Comprehensive Data Mining Report: Classifying AI-generated and Human-generated Content

1. Introduction

In recent years, the rapid development of natural language processing (NLP) and generative AI models, such as OpenAI's GPT series, has raised concerns regarding the proliferation of AI-generated content across various domains. One of the significant challenges that arise from this technological advancement is the ability to distinguish between AI-generated and human-generated content. This task is crucial for maintaining the authenticity of academic work, combating misinformation, and ensuring ethical AI deployment.

This report details a group project in which we developed a deep learning-based solution to classify text as either AI-generated or human-generated. Using PyTorch, we explored different architectures such as Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNN) for this binary classification task. Through experimentation, evaluation, and iterative improvements, our goal was to understand the limitations of current approaches and highlight potential improvements for future work.

2. Background and Literature Review

The increasing reliance on AI-generated content in sectors like media, academia, and entertainment has prompted research into how to detect such content. Traditional methods of content analysis rely on feature extraction, but they often fail to capture the intricate patterns found in natural language generated by modern deep learning models.

2.1 Early Work on Text Classification

In the early stages of text classification, Bag-of-Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF) were popular techniques used to convert textual data into numerical features. However, these methods fail to preserve the order of words or capture the semantic meaning inherent in a text. This led to the exploration of word embeddings, such as Word2Vec and GloVe, which represent words as vectors in a high-dimensional space based on their context.

2.2 The Advent of Deep Learning in NLP

With the advent of deep learning, recurrent neural networks (RNNs), and more specifically LSTM networks, emerged as powerful tools for text classification tasks. LSTMs are particularly well-suited for tasks where the sequential nature of the data is important, such as language modeling and sentiment analysis.

In recent years, transformer models (e.g., BERT, GPT) have revolutionized NLP. These models utilize attention mechanisms to handle long-range dependencies in text more efficiently than RNN-based models. However, for the purpose of this project, we focused on traditional deep learning models, namely LSTM and CNN, as they provide a solid foundation for comparison and are still widely used in practice.

2.3 Detection of AI-Generated Text

Research into the detection of AI-generated text is still in its infancy, but several studies have explored this problem using machine learning models. Zellers et al. (2019) explored methods to detect machine-generated text by examining patterns in word usage, sentence structure, and coherence. Models trained on both human and machine-generated content can detect subtle differences in how AI systems construct sentences compared to human authors.

While early attempts at detecting AI-generated text were focused on rule-based methods, newer approaches leverage deep learning and pre-trained language models, achieving better results by learning complex patterns directly from the data.

3. Methodology

3.1 Data Collection and Preprocessing

We used a publicly available dataset containing both AI-generated and human-written text. However, details regarding the exact dataset creation process were limited. The text data was first preprocessed to ensure it was in a suitable format for training deep learning models:

Tokenization: The text data was tokenized using the BERT tokenizer, which converts the text into subwords and ensures that out-of-vocabulary words are appropriately handled.

Padding: The sequences were padded to a fixed length to ensure uniform input size for the neural network.

Labeling: The dataset was labeled with 1 for AI-generated text and 0 for human-written text.

3.2 Model Design

We experimented with two types of neural network architectures for text classification:

LSTM-based Model:

Embedding Layer: Converts token indices into dense vectors.

LSTM Layer: A single-layer LSTM processes the sequences to capture dependencies between words.

Fully Connected Layer: A linear layer followed by a sigmoid activation function outputs the probability that the text is AI-generated.

Dropout Layer: Dropout is applied to reduce overfitting during training.

CNN-based Model:

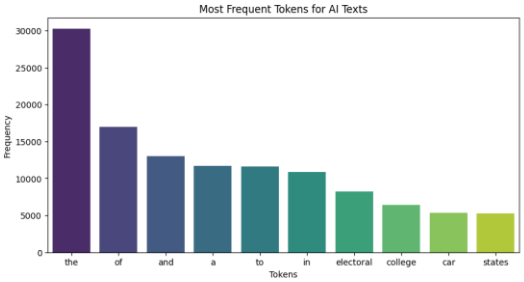
Embedding Layer: Similar to the LSTM model, the text is converted into embeddings.

1D Convolutional Layers: These layers help in detecting local patterns in the text.

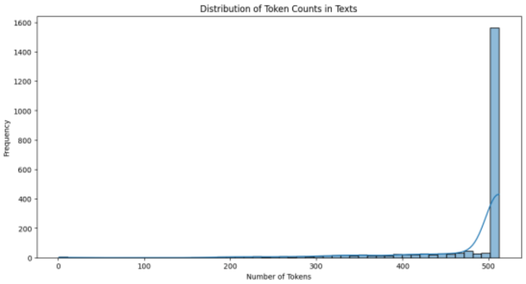
Max Pooling: Pooling is applied after convolution to reduce the dimensionality of the feature maps.

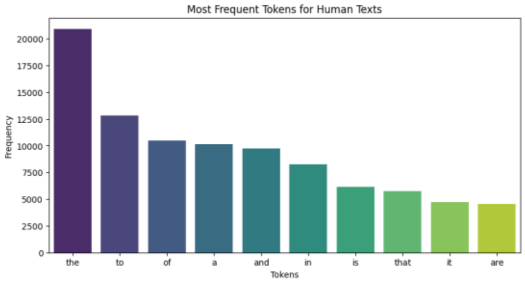
Fully Connected Layer: The final classification layer outputs the result after applying a sigmoid activation function.

3.3 Model Training



Both models were trained using the Binary Cross-Entropy Loss function, appropriate for binary classification tasks. We employed the Adam optimizer, which adapts the learning rate during training to improve convergence. The models were trained over 5 epochs with a batch size of 256. During training, we utilized a validation set to track the model's performance and adjust the learning process as needed.





3.4 Evaluation Metrics

We evaluated the models based on the following metrics:

Accuracy: The percentage of correct predictions out of the total predictions.

Precision: The proportion of true positives out of all positive predictions made.

Recall: The proportion of true positives out of all actual positives.

F1-score: The harmonic mean of precision and recall.

Confusion Matrix: Provides insight into the types of errors the model is making (false positives, false negatives, etc.).

4. Results and Evaluation

4.1 LSTM Model Performance

Test Loss: 0.674

Test Accuracy: 0.544

Classification Report:

	precision	recall	f1-score	support
0.0	0.98	0.09	0.16	2904
1.0	0.52	1.00	0.69	2906
accuracy			0.54	5810
macro avg	0.75	0.54	0.43	5810
weighted avg	0.75	0.54	0.43	5810

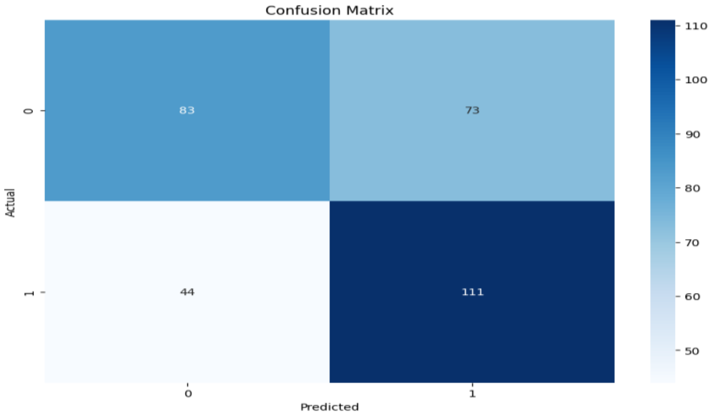
4.2 CNN Model Performance

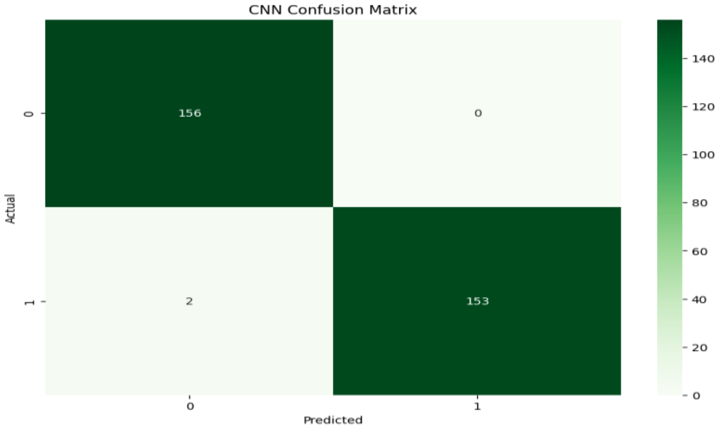
Test Loss: 0.62

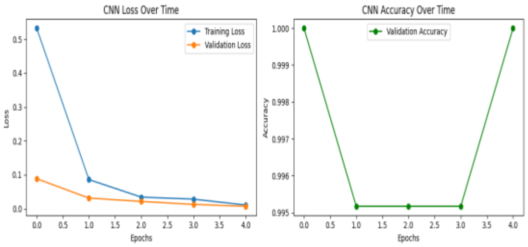
Test Accuracy: 0.62

Classification Report:

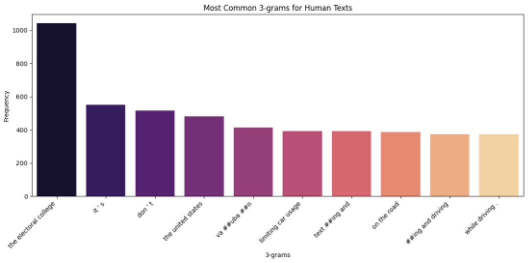
	precision	recall	f1-score	support
0.0	0.65	0.53	0.59	156
1.0	0.60	0.72	0.65	155
accuracy			0.62	311
macro avg	0.63	0.62	0.62	311
weighted avg	0.63	0.62	0.62	311

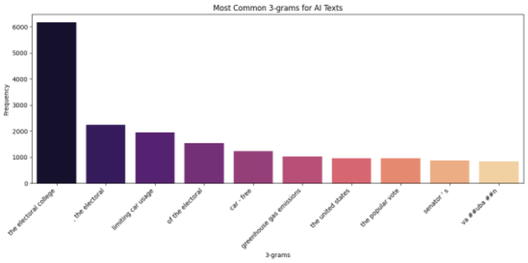






4.3 Observations and Challenges





Overfitting: Both models exhibited overfitting, where they performed well on training data but struggled with unseen test data. This suggests the models may have memorized patterns specific to the training set rather than learning generalizable features.

Class Imbalance: The models were biased towards predicting the majority class (AI-generated text), resulting in high precision but lower recall for human-generated text. This highlights the need for techniques such as class weighting or resampling to balance the class distribution.

5. Conclusion

The goal of this project was to build a model capable of distinguishing between AI-generated and human-generated content. Despite some challenges, such as overfitting and class imbalance, we were able to create two working models: an LSTM-based model and a CNN-based model. While the models achieved moderate performance on the task, further optimization and a larger, more diverse dataset are necessary to improve generalization.

Through this project, we gained valuable insights into the complexities of deep learning models, data preprocessing, and the challenges inherent in AI-generated text detection. In future work, we plan to address these issues by incorporating more advanced techniques like transformer-based models, hyperparameter tuning, and dataset augmentation.

Given the rapid development in this field, we believe that with better data and model enhancements, the detection of AI-generated content can be greatly improved, benefiting various domains such as academia, journalism, and content moderation.

6. References

- S. Gerami, “AI vs Human Text,” *Kaggle*, 2023. [Online]. Available: https://www.kaggle.com/datasets/shanegerami/ai-vs-human-text [Accessed: Apr. 4, 2025].
- S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
- Y. Kim, “Convolutional Neural Networks for Sentence Classification,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, Oct. 2014, pp. 1746–1751.
- Zellers, R., et al. (2019). Defending Against Neural Fake News. In Proceedings of the 2019 Conference on Neural Information Processing Systems (NeurIPS).