MOCSA: Grapheme-Aware Orthographic Modeling for AI-Generated Text Detection in Indic Languages

Anonymous ACL submission

Abstract

The increasing fluency of large language models (LLMs) has intensified the need for reliable AI-generated text detection across multilingual settings. While existing detectors primarily rely on semantic or probabilistic signals, they are largely developed and evaluated in English-centric contexts, leaving orthographically rich Indic languages comparatively underexplored. In this work, we investigate whether script-level orthographic structure can improve multilingual AI-text detection for Indic languages. In this paper we propose **MOCSA**, a fusion-based framework that jointly models semantic and orthographic signals through a frozen multilingual semantic encoder (XLM-R) and a grapheme-aware orthographic encoder designed to capture Indic script regularities. Unlike prior approaches that predominantly rely on semantic representations, our method explicitly models grapheme-level structure, including conjunct formation and orthographic transitions characteristic of Indic writing systems. Experiments across 14 Indic languages and three multilingual generator families (Aya-8B, GPT-OSS, and Qwen2.5-7B) reveal that orthographic modeling substantially outperforms semantic-only detection, achieving an AUROC of 0.945 compared to 0.852 for semantic representations. Our proposed fusion framework further improves performance to an AUROC of 0.956, with statistically significant gains over orthographic-only modeling. These findings suggest that orthographic signals provide highly discriminative cues for Indic AI-generated text detection and highlight the importance of script-aware modeling in multilingual settings.

1 Introduction

The rapid proliferation of large language models (LLMs) has enabled scalable generation of fluent and increasingly human-like text across diverse domains, including journalism, education, social media, and conversational systems. While such capabilities unlock substantial opportunities, they also raise concerns regarding misinformation, authorship verification, academic integrity, and content authenticity (Solaiman et al., 2019; Wu et al., 2024). These concerns have motivated growing interest in automated AI-generated text detection, particularly as synthetic text becomes progressively harder to distinguish from human-authored content. Despite recent advances, existing detection methods remain predominantly centered on English and other high-resource languages, leaving multilingual and low-resource language settings comparatively underexplored.

Existing AI-text detection approaches broadly rely on semantic, contextual, or statistical signals to distinguish machine-generated text from human writing. Probability-based methods such as DetectGPT (Mitchell et al., 2023) and Fast-DetectGPT(Bao et al., 2024) exploit perturbation sensitivity and probability curvature in language model outputs , while supervised detectors commonly employ pretrained transformer encoders to model semantic regularities in generated text (Macko et al., 2023; Spiegel and Macko, 2024). Implicitly, these approaches assume that semantic inconsistencies or statistical irregularities are sufficient signals for identifying machine-generated content. However, such assumptions may not generalize effectively to Indic languages, which exhibit substantially richer orthographic structure through conjunct formation, grapheme composition, ligatures, and script-specific writing conventions (Ishida, 2002; Manodnya and Giri, 2023; Kakwani et al., 2020). Unlike Latin-script languages, meaningful writing units in Indic scripts frequently span multiple Unicode codepoints, potentially encoding discriminative orthographic cues that remain overlooked by semantic-centric detectors.

In this work, we investigate a simple but underexplored question: *can orthographic structure itself serve as a reliable signal for Indic AI-text detection?* Surprisingly, we observe that orthographic

modeling alone substantially outperforms multilingual semantic encoders across Indic languages. In our experiments, a grapheme-aware orthographic model achieves an AUROC of 0.945, markedly surpassing a semantic-only XLM-R (Conneau et al., 2020) baseline (0.852). This finding suggests that AI-generated Indic text exhibits detectable regularities in grapheme transitions, conjunct usage, spacing behavior, and orthographic normalization (Ansary et al., 2024) patterns that are not fully captured through semantic representations alone.

Motivated by this observation, we propose **MOCSA** (Multilingual Orthographic–Contextual Signal Architecture), a fusion-based framework for Indic AI-generated text detection that jointly models semantic and orthographic information. The framework combines a frozen multilingual semantic encoder based on XLM-R (Conneau et al., 2020) with a grapheme-aware orthographic encoder that explicitly models Indic script structure using Unicode grapheme (Bhogale et al., 2026; Manodnya and Giri, 2023) sequences. Both views are projected into a shared latent space and fused to produce robust document-level predictions. Rather than replacing semantic information, our goal is to investigate whether orthographic structure provides complementary signals for multilingual detection (Bhogale et al., 2026).

We evaluate **MOCSA** across 14 Indic languages spanning 10 Indic scripts and three multilingual generator families: Aya-8B (Dash et al.), GPT-OSS (Agarwal et al., 2025), and Qwen2.5-7B (Hui et al., 2024). Experimental results show that **MOCSA** achieves a test AUROC of **0.956**, substantially outperforming representative probability-based detectors and multilingual semantic baselines. Orthographic modeling alone achieves strong performance (**0.945** AUROC), consistently surpassing semantic representations and highlighting the importance of script-aware modeling for Indic AI-generated text detection. Combining semantic and orthographic signals through **MOCSA** yields further gains, indicating that semantic information provides complementary robustness despite orthographic signals dominating overall performance.

Specifically, our key contributions include:

1. The first systematic study investigating orthographic signals for AI-generated text detection in Indic languages.

2. The first empirical demonstration that grapheme-aware orthographic modeling substantially outperforms multilingual semantic-only representations for Indic AI-text detection.

3. **MOCSA**, a fusion-based architecture that jointly models multilingual semantic and orthographic signals through grapheme-aware fusion.

2 Related Work

Existing AI-generated text detectors broadly fall into two categories: *probability-based* and *supervised* methods. Probability-based approaches exploit statistical irregularities in generated text, often assuming token likelihood geometry or perturbation sensitivity transfers across languages and scripts. Representative examples include DetectGPT (Mitchell et al., 2023), Fast-DetectGPT (Bao et al., 2024), and related zero-shot methods leveraging perplexity, token cohesiveness, or log-rank statistics (Su et al., 2023; Ma and Wang, 2024; Dmitrii et al., 2024).

Supervised detectors instead formulate AI-text detection as a classification problem using pretrained language models such as RoBERTa, DeBERTa (He et al., 2020), and multilingual transformer encoders (Liu et al., 2019; Spiegel and Macko, 2024; Macko et al., 2023, 2025). While effective, these approaches primarily model semantic and contextual signals and are predominantly evaluated in English or other high-resource settings.

Multilingual pretrained language models such as mBERT (Devlin et al., 2019), XLM-R(Conneau et al., 2020), MuRIL(Khanuja et al., 2021), and IndicBERT have substantially advanced NLP for low-resource and multilingual languages (Doddapaneni et al., 2023). However, Indic languages exhibit orthographic properties that differ substantially from Latin-script languages, including conjunct formation, ligatures, virama-based composition, and grapheme clusters spanning multiple Unicode codepoints (Ishida, 2002; Kakwani et al., 2020). Prior work on character- and grapheme-level modeling has shown advantages for morphology-rich and script-sensitive settings, where visible writing units may otherwise be fragmented under naïve tokenization(Ansary et al., 2024; Bhogale et al., 2026).

To the best of our knowledge, no prior work explicitly investigates orthographic signals for Indic AI-generated text detection or combines grapheme-aware modeling with multilingual semantic repre-

sentations. We address this gap through **MOCSA**, a multi-view framework that jointly models semantic and orthographic information for Indic AI-text detection.

3 Methodology

This section introduces **MOCSA**, a novel framework for multilingual Indic AI-generated text detection. We first formulate the task in section 3.1 followed by the semantic and orthographic encoders in section 3.2 and 3.3 the fusion architecture in section 3.4, and training details in section 3.5

3.1 Problem Formulation

We formulate Indic AI-text detection as a document-level binary classification task. Given an input text sample $x = \{t_1, t_2, \ldots, t_n\}$, where t_i denotes the i-th textual unit in the sequence, the objective is to predict whether the text is human-written or machine-generated. Each input instance corresponds to a complete text sample and is assigned a binary label $y \in \{0, 1\}$, where $y = 0$ represents human-authored text and $y = 1$ denotes AI-generated text.

Unlike prior multilingual detection approaches that primarily rely on semantic representations, we model complementary semantic and orthographic representations of Indic text. Specifically, an input document x is transformed into two parallel representations: x^{sem}, capturing contextual multilingual semantics, and x^{orth}, capturing grapheme-level orthographic structure. Our proposed architecture, **MOCSA**, learns a joint representation from both views to estimate the probability of machine generation:

$$\hat{y} = f(x^{sem}, x^{orth}),$$

where $f(\cdot)$ denotes the proposed orthographic–contextual detection model and $\hat{y}$ is the predicted class probability.

3.2 Semantic Encoder

To capture multilingual contextual semantics, we employ a frozen `xlm-roberta-base` (Conneau et al., 2020) encoder as the semantic branch of **MOCSA**. Given an input text sample x, tokenization is performed using the XLM-R tokenizer with truncation and padding applied to a maximum sequence length of 256 tokens.

Rather than fine-tuning the transformer backbone, all XLM-R parameters are frozen during training, and only lightweight projection and classification layers are optimized. This design reduces computational overhead, mitigates overfitting, and enables a cleaner assessment of orthographic contributions through controlled ablations.

Let

$$H = \text{XLM-R}(x) \in \mathbb{R}^{T \times 768},$$

denote the final hidden representations produced by XLM-R, where T is the token sequence length. Following standard practice, we use the contextualized representation of the first token (`<s>`), equivalent to CLS pooling, as the semantic representation:

$$h_{\text{sem}} = H_{[0]}, \quad \text{where } h_{\text{sem}} \in \mathbb{R}^{768}.$$

To align semantic features with the orthographic branch, the representation is projected into a shared latent space through a lightweight projection head:

$$z_{\text{sem}} = \phi_{\text{sem}}(h_{\text{sem}}),$$

where $\phi_{\text{sem}}(\cdot)$ consists of a linear projection ($768 \rightarrow 128$), followed by Layer Normalization, ReLU activation, and dropout ($p = 0.2$). The resulting semantic representation is:

$$z_{\text{sem}} \in \mathbb{R}^{128}.$$

The semantic branch captures contextual multilingual information while remaining computationally efficient through frozen pretrained representations.

3.3 Orthographic Encoder

Unlike Latin-script languages, Indic scripts are inherently grapheme-centric, where visually meaningful units often consist of multiple Unicode characters composed through conjunct formation, diacritics, and script-specific orthographic rules. Consequently, character-level modeling may fragment meaningful script units and obscure orthographic patterns relevant to AI-text detection. To preserve script integrity, we model Indic text at the grapheme level.

Grapheme Extraction We represent text using Unicode extended grapheme clusters, which correspond more closely to visually interpretable writing units in Indic scripts (Ansary et al., 2024). For example, the Hindi word **क्षमता** is segmented as: **क्षमता** $\rightarrow$ **[क्ष, म, ता]**, illustrating how conjunct formations are preserved as single orthographic units.

Prior to segmentation, input text is normalized using Unicode NFC normalization to ensure canonical consistency across scripts. Grapheme extraction (Manodnya and Giri, 2023) is subsequently

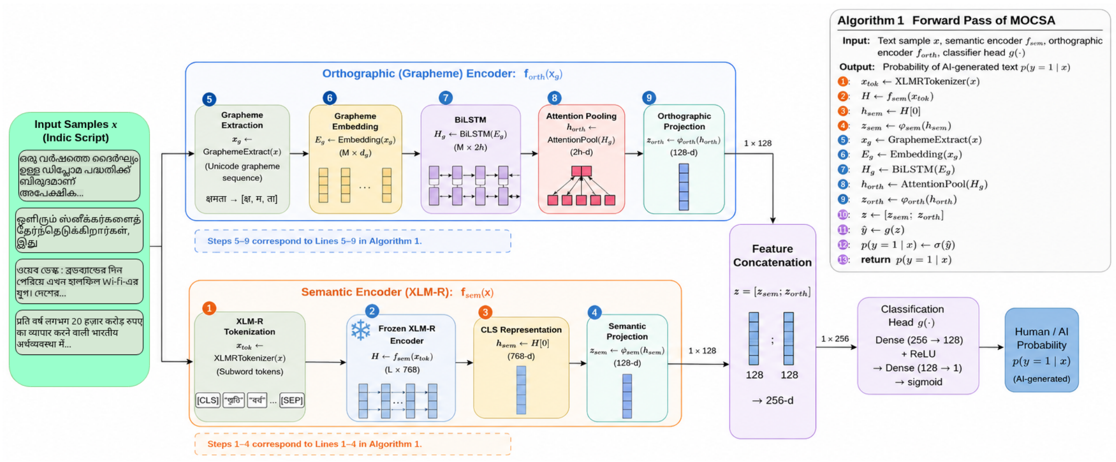

Figure 1: Architecture of **MOCSA**, a multi-view framework for Indic AI-generated text detection. The model jointly encodes semantic representations using a frozen XLM-R encoder and grapheme-aware orthographic representations using a BiLSTM with additive attention. Both views are projected into a shared latent space and fused through feature concatenation for document-level binary classification.

performed using Unicode-aware grapheme clustering(Ansary et al., 2024), while preserving whitespace information, as spacing itself may contribute to stylistic and orthographic regularities.

Our approach naturally supports script-specific conjunct formation across all supported Indic writing systems without requiring handcrafted language-specific tokenization rules.

Grapheme Vocabulary Let

$$g = \{g_1, g_2, \ldots, g_m\}$$

denote the grapheme sequence extracted from an input text sample, where g_i represents an individual grapheme cluster. A grapheme vocabulary comprising 14,157 unique symbols is derived solely from the training split and reused without modification during validation and testing, thereby preventing information leakage. Each grapheme is mapped to an integer index and embedded into a dense representation space. For computational consistency, grapheme sequences are truncated or padded to a fixed maximum length of 256 graphemes.

Orthographic Sequence Encoder Each grapheme token is first mapped to a trainable 128-dimensional embedding:

$$E(g_i) \in \mathbb{R}^{128}.$$

The embedded sequence is subsequently processed using a bidirectional Long Short-Term Memory (BiLSTM) network with hidden size 128 per direction. Given the grapheme sequence g, the BiLSTM produces contextualized hidden states:

$$H = \{h_1, h_2, \ldots, h_m\}, \quad \text{where } h_i \in \mathbb{R}^{256}.$$

Bidirectional recurrence enables the model to capture both preceding and succeeding grapheme dependencies, which are particularly important for conjunct-rich Indic scripts.

To aggregate sequence-level information, we employ learned additive attention over the BiLSTM hidden states:

$$e_i = W_2 \tanh(W_1 h_i),$$

$$\alpha_i = \frac{\exp(e_i)}{\sum_j \exp(e_j)},$$

where α_i denotes the normalized attention weight for the i-th grapheme.

The orthographic representation is then obtained as:

$$h_{\text{orth}} = \sum_{i=1}^{m} \alpha_i h_i.$$

Finally, the pooled representation is projected into a shared latent space:

$$z_{\text{orth}} = \phi_{\text{orth}}(h_{\text{orth}}),$$

where ϕ_{orth} consists of a linear projection (256 → 128), Layer Normalization, ReLU activation (Glorot et al., 2011), and dropout ($p = 0.2$). The resulting orthographic representation is:

$$z_{\text{orth}} \in \mathbb{R}^{128}.$$

The orthographic encoder is designed to capture script-level regularities, including grapheme transitions, conjunct consistency, spacing behavior, and orthographic normalization (Ansary et al., 2024) patterns that may differ between human- and machine-generated text.

3.4 MOCSA (Multilingual Orthographic–Contextual Signal Architecture))

Semantic and orthographic representations capture complementary but distinct properties of Indic text. While semantic representations encode contextual multilingual meaning, orthographic representations capture script-level regularities such as grapheme composition, conjunct usage, spacing patterns, and orthographic consistency. Motivated by the hypothesis that these signals jointly contribute to AI-generated text detection, **MOCSA** adopts a fusion-based architecture that explicitly integrates both representations through feature-level fusion.

Given an input text sample x, the semantic encoder produces a contextual representation $z_{\text{sem}} \in \mathbb{R}^{128}$, while the orthographic encoder generates a grapheme-aware representation $z_{\text{orth}} \in \mathbb{R}^{128}$. Prior to fusion, both modalities are projected into a shared latent space to ensure representational compatibility and mitigate modality imbalance arising from heterogeneous feature scales.

The fused representation is obtained through vector concatenation:

$$z = [z_{\text{sem}}; z_{\text{orth}}], \quad \text{where } z \in \mathbb{R}^{256}.$$

This concatenation preserves modality-specific information while enabling downstream layers to learn interactions between semantic and orthographic cues. The fused representation is subsequently passed through a lightweight classification head:

$$\hat{y} = f(z),$$

where $f(\cdot)$ consists of a fully connected layer $(256 \rightarrow 128)$, ReLU activation (Glorot et al., 2011), dropout ($p = 0.3$), and a final linear layer $(128 \rightarrow 1)$ producing a binary classification logit.

The final prediction probability is computed using a sigmoid transformation:

$$P(y = 1 \mid x) = \sigma(\hat{y}),$$

where $y = 1$ denotes machine-generated text.

Unlike prior multilingual detectors that predominantly rely on semantic information, **MOCSA** explicitly models orthographic structure as a complementary signal for Indic AI-generated text detection. This design allows the model to capture both contextual meaning and script-specific generation artifacts within a unified framework. For completeness, Appendix A.6 summarizes the forward inference pipeline of **MOCSA**.

3.5 Training Setup

Experiments are conducted on the IndicTextFake benchmark using the predefined train, validation, and test partitions containing 75,104, 9,264, and 9,340 samples, respectively. All models are trained for text-level binary classification, where each input corresponds to a complete text sample labeled as either human-authored text or machine-generated text, generated by Aya-8B (Dash et al.), GPT-OSS (Agarwal et al., 2025), and Qwen2.5-7B (Hui et al., 2024). Optimization is performed using AdamW (Loshchilov and Hutter, 2017) with an initial learning rate of 2×10^{-5}, weight decay of 0.01, and batch size 8. Learning rates are adaptively updated using `ReduceLROnPlateau` with a decay factor of 0.5 and patience of one epoch based on validation AUROC. During late-stage optimization of the fusion model, the learning rate is reduced to 5×10^{-6}. We optimize models using binary cross-entropy with logits (`BCEWithLogitsLoss`). Semantic-only and orthographic-only variants are trained for up to 15 epochs, while **MOCSA** is trained for up to 25 epochs with early stopping (patience = 3) based on validation AUROC.

4 Experiments

This section evaluates the effectiveness of **MOCSA** for multilingual Indic AI-generated text detection compared with baselines constituting Semantic-only detector and Orthrographic-only detector

4.1 Experimental Setup

Dataset. We evaluate **MOCSA** using IndicTextFake, a multilingual benchmark for AI-generated text detection spanning fourteen Indic languages from both Indo-Aryan and Dravidian language families, including Assamese, Bengali, Gujarati, Hindi, Kannada, Maithili, Malayalam, Marathi, Odia, Punjabi, Sanskrit, Tamil, Telugu, and Urdu. The benchmark covers diverse Indic writing systems, including Devanagari, Bengali-Assamese, Gujarati, Gurmukhi, Kannada,

Malayalam, Odia, Tamil, Telugu, and Perso-Arabic scripts.

Human-authored text is sourced from IndicCorp v2 (Doddapaneni et al., 2023), while machine-generated samples are produced using three multilingual generators: Aya-8B, GPT-OSS, and Qwen2.5-7B. We use the predefined train, validation, and test splits consisting of 127,384, 15,639, and 14,783 samples, respectively.

To evaluate robustness under lexical and surface-form variation, the benchmark additionally includes back-translation-based paraphrastic perturbations of machine-generated text. Grapheme vocabularies are constructed exclusively from the training split to prevent information leakage. All experiments are formulated as text-level binary classification between human- and machine-generated text.

Metric Following prior work on AI-text detection, we report Area Under the Receiver Operating Characteristic Curve (AUROC) as the primary evaluation metric due to its threshold-independent assessment of binary discrimination performance.

Task setup We evaluate whether grapheme-aware orthographic signals improve multilingual AI-generated text detection for Indic languages. Specifically, we compare three modeling paradigms: (i) a semantic-only detector based on multilingual contextual representations, (ii) an orthographic-only detector operating over grapheme sequences, and (iii) **MOCSA**, which jointly models semantic and orthographic information through feature-level fusion.

We formulate detection as a text-level binary classification task, where each input corresponds to a single complete text sample and is classified as either human-authored or machine-generated. Inputs are tokenized using XLM-R with a maximum sequence length of 256 tokens, while grapheme sequences are truncated or padded to 256 symbols. Grapheme vocabularies are constructed exclusively from the training split to prevent vocabulary leakage into validation and test data.

4.2 Detection Performance

Table 1 summarizes overall detection performance across representative baseline detectors and proposed model variants.

Probability-based detectors transfer poorly to Indic AI-generated text detection, with DetectGPT and FastDetectGPT performing near random chance and MuRIL-based PLL achieving only modest gains. In contrast, multilingual semantic modeling using frozen XLM-R substantially improves performance, reaching an AUROC of 0.852. Orthographic modeling yields a large additional improvement, increasing AUROC to 0.945 and indicating that grapheme-level structure provides highly discriminative signals for Indic AI-generated text detection. Combining semantic and orthographic representations through **MOCSA** further improves performance to 0.956 AUROC, suggesting that semantic information contributes complementary signal despite orthographic representations dominating overall performance. Figure 2 further illustrates this trend, where semantic-only representations exhibit substantially weaker separability than orthographic and fusion-based models.

Model	**Test AUROC**
DetectGPT	0.497
FastDetectGPT	0.493
PLL (MuRIL)	0.608
XLM-R (Frozen)	0.852
Orthographic Only	0.945
Fusion (**MOCSA**)	**0.956**

Table 1: Overall detection performance across representative baseline detectors and proposed model variants.

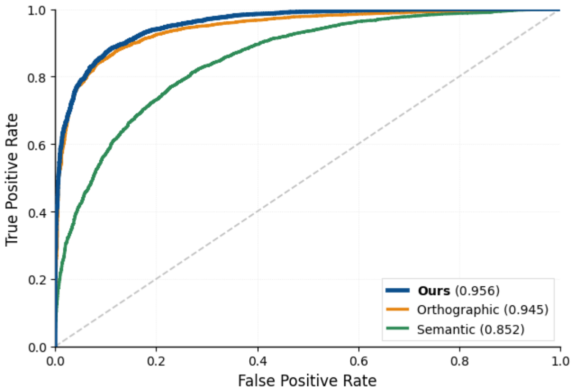

Figure 2: ROC curves for semantic-only, orthographic-only, and fusion-based detection models.

4.3 Language-wise Analysis

To evaluate multilingual robustness, we analyze performance across all fourteen Indic languages. Figure 3 summarizes language-wise AUROC for semantic, orthographic, and fusion-based models.

Across nearly all languages, orthographic and fusion-based approaches substantially outperform semantic-only detection. Improvements are especially pronounced in orthographically rich scripts

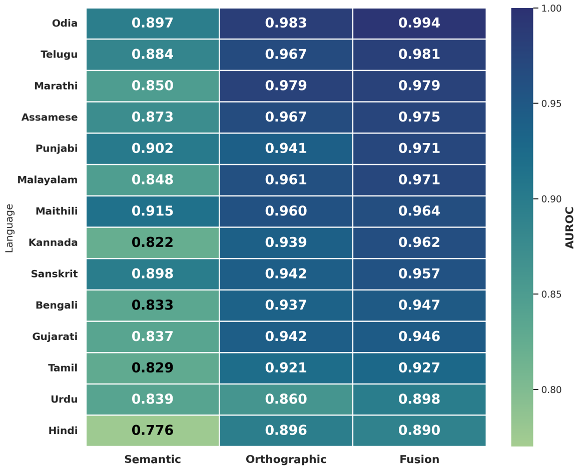

Figure 3: Language-wise AUROC across semantic, orthographic, and fusion models.

Generator	Semantic	Orthographic	Fusion
Aya-8B	0.832	0.907	0.948
GPT-OSS	0.859	**0.965**	0.961
Qwen2.5	0.860	0.951	**0.958**

Table 2: Generator-wise detection performance across multilingual generators.

Comparison	Δ AUROC	95% CI
Fusion vs Orthographic	+0.0117	[0.0074, 0.0162]
Fusion vs Semantic	+0.1044	[0.0969, 0.1117]

Table 3: Bootstrap confidence intervals for AUROC improvements.

such as Odia, Telugu, Marathi, and Assamese, where fusion performance exceeds 0.97 AUROC. For example, Odia achieves an AUROC of 0.994 under feature-level fusion, suggesting that grapheme-level regularities provide highly discriminative signals for AI-generated text detection.

Interestingly, orthographic-only modeling remains highly competitive with the fusion-based approach in several languages. In Marathi, orthographic modeling achieves an AUROC of 0.979, comparable to fusion performance (0.979). Similarly, GPT-OSS-generated text exhibits marginally stronger orthographic than fusion performance. These findings suggest that orthographic signals frequently dominate semantic representations in Indic AI-text detection, while semantic information contributes complementary gains in aggregate performance.

Performance is comparatively weaker for Hindi and Urdu, where fusion achieves AUROC scores of 0.890 and 0.898, respectively. We hypothesize that greater stylistic diversity, higher corpus heterogeneity, and stronger multilingual generator familiarity with these languages may reduce separability between human and machine-generated text.

4.4 Generator-wise Generalization

We next examine robustness across different multilingual text generators. Table 2 reports AUROC across Aya-8B, GPT-OSS, and Qwen2.5.

The proposed framework generalizes consistently across multilingual generators, achieving strong performance in all evaluated settings. Qwen2.5 and GPT-OSS are comparatively easier to detect, with fusion AUROC values of 0.958 and 0.961, respectively, whereas Aya-8B presents a more challenging detection setting (0.948).

Interestingly, orthographic modeling alone achieves the strongest performance for GPT-OSS-generated text, further reinforcing the importance of grapheme-level regularities in generated Indic text. We hypothesize that stronger multilingual generators may produce more natural semantic content while still exhibiting detectable orthographic regularities. Although evaluation spans multiple multilingual generators, our experiments do not explicitly assess unseen-generator transfer, which we leave for future work.

4.5 Statistical Significance

To determine whether improvements from fusion-based approach are statistically meaningful, we estimate bootstrap confidence intervals over AUROC differences.

As shown in Table 3, fusion significantly improves over both semantic-only and orthographic-only baselines. Although the gain over orthographic modeling is relatively modest (+0.0117 AUROC), the confidence interval excludes zero, indicating that semantic information contributes statistically significant complementary information despite orthographic signals dominating overall performance.

4.6 Robustness to Paraphrastic Perturbations

We evaluate robustness under lexical and surface-form variation using two back-translation perturbation settings (*P1* and *P2*) that preserve semantic content while modifying textual realization.

Table 4 shows that **MOCSA** remains consistently strongest under paraphrastic perturbation, achieving AUROC scores of 0.937 and 0.921 under P1 and P2, respectively. While orthographic-only performance

Model	Clean	P1	P2	Δ(P1)	Δ(P2)
Semantic Only	0.852	0.848	0.833	-0.004	-0.020
Orthographic Only	0.945	0.890	0.876	-0.054	-0.069
Fusion (**MOCSA**)	**0.956**	**0.937**	**0.921**	-0.020	-0.036

Table 4: Robustness under paraphrastic perturbations.

degrades more substantially under paraphrasing, semantic representations remain comparatively stable.

These findings suggest a complementary trade-off between robustness and discriminative power: orthographic representations provide strong detection signals, whereas semantic representations contribute stability under lexical variation. By combining both views, **MOCSA** maintains strong performance under surface-form changes, indicating that the model does not rely solely on brittle orthographic artifacts.

4.7 Error Analysis

To better understand model behavior, we qualitatively examine representative failure cases of the fusion model. False positives primarily occur for highly structured human-authored documents, particularly institutional, informational, and news-style writing. Such texts exhibit strong stylistic regularity and stable orthographic patterns that occasionally resemble machine-generated outputs. For example, Bengali and Hindi informational articles are sometimes assigned high AI probabilities, suggesting that highly regular orthographic structure may resemble generation artifacts.

In contrast, false negatives commonly arise from narrative, encyclopedic, or conversational AI-generated content, particularly from Qwen2.5 and GPT-OSS. These outputs often display comparatively natural grapheme transitions and reduced surface-level artifacts, making them harder to distinguish from human-authored text.

Overall, these findings reinforce the central hypothesis of this work: orthographic signals are highly discriminative for Indic AI-generated text detection, but their effectiveness depends on the extent to which generated text preserves natural grapheme-level regularities. Figure 4 further examines model behavior across document lengths.

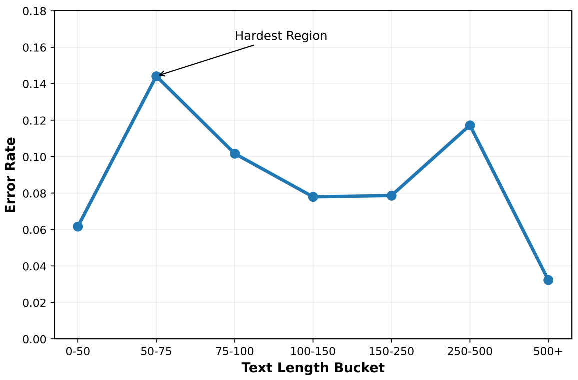

Figure 4: Prediction performance across text lengths.

5 Conclusion

In this work, we investigated the role of orthographic structure in multilingual AI-generated text detection for Indic languages. Our findings show that grapheme-aware orthographic representations provide substantially stronger discriminative signals than semantic-only multilingual encoders, highlighting the importance of script-aware modeling for orthographically rich writing systems. Motivated by this observation, we proposed **MOCSA**, a multi-view framework that jointly models semantic and orthographic information through frozen multilingual representations and grapheme-aware sequence modeling. Robustness analysis further suggests that semantic representations contribute stability under lexical variation, while orthographic signals retain strong discriminative power, making their combination particularly effective for multilingual detection. More broadly, our results indicate that assumptions underlying English-centric AI-text detection do not transfer directly to Indic languages, emphasizing the importance of explicit orthographic modeling for multilingual and script-diverse settings.

Potential Risks. AI-generated text detectors may produce false positives, potentially misclassifying human-authored content as machine-generated, particularly in educational, journalistic, or authorship-sensitive settings. Performance may further vary across domains, languages, and generator families, necessitating human oversight in high-stakes deployment scenarios. As with other automated detection systems, predictions should therefore be interpreted as assistive signals rather than definitive evidence of authorship.

Limitations. The primary limitation of our work is that our findings are specific to Indic languages and may not generalize directly to other multilingual or orthographically complex language families. However, since existing detectors do not account for Indic-specific cases, this limitation itself may pave

the way for future research. Moreover, robustness is evaluated only under paraphrastic perturbations; broader adversarial settings, such as style transfer or human-guided rewriting, remain to be explored.

References

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, and 1 others. 2025. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*.

Nazmuddoha Ansary, Quazi Adibur Rahman Adib, Tahsin Reasat, Asif Shahriyar Sushmit, Ahmed Imtiaz Humayun, Sazia Mehnaz, Kanij Fatema, Mohammad Mamun Or Rashid, and Farig Sadeque. 2024. Unicode normalization and grapheme parsing of indic languages. In *Proceedings of the 2024 joint international conference on computational linguistics, language resources and evaluation (LREC-COLING 2024)*, pages 17019–17030.

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. 2024. Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature. In *International Conference on Learning Representations*, volume 2024, pages 24814–24836.

Kaushal Santosh Bhogale, Tahir Javed, Greeshma Susan John, Dhruv Rathi, Akshayasree Padmanaban, Niharika Parasa, and Mitesh M Khapra. 2026. Towards orthographically-informed evaluation of speech recognition systems for indian languages. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 17092–17096. IEEE.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 8440–8451.

Saurabh Dash, Yiyang Nan, John Dang, Arash Ahmadian, Shivalika Singh, Madeline Smith, Bharat Venkitesh, Vlad Shmyhlo, Viraat Aryabumi, Walter Beller-Morales, and 1 others. Aya vision: Advancing the frontier of multilingual multimodality, 2025. *URL https://arxiv. org/abs/2505.08751*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Poimanov Dmitrii, Leonid Mestetsky, and Andrey Grabovoy. 2024. N-gram perplexity-based ai-generated text detection. In *2024 Ivannikov Ispras Open Conference (ISPRAS)*, pages 1–8. IEEE.

Sumanth Doddapaneni, Rahul Aralikatte, Gowtham Ramesh, Shreya Goyal, Mitesh M Khapra, Anoop Kunchukuttan, and Pratyush Kumar. 2023. Towards leaving no indic language behind: Building monolingual corpora, benchmark and models for indic languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12402–12426.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323. JMLR Workshop and Conference Proceedings.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, and 1 others. 2024. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*.

Richard Ishida. 2002. An introduction to indic scripts. In *Proceedings of the 22nd Int. Unicode Conference*, volume 8.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul NC, Avik Bhattacharyya, Mitesh M Khapra, and Pratyush Kumar. 2020. Indicnlpsuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for indian languages. In *Findings of the association for computational linguistics: EMNLP 2020*, pages 4948–4961.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, and 1 others. 2021. Muril: Multilingual representations for indian languages. *arXiv preprint arXiv:2103.10730*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Shixuan Ma and Quan Wang. 2024. Zero-shot detection of llm-generated text using token cohesiveness. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17538–17553.

Dominik Macko, Jakub Kopal, Robert Moro, and Ivan Srba. 2025. Multisocial: Multilingual benchmark of machine-generated text detection of social-media texts. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 727–752.

Dominik Macko, Robert Moro, Adaku Uchendu, Jason Lucas, Michiharu Yamashita, Matúš Pikuliak, Ivan Srba, Thai Le, Dongwon Lee, Jakub Simko, and 1 others. 2023. Multitude: Large-scale multilingual machine-generated text detection benchmark. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9960–9987.

KH Manodnya and Animesh Giri. 2023. Orthographic syllable pair encoding for language modelling tasks in indic languages. In *2023 IEEE MIT Undergraduate Research Technology Conference (URTC)*, pages 1–6. IEEE.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International conference on machine learning*, pages 24950–24962. PMLR.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, and 1 others. 2019. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*.

Michal Spiegel and Dominik Macko. 2024. Kinit at semeval-2024 task 8: Fine-tuned llms for multilingual machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 558–564.

Jinyan Su, Terry Zhuo, Di Wang, and Preslav Nakov. 2023. Detectllm: Leveraging log rank information for zero-shot detection of machine-generated text. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12395–12412.

Junchao Wu, Runzhe Zhan, Derek F Wong, Shu Yang, Xinyi Yang, Yulin Yuan, and Lidia S Chao. 2024. Detectrl: Benchmarking llm-generated text detection in real-world scenarios. *Advances in Neural Information Processing Systems*, 37:100369–100401.

A Implementation Details and Reproducibility

A.1 Training Hyperparameters

Table 5 summarizes the training configuration used across experiments.

Hyperparameter	Value
Optimizer	AdamW
Initial Learning Rate	2×10^{-5}
Fusion LR (late stage)	5×10^{-6}
Weight Decay	0.01
Batch Size	8
Scheduler	ReduceLROnPlateau
Loss Function	BCEWithLogitsLoss
Early Stopping	3 epochs
Max Token Length	256
Max Grapheme Length	256
Grapheme Vocabulary Size	14,157
Semantic Encoder	XLM-R Base
Orthographic Encoder	BiLSTM + Attention

Table 5: Training configuration and hyperparameters.

A.2 Computational Resources

Experiments were conducted on a single NVIDIA RTX 3060 GPU (12 GB VRAM). The semantic branch employs `xlm-roberta-base` (∼278M parameters), while the orthographic encoder consists of a lightweight BiLSTM-based architecture with substantially fewer trainable parameters.

Training required approximately 3.1 GPU hours for the semantic-only model, 0.4 GPU hours for the orthographic-only model, and 5.4 GPU hours for the fusion model. The fusion model was trained in two stages totaling 25 epochs. Freezing the semantic backbone substantially reduced computational cost by restricting optimization to lightweight projection and classification layers.

A.3 Model Capacity and Trainable Parameters

Table 6 summarizes parameter counts across model variants. Since the semantic backbone is frozen during training, only lightweight projection and classification layers are optimized in the semantic and fusion models, substantially reducing trainable capacity relative to full transformer fine-tuning.

Model	Total Params	Trainable Params
Semantic Only	278.4M	0.39M
Orthographic Only	4.5M	4.5M
Fusion (**MOCSA**)	280.4M	2.31M

Table 6: Model capacity and trainable parameter counts.

Despite incorporating a pretrained multilingual encoder, the frozen semantic backbone keeps optimization lightweight, with only a small fraction of parameters updated during training. This design reduces computational cost while enabling clearer isolation of orthographic contributions.

Algorithm 1 Forward Pass of **MOCSA**

Input: Text sample x, semantic encoder f_{sem}, orthographic encoder f_{orth}, classifier head $g(\cdot)$
Output: Probability of AI-generated text $p(y = 1 \mid x)$

$x_{\text{tok}} \leftarrow \text{XLMRTokenizer}(x)$
$H \leftarrow f_{\text{sem}}(x_{\text{tok}})$
$h_{\text{sem}} \leftarrow H_{[0]}$
$z_{\text{sem}} \leftarrow \phi_{\text{sem}}(h_{\text{sem}})$
$x_g \leftarrow \text{GraphemeExtract}(x)$
$E_g \leftarrow \text{Embedding}(x_g)$
$H_g \leftarrow \text{BiLSTM}(E_g)$
$h_{\text{orth}} \leftarrow \text{AttentionPool}(H_g)$
$z_{\text{orth}} \leftarrow \phi_{\text{orth}}(h_{\text{orth}})$
$z \leftarrow [z_{\text{sem}}; z_{\text{orth}}]$
$\ell \leftarrow g(z)$
$p(y = 1 \mid x) \leftarrow \sigma(\ell)$
return $p(y = 1 \mid x)$

A.4 Software and Implementation Details

Experiments were implemented in PyTorch using the HuggingFace `Transformers` library for XLM-R-based semantic encoding. Unicode grapheme segmentation was performed using Unicode-aware grapheme clustering with NFC normalization to preserve script-specific orthographic structure.

Experiments were conducted using Python 3.12.3, PyTorch 2.10.0 with CUDA 12.8 support, and `transformers` version 4.41.2.

A.5 Reproducibility

We report architectural details, hyperparameters, training settings, and evaluation protocols necessary for reproducibility. Additional implementation details, robustness analyses, and qualitative error cases are provided in the appendix to facilitate independent verification.

A.6 Inference Pipeline of MOCSA

For completeness, Algorithm 1 summarizes the forward inference procedure of **MOCSA**, illustrating how semantic and orthographic representations are jointly computed and fused for document-level prediction.

A.7 Data and Artifact Usage

This work uses publicly available multilingual corpora and pretrained language models in accordance with their intended research usage conditions. Experiments additionally rely on a multilingual benchmark for Indic AI-generated text detection developed for research purposes. Dataset characteristics relevant to evaluation are described in the main paper.

Language	Sem.	Orth.	Fusion
Odia	0.897	0.983	**0.994**
Telugu	0.884	0.967	**0.981**
Marathi	0.850	**0.979**	0.979
Assamese	0.873	0.967	**0.975**
Punjabi	0.902	0.941	**0.971**
Malayalam	0.848	0.961	**0.971**
Maithili	0.915	0.960	**0.964**
Kannada	0.822	0.939	**0.962**
Sanskrit	0.898	0.942	**0.957**
Bengali	0.833	0.937	**0.947**
Gujarati	0.837	0.942	**0.946**
Tamil	0.829	0.921	**0.927**
Urdu	0.839	0.860	**0.898**
Hindi	0.776	**0.896**	0.890

Table 7: Language-wise AUROC across semantic(Sem.), orthographic (Orth.), and fusion-based models.

A.8 AI Assistance Disclosure

AI assistants were used for language refinement and manuscript editing. Scientific claims, experimental design, implementation, and result verification were conducted by the authors.

B Additional Experimental Results

B.1 Language-wise Detection Performance

Table 7 reports language-wise AUROC across semantic, orthographic, and fusion-based models. Consistent with findings in the main paper, orthographic and fusion-based models substantially outperform semantic-only detection across most Indic languages. Improvements are especially pronounced in orthographically rich scripts such as Odia, Telugu, and Marathi.

B.2 Prediction Confidence Analysis

To assess prediction behavior beyond threshold-independent metrics, we analyze the distribution of prediction confidence for human- and machine-generated samples. Table 8 summarizes class-wise prediction statistics for the fusion model.

Class	Mean Score	Std. Dev.
Human-authored	0.213	0.268
Machine-generated	0.883	0.209

Table 8: Prediction confidence statistics of the fusion model.

The model exhibits strong confidence separation between human- and machine-generated text, with machine-generated samples receiving substantially higher prediction scores on average. The strong separation between confidence distributions suggests

that **MOCSA** learns stable discriminative boundaries rather than relying on threshold-specific behavior.

B.3 Generator-wise Robustness under Paraphrastic Perturbation

Table 9 reports generator-specific robustness under paraphrastic perturbations. Across all perturbation settings, the fusion model consistently achieves the strongest performance, indicating improved robustness under lexical and surface-form variation.

B.4 Performance Across Text Lengths

We further analyze prediction performance across text lengths to assess whether detection quality is sensitive to document size. Table 10 summarizes error rates across length bins.

Performance remains comparatively stable across shorter and longer text samples, suggesting that **MOCSA** does not rely solely on long-range context and remains effective across varying document lengths.

B.5 Qualitative Error Analysis

Tables 13 and 14 present representative failure cases of the fusion model.

False positives frequently correspond to highly structured human-authored text, including institutional, informational, and news-like writing exhibiting strong stylistic regularity. In contrast, false negatives commonly arise from narrative or encyclopedic AI-generated content with comparatively natural orthographic flow.

B.6 Language-wise Error Analysis

Table 11 reports language-wise prediction error rates for the fusion model. Performance varies moderately across languages, with Urdu and Hindi presenting the highest error rates, while Odia and Telugu exhibit the strongest performance.

The higher error rates observed for Hindi and Urdu suggest comparatively weaker separability between human- and machine-generated text, potentially due to broader lexical diversity and stylistic overlap. In contrast, stronger performance on Odia and Telugu suggests that orthographic regularities may be more consistently preserved across generated content.

B.7 Generator-wise Error Analysis

Table 12 summarizes prediction accuracy across generator families. GPT-OSS outputs are detected most reliably, whereas Aya-8B presents the most challenging setting among evaluated generators.

Detection difficulty varies across generators, with Aya-8B producing comparatively harder-to-distinguish outputs. Nevertheless, performance remains consistently strong across generator families, indicating that orthographic cues generalize across multilingual generation styles.

Generator	Perturbation	Semantic	Orthographic	Fusion
Aya-8B	P1	0.831	0.854	**0.941**
Aya-8B	P2	0.802	0.829	**0.902**
Qwen2.5	P1	0.825	0.877	**0.920**
Qwen2.5	P2	0.822	0.874	**0.917**
GPT-OSS	P1	0.876	0.924	**0.946**
GPT-OSS	P2	0.861	0.909	**0.935**

Table 9: Generator-wise robustness under paraphrastic perturbations.

Length Bin	Error Rate	Accuracy
35–52	0.085	0.915
52–65	0.155	0.845
65–93	0.101	0.899
93–684	0.083	0.917

Table 10: Prediction performance across document length ranges.

Language	Error Rate	Accuracy
Odia	0.040	0.960
Telugu	0.061	0.939
Assamese	0.074	0.926
Kannada	0.079	0.921
Marathi	0.079	0.921
Malayalam	0.088	0.912
Punjabi	0.090	0.910
Maithili	0.092	0.908
Bengali	0.112	0.888
Sanskrit	0.115	0.885
Gujarati	0.132	0.868
Tamil	0.150	0.850
Hindi	0.170	0.830
Urdu	0.170	0.830

Table 11: Language-wise prediction error rates of the fusion model.

Generator	Error Rate	Accuracy
GPT-OSS	0.065	0.935
Qwen2.5	0.081	0.919
Aya-8B	0.099	0.901
Human	0.164	0.836

Table 12: Generator-wise prediction error rates of the fusion model.

Language	Language Family	Snippet	True Label	Prediction
Hindi	Indo-Aryan	Ganatantra divas per bhashan 2021 – गणतंत्र दिवस पर दमदार भाषण 26 जनवरी – नमस्कार दोस्तों आपका स्वागत है हमारे हिंदी ब्लॉग मे और आज मैं आपके लिए ले के आया हूँ 26 january par bhashan 2021 और मई उम्मीद करूंगा कि आपको यह प्यारा गणतंत्र दिवस पर भाषण 2021 पसंद आएगा और अगर आपको यह ganatantra diwas speech in hindi 2021 पसंद आए तो आप इस 26 जनवरी भाषण 2021 को अपने स्कूल अथवा कालेज मे जरूर सांझा करें और 26 जनवरी गणतंत्र दिवस 2021 के दिन अपने विचार अपने देशवासियों के साथ साँझ करे । तो दोस्तों बिना आपका समय लिए शुरू करते है इस प्यारे gantantra diwas bhahan 2021 को और अगर आपको ये republic day speech in hindi पसंद आए तो इस पोस्ट को अपने दोस्तों के साथ साँझान जरूर करें। छलिये शुरू करते हैं bhashan 26 january and 26 bhashan in hindi	0	0.996
Maithili	Indo-Aryan	उत्तर. एनटीएमकेर लक्ष्य सभमेसँ एक मुख्य अछि अनुवादक शिक्षण, जे एहि क्रियाकलाप हेतु विशेष प्रशिक्षणक रुपमे आवश्यक अछि। एनटीएम सहज बनाओत-संक्षिप्त अवधिकेर प्रशिक्षण कार्यक्रमक आयोजन द्वारा भावी अनुवादक, अनुवादक हेतु पाठ्यक्रम मापदण्ड आ संपुष्टिक परिस्थितिक प्रादुर्भाव, अनुवाद तकनीकमे विशिष्ट पाठ्यक्रमक विकासमे उत्साह, समर्थन आ सहायता, शोध परियोजनाक संबर्द्धन, अध्येतावृत्ति कार्यक्रमक संगठन, कार्यशालाक आयोजन आओर पुनरीक्षण, सम्पादन आ कॉपी-सम्पादन अनुवादमे अनुवादककेँ जोड़बाजन्य आओरो बेसी अङ्गीकार करबामे।	0	0.996
Kannada	Dravidian	ಪ್ರಚಾರದ ಎನ್ಜಿಒಗಳು "ರಾಜಕೀಯ ವ್ಯವಸ್ಥೆಯ ಪ್ರಭಾವದ ಮೂಲಕ ಪರೋಕ್ಷವಾಗಿ ಉತ್ತೇಜಿಸಲ್ಪಟ್ಟ ದೊಡ್ಡ-ಪ್ರಮಾಣದ ಬದಲಾವಣೆಯನ್ನು ಸಾಧಿಸಲು" ಪ್ರಯತ್ನಿಸುತ್ತವೆ. ಅವರಿಗೆ ಸಕ್ರಿಯ, ಪರಿಣಾಮಕಾರಿ ವೃತ್ತಿಪರ ಸದಸ್ಯರ ಅಗತ್ಯವಿರುತ್ತದೆ, ಅವರು ಬೆಂಬಲಿಗರಿಗೆ ಮಾಹಿತಿ ಮತ್ತು ಪ್ರೇರಣೆ ನೀಡಬಹುದು. ಪ್ರಚಾರ ಮಾಡುವ ಎನ್‌ಜಿಒಗಳು ಮಾಧ್ಯಮಗಳನ್ನು, ಅವುಗಳ ವ್ಯಾಖ್ಯಾನಿಸುವ ಚಟುವಟಿಕೆಯನ್ನು ಆಕರ್ಷಿಸುವ ಪ್ರದರ್ಶನಗಳು ಮತ್ತು ಘಟನೆಗಳನ್ನು ಯೋಜಿಸಬೇಕು ಮತ್ತು ಆಯೋಜಿಸಬೇಕು. ಪ್ರಚಾರ ಮಾಡುವ ಎನ್ಜಿಒಗಳು ಸಾಮಾನ್ಯವಾಗಿ ಮಾನವ ಹಕ್ಕುಗಳು, ಮಹಿಳಾ ಹಕ್ಕುಗಳು ಮತ್ತು ಮಕ್ಕಳ ಹಕ್ಕುಗಳಿಗೆ ಸಂಬಂಧಿಸಿದ ಸಮಸ್ಯೆಗಳೊಂದಿಗೆ ವ್ಯವಹರಿಸುತ್ತವೆ ಮತ್ತು ಅವರ ಪ್ರಾಥಮಿಕ ಉದ್ದೇಶವು ಒಂದು ನಿರ್ದಿಷ್ಟ ಕಾರಣವನ್ನು ರಕ್ಷಿಸುವುದು (ಅಥವಾ ಉತ್ತೇಜಿಸುವುದು).	0	0.993
Bengali	Indo-Aryan	রেলপথ মন্ত্রণালয়ের প্রকল্প সম্পর্কে মন্ত্রী বলেন, অনেক প্রকল্প চলমান আছে এবং অনেক প্রকল্প হাতে নেওয়া হয়েছে। পুরনো লাইন ও ব্রিজ সংস্কার করা হচ্ছে। নতুন নতুন কোচ এবং ইঞ্জিন নিয়ে আসা হচ্ছে। নতুন ট্রেনগুলোতে বায়ো টয়লেটের ব্যবস্থা করা হচ্ছে যাতে পরিবেশ নষ্ট না হয়। আগামী বছর খুলনা থেকে মংলা নতুন রেললাইন চালু হবে। এছাড়া টঙ্গী-জয়দেবপুর পর্যন্ত ডাবল লাইন চালু হবে।	0	0.978

Table 13: Representative false positives (human-authored text predicted as AI-generated). The selected examples exhibit repetitive, formal, or highly structured discourse patterns, illustrating challenging cases where human writing may stylistically resemble machine-generated text.

Language Family	Language	True Label	Generator	Snippet	Prediction
Indo-Aryan	Marathi	1	Qwen2.5	काही राष्ट्रवादी पुढाऱ्यांनी (उदा. बोरीस जॉन्सन) युके युरोपीयन युनियनमधे रहाणे तोट्याचे आहे, आपण युरोपीयन युनियनच्या बोंडल्याने दूध पितोय, आपण परावलंबी आहोत अशी हाकाटी केली. भारतातल्या हिंदुत्व परिवाराला लाजवेल अशा रीतीने आपण निषेधक आहोत. इतर देशातील समस्यांची भाग्यवाणी करून आमंत्रित झाली, परंतु राष्ट्रवादीही नेते खुप सरदार विश्वासाचा मानले. (कृपया या लाञ्छनांमध्ये अंग्रेजी कडक देण्यात आलेल्या वाक्यांचे मराठीत निरुद्ध करणे हवे)	0.018
Dravidian	Telugu	1	GPT-OSS	సాదారణంగా హీరోయిన్స్ ఆహారం విషయంలో చాలా జాగ్రత్తగా ఉంటారు. ఎంతగా తినాలని మనసులో ఉన్నా కూడా కట్టి పెట్టి మరీ డైట్ ను ఫాలో అవుతారు. తమ ఫిజిక్ విషయంలో జాగ్ర త్తలు తీసుకోవడం కోసం నోరు కుట్టేసుకుని ఉంటారు అనడంలో సందేహం. ఈ నెలలో హైదరాబాద్‌లో జరిగిన 'ఫిట్‌నెస్ ఫ్యాష న్ షో'లో అనేక హీరోయిన్లు తమ డైట్ ప్లాన్‌ను పంచుకున్నారు. ప్రసిద్ధ నటీమణి సుమన్యా గారూ, రోజుకు మూడు భోజనాలకే పరిమితం చేసి, సాయంత్రం యోగా సెషన్‌లో పాల్గొంటున్నారని తెలిపారు. ఇతరులు కూడా హై ప్రోటీన్ షేక్, సలాడ్లు, మరియు సూపర్‌ఫుడ్‌లను రోజువారీ ఆహారంలో చేర్చుకుంటున్నారు. ఈ విధానంతో వారు ఆరోగ్యంగా ఉండటమే కాదు, ఫిల్మ్ షూటింగ్‌లో కూడా శక్తివంతంగా కనిపిస్తున్నారు. సినిమా నిర్మాతలు కూడా ఆరోగ్యకరమైన ఆహారంపై ప్రత్యేక శ్రద్ధ చూపిస్తున్నారు	0.018
Indo-Aryan	Bengali	1	Qwen2.5	ট্রেন নিয়ে আমার একটা অভিজ্ঞতা আছে। সেইটাই বলি। আমি থাকি জুড়াইন। আর পড়াতে যাই উত্তরায়। মানে ঢাকার এক প্রান্ত থেকে আরেক প্রান্তে আর কী। এইটা ১৯৯৮ সালের কথা বলছি। বন্যার শহরে গিয়ে পৌঁছাতে হয়েছিল, অত্যন্ত ট্রেনে আটকে পড়া। কিছুদিন ধরে সেখানেই থাকতে □বাই হয়েছিল, যাত্রার অসুবিধার মধ্যে পড়ে আমি। বন্যা শহরটা দেখতে ভালো লে□ছিল, কিন্তু সেই অভিজ্ঞতাটা প্রতিষ্ঠানের মধ্যেও থাকবে।	0.019
Indo-Aryan	Gujarati	1	Aya-8B	ભૂતપૂર્વ ભારતીય ક્રિકેટ કેપ્ટન અને હાલમાં BCCI પ્રમુખ સૌરવ ગાંગુલી (Sourav Ganguly) અને ક્રિકેટર વિરેન્દ્ર સહેવાગની (Virendra Sehwag) મજબૂત જોડી, ક્રિકેટ મેદાન બાદ હવે 'કૌન બનેગા કરોડપતિ' સિઝન 13 (KBC-13) અને છૂટના મુખ્યકારીની હાલમાં સિઝન 14 (KBC-14) જોડી, તેવી પછી ક્રિકેટ મેદાન બાદ હવે 'કૌન બનેગા કરોડપતિ' સિઝન 15 (KBC-15).	0.022

Table 14: Representative false negatives (AI-generated text predicted as human). The selected examples are highly fluent and structurally coherent, illustrating challenging failure cases for the detector.