Team #26- Power$^{\text{Set}}$
AI-Generated Content Detection

Anosh Kurian Vadakkeparampil; 40303184; anosh.k.v@gmail.com
and
Oven Van Esbroek; 40073056; mofvanes@gmail.com

December 30, 2024

Abstract. In an era where AI-generated content is increasingly hard to distinguish from human-written text, this project aims to develop a transformer-based model to effectively detect AI-generated work. Leveraging three diverse datasets, we evaluated multiple models, including Naive Bayes, and three transformer-based classifiers. While Naive Bayes performed decently in finding surface-level features, transformer-based models showed superior potential in leveraging contextual embeddings. However, challenges such as detecting hybrid texts revealed the models' inability.

Contents

1 Goal of the project

In this project, our team set out to develop a robust transformer model capable of effectively distinguishing AI-generated content from authentic human-created work. As AI-generated content becomes not much different from human, this effort addresses a critical need to maintain trust and authenticity in digital communication.

Our goal is to test and compare different high-performance models, and to also test their effectiveness against possible countermeasures.

2 Methodology

Over the course of this project, we utilized three distinct datasets to ensure the validity and robustness of our experimental results. These datasets are as follows:

- GenAI Content Detection Dataset (English Texts): Comprising 2,096 training examples and 1,626 testing examples[3].
- GenAI Content Detection Dataset (Arabic Texts): Comprising 2,070 training examples and 481 testing examples[3].
- Subset of the DAIGT Dataset: Including 3,000 examples for training and 1,000 examples for testing[2].

To evaluate and improve performance, we conducted multiple iterations of model development, analyzing their effectiveness and relative advantages. Two primary classes of models were employed, a basic Bag-of-Words(BoW) type model and transformer-based deep learning models.

The following section outlines the specific models utilized in our project.

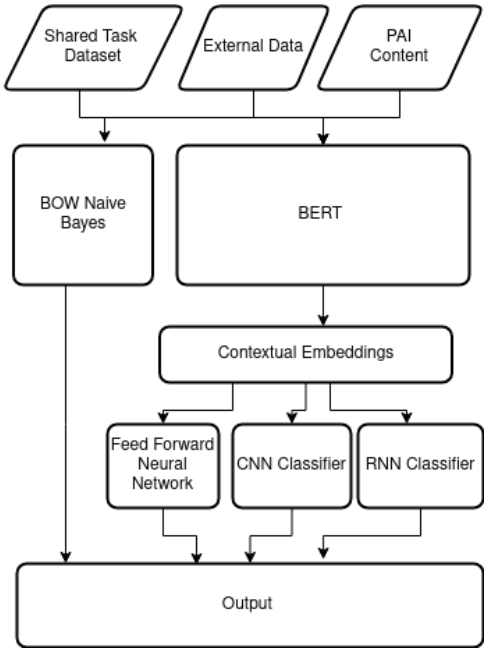
- **Naive Bayes**: This model largely stands as a baseline to help understand and highlight the performance difference that is gained by the more advanced deep learning model.
- **Bert**: We use hugging face's transformers library for using pretrained.
  - **NN**: The default head for `BertForSequenceClassification`, which uses a simple dense layer over the sentence embedding. [1]
  - **CNN**: Custom Model with a 1D CNN betwee the word embeddings and the classification layer.

– **LSTM**: Custom Model using a LSTM RNN between the word embeddings and the classification layer.

To conduct the Bert training, we used the bert-base-cased weights for the English and DAIGT dataset tokenizers, and bert-base-multilingual-cased weights for the Arabic dataset tokenizers

We used a modified Adam algorithm named AdamW, that modifies the weight decay step to achieve better performance.[5]. We set an initial learning rate of 5×10^{-5}, had a batch size of 16, and trained for 20 epochs.

For the CNN and LSTM, to improve training stability, we also clipped the gradient values.



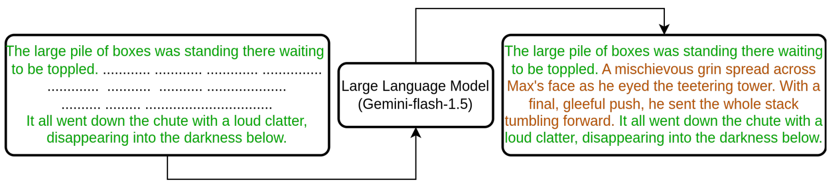
[4]

3 Countermeasures

We explored the concept of using partially human-written and partially AI-generated data for training. Although this approach was beyond our current capacity to fully implement, we opted to test such cases against our trained models instead.

To simulate this scenario, we generated 754 essays by prompting a large language model (LLM) to produce content that incorporated specific sentences from the human-written English essays. This method aimed to mimic

the cases where AI-generated text is subtly blended with human input. The resulting essays were then evaluated to determine whether our models could still effectively identify the predominantly AI-generated nature of the content.



[4]

3.1 Results

Classifier	Language	Precision	Recall	F_1-Score
Naive Bayes	English	0.56	1.00	0.72
	Arabic	0.72	1.00	0.83
	DGAIT	0.96	0.99	0.98
FFNN	English	0.95	0.90	0.93
	Arabic	1.00	0.87	0.93
	DGAIT	0.95	0.98	0.97
CNN	English	1.00	0.80	0.89
	Arabic	1.00	0.85	0.92
	DGAIT	0.94	0.98	0.96
RNN	English	0.99	0.87	0.93
	Arabic	1.00	0.88	0.93
	DGAIT	0.95	0.97	0.96

Table 1: Consolidated metrics for classifiers across languages and models, with AI class

3.2 Analysis

- Effectiveness of Naive Bayes: The Naive Bayes classifier exceeded expectations, likely due to its ability to leverage simple yet telling mark-

	FFNN	CNN	RNN
Detection rate	14/754	1/754	6/754

Table 2: Model Performance on AI-Expanded Essays

ers such as irregular punctuation or errant capitalization. While preprocessing and tokenization often discard these features, their presence strongly indicates human authorship when retained. This performance highlights how even basic models can effectively exploit surface-level patterns in distinguishing human- and AI-generated text.

- Comparison of CNN and RNN Heads on BERT: The CNN classification head showed a slight performance edge over the RNN head, with both performing on par with standard BERT classification. This outcome suggests that these architectures capitalize on superficial patterns, much like Naive Bayes, rather than fully leveraging the rich contextual embeddings that BERT provides. The CNN's ability to capture localized features may give it an advantage in identifying distinctive yet shallow markers of authorship. However, this approach under-utilizes the nuanced semantic understanding that transformers like BERT are designed to offer.

- Challenges in Detecting Hybrid Texts: All models faced significant difficulties when tasked with detecting hybrid texts where AI expanded on human-written sentences. Such integration effectively conceals the surface-level indicators that the models predominantly rely on, rendering them less effective in these cases. This limitation reveals a critical gap in the models' ability to perform deeper semantic analysis and contextual reasoning. Bridging this gap would require developing techniques that go beyond surface patterns, potentially incorporating more advanced contextual or generative adversarial strategies to detect the subtle interplay between human and AI inputs.

4 Role of each team member

Both team members collaborated effectively to research, and ideated in tandem and finally contributed to the project code and write-ups equally.

5 Limitations

- Binary Assumption of Text Origins: The datasets used in this study inherently assume a binary distinction between *Human* and *AI* text. However, in real-world scenarios, text often exists on a spectrum, for instance, a human-written essay augmented with AI-generated sugges-

tions and corrections. This binary framing limits the models' ability to handle hybrid content effectively.

- Linguistic Similarities in Short Essays: For shorter texts, the differences between well-written human essays and generated essays can become negligible. In such cases, linguistic markers are often insufficient for reliable classification, resulting in diminished model performance.

- Resource Constraints: Our ability to train and optimize the model was restricted by limited computational resources. This limitation hindered the scope for iterative model improvements and prevented us from fully exploring the potential of more computationally demanding approaches. Additionally, unrestricted access to state-of-the-art large language models (LLMs) for dataset generation was beyond our capacity, limiting the diversity and representativeness of our datasets.

6 Future Work

- Exploring Tokenizer Dependencies: Investigate whether the choice of tokenizer significantly impacts classification effectiveness, particularly for detecting hybrid or paraphrased AI-generated content. Tokenization strategies tailored to this task might yield better results.

- Evaluating Alternative Classification Heads: Beyond the CNN and RNN heads tested, explore other classification head architectures, such as attention-based heads or hybrid models that combine multiple strategies, to enhance performance and capture deeper contextual nuances.

- Incorporating Hybrid Datasets: Generate and train models on datasets that include hybrid texts—content created collaboratively by humans and AI. This approach would better simulate real-world scenarios and push the boundaries of what these models can detect.

- Resource Scaling and Advanced Model Training: With greater computational resources, explore training more complex models or fine-tuning existing state-of-the-art LLMs to address nuanced challenges in text classification. Collaborative access to cutting-edge LLMs for dataset generation could further enrich the dataset quality and diversity.

7 Difference with your original proposal

Our initial proposal had us using DistilBert, but we ended up being capable of using the full BERT model. We added the DAIGT dataset to confirm our results with another dataset. We also explore partial authorship content as an avenue, and have uncovered concrete ideas for future research towards more concrete means of detection.

8 Conclusions

This project aimed to develop a transformer-based model to distinguish between human-written and AI-generated text. By using diverse datasets and iterating on model designs, we achieved our goal, however challenges like hybrid texts and highlight room for model improvement.

Our findings emphasize the need for advanced architectures and hybrid datasets to better handle nuanced cases. Future work should explore tokenizer dependencies, alternative classification heads, and richer training data to enhance detection capabilities.

9 References

References

[1] Bertforsequenceclassification source code. Accessed 12-12-2024.

[2] Detect AI Generated Text Training Dataset. Kaggle repository.

[3] GenAI Content Detection Dataset. Academic Essay Authenticity Challenge. GitLab repository. Consent form must be signed for access to the dataset.

[4] Google AI. Google Gemini API Quickstart. Google AI Developer Documentation. Python SDK usage instructions.

[5] Loshchilov, I., and Hutter, F. Fixing weight decay regularization in adam. *CoRR abs/1711.05101* (2017).