Detecting Arabic AI-Generated Text Using Machine Learning

Rashid Binkulaib – 444106414
Mohammed Alomar – 444101583
Nawaf Alwazrah – 444101463

December 2025

Abstract

The increasing use of large language models to generate Arabic text has raised concerns regarding authorship attribution and the reliability of online content. Detecting AI-generated writing in Arabic remains challenging due to the language's rich morphology, irregular punctuation styles, and diverse writing forms across formal and informal contexts. This work investigates practical machine learning methods for distinguishing human-written and AI-generated Arabic text, emphasizing realistic dataset construction and stylistic variability.

Human samples were collected from online Arabic sources, including news platforms and user comments. AI text was generated by rephrasing these samples with a temperature of 1.0, after observing that topic-based generation produced unrealistically formal outputs and resulted in misleading model performance. Several detection models were evaluated, including a Naive Bayes baseline, AraBERT-v2 fine-tuning, a linguistic classifier using Farasa morphological features with XGBoost, and a character N-gram model. Results show that stylistic and structural cues provide stronger indicators than semantic features alone. A hybrid model combining N-gram patterns with linguistic attributes achieved the highest performance, reaching 93.16% on internal evaluation and maintaining strong generalization on external data from different sources and models.

1 Introduction

The rapid advancement of large language models has made it increasingly difficult to distinguish between human-written and machine-generated Arabic text.

Although global research has investigated AI-text detection in English, the task remains underexplored in Arabic due to its complex morphology, irregular punctuation patterns, and wide stylistic variation across formal and informal writing. These challenges make Arabic particularly sensitive to surface-level cues, making semantic-based detection approaches insufficient on their own.

This project investigates practical techniques for detecting AI-generated Arabic text using machine learning models. A dataset was constructed using real human writing collected from online sources, including formal news articles and informal YouTube comments. AI-generated samples were initially produced from multiple language models, but only outputs from Gemini were retained due to their more natural sentence structure and higher resemblance to human writing. To avoid unrealistic performance caused by topic-based generation, all AI text was created by rephrasing human samples with a temperature of 1.0, ensuring controlled content and natural stylistic variation.

Several modeling approaches were explored, including Naive Bayes (as a baseline test of task difficulty), transformer-based AraBERT fine-tuning, a linguistic classifier using Farasa morphological features with XGBoost, and a statistical model based on character N-grams. The results demonstrate that models relying on stylistic and structural features outperform semantic transformers. A hybrid approach that combines N-gram patterns with linguistic features achieves the strongest performance and remains effective on unseen data from external sources.

2 Literature Review

Automated detection of machine-generated text has gained importance with the growing use of large language models (LLMs) in academic, commercial, and online platforms. Early work focused on surface-level statistical indicators such as text perplexity, token repetition, and frequency-based irregularities, assuming that machine-generated text follows more predictable distributions than human writing. These approaches were initially effective, but their reliability decreases as modern models produce increasingly coherent and diverse content.

Recent advances introduced neural approaches that treat AI-text detection as a classification problem using pre-trained language models, including variants of BERT-like architectures and model-specific discriminators. These systems often improve accuracy due to their ability to capture higher-level linguistic representations. However, their predictions typically lack interpretability, offering binary outputs without explaining why a text is classified as human or machine-generated.

For Arabic, research remains limited. The lack of comprehensive datasets, the coexistence of Modern Standard Arabic with informal dialects, and the morphological richness of the language create additional challenges. Most available Arabic efforts rely on neural classifiers trained on small or domain-specific corpora, often focusing on plagiarism detection rather than modern LLM-generated text. Moreover, there is no widely adopted Arabic benchmark that combines formal and informal writing with contemporary generative outputs.

In summary, existing studies highlight progress in English detection but limited exploration for Arabic. Current methods, whether statistical or neural, tend to capture patterns associated with the current generation of language models, which means their effectiveness must be validated as models evolve. This positions AI-text detection as a practical, evaluation-driven task rather than a solved problem, especially for morphologically rich languages such as Arabic.

3 Methodology

3.1 Dataset Construction

The dataset contains human-written and machine-generated Arabic text. Human samples were collected from publicly available sources, including formal news articles and informal user comments from social media platforms and YouTube. These sources provide a range of writing styles, from structured Modern Standard Arabic (MSA) to dialectal and casual expressions.

Machine-generated samples were produced by rephrasing human texts using large language models to maintain topic diversity while introducing stylistic variation. A temperature value of 1.0 was applied to increase lexical variety. After manual filtering, only high-quality outputs produced by the Gemini model were retained.

In total, the dataset consists of 12,796 samples, evenly balanced between human-written and machine-generated text. A standard split of 70%, 20%, and 10% was used for training, validation, and testing, respectively, while preserving class balance across all partitions.

3.2 Modeling Approaches

Multiple modeling strategies were evaluated to detect AI-generated Arabic text. The objective was not only to compare performance levels, but also to assess whether shallow linguistic and statistical patterns can distinguish machine-generated writing without relying exclusively on deep transformer architectures.

3.2.1 Naive Bayes Baseline

A multinomial Naive Bayes classifier was first implemented as a sanity check to evaluate whether the task could be solved using a very simple model. Following the spirit of Occam's Razor, this step tests the possibility that a basic frequency-based classifier might already separate human and machine-written text, which would make more complex models unnecessary. Since Naive Bayes relies mainly on word distributions without capturing contextual or stylistic cues, its weak performance showed that the distinction is not trivially learnable. This result motivated the need for more expressive models in the subsequent experiments.

3.2.2 Character N-grams Classifier

A logistic regression model trained on TF-IDF character-level n-grams was used to capture statistical surface patterns such as punctuation habits, spacing, recurrent morphological fragments, and common token endings. Unlike word-level features, character-level units are robust to dialectal variation and spelling inconsistencies, which are frequent in informal Arabic writing. This approach demonstrated strong predictive capability, suggesting that statistical cues alone can reveal implicit structural regularities in machine-generated text.

3.2.3 Linguistic Features + XGBoost

To explore whether machine-generated text exhibits distinctive linguistic behavior, morphological features were extracted using Farasa segmentation, including part-of-speech ratios, average word length, determiners, punctuation usage, and the proportion of unknown tokens produced by the segmenter. The latter often reflects misspellings and informal or nonstandard forms that humans frequently produce but language models tend to avoid. These handcrafted indicators were modeled using an XGBoost classifier, enabling non-linear decision boundaries between human and AI writing styles. Although performance was lower than the n-gram model, this approach offered greater interpretability by linking predictions to observable stylistic differences in the text.

3.2.4 AraBERT Fine-Tuning

A transformer-based Arabic model (AraBERTv2) was fine-tuned for binary classification. By leveraging contextual representations, the model learns semantic and syntactic distinctions beyond surface statistics. This approach provided high-quality results, confirming that deep contextual modeling can effectively separate AI-generated content from human-written text. However, its predictions remain difficult to interpret, as the model does not explicitly reveal which linguistic cues determine classification.

3.2.5 Hybrid Model

Finally, a hybrid classifier was developed by combining character-level n-gram representations with the linguistic features extracted from Farasa. The resulting feature vector was trained using an XGBoost classifier, allowing both statistical and stylistic cues to contribute to the final decision. This approach yielded the highest overall performance while retaining partial interpretability, as the linguistic indicators can be traced to observable text characteristics. The hybrid design demonstrates that shallow and interpretable features can complement statistical signals without relying solely on deep contextual embeddings.

3.3 Preprocessing and Feature Extraction

Different preprocessing and feature extraction strategies were applied depending on the type of model. Character-based models relied primarily on surface text statistics, whereas the linguistic model used morphological analysis from Farasa, and the transformer model followed its own tokenization scheme.

3.3.1 Character-Level Models

For the Naive Bayes and character n-grams classifiers, the text was kept largely in its original form in order to preserve punctuation, spacing, and orthographic variation. Basic cleaning was limited to removing empty lines and obvious non-text artifacts, while retaining characters such as commas, full stops, question marks, and exclamation marks. This allowed the TF–IDF character n-gram representation to capture punctuation habits, common affixes, and local character patterns that differ between human and machine-generated writing.

3.3.2 Linguistic Feature Extraction with Farasa

For the linguistic model, features were extracted using a Farasa-based part-of-speech tagger applied directly to the raw Arabic text. Sentence boundaries were approximated by splitting on punctuation marks such as the period, question mark, and exclamation point, and the average sentence length (**avg_sentence_len**) was then computed as the average number of whitespace-separated tokens per sentence.

The tagged output from Farasa produces tokens in the form `word/TAG`. Each token was therefore separated into its lexical unit and its corresponding POS tag. A list of "clean" words was constructed by stripping off tag information. These clean words were used to compute two global lexical statistics:

- **avg_word_len**: the average word length in characters, obtained by dividing the total number of characters in all clean words by the number of clean words.
- **TTR_ratio**: the type–token ratio (lexical diversity), defined as the number of unique clean words divided by the total number of clean words.

Next, several POS-based ratios were computed by counting how often certain tag types appeared and normalizing by the total number of tokens. The feature set included:

- **NOUN_ratio**: counts tokens tagged as nouns or nominal forms (e.g., tags starting with `S`, or tags such as `NOUN` or `FOREIGN`).
- **VERB_ratio**: counts verbs (tags starting with `V`).
- **PART_ratio**: aggregates function words such as conjunctions, prepositions, pronouns, and other particles (e.g., `CONJ`, `PREP`, `PRON`, `H`, or tags beginning with `PART`).

- **ADJ_ratio**: counts adjectives.
- **NUM_ratio**: counts numerals and related suffixes (e.g., tags beginning with `NUM` or `NSUFF`).
- **PRON_ratio**: counts pronominal tags.
- **DET_ratio**: counts determiners.
- **PUNC_ratio**: counts punctuation tokens.

An additional **UNKNOWN_ratio** feature was computed by counting tokens that could not be assigned a valid POS tag. These cases typically arise from misspellings, creative spellings, dialectal words, or informal writing conventions that are more common in human text than in language-model output. All POS-based counts were normalized by the total number of tokens, and a simple **word_count** feature was retained to capture overall text length.

Together, these features form a compact stylometric profile of each sample. They capture differences in part-of-speech usage, lexical diversity, sentence structure, and punctuation behavior without relying on contextual embeddings. These engineered features were then used to train an XGBoost classifier, serving as an interpretable, feature-based component in our detection pipeline.

3.3.3 Transformer Model Preprocessing

For the AraBERTv2 model, preprocessing followed the standard requirements of the underlying tokenizer. The Arabic text was tokenized using the AraBERT-specific tokenizer, which applies its own normalization and subword segmentation. Special classification and separator tokens were added, and sequences were padded or truncated to a fixed maximum length to form uniform batches. The resulting token IDs and attention masks were then fed to the transformer model during fine-tuning for binary classification.

3.4 Evaluation Setup and Metrics

All models were first evaluated on the main dataset using a standard train–validation–test split of 70%, 20%, and 10%, respectively, with balanced class proportions in each split. The training set was used to fit the models, the validation set was used to select hyperparameters and compare approaches, and the final internal results were reported on the held-out test set only.

In addition to this internal evaluation, an external test set was constructed to assess generalization beyond the original data. This external set consisted of two new files: one containing fresh human-written Arabic texts scraped from different sources than the main dataset, and another containing AI-generated texts produced by several large language models, including systems not seen during training. Only the best-performing hybrid model was evaluated on this external set, and its performance is reported separately from the internal results.

For all evaluations, we report class-wise and overall performance using precision, recall, and F1-score, alongside overall accuracy. These metrics provide a more informative picture than accuracy alone, especially in scenarios where misclassifying human text as AI (or the opposite) carries different practical implications. Internal and external results are summarized in comparative tables in the Results section.

3.5 Implementation Overview

All experiments were conducted using Python, with data preparation, feature extraction, model training, and evaluation completed in an end-to-end pipeline. The implementation relied on widely adopted open-source libraries: `pandas` and `numpy` for data preprocessing, `nltk` and Farasa for morphological tagging, and `scikit-learn` and `XGBoost` for training the classification models.

To ensure practicality rather than theoretical experimentation, evaluations were conducted under realistic resource constraints. The full system was executed on a local Apple M3 Max workstation (48 GB RAM) and replicated on a Google Colab environment. No GPU acceleration was required, since all models were lightweight and trained rapidly even on CPU-only settings.

The complete pipeline consisted of four main stages:

1. **Dataset Construction:** scraping human-written Arabic text from online platforms, generating artificial text using LLMs, and filtering model outputs for coherence prior to inclusion.

2. **Preprocessing:** normalization, sentence segmentation, token cleaning, and part-of-speech tagging.

3. **Feature Engineering and Modeling:** extraction of stylometric and linguistic features using Farasa, followed by training three classifiers (Naive Bayes, XGBoost, and a Hybrid Ensemble combining the linguistic model with TF–IDF embeddings).

4. **Evaluation:** standard train–test validation (70/20/10 split) and external validation on a newly collected human dataset and LLM-generated outputs not seen during training.

This implementation reflects a practical pipeline for AI-content detection in Arabic, emphasizing reproducibility, feature interpretability, and robustness against unseen model outputs.

4 Results and Discussion

Table 1 summarizes internal performance across all models trained on the constructed dataset. The hybrid classifier, which combines character-level n-grams with stylometric features, achieved the strongest overall results, outperforming both deep transformer models and standalone linguistic or statistical approaches.

4.1 Internal Evaluation

Table 1: Internal evaluation metrics for all models on the Arabic AI-text detection task.

Model	**Human**			**AI**			**Acc.**	**Support**
	P	**R**	**F1**	**P**	**R**	**F1**		
Naive Bayes	0.57	0.44	0.49	0.55	0.67	0.61	0.5566	2560
Char N-grams (LogReg)	0.87	0.86	0.87	0.87	0.87	0.87	0.8699	2560
Farasa + XGBoost	0.85	0.80	0.82	0.77	0.83	0.80	0.8118	2875
AraBERTv2 (FT)	0.8552	0.8485	0.8519	0.8495	0.8562	0.8528	0.8523	1517
Hybrid (N-grams + Ling.)	0.94	0.92	0.93	0.92	0.94	0.93	0.9316	2560

To examine generalization beyond the primary dataset, an external evaluation was conducted using newly collected human texts and AI outputs from multiple large language models not seen during training. Table 2 reports the results of the hybrid classifier on this unseen data.

4.2 External Evaluation

Table 2: External evaluation of the hybrid model on unseen data.

Model	**Human**			**AI**			**Acc.**	**Support**
	P	**R**	**F1**	**P**	**R**	**F1**		
Hybrid (External Test)	0.91	0.80	0.85	0.82	0.93	0.87	0.86	348

4.3 Error Analysis

Finally, Table 3 shows the confusion matrix for the hybrid model on the internal test set, illustrating that most errors come from misclassifying human text as AI rather than the reverse.

Table 3: Confusion matrix of the hybrid model on the internal test set.

	Predicted Human (0)	**Predicted AI (1)**
Actual Human (0)	1163	102
Actual AI (1)	73	1222

4.4 Interpretability: Feature-Based Detection Signals

To understand why the hybrid model succeeds, a feature attribution analysis was conducted on both the linguistic component (Farasa features) and the character n-gram classifier. Table 4 shows the most influential hand-engineered signals used by XGBoost.

Table 4: Top linguistic feature influences in the hybrid model.

Feature	Importance Score (XGBoost)
avg_word_len	2166.0
PUNC_ratio	1732.0
PART_ratio	1661.0
NOUN_ratio	1454.0
word_count	1374.0
DET_ratio	1366.0
VERB_ratio	1314.0
ADJ_ratio	1139.0
NUM_ratio	1052.0

These linguistic indicators suggest that AI-generated Arabic tends to produce more consistent punctuation, longer average word lengths, and a higher proportion of determiners and adjectives, while informal or erroneous constructions are less frequent than in human-written text.

Complementing these linguistic cues, the character-level model highlights specific substrings that strongly differentiate AI and human writing. Table 5 summarizes the highest-weighted character n-grams extracted by the logistic regression model.

Table 5: Examples of strong character n-gram indicators for AI and human text.

AI-Associated N-grams (High Weight)	Human-Associated N-grams (Low/Negative Weight)
، (12.61)	ا (-4.63)
اً (6.03)	ه (-3.33)
. (5.87)	، (-3.22)
ة. (4.95)	و (-2.93)
أ (4.20)	. (-2.89)
! (3.69)	؟ (-2.86)
، م (2.77)	، (-2.78)
، ا (2.72)	- (-2.66)

This comparison shows that AI-generated Arabic tends to rely on clean punctuation (e.g., ،, .), formalized morphological endings (e.g., اً), and more structured clause openings (e.g., أ). In contrast, human writing often contains informal spacing, dialectal fragments, hesitation markers (e.g., و , ،), and misspellings, which increase the presence of low-confidence tokens in morphological tagging.

Overall, this interpretability confirms that the hybrid model does not rely solely on semantic understanding. Instead, its predictions stem from observable stylistic and statistical differences that are currently hard for machine-generated Arabic to fully mimic, particularly in informal discourse and noisy user-generated text.

5 Conclusion and Future Work

This work presented a practical investigation into detecting AI-generated Arabic text using a combination of stylometric, statistical, and transformer-based models. By constructing a balanced dataset of human-written and machine-generated content and systematically comparing diverse modeling strategies, the experiments demonstrated that Arabic AI-text detection does not require reliance on deep contextual architectures alone. The proposed hybrid model, which integrates character-level n-grams with interpretable linguistic indicators extracted from Farasa, achieved the highest overall performance internally and maintained strong generalization when evaluated on an external dataset containing text from previously unseen language models. These findings suggest that detectable distributional and stylistic asymmetries exist in current Arabic LLM outputs, and that shallow, interpretable models can exploit them effectively.

Nevertheless, the study revealed several limitations that stem both from modeling choices and from the evolving nature of generative systems. Attempts to incorporate perplexity as an additional feature did not improve performance and in some cases reduced classifier accuracy, likely due to the increasingly fluent distributions produced by modern LLMs. Furthermore, although watermarking techniques offer robust model-side traceability, they were not explored beyond preliminary review due to their dependence on proprietary architectures, API access, and computational resources unavailable in this project. These approaches remain impractical for open, detector-side solutions intended for diverse online usage.

The dataset itself, while balanced and sufficient for empirical evaluation, remains limited in scope relative to the rapidly growing diversity of Arabic text generation. Expanding the dataset to include more genres, dialectal variation, and unedited user queries would further stress-test detection methods but is not required to validate the main conclusions of this study. Instead, such expansion would primarily support stronger generalization against future Arabic models rather than correcting deficiencies in the current pipeline.

Future work should consider adaptive or online detection models that evolve alongside new text generators, mitigating the risk of detector obsolescence as newer systems erase stylistic artifacts. Finally, ethical considerations surrounding Arabic AI detection—particularly in educational and journalistic environments—call for solutions that are not only accurate but also interpretable, auditable, and compatible with transparent deployment policies.