Final Report: AI-Generated Text Classification

SEA 820 NLP Final Project
Name : Kannav Sethi
Course: Final-Year Undergraduate NLP

Executive Summary

This report presents a comprehensive analysis of developing machine learning models to classify AI-generated versus human-generated text. While achieving high performance metrics (99.97% accuracy), My investigation reveals critical overfitting issues and concerning classification patterns that highlight the complexity of this task and its ethical implications.

---

1. Introduction and Problem Statement

1.1 Objective

The primary goal was to develop a robust classifier capable of distinguishing between AI-generated and human-generated text.
This task has become increasingly important as AI text generation tools become more sophisticated and widespread.

1.2 Dataset Overview

- **Source**: AI vs Human Text dataset from Kaggle
- **Size**: 487,235 text samples
- **Distribution**: Binary classification (AI-generated vs Human-generated)
- **Features**: Raw text content with binary labels

---

2. Methodology

2.1 Experimental Design

I had implemented a three-tier approach to model development:

1. **Baseline Traditional Models**: Logistic Regression and Support Vector Machine, I didn't want to use Naive Bayes Model, just because the dataset is too much to go about.
2. **Advanced Deep Learning**: Fine-tuned DistilBERT transformer model, specifically `distillbert-base-uncased`
3. **Error Analysis**: Systematic investigation of misclassification patterns

2.2 Data Preprocessing

2.2.1 Baseline Models

- **Text Normalization**: Converted to lowercase
- **Punctuation Removal**: Stripped punctuation marks
- **Stop Word Removal**: Eliminated common English stop words
- **Feature Extraction**: TF-IDF vectorization with:
  - N-gram range: (1, 2)
  - Minimum document frequency: 2
  - Maximum document frequency: 0.9
  - Maximum features: 20,000

2.2.2 DistilBERT Model

- **Tokenization**: Used DistilBERT tokenizer with padding and truncation
- **Data Splitting**: 80% training, 10% validation, 10% testing
- **Preprocessing**: Minimal preprocessing to preserve linguistic nuances

2.3 Model Architectures

2.3.1 Baseline Models

1. **Logistic Regression**

   - Maximum iterations: 1000
   - Default scikit-learn parameters

2. **Support Vector Machine**

   - Linear kernel (LinearSVC)
   - Default regularization parameters

2.3.2 Fine-tuned DistilBERT

- **Base Model**: distilbert-base-uncased
- **Architecture**: Added classification head for binary classification
- **Training Configuration**:
    - Learning rate: 2e-5
    - Batch size: 16 (train and eval)
    - Epochs: 3
    - Weight decay: 0.01
    - Optimizer: AdamW (default)

2.4 Training Strategy

- **Evaluation Strategy**: Per-epoch evaluation
- **Model Selection**: Best model based on validation performance
- **Early Stopping**: Load best model at end of training
- **Metrics**: Accuracy, F1-score, Precision, Recall

3. Results and Performance Analysis

3.1 Quantitative Results

Model	Accuracy	F1-Score	Precision	Recall
Logistic Regression	0.9949	0.9931	0.9969	0.9893
SVM	0.9988	0.9984	0.9996	0.9972
DistilBERT (Fine-tuned)	**0.9997**	**0.9996**	**0.9996**	**0.9997**

As can be seen from the evaluation, I saw that DistilBERT was better than all of those.

3.2 Performance Interpretation

The fine-tuned DistilBERT model achieved state-of-the-art performance metrics, showing marginal but consistent improvements over traditional baseline models.
However, I did find one major issue, which was overfitting to the dataset.

4. Comprehensive Error Analysis

4.1 Classification Pattern Investigation

So I made a file called as `misclassification_analysis.py` , whose task was to take the model, give me predictions based on several outlier cases

I chose some test cases, the reason for choosing such ones is that it was designed keeping in mind to test whether the model learned genuine linguistic differences or merely memorized superficial statistical patterns

4.1.1 Short Text Bias

Test Cases:

- 'Yes.' → AI-generated (confidence: 0.998)
- 'I agree.' → AI-generated (confidence: 1.000)
- 'Thanks!' → AI-generated (confidence: 1.000)

Analysis:
What I discovered was that the model consistently misclassifies short, simple human expressions as AI-generated, suggesting it has learned to associate brevity with artificial generation, again this entails to overfitting of the model.

4.1.2 False Positives: Formal Language Bias

Test Cases:

- 'I believe this approach would be beneficial for achieving optimal results.' → AI-generated (confidence: 1.000)
- 'Furthermore, we should consider the implications of this decision.' → AI-generated (confidence: 1.000)
- 'It is important to note that various factors contribute to this outcome.' → AI-generated (confidence: 1.000)

Analysis: The model incorrectly flags formal, academic, or professional language as AI-generated. This reveals a dangerous bias against:

- Academic writing styles
- Professional communication
- Non-native speakers who may use more formal constructions

I think normally this shouldn't happen, but since it overfit, its treating every other line as AI-Generated

4.1.3 False Negatives: Informal Language Misclassification

Test Cases:

- 'Honestly, I think this whole thing is pretty messed up, you know?' → AI-generated (confidence: 1.000)
- 'Ugh, don't even get me started on that topic' → AI-generated (confidence: 1.000)
- 'My bad! Totally forgot about that meeting yesterday.' → AI-generated (confidence: 0.961)

Analysis: informal human language is also classified as AI-generated, indicating the model has learned bad correlations rather than genuine linguistic differences.

4.2 Root Cause Analysis: Overfitting

1. **Extreme Confidence Scores**: Most predictions show confidence levels of 0.998-1.000, indicating overconfident predictions
2. **Systematic Misclassification**: The model consistently misclassifies diverse text types, suggesting it learned dataset-specific patterns rather than generalizable features
3. **Contradictory Logic**: Both formal and informal language are flagged as AI-generated, indicating the model lacks coherent decision boundaries

4.2.2 Potential Causes

1. **Dataset Bias**: The training data contained the following differences, as observed in the EDA
    - Text length distributions
    - Topic domains distribution, I had more AI-generated texts rather than the human generated texts
2. **Limited Training Diversity**: The model has memorized training patterns rather than learning transferable linguistic features, as can be seen, the overfitting was present there

---

5. Ethical Considerations and Societal Impact

When we talk about ethical considerations, I believe that a lot of factors are involved in it.

5.1 Stakeholder Analysis

5.1.1 Potential Beneficiaries

1. **Educational Institutions**: Detecting academic dishonesty
2. **Publishers**: Ensuring content authenticity
3. **Researchers**: Understanding AI text generation capabilities

5.1.2 Potentially Harmed Groups

1. **Non-Native English Speakers**: Formal language constructions may trigger false positives
2. **Academic Writers**: Professional writing styles unfairly flagged
3. **Professional Communicators**: Business language may be incorrectly identified

5.2 Bias and Discrimination Concerns

Through my analysis, I've identified several concerning patterns of bias that emerged from the model's training process. The linguistic discrimination I observed is particularly troubling because the model penalizes both formal and informal communication styles, effectively creating an unrealistic narrow definition of what constitutes "acceptable" human language.

The cultural bias present in the system stems from the likelihood that the training data reflects specific cultural and linguistic contexts. What concerns me most is that this limitation means the model may not generalize effectively to global English varieties, potentially creating systemic bias against non-traditional communication patterns. This could unfairly penalize speakers who use different rhetorical structures, cultural references, or linguistic patterns that are perfectly natural in their contexts but differ from the dominant patterns in the training data.

Perhaps most concerning from an educational perspective is the model's tendency to flag academic language as AI-generated. This educational bias could have far-reaching consequences, potentially discouraging students from developing sophisticated writing skills and harming the fairness of educational assessments.I am a student as well, If I was to encounter such a case where my personal written content was classified as AI-generated, it would discourage me from using proper and correct lexical sound grammar.

5.3 Privacy and Surveillance Concerns

My research has uncovered significant privacy implications that extend beyond the immediate classification task. The systematic analysis of written communication that these models enable raises serious concerns about surveillance and behavioral monitoring. What particularly troubles me is the potential for these systems to develop behavioral profiling capabilities, where individual writing patterns could be identified and tracked across different platforms and contexts.

6. Technical Limitations and Future Directions

6.1 Current Limitations

6.1.1 Overfitting Issues

- **Problem**: Model memorizes training patterns rather than learning generalizable features

- **Evidence**: Contradictory classification patterns and extreme confidence scores
- **Impact**: Poor real-world generalization

6.1.2 Evaluation Methodology

- **Problem**: High test metrics don't reflect real-world performance
- **Cause**: Test set may share biases with training data
- **Solution**: Need for more diverse, adversarial evaluation sets

6.2 Recommended Improvements

Based on my comprehensive analysis of the model's failures, I believe several technical enhancements could address the overfitting issues I observed. Advanced regularization techniques represent the most immediate opportunity for improvement, particularly through implementing dropout layers in the classification head, apart from all of this, I think having a balanced dataset and using LoRa Adapaters would be much better in this case, rather than fine-tuning the entire model. These techniques could help combat the extreme confidence scores and rigid decision boundaries that characterize the current model's failures.

The training strategy itself also needs refinement, particularly through curriculum learning approaches that gradually expose the model to increasingly complex examples, domain adaptation techniques that could improve cross-domain generalization, and more sophisticated cross-validation with stratified sampling to ensure robust evaluation.

The data quality improvements I would prioritize focus on addressing the systematic biases I identified in the training process. Dataset diversification is crucial, requiring representation across multiple text domains, various writing styles, and content generated by different AI models to prevent the narrow pattern memorization I observed. Equally important is bias mitigation through balanced representation across demographic groups.

6.3 Alternative Approaches

Given the fundamental limitations I've identified in the current approach, I believe exploring alternative methodologies could offer more robust solutions to AI text detection. Ensemble methods represent one particularly promising direction, where combining multiple model types could help reduce the individual model biases that were not good for my current implementation. By aggregating predictions from diverse architectures—perhaps combining transformer-based models with traditional statistical approaches, I could achieve improved robustness and reduce the likelihood of systematic failures across the entire system.

LoRA (Low-Rank Adaptation) fine-tuning offers another compelling alternative that directly addresses the overconfidence issues I observed through parameter-efficient adaptation. Rather than relying on full model fine-tuning or static pre-trained models, LoRA would enable the targeted adaptation of specific model layers while preserving the original model's general capabilities. This approach could incorporate domain-specific fine-tuning that adapts models to particular writing styles or contexts, multi-task learning that simultaneously trains on detection and related tasks to improve generalization, and adaptive learning rates that allow the model to adjust its confidence based on input characteristics. LoRA's efficiency makes it practical to maintain multiple specialized variants for different domains while requiring significantly fewer computational resources than traditional fine-tuning approaches.

---

7. My Recommendations

My experience with this project has convinced me that normal people working on AI detection systems should change to approach evaluation and validation.
Comprehensive evaluation should become the norm, requiring systematic testing on diverse, adversarial examples that go well beyond standard test sets. I think one of the thing that became evident is that there should be a mandatory bias assessment across demographic groups, as the patterns I discovered suggest that many systems could have hidden biases that only become apparent through targeted analysis.

---

8. Conclusion

This comprehensive analysis reveals a critical disconnect between quantitative performance metrics and real-world applicability in AI text classification. While my fine-tuned DistilBERT model achieved impressive accuracy scores (99.97%), systematic error analysis exposed severe overfitting issues and concerning bias patterns.

Appendix

A. Technical Specifications

- **Hardware**: GPU-enabled training environment
- **Software**: PyTorch, Transformers, scikit-learn
- **Model Checkpoint**: Available at `./models/checkpoint-36543`

B. Additional Resources

- **Notebooks**: `baseline_model_final.ipynb`, `fine_tuned_model_final.ipynb`
- **Analysis Scripts**: `misclassification_analysis.py`, `evaluation_metrics.py`

- **Data**: AI vs Human Text dataset (487K samples)