CIS 5300: Natural Language Processing - Term Project

AI-GENERATED OR HUMAN-WRITTEN ESSAY?

Group : Tanisha Admane, Ishwari Mulay, Aeshon Balasubramanian, Nhat-ha Pham

December 2025

PROBLEM STATEMENT

Consider two university essays on the same topic. One is written by a student, reflecting their unique voice, learning, and potential struggles. The other is generated by an advanced AI, exhibiting perfect grammar, coherent structure, but perhaps lacking genuine insight or human-like imperfections.

How do we reliably distinguish between these two?
This is the core challenge our project addresses.

ILLUSTRATIVE EXAMPLE

Example Prompt: "Discuss the impact of social media on modern communication."

Human-written text tends to include:

- Natural variation in sentence length
- Minor stylistic imperfections
- Personal voice or opinion
- Emotional or colloquial expressions

AI-generated text often shows:

- Low perplexity, overly smooth structure
- Even sentence lengths
- Neutral, encyclopedic tone
- High lexical diversity but limited individuality

PROJECT GOALS

Build a model that determines whether a given essay is AI-generated or human-written.

- Input:
  - Raw essay text
- Output:
  - Binary label: 0 = Human, 1 = AI-Generated
  - (Optional) model confidence probability
- Core Challenges:
  - AI text is increasingly indistinguishable from human text
  - Dataset imbalance
  - Need for robust features + context-aware representations
  - Generalizing across prompts and writing styles

BACKGROUND & PRIOR WORK

What Existing Research Shows?

From the literature review:

- Classic ML models (SVM, TF-IDF, XGBoost) provide solid baselines but can fail against advanced LLMs.
- BERT-based detectors outperform traditional methods by capturing contextual signals.
- Adversarial attacks (e.g., paraphrasing, prompt variation) can break naïve detectors.
- Stylometric cues remain useful: POS patterns, perplexity, n-grams, writing complexity.

Key Takeaway:

Robust AI detectors require more than surface-level features, they need contextual understanding and adversarial resilience.

ORIGINAL DATASET

- ~ 500,000 essays, both human-written and AI-generated texts.
- 2 columns: Essay & Label

- Split into training, development, and test sets:

  - Training set size: 389788 rows
  - Development set size: 48723 rows
  - Test set size: 48724 rows

METRICS

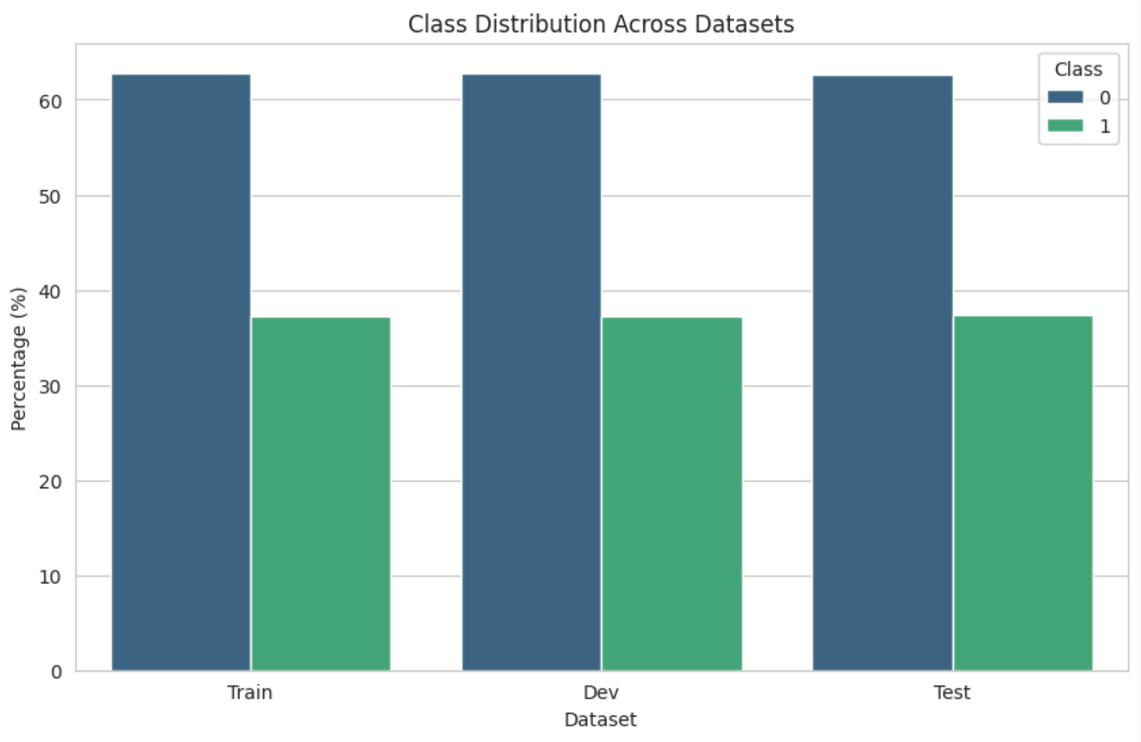
Weighted precision, recall, F1 Score were used in order to account for class imbalance in data (see right).

Imbalanced data could cause precision & recall to be high even with low performance on smaller class

$$\text{Precision}_{\text{weighted}} = \sum_{i \in C} \frac{n_i}{\sum_{j \in C} n_j} \cdot \frac{TP_i}{TP_i + FP_i},$$

$$\text{Recall}_{\text{weighted}} = \sum_{i \in C} \frac{n_i}{\sum_{j \in C} n_j} \cdot \frac{TP_i}{TP_i + FN_i},$$

$$F_{1,\text{weighted}} = \sum_{i \in C} \frac{n_i}{\sum_{j \in C} n_j} \cdot \frac{2 \cdot \text{Precision}_i \cdot \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i}.$$



SIMPLE BASELINE

MAJORITY CLASS PREDICTOR

Majority class identified in training data:
→ **Class 0 (Human-written)**

This baseline predicts the majority label for every example, providing a minimal reference point that all later models must improve upon.

Interpretation:

- The model achieves relatively high recall simply by predicting the dominant class.
- Very low precision indicates poor class discrimination.
- Serves as a baseline for evaluating the effectiveness of more advanced models.

Development Set Performance

- Accuracy: 0.6282
- Precision: 0.3946
- Recall: 0.6282
- F1-Score: 0.4848

Test Set Performance

- Accuracy: 0.6261
- Precision: 0.3919
- Recall: 0.6261
- F1-Score: 0.4821

FEATURE ENGINEERING

CORE FEATURES

- Text length
- Flesch Reading Ease and Automated Readability Index for text comprehension.
- Words & Sentence metrics: Counts of words, unique words, sentences, and average lengths to assess complexity.
- Type-Token Ratio and Stop Word Ratio to quantify vocabulary richness and style.
- Punctuation count

CONTENT CATEGORIZATION

1

Keyword-Based

Initial classification into categories like 'Opinion', 'Informative', or 'Creative Writing' using predefined keywords.

2

Encoded Categories

Numerical representation of keyword-based categories using LabelEncoder for ML compatibility.

3

ML-Derived Categories

Refined classification using a Logistic Regression model trained on 5000 TF-IDF features (accuracy = 0.8959)

STRONG BASELINE

LINEAR SVM + TF-IDF

Why SVM?

- Performs well in high-dimensional text settings
- Strong classical baseline for NLP classification
- Captures discriminative lexical patterns better than simple baselines

Key Insight:

SVM delivers a significant jump over the simple baseline (0.62 → ~0.90 accuracy), establishing a strong foundation for our Milestone 3 extensions.

Development Set Results

- Accuracy: 0.8996
- Precision (weighted): 0.8993
- Recall (weighted): 0.8996
- F1-Score (weighted): 0.8987

Test Set Accuracy

- 0.8990

EXTENSION 1: BIDIRECTIONAL LSTM

To better capture long-range and contextual dependencies, we implemented a Bi-LSTM neural network, learning representations directly from raw text.

0.9801

Accuracy

Overall correct predictions.

0.9700

Precision

Reliability of positive predictions.

0.9768

Recall

Sensitivity to actual positives.

0.9734

F1-Score

Harmonic mean of precision & recall.

This Bi-LSTM architecture significantly outperformed both the majority baseline and the SVM.

EXTENSION 2: FINE-TUNING BERT

We selected bert-base-uncased because its deep, bidirectional understanding of linguistic context allows it to capture the subtle nuances necessary for this task.

0.9988	0.9986	0.9988	0.9987
Accuracy	Precision	Recall	F1-Score
Overall correct predictions.	Reliability of positive predictions.	Sensitivity to actual positives.	Harmonic mean of precision & recall.

The BERT-based classifier demonstrates near-perfect performance on this dataset, effectively distinguishing between the two classes with ~99.9% accuracy.

NEXT STEPS

- Use supervised fine tuning to build a model that is tailored for our task
  - Would require getting additional essay data for training LLM
- Use larger off-the-shelf LLMs like ChatGPT or Claude through APIs
  - Could see much better performance than from open-source models

THANK YOU