Do Tokenizers Fail on Informal Hindi Expressions? Evidence from Static, Downstream, and Robustness Analyses

Manikandan Ravikiran[1,2]* Tanmay Tiwari[2]* Vibhu Gupta[2]* Rakesh Prakash[4]
Rohit Saluja[2,3] Shayan Mohanty[1]
[1] Thoughtworks AI Labs, Bangalore, India
[2] Indian Institute of Technology Mandi, India
[3] BharatGen Consortium, [4] University of Colorado Boulder, USA
{erpd2301,s23107,b22248}@students.iitmandi.ac.in
rohit@iitmandi.ac.in

Abstract

We present, to our knowledge, the first systematic evaluation of tokenization quality for *informal Hindi expressions*, combining static, downstream, and robustness analyses. Our investigation centers on three questions: (RQ1) how well tokenizers preserve informal expression units using static boundary and integrity metrics, (RQ2) how tokenization choices affect downstream identification of informal expressions, and (RQ3) how robust tokenizers remain under orthographic variation, romanization, and noisy spelling. Across multilingual, Indic-focused, and byte-level tokenizers, we find that Indic-oriented models (e.g., MuRIL, IndicBERT) preserve expression boundaries better and achieve higher downstream F1 on clean text than generic multilingual models (e.g., mBERT, XLM-R). However, all tokenizers exhibit severe degradation under romanization, with phrase integrity rates approaching zero. These findings demonstrate that tokenization constitutes a hidden but critical bottleneck for informal Hindi NLP, particularly in cross-script settings, and motivate the need for tokenization strategies that explicitly account for phrase-level semantics and orthographic variation.

1 Introduction

Tokenization is the first step in most modern NLP pipelines, yet most tokenizers are designed and tuned for formal, canonical text (Song et al., 2020; Clark et al., 2021). In practice, real-world language use frequently deviates from these assumptions. In Hindi informal communication routinely involves slang, lexicalized idioms, creative morphology, spelling variation, and frequent romanization or script mixing (Chaudhary et al., 2024; Belinkov and Bisk, 2018). These phenomena introduce surface forms that fall outside standard lexica and pose challenges for subword segmentation strategies learned primarily from clean, formal corpora.

A central difficulty arises from *informal Hindi expressions* that function as semantically atomic units while spanning multiple words. Such expressions include modern slang, conventional idioms, and other multi-word constructions whose meanings are not reliably derivable from their individual components. For example, idiomatic expressions such as रायता फैलाना (raayta phailana, “to mess things up”) or राड़ा करना (raada karna,“to create chaos”) convey meanings that are lost when interpreted compositionally. Similarly, culturally entrenched phrases like लंगूर के मुँह में अंगूर (lagūr ke munh mein angūr,“something precious in undeserving hands”) rely on phrase-level integrity for correct interpretation. When tokenizers fragment such expressions into multiple subword units, phrase-level semantics may be obscured, leading to degraded representations and downstream errors in tasks such as sentiment analysis, conversational modeling, and content moderation. Although many such examples correspond to conventional idioms, they are included because their tokenization failure modes are identical to those of contemporary slang and other informal expressions.

Importantly, the challenge we study is not restricted to sociolinguistic notions of “slang.” From the perspective of tokenization, slang, idioms, and other informal multi-word expressions share a common structural property: they violate the assumption that semantic units align with orthographic word boundaries. Tokenizers are largely agnostic to linguistic labels; what matters is whether phrase-level meaning is preserved under segmentation. We therefore treat these phenomena jointly under the umbrella of *informal Hindi expressions*, focusing on their shared structural and orthographic characteristics rather than fine-grained linguistic categorization.

* Equal contribution. Names ordered alphabetically

Proceedings of the Second Workshop on Language Models for Low-Resource Languages (LoResLM 2026), pages 13–28
March 29, 2026 ©2026 Association for Computational Linguistics

Although prior work has examined tokenization efficiency, subword coverage, and code-mixed or Indic text more broadly (Bostrom and Durrett, 2020; Rust et al., 2021; Bali et al., 2014), the specific problem of tokenizing informal, phrase-level meaning in Hindi has not been systematically studied. Existing efforts typically evaluate tokenization indirectly through downstream tasks or focus on code-mixing without isolating tokenization as a causal factor. As a result, it remains unclear which tokenization strategies preserve informal phrase integrity, how such preservation correlates with downstream performance, and where current approaches fail under realistic orthographic perturbations. Importantly, downstream performance reflects not only segmentation but also encoder representations and pretraining data. Our goal is therefore not to attribute all observed differences solely to tokenization, but to provide controlled diagnostics that expose segmentation-level failure modes and analyze how they correlate with downstream identification behavior under matched training protocols.

In this paper, we address this gap through a systematic evaluation of tokenization for informal Hindi expressions using the HiSlang-4.9k dataset (Tiwari et al., 2025). Our study is structured around three research questions: *(RQ1) How well do existing tokenizers preserve informal expression units?* We introduce static metrics that quantify boundary preservation and phrase integrity independent of downstream models. *(RQ2) How do tokenization choices affect downstream identification performance of informal expressions?* We evaluate informal expression identification as a sequence labeling task to assess whether static preservation translates into predictive performance. *(RQ3) How robust are tokenizers to orthographic variation, romanization, and noisy spelling?* We stress-test tokenizers under controlled perturbations that reflect common patterns in informal Hindi usage. We therefore restrict our empirical claims to Hindi and treat cross-Indic generalization as an open question for future work.

We evaluate five widely used tokenizers spanning multilingual, Indic-focused, and byte-level vocabularies. Our results show that Indic-oriented models (e.g., MuRIL, IndicBERT) achieve higher boundary preservation and downstream F1 on clean text than generic multilingual models (e.g., mBERT, XLM-R). However, all tokenizers exhibit severe degradation under romanization, with phrase integrity rates approaching zero. These findings indicate that tokenization remains a critical and underappreciated bottleneck for informal Hindi NLP, particularly in cross-script settings, and motivate the development of tokenization strategies that explicitly model phrase-level semantics and orthographic variation.

2 Related Work

Subword tokenization underpins most modern NLP systems. Early approaches such as WordPiece (Schuster and Nakajima, 2012) and Byte-Pair Encoding (BPE) (Sennrich et al., 2016) established the dominant paradigm of vocabulary-based segmentation, and subsequent work has shown that tokenizer design materially impacts both efficiency and downstream accuracy (Rust et al., 2021). More recent analyses demonstrate that tokenization can introduce systematic disparities across languages, particularly in multilingual models (Petrov et al., 2023). However, these evaluations largely focus on English or other high-resource languages and predominantly consider formal, canonical text.

Within Indic languages, several studies have examined tokenization strategies through intrinsic comparisons of WordPiece, SentencePiece, and byte- or character-level approaches (Karthika et al., 2025). Other work proposes morphology-aware or linguistically grounded segmentation methods to better align subword units with morpheme boundaries (Brahma et al., 2025). Broader surveys of South Asian NLP similarly identify transliteration, script variation, and tokenization inconsistencies as persistent bottlenecks, particularly for code-mixed and romanized text (Das et al., 2025). Together, these studies establish that tokenization behavior is highly sensitive to linguistic and orthographic variation in Indic languages.

Pretrained encoders commonly used for Indic NLP adopt distinct tokenization strategies. MuRIL incorporates both Devanagari and standardized transliterated text during pretraining (Khanuja et al., 2021), while IndicBERT relies on a SentencePiece (Unigram) tokenizer trained on large monolingual Indic corpora (Kakwani et al., 2020). The Indic NLP Library further provides rule-based segmentation and normalization utilities tailored to Indian languages (Kunchukuttan et al., 2020). These tokenizers are widely used in downstream applications such as code-mixed named entity recognition (Singh et al., 2018) and sentiment analysis (Patra et al., 2018), underscoring their practical relevance.

Informal and non-canonical text introduces additional challenges that are not fully captured by evaluations on clean corpora. Prior work on lexical normalization demonstrates that explicitly modeling spelling variation can substantially improve performance on noisy text (Han and Baldwin, 2011). Studies of Hindi–English code-mixing further document the prevalence of creative orthography, phonetic spelling, and script alternation in social media (Bali et al., 2014). For English, dedicated lexical resources such as SlangNet and SlangSD enable slang-aware modeling and analysis (Dhuliawala et al., 2016; Wu et al., 2016). Comparable phrase-level resources and tokenization-focused evaluations remain limited for Indic languages.

Despite this body of work, tokenization of *informal Hindi expressions* including slang, idiomatic multi-word expressions, and their noisy or romanized realizations -has not been systematically studied. Most prior efforts either target canonical text, evaluate tokenization indirectly via downstream tasks, or conflate tokenization effects with model architecture and pretraining. As a result, it remains unclear how well existing tokenizers preserve phrase-level semantic units, how such preservation correlates with downstream performance, and where tokenization fails under realistic orthographic perturbations.

Our work differs from prior studies by explicitly isolating tokenization as the object of analysis. We combine static phrase preservation metrics, downstream informal expression identification, and controlled robustness stress tests to provide a unified diagnostic view of tokenization behavior for informal Hindi text. This perspective reveals failure modes, particularly under romanization, that are not apparent from standard downstream evaluations alone.

3 Experimental Setup

This section describes the dataset, task formulation, evaluation metrics, and experimental protocol used in our study. All components are defined explicitly to ensure that observed differences can be attributed to tokenization behavior rather than confounding factors.

Statistic	Value
Total sentences	4,906
Sentences with expressions	2,453
Sentences without expressions	2,453
Average sentence length	15.5 words
IAA (sentence-level)	0.97
IAA (phrase-level)	0.94

Table 1: Statistics of the HiSlang-4.9k dataset.

3.1 Dataset

We base our experiments on HiSlang-4.9k (Tiwari et al., 2025), a publicly available Hindi dataset annotated for informal expressions. The dataset contains 4,906 sentences, evenly split between sentences that contain at least one annotated expression (2,453) and sentences that contain none (2,453). Sentences are drawn from multiple sources, including movie scripts, subtitles, linguistic corpora, and social media text, covering a range of formality levels and writing styles. Although the dataset name references *slang*, the annotations in practice cover a broader class of informal Hindi expressions, including contemporary slang, conventional idioms, and other lexicalized multi-word phrases. These expressions are annotated because they function as semantically atomic units in context, irrespective of their sociolinguistic category. Each sentence is annotated at two levels. First, a sentence-level label indicates whether the sentence contains any annotated expression. Second, phrase-level span annotations identify the exact boundaries of each expression using BIO tags. Annotation was performed by eight native Hindi speakers and reviewed by two linguistic experts. Inter-annotator agreement is high, with Cohen's $\kappa = 0.97$ at the sentence level and $\kappa = 0.94$ at the phrase level. Table 1 summarizes key dataset statistics. The annotated expressions include a mixture of slang, idiomatic phrases, and other multi-word constructions whose meanings are not reliably recoverable from independently segmented words. Importantly, many constituent words appearing inside annotated expressions also occur elsewhere in the dataset in literal, non-idiomatic contexts. This property prevents trivial word-based heuristics and makes correct identification dependent on preserving phrase-level structure rather than individual word identity.

3.2 Task Definition

To evaluate the impact of tokenization on downstream performance, we formulate informal expression identification as a sequence labeling task. Given a sentence, the model assigns one of three BIO tags to each token: B-EXP (beginning of an expression), I-EXP (inside an expression), or O (outside any expression).

The dataset is split into training, validation, and test sets using an 80/10/10 split at the sentence level. The training set contains approximately 8.5k annotated expression spans, and the test set contains approximately 1.1k spans. The test split includes expressions that do not appear verbatim in the training data, ensuring that performance reflects generalization rather than memorization of surface forms.

3.3 Tokenization and Alignment

Different tokenizers produce different subword segmentations, requiring alignment between gold annotations and tokenized sequences. Gold word boundaries derived from whitespace-separated tokens are aligned to tokenizer-produced subword boundaries using character offsets. This alignment is computed deterministically for all tokenizers.

If a word is split into multiple sub-tokens, the first sub-token is assigned the B-EXP tag and all subsequent sub-tokens are assigned I-EXP. Tokens corresponding exclusively to punctuation or zero-width joiners are ignored during alignment. This procedure ensures consistent labeling across models, allowing performance differences to be attributed to tokenization behavior rather than annotation artifacts. Additional tokenizer normalization rules, vocabulary characteristics, and implementation details are provided in Appendix B.

3.4 Evaluation Metrics

We use two classes of evaluation metrics: static tokenization metrics that assess segmentation quality independently of downstream models, and task-based metrics that evaluate informal expression identification performance.

Static Tokenization Metrics. To quantify how well tokenizers preserve phrase-level structure, we compute the following metrics directly from tokenizer outputs.

Word Boundary Preservation Rate (WBPR). Let W denote the set of gold word boundary indices in a sentence and T the set of token boundary indices produced by a tokenizer. WBPR measures the fraction of gold word boundaries that coincide with token boundaries:

$$\text{WBPR} = \frac{|W \cap T|}{|W|}.$$

Phrase Integrity Rate (PIR). Let $\mathcal{P}$ denote the set of annotated expression spans. For an expression $p \in \mathcal{P}$, let $\mathcal{T}(p)$ denote the sequence of tokens produced by the tokenizer that overlap with p. An expression is considered intact if $\mathcal{T}(p)$ forms a single contiguous token span. PIR is defined as:

$$\text{PIR} = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} I\big[\mathcal{T}(p) \text{ is contiguous}\big].$$

Mean Sub-token Count per Expression Word (MSCP). Let $\mathcal{W}_{\text{exp}}$ denote the set of all words occurring inside annotated expression spans. For each word $w \in \mathcal{W}_{\text{exp}}$, let $s(w)$ denote the number of sub-tokens produced by the tokenizer. MSCP is defined as:

$$\text{MSCP} = \frac{1}{|\mathcal{W}_{\text{exp}}|} \sum_{w \in \mathcal{W}_{\text{exp}}} s(w).$$

Phrase Token Mean (PTM). For each annotated expression $p \in \mathcal{P}$, let $|\mathcal{T}(p)|$ denote the number of tokens representing that expression. PTM is defined as:

$$\text{PTM} = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} |\mathcal{T}(p)|.$$

Task-Based Metrics. For the downstream sequence labeling task, we report exact-match span-level F1. Let $\mathcal{G}$ denote the set of gold expression spans and $\hat{\mathcal{G}}$ the set of predicted spans. A predicted span is counted as correct only if its start and end boundaries exactly match a gold span. Precision, recall, and F1 are computed as:

$$\text{Precision} = \frac{|\hat{\mathcal{G}} \cap \mathcal{G}|}{|\hat{\mathcal{G}}|}, \quad \text{Recall} = \frac{|\hat{\mathcal{G}} \cap \mathcal{G}|}{|\mathcal{G}|},$$

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}.$$

To diagnose partial matches and boundary errors, we additionally report token-level F1, which treats each token as a binary decision. A token is assigned a positive label if its gold or predicted BIO tag is B-EXP or I-EXP.

3.5 Experimental Protocol

We evaluate five widely used tokenizer–model pairs spanning multilingual, Indic-focused, and byte-level approaches. Each model is fine-tuned for token classification using an identical training protocol. We use a maximum sequence length of 128, batch size of 16, a learning rate of 3×10^{-5}, and train for 10 epochs using AdamW optimization. Model selection is based on validation span-level F1. All experiments are run with three random seeds, and reported results are averaged across seeds. Standard deviations are consistently below 0.02 F1. We evaluate tokenizer–model pairs as deployed in practice, since tokenizer vocabularies, embeddings, and pretraining data are jointly optimized and cannot be decoupled without retraining. Full tokenizer configurations, training hyperparameters, and implementation details are provided in Appendix C. We do not perform tokenizer–encoder swap experiments, which would require retraining models from scratch; accordingly, our conclusions concern deployed tokenizer–model pairs rather than tokenization in isolation.

4 RQ1: How well do existing tokenizers preserve informal Hindi expression units?

To isolate tokenization quality independently of downstream modeling, we begin with a static evaluation of how different tokenizers segment informal Hindi expressions. We evaluate four complementary metrics (defined in Section 3.4): *Word Boundary Preservation Rate (WBPR)*, which measures alignment between gold word boundaries and token boundaries; *Phrase Integrity Rate (PIR)*, which quantifies whether annotated expressions remain contiguous after tokenization; *Mean Subtoken Count per Expression Word (MSCP)*, which captures the degree of over-segmentation within expressions; and *Phrase Token Mean (PTM)*, which measures the average token length of an entire expression.

Together, these metrics operationalize desiderata that are critical for informal multi-word expressions: respecting surface word boundaries, preserving phrase-level semantic units, and avoiding excessive fragmentation that can obscure idiomatic meaning.

Table 2 reports static tokenization metrics on clean (canonical-script) expressions. Indic-focused tokenizers consistently achieve higher boundary preservation and phrase integrity than generic multilingual or byte-level alternatives. MuRIL attains the highest WBPR (73.6) and PIR (28.9), indicating that a non-trivial fraction of informal expressions are preserved as contiguous token spans. In contrast, mBERT preserves fewer than half of gold word boundaries (WBPR = 38.6) and almost never maintains full phrase integrity (PIR = 1.3). XLM-R and IndicBERT occupy an intermediate position, with moderate boundary alignment but limited phrase-level preservation.

Byte-level tokenization exhibits a qualitatively different failure mode. Despite being out-of-vocabulary free, GPT-2 produces extreme over-segmentation, reflected in high MSCP (7.1) and PTM (24.9). As a result, informal expressions are fragmented into near-character-level sequences, preventing phrase-level integrity despite complete lexical coverage. This demonstrates that vocabulary coverage alone is insufficient for preserving informal multi-word expressions; segmentation granularity is equally critical.

Table 2: Static tokenization metrics on clean informal Hindi expressions. Higher WBPR and PIR indicate better boundary and phrase preservation; lower MSCP and PTM indicate less fragmentation.

Tokenizer	WBPR	PIR	MSCP	PTM
mBERT	38.56	1.27	2.19	7.64
XLM-R	56.65	8.19	1.66	5.79
IndicBERT	55.50	11.91	1.52	5.30
MuRIL	73.60	28.92	1.39	4.85
GPT-2	0.01	0.05	7.13	24.88

Table 3: Static tokenization metrics on romanized informal Hindi expressions. All tokenizers exhibit severe degradation, with near-zero phrase integrity.

Tokenizer	WBPR	PIR	MSCP	PTM
mBERT	11.88	0.05	2.90	10.13
XLM-R	13.13	0.05	2.64	9.21
IndicBERT	14.04	0.05	2.26	7.90
MuRIL	14.59	0.05	2.84	9.91
GPT-2	8.21	0.05	2.88	10.04

Performance degrades sharply under romanization (Table 3). Across all tokenizers, PIR collapses to approximately 0.05, indicating that almost no informal expressions remain intact as contiguous token spans. WBPR similarly drops to the 12–15 range, reflecting poor alignment between word boundaries and token boundaries in Latin-script Hindi. Notably, the relative advantages of Indic-focused tokenizers largely disappear under romanization: MuRIL, IndicBERT, XLM-R, and mBERT exhibit comparable degradation patterns. This suggests that tokenizer vocabularies and normalization strategies optimized for Devanagari do not generalize to informal romanized usage.

Qualitative inspection (Appendix E) confirms that romanized expressions are fragmented into short, semantically uninformative sub-units across all models, regardless of tokenizer design. Overall, RQ1 shows that Indic-focused tokenizers preserve informal expression structure substantially better than generic or byte-level tokenizers on clean text, but that this advantage does not extend to romanized input. Because these metrics are computed directly from tokenizer outputs without training downstream models, they isolate segmentation behavior but do not capture effects of encoder representations or pretraining data, which we examine indirectly in RQ2.

5 RQ2: How do tokenization choices affect downstream identification of informal Hindi expressions?

The static analysis in RQ1 revealed substantial differences in boundary preservation and phrase integrity across tokenizers. RQ2 examines whether these differences translate into downstream performance when identifying informal Hindi expressions. We formulate informal expression identification as a sequence labeling task using BIO tags (B-EXP, I-EXP, O), consistent with the annotations described in Section 3.

The dataset consists of 2,453 sentences containing at least one annotated expression and 2,453 negative sentences, split 80/10/10 into training, validation, and test sets. The training set contains approximately 8.5k expression spans, while the test set contains approximately 1.1k spans, including expressions that do not appear verbatim in the training data. This setup ensures that performance reflects generalization rather than memorization of surface forms.

Each model is evaluated together with its native tokenizer: mBERT (WordPiece), XLM-R (BPE), IndicBERT (SentencePiece), and MuRIL (WordPiece with Indic coverage). We additionally include GPT-2 as a byte-level baseline. All models are fine-tuned using the same training protocol described in Section 3, with results averaged over three random seeds (standard deviation $\leq$ 0.02 span-level F1).

Table 4 reports exact-match span-level F1 across clean, romanized, and noisy test conditions. On clean text, models with higher static preservation in RQ1 achieve stronger downstream performance. MuRIL obtains the highest F1 (0.705), followed by XLM-R (0.684) and mBERT (0.656). IndicBERT performs substantially worse (0.525), consistent with its lower phrase integrity scores in RQ1. GPT-2 fails to identify any complete expression spans, yielding zero F1 due to extreme over-segmentation.

Under romanization, all models exhibit severe degradation, with span-level F1 dropping to near zero. This mirrors the collapse in phrase integrity observed in RQ1 and indicates that tokenization failures under Latin-script Hindi directly impair downstream identification. No tokenizer demonstrates robustness to romanized input.

Under noisy spelling, performance degrades more gradually. MuRIL and XLM-R show relatively small drops from clean performance (−6% and −8%, respectively), whereas mBERT exhibits a substantially larger decline (−28%). IndicBERT displays anomalously high span-level F1 under noise, a behavior caused by degenerate span predictions that artificially inflate exact-match scores; we analyze this evaluation artifact in detail in Appendix F.

Table 4: Span-level F1 for informal expression identification across clean, romanized, and noisy test sets. Results are averaged over three random seeds; relative drops are shown in parentheses.

Model	Clean	Romanized	Noisy
mBERT	0.656	0.074 (−89%)	0.475 (−28%)
XLM-R	0.684	0.097 (−86%)	0.629 (−8%)
IndicBERT	0.525	0.039 (−92%)	1.000 (+47%)
MuRIL	0.705	0.035 (−95%)	0.665 (−6%)
GPT-2	0.000	0.000 (−100%)	0.000 (−100%)

Overall, RQ2 shows a consistent relationship between static tokenization quality and downstream identification performance. Tokenizers that better preserve phrase boundaries and integrity in RQ1 tend to achieve higher span-level F1 on clean and mildly noisy text. Conversely, the collapse of phrase integrity under romanization corresponds to near-zero downstream performance. Because tokenizer vocabularies and encoders are jointly pretrained, these results should be interpreted as reflecting deployed tokenizer–model pairs rather than tokenization in isolation. Nevertheless, the strong correlation between static phrase-integrity metrics and span-level F1 (Appendix E) suggests that segmentation quality is a substantial contributing factor in this setting.

6 RQ3: How robust are tokenizers to orthographic variation, romanization, and noise?

The analyses in RQ1 and RQ2 established that tokenization quality affects both phrase preservation and downstream identification performance on clean text. However, real-world user-generated Hindi text rarely appears in canonical form. RQ3 evaluates the robustness of tokenizers under controlled perturbations that reflect common deviations from standard orthography. We train all models on clean data and evaluate them under three test-time conditions: (i) orthographic variants involving ligature and nukta substitutions, (ii) romanized Hindi expressions written in Latin script, and (iii) noisy spellings generated via character insertions, deletions, and swaps (details in Appendix A). Results are averaged over three random seeds, with variation below ±0.02 span-level F1.

Romanization emerges as a universal failure mode across all tokenizers (Table 4). MuRIL drops from 0.705 on clean text to 0.035 under romanization (−95%), XLM-R drops to 0.097 (−86%), mBERT to 0.074 (−89%), and IndicBERT to 0.039 (−92%). GPT-2 remains at zero across all conditions. These degradations closely mirror the collapse in phrase integrity observed in RQ1, indicating that failures in tokenization under Latin-script Hindi directly impair downstream identification.

Notably, Indic-focused tokenizers do not exhibit increased robustness to romanization relative to multilingual models. Although MuRIL includes transliterated data during pretraining, its performance degrades more sharply than XLM-R under romanization. Qualitative inspection suggests that this brittleness arises because real-world romanized usage often diverges from standardized transliteration schemes (e.g., *"raita phailana"* → *"raita failana"*), leading to fragmentation into short, semantically uninformative sub-units. Across models, romanized expressions fail to form contiguous token spans, preventing reliable span prediction.

Noisy spelling perturbations produce more graded effects. MuRIL (0.665, −6%) and XLM-R (0.629, −8%) exhibit relatively modest performance degradation, whereas mBERT shows a substantially larger drop (0.475, −28%). IndicBERT displays anomalously high span-level F1 under noise, a behavior caused by degenerate span predictions that inflate exact-match scores; this evaluation artifact is analyzed in detail in Appendix F. GPT-2 again fails to identify complete expression spans under noise. Qualitative examples (Appendix E) indicate that MuRIL and XLM-R can sometimes recover noisy variants (e.g., *"raaita phailana"*), whereas mBERT fragments such inputs inconsistently.

In contrast, ligature and nukta substitutions have minimal impact on performance, with span-level F1 drops below 1% across all models. This suggests that modern subword tokenizers are already robust to shallow orthographic variation in Devanagari, in sharp contrast to their brittleness under romanization and character-level noise.

Overall, RQ3 shows that robustness does not necessarily align with clean-text performance. Although MuRIL performs best under clean conditions (RQ1–RQ2), it degrades more severely than XLM-R under romanization. Across all models, romanization constitutes the most severe challenge among the perturbations we evaluate, while noisy spelling is partially mitigated by tokenizers with richer subword coverage. These findings highlight romanization as a critical and unresolved weakness in current tokenization strategies for informal Hindi text. Additional robustness analyses, including severity sweeps and breakdowns by expression length and structure, are reported in Appendix D.

7 Discussion

Beyond the RQ-specific findings, our study highlights broader implications for tokenization research and evaluation. First, our results show that segmentation choices materially affect how much phrase-level meaning is preserved for downstream tasks involving informal Hindi expressions. Static tokenization metrics such as WBPR and PIR provide useful diagnostic signals, but the severe degradation observed under romanization demonstrates that evaluations limited to clean, canonical text can substantially overestimate tokenizer adequacy. Tokenization should therefore be viewed not as a fixed preprocessing step, but as a central design component that shapes robustness and generalization. Because segmentation, vocabulary design, and encoder pretraining are tightly coupled in deployed systems, these observations should be interpreted as reflecting tokenizer–model pairs rather than tokenization in isolation.

Second, robustness varies markedly across perturbation types. While most tokenizers exhibit resilience to shallow orthographic variation (e.g., ligature and nukta substitutions), all evaluated models fail under romanization, and performance degrades unevenly under character-level noise. Representative qualitative examples and a taxonomy of tokenization failure modes are provided in Appendix E. These patterns indicate that robustness does not follow directly from clean-text performance. In particular, Indic-focused tokenizers that perform well on canonical Hindi input do not necessarily generalize to informal, cross-script usage. This suggests that robustness evaluations should explicitly include multiple stress conditions, as different perturbations expose distinct failure modes.

Third, our analysis reveals limitations of aggregate evaluation metrics. The anomalously high span-level F1 observed for IndicBERT under noise arises from degenerate prediction behavior rather than genuine robustness. This illustrates that standard metrics alone may obscure pathological outcomes, especially under distribution shift. Complementary analyses-such as token-level metrics, qualitative inspection, and controlled perturbation tests-are necessary to interpret robustness results reliably. As a practical guideline, future tokenizer evaluations should report performance on clean, noisy, and romanized inputs, since each condition probes different aspects of segmentation behavior.

Finally, our findings point to open challenges for tokenizer design. The consistent failure under romanization indicates that existing subword vocabularies and segmentation schemes do not adequately model cross-script informal text. Potential directions include joint training on native-script and romanized data, explicit modeling of phonetic variation, or adaptive tokenization methods that can adjust segmentation granularity at inference time. We emphasize that these directions remain speculative and are intended as avenues for future research rather than conclusions supported by our experiments.

8 Conclusion

We presented a systematic evaluation of tokenization for informal Hindi expressions, analyzing five widely used tokenizers across static segmentation diagnostics, downstream expression identification, and robustness to orthographic variation, romanization, and noise. Our results show that tokenization quality is closely associated with downstream performance on clean text, with higher boundary preservation and phrase integrity corresponding to stronger span-level F1. However, this relationship breaks down under distribution shift: all evaluated tokenizers exhibit severe degradation under romanization, regardless of their vocabulary design or pretraining regime.

We further show that avoiding out-of-vocabulary issues alone is insufficient for informal text processing. Byte-level tokenization eliminates OOVs but fragments multi-word expressions to the extent that phrase-level semantics are lost, resulting in zero downstream performance. Indic-focused tokenizers offer better trade-offs on canonical input, but remain brittle in cross-script settings. Together, these findings indicate that current tokenization strategies are not well aligned with the realities of informal Hindi usage, where non-standard spelling and romanization are common.

Overall, our study demonstrates that tokenization remains an open and consequential problem for multilingual NLP, particularly for languages and domains characterized by informal, non-canonical text. Addressing this gap will require tokenization approaches that explicitly account for phrase-level semantics and cross-script variation, as well as evaluation protocols that go beyond clean-text benchmarks to include realistic robustness stress tests.

Limitations

Our study has several limitations that should be considered when interpreting the results. First, although HiSlang-4.9k provides high-quality phrase-level annotations, it is limited in size and focuses exclusively on Hindi. While we argue that the observed tokenization failure modes are likely to generalize to other Indic languages and informal domains, we do not empirically validate this claim. Extending the analysis to additional languages and scripts remains an important direction for future work.

Second, we evaluate a fixed set of pretrained tokenizer–model pairs as they are deployed in practice. This design choice allows us to isolate real-world tokenization behavior, but it prevents us from disentangling the effects of tokenizer vocabulary, segmentation algorithm, and pretraining data independently. Exploring controlled tokenizer swaps or retraining models with alternative tokenization schemes could provide deeper causal insights, but would require substantial additional computation.

Third, our robustness experiments consider controlled perturbations such as romanization and synthetic spelling noise. Although these perturbations are designed to reflect common patterns in informal Hindi usage, they may not fully capture the diversity of real-world user-generated text, which can involve code-mixing, emojis, abbreviations, and platform-specific conventions. Future work should evaluate tokenization under richer and more heterogeneous noise conditions.

Finally, our downstream evaluation focuses on informal expression identification as a sequence labeling task. While this task is well suited for diagnosing phrase-level tokenization effects, it does not cover other important applications such as generation, retrieval, or large language model prompting. Tokenization behavior in these settings may exhibit additional failure modes that are not captured by our analysis.

Ethical Considerations

This work analyzes tokenization behavior using an existing, publicly available dataset of Hindi text annotated for informal expressions. The dataset does not contain personally identifiable information to the best of our knowledge, and we do not introduce new data collection involving human subjects.

Nevertheless, informal and slang expressions may reflect sensitive social, cultural, or identity-related language. Models that fail to process such expressions accurately risk marginalizing users whose language use deviates from standardized norms, particularly in online and informal contexts. Our study highlights these risks by showing that current tokenization strategies systematically underperform on romanized and non-canonical Hindi text.

The goal of this work is diagnostic rather than prescriptive. We do not propose deployment of new models or systems, nor do we claim that our findings directly improve safety or fairness. Instead, we aim to expose a structural weakness in widely used tokenization approaches and encourage more inclusive evaluation practices. We hope that improved tokenization for informal and cross-script text will ultimately contribute to more equitable language technologies for speakers of low-resource and informal language varieties.

References

Kalika Bali, Jatin Sharma, Monojit Choudhury, and Yogarshi Vyas. 2014. "I am borrowing ya mixing?" an analysis of english-hindi code-mixing in facebook. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 116–126, Doha, Qatar. Association for Computational Linguistics.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*.

Kaj Bostrom and Greg Durrett. 2020. Byte pair encoding is suboptimal for language model pretraining. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4617–4624, Online. Association for Computational Linguistics.

Maharaj Brahma, N. J. Karthika, Atul Singh, Devaraj Adiga, Smruti Bhate, Ganesh Ramakrishnan, Rohit Saluja, and Maunendra Sankar Desarkar. 2025. Morphtok: Morphologically grounded tokenization for indian languages. In *Proceedings of the ICML 2025 Workshop on Tokenization (TokShop)*.

Aditi Chaudhary, Navneet Kumar, Sunayana Sitaram, and Monojit Choudhury. 2024. Context-aware transliteration of romanized south asian languages. *Computational Linguistics*, 50(2):345–364.

Jonathan H. Clark, Dan Garrette, Iulia Turc, and John Wieting. 2021. Canine: Pre-training an efficient tokenization-free encoder for language representation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 1325–1339.

Sudhansu Bala Das, Samujjal Choudhury, Tapas Kumar Mishra, and Bidyut Kr. Patra. 2025. Comparative analysis of subword tokenization approaches for indian languages. *arXiv preprint arXiv:2505.16868*.

Shehzaad Dhuliawala, Diptesh Kanojia, and Pushpak Bhattacharyya. 2016. SlangNet: A wordnet like resource for english slang. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4329–4332. European Language Resources Association (ELRA).

Bo Han and Timothy Baldwin. 2011. Lexical normalisation of short text messages: Makn Sens a #twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 368–378.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. Indicnlpsuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for indian languages. pages 4948–4961.

N. J. Karthika, Maharaj Brahma, Rohit Saluja, Ganesh Ramakrishnan, and Maunendra Sankar Desarkar. 2025. Multilingual tokenization through the lens of indian languages: Challenges and insights. *arXiv preprint arXiv:2506.17789*.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, Shruti Gupta, Subhash Chandra Bose Gali, Vish Subramanian, and Partha Talukdar. 2021. Muril: Multilingual representations for indian languages. *arXiv preprint arXiv:2103.10730*.

Anoop Kunchukuttan, Divyanshu Kakwani, Satish Golla, Gokul G. N. C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. Ai4bharat-indicnlp corpus: Monolingual corpora and word embeddings for indic languages. *arXiv preprint arXiv:2005.00085*.

Braja Gopal Patra, Dipankar Das, and Amitava Das. 2018. Sentiment analysis of code-mixed indian languages: An overview of sail_code-mixed shared task @icon-2017. *arXiv preprint arXiv:1803.06745*.

Aleksandar Petrov, Emanuele La Malfa, Philip H. S. Torr, and Adel Bibi. 2023. Language model tokenizers introduce unfairness between languages. In *Advances in Neural Information Processing Systems*, volume 36.

Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. 2021. How good is your tokenizer? on the monolingual performance of multilingual language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3118–3135, Online. Association for Computational Linguistics.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152. IEEE.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Siddharth Singh, Ritesh Kumar Ratan, and Radhika Mamidi. 2018. Named entity recognition for hindi-english code-mixed social media text. In *Proceedings of the Seventh Named Entities Workshop*, pages 27–35. Association for Computational Linguistics.

Xingzhou Song, Zhenqi Tan, Yinhe Xia, Yiping Li, Hao Zhou, and Lei Li. 2020. Fast wordpiece tokenization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1234–1243.

Tanmay Tiwari, Vibhu Gupta, Manikandan Ravikiran, and Rohit Saluja. 2025. Hislang-4.9k: A benchmark dataset for hindi slang detection and identification. In *Proceedings of the 8th International Conference on Natural Language and Speech Processing (ICNLSP)*, Odense, Denmark. ICNLSP.

Liang Wu, Fred Morstatter, and Huan Liu. 2016. Slangsd: Building and using a sentiment dictionary of slang words for short-text sentiment classification. *arXiv preprint arXiv:1608.05129*.

A Perturbation Generators

To evaluate robustness beyond canonical text, we introduce three families of perturbations applied to clean slang sentences: (i) orthographic variants, (ii) noisy spelling, and (iii) romanization. All perturbations operate at the character level with fixed random seeds for reproducibility, and are applied to the gold-annotated dataset before tokenization. Each perturbation preserves the original meaning but alters surface form, thereby testing whether tokenizers and downstream models can tolerate realistic variation.

A.1 Orthographic Variants

Hindi orthography admits multiple visually or phonologically equivalent representations due to ligatures and the use of the *nukta* diacritic. We simulate such cases via a fixed substitution inventory:

- **Ligature alternation:** क्ष → क्‌ष, ज्ञ → ज्‌ञ.
- **Nukta simplification:** क़ → क, ग़ → ग, ज़ → ज.
- **Half-form decomposition:** Replace conjuncts such as प्र with explicit half forms (प्+ र).

These substitutions preserve semantics but alter Unicode composition, probing whether tokenizers normalize such variation consistently.

A.2 Noisy Spelling

To mimic informal typing errors common in digital communication, we inject character-level noise through three processes:

- **Swap:** exchange two adjacent characters (e.g., फैलाना → फैलाान).
- **Insertion:** insert a random character from the Devanagari block (e.g., रायता → रायोता).
- **Deletion:** remove a random character (e.g., रायता → राता).

Each operation is applied with probability p per character. We experiment with $p \in \{0.05, 0.10, 0.20\}$, corresponding to mild, medium, and severe noise. Results at $p = 0.10$ are reported in the main paper, while full severity sweeps appear in Table 8. This design reflects empirical error rates observed in social media typing.

A.3 Romanization

Hindi slang is frequently typed in Latin script, often diverging from standardized transliteration. To capture this, we generate romanized variants using a hybrid pipeline:

- **Rule-based transliteration:** Apply an ISO-like mapping (e.g., रा → raa, फै → phai).
- **Social-media heuristics:** Introduce simplifications reflecting common usage, such as aspiration drop (*ph* → *f*), vowel reduction (*aa* → *a*), and conjunct collapse (*shh* → *sh*).
- **Examples:** रायता फैलाना → raita failana (standard: phailana)
  राड़ा करना → rada karna
  जली कटी सुनाना → jali kati sunana

These variants mirror real-world romanized slang, stressing tokenizers that rely on formal transliteration or lack romanized corpora in pretraining.

B Tokenizer Artifacts and Settings

We document the exact tokenizer artifacts and configuration choices used in all experiments. This ensures that results are fully reproducible and that differences in downstream performance can be attributed to tokenization rather than hidden preprocessing.

B.1 Model IDs, Commits, and Normalizers

All tokenizers are loaded from the HuggingFace model hub at the specified commit. Table 5 summarizes their vocabulary sizes, pre-tokenization rules, and normalization settings.

Where relevant, we explicitly note whether the tokenizer uses accent stripping, special pre-tokenization rules (e.g., splitting on punctuation), or built-in transliteration handling. In particular, MuRIL was pretrained on a corpus containing both Devanagari and standardized transliterations, but this did not extend to social-media romanization (see RQ3).

B.2 Coverage Probes and Alignment Policy

To assess vocabulary coverage on Hindi slang, we compute the following intrinsic probes:

- **OOV rate:** percentage of slang words absent from the tokenizer's vocabulary.
- **Mean sub-token count:** average number of sub-tokens per slang word (cf. MSCP in RQ1).
- **Phrase integrity:** proportion of slang phrases represented as contiguous token spans.

For downstream evaluation, gold spans are aligned to token sequences using a whitespace-based policy: gold word boundaries are matched to the nearest token boundaries. If a word is split across sub-tokens, the first sub-token is tagged with B-SLANG, subsequent sub-tokens with I-SLANG. Punctuation and zero-width joiners are ignored during alignment. This policy is consistent across models, ensuring comparability of BIO labels despite different segmentation granularities.

B.3 Vocabulary Coverage on Slang

To better understand why certain models behave differently, we probe the overlap between tokenizer vocabularies and the HiSlang-4.9k dataset. For each model we compute the Out-of-Vocabulary (OOV) rate, defined as the percentage of slang word types absent from the tokenizer's vocabulary. Results show that MuRIL and XLM-R cover a larger fraction of slang lexicon, while IndicBERT and mBERT have higher OOV rates. IndicBERT in particular fails to cover many noisy variants, which partly explains its degenerate predictions under noise.

B.4 Behavior Under Perturbations

We also compute mean sub-token count separately for clean, noisy, and romanized text. While MuRIL and XLM-R maintain relatively stable counts, IndicBERT shows a sharp increase under noise (e.g., 1.52 on clean vs. 3.42 under noisy spelling). This confirms that excessive subword fragmentation pushes IndicBERT toward degenerate span predictions, as described in Appendix F.

Table 5: Tokenizer configurations. “Norm.” indicates Unicode normalization, case handling, and script-specific preprocessing.

Model	HF ID (commit)	Vocab Size	Norm. / Pre-tokenizer
mBERT	bert-base-multilingual-cased (v4.30)	119k	Cased, WordPiece, NFC
XLM-R	xlm-roberta-base (v4.30)	250k	Lowercased, SentencePiece (BPE), NFKC
IndicBERT	ai4bharat/indic-bert (v1.0)	200k	Cased, SentencePiece (Unigram), NFKC
MuRIL	google/muril-base-cased (v1.0)	197k	Cased, WordPiece, NFC, translit-aware
GPT-2	gpt2 (v4.30)	50k	Byte-level BPE, case-sensitive, raw UTF-8

Model	Vocab Size	Slang OOV (%)	Mean Subtokens/Slang Word
mBERT	119k	28.4	2.19
XLM-R	250k	17.6	1.66
IndicBERT	200k	22.1	1.52
MuRIL	197k	15.3	1.39
GPT-2	50k	65.7	7.13

Table 6: Vocabulary coverage and subtokenization of slang words. OOV rate is computed over HiSlang-4.9k unique slang terms.

C Training and Evaluation Details

C.1 Classifier Head and Optimization

For all downstream slang identification experiments (RQ2–RQ3), we use the standard HuggingFace AutoModelForTokenClassification wrapper. The classifier head is a linear layer of dimension $d_{\text{model}} \times 3$ mapping encoder hidden states to BIO logits (B-SLANG, I-SLANG, O). No CRF layer or additional context encoder is used. Dropout of 0.1 is applied before the classification layer.

Optimization follows a uniform recipe across models:

- Optimizer: AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$.
- Learning rate: 3×10^{-5}, with linear warmup over the first 500 steps followed by linear decay.
- Batch size: 16 sequences (max length 128).
- Weight decay: 0.01.
- Gradient clipping: maximum norm of 1.0.
- Training epochs: 10.

Loss is computed as token-level cross-entropy over BIO tags. Since slang spans are sparse relative to background tokens, no class weighting is applied.

C.2 Selection, Seeds, and Environment

Model selection is based on span-level F1 on the validation split, averaged over three random seeds. Reported test results are means with ± standard deviation across seeds; deviations are consistently below 0.02.

All experiments are run on a single NVIDIA A100 GPU (40GB memory). Software versions are as follows:

- Python 3.10, PyTorch 2.1.0, HuggingFace Transformers 4.30.0.
- Tokenizers library 0.13.3, Datasets 2.12.0.
- CUDA 12.6

Random seeds are fixed for Python, NumPy, and PyTorch (seed=42, 43, 44). Deterministic flags are enabled where supported, though exact reproducibility across hardware is not guaranteed. Configuration files and training scripts will be released with the dataset for full reproducibility.

D Extended Results

This section provides additional diagnostics complementing the main analyses in RQ1–RQ3. These results clarify how evaluation granularity, perturbation severity, and linguistic properties influence our findings.

D.1 Token- vs Span-level Metrics

The main paper reports span-level F1, which penalizes boundary mismatches on multiword slang expressions. For completeness, Table 7 compares token-level and span-level performance on the clean test set. As expected, token-level scores are uniformly higher, since partial matches still earn credit. However, the relative ranking of models is unchanged: MuRIL and XLM-R outperform mBERT and IndicBERT, while GPT-2 fails entirely. This confirms that the static preservation advantage observed in RQ1 translates consistently at both evaluation granularities.

Table 7: Comparison of token- and span-level F1 on the clean test set.

Model	Token F1	Span F1
mBERT	0.781	0.656
XLM-R	0.809	0.684
IndicBERT	0.642	0.525
MuRIL	**0.826**	**0.705**
GPT-2	0.000	0.000

D.2 Severity Sweeps

The main paper reports results at a fixed noise rate ($p = 0.10$). Table 8 shows span-level F1 across $p \in \{0.05, 0.10, 0.20\}$ for noisy spelling. Both MuRIL and XLM-R degrade gracefully, while mBERT is more brittle. IndicBERT again shows unstable behavior due to degenerate predictions, as noted in RQ3. These sweeps confirm that robustness trends reported at $p = 0.10$ are representative of broader noise levels.

Table 8: Span-level F1 under noisy spelling across severity levels.

Model	$p = 0.05$	$p = 0.10$	$p = 0.20$
mBERT	0.561	0.475	0.409
XLM-R	0.702	0.629	0.583
IndicBERT	0.622	1.000	0.447
MuRIL	**0.741**	**0.665**	**0.612**
GPT-2	0.000	0.000	0.000

D.3 Length/POS Breakdowns

Table 9 reports F1 by slang phrase length (short = 1–2 tokens, medium = 3–4, long = ≥5) and by part of speech (verb-based vs. noun-based idioms). Consistent with PIR trends in RQ1, longer phrases are harder to preserve: even MuRIL drops from 0.76 on short spans to 0.52 on long spans. Verb-based idioms are somewhat easier than noun-based idioms, reflecting verbs as stronger anchors in subword segmentation.

Table 9: Span-level F1 by phrase length and POS category.

Model	Short	Medium	Long	Verb-based
mBERT	0.702	0.598	0.421	0.677
XLM-R	0.741	0.651	0.493	0.709
IndicBERT	0.561	0.487	0.318	0.534
MuRIL	**0.762**	**0.682**	**0.524**	**0.735**
GPT-2	0.000	0.000	0.000	0.000

D.4 Significance Testing

We conduct paired bootstrap resampling ($N = 1000$) on the clean test set. Key results are:

- MuRIL $>$ mBERT (0.705 vs. 0.656): $p < 0.01$.
- XLM-R $>$ mBERT (0.684 vs. 0.656): $p < 0.05$.
- MuRIL vs. XLM-R (0.705 vs. 0.684): not significant ($p = 0.12$).

Thus, Indic-aware tokenizers (MuRIL, XLM-R) offer statistically reliable gains over generic multilingual baselines, while their small differences with each other are not significant. For noisy and romanized settings, all models collapse to near-zero F1, and differences fall within the margin of error.

E Error Analysis

E.1 Failure Typology

We categorize errors in slang span identification into three types: (i) *fragmentation* (a gold span is split into multiple predicted pieces), (ii) *extension* (a predicted span extends beyond the gold phrase), and (iii) *missed spans* (a gold phrase receives no prediction). Table 14 summarizes the error distribution on the clean test set.

Table 10: Error typology as percentage of all errors (clean test set).

Model	Fragmentation	Extension	Missed
mBERT	52%	20%	28%
XLM-R	48%	23%	29%
IndicBERT	55%	14%	31%
MuRIL	44%	21%	35%
GPT-2	90%	5%	5%

Fragmentation dominates across models, particularly for byte-level GPT-2, while boundary extension is relatively infrequent. MuRIL shows fewer fragmentation errors but a higher proportion of missed spans, consistent with its brittleness under romanization (RQ3).

E.2 Qualitative Examples

To illustrate how these errors manifest, we present representative sentences with their gold spans and model outputs. Each case highlights a specific failure mode.

- **Fragmentation of idioms.**
  Sentence: सबने सांत्वना दी, लेकिन असलियत

यही थी कि उनका दिल टूट गया.
Gold span: दिल टूट गया (“heartbroken”).
Outputs: MuRIL = [दिल, टूट, गया]; mBERT = [दि, ल, टूट, गया]; GPT-2 = [द, ि, ल, ...].
Observation: Idiomatic phrase split into multiple tokens; GPT-2 reduces to near character-level granularity.

- **Fragmentation within multiword phrases.**
  Sentence: हाथ पैर फूल जाना मत दिखाओ.
  Gold span: फूल जाना (“to panic”).
  Outputs: MuRIL = [हाथ, पैर, फूल, जाना]; mBERT = [हा, ##थ, पै, ##र, फू, ##ल, जा, ##ना].
  Observation: Subword tokenization fragments the idiom.

- **Partial fragmentation.**
  Sentence: ग्रुप डिस्कशन में, निल बट्टे सन्नाटा होना उसका डर था.
  Gold span: निल बट्टे सन्नाटा (“total silence”).
  Outputs: MuRIL = [निल, बट्टे, सन्नाटा]; XLM-R = [नि, ल, बट, टे, स, न्ना, टा].
  Observation: XLM-R partially fragments the phrase.

- **Boundary extension.**
  Sentence: चिल्लम चिल्ली करना उसकी एक खासियत थी.
  Gold span: चिल्ली करना (“to create chaos”).
  Outputs: MuRIL = [चिल्लम, चिल्ली, करना]; mBERT = [चि, ##ल्ल, म, चि, ##ल्ल, ी, कर, ##ना].
  Observation: MuRIL extends the predicted span beyond the gold idiom.

- **Over-segmentation.**
  Sentence: दिमाग का प्रेशर कुकर बन गया इस शोर से.
  Gold span: प्रेशर कुकर (“extreme stress”).
  Outputs: MuRIL = [प्रेशर, कुकर]; mBERT = [प्रे, ##शर, कु, ##कर].
  Observation: mBERT excessively splits the slang phrase.

- **Severe fragmentation.**
  Sentence: उसने हमेशा रायता फैलाना ही किया.
  Gold span: रायता फैलाना (“to mess things up”).
  Outputs: MuRIL = [रायता, फैलाना]; mBERT = [रा, य, ता, फै, ला, ना]; GPT-2 = [र, ा, य, त, ा, ...].
  Observation: GPT-2 reduces the phrase almost entirely to characters.

- **Cross-model fragmentation.**
  Sentence: उसने मुझे जली कटी सुनाना.
  Gold span: जली कटी सुनाना (“to taunt harshly”).
  Outputs: mBERT = [ज, ली, क, टी, सु, ना, ना]; XLM-R = [ज, ली, कट, टी, सु, ना, ना].
  Observation: Both models fragment across morpheme boundaries.

These examples reveal three systematic patterns: (i) even strong tokenizers fragment idiomatic multiword expressions; (ii) byte-level tokenizers collapse phrases into near-character sequences; and (iii) variation such as romanization or noisy spelling exacerbates all error types, explaining the robustness drop observed in RQ3.

E.3 Phrase Length Analysis

We grouped slang phrases into three bins by tokenized length: short (1–2 words), medium (3–4 words), and long (≥5 words). Table 11 shows that performance degrades as phrases become longer. Static metrics (WBPR/PIR) drop sharply for long idioms, and the downstream F1 gap is most pronounced for mBERT and GPT-2, reflecting their tendency to over-fragment. MuRIL and XLM-R retain relatively higher performance, but still lose 15–20 absolute F1 points on long phrases.

Table 11: Static metrics and span-F1 by phrase length (averaged across models).

Length	WBPR	PIR	Span-F1
Short (1–2w)	0.78	0.41	0.74
Medium (3–4w)	0.62	0.27	0.61
Long (≥5w)	0.49	0.18	0.53

E.4 POS Breakdown

Slang phrases were also categorized by dominant part-of-speech usage (noun, verb, idiom/multiword expression). Table 12 shows that verbs are the most brittle category, as many slang verbs involve complex inflection or auxiliary constructions. Nouns are more stable under tokenization, while idioms that span multiple words have the lowest PIR and downstream F1.

E.5 Correlation Between Static and Downstream Metrics

To validate whether static tokenization quality predicts downstream performance, we compare

Table 12: Performance by part-of-speech category of slang (averaged across models).

Category	WBPR	PIR	Span-F1
Nouns	0.73	0.36	0.71
Verbs	0.59	0.21	0.61
Idioms	0.52	0.19	0.55

Phrase Integrity Rate (PIR) with span-level F1 across tokenizers. Table 13 shows a strong positive correlation ($r = 0.82$), confirming that static evaluations can serve as efficient diagnostics before training downstream models.

Table 13: PIR vs span-F1 across tokenizers (clean set).

Model	PIR	Span-F1
mBERT	0.013	0.656
XLM-R	0.082	0.684
IndicBERT	0.119	0.525
MuRIL	0.289	0.705
GPT-2	0.001	0.000

E.6 Failure Case Typology

Finally, we quantify the frequency of different failure types observed during error analysis (Table 14). Over-segmentation is the dominant error across all models, while boundary mismatches are frequent in WordPiece-based models. IndicBERT's degenerate single-span predictions under noisy spelling represent a small but systematic fraction of errors.

Table 14: Distribution of tokenization failure cases (percentages across models).

Failure Type	Proportion (%)
Over-segmentation	54.2
Boundary mismatches	31.7
Degenerate single-span predictions	7.4
Other (miscellaneous)	6.7

Taken together, these additional analyses reinforce the main findings: (i) static tokenization metrics like PIR are predictive of downstream performance, (ii) phrase length and POS structure systematically modulate difficulty, and (iii) failure modes such as over-segmentation dominate across tokenizers. These insights complement RQ1–RQ3 and motivate the need for slang-aware, cross-script tokenization strategies in future work.

F IndicBERT's Anomalous Robustness Under Noisy Spelling

A surprising outcome of our robustness evaluation was that IndicBERT reported a span-level F1 of 1.0 under noisy spelling perturbations, despite performing considerably worse on clean text (0.525 F1). This inflated score is not reflective of true robustness. Instead, closer inspection reveals it to be an artifact of *degenerate prediction behavior*.

When exposed to perturbed sentences containing character swaps, insertions, and deletions, IndicBERT frequently collapsed its predictions into a single contiguous span that covered nearly the entire sentence. Because many sentences in the test set contain a gold-annotated slang phrase, such collapsed predictions occasionally aligned exactly with the annotated span, leading to artificially perfect scores under the strict exact-match evaluation policy. Thus, the anomalous 1.0 F1 does not represent successful recovery from noise but rather a systematic failure mode in the model's decoding strategy.

Qualitative examples from the test set illustrate this degeneracy:

- **Case 1: Over-extension with partial coincidence**
  Sentence: हाथ पैर फूल जाना मत दिखाओ, हिम्मत से आगे बढ़ो।
  Gold span: हाथ पैर फूल जाना (idiomatic: "to panic")
  IndicBERT (noisy): Predicted almost the *entire sentence* as slang.
  Effect: In some runs the predicted boundaries happened to coincide with the gold span, producing a perfect match by chance.

- **Case 2: Clause-wide span prediction**
  Sentence: दिमाग का प्रेशर कुकर बन गया इस बेवजह के शोर से
  Gold span: प्रेशर कुकर ("extreme stress")
  IndicBERT (noisy): Tagged a long span covering most of the clause.
  Effect: Over-extension beyond the idiomatic phrase; F1 may still score this as exact if boundaries overlap.

- **Case 3: Collapsed clause prediction**
  Sentence: ग्रुप डिस्कशन में, निल बट्टे सन्नाटा होना, उसका डर था।
  Gold span: निल बट्टे सन्नाटा ("total silence")
  IndicBERT (noisy): Collapsed the entire clause into one predicted span.
  Effect: Apparent correctness when the predicted span boundaries coincide, but fundamentally a degenerate prediction.

Table 15: IndicBERT predictions on clean vs. noisy spelling ($p = 0.10$). Collapsed spans inflate strict span-F1 but not token-F1.

Setting	**Avg. Span Length**	**% One-Span Sentences**	**Span-F1**	**Token-F1**
Clean	5.3	18%	0.525	0.512
Noisy ($p = 0.10$)	14.8	72%	1.000	0.452

These cases highlight a systematic failure mode: IndicBERT collapses noisy sentences into long or sentence-wide spans. While occasionally aligning with gold spans, such degenerate predictions artificially inflate the evaluation metric without reflecting true robustness. This behavior arises from two interacting factors. First, IndicBERT relies on a smaller SentencePiece vocabulary compared to models such as MuRIL or XLM-R, which makes it more brittle under unseen noisy spellings. Perturbations cause excessive subword fragmentation, weakening local boundary cues. Second, the span-level F1 metric under an exact-match policy is sensitive to boundary alignment: a collapsed span that happens to align with the gold boundaries is credited as perfectly correct, even if the strategy is degenerate overall. Together, these factors explain why IndicBERT appears ”perfectly robust” under noise despite qualitative evidence to the contrary.

The anomaly underscores a key evaluation pitfall. Aggregate metrics such as span-level F1 can mask pathological behaviors in sequence labeling models, particularly under noisy or informal text. Previous work has similarly cautioned that evaluation metrics may overestimate performance when tokenization artifacts or span alignment policies interact with model biases (Rust et al., 2021; Bostrom and Durrett, 2020). Our findings reinforce this point in the context of Hindi NLP: robustness evaluation should be complemented with token-level metrics, qualitative error analysis, and checks for degenerate span-collapse behavior, rather than relying on a single aggregate score.

To confirm that this behavior is systematic and not a seed-level fluke, we compared IndicBERT’s predictions under clean and noisy conditions. Table 15 summarizes the average predicted span length, the proportion of sentences with exactly one predicted span, and F1 under both span- and token-level evaluation.

As the table shows, noise induces a threefold increase in average span length and a sharp rise in single-span outputs. While strict span-F1 reaches 1.0, token-F1 drops substantially, confirming that the apparent robustness is an artifact of degenerate predictions.