Final Project Report: AI Text Detection System

Peilin Rao (jackrao@g.ucla.edu)**, Nicole Ju** (nicoleju@g.ucla.edu)**,**
Zinnia Kwan (zinniakwan@g.ucla.edu), **Ryan Phua** (rphua@g.ucla.edu),

1 colab link

https://colab.research.google.com/drive/1JJjZtHZ0BlNJwF5QRiqz2lt-0tfV4pnG

2 Data Collection and Preparation

Our training dataset consists of both human and AI-generated text drawn from diverse domains, including academic essays, news articles, question/answer formats, and research abstracts. Examining the provided dev set, we noticed that the arxiv dataset consists of scientific/academic samples, while the reddit dataset includes more informal, essay-like writing. We tried to reflect these domains, as well as add additional variety, in our training data. We pulled from the following datasets:

- **DAIGT V2**[1] **:** includes 44,868 samples of human-written and AI-generated essays. The human essays are drawn from the Persuade corpus[2], which consists of 25,000 argumentative essays produced by 6th-12th grade students. The AI-generated essays came from a variety of models such as ChatGPT, PaLM from Google Gen-AI, and Claude.
- **TURINGBENCH**[3] **:** created by collecting 10K news articles and passing their titles through 19 AI text generators. This produced 200K articles and 20 labels in total. To maintain a better overall balance between human and AI-generated texts, we randomly sampled 30K of the AI-generated samples.
- **HC3**[4] **:** consists of 24,300 questions alongside their corresponding human and ChatGPT answers (yielding 85,449 text-label pairs in total). We chose this dataset to better cover informal and conversational discourse, as seen in the reddit dataset.
- **ChatGPTResearchAbstracts**[5] **:** includes 10k human-writen and 10K AI-generated scientific research abstracts. Utilizing this dataset enabled us to better generalize to academic writing as seen in the arxiv dataset.

After combining these datasets, our dataset consisted of around 180K entries formatted as (text, label) pairs. It was relatively balanced between human and AI text, with 104771 human and 80962 AI-generated entries. Before training, we split the dataset into 70% training, 15% validation, and 15% test.

2.1 Baseline Comparison

We created a Naive-Bayes Model as a baseline model to compare against our proposed model architecture. The training data was from DAIGT V2, TURINGBENCH, and HC3. On its own, this model performs better than random, but not by much. This demonstrates that there is much room for improvement with a more advanced model.

3 Modeling Approach

Metric	Score
Accuracy	0.5571
Precision	0.5925
Recall	0.3657
F1 Score	0.4522

Figure 1: Baseline Model Performance on Devset

[1] https://www.kaggle.com/datasets/thedrcat/daigt-v2-train-dataset

[2] https://www.kaggle.com/datasets/nbroad/persaude-corpus-2/

[3] https://turingbench.ist.psu.edu/

[4] https://huggingface.co/datasets/Hello-SimpleAI/HC3

[5] https://huggingface.co/datasets/NicolaiSivesind/ChatGPT-Research-Abstracts

3.1 Feature Extraction

We have implemented a hybrid feature extraction method that combines semantic content analysis with writing style analysis.

1. **Semantic Embeddings:** For our training and evaluation datasets, pre-computed dense vector representations ($\boldsymbol{e} \in \mathbb{R}^{768}$), stated to be from the Google `text-embedding-004` model, were utilized. These embeddings capture the fundamental meaning of the text and are loaded from `semantic_embeddings.csv`.

2. **Stylometric Features:** We have developed a pipeline to extract a feature vector ($\boldsymbol{s} \in \mathbb{R}^{50}$) designed to capture subtle stylistic cues that may not be adequately represented by the semantic embedding. This pipeline, implemented in our Python script (`extract_stylometric_features`), computes features across multiple categories:
    - **Lexical features**: average word length, type-token ratio, hapax legomena (rarity), and repetition score. Human writers tend to show more variation.
    - **N-gram diversity and entropy**: character trigram and word bigram diversity/entropy. AI text often reuses common patterns.
    - **Grammar and syntax features**: POS tag distribution, subordinate clause ratio, syntactic surprise, and parse tree depth. These features capture grammatical complexity that may differ between human and AI writers.
    - **Punctuation and formatting**: frequency of various punctuation marks and capitalization ratio.
    - **Readability metrics**: Flesch Reading Ease and Gunning Fog Index, which quantify how difficult the text is to read.
    - **Discourse and fluency**: sentence length, contraction usage, typo frequency, and burstiness (variation in word re-use).

    We collected the computed features into single fixed-length vector ($\boldsymbol{s} \in \mathbb{R}^{47}$), where each component corresponds to a stylometric element. POS tag distributions and punctuation frequencies, which are dictionaries, are expanded into individual scalar values before being appended to the final feature vector.

3.2 Model Architecture

The main classification model, which has been implemented, utilizes a neural network structure that unites semantic information with attended stylometric features.

- **Stylometric Feature Attention:** The stylometric feature vector $\boldsymbol{s}$ is processed by a multi-head self-attention mechanism, implemented as the `SelfAttention` module in our code. This module generates an attention-weighted stylometric representation $\boldsymbol{s}_{att}$ by determining the relative importance of stylistic features and their internal dependencies.

- **Feature Fusion:** The semantic embedding $\boldsymbol{e}$ is then concatenated with the attended stylometric vector $\boldsymbol{s}_{att}$ to create the final classifier input vector $\boldsymbol{x}_{att} = [\boldsymbol{e}; \boldsymbol{s}_{att}]$.

- **Classifier:** A Multi-Layer Perceptron (MLP) takes the combined vector $\boldsymbol{x}_{att}$ as input to make predictions about text origin (human vs. AI). Our implemented `TextClassifier` uses an MLP with hidden layers utilizing ReLU activation functions and an output layer with a Sigmoid activation function for binary classification.

4 Mid-Project Results & Model Evaluation

3.1 Model Performance on Evaluation Metrics

Our mid-project model configuration consists of:

- Semantic embedding dimension: 768 (from Google text-embedding-004)

- Stylometric embedding dimension: 50 (engineered features)

The model was trained on 262,124 samples with a 70-15-15 train-validation-test split. Training employed early stopping with patience of 5 epochs, monitoring validation AUC as the primary metric. Additional model specifications:

- **Hidden layers**: [256, 128, 64] with ReLU activation

- **Total parameters**: 261,081

- **Optimizer**: Adam optimizer with (learning rate=0.001) and (weight decay=1e-5)

During training, the model converged within 10 epochs. Training loss decreased consistently from 0.0714 in epoch 1 to 0.0171 in the final epoch, while validation loss reached a minimum of 0.0154.

3.2 Internal Test Set Evaluation

Performance on the internal test set:

Metric	Score
Accuracy	0.9948
Precision	0.9959
Recall	0.9973
F1 Score	0.9966
AUC	0.9997

Table 1: Model Performance on Internal Test Set

The model demonstrates strong discriminative capability, with minimal misclassification across both classes. However, these extremely high performance metrics indicate potential overfitting to the internal training data. The model is slightly more conservative in predicting AI-generated text.

3.3 Development Set Evaluation

The model's performance was also evaluated on the provided Development Set (Devset):

Metric	Score
Accuracy	0.7621
Precision	0.7110
Recall	0.7722
F1 Score	0.7404
AUC	0.8593

Table 2: Model Performance on Development Set (Devset)

The confusion matrix for the Development Set in Figure 2 shows the model's performance on this dataset. While it still discriminates between the classes, there are notably more misclassifications compared to the internal test set. This indicates a potential generalization gap between the internal test set and the Devset, which may be due to overfitting or differences in data distribution. The asymmetry in errors (2,141 false negatives vs. 2,950 false positives) show that on the Devset, the model is more likely to incorrectly label human-written text as AI-generated. This could mean that human-written texts in the Devset might possess characteristics that the model, trained on other corpora, associates with AI generation, or that the AI-generated

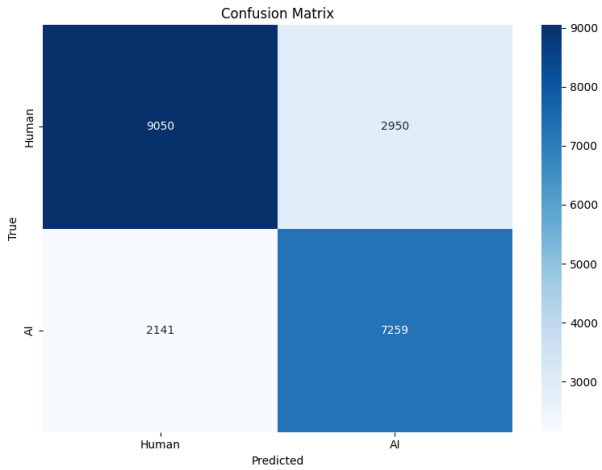

Figure 2: Confusion Matrix on Development Set (Devset). The model correctly classified 7,259 AI-generated samples and 9,050 human-written samples. There were 2,950 false positives (human text classified as AI) and 2,141 false negatives (AI text classified as human).

texts in the Devset are less distinguishable by the current feature set.

5 Mid-project Improvements

5.1 Dataset Analysis and Improvements

Our intial training datasets included the DAIGT V2, TURINGBENCH (without random sampling), and HC3 datasets, yielding 262,124 samples. These sets were mainly chosen for domain variety, but not examined closely for balance. Further analysis revealed that the TURINGBENCH dataset, with 200,000 news articles, contributed a large majority of all the training samples. This skewed the data toward more journalistic, opinionated writing. Furthermore, we investigated the balance between human and AI samples. Within the TURINGBENCH set, there were only 10K human and 190K AI-generated samples, likely contributing to the model's high false positive rate.

To address both these issues, we decided to randomly sample from TURINGBENCH's AI-generated articles. Since the other training datasets were slightly skewed toward human texts, we decided to sample 30K AI-generated articles in addition to the 10K human-written ones.

Next, we evaluated performance on the reddit vs arxiv sets from the Devset to analyze domain generalization. Tables 3 and 4 show the performance on each dataset.

The model achieved better overall performance on the reddit dataset, with a very high recall score.

Metric	Score
Accuracy	0.7956
Precision	0.6480
Recall	0.9521
F1 Score	0.7712
AUC	0.9326

Table 3: Model Performance on Reddit Set (Devset)

Metric	Score
Accuracy	0.7358
Precision	0.7714
Recall	0.6703
F1 Score	0.7173
AUC	0.8171

Table 4: Model Performance on Arxiv Set (Devset)

These results indicate that the model was overfitting to more informal, essay-like writing styles and failed to generalize to the scientific papers seen in the arxiv dataset. To improve generalization, we incorporated a new Hugging Face dataset with scientific research abstracts: ChatGPTResearchAbstracts[6].

5.2 Hyperparameter Tuning

To optimize our model's performance, we performed hyperparameter tuning through a Grid Search approach on 10% of training data, using only 50% of Devset as validation and the rest 50% as test set later to avoid data leakage. We explored variations of learning rate, dropout rate, weight decay, and the number of hidden layers and their dimensions. After extensive experimentation, we identified the following hyperparameters as yielding the best validation performance on 10% training data (validation AUC = 0.9001):


```
{
    "hyperparameters": {
        "lr": 0.0005,
        "weight_decay": 0.0005,
        "dropout_rate": 0.25,
        "hidden_dims": [128, 64]
    },
   "validation_auc": 0.9000554787234043,
   "timestamp": "2025-06-06T00:17:20.025974"
}
```


These optimized parameters significantly improved generalization compared to earlier configurations, demonstrating balanced regularization and sufficient model capacity for capturing complex patterns in the stylometric and semantic features.

5.3 Stylometric Feature Refinement

We aimed to better understand and optimize the contribution of stylometric features to classification performance. First, we applied Principal Component Analysis (PCA) and found that the top 35 components captured approximately 95% of the total variance, suggesting that some features contributed minimally and could introduce noise. Then, we employed a Random Forest Classifier to compute feature importance, which revealed that a few features, specifically `passive_ratio`, `sentiment`, and some punctuation usages had near zero importance scores. To visualize and support these findings, we generated kernel density plots by label, which highlighted which features showed meaningful separation between human and AI text. Aligning with results from earlier, `passive_ratio`, `sentiment`, and – frequency demonstrated near-identical distributions across labels.

Ultimately, we removed the three low importance features, reducing the dimensionality from 50 to 47. This refinement aimed to reduce the input space (improving model training speed), and reduce overfitting on noisy/noninformative features.

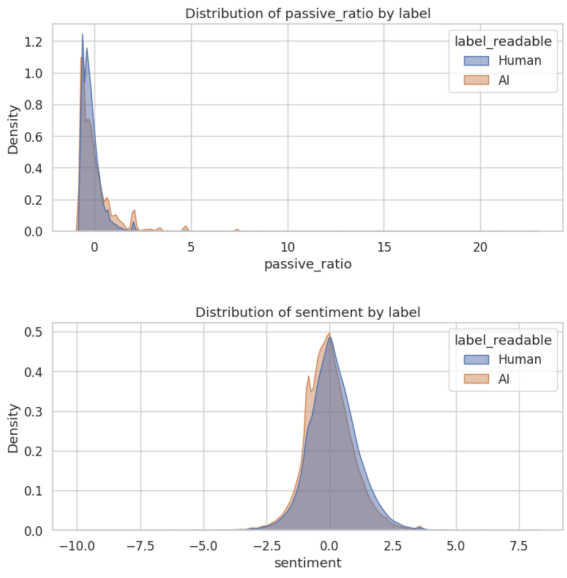

Figure 3: Graphs showing features with low distribution differences.

[6]https://huggingface.co/datasets/NicolaiSivesind/ChatGPT-Research-Abstracts

6 Final Results

6.1 Development Set Final Evaluation

The final model performed noticeably better on the Devset. It achieved an accuracy of 0.8899, as compared to the original accuracy of 0.7358. Hyperparameter tuning such as a smaller learning rate and a higher weight decay, along with the inclusion of research abstracts in the dataset, helped address issues with overfitting. This enabled the model to generalize better to the Devset. Referencing Figure 5, there are noticeably fewer false positives (77 compared to 2950), suggesting that balancing AI to human-written text helped address the overprediction of AI. Before our changes, there were more false positives than false negatives, but we now observe more false negatives (1101), meaning the model was more conservative in predicting text as AI. This may be due to the slight skew toward human-written samples in our updated training dataset, or other distributional differences between the datasets.

Metric	Score
Accuracy	0.8899
F1 Score	0.8594
AUC	0.9442

Figure 4: Model Performance on Devset

6.2 Ethics Sets Final Evaluation

We evaluated our model on three distinct ethics datasets: TOEFL Essays, Hewlett Essays, and German Wikipedia excerpts. The results can be visualized from the confusion matrices in Figures 5 to 7 below.

TOEFL Essays The model demonstrated poor discrimination capabilities on the TOEFL Essays dataset, as seen in Figure 5., with an accuracy of only 52.75%. All 91 samples were predicted as human-written, yielding no true positives for AI-generated content. This resulted in undefined precision, recall, and F1-score (all effectively zero). Such outcomes suggest significant biases or inadequacies in detecting AI-written content in formal academic English.

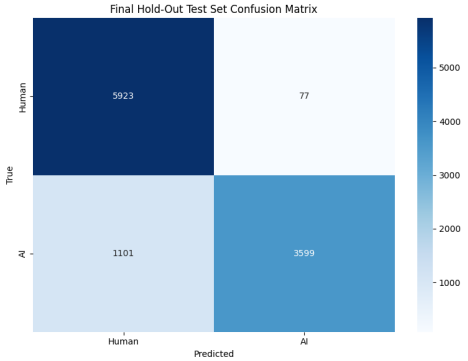

Figure 5: Final Confusion Matrix for Devset

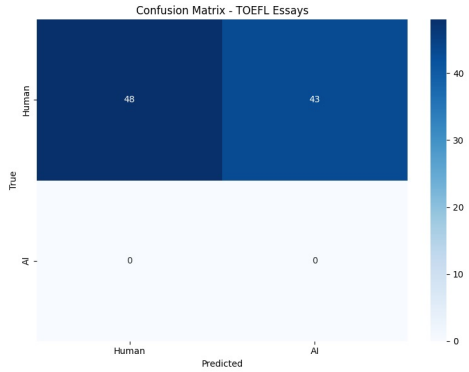

Figure 6: Confusion Matrix for TOEFL Essays

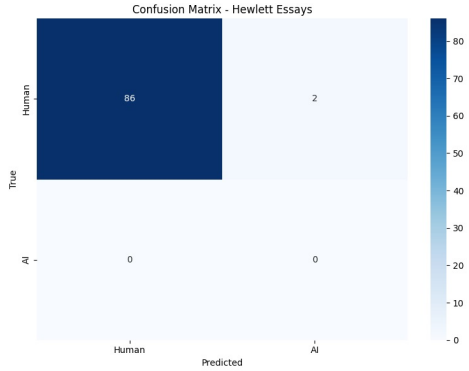

Figure 7: Confusion Matrix for Hewlett Essays

Hewlett Essays On the Hewlett Essays dataset, the model achieved high accuracy (97.73%), but again failed completely to identify AI-generated essays, classifying all 88 samples as human-written. Despite accuracy being high due to dataset imbalance (predominantly human samples), this indicates the model's inability to generalize its detection capabilities to nuanced or formal essay writing.

German Wikipedia Performance on the German Wikipedia dataset was notably better, albeit still with limitations. The accuracy was moderate (67.70%), with a precision of 89.33% and recall of 40.20%. The relatively low recall indicates difficulty in consistently detecting AI-generated German text, possibly due to language-specific stylometric and semantic differences that the model,

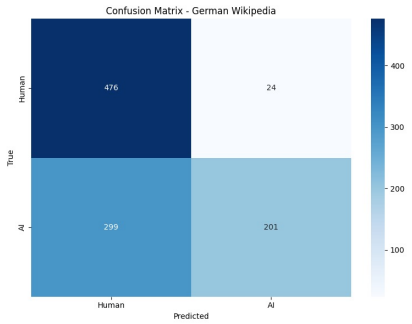

Figure 8: Confusion Matrix for German Wikipedia Dataset

primarily trained on English datasets, cannot capture effectively.

The combined performance metrics are summarized in Table 5 below.

Metric	TOEFL Essays	Hewlett Essays	German Wikipedia
Accuracy	0.5275	0.9773	0.6770
Precision	0.0000	0.0000	0.8933
Recall	0.0000	0.0000	0.4020
F1 Score	0.0000	0.0000	0.5545

Table 5: Combined Model Performance on Ethics Datasets

The evaluation results on the ethics datasets highlight significant generalization challenges faced by our model. On the TOEFL Essays and Hewlett Essays datasets, precision, recall, and F1 scores are all zero, indicating complete failure to detect AI-generated essays. This outcome is primarily caused by dataset imbalance and domain mismatch. The model was predominantly trained on informal or journalistic text, leading it to misclassify formal academic essays entirely as human-generated. In contrast, the German Wikipedia dataset exhibited moderate performance with notably high precision (0.8933) but relatively lower recall (0.4020), suggesting the model successfully identifies some distinct AI-generated features but lacks comprehensive coverage, likely due to linguistic and stylistic differences inherent to German-language text.

6.3 Note on Evaluation Script

Our deliverables include two unnormalized stylometric embedding files (to properly initialize the StandardScaler) along with our trained model. Note that our script takes a while to generate the stylometric embeddings for the test data, so please be mindful of that.

7 Contribution Statement

Name	Contribution
Peilin	Perform semantic embedding. Train model on whole dataset and test on Devset. Ethics and bias evaluation + hyperparameter tuning
Nicole	Researched training datasets, built pipeline to extract stylometric features (and generated stylometric embeddings for training), performed data analysis and restructured the training data.
Ryan	Implemented PyTorch ML pipeline in Colab, including dataloader classes, model architecture, train/validation loop, and end-to-end testing, detailed evaluation + hyperparameter tuning
Zinnia	Built pipeline to extract stylometric feature, normalized stylometric embeddings, generated naive-bayes baseline model, analyze Semantic Embedding impact (potentially with PCA), drop low-impact embeddings to reduce noise