TECHNICAL REPORT: AI TEXT DETECTION MODEL TRAINING

1. Project Overview

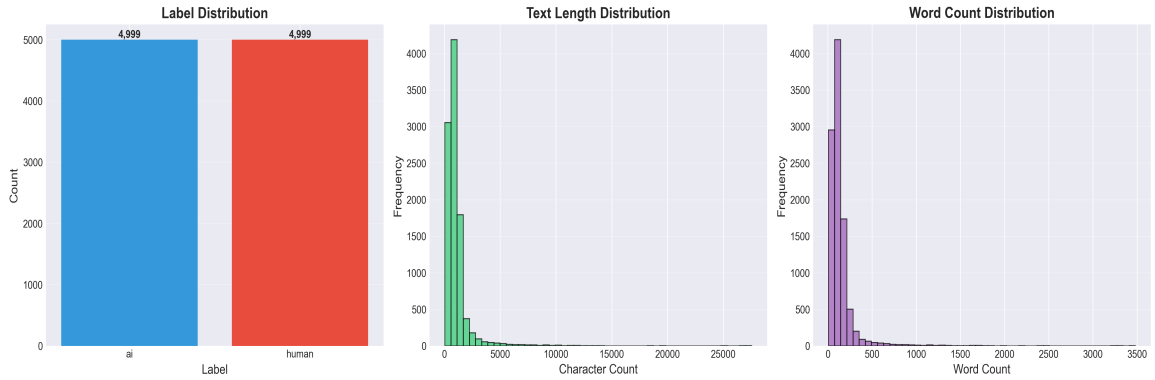
This project focuses on developing a machine learning system capable of distinguishing between human-written text and AI-generated text. The objective of this stage is to establish a baseline classification model using traditional machine learning approaches with feature engineering techniques. The goal is to achieve high accuracy in binary classification, with a target of 99% accuracy, though the current implementation achieves 97.40% accuracy on the test set.

The project addresses the growing concern of AI-generated content in various domains, including journalism, academic writing, and online content. The ability to reliably detect AI-generated text has important implications for content verification and authenticity assessment.

2. Dataset Description

The dataset used in this training pipeline consists of 9,998 text articles, with an equal distribution between AI-generated and human-written content. Specifically, the dataset contains 4,999 articles labeled as "ai" and 4,999 articles labeled as "human", resulting in a perfectly balanced dataset.

The text articles vary significantly in length, with an average of approximately 1098 characters and 142 words per article. The minimum article length is 51 characters, while the maximum extends to 27,550 characters, indicating substantial diversity in content length.



3. Data Preprocessing

The preprocessing pipeline applies several text cleaning operations to standardize the input data. The cleaning function removes URLs and email addresses using regular

expressions, as these elements do not contribute to distinguishing AI from human text and may introduce noise. The preprocessing also normalizes whitespace by collapsing multiple spaces into single spaces.

Special Unicode characters are handled, including non-breaking spaces and zero-width characters that may have been introduced during data collection or processing. Excessive punctuation is normalized, with multiple consecutive exclamation marks, question marks, or periods reduced to single or standard forms.

4. Feature Extraction

The feature extraction process employs a dual approach using both word-level and character-level n-gram features, combined through TF-IDF (Term Frequency-Inverse Document Frequency) vectorization. This approach captures both semantic patterns at the word level and stylistic patterns at the character level.

Word-level features are extracted using TF-IDF vectorization with n-grams ranging from 1 to 3, meaning the model considers individual words, two-word phrases, and three-word phrases. The word vectorizer is configured with a maximum of 20,000 features, using sublinear term frequency scaling to reduce the impact of very frequent terms.

Character-level features are extracted using character n-grams ranging from 3 to 6 characters, capturing stylistic patterns such as common character sequences, punctuation usage, and writing style markers that may differ between AI and human text. The character vectorizer uses 30,000 maximum features with similar filtering parameters as the word vectorizer.

The two feature sets are combined using sparse matrix concatenation, resulting in a total of 50,000 features per document. This high-dimensional feature space allows the model to capture subtle differences in writing patterns, vocabulary usage, and stylistic elements that distinguish AI-generated text from human-written text.

5. Model Used

Two classification models were trained and evaluated: Logistic Regression as a baseline model and Support Vector Machine (SVM) as the primary model.

Logistic Regression was chosen as a baseline because it is a simple, interpretable linear classifier that provides fast training and serves as a performance benchmark. The Logistic Regression model uses the L-BFGS solver with balanced class weights to handle any potential class imbalance, and it was trained with a maximum of 2,000 iterations.

The Support Vector Machine with a linear kernel was selected as the primary model because linear SVMs are well-suited for high-dimensional sparse feature spaces, such as those created by TF-IDF vectorization. Linear SVMs can effectively find decision boundaries in high-dimensional spaces while being computationally efficient compared to non-linear kernels.

6. Training Setup

The dataset was split into training and testing sets using an 80-20 split, with stratification to maintain the class distribution in both sets. This resulted in 7,998 training samples and 2,000 test samples. Stratification ensures that both sets have approximately equal representation of AI and human text, which is important for fair evaluation.

For the SVM model, hyperparameter tuning was performed using grid search with cross-validation. The grid search was optimized for speed by using a subset of 3,000 training samples and 3-fold cross-validation instead of the full dataset and 5-fold cross-validation. The hyperparameter grid explored three values of the regularization parameter C (1.0, 2.0, and 5.0) and balanced class weights.

The best hyperparameters found were C=5.0 with balanced class weights. The final SVM model was then trained on the complete training set of 7,998 samples using these optimal parameters.

7. Evaluation Metrics & Results

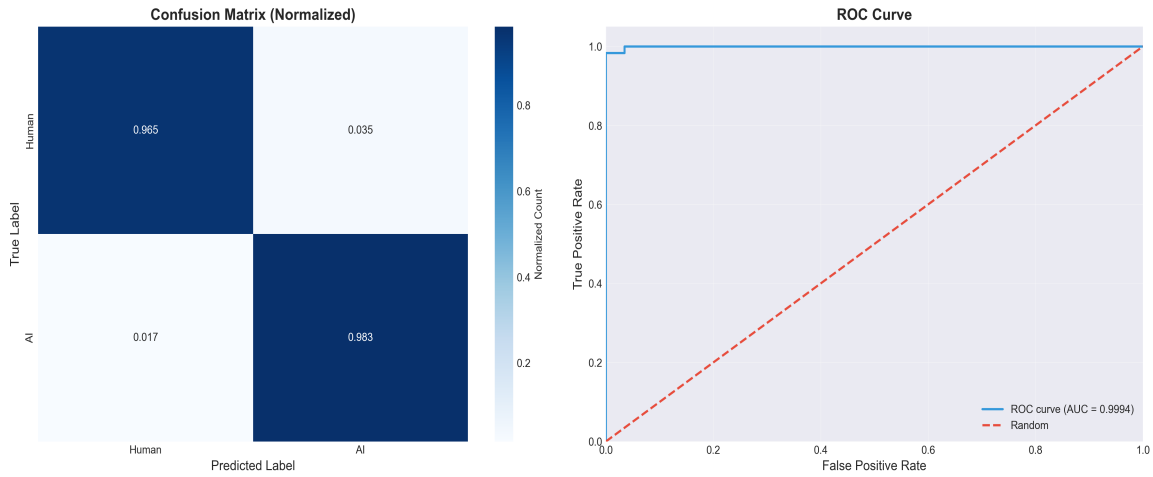
The models were evaluated using multiple metrics including accuracy, precision, recall, F1-score, and cross-validation scores. The primary evaluation was performed on the held-out test set of 2,000 samples.

The SVM model achieved an accuracy of 97.40% on the test set, with an F1-score of 0.9742. The per-class performance metrics show that the model performs well on both classes. For human-written text (Class 0), the precision is 0.9827, recall is 0.9650, and F1-score is 0.9738. For AI-generated text (Class 1), the precision is 0.9656, recall is 0.9830, and F1-score is 0.9742.

The confusion matrix reveals the following breakdown: 965 true negatives (human text correctly identified as human), 35 false positives (human text incorrectly identified as AI), 17 false negatives (AI text incorrectly identified as human), and 983 true positives (AI text correctly identified as AI).

The Logistic Regression baseline model achieved an accuracy of 95.65% with an F1-score of 0.9579, demonstrating that the SVM model provides a meaningful improvement over the baseline.

Cross-validation was performed on the training set using 3-fold stratified cross-validation, resulting in a mean accuracy of 96.79% with a standard deviation of 0.0018. This low standard deviation indicates that the model's performance is stable across different data splits, suggesting good generalization capability.



8. Limitations of Current Approach

Several limitations constrain the current approach and prevent it from reaching the target accuracy of 99%.

The dataset size, while substantial at approximately 10,000 articles, may not be sufficient to capture the full diversity of AI-generated text patterns, especially as AI text generation models continue to evolve and improve. The dataset may also be limited in terms of domain diversity, potentially containing text from specific sources or topics that may not generalize to all types of content.

The feature extraction approach, while comprehensive, relies on traditional n-gram and TF-IDF features that may not capture deeper semantic or contextual patterns that distinguish AI from human text. These surface-level features may be insufficient as AI text generation becomes more sophisticated and human-like.

The models used are linear classifiers, which may not be able to capture complex non-linear relationships between features that could improve classification accuracy. While linear models are computationally efficient, they may be limited in their ability to learn sophisticated patterns.

9. Next Steps

Several improvements can be pursued to enhance model performance and move closer to the 99% accuracy target.

First, expanding the dataset size and diversity would provide more training examples and help the model generalize better to different types of content. This could include collecting text from various domains, languages, and AI generation models to ensure robust performance across different scenarios.

Second, exploring deep learning approaches could capture more sophisticated patterns. Transformer-based models such as BERT or RoBERTa could be fine-tuned for this task, potentially learning semantic and contextual features that traditional n-gram features miss. These models have shown strong performance in text classification tasks and may provide the necessary boost to reach 99% accuracy.

Third, incorporating additional features beyond n-grams could improve discrimination. This might include syntactic features such as parse tree structures, semantic features from word embeddings, or statistical features such as readability scores, sentence length distributions, or punctuation patterns.

Fourth, ensemble methods could combine multiple models to improve overall performance. Combining predictions from different model types or different feature sets might capture complementary patterns and reduce individual model errors.

Finally, continuous evaluation and retraining as new AI text generation models emerge would ensure the detector remains effective against evolving generation techniques. This would require establishing a pipeline for collecting new data and updating models regularly.