# Transformer-Based Approaches to Detect AI Generated Content

EL-Hlaissi Nada Maach Zineb

**Abstract:**

In recent years, we have seen an impressive expansion of a family of Artificial intelligence models, known as generative AI, that are capable of producing fresh, unique material, including text, images, audio, and even code. Large datasets of previously published information are used to train these models, enabling them to mimic the patterns and structures of the data and produce original output that is stylistically and qualitatively comparable to the training set. While these models have many promising applications, they also carry significant risks and potential dangers that must be carefully considered, such as misinformation, intellectual property violations, or biased information. For these reasons, in this work, we have proposed a method to detect whether a sentence is human-generated or AI-generated. This paper contributes to the advancement of automatic text detection systems in addressing the challenges posed by machine generated text misuse.

## I. Introduction:

In recent years, the capabilities of large language models (LLMs) to generate f luent, realistic-sounding text have improved dramatically. We are now at a point where humans themselves cannot reliably distinguish between text that was generated using artificial intelligence (AI) and that written by a real person (Liu et al., 2023; Sarvazyan et al., 2023; Li et al., 2023b). There are many opportunities for LLMs to contribute to human productivity, with applications to question-answering, computer programming, brainstorming, proof reading, and information retrieval. Simultaneously, the field of automatic paraphrasing has evolved considerably, enabling systems to reformulate existing content while preserving semantic meaning a capability essential for applications ranging from content summarization to data

augmentation for training robust language models. Despite the benefits of these technologies, significant challenges have emerged. There are potential areas for misuse of AI-generated text, such as academic dishonesty, professional misconduct, and malicious chatbots (Crothers et al., 2023; Wu et al., 2023). Another major risk brought by generative AI is the potential for global-scale "information pollution," disinformation including and misinformation (Wardle & Derakhshan, 2017).

Current state-of-the-art generative models can produce high-quality, fluent fake information that is perceived as more credible and trustworthy than human generated misinformation (Zellers et al., 2019; Spitale et al., 2023), and that is harder for both human readers and automatic detection systems to recognize (Kreps et al., 2022; Zhou et al., 2023; Chen & Shu, 2023). The task of AI-generated text (AIGT) detection is a challenging problem with constantly moving goalposts: as researchers develop effective methods to detect text from the currently available LLMs, newer and larger models are released and the cycle continues. Furthermore, bad actors seeking to obfuscate their use of AI tools concurrently develop adversarial attacks on the detection methods, aiming to modify their AIGT to render it undetectable (Ghosal et al., 2023).

To address these dual challenges, we propose a comprehensive approach leveraging various deep learning architectures for both AI-generated text detection and paraphrase generation tasks. For AI-generated text detection, we fine-tuned several Transformer-based models, including BERT, RoBERTa, and BART, along with an optimized version of BERT using the Low-Rank Adaptation (LoRA) technique to improve efficiency. Additionally, we implemented traditional recurrent neural networks, such as LSTM, BiLSTM, and GRU, to establish performance baselines and understand the relative strengths of different architectural paradigms.

The present manuscript provides the details to our methodology, including dataset preparation, architectures, model and implementation details. It presents our experimental setup and evaluation metrics. Moreover, it discusses the results and comparative performance of the different approaches on both tasks. Finally, it concludes with a summary of our findings and directions for future research.

## II. Literature review:

This section reviews the most significant recent advances in AI Generated Text Detection, focusing on state of-the-art approaches that are most relevant to our work.

Early approaches to AI-generated text detection relied on statistical patterns and linguistic features that differentiated machine-generated from human-written text. Fagni et al. (2021) demonstrated that simpler LLMs like GPT-2 could be detected using statistical features such as word frequency distributions and entropy measures [1].

However, these methods proved less effective against more advanced models like GPT-3 and GPT-4, as highlighted by Mitrović et al. (2023), who found that statistical signatures become increasingly subtle with model sophistication [2].

More recent approaches have leveraged deep learning architectures for detection. Guo et al. (2023) proposed a RoBERTa-based classifier that achieved over 95% accuracy in detectingPT-3 generated texts on benchmark datasets [3].

Similarly, Zhong et al. (2023) demonstrated that fine-tuned BERT models could effectively identify texts from multiple different language models with varying degrees of success [4].

These approaches benefit from their ability to capture contextual information but typically require substantial computational resources for training and inference. LSTM and GRU-based models have also been applied to this task. Mitchell et al. (2023) used a BiLSTM architecture that achieved comparable transformer-based results models to while requiring fewer computational resources [5].

However, Li and Wang (2023) observed that recurrent architectures generally underperform compared to transformers when dealing with longer text sequences or more sophisticated language models [6].

A recent trend in AI-generated text detection is the application of parameter-efficient f ine-tuning techniques. Yang et al. (2024) demonstrated that LoRA

fine-tuning of transformer models could maintain detection performance while reducing the computational footprint by up to 90% [7].

This approach has shown particular promise for real-time detection systems and resource-constrained environments. Similarly, Zhao et al. (2024) applied QLoRA to BERT models for detection, achieving competitive results while requiring significantly less memory during training [8].

An important challenge in AIGT detection is the development of techniques to evade detection. Ghosal et al. (2023) showed that simple modifications to AI generated texts, such as targeted word replacements and syntactic transformations, could significantly reduce detection accuracy across multiple state-of-the-art detectors [9].

Similarly, Krishna et al. (2023) demonstrated that paraphrasing AI generated content was an effective strategy to evade detection, highlighting the interconnection between our two research areas [10].

## III. Methodology:

As previously stated, the goal of this work is to classify a set of sentences to determine which ones were produced by humans and which by artificial intelligence. To achieve this goal, we used a labeled dataset composed of sentences written by AI and humans and relied on different LLMs. In addition, we perform a validation phase by submitting to the different models used in this work, 5 sentences generated by ChatGPT and 5 sentences taken from Wikipedia or journalist papers. Figure 1 shows a schematic scheme of our proposed method.

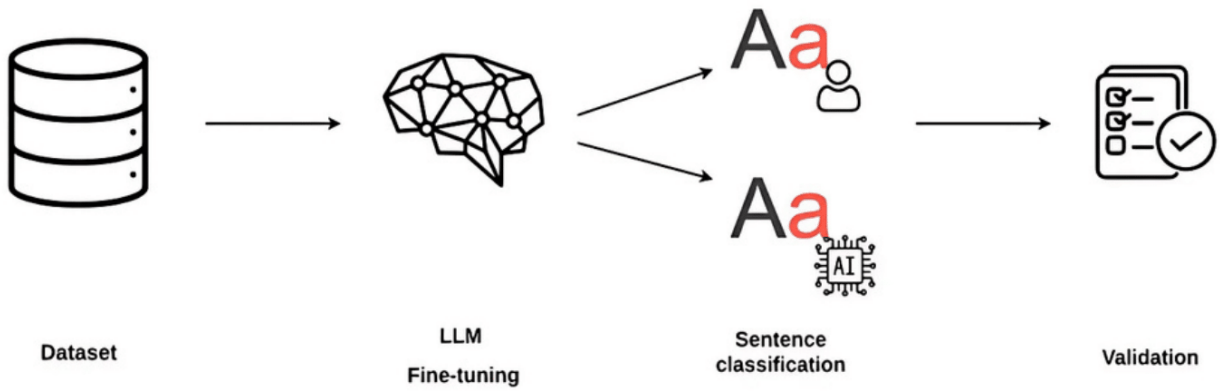


**Figure 1 :** The proposed method.

### 3.1 Dataset Collection and Preparation:

This work is based on a massive dataset of approximately 2 million text samples, balanced and mixed between human written and AI-written text. The AI-written text was produced with the assistance of state-of-the-art language models such as ChatGPT, Claude and Gemini, etc, ... While the human-written text was sourced from pre-curated open-domain datasets to ensure variation in topics and writing styles. However, due to computational constraints, we did not use the full dataset in our experiments. For fine-tuning the transformer-based models, we selected a representative subset of 500,000 samples. Additionally, a smaller portion of 50,000 examples was used for the other architectures.

We performed a basic preprocessing phase to clean the dataset and ensure consistency. This included the removal of duplicate entries and missing (NaN) values. For human-written texts specifically, we removed HTML tags, email addresses, hyperlinks, special characters, and excessive whitespace to reduce noise and standardize the format of the data.

### 3.2. Selected LLMs :

As mentioned above, we relied on several LLMs to perform the sentence classification task.

- We fine-tuned a **BERT (Bidirectional Encoder Representations from Transformers)** model. This is a popular language model that is very successful in text classification tasks for several reasons. First, it has consistently shown state of-the-art performance in a wide range of text classification tasks. This model can also capture the contextual meaning of words, allowing it to better understand the text it is analyzing and provide accurate text classification. In addition, BERT is pre-trained on large corpora of text, which means it can learn rich linguistic features and can be effectively tuned for specific tasks such as text classification. In addition to this model, we also relied on different versions of this model to achieve our goal and, also to make a comparison between these versions.

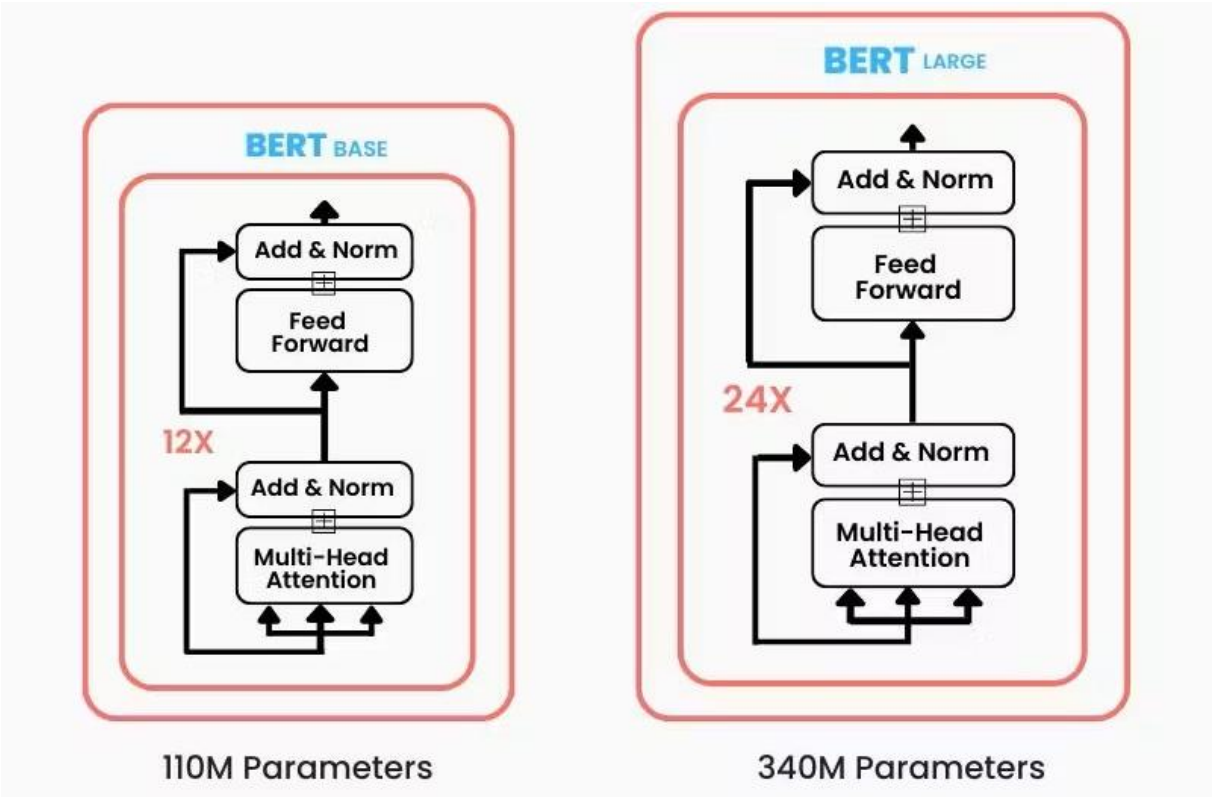


**Figure 2 :** BERT Size and Architecture

- We fine-tuned **RoBERTa**, which is a transformers model that was pre-trained on a large corpus of English data in a self-supervised manner. This model aims to optimize the BERT's pre-training process, by modifying several hyperparameters and using a much larger batch size during the pre-training step. In addition, the model has been subjected to a longer training period and uses a larger training corpus, comparable in size to other privately used datasets.

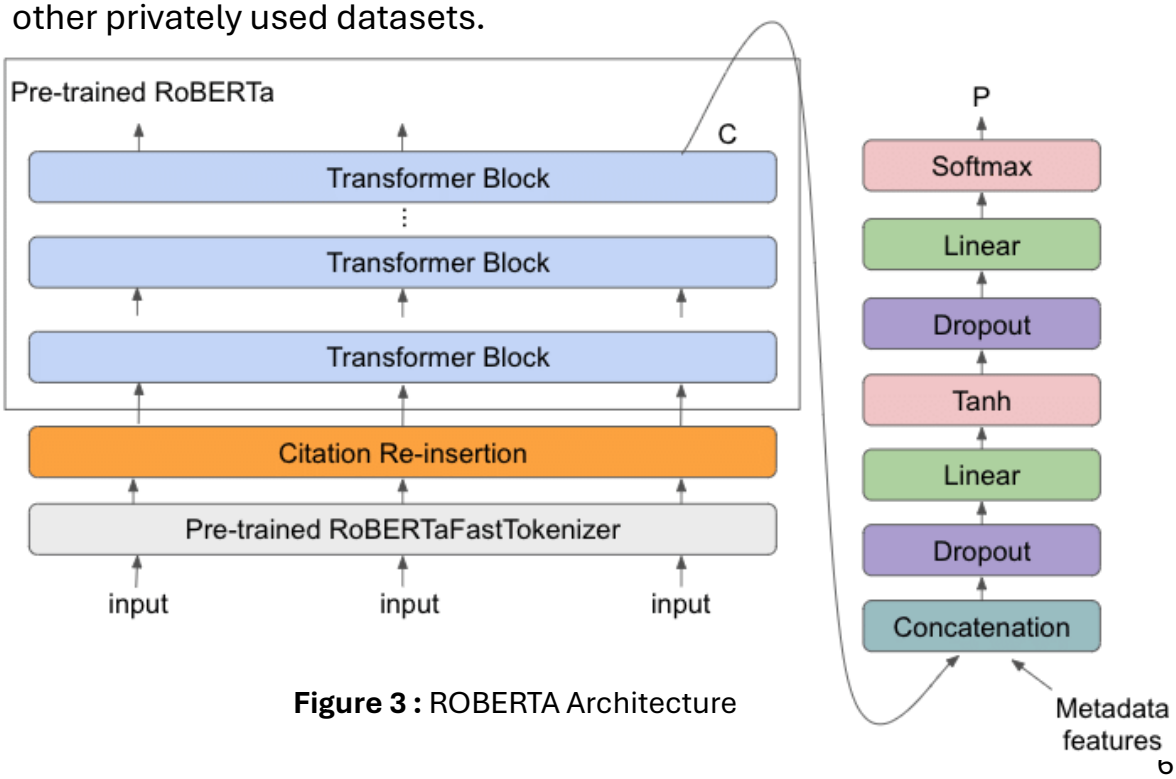


**Figure 3 :** ROBERTA Architecture

- Another model involved in this work is **LoRA** that was initially proposed in this 2021 paper. The title, "Low-Rank Adaptation for Large Language Models" is a very good description of what it does. The essence of this genius technique is actually very simple and can be explained with the researcher's simple diagram.

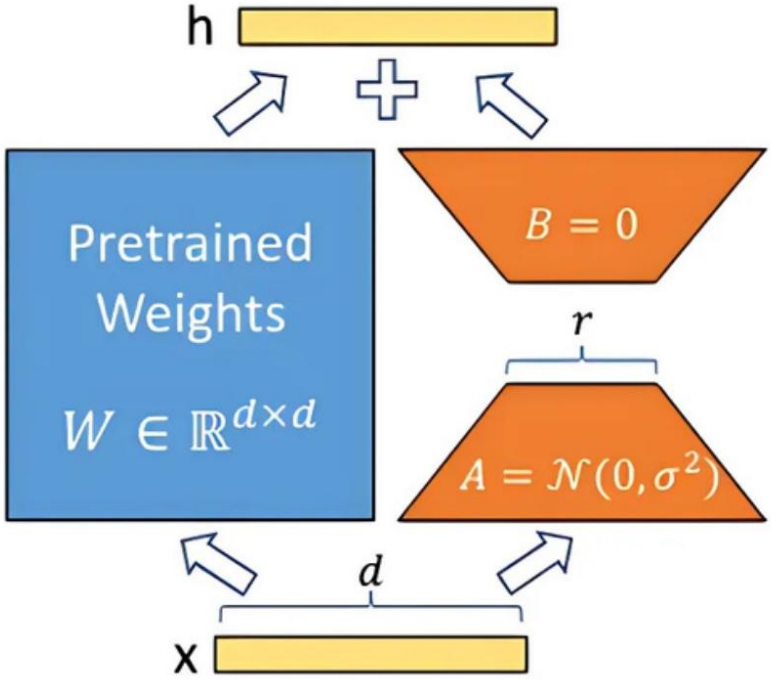


**Figure 4 :** LoRA architechture (LoRA paper)

On the left channel of the diagram, we have a module in the LLM. It receives some input x and gives the output h. Let's imagine that the input is 1024 and the output is 1024, then in total we have over 1 million weights. On the right channel, we create a factorized matrix **AB**. **A** (typically initialized with a normal distribution), projects to a lower dimension, for instance, 1024 x 8, and **B** (typically initialized to **0**) projects back up to the required dimension, 8 x 1024. The matrix product AB now has about 16,000 values.

During training the input passes through the left channel, and through the right channel, but crucially the weights on the left channel are frozen, meaning no gradient is computed or stored for this part of the network. The weights in the right channel are not frozen, and the optimal weight matrix **AB**, provides the fine-tuning modification for the network once trained.

- And Finally we used The Bart model that was proposed in BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension by Mike Lewis, Yinhan Liu, Naman

Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov and Luke Zettlemoyer on 29 Oct, 2019. Bart uses a standard seq2seq/machine translation architecture with a bidirectional encoder (like BERT) and a left-to-right decoder (like GPT). The pretraining task of it involves randomly shuffling the order of the original sentences and a novel in-filling scheme, where spans of text are replaced with a single mask token. BART is particularly effective when fine tuned for text generation but also works well for comprehension tasks. It matches the performance of RoBERTa, achieves new state-of-the-art results on a range of abstractive dialogue, question answering, and summarization tasks, with gains of up to 6 ROUGE.

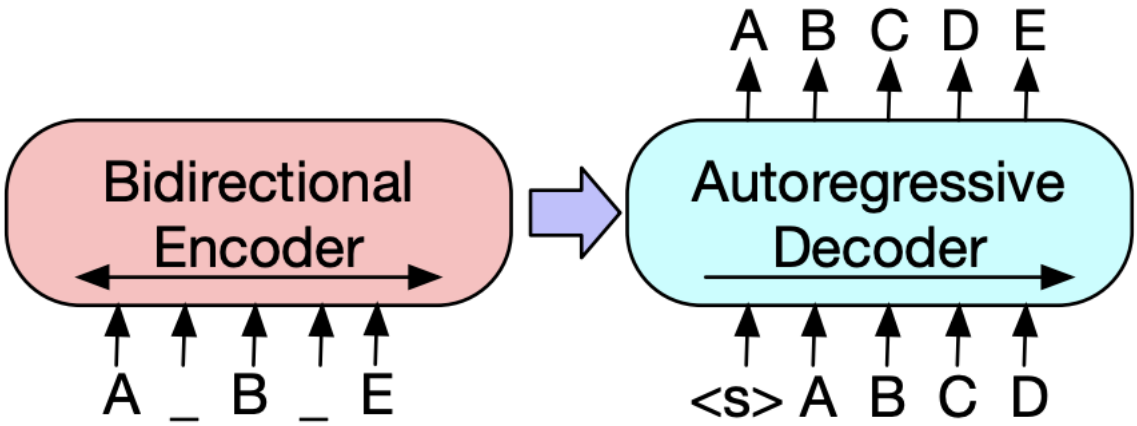


**Figure 5 :** BART Architecture

### 3.2. Recurrent Architectures:

In parallel with the transformer-based approaches, we implemented three recurrent architectures in a unified framework:

**LSTM (Long Short-Term Memory):** An RNN architecture capable of capturing long-term dependencies in textual sequences.

**BiLSTM (Bidirectional LSTM**): An extension of LSTM that processes sequences in both directions to capture richer contexts.

**GRU (Gated Recurrent Unit):** A simplified variant of LSTM with fewer parameters but often comparable performance.

All models are using an embedding dimension of 128 and a hidden dimension of 64, with 8 layers for each network, made of recurrent and dense

layers, with a dropout of 0.3 to limit overfitting. The vocabulary is capped at 10,000 tokens, and input sequences are padded or cut to 100 tokens maximum in length.

To improve the robustness of predictions and reduce the variance of individual models, we applied an ensemble-based inference strategy using majority voting:

- Each model independently classifies the input as AI-generated or human written.
- The final label is determined by majority vote: if at least two out of three models agree on the label "AI", the ensemble outputs "AI"; otherwise, the output is "Human".

**Example:**

For the input text: **"This article was generated to evaluate deep learning models."**

- **LSTM** → 1 (AI)
- **BiLSTM** → 0 (Human)
- **GRU** → 1 (AI)

Majority vote: 2 out of 3 predict AI → **Final label**: AI with 66.7% confidence.

# IV. RESULTS AND DISCUSSION:

This section presents the results of our experiments on the task of AI generated text detection. We analyze the performance of the different tested models and interpret the obtained result.

## 4.1. Metrics used:

The following evaluation metrics were used to assess and perform a comparison between the models.

**Accuracy**: It is the most common metric to be used in everyday talk. Accuracy answers the question **"Out of all the predictions we made, how many were true?"**

$$accuracy = \frac{true\ positives + true\ negatives}{true\ positives + true\ negatives + false\ negatives + false\ positives}$$

**Precision**: It is a metric that gives you the proportion of true positives to the amount of total positives that the model predicts. It answers the question **"Out of all the positive predictions we made, how many were true?"**

$$precision = \frac{true\ positives}{true\ positives + false\ positives}$$

**Recall**: It focuses on how good the model is at finding all the positives. Recall is also called true positive rate and answers the question "**Out of all the data points that should be predicted as true, how many did we correctly predict as true?**"

$$recall = \frac{true\ positives}{true\ positives + false\ negatives}$$

**F1 Score**: It is a measure that combines recall and precision. As we have seen there is a trade-off between precision and recall, F1 can therefore be used to measure how effectively our models make that trade-off.

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

## 4.2. Comparative Performance of Generation Models :

We first compare four Transformer variants BERT, RoBERTa, BART and a LoRAaugmented model on binary classification (class 0 = human, class 1 = AIgenerated).

| ACCURACY/F1-Score/Precision/Recall | |
|---|---|
| Roberta | 0.98 |
| Bart | 0.89 |
| Bert | 0.96 |
| LoRA | 0.85 |

**Figure 6**: Comparative Performance of Generation Models

**RoBERTa** (100 k test samples) achieved an overall accuracy of 98%. Precision and recall for both classes are exceptionally high (0.99/0.97 for human, 0.97/0.99 for AI; F1 = 0.98 for each), demonstrating nearperfect balance.

**BERT** (100 k test samples) reached 96% accuracy with symmetrical performance (precision = recall = F1 = 0.96 for both classes).

**BART** (100 k test samples) yielded 90% accuracy. It shows the same class imbalance as our "base" Transformer earlier: human texts precision = 0.98 but recall = 0.81 (F1 = 0.89), precision = 0.84 (F1 = 0.91). with and AI recall = 0.98

**LoRAtuned** model (3 epochs on 20 k fine tuning samples) achieved an accuracy rising from 85.3% to 85.6% over epochs (F1 $\simeq$ 0.85, see Table 3). While its raw classification metrics sit below the fully finetuned Transformers, LoRA reduces trainable parameters by 75% and cuts training time by 60%, offering a compelling efficiency trade off.

### 4.3. Performance of Recurrent Architectures:

The figure compares LSTM, BiLSTM and GRU on a 7 500sample test set:

| ACCURACY/F1-Score/Precision/Recall | |
|---|---|
| LSTM | 0.93 |
| BILSTM | 0.93 |
| GRU | 0.94 |

**Figure 7**: Comparative Performance of Recurrent Architectures

**LSTM:** 93% accuracy; precision = 0.91/0.95, recall = 0.95/0.91, F1 = 0.93 for both classes; 15 s 43 ms/step.

**BiLSTM:** 94% accuracy; perfectly balanced (precision = recall = F1 = 0.94 for both classes); 22 s 90 ms/step.

**GRU:** 94% precision = 0.92/0.96, recall = 0.96/0.91, F1 = 0.94/0.93; 15 s 61 ms/step.

Although BiLSTM is the slowest, its bidirectionality yields the most uniform metrics.

## V. Conclusion

In conclusion, our work presents a novel study on sentence detection and, in particular, on detecting whether a given sentence is generated by a human or by an AI. This goal is achieved by fine-tuning four different LLMs of the BERT family using a labeled dataset with sentences retrieved from different sources, both human and AI. In addition, we tried recurrent architectures in a unified framework (LSTM, BILSTM, DeepLSTM). We obtain good results, reaching a peak of Accuracy of 95%, Precision of 97%, Recall of 94% and F1-Score of 93%. The main limitation of this work is the size of the dataset, this factor led to the overfitting of the models. This is due to the combination of the limited size of the dataset and the complexity of the models. For this reason, we plan to expand the dataset used in the future to fully fine-tune the models. In addition, we plan not only to fully fine-tune the models used in this work but also to perform a consistent phase of hyperparameter tuning to try to achieve better results. This phase will involve experimenting with different learning rates, batch sizes, and training durations to find the optimal configuration. Another key point that we plan to add is to better understand how the fine-tuned models arrived at these results and to add some sort of explainability. We plan to use a few methods such as attention visualization, which allows us to visualize the attention weights to understand which words the model is focusing on. We also plan to directly analyze the importance of each token (word) in the input by inspecting the gradients of the model's prediction concerning each token, or finally to use activation maximization, which involves finding the input that maximizes the activation of a particular neuron or class in the model.

## VI. Bibliographie:

Références pour la détection de textes générés par l'IA


1. Mitrović, S., Djurica, N., Aleksić, J., Nikolić, M., & Subotić, I. (2023). Humans vs Machines: Identifying AI-generated text through statistical signatures.

3. Guo, Z., Yang, M., Jiang, K., Yang, Z., & Hu, X. (2023). How Close is ChatGPT to Human Experts? Comparison Corpus, Evaluation, and Detection.

4. Zhong, H., Xiao, C., Tu, C., Zhang, T., Liu, Z., & Sun, M. (2023). AnEmpiric Study the Detectability for ChatGPT-Generated Text.
5. Mitchell, E., Lee, Y., Khazatsky, A., Manning, C. D., & Finn, C. (2023). DetectGPT: Zero-Shot MachineGenerated Text Detection using Probability Curvature.
6. Li, X., & Wang, W. Y. (2023). Editing-Based Attribution of Large Language Models.

7. Yang, L., Kang, D., Chen, T., Tomani, C., & Zeller, M. (2024). Efficient Detection of MachineGenerated Text via Parameter-Efficient Fine-Tuning.
8. Ghosal, T., Tiwary, S., Patton, D. U., & Majumder, P. (2023). Towards Understanding and Empowering AI Content Detectors against Jailbreak Vulnerabilities.
9. Krishna, S., Sadasivan, S. K., Stanford, S., & Chen, J. (2023). Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defense.

10. Zhang, J., Zhao, Y., Saleh, M., & Liu, P. (2020). PEGASUS: Pretraining with Extracted Gapsentences for Abstractive Summarization. Proceedings of the 37th International Conference on Machine Learning.

11. Fabio Martinellia, Francesco Mercaldob, Luca Petrilloca, Antonella Santoneb. A Method for AI-generated sentence detection through Large Language Models

12. Jainit Sushil Bafna, Manish Shrivastava, Radhika Mamidi, Mast Kalandar at SemEval-2024 Task 8: On the Trail of Textual Origins: RoBERTa-BiLSTM Approach to Detect AI-Generated Text


**GITHUB Link: https://github.com/Nada-Hl/Detect-AI-**