task of fine-tuning is essential for a good performance of the transformer model on a specific task. This fine-tuning allows a small adjustment on a pre-trained model to drastically improve its performance on a designated task.

However, first the model was tested without any previous fine-tuning and with common treatments of the data. The data was simply inputted as a Comma-Separated Values (CSV) file with the format described before. Then, a treatment was applied to the database which includes the following: (i) removal of special characters and numbers: remove all the special characters in the text such as commas or parentheses. Numbers were also removed from the text. The removal of such characters did not show meaningful differences in the final result as the model is focused on the tokens (i.e.: words) and not on punctuation, for example; (ii) convert to lower case: convert the text to lower case to make it completely uniform; (iii) combine the text into a single line: in case an abstract has several paragraphs, it was assured that all the text fits to a single line; (iv) remove extra spaces: delete any extra spaces between words resulting of previous data treatment, leaving a single space between words.

The application of this treatment to the data is crucial for enabling the model to capture significant patterns while mitigating noise and potential bias resulting from variations in the text or distinctive writing styles. Two additional treatments were considered, but their incorporation into the code was ultimately excluded. The initial treatment involves eliminating stop words (e.g., a, the, is), and the second treatment involves applying stemming to the text, reducing each word to its base form (e.g., transforming "creating" and "creative" to "create"). The rationale behind not applying these treatments is rooted in the endeavor to execute a complex text classification task. In this context, the composition and structure of the phrase hold paramount importance. The implementation of stop word removal and stemming could compromise the contextual integrity of the phrase, rendering the AI-generated text indistinguishable from human-generated text. The test utilizing the standard transformer model without any fine-tuning was conducted using the code chunk presented in Table 4 (i.e., in this instance, for BERT). The test database and model were loaded, enabling the computation of performance metrics.

Subsequently, the fine-tuning operation was executed using the code chunk provided in Table 5. In this step, the AdamW optimizer was employed to refine the model. AdamW represents a variation of the Adam optimization algorithm

Table 4 Code chuck used to test the transformer models without fine-tuning

```
import pandas as pd
import numpy as np
import torch
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score,
roc_auc_score, average_precision_score
from transformers import BertForSequenceClassification, BertTokenizer

# Load pre-trained BERT model and tokenizer
model = BertForSequenceClassification.from_pretrained('bert-large-uncased')
tokenizer = BertTokenizer.from_pretrained('bert-large-uncased')

# Load dataset with human and AI-generated text
df = pd.read_csv("PATH_OF_DATA/TEST.csv")

# Tokenize text and convert to PyTorch tensors
inputs = tokenizer(df['text'].tolist(), padding=True, truncation=True, max_length=512,
return_tensors='pt')
labels = torch.tensor(df['label'].tolist())

# Evaluate model on dataset
outputs = model(**inputs, labels=labels)
loss = outputs.loss.item()
predictions = torch.argmax(outputs.logits, axis=1)
accuracy = accuracy_score(labels.detach().cpu().numpy(), predictions.detach().cpu().numpy())
precision = precision_score(labels.detach().cpu().numpy(), predictions.detach().cpu().numpy())
recall = recall_score(labels.detach().cpu().numpy(), predictions.detach().cpu().numpy())
f1 = f1_score(labels.detach().cpu().numpy(), predictions.detach().cpu().numpy())
auc_roc = roc_auc_score(labels.detach().cpu().numpy(), predictions.detach().cpu().numpy())
auc_pr = average_precision_score(labels.detach().cpu().numpy(),
predictions.detach().cpu().numpy())

print('Loss: {:.4f}'.format(loss))
print('Accuracy: {:.2f}%'.format(accuracy * 100))
print('Precision: {:.2f}%'.format(precision * 100))
print('Recall: {:.2f}%'.format(recall * 100))
print('F1 score: {:.2f}%'.format(f1 * 100))
print('AUC-ROC: {:.2f}%'.format(auc_roc * 100))
print('AUC-PR: {:.2f}%'.format(auc_pr * 100))
```

Table 5 Code chuck used to fine-tune the models

```
# Fine-tune BERT model on the train set
optimizer = AdamW(model.parameters(), lr=2e-5)
for epoch in range(num_epochs):
    optimizer.zero_grad()
    outputs = model(**train_inputs, labels=train_labels)
    loss = outputs.loss
    loss.backward()
    optimizer.step()
```

(Adaptive Moment Estimation) initially introduced by Loshchilov & Hutter (2019). Notably, AdamW incorporates an additional weight decay term compared to the conventional Adam, specifically designed to mitigate overfitting. The inclusion of the weight decay term in AdamW serves to penalize substantial weights within the model, a factor known to contribute to overfitting. Although Adam has demonstrated effectiveness across a broad spectrum of deep learning tasks, it occasionally grapples with overfitting challenges. The introduction of AdamW addresses this concern by explicitly regulating the model through weight decay.

The code in Table 5 defines an optimizer using the AdamW algorithm with a learning rate of $2e^{-5}$, and then fine-tunes a pre-trained model for a specified number of epochs. During each epoch, the optimizer's gradients are reset to zero using the *optimizer.zero_grad()* method. Then, the model's forward propagation is performed with the input data *train_inputs* and target labels *train_labels*, and the resulting loss is calculated. The backward propagation is then performed using the *loss.backward()* method to compute gradients, and the optimizer's *step()* method is called to update the model parameters based on these gradients. Overall, this code implements the basic training loop for fine-tuning a pre-trained language model using the AdamW optimizer.

The learning rate of $2e^{-5}$ means that the optimizer adjusts the model's parameters by a factor of $2e^{-5}$ times the computed gradients during each update step. The learning rate is a hyperparameter that controls the step size of the optimizer during parameter updates. It determines how much the model's parameters should be adjusted based on the computed gradients. A smaller learning rate means that the model parameters will be updated more slowly and cautiously, which can help prevent overshooting the optimal values. However, it also means that the optimization process may take longer to converge. On the other hand, a larger learning rate means that the model parameters will be updated more aggressively, which can result in faster convergence but may also cause the optimizer to overshoot the optimal values.

Performance measures

Evaluating the model's performance is a crucial aspect of this study, encompassing assessments for both non-fine-tuned models and at each epoch during the fine-tuning process. This comprehensive approach facilitates the computation of average performance metrics over multiple epochs. The evaluation employs the test database for all metrics except for loss, which is computed during fine-tuning. The measures utilized include firstly the loss, which is a measure of the error of the model on the training data. It is computed using the cross-entropy loss function as per Eq. (1).

$$Cross\ Entropy\ Loss = -\sum_i y_i \bullet \log(p_i) \tag{1}$$

where: y_i is the true probability of class i, and p_i is the predicted probability of class i

Secondly, accuracy is the proportion of correctly classified examples out of all examples. It is computed as the number of true positives and true negatives divided by the total number of examples as per Eq. (2).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

where: TP (true positives), TN (true negatives), FP (false positives), and FN (false negatives)

Thirdly, precision is the proportion of true positives among all examples classified as positive. It is computed as the number of true positives divided by the total number of positive predictions as per Eq. (3).

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

where: TP (true positives) and FP (false positives)

Fourthly, recall is the proportion of true positives among all actual positive examples. It is computed as the number of true positives divided by the total number of positive examples as per Eq. (4).

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

where: TP (true positives) and FN (false negatives)

Fifthly, F1-score is the mean of precision and recall. It is a measure of the balance between these two measures as per Eq. (5).

$$F1 = \frac{2 * Precision + Recall}{Precision + Recall} \tag{5}$$

Sixthly, AUC-ROC is the area under the receiver operating characteristic (ROC) curve. It is a measure of the model's ability to distinguish between positive and negative examples. The ROC curve plots the true positive rate (TPR) against the false positive rate (FPR) for different threshold values. The AUC-ROC is computed as the area under the ROC curve. Lastly, AUC-PR: AUC-PR is the area under the precision-recall curve. It is a measure of the model's ability to retrieve positive examples. The precision-recall curve plots the precision against the recall for different threshold values. The AUC-PR is computed as the area under the precision-recall curve.

Analyze an entire document

After the fine-tuning and the definition of the performance measures, the model can be saved and is ready to use in the task defined. In real world scenario, it may be important to input an entire document and verify the probability of such a document being written by AI. This can be a biased analysis as parts of the document may be written by AI and other by humans, and the result will depend on the proportions of the text authorship. The best option would be to analyze smaller portions of text but as this is not always possible or comfortable, the transformer models allow an analysis of an entire document.

The suggestion is that for such analysis, the user should import an entire PDF file and convert it into simple text as in Table 6. After that, the text preprocessing techniques already explored should be applied. If the process is well conducted, the data will end up being exactly as a simple text input but with a considerable larger content. The transformer models have a limit of tokens which they can analyze at once. For example, the BERT base model can analyze 512 tokens. Therefore, it is crucial to divide the imported file into text chucks not exceeding the capacity limit of the

Table 6 Code chuck used to import the entire PDF document

```
import PyPDF2
with open('paper.pdf', 'rb') as pdf_file:
    pdf_reader = PyPDF2.PdfFileReader(pdf_file)
    text_to_analyze = ''
    for page_num in range(pdf_reader.numPages):
        page = pdf_reader.getPage(page_num)
        text_to_analyze += page.extractText()
```

model selected. Then, a loop should be created to sum the probability of each text chunk being generated by AI and later a simple mean should be performed.

Python and libraries used

To execute the analysis proposed in this research, it is essential to utilize the latest version of the Python coding language (currently Python 3.11). In this case, Python was employed, installed through Anaconda, allowing for the straightforward construction of environments and library installation. Specifically for the tasks outlined, heavy reliance was placed on the Transformers library, which offers state-of-the-art Natural Language Processing (NLP) capabilities. This library is constructed on top of PyTorch and TensorFlow, featuring pre-trained models for various NLP tasks, including text classification, question answering, and language translation. Additionally, the Torch library, designed for building and training neural networks, was utilized, particularly in the realm of deep learning and scientific computing. The Sklearn library, with a focus on the Metrics module, played a crucial role, providing a suite of functions for evaluating machine learning model performance. Finally, foundational libraries such as Numpy and Pandas were incorporated into the toolkit.

Results

As examined earlier, the primary aim of the research is to assess the probability of a text being composed by AI. To accomplish this objective, pre-trained Transformer models were employed. Initially, pre-trained models were utilized without any fine-tuning. Following this, the models underwent fine-tuning to discern whether a scientific abstract was authored by an AI tool, relying on the distinctive writing styles inherent in both human and AI-generated content.

Standard models

When employing the standard version of Transformer models (i.e., without fine-tuning), the results of performance measures exhibit inconsistency, as detailed in Table 7. Subpar outcomes were observed across all performance metrics. Notably, there were elevated values for loss and diminished values for accuracy. Among the models assessed, BERT demonstrated the most favorable performance, albeit with an accuracy of only 53.75% and a precision of 52.17%. BERT and XLNet displayed

Table 7 Results of the transformer models without fine-tuning

	ALBERT	BERT	ELECTRA	RoBERTa	XLNet
Loss	0.7478	**0.6868**	0.6904	0.6961	0.7183
Accuracy	33.75%	**53.75%**	50.00%	50.00%	51.25%
Precision	38.98%	**52.17%**	0.00%	0.00%	50.70%
Recall	57.50%	**90.00%**	0.00%	0.00%	**90.00%**
F1	46.46%	**66.06%**	0.00%	0.00%	64.86%
ROC-AUC	33.75%	**53.75%**	50.00%	50.00%	51.25%
PR-AUC	43.67%	**51.96%**	50.00%	50.00%	50.63%
TimeF (min.)	1.77	**1.67**	1.72	1.69	3.49

The bold means that were the best results in a given test

commendable results for recall, indicating their ability to correctly identify positive values. However, considering all the other metrics, utilizing the models in their standard configuration for text classification, especially in the context of the intricate task outlined in this research, may be no more precise than a random guess.

Fine-tuned models

Considering the results obtained for the use of the standard models without any fine-tuning, it becomes clear that this step is crucial to obtain a more precise model. Hence, the fine-tuning of the models was performed and the results are summarized in Table 8.

Notable advancements are apparent with the implementation of the fine-tuning step, leading to improved performance measures across all models. Even after only 12 epochs, a relatively limited number of training periods, BERT emerges as the most superior model among its peers. The model demonstrates robust values, with an accuracy of 94.52% and a precision of 93.66%. Significantly, the recall value is exceptionally high at 96.90%, second only to the ELECTRA model with a value of 97.62%. Other metrics also reveal similarly elevated results, hovering around the 95% mark. In the context of our specific objective, emphasis is placed on the loss metric, representing the model's error. BERT maintains an average loss of 0.3513, which, though improved, remains relatively high compared to the lowest recorded value of 0.1572. Additionally, BERT distinguishes itself for its swift convergence, achieving this in 87.35 min in the described environment. The BERT model unequivocally stands out as the top performer, closely followed by the ELECTRA model after fine-tuning. Consequently, the number of epochs was extended to 20, and both the BERT and ELECTRA models were fine-tuned, with the results detailed in Table 9.

With an increased number of epochs, there is evident improvement in the results for performance measures. Considering the loss as a pivotal metric, BERT reduces the average to 0.3179, with the minimum value reaching a mere 0.0521. While the average values for all other measures remain similar, the noteworthy enhancement

Table 8 Results of the transformer models after fine-tuning

	ALBERT			BERT			ELECTRA			RoBERTa			XLNet		
	Minimum	Average	Maximum	Minimum	Average	Maximum	Minimum	Average	Maximum	Minimum	Average	Maximum	Minimum	Average	Maximum
Loss	0.1321	0.3714	0.7856	0.1572	**0.3513**	0.7171	0.4134	0.5872	0.6994	0.3263	0.5838	0.6912	0.0411	0.2728	0.6868
Accuracy	50.00%	88.81%	100.00%	75.71%	**94.52%**	100.00%	52.86%	87.38%	95.71%	50.00%	77.38%	98.57%	65.71%	87.26%	95.71%
Precision	0.00%	82.26%	100.00%	67.31%	**93.66%**	100.00%	51.47%	84.30%	94.44%	50.00%	85.99%	100.00%	65.38%	84.27%	92.11%
Recall	0.00%	91.43%	100.00%	71.43%	96.90%	100.00%	94.29%	**97.62%**	100.00%	48.57%	80.71%	100.00%	40.00%	92.62%	100.00%
F1	0.00%	86.04%	97.22%	74.63%	**94.92%**	100.00%	67.96%	89.68%	95.89%	60.00%	78.43%	98.59%	53.85%	87.23%	95.89
ROC-AUC	50.00%	88.81%	100.00%	75.71%	**94.52%**	100.00%	52.86%	87.38%	95.71%	50.00%	77.38%	98.57%	65.71%	87.26%	95.71
PR-AUC	50.00%	86.31%	100.00%	67.61%	**92.64%**	100.00%	51.47%	83.32%	93.17%	50.00%	76.46%	97.22%	65.94%	81.83%	92.11
Epochs	12			12			12			12			12		
TimeF(min.)	96.98			**87.35**			98.48			94.59			173.18		

The bold means that were the best results in a given test

Table 9 Results of the transformer models fine-tuned with 20 epochs

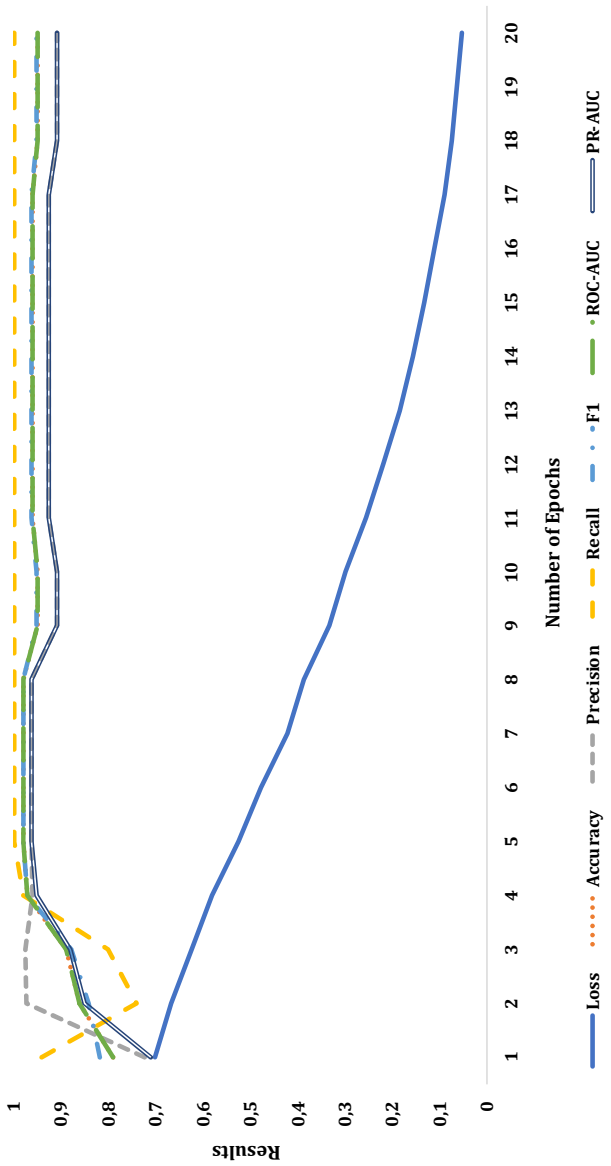
	BERT			ELECTRA		
	Minimum	Average	Maximum	Minimum	Average	Maximum
Loss	0.0521	**0.3179**	0.7012	0.1895	0.4833	0.7055
Accuracy	79.00%	**94.50%**	98.00%	50.00%	89.30%	97.00%
Precision	72.31%	**92.53%**	97.56%	50.00%	89.22%	100.00%
Recall	74.00%	**97.30%**	100.00%	76.00%	95.30%	100.00%
F1	81.74%	**94.61%**	100.00%	66.67%	91.01%	97.09%
ROC-AUC	79.00%	**94.50%**	98.00%	50.00%	89.30%	97.00%
PR-AUC	70.97%	**91.33%**	96.15%	50.00%	86.95%	95.04%
Epochs	20			20		
TimeF(min.)	**209.01**			226.80		

The bold means that were the best results in a given test

lies in the lowest values for these measures, which are distinctly higher after the epoch increase. BERT maintains its position as the fastest model, converging in 209.01 min compared to ELECTRA's 226.80 min. However, ELECTRA still exhibits poorer metrics when compared to BERT, underscoring the need for a further increase in the number of epochs to mitigate the loss. It is essential to approach this increase cautiously, as an excessively high number of epochs poses the risk of overfitting. The determination of a reasonable number of epochs is crucial, considering that performance values tend to plateau and show minimal improvement beyond a certain point. Graph 1 reinforces this observation, illustrating that as the loss decreases, the remaining metrics reach a plateau with only marginal differences in their results as the number of epochs increases. To facilitate comparison with the loss metric, all measures were converted to their original values rather than being presented as percentages. In practice, continuous testing of the model with real scenarios after several epochs is deemed crucial. Following the fine-tuning with 12 epochs, a real test to assess the model's performance on the proposed task is recommended. The same applies after increasing the number of epochs to 20 and beyond if warranted. In the context of this study and the task at hand, the BERT model exhibits robust results after 20 epochs, particularly considering the low value for the loss. However, ELECTRA evidently requires a higher number of epochs to yield improved results.

Experiment with the fine-tuned BERT model

The BERT model has been identified as exhibiting the most promising accuracy results across all conducted steps. Consequently, it has been chosen for a conclusive real test, and the ensuing results are presented. Emphasizing the significance of conducting such tests throughout the fine-tuning phase to calibrate the number of epochs is crucial. In this study, 20 epochs are deemed adequate to achieve a commendable



Graph 1 Results for the BERT model along fine-tuning with 20 epochs

model performance. However, in more intricate real-world scenarios, the necessary number of epochs might escalate.

To this end, BERT was fine-tuned, and both the model and the tokenizer employed (the model's tokenizer available in the Transformers' library) were saved. Subsequently, both elements were loaded, and functions were defined for text pre-processing and computing the probability of inputted text being generated by AI, based on the outcomes of our pre-trained and fine-tuned BERT model. The pre-processing of text holds significance to maintain uniformity with the input text format utilized during model fine-tuning. In our research, special characters and numbers were removed, the text was converted to lowercase, words were combined into a single line, and unnecessary spaces between words were eliminated.

Ultimately, the probability of a text being AI-generated is computed using the designated function. The Softmax function is applied to normalize the outputs, yielding probabilities that sum up to 1—forming a valid probability distribution over the binary output classes (AI-generated or not). Consequently, the probability returned by the predict function signifies the model's confidence in the input text being AI-generated, considering the model's training data and the fine-tuning process. The probability is then scaled up by 100 and presented with two decimal points, depicting it as a percentage value. The entire process is outlined in Table 10, accompanied by an example of the utilized code chunk.

Table 10 Code chuck used to test the fine-tuned models

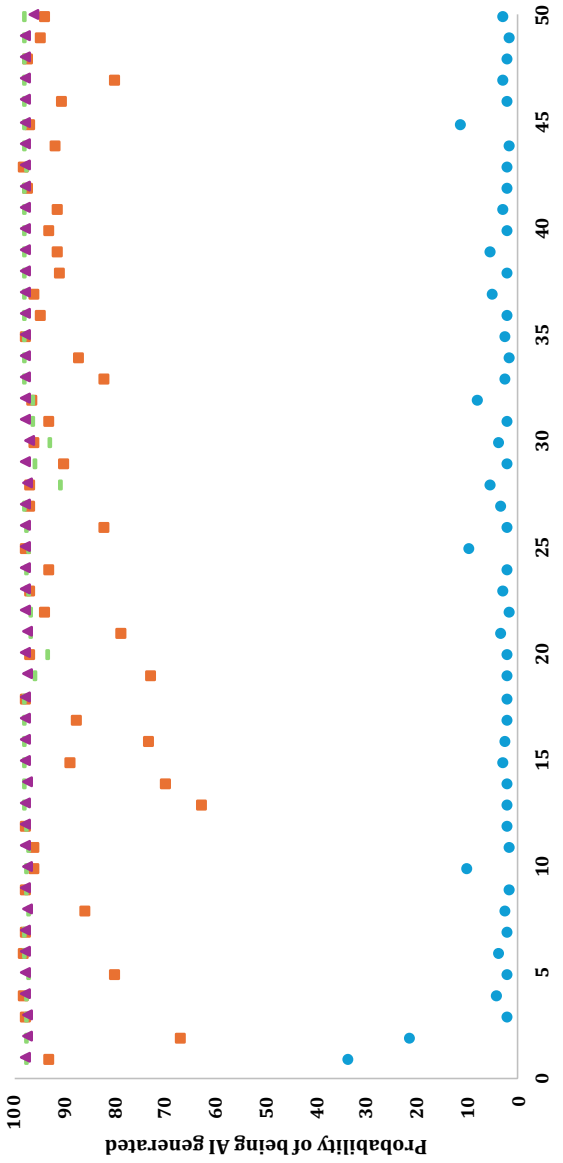
```
import re
import torch
from transformers import BertForSequenceClassification, BertTokenizer

# Load fine-tuned BERT model and tokenizer
model = BertForSequenceClassification.from_pretrained('fine_tuned_model')
tokenizer = BertTokenizer.from_pretrained('fine_tuned_model')

# Define function to preprocess text
def preprocess(text):
    # Remove special characters and numbers
    text = re.sub('[^A-Za-z\s]+', '', text)
    # Convert text to lowercase
    text = text.lower()
    # Combine all words into a single line
    text = ' '.join(text.split())
    # Remove more than one space between words
    text = re.sub('\s+', ' ', text)
    return text

# Define function to predict probability of input text being generated by AI
def predict(text):
    # Preprocess text
    text = preprocess(text)
    # Tokenize text and convert to PyTorch tensors
    inputs = tokenizer(text, padding=True, truncation=True, max_length=512, return_tensors='pt')
    # Make prediction using fine-tuned BERT model
    outputs = model(**inputs)
    # Extract probabilities from model output
    probs = torch.nn.functional.softmax(outputs.logits, dim=-1)
    # Return probability of input text being generated by AI
    return probs[0][1].item()

text="ABSTRACT HERE"

prob = predict(text)*100
print(f"Probability of text being generated by AI: {prob:.2f}%")
```

To evaluate the outcomes and assess their practical relevance, a novel database comprising 150 observations was meticulously curated. This dataset is equally divided between authentic abstracts extracted from Scopus in 2023, employing the keyword "machine learning," and synthetic abstracts generated for the same papers by three distinct models: (i) ChatGPT (supplied by OpenAI using GPT-3.5); (ii) Bing (supplied my Microsoft using GPT-4); (iii) Bard (supplied by Google). The models were invoked with identical commands. Graph 2 visually represents the outcomes, illustrating the precision of predictions made by the fine-tuned model. Our model adeptly distinguishes human-authored texts, assigning them significantly low probabilities of being machine-generated. Conversely, it consistently attributes high probabilities to texts originated from AI models. Specifically, when analyzing texts produced by GPT-3.5, the model consistently yields elevated probability values, consistently exceeding the 95% threshold. A similar scenario is verified when analyzing texts produced by Bing which used GPT-4. Nevertheless, the results unveil a discernible decline in accuracy when Bard is utilized in lieu of the other two models.

According to the data presented in Table 11, 100% of the predictions associated with texts generated by GPT-3.5 exhibit probabilities exceeding 95%, indicating a high confidence in identifying these texts as AI-generated. For Bing produced texts, 100% of the predictions made by our model are above the 90% probability threshold and 94% of the predictions are above the 95% probability threshold. However, the consistency of these values diminishes when employing Bard, where only 44% of predictions for texts generated by Bard surpass the 95% probability threshold of being AI-generated. Despite this variability, the predictions for Bard-generated texts maintain a commendable level of reliability. Specifically, 94% of the predictions surpass a 70% probability threshold, and 70% of the predictions exceed a 90% probability threshold, showcasing the robustness of the model in ascertaining the AI origin of the generated texts.

The obtained results, particularly the diminished accuracy observed for Bard-generated texts, align with the fine-tuning process applied to the model. The fine-tuning exclusively utilized texts from GPT-3.5 and human-authored sources, showcasing optimal performance in those specific scenarios. The same scenario is verified with Bing as it is based on the same baseline GPT model, although a more recent version is employed. However, as the evaluation to different generative models is extend, the efficacy of our fine-tuned detection model varies, contingent upon the generative model employed and the data used in its training. Consequently, the adaptability of tools designed for detecting AI-generated writing necessitates adjustments according to the specific generative model in use. Fine-tuning the detection model with texts from diverse sources is a potential approach. However, this may compromise overall accuracy as patterns between different types of AI writing become more challenging to discern. Such an approach would also demand increased computational resources, owing to the requirement for larger datasets and extended fine-tuning epochs.

Our research methodology suggests the prospect of creating distinct tools tailored to different AI writing models. These specialized tools, fine-tuned for specific models, could be more efficient and accurate. Notably, the availability of sufficient data for successful generative model training remains a challenge, limiting the diversity of models. As a result, while certain models may proliferate based on established



Graph 2 Probability of a text being generated by AI for Bard, Bing, GPT-3.5 and human texts

Table 11 Results breakdown for the probability of a text being generated by AI

%AI	Bard (%)	Bing (%)	GPT-3.5 (%)	Human (%)
> 10%	100	100	100	6
> 20%	100	100	100	4
> 30%	100	100	100	2
> 40%	100	100	100	0
> 50%	100	100	100	0
> 60%	100	100	100	0
> 70%	94	100	100	0
> 80%	84	100	100	0
> 90%	70	100	100	0
> 95%	44	94	100	0

methodologies, this study provides compelling evidence that tools developed for detecting text generated by prominent generative models can achieve high accuracy.

It is essential to acknowledge that the accuracy of our fine-tuned model is contingent upon data consistency between the fine-tuning and testing phases. In this study, the model was fine-tuned using abstracts, and its effectiveness in detecting AI writing is demonstrated on texts of the same type. However, caution is warranted when extending this accuracy to other text types, as the optimal fine-tuning approach for different text categories remains uncertain, even though accuracy may remain high.

Discussion

The Transformer methodology, initially introduced by Vaswani et al. (2017), has brought about a revolutionary shift in machine learning, particularly in Natural Language Processing (NLP) tasks. This study proposes a classification task aimed at identifying texts generated by AI, assigning a probability percentage to such occurrences. This task proves valuable in essay grading and control tasks designed for university-level students.

The Transformer methodology facilitated the development of pre-trained models that leverage extensive data, a resource typically unavailable at an individual level. This scarcity complicates the training process, rendering it exceptionally intricate. The utility of pre-trained models lies in their ability to achieve high performance across various tasks with a less complex fine-tuning process. This research substantiates the significance of this process by comparing performance measures of leading pre-trained Transformer models. When employing the standard model without fine-tuning, all models exhibited poor performance. This contrasts with the significantly improved results observed post fine-tuning, utilizing a database comprising abstracts from real academic articles and those generated by GPT-3.5.

The BERT model introduced by Devlin et al., (2019) was confirmed as the top-performer model analyzed for the task proposed in this research. Using the model without any fine-tuning process reveals levels of loss of 0.6868, accuracy of 53.75%,

and precision of 52.17% for the BERT model, which was still the best performer among its peers. After a fine-tuning process of 20 epochs, it achieved average values of 0.3179 for loss with a minimum value in the last epoch of 0.0521. As for accuracy, the model showed average values of 94.50% with a minimum value of 79% and a maximum value of 98%. Similarly, the precision results show an average value of 92.53% with a minimum value of 73.31% and a maximum value of 97.56%. Several studies have previously highlighted the need for AI writing detection tools as Abd-Elaal et al., (2022) showed the growing concern regarding this issue and the need to raise awareness among academics of the utilization of AI tools. Many tools for AI writing detection were developed, particularly since the launch of the disruptive tool ChatGPT in 2018 by OpenAI. However, as noticed by Weber-Wulff et al., (2023), the current AI writing detection tools show poor levels of accuracy as all detection tools scored below 80% of accuracy and only 5 over 70%. The reasoning of this paper is that the lack of accuracy is due to generalization as tools try to identify and be compatible with every type of text. Generative AI models can adapt to different circumstances and the result will show this adaptability. For example, a command to write in academic manner will provide a different result than a command to write in a creative manner. Hence, a generic tool will likely show reasonable levels of accuracy but still lacking specificity and eventually showing poorer results when compared with fine-tuned tools. This can be verified in the experiment conducted with the fine-tuned model. After this process, BERT was able to clearly discern between human written texts and the ones generated by GPT-3.5 and GPT-4. However, while using the same model on Bard generated texts, a reduction in accuracy was verified, showing the need for a fine-tuning process on a specific type of text.

Hence, if the classification task purpose is to detect AI writing in abstracts, the tools should be fine-tuned with this type of data. Similarly, the data used for fine-tuning should be adapted to the specific writing level or style. This research proposes the need to implement a control for detection of AI writing at university level. Therefore, the tool used needs to be fine-tuned to the type of writing at this schooling level. Analogously, if such tool would be used in different levels of the education system, it would need adaptation for the expected type of writing as the fine-tuning process needs to account for data provided by students of the schooling level analyzed. Hence, the fine-tuning process is a simple but crucial process to improve the model's performance. This process needs adaptations according to the pre-trained Transformer model used (i.e.: can be found in the accompanying information), particularly at the model and tokenizer steps identified in Table 10. It is important to emphasize the need for removing noise from the data with the preliminary cleaning process. The approach followed suggests removing special characters, converting the entire text to lower case, combining the text into a single line, and removing extra blanks spaces. This process will allow the model to focus only on the essential tokens providing better outputs. Lastly, the definition of the optimal number of epochs is a tricky process as the model should not be fine-tuned excessively to avoid overfitting to the data. The fine-tuning (i.e.: as well as a training process) should stop as soon as the performance measures achieve a plateau. Therefore, the process should be stopped manually, or with an early stop mechanism activated as soon as

the measures are not improving beyond the desired value. Several models should be fine-tuned and tested to get a perception of which process provided better outcomes.

In the attempt to provide solutions for AI detection tools, several research proposed early solutions such as Tien & Labbé, (2017) and Shahmohammadi et al., (2020). However, they were not able to build on the current techniques and models mainly developed after the introduction of BERT in 2019. Shahmohammadi et al., (2020) proposes the use of such techniques in future research as the authors have found this gap in the literature. This research tries to contribute to this line of thought including Transformer models in such AI writing detecting tools. Indeed, seemingly due to a fine-tuning process of a pre-trained viable model and the use in a specific context, the approach proposed here provides better results than the ones previously identified by Weber-Wulff et al., (2023) and Johnson and et al., (2023). Indeed, Shahmohammadi et al., (2020) have identified the best performance model to show an accuracy of 88.5% and a f-measure of 70.3%. After fine-tuning, the BERT model used in this research showed average values of accuracy of 94.5% with a minimum value of 79%, and average F1 values of 94.61% with a minimum value of 87.74%. A caveat for these results is the amount of data tested (i.e.: less amount of data needed due to previous training), and the specificity of the task requested as other studies seem to have adopted general detection of AI writing. Another differentiated feature of our analysis is the use of the probability and not a binary outcome. The BERT model allows the computation of a percentage showing the probability of a certain text being written by AI tools. This approach allows a nuanced view instead of a static one. The percentage allows the user to understand the level of certainty of the model and if it falls in uncomfortable levels (i.e.: between 40 and 60%), this should be an alert that a definitive conclusion should not be taken.

The use of tools such as the one proposed here, is in its initial phase and tremendous improvements and adjustments must be made. As this research proposes an alternative based on previous pre-trained models, several caveats arise from the use of such solutions. Liang et al. (2021) have shown convincing proof of a bias of AI writing detection tools towards non-native English speakers. Indeed, while in theoretical realms it might not be crucial, the existence of false negatives but particular the existence of false positives, might be an important issue. Any tool developed will show errors while in use, but the consequences of such errors might have several implications that are not currently foreseen. To account for true positives, our approach was to compute the recall value which for BERT showed an average value of 97.30% with a minimum value of 74% in the fine-tuning process. While the percentage of true positives is promising, it will certainly vary according to the task and in non-controlled environments. Furthermore, if the average value is considered, there are still 2.7% of wrongly classified observations.

As Lund & Wang, (2023) have identified, AI tools have tremendous impacts in society particularly in academia and education. Chen et al., (2020) have identified positives aspects of the adoption of such tools in education such as positive results in engagement and learning. However, several studies have identified possible fragilities of AI tools which are still in an early stage of development (Liang et al. 2021; Mathew, 2023). Certainly, the BERT or other similar model needs further research and experimentation before being largely applied. These tools should be adapted to

the reality of the task they are proposed to solve and undergo a long and thorough testing time before being implemented. Plagiarism tools are already implemented in most universities to make the scientific writing more rigorous and avoid deviant behaviors. The implementation of AI writing detection tools might undergo the same process of verifying their adaptability to the organization. It is crucial that, once the correct model was defined, it goes through an experimental period in which it is tested in real world scenarios. This experimental period should confirm the adaptability and accuracy of the tool as well as the capability of the infrastructure available.

As Chugh et al., (2023) point out in a study on the implementation of technologies in higher education, the type of technology being implemented plays a crucial role for the success of its implementation. Therefore, it is critical that the users know the technology and their fundamentals. As Esteve-Mon et al. (2021) states, the success of the implementation of digital tools depends critically on training and training strategies. Therefore, it is crucial to highlight the need for training of the users who are the ones evaluating the outputs of the tool. Human intelligence will always be crucial and irreplaceable, being the last resort to ascertain critically the accuracy of the model outputs. It is also interesting to note an apparent inherent contradiction of using AI to undo AI's work of convincingly mimicking the conventions of human writing. The implementation of AI tools in education has positive impacts (Chen et al., 2020) and its usage is inevitable. As technology implementation proceeds, negative effects might be found and they will need to be corrected. This research is focused on the detection of AI writing in essays at university level where such tools should not be used as a primary source, as they could create a bias in students' evaluation and jeopardize their critical thinking. Therefore, AI tools could be implemented under close human supervision to detect undesirable behaviors and improve fairness while technological solutions are fomented.

The present study holds significant educational implications owing to its innovative approach to the subject of AI-generated writing. The primary implication lies in the enhancement of essay grading efficiency through the automation of identifying AI-generated content, especially in contexts where its use is inappropriate. When combined with other tools like automatic grading, this process stands to become more efficient, less prone to bias, and quicker. Consequently, the promotion of academic integrity can be achieved by identifying and preventing instances of AI-generated content, thereby curbing plagiarism and fostering originality in academic endeavors. Such tools also have the potential to cultivate critical thinking skills. However, it's imperative to exercise critical thinking in their utilization. Maintaining human oversight and critically evaluating outputs, particularly in tasks such as essay grading, is indispensable. Educators face the challenge of striking a balance between the benefits of AI technology and its inherent limitations, ensuring it complements rather than supplants human intelligence and critical thinking abilities. Moreover, there's a pressing need for training and awareness regarding the technology and its underlying principles to effectively evaluate the outputs of AI writing detection tools and comprehend their limitations. This becomes especially crucial with models that require fine-tuning to enhance their effectiveness. The integration of critical thinking

and comprehensive training is paramount in mitigating ethical concerns, particularly those pertaining to biases and potential inaccuracies. Educators must remain vigilant in addressing these issues to ensure fair and impartial assessments.

The process outlined in this research holds promise for broader future developments that warrant attention. While this study serves as a preliminary exploration, showcasing a functional methodology capable of yielding compelling results, it's essential to recognize that the data compiled thus far is primarily for testing purposes. Consequently, it becomes imperative to apply this methodology to real-world data with genuine objectives. This approach facilitates the refinement of the model through a fine-tuning process and enables rigorous testing of its accuracy. Although the methodology demonstrates potential applicability across various domains, it's advisable to tailor its implementation specifically to academic contexts. Therefore, the methods employed herein should be adapted and applied across different educational levels to assess the consistency of results, given the varying nature of writing styles encountered. For instance, tools deployed at the university level should undergo fine-tuning to discern AI-generated writing specific to that academic setting, ensuring precision and relevance in detection. Moreover, it's crucial to employ these methods with diverse datasets comprising students from varied backgrounds and academic years, following a panel data structure. This approach enables the assessment of accuracy while also facilitating the identification of potential biases.

Conclusion

This research explores the impact of the Transformer methodology, particularly in the realm of Natural Language Processing (NLP). The proposed classification task, aimed at identifying texts generated by AI, presents a valuable application in the context of essay grading and control tasks for university-level students. Leveraging pre-trained Transformer models, this study highlights the significance of fine-tuning processes in enhancing model performance. Even without fine-tuning, BERT demonstrated superiority over its counterparts, and after a fine-tuning process, it exhibited improvements in key performance metrics.

Addressing the limitations observed in existing AI writing detection tools, this research underscores the importance of task-specific fine-tuning to enhance model reliability and accuracy. The proposed methodology emphasizes the need to tailor the fine-tuning process to the specific writing level, style, and context. Moreover, the significance of data cleaning, optimal epoch determination, and noise reduction in the preliminary stages of the process is highlighted.

While acknowledging the initial phase of AI writing detection tools, this research suggests caution in their implementation due to potential biases. As technology, including AI tools, continues to impact academia and education, it is crucial to approach their implementation with care and consideration. The study reinforces the importance of thorough testing and adaptation to the specific organizational context before the widespread deployment of AI writing detection tools. Human intelligence remains indispensable, serving as the ultimate arbiter to critically evaluate model outputs.

Practical implications

This research can have important implications in several fields and tasks. Particularly, we have applied our approach to education, as this area will most certainly be highly affected by the development of AI. The use of AI shows significant benefits in improving learning outcomes and providing personalized education. However, the development of tools that allow for automated text generation, such as GPT-4, can raise concerns about academic integrity and individual reasoning. The findings of the study suggest that pre-trained transformer models can effectively detect whether a given text was written by AI or by humans, which could help in identifying instances of academic dishonesty. This technology could be utilized by educational institutions to monitor the use of automated text generation tools and ensure that students are developing their own reasoning and critical thinking skills.

It is likely that such tools will be present more often in the future in other areas apart from education. This study may also have provided a contribution to this application and to the discussion of AI tools capable of generating text via the input of prompts. The study raises the awareness of the importance of developing human capabilities such as critical thinking and reasoning while adopting disruptive solutions that can certainly improve the way of teaching and learning, but also innumerous other aspects of daily life.

Limitations and suggestions

Although the interesting and encouraging results of this research, it is crucial to highlight some if its limitations. First, it is important to highlight the use of pre-trained models for a text classification task. The BERT model has revealed to be particularly accurate in this task. However, it can provide different results for different tasks or for different types of text. Second, and in order to guarantee accuracy, it is crucial to fine-tune the model. Nevertheless, the success of the fine-tuning process is completely dependent on the database's quality. The data should be as similar as possible to the one to which the model will be finally applied. Third, this study concluded that 20 epochs of fine-tuning process provide very accurate results. Nevertheless, this can differ depending on the task, on the database used, or the complexity of the real-world text. Fourth, the computational resources used were limited to show evidence of the application of such approach to real world scenarios. However, this also limits the amount of data used in the fine-tuning process and may make the process much slower. The approach proposed will become more accurate as the amount of data used for fine-tuning increases. Fifth, it is clear that the potential of transformer models in detecting text written by AI is large. Nonetheless, the approach proposed is not definitive and certainly has margin for improvement. It is crucial to bear in mind the possibility for error in such tools, particularly dependent on the fine-tuning process. This limitation is visible in the experiment conducted where applying a model fine-tuned using GPT-3.5 text in texts generated by Bard, revealed lower performances. Hence, the fine-tuning process should be adapted to the specific task requested and might not show similar performance levels for other tasks

(e.g.: the model fine-tuned with the simple purpose of detecting purely AI generated text might show lower performance when detecting paraphrasing). Furthermore, the bias propensity of such AI tools must not be overlooked and should also be considered when applying these solutions.

The approach here proposed will benefit from testing the results in much larger databases. This will improve the certainty that our proposed approach is reliable for larger sources of data. This should also be done using other data types and not only abstracts of academic papers. Furthermore, the need to test this approach with real experimental data is crucial. The approach here developed should be tested in real world scenarios to ascertain its real accuracy. In real world scenarios several limitations and drawbacks might arise which were not foreseen in controlled environments. For instance, the need to analyze an entire document might reveal several challenges as only part of the document might have been generated by an AI model. Although the tool might reveal effective in such scenarios, this will certainly affect the final percentage result leading to uncertainty. Therefore, the approach here proposed might also be tested with full documents and not only plain text.

Due to the high pace at which technology evolves, the approach here proposed might be outdated soon and other more promising solutions might arise. Nevertheless, we aim at contributing to the development of this field and to the discussion of such pressing matters. Furthermore, it is important to notice that our approach is not the only available to perform the task of text classification. It should be possible to perform such task using a logit model or a RNN model, if trained and designed for such purpose. This alternative approach will require a large amount of data. This burden is overcome using the pre-trained Transformer model. Nevertheless, it would be important to confirm the accuracy and usability of such approaches. Lastly, we suggest the application of the proposed approach to different languages. In this research, we have applied our approach only to the English language. Therefore, we cannot assume the same behavior in other languages, but it is expected that the pre-trained fine-tuned model will also show a good performance due to its pre-training phase.

Acknowledgements The author José Campino acknowledges the financial support of Fundação para a Ciência e Tecnologia through the project number PTDC/EGE-ECO/7493/2020.

Funding Open access funding provided by FCT|FCCN (b-on). The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Data availability The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

Ethical approval All of the followed procedures were in accordance with the ethical and scientific standards. This article does not contain any studies with human participants performed by the author.

References

Abd-Elaal, E.-S., Gamage, S. H., & Mills, J. E. (2022). Assisting academics to identify computer generated writing. *European Journal of Engineering Education*. https://doi.org/10.1080/03043797.2022.2046709

Akgun, S., & Greenhow, C. (2022). Artificial intelligence in education: Addressing ethical challenges in K-12 settings. AI Ethics.

Chen, L., Chen, P., & Lin, Z. (2020). Artificial Intelligence in Education: A Review. IEEE.

Cho, K., Merrienboer, B. v., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. Conference on Empirical Methods in Natural Language Processing. https://aclanthology.org/D14-1179.pdf.

Chugh, R., Turnbull, D., Cowling, M. A., Vanderburg, R., & Vanderburg, M. A. (2023). Implementing educational technology in Higher Education Institutions: a review of technologies, stakeholder perceptions, frameworks and metrics. *Education and Information Technologies*. https://doi.org/10.23919/EECSI56542.2022.9946579

Clark, K., Luong, M.-T., Le, Q. V., & Manning, C. D. (2020). ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. ICLR 2020. https://arxiv.org/pdf/2003.10555.pdf.

Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q. V., & Salakhutdinov, R. (2019). Transformer-XL: attentive language models beyond a fixed-length context. ACL. https://arxiv.org/pdf/1901.02860.pdf.

Devedžić, V. (2004). Web intelligence and artificial intelligence in education. Educational Technology & Society.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technology. https://arxiv.org/pdf/1810.04805.pdf.

Dimitriadou, E., & Lanitis, A. (2023). A critical evaluation, challenges, and future perspectives of using artificial intelligence and emerging technologies in smart classrooms. Smart Learning Environments.

Esteve-Mon, F. M., Postigo-Fuentes, A. Y., & Castañeda, L. (2021). A strategic approach of the crucial elements for the implementation of digital tools and processes in higher education. *Higher Education Quarterly*. https://doi.org/10.1111/hequ.12411

François, T., & Miltsakaki, E. (2012). Do NLP and machine learning improve traditional readability formulas? NAACL-HLT 2012. https://aclanthology.org/W12-2207.pdf.

Johnson, D., Goodman, R., Patrinely, J., Stone, C., Zimmerman, E., Donald, R., Wheless, L. (2023). Assessing the accuracy and reliability of AI-generated medical responses: an evaluation of the chatGPT model. Research Square.

Jozefowicz, R., Vinyals, O., Schuster, M., Shazeer, N., & Wu, Y. (2016). Exploring the limits of language modeling.

Kiros, R., Zhu, Y., Salakhutdinov, R. R., Zemel, R., Urtasun, R., Torralba, A., & Fidler, S. (2015). Skip-thought vectors. Advances in Neural Information Processing Systems.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2020). ALBERT: a lite BERT for self-supervised learning of language representations. ICLR 2020. https://arxiv.org/pdf/1909.11942.pdf.

Liang, W., Yuksekgonul, M., Mao, Y., Wu, E., & Zou, J. (2021). GPT detectors are biased against non-native English writers. *Cell Press*. https://doi.org/10.1016/j.patter.2023.100779

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Stoyanov, V. (2019). RoBERTa: a robustly optimized BERT pretraining approach.

Loshchilov, I., & Hutter, F. (2019). Decoupled weight decay regularization. ICLR. https://arxiv.org/pdf/1711.05101.pdf.
Lund, B. D., & Wang, T. (2023). Chatting about ChatGPT: how may AI and GPT impact academia and libraries? Library Hi Tech News.
Mathew, A. (2023). Is artificial intelligence a world changer? a case study of OpenAI's chat GPT. Recent Progress in Science and Technology.
McMurtrie, B. (2023). AI and the future of undergraduate writing. Retrieved from The Chronicle of of Higher Education: https://www.chronicle.com/article/ai-and-the-future-of-undergraduate-writing
Merity, S., Xiong, C., Bradbury, J., & Socher, R. (2016). Pointer sentinel mixture models.
O'Shea, K., & Nash, R. (2015). An introduction to convolutional neural networks.
Peters, M. E., Neumann, M., Iyyer, M., & Gardner, M. (2018). Deep contextualized word representations. NAACL-HLT 2018. https://aclanthology.org/N18-1202.pdf.
Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI Blog.
Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Li, W. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research.
Rudolph, J., Tan, S., & Tan, S. (2023). ChatGPT: Bullshit spewer or the end of traditional assessments in higher education? Journal of Applied Learning & Teaching.
Salehinejad, H., Sankar, S., Barfett, J., Colak, E., & Valaee, S. (2017). Recent advances in recurrent neural networks.
Shahmohammadi, H., Dezfoulian, M., & Mansoorizadeh, M. (2020). Paraphrase detection using LSTM networks and handcrafted features. *Multimedia Tools and Applications*. https://doi.org/10.1007/s11042-020-09996-y
Shi, W., & Demberg, V. (2019). Next sentence prediction helps implicit discourse relation classification within and across domains. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). https://aclanthology.org/D19-1586.pdf.
Sinha, K., Jia, R., Hupkes, D., Pineau, J., Williams, A., & Kiela, D. (2021). Masked language modeling and the distributional hypothesis: order word matters pre-training for little.
Taghizadeh, M. E., Abidin, M. J., Naseri, E., & Hosseini, M. (2020). In the importance of EFL learners' writing skill: is there any relation between writing skill and content score of english essay test? SciPress Ltd.
Tien, N. M., & Labbe, C. (2017). Detecting automatically generated sentences with grammatical structure similarity. *Scientometrics*. https://doi.org/10.1007/s11192-018-2789-4
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Polosukhin, I. (2017). Attention Is All You Need. In 31st Conference on Neural Information Processing Systems. https://arxiv.org/pdf/1706.03762.pdf.
Weber-Wulff, D., Anohina-Naumeca, A., Bjelobaba, S., Foltýnek, T., Guerrero-Dib, J., Popoola, O., & Waddington, L. (2023). Testing of detection tools for AI-generated text. *International Journal for Educational Integrity*. https://doi.org/10.1007/s40979-023-00146-z
Xu, W., & Ouyang, F. (2022). The application of AI technologies in STEM education: a systematic review from 2011 to 2021. International Journal of STEM Education.
Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., & Le, Q. V. (2019). XLNet: Generalized Autoregressive Pretraining for Language Understanding. 33rd Conference on Neural Information Processing Systems. https://arxiv.org/pdf/1906.08237.pdf.

José Campino PhD professor and researcher with a track record of international publications and participation in international events. Has coordinated several classes, including international classes, in the areas of management and strategy. Experienced professional in the financial/banking area who has worked for internationally recognized institutions. The research interests concern innovation, management, strategy, entrepreneurship, and finance.