Detection of Machine-Generated Text using BERT Models

Adarsh Yadav
Chemical Engineering, 4th Year
Indian Institute of Technology Roorkee
Enrollment No: 22112007
GitHub: codebreaker0001

August 12, 2025

Contents

1 Introduction

With the advancement of large language models (LLMs), the boundary between AI-generated and human-written content has become increasingly blurred. This raises significant concerns regarding authenticity, trustworthiness, and ethical use. The goal of this project is to utilize **BERT** (Bidirectional Encoder Representations from Transformers) to identify whether a given text is authored by a human or generated by an AI model. The emphasis lies in maintaining content integrity while embracing the benefits of AI.

2 Workflow

The workflow adopted for this project consists of:

1. **Preprocessing:** Removing stop words, punctuation, special characters, and non-alphabetic terms. Tokenization and input formatting are handled using the BERT preprocessing module.
2. **Dataset Expansion:** Additional publicly available datasets were aggregated from multiple sources, expanding the dataset size from 1,378 to roughly 50,000 entries. This diversity aids in capturing different writing styles.
3. **Model Training:** A `bert-base-uncased` model was used for sequence classification. Although `bert-large-cased` could offer deeper analysis, it demands more computational resources.
4. **Prediction:** The trained model evaluates unseen data and labels potential AI-generated content.
5. **Result Storage:** Outputs are stored in CSV format for further analysis or contest submission.

3 Detailed Explanation of BERT's Role

BERT is designed to understand contextual relationships between words by considering both left and right context simultaneously. In detecting AI-generated text, it captures:

- Semantic structures and coherence.
- Vocabulary usage frequency and diversity.
- Grammatical tendencies and stylistic patterns.
- Emotional tone variations.

3.1 Key Questions Explored

1. How do black-box and white-box detection methods differ, and which applies here?
2. What role do watermark embeddings play in text detection?
3. Why is AI hallucination a problem, and how can it be mitigated?
4. What pipeline structure best supports robust classification?
5. Which edge cases should be considered in model evaluation?

4 Edge Cases

During testing, some text samples exhibited deliberate typographical errors, unusual grammar, or rare word usage. These can confuse models trained solely on clean datasets. Documenting such scenarios helps refine the detection approach.

5 Tips and Recommendations

- Use `bert-base` for balanced performance; `bert-large` may be impractical under strict runtime constraints.
- Maintain consistent dataset paths to prevent execution errors.
- Favor BERT-specific preprocessing pipelines for better compatibility.
- Experiment with hyperparameters such as learning rate, optimizer, and batch size for improved results.

6 Variations and Experiments

- Smaller batch sizes improved offline accuracy but sometimes reduced leaderboard scores.
- The Adam optimizer generally outperformed alternatives in this task.
- SGD provided better generalization in some cases but at the cost of peak accuracy.

7 Notable Observations

Based on linguistic comparisons:

1. Human-written texts typically have more varied vocabulary and unique word usage.

2. AI-generated texts often favor consistent sentence structures with less emotional expression.
3. LLM outputs may repeat certain phrases more frequently.
4. Machine-generated sentences can be syntactically complex yet lexically simple.
5. Typos and irregularities are uncommon in AI output unless artificially inserted.

8 CV vs LB Score Discrepancies

Cross-validation scores tend to be higher than leaderboard scores due to mismatches between public datasets and hidden evaluation sets. Hidden datasets may contain AI-generated text modified with human-like imperfections, making detection harder.

9 Analysis of the Discrepancy

- AI typically produces near-perfect spelling without explicit instructions to add errors.
- Human-like noise is sometimes deliberately added to evaluation datasets to challenge detection methods.
- Overfitting models to leaderboard-specific patterns reduces generalizability.

10 Improving Leaderboard Scores

While deep transformers like BERT are powerful, simpler TF-IDF models combined with ensemble classifiers (Logistic Regression and SGDClassifier) achieved competitive scores with shorter runtimes. Introducing noise during training can further improve robustness.

11 Result Summary

1. Transformers struggle when input text has heavy spelling errors.
2. TF-IDF remains highly effective for stylistic classification tasks.
3. Adding noise to training data improves adaptability.
4. Ensembling models increases accuracy when base models exceed 50% performance.

12 References

- A Survey on LLM-Generated Text Detection: Necessity, Methods, and Future Directions. Available at: https://aclanthology.org/2025.cl-1.8.pdf
- Zero-Shot Detection of LLM-Generated Text using Token Cohesiveness. Available at: https://arxiv.org/pdf/2409.16914
- LLM4DV: Using Large Language Models for Hardware Test Stimuli Generation. Available at: https://arxiv.org/pdf/2310.04535
- Implementing BERT and Fine-Tuned RoBERTa to Detect AI-Generated News by ChatGPT. Available at: https://arxiv.org/pdf/2306.07401