# AI Text Detection: Comparing Classical Machine Learning and Transformer Approaches

# Introduction

The increase in use of AI text generation tools like ChatGPT, has created challenges in the academic industry where students might tend to use these tools to complete assignments instead of doing the work themselves. This makes it crucial to have a reliable techniques, methods or tools for detecting text that is ai generated. My project compared how traditional machine learning approaches work against modern transformer based methods for detecting AI generated text.

The main challenge was that the modern language models generate very human – like text, which makes such detections difficult. In this project, I experimented to answer one question, that is do we need expensive transformer models for detecting ai generated text, or simpler classical methods also perform or competes at the same level?

I tested four different approaches:-

* TF – IDF with Naïve Bayes classifier
* TF-IDF with Logistic Regression Classifier
* Fine-tuned DistilBERT transformer
* DistilBERT with LoRA (parameter-efficient training)

This comparison is important because classical methods are very fast and cost efficient to use, as compared to the transformers which are tend to be more accurate but require costly computational resources to run.

## Methods

**Dataset and Setup: -** I used a labelled dataset containing 487,235 text samples which includes both human-written and AI-generated content. In this labelled dataset 0 represents human written text and 1 represents AI text. This was an imbalanced dataset with around 305,797 human samples and 181,438 AI samples, a ratio of 63% : 37%. A 80/20 train-test split is used to maintain class balance across both sets.

## Classical Machine Learning Baselines

### TF-IDF Feature Extraction

I used Term Frequency-Inverse Document Frequency (TF-IDF) to convert text into the numerical features. In this method important words are identified by assigning higher weights to terms that appear more often in a document but are rare across the entire collection.

### Naïve Bayes Classifier

The first model I used combined TF-IDF features with Multinomial Naive Bayes. This is the type of classifier which perceives words independently with the given class label, which is AI or Human. This classifier works well for text classification tasks even though it assumes words appear independently.

## Transformer Based Approaches

### DistilBERT Fine-tuning

DistilBert is the transformer model of choice because it is 60% smaller than BERT while maintain 97% of its capabilities. Started from the pre-trined distilBert-base uncased model, a classification layer for binary prediction was trained, and all parameters were fine tuned using our dataset.

Training was done using these settings: - learning rate 2e-5, batch size 16, and 3 epochs. The model processes the text by breaking it into smaller subwords units, then passes them through transformer layers to generate contextual representations for classification.

### DistilBERT with LoRA

The final approach used LoRa (Low-Rank Adaptation) for parameter – efficient fine-tuning. In this technique, we do not need to update all model weights, it freezes the original parameters, so we only need to train small additional matrices. I applied LoRa to the query and value projection layers with rank =8 and alpha = 32, which significantly reduces trainable parameters while maintaining performance.

### Evaluation

All models were evaluated using the accuracy, precision, recall and F1-Score on the same test set. I also performed some sort of error analysis to analyze or understand what types of text make different models to fail on predictions.

## Results

The experiments described above revealed important differences in both performance and computational requirements across the four approaches.

## Model Performance Comparison Table: -

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Precision** | **Recall** | **F1-Score** | **Training Time** |
| TF-IDF + Naïve Bayes | 96% | 96% | 96% | 96% | ~ 2-3 minutes |
| TF-IDF + Logistic Regression | 99% | 99% | 99% | 99% | ~ 3 minutes |
| DistilBERT (Full) | 99.92% | 99.92% | 99.92% | 99.92% | ~ 64 minutes |
| DistilBERT + LoRA | 99.94% | 99.94% | 99.94% | 99.94% | ~ 48 minutes |

## Key Findings

* **Best Overall Performance:** - Surprisingly, DistilBERT with LoRA achieved the highest accuracy at 99.94%. Its accuracy is slightly higher than full-fine tuning while it is more computationally efficient.
* **Excellent Classical Performance: -** TF-IDF with Logistic Regression performed remarkably well at 99% accuracy, which is significantly higher than Naïve Bayes (96%), and is very close to the transformer approaches.
* **LoRA Superiority:** -LoRA not only achieved the best accuracy (99.94%) but also trained around 25% faster than full fine-tuning. It also used fewer trainable parameters. This was the most surprising finding for this task.
* **Speed vs Accuracy Trade-Off:** Classical methods trained 15-20x faster but also showed a 3-4% gap in performance if we compare them to transformers.

# Analysis and Discussion

## Error Analysis

I also conducted detailed error analysis on the best performing model’s misclassifications, which is LoRA:

* **Total Errors:-** only 62 misclassified examples out of 97,447 test samples.
* **False Positives:-** 39 cases where human written text was labeled as AI-generated
* **False Negatives:** 23 cases where AI text was labeled as human-written

The model demonstrated bias towards predicting AI-generated text [False Positives], which can be very problematic in educational contexts where false cases of academic integrity violations might be very unfair for students.

## Pattern Analysis

My analysis of misclassified texts showed several interesting patterns:

1. **Length Variations:** Both very short and very long texts were more likely to be misclassified.
2. **Domain Specificity:** Model struggled in Technical or highly specialized content
3. **Writing Style:** Formal vs Informal writing patterns tend to confuse the model

## Performance Trade-Offs

For AI text detection applications, the choice between accuracy and computational efficiency depends on the specific use case:

* **High-Stakes Applications –** LoRA provides the best accuracy with the benefit of being computational requirements.
* **Resource-Constrained Environments-** TF-IDF + Logistic Regression offers excellent performance (99%), which is just a little less from transformer models, with benefit of training on very minimal computational requirements.
* **Balance:** LoRA emerges as the winner, which provides top performance with efficiency.

# Ethical Considerations

My work as described raises important ethical questions about AI Text detection in educational settings:

**False Positive Impact:** When human written text is falsely labelled as AI generated text, students might have to face unfair academic consequences. The analysis as described above shows that even the best model has a 0.06% false positive rate, this means 6 out of 1000 students would be falsely accused, this could affect innocent students.

**Bias Against Non-Native Speakers:** AI detection systems may discriminate against students whose writing patterns differ from native speakers in the training data, which can potentially flag legitimate work as AI-generated.

Detection Arms Race: - As detection systems improve, AI writing tools might also evolve to erase those patterns in the text that can be detected, which can create a never ending technological competition ignoring the underlying educational challenges.

Privacy Concerns: Text Analysis for AI detection requires processing student writings, which might raise concerns of institutions or students about data privacy and institutional surveillance.

Over-reliance on Technology – Automated detection should aid or help, but not replace human judgement in evaluating student’s work and addressing academic integrity concerns.

## Conclusion

This study shows that both classical and transformer-based approaches can achieve high performance for AI text detection, but have their significant trade-offs. Most surprisingly, LoRA achieved the highest accuracy(99.94%) while also being computationally efficient than full fine tuning, making it the recommended approach.

TF-IDF with Logistic Regression also provided surprisingly competitive results (99%), with very minimal computational requirements. This makes it suitable for institutions with limited resources.

In conclusion, I recommend that educational institutions should consider LoRA based approaches for AI detection systems when computational resources are available, otherwise classical methods remain as reliable alternatives.

## References

1. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.
2. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv:1810.04805.
3. Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108.
4. Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., ... & Chen, W. (2021). LoRA: Low-Rank Adaptation of Large Language Models. arXiv preprint arXiv:2106.09685.
5. Manning, C., Raghavan, P., & Schütze, H. (2008). Introduction to information retrieval. Cambridge university press.
6. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. Journal of machine learning research, 12(Oct), 2825-2830.
7. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... & Rush, A. M. (2020). Transformers: State-of-the-art natural language processing. In Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations (pp. 38-45).
8. Kenton, J. D. M. W. C., & Toutanova, L. K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of NAACL-HLT (pp. 4171-4186).
9. Rish, I. (2001). An empirical study of the naive Bayes classifier. In IJCAI 2001 workshop on empirical methods in artificial intelligence (Vol. 3, No. 22, pp. 41-46).
10. Hosmer Jr, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). Applied logistic regression (Vol. 398). John Wiley & Sons.
11. Rogers, A., Kovaleva, O., & Rumshisky, A. (2020). A primer in BERTology: What we know about how BERT works. Transactions of the Association for Computational Linguistics, 8, 842-866.
12. Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N., & Huang, X. (2020). Pre-trained models for natural language processing: A survey. Science China Technological Sciences, 63(10), 1872-1897.
13. Liu, X., He, P., Chen, W., & Gao, J. (2019). Multi-task deep neural networks for natural language understanding. arXiv preprint arXiv:1901.11504.
14. Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2023). QLoRA: Efficient Finetuning of Quantized LLMs. arXiv preprint arXiv:2305.14314.
15. Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., ... & Gelly, S. (2019). Parameter-efficient transfer learning for NLP. In International Conference on Machine Learning (pp. 2790-2799). PMLR.
16. Mitchell, E., Lee, Y., Khazatsky, A., Manning, C. D., & Finn, C. (2023). DetectGPT: Zero-shot machine-generated text detection using probability curvature. arXiv preprint arXiv:2301.11305.
17. Su, J., Xu, J., Qiu, X., & Huang, X. (2022). RoFormer: Enhanced transformer with rotary position embedding. Neurocomputing, 568, 127063.
18. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI blog, 1(8), 9.
19. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. Advances in neural information processing systems, 33, 1877-1901.
20. Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., & Artzi, Y. (2019). BERTScore: Evaluating text generation with BERT. arXiv preprint arXiv:1904.09675.
21. Scikit-learn Development Team. (2023). TF-IDF Vectorizer Documentation. Retrieved from <https://scikit-learn.org/stable/modules/feature_extraction.html>
22. Scikit-learn Development Team. (2023). Naive Bayes Documentation. Retrieved from <https://scikit-learn.org/stable/modules/naive_bayes.html>
23. Scikit-learn Development Team. (2023). Logistic Regression Documentation. Retrieved from <https://scikit-learn.org/stable/modules/linear_model.html>
24. Hugging Face Team. (2023). Trainer API Documentation. Retrieved from <https://huggingface.co/docs/transformers/main_classes/trainer>
25. PEFT Documentation. (2023). Parameter-Efficient Fine-Tuning. Retrieved from <https://github.com/huggingface/peft>