𝓐𝓗-*Translit*: A Multi-Domain Dataset and Benchmark for Arabic-to-Hindi Transliteration

Anonymous ACL submission

Abstract

Arabic-to-Hindi transliteration is challenging because Arabic orthography often omits pronunciation-relevant information that Hindi orthography requires explicitly, yet the lack of large-scale resources has limited systematic evaluation of this task. We introduce ***𝓐𝓗-Translit***, the first large-scale multi-domain dataset for Arabic-to-Hindi transliteration with over 290K Arabic and 291K Hindi words. We also present *𝓐𝓗-Translit-Bench*, a 6K expert-verified benchmark spanning Quranic, contemporary, and entity-rich Arabic. We additionally propose a linguistic-informed analysis for fine-grained error diagnosis. We find that models substantially degrade under domain shift despite strong in-domain performance, with errors concentrated in vowel realization while consonants are preserved. Overall, our findings suggests that Arabic-to-Hindi transliteration is a reconstruction problem arising from pronunciation-critical information left implicit in Arabic orthography.

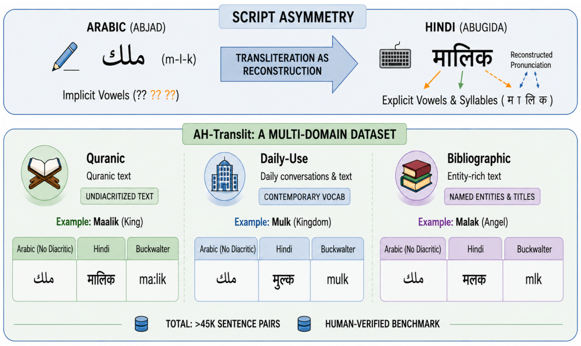


Figure 1: Overview of ***𝓐𝓗-Translit*** and *𝓐𝓗-Translit-Bench*. The benchmark spans Quranic, Daily-Use, and Bibliographic domains and supports both in-domain and cross-domain evaluation of Arabic–Hindi transliteration robustness.

1 Introduction

Transliteration enables cross-script access to information by converting text between writing systems while preserving pronunciation. Accurate transliteration supports multilingual communication, information accessibility, language learning, and cultural preservation. The need is particularly pronounced for Arabic and Hindi, two widely used languages that interact across migration, education, religion, and digital content consumption. Millions of Hindi speakers encounter Arabic text in public services and educational resources, while many non-Arabic-speaking South Asian Muslims rely on transliterated Arabic for Quranic recitation and religious observances [Harahap et al., 2023, Shariq, 2023].

Despite its practical importance, Arabic-to-Hindi transliteration remains challenging and relatively underexplored. Arabic is largely unvocalized, omitting most short vowels, whereas Devanagari requires explicit realization of vowels and syllabic structure [Habash, 2010, Boumaraf et al., 2022]. Consequently, multiple Hindi renderings may correspond to the same Arabic form, requiring models to infer information not explicitly encoded in the source script. The task is further complicated by phonetic mismatches and variation across domains, ranging from classical Quranic Arabic and Modern Standard Arabic (MSA) to named-entity-rich bibliographic text. Yet existing resources are limited in scale, confined to specific domains, or unsuitable for systematic evaluation, leaving cross-domain generalization and transliteration failure modes insufficiently understood.

To address this gap, we introduce ***𝓐𝓗-Translit***, the first large-scale public resource for Arabic-to-Hindi transliteration. ***𝓐𝓗-Translit*** contains over 45K parallel sentence pairs spanning Quranic Arabic, Modern Standard Arabic, and bibliographic text, exposing models to substantially different lexical, syllabic, and orthographic conditions. We further establish *𝓐𝓗-Translit-Bench*, a manually verified benchmark for standardized in-domain and

cross-domain evaluation, enabling study of transliteration robustness.

Using ***AH-Translit***, we evaluate representative transliteration architectures across domains and training regimes. The benchmark reveals substantial gaps between in-domain and cross-domain performance, even for strong models. Mixed-domain training improves robustness under distribution shift. Moreover, fine-grained linguistic analysis reveals structured error patterns concentrated in specific phonetic and orthographic phenomena. These findings underscore the importance of robustness-oriented evaluation and fine-grained error analysis beyond character-level accuracy.

Our contributions are summarized as follows:

- **A Dataset.** We introduce ***AH-Translit***, the first public Arabic-to-Hindi transliteration dataset containing over 45K parallel sentence pairs spanning Quranic, Modern Standard Arabic, and bibliographic domains.
- **A Benchmark.** We establish *AH-Translit-Bench*, a manually verified benchmark that supports reliable in-domain and cross-domain evaluation of Arabic-to-Hindi transliteration systems.
- **Baselines.** We benchmark representative transliteration architectures, and analyze robustness and generalization across domains.
- **Linguistically-Informed Error Analysis.** We develop an analysis framework for characterizing transliteration failures and use it to uncover structured error patterns across linguistic layers.

2 Related Work

Machine Transliteration and Benchmarks Machine transliteration has evolved from rule-based and statistical approaches to neural sequence-to-sequence and Transformer-based models, with benchmark datasets and shared tasks playing a central role in enabling systematic progress [Kaur and Garg, 2022, Li et al., 2009, Jayakumar et al., 2026]. Existing resources, however, primarily target related-script language pairs, multilingual Indic benchmarks [Jayakumar et al., 2026, Roark et al., 2020, Madhani et al., 2023], or Arabic-centric tasks such as Arabizi normalization, Arabic–Latin transliteration, and script conversion [Maqbool et al., 2015, Hajbi et al., 2025, Talafha et al., 2021, Hadj Ameur et al., 2017]. However, none directly target Arabic–Hindi transliteration, leaving this setting underrepresented in existing transliteration benchmarks and resources.

Arabic–Hindi Transliteration Challenges Arabic–Hindi transliteration is particularly challenging because Arabic is an *abjad* in which short vowels are often omitted, whereas Devanagari is an *abugida* that explicitly encodes vowel information [Daniels, 1990, Rogers, 2005, Maqbool et al., 2015]. Consequently, systems must infer pronunciation-relevant information that is absent from the source spelling, creating ambiguity in vowel realization and grapheme-to-phoneme alignment [Knight and Graehl, 1998, Hadj Ameur et al., 2017, Hajbi et al., 2025]. This asymmetry makes Arabic–Hindi substantially more challenging than transliteration between more phonologically explicit or closely related scripts.

Synthetic Data Construction Synthetic parallel data is increasingly used to address data scarcity in low-resource NLP and transliteration tasks [Nadăş et al., 2025, Jayakumar et al., 2026]. However, synthetically generated transliteration pairs can introduce inconsistent mappings and script-specific artifacts, particularly in settings involving substantial script and phonological divergence [Nadăş et al., 2025]. These limitations make human validation and iterative refinement important for constructing reliable resources in settings such as Arabic–Hindi transliteration.

Fine-Grained Evaluation Evaluating transliteration systems requires more than exact orthographic matching, since multiple surface forms may preserve the same pronunciation [Kaur and Garg, 2022, Durrani and Hussain, 2010]. Character Error Rate (CER) remains a useful diagnostic, but by itself it may not fully capture pronunciation-preserving variation [Hajbi et al., 2025]. This motivates evaluation that goes beyond surface error alone and examines finer-grained preservation of linguistic information.

3 The *AH-Translit* Dataset

3.1 Dataset Curation

Data Sources. We construct ***AH-Translit*** from three complementary sources spanning classical, contemporary, and entity-rich Arabic. We incorporate two expert-curated resources that serve as

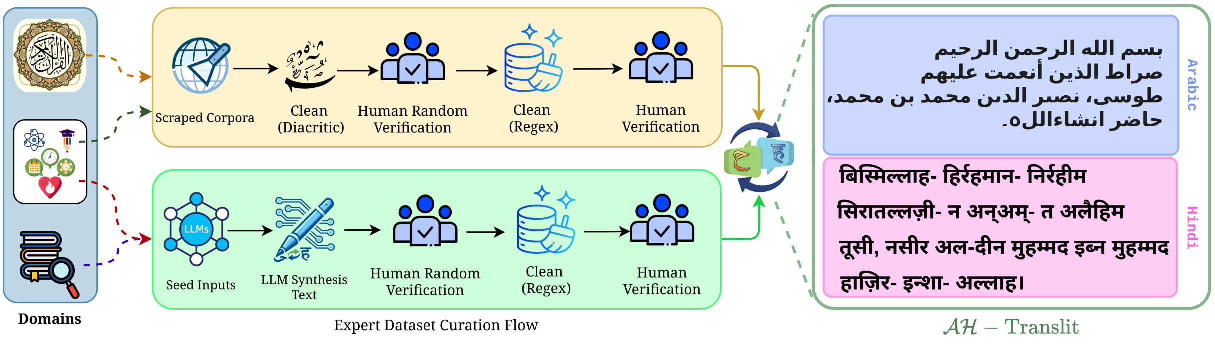


Figure 2: **The $\mathcal{AH}$-Translit** curation process uses two pipelines. The top pipeline processes Human-expert data from the Quranic and MSA domains via direct validation. The bottom pipeline processes the Bibliographic corpus synthetically, using LLM for parallel data generation and iterative validation.

Dataset	XL	Direct	Sent.	Multi	Pron.	#Pairs
NEWS Tasks	✓	✗	✗	✗	✗	∼15K
ANETAC	✓	✗	✗	✗	✗	79K
Dakshina	✗	✗	✓	✗	✗	10M+
Arabizi Corpora	✗	✗	✓	✗	✗	∼100K
Urdu–Hindi Corpora	✗	✓	✓	✗	✗	∼50K
Aksharantar	✓	✓	✗	✓	✗	26M
$\mathcal{AH}$-Translit (Ours)	✓	✓	✓	✓	✓	**45K**

Table 1: Comparison with representative transliteration datasets and benchmarks. XL = Cross-Lingual Transliteration, Direct = Direct Non-Latin Script Transliteration, Sent. = Sentence-Level Context, Multi = Multi-Domain Coverage, Pron. = Pronunciation Awareness. ***$\mathcal{AH}$-Translit*** is the only resource that combines direct Arabic–Hindi transliteration, sentence-level context, multi-domain coverage, and pronunciation-aware validation.

a high-quality foundation: **(1)** a *Quranic* domain [HindiQuran.in, 2025], consisting of fully vocalized Classical Arabic paired with human-curated Hindi transliterations and governed by *Tajwid*[1]; and **(2)** a *Contemporary MSA* domain combining Modern Standard Arabic (MSA) phrases from an expert-authored educational resource [A.M.K. SIDDIQUI, 2014] with additional scientific and educational content. To capture transliteration challenges associated with named entities, we further include a *Bibliographic* domain derived from the Arabic bibliographic corpus of [Eryani and Habash, 2021]. Together, these sources provide coverage of classical, contemporary, and entity-rich Arabic while capturing diverse transliteration phenomena.

Data Collection. We employ two complementary collection strategies. For the Quranic and expert-curated MSA domains, we develop domain-specific scrapers to extract fully vocalized Arabic text and corresponding human-transliterated Hindi texts from verified scholarly repositories. These examples preserve pronunciation cues often absent from unvocalized Arabic. For the Bibliographic domain and auxiliary MSA subsets lacking Hindi transliterations, we use an LLM-assisted generation pipeline using Gemini 2.5 Pro[2]. We use a few-shot prompting strategy for generation, with prompt templates provided in Appendix A.2. The resulting corpus combines expert-curated and automatically generated transliterations across diverse domains.

Quality Assurance. To improve phonetic fidelity and orthographic consistency, we apply a three-stage quality-assurance pipeline. **(1)** *Audit*. We conduct a manual audit of 2,000 randomly sampled transliteration pairs from each domain to identify recurring artifacts introduced during data generation or scraping. Common issues include unsupported symbols, punctuation errors, malformed consonant clusters, and other formatting irregularities (eg. whitespace). **(2)** *Normalization*. Based on the observed error patterns, we develop regex-based normalization rules that remove systematic artifacts and standardize corpus formatting prior to manual review. **(3)** *Expert Validation*. Three bilingual annotators with expertise in Arabic phonology, Hindi orthography, and Devanagari transliteration independently review 1,000 randomly sampled pairs from each domain using a shared validation protocol. Evaluation focuses on: (i) phonetic correspondence between Arabic and Hindi forms, particularly for emphatic and guttural consonants; (ii) realization of short/long vowels through Devana-

[1]Tajwid refers to the rules governing Quranic pronunciation during recitation.

[2]Gemini 2.5 Pro was selected for its strong handling of Perso-Arabic and Devanagari scripts [Comanici et al., 2025].

Table 2: Corpus overview of *𝒜ℋ-Translit*. Statistics represent counts and mean sentence length per domain. All numbers are in thousands

Domain	Pairs	Words		Vocab		Chars/S	
		Ar	Hi	Ar	Hi	Ar	Hi
Quranic	6.3	76.8	73.8	20.7	32.7	64.2	90.3
Daily-Use	16.8	67.5	67.7	22.4	23.0	20.9	27.4
Bibliographic	21.6	145.7	150.2	31.4	33.0	39.1	50.7
Total	**44.7**	**290.0**	**291.7**	**74.4**	**88.7**	**35.8**	**47.5**

Table 3: Linguistic profile of *𝒜ℋ-Translit*: Avg. Words ($\mu \pm \sigma$), TTR, and NER Density. The three domains differ substantially in sequence length, lexical diversity (TTR), and named-entity density, providing complementary transliteration challenges.

Domain	Words ($\mu \pm \sigma$)		TTR		
	Ar	Hi	Ar	Hi	NE
Quranic	12.24±9.37	11.78±8.88	0.33	0.49	0.12
Daily-Use	4.01±2.05	4.02±2.06	0.44	0.44	0.20
Bibliographic	6.73±4.88	6.94±5.26	0.32	0.32	1.04

gari matras; and (iii) orthographic correctness, including Unicode consistency, punctuation, spacing conventions, script mixing, and consonant-cluster formation. Annotators correct residual errors remaining after normalization while enforcing consistent pronunciation-preserving transliterations. Detailed annotation guidelines are provided in Appendix A.3.

Annotation Reliability. We measure annotation consistency through an inter-annotator agreement (IAA) study on 2000 randomly selected samples. Each sample is independently evaluated by three annotators using the same validation protocol employed during expert review. Agreement is evaluated across phonetic alignment, vowels realization and Devanagari matras, and orthographic correctness and quantified using Fleiss' κ. The resulting score exceeds 0.85, indicating high agreement among annotators. The observed agreement supports the reliability of ***𝒜ℋ-Translit*** as a benchmark for Arabic–Hindi transliteration research.

3.2 Dataset Statistics

Corpus Overview. The final curated *𝒜ℋ-Translit* corpus contains 44.7K parallel Arabic–Hindi sentence pairs drawn from three linguistically distinct domains: Quranic text, Daily-Use, and Bibliographic records (Table 2). The corpus comprises over 290K Arabic words and 291K Hindi words, with approximately 74K and 88K unique Arabic and Hindi tokens, respectively. The Bibliographic domain contributes the largest number of examples, while Quranic and MSA data provide coverage of religious and conversational language. We release standard train, validation, and test splits for all domains; detailed split statistics are reported in Appendix B.

Domain-Level Diversity. The three domains occupy distinct linguistic regimes that vary substantially in structural complexity, lexical diversity, and semantic content (Table 3). Quranic text contains the longest and most variable sequences (12.24 ± 9.37 words per sentence), reflecting its syntactically rich and compositional structure. In contrast, Daily-Use consists primarily of short conversational expressions (4.01 ± 2.05 words). The Bibliographic domain exhibits intermediate sequence lengths (6.73 ± 4.83 words) but the highest named-entity density (1.04 entities per sentence), owing to its concentration of author and title names. Lexical diversity also differs across domains, with Quranic and MSA data exhibiting higher type-token ratios than the entity-dominated Bibliographic corpus.

Script-Level Characteristics. Beyond domain-level variation, the corpus exhibits systematic differences between Arabic and Devanagari representations. Across domains, Hindi consistently exhibits larger vocabularies and higher type-token ratios than Arabic, with approximately 88K unique Hindi tokens compared to 74K Arabic tokens overall. This pattern is consistent with the transliteration setting, where a single unvocalized Arabic form may correspond to multiple phonetically distinct Hindi transliterations, increasing lexical variation in the target script. Similar effects appear at the character level: Hindi sentences average 47.45 characters compared to 35.76 characters for Arabic. This expansion largely arises because Devanagari explicitly encodes vowels through *matras* and represents consonant clusters orthographically, whereas standard Arabic typically omits them. Additional statistics are provided in Appendix B.

Evaluation Implications. The observed differences in sequence length, lexical diversity, named-entity density, and orthographic representation create complementary transliteration challenges across domains. Cross-script lexical and orthographic asymmetries further introduce ambigu-

ity characteristic of Arabic–Hindi transliteration. These properties make $\mathcal{AH}$-*Translit* a challenging testbed for evaluating generalization and transfer across diverse linguistic conditions.

3.3 Benchmark Design

To avoid evaluation bias toward the largest domain, we construct ***$\mathcal{AH}$-Translit*-Bench**, a balanced 6,000-pair benchmark comprising 2,000 pairs each from the Quranic, MSA, and Bibliographic domains. Equal domain enables fair comparison across heterogeneous linguistic conditions and supports systematic in-domain performance and cross-domain generalizations.

3.4 Ethical Consideration.

All source materials are publicly available and were collected in accordance with their respective terms of use. LLM-generated transliterations were subjected to filtering, normalization, and expert validation before inclusion. As transliteration conventions may vary across speakers and communities, some alternative valid renderings may exist despite our efforts to maintain consistency. We will release ***$\mathcal{AH}$-Translit*** and $\mathcal{AH}$-*Translit-Bench* for research use only.

4 Evaluation

4.1 Evaluation Metrics

We evaluate transliteration quality using complementary orthographic, phonemic, and diagnostic metrics. For orthographic evaluation, we report Character Error Rate (CER) and Word Error Rate (WER) [Morris et al., 2004]. CER serves as the primary metric because even single-character errors, such as misplaced *nukta* markers or omitted vowels, can alter pronunciation.

To assess phonemic fidelity, we convert reference and predicted transliterations into IPA (International Phonetic Alphabet) transcriptions using Epitran [Mortensen et al., 2018] and compute Phoneme Error Rate (PER) and Weighted Phoneme Error Rate (WPER). PER measures phoneme-level edit distance, whereas WPER uses PanPhon articulatory features [Mortensen et al., 2016] to weight substitutions by phonetic similarity, assigning larger penalties to phonetically distant confusions.

Aggregate metrics provide limited insight into the linguistic sources of errors. To facilitate fine-grained analysis, we partition Devanagari characters into four functional layers defined in section 4.2: *Orthographic*, *Vocalic*, *Syllabic*, and *Consonantal*. For each layer L, we compute a survival rate measuring the proportion of reference units preserved

$$SR_L = \max\left(0, 1 - \frac{E_L}{U_L}\right) \times 100,$$

where U_L denotes the number of reference units and E_L the corresponding alignment errors for layer L. This yields Consonant Survival Rate (CSR), Vowel Survival Rate (VSR), Syllabic Survival Rate (SSR), and Orthographic Survival Rate (OSR).

4.2 Linguistic Divergence Analysis

CER measures the number of character-level errors but does not indicate their linguistic impact. A single edit may reflect a punctuation mismatch with negligible effect on transliteration, the omission of a vowel marker that alters pronunciation, a modification to syllable structure, or the substitution of a consonant that changes lexical identity. To enable systematic interpretation of model failures, we introduce a *Linguistic Divergence Analysis Framework* that maps character-level edit operations to linguistically meaningful categories of divergence between Arabic and Hindi orthography.

The framework is motivated by the asymmetry between the two writing systems. Arabic is an abjad in which short vowels are often omitted, whereas Devanagari is an abugida that explicitly encodes vowel realization and syllable structure. Consequently, transliteration errors may affect different *layers* of linguistic representation beyond surface character accuracy.

To capture these distinctions, we organize divergences into four linguistically motivated categories (Table 4): *Orthographic*, *Vocalic*, *Syllabic*, and *Consonantal*. Collectively, these categories span progressively deeper layers of linguistic representation, primarily affecting surface form, pronunciation, syllable structure, and lexical identity, respectively. Additional category descriptions and examples are provided in Appendix E.

Error Attribution Pipeline For each prediction-reference pair, we compute a minimum-edit-distance alignment and extract all insertions, deletions, and substitutions. Each edit is assigned to the affected Devanagari character and mapped to a divergence category using Table 4. The resulting

Table 4: **Character-to-category mapping used in the Linguistic Divergence Analysis framework.** Each character involved in an error (insertion, deletion, or substitution) is attributed to exactly one category: Orthographic, Vocalic, Syllabic, or Consonantal.

Divergence Category	Assigned Characters Hindi (Romanized)
Orthographic	◌़ (nukta), ऽ (avagraha), hyphen (-), space, ०--९ (0–9), . , ? ! ( ) [ ] { } : ; _ / \
Vocalic	अ (a), आ (ā), इ (i), ई (ī), उ (u), ऊ (ū), ऋ (r), ए (e), ऐ (ai), ओ (o), औ (au), अं (aṃ), अः (aḥ), अँ (ã), ◌ा (ā), ◌ि (i), ◌ी (ī), ◌ु (u), ◌ू (ū), ◌ृ (r), ◌े (e), ◌ै (ai), ◌ो (o), ◌ौ (au), ◌ं (ṃ), ◌ः (ḥ), ◌ँ (~), ◌ॄ (r̄), ◌ॢ (l), ◌ॣ (l̄)
Syllabic	Halant (◌्), Danda (।)
Consonantal	क (k), ख (kh), ग (g), घ (gh), च (c), छ (ch), ज (j), झ (jh), ट (ṭ), ठ (ṭh), ड (ḍ), ढ (ḍh), त (t), थ (th), द (d), ध (dh), प (p), फ (ph), ब (b), भ (bh), म (m), य (y), र (r), ल (l), व (v), श (ś), ष (ṣ), स (s), ह (h)

attributions are aggregated to analyze error patterns across domain, model architecture, and training regime.

This framework complements aggregate evaluation metrics by revealing which linguistic representation layers contribute most strongly to transliteration degradation under different evaluation settings.

5 Experiments

Models. We evaluate two character-level supervised baselines: **GRU-Seq2Seq**, a three-layer GRU encoder-decoder with Bahdanau attention [Bahdanau et al., 2015], and a **Vanilla Transformer** [Vaswani et al., 2017, Jayakumar et al., 2026]. We additionally evaluate four multilingual **LLMs** (table 5) in few shot settings; prompting details are provided in Appendix A.2. Supervised decoding uses beam search with beam size $k = 5$.

Training. For each supervised architecture, we train domain-specific specialists on the Quranic, Daily-Use, and Bibliographic training splits; exact splits are reported in Appendix C.2. We further train two mixed-domain generalists using fixed-size 30K-pair corpora: `Equal-Mix` balances domain contributions through upsampling, whereas `Prop-Mix` follows the original AH-Translit domain distribution. Holding corpus size constant isolates the effect of sampling strategy. Models are evaluated on the held-out Quranic, Daily-Use, and Bibliographic benchmark splits, producing both in-domain and cross-domain conditions.

Implementation. All supervised models are trained on four NVIDIA RTX 2080 GPUs (12 GB VRAM). Architectural specifications, optimization settings, prompting details, and training hyperparameters are reported in Appendix C.

6 Results

6.1 Main Results and Cross-Domain Robustness

Across configurations, VanillaT consistently outperforms GRU-Seq2Seq and all LLM baselines, with the gap widening under specialist transfer (table 5). In-domain performance is strong for both supervised architectures, but cross-domain evaluation reveals substantial degradation. The VanillaT Bibliographic specialist increases from 3.16 CER on Bibliographic text to 38.93 on Quranic, while the Daily-Use specialist rises from 4.73 to 42.14 CER under the same transfer. GRU-Seq2Seq exhibits the same pattern, with the Bibliographic and Daily-Use specialists increasing from 3.68 and 5.68 CER in-domain to 39.86 and 40.27 on Quranic, respectively (table 5).

Transfer failures are asymmetric across domains. Transfers into Quranic produce the largest degradations, whereas the reverse direction does not recover consistently across domains. Quranic is also the hardest evaluation setting overall, with the highest average CER and lowest average chrF+ in the benchmark. These results show that strong matched-domain performance does not imply cross-domain generalization (table 5).

6.2 Effect of Mixed-Domain Training

Mixed-domain training is performs best for reducing cross-domain variance. For VanillaT, Prop-Mix and Equal-Mix achieve the best macro CER values (7.57 and 7.60), while reducing cross-domain variance relative to single-domain specialists; GRU-Seq2Seq exhibits the same pattern, with mixed-domain training both lowering macro CER and reducing variance. Equal-Mix also outperforming the Quranic-only specialist on

Table 5: Cross-domain evaluation (CER%). Macro average and standard deviation (Std. Dev.) are computed across the three test domains. **Best** and second-best results within each architecture are highlighted. Best and worst denote the best and worst values in each column, respectively. Mixed-domain Transformer training (`Prop-Mix` and `Equal-Mix`) yields the most robust cross-domain performance.

Architecture	Model	CER (%) ↓			WPER (%) ↓			Consistency ↓	
		Qur.	Daily	Bib.	Qur.	Daily	Bib.	MaCER	Std.
Transformer	Quranic-only	14.74	40.40	46.07	14.69	40.61	48.18	33.74	16.69
	Daily-only	42.14	**4.73**	17.73	43.53	**4.73**	18.12	21.53	18.99
	Bib-only	38.93	23.14	**3.16**	40.57	23.99	**3.21**	21.74	17.93
	Prop-Mix	14.37	5.07	3.28	14.32	5.06	3.34	**7.57**	5.95
	Equal-Mix	**13.48**	5.44	3.87	**13.44**	5.42	3.96	7.60	**5.16**
GRU-Seq2Seq	Quranic-only	**18.39**	40.64	45.35	**18.67**	41.68	48.51	34.79	14.40
	Daily-only	40.27	5.68	19.48	41.40	5.75	20.04	21.81	17.41
	Bib-only	39.86	24.52	3.68	41.59	25.49	3.77	22.69	18.16
	Prop-Mix	15.98	6.39	3.81	16.09	6.55	3.93	**8.73**	6.41
	Equal-Mix	16.54	8.41	5.93	16.72	8.66	6.16	10.29	**5.55**
LLM	Qwen 3.5 2B	70.89	67.73	63.43	73.27	70.12	65.52	67.35	3.74
	Qwen 3.5 4B	49.21	44.58	42.51	51.65	47.62	45.32	45.43	**3.43**
	Qwen 3.5 9B	59.02	47.93	40.06	60.28	49.88	42.84	49.00	9.53
	Gemma 4 26B	**22.11**	**22.15**	**15.87**	**22.37**	**22.10**	**15.92**	**20.04**	3.61

Quranic, where it achieves 13.48 CER versus 14.74 for the Quranic specialist (table 5).

The two mixture strategies expose a small trade-off. Prop-Mix achieves the best macro CER, while Equal-Mix yields lower variance and slightly stronger performance on Quranic. This suggests that broad domain coverage matters more than precise mixture weighting in this setting, and that balanced exposure is especially useful for difficult transfer targets.

6.3 Error Dynamics and Linguistic Preservation

To understand why some settings fail more severely than others, we examine edit composition. Across both VanillaT and GRU-Seq2Seq, deletions are the dominant error type, while insertions are less frequent and substitutions remain comparatively limited (fig. 3). The most brittle transfer settings amplify this behavior, with cross-domain evaluation into Quranic producing especially deletion-heavy error profiles. At the edit level, these deletions are concentrated in realization-sensitive markers such as the halant and short vowel signs, rather than in consonantal material. This indicates that domain shift does not merely increase the error rate, but also changes the structure of failure toward omission-dominant transliteration.

This degradation is not uniform across linguistic layers. Across models and evaluation settings, preservation consistently follows the hierarchy CSR > VSR > SSR, showing that consonantal structure is retained more reliably than vocalic and syllabic information (figs. 4 and 5). As CER increases, VSR and especially SSR decline more rapidly than CSR, indicating that transliteration loss is concentrated in realization-sensitive components rather than in consonantal identity. The same pattern persists across architectures and under length-controlled evaluation, suggesting that this hierarchy reflects a general property of translit-

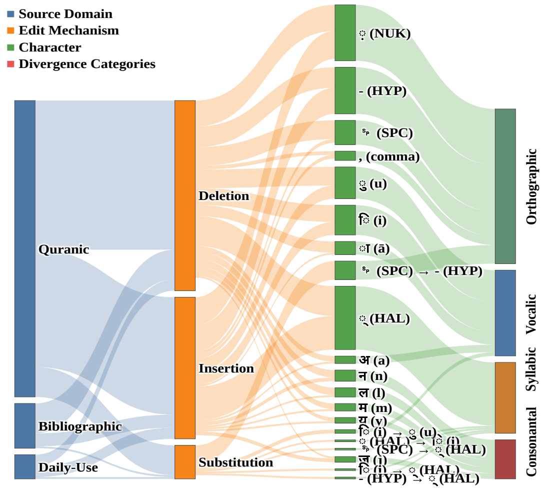


Figure 3: **Character-level error flow for the Transformer (Equal Mix)**. Deletions dominate cross-domain failures, particularly on Quranic transfer, and primarily affect vocalic and syllabic markers rather than consonantal units.

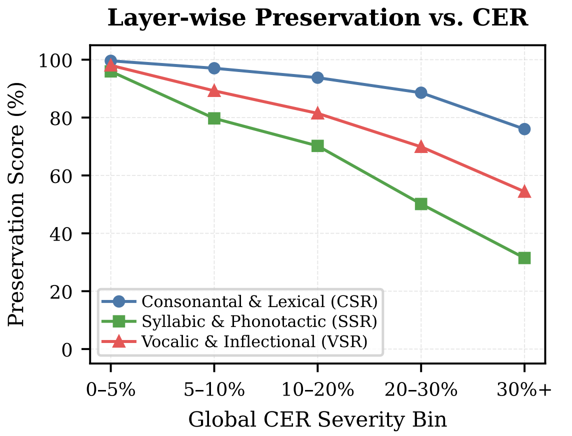


Figure 4: Layer-wise linguistic preservation across CER severity bins. Consonantal structure (CSR) remains highly resilient even as syllabic (SSR) and vocalic (VSR) layers sharply collapse, demonstrating non-uniform degradation.

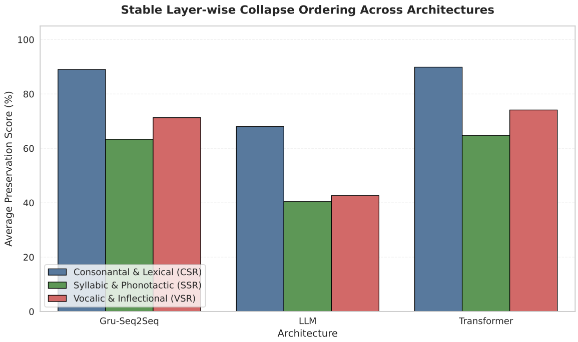


Figure 5: Stable layer-wise collapse ordering across model architectures. Vocalic and syllabic markers are more susceptible to degradation, while consonantal structure remains comparatively robust.

eration degradation rather than a model-specific artifact.

These findings indicate a systematic error profile: deletions dominate and are concentrated in vowel and syllable markers, while consonantal structure is preserved more robustly. This suggests that the bottleneck lies in recovering realization-sensitive information rather than in consonant identity. This aligns with the view of Arabic-to-Hindi transliteration as a reconstruction problem under orthographic asymmetry.

Phonetic Softness To quantify how much character-level error is discounted in phonemic space, we define phonetic softness as

$$\phi = 1 - \mathrm{WPER}/\mathrm{CER}$$

. Higher ϕ indicates a larger gap between orthographic and phonemic error, deviations that remain phonetically similar to the target.

The highest softness values occur under cross-domain transfer. A Quranic-only Transformer evaluated on Daily-Use yields $\phi \approx 9.1\%$, whereas the reverse direction yields $\phi \approx 5.0\%$ (table 5). The larger CER-WPER gap in the former suggests that its errors remain phonetically closer to the reference despite similar CER. We report ϕ as a complementary diagnostic alongside the edit-composition and preservation analyses.

7 Conclusion

We introduce ***AH-Translit***, the first large-scale multi-domain dataset, and *AH-Translit-Bench*, the first benchmark for Arabic-to-Hindi transliteration, along with a Linguistic Divergence Analysis framework that attributes transliteration errors to orthographic, vocalic, syllabic, and consonantal factors. Evaluations show that strong in-domain performance degrades sharply under domain shift. Although mixed-domain training improves domain transferability, errors remain systematic, arising mainly from vocalic and syllabic reconstruction while consonants are largely preserved. These findings suggest that Arabic-to-Hindi transliteration is fundamentally a reconstruction problem: Arabic orthography leaves pronunciation-critical information implicit, whereas Hindi orthography requires it explicitly. Progress will likely depend on better recovery of the vocalic and syllabic information omitted from the source script, and we hope our resources facilitate future research toward this goal.

8 Limitations

Although AH-Translit covers Quranic, contemporary, and entity-rich Arabic, it excludes dialectal Arabic, social media or code-switched content, limiting the scope of our cross-domain analysis. Our interpretation of transliteration as a reconstruction problem is based on character error patterns rather than direct pronunciation annotations. Likewise, our analysis framework attributes errors to linguistic catgeories but doesn't establish the causal mechanisms underlying them. Finally, due to computational constraints, we do not evaluate larger frontier LLMs, leaving their behavior on this benchmark for future work.

References

A.M.K. SIDDIQUI. 2014. JOUDAH INFO-LINGO TECH PVT LTD.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *Preprint*, arXiv:1409.0473.

Assia Boumaraf, Sonia Bekal, and Joël Macoir. 2022. The orthographic ambiguity of the arabic graphic system: Evidence from a case of central agraphia affecting the two routes of spelling. *Behavioural Neurology*, 2022:11 pages.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*. Preprint.

Peter T. Daniels. 1990. Fundamentals of grammatology. *Journal of the American Oriental Society*, 110(4):727–731.

Nadir Durrani and Sarmad Hussain. 2010. A transliteration scheme for urdu-to-hindi/marathi cross-language information retrieval. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 80–85. Association for Computational Linguistics.

Fadhl Eryani and Nizar Habash. 2021. Automatic Romanization of Arabic bibliographic records. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 213–218, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

N.Y. Habash. 2010. *Introduction to Arabic Natural Language Processing*. Synthesis digital library of engineering and computer science. Morgan & Claypool Publishers.

Mohamed Seghir Hadj Ameur, Farid Meziane, and Ahmed Guessoum. 2017. Arabic machine transliteration using an attention-based encoder-decoder model. *Procedia Computer Science*, 117:287–297. Arabic Computational Linguistics.

Soufiane Hajbi, Omayma Amezian, Mouhssine Ziyad, Issame El Kaime, Redouan Korchiyne, and Younes Chihab. 2025. Transformer-based model for moroccan arabizi-to-arabic transliteration using a semi-automatic annotated dataset. *International Journal of Information Management Data Insights*, 5(2):100351.

Yunan Harahap, Talha Ansari, and 1 others. 2023. The impact of transliteration method in the tradition of reading and understanding the quran at madrasah aliyah swasta amaliyah. *Proceeding of The Annual International Conference on Islamic Education*, 2(1):1–13.

HindiQuran.in. 2025. Quran in hindi.

Thanmay Jayakumar, Deepon Halder, and Raj Dabre. 2026. Scripts through time: A survey of the evolving role of transliteration in nlp. *Preprint*, arXiv:2604.18722.

Palakpreet Kaur and Kamal Deep Garg. 2022. Machine transliteration for indian languages: Survey. In *2022 Seventh International Conference on Parallel, Distributed and Grid Computing (PDGC)*, pages 462–467.

Kevin Knight and Jonathan Graehl. 1998. Machine transliteration. *Computational Linguistics*, 24(4):599–612.

Haizhou Li, A Kumaran, Vladimir Pervouchine, and Min Zhang. 2009. Report of NEWS 2009 machine transliteration shared task. In *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration (NEWS 2009)*, pages 1–18, Suntec, Singapore. Association for Computational Linguistics.

Yash Madhani, Sushane Parthan, Priyanka Bedekar, Gokul Nc, Ruchi Khapra, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh Khapra. 2023. Aksharantar: Open Indic-language transliteration datasets and models for the next billion users. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 40–57, Singapore. Association for Computational Linguistics.

Saqib Maqbool, Nisar Ahmed, Gulshan Saleem, and Muhammad Munawar. 2015. An efficient hindi-urdu transliteration system. *Oral Science International*, 27:4549–4553.

Andrew C. Morris, Viktoria Maier, and Phil D. Green. 2004. From wer and ril to mer and wil: improved evaluation measures for connected speech recognition. In *Interspeech*.

David R. Mortensen, Siddharth Dalmia, and Patrick Littell. 2018. Epitran: Precision G2P for many languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

David R. Mortensen, Patrick Littell, Akash Bharadwaj, Kartik Goyal, Chris Dyer, and Lori S. Levin. 2016. Panphon: A resource for mapping IPA segments to articulatory feature vectors. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3475–3484. ACL.

Mihai Nadăş, Laura Dioşan, and Andreea Tomescu. 2025. Synthetic data generation using large language models: Advances in text and code. *IEEE Access*, 13:134615–134633.

Brian Roark, Lawrence Wolf-Sonkin, Christo Kirov, Sabrina J. Mielke, Cibu Johny, Isin Demirsahin, and Keith Hall. 2020. Processing South Asian languages written in the Latin script: the Dakshina dataset. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2413–2423, Marseille, France. European Language Resources Association.

Henry Rogers. 2005. *Writing Systems: A Linguistic Approach*. Blackwell Publishing.

Mohammad Shariq. 2023. Use of arabic by the 'others' in saudi arabia: A sociolinguistic study of communication needs leading to interlanguage development. *Migration Letters*, 20(S7):1334–1343.

Bashar Talafha, Analle Abuammar, and Mahmoud Al-Ayyoub. 2021. Atar: Attention-based lstm for arabizi transliteration. *International Journal of Electrical and Computer Engineering*, 11:2327–2334.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

A Dataset Curation Details

A.1 Curation Summary

Table 6 summarizes the major stages of the dataset curation and validation process.

Table 6: Summary of dataset curation and validation protocols.

Metric	Value
Domains	3
Audit Samples (per domain)	2,000
Validation Samples (per domain)	1,000
Annotators (count)	3
IAA Sample Size	2000
Validation Effort	≈1,620 hrs
Fleiss' κ	085
Agreement Threshold	Majority

A.2 System Prompt

A.3 Annotation Guidelines

Annotators followed a standardized set of operational guidelines throughout the validation process to ensure consistent transliteration decisions across domains.

- Preserve pronunciation over literal character mapping whenever ambiguity exists.
- Ensure that Hindi transliterations remain naturally pronounceable for native Hindi speakers while remaining faithful to Arabic phonetics.
- Avoid hallucinated vowel insertions unless phonologically necessary.
- Maintain consistent transliteration decisions for recurring phonetic patterns throughout the corpus.
- Preserve pronunciation-sensitive structures such as consonant gemination, semivowel transitions, and domain-specific phonetic conventions.
- Correct malformed consonant clusters introduced through noisy scraping or automatic generation.
- Remove unsupported Unicode symbols, formatting artifacts, script-mixing errors, and spacing inconsistencies.

These guidelines were distributed to all annotators prior to validation and were applied consistently throughout the review process.

A.4 Human Validation Protocol

Manual validation was conducted by three bilingual annotators with expertise in Arabic phonology, Hindi orthography, and Devanagari transliteration standards.

Each annotator contributed approximately six hours per day over a three-month period, resulting in nearly 1,620 person-hours devoted to corpus validation, correction, and quality assurance.

For validation, we randomly sampled 1,000 transliteration pairs from each domain after data collection and automatic normalization. All sampled pairs were independently reviewed in a blind-review setting using the common annotation guidelines described in Appendix A.3.

Evaluation focused on three primary criteria:

1. **Phonetic Alignment:** Accurate mapping of Arabic consonants, particularly emphatic, guttural, and pharyngeal sounds, to their closest Hindi phonetic realizations.
2. **Vowel Realization:** Correct rendering of short and long vowels through Devanagari *matras*, including implicit vowels and elongated vowel forms.
3. **Orthographic Correctness:** Detection and correction of malformed consonant clusters, Unicode inconsistencies, punctuation artifacts, duplicated whitespace, unsupported symbols, and script-mixing errors.

Validation additionally emphasized accurate rendering of Arabic diacritics, preservation of pronunciation-sensitive structures, including Quranic recitation patterns, consonant gemination, and semivowel transitions, and correct realization of vowel mappings in Devanagari.

Residual transliteration errors that remained after automatic normalization were manually corrected prior to inclusion in the final benchmark. To reduce reviewer bias and improve consistency, sample ordering and domain assignments were periodically reshuffled during the validation process.

A.5 Inter-Annotator Agreement

To assess annotation consistency, we conducted an inter-annotator agreement (IAA) study on a randomly selected subset of 100 samples.

Each sample was independently evaluated by all three annotators using the same validation protocol employed during dataset construction.

System Prompt

Task: Transliterate the following text from Arabic (Perso-Arabic Script) to Hindi (Devanagari Script).

Rules:

- Perform transliteration only, not translation.
- Preserve pronunciation as closely as possible.
- Output only the Hindi transliteration.
- Do not add explanations or extra text.
- Input and output line must be equal.

Examples:

Ex 1:

السلام عليكم → अस्सलाम अलैकुम

Ex 2:

بسم الله الرحمن الرحيم → बिस्मिल्लाह-हिर्रहमान निर्रहीम

Input:

- {input_text}

Action:

- Now transliterate

Figure 6: **Prompt used for Arabic-to-Hindi transliteration generation.** The structured instruction template defines transliteration constraints, pronunciation-preservation rules, aligned examples, and output formatting requirements used to guide the model during inference across cross-domain Arabic text inputs.

Agreement was assessed across the following dimensions:

1. Arabic–Hindi phonetic alignment.
2. Correct realization of vowels and Devanagari *matras*.
3. Orthographic and formatting correctness.

Multi-rater agreement was quantified using Fleiss' κ. Table 7 reports the agreement statistics.

Table 7: Inter-annotator agreement statistics.

Metric	**Value**
Samples Evaluated	2000
Annotators	3
Agreement Threshold	Majority
Fleiss' κ	085

Samples failing to satisfy the predefined agreement criterion were reviewed and corrected prior to final inclusion in the benchmark.

A.6 Error Audit and Corpus Cleaning

Prior to large-scale validation, we performed a manual audit of 2,000 randomly sampled pairs from each domain to identify recurring artifacts introduced during data acquisition and automatic generation.

The audit revealed several recurring error categories:

- Redundant whitespace and formatting inconsistencies.
- Unsupported Unicode symbols and special characters.
- Punctuation normalization issues.
- Script-mixing artifacts.

- Malformed consonant clusters.
- Inconsistent transliteration of recurring phonetic patterns.
- Non-phonetic character repetitions.
- Garbled text arising from scraping or generation artifacts.

Based on these observations, we designed regex-based normalization rules to remove systematic artifacts before human review.

Representative examples are shown in Table 8.

The normalized corpus was subsequently subjected to manual validation and agreement assessment before inclusion in the final benchmark release.

B Dataset Details

B.1 Detailed Statistics and Linguistic Profiles

Table 9 provides the definitive statistics for the base domains and the 30K-pair mixtures.

B.2 Exploratory Data Analysis and Visualizations

To further characterize the ***AH-Translit*** corpus, we provide a suite of visualizations highlighting scale, structural alignment, and linguistic diversity.

C Experimental and Implementation Details

This section provides the technical specifications required for full reproducibility, documenting our architectural choices, hyperparameter settings, and the multi-stage curation pipeline.

C.1 Training Hyperparameters

Experiments were conducted using both recurrent (GRU-Seq2Seq) and attention-driven (Vanilla Transformer) architectures. Table 10 details the specific configurations used. Both models were trained using the AdamW optimizer with a learning rate of 3×10^{-4} and a batch size of 16.

C.2 Dataset Training Splits

Domain-specific specialist models are trained on the corresponding training splits in table 11. Validation splits are used for model selection, and all reported results are computed on the held-out test sets.

The Equal-Mix corpus is constructed by allocating equal contributions from each domain, with upsampling applied where required. Prop-Mix follows the original domain distribution of ***AH-Translit***. Both mixtures use identical train and validation sizes (table 12).

D Detailed Evaluation Metrics

This appendix provides additional details on the evaluation metrics used throughout the paper. While the main text introduces the evaluation framework at a high level, this section presents the formal definitions, implementation details, and illustrative examples used to assess transliteration quality and analyze linguistic degradation patterns.

D.1 Surface-Level Evaluation Metrics

We report Character Error Rate (CER), Word Error Rate (WER), and Exact Match (EM) as primary orthographic evaluation metrics. CER measures the minimum number of insertions (I), deletions (D), and substitutions (S) required to transform a prediction into the reference sequence, normalized by the reference length:

$$CER = \frac{S + D + I}{|R|} \times 100,$$

where $|R|$ denotes the number of characters in the reference transliteration.

CER serves as the primary evaluation metric because transliteration quality depends on accurate character-level realization of pronunciation. Even seemingly minor orthographic deviations, such as omitted vowels or misplaced *nukta* markers, may alter the intended pronunciation.

Since transliteration is evaluated at the word level, even a single character error typically renders the entire word incorrect. We additionally report Exact Match (EM), which assigns a score of one only when the predicted transliteration exactly matches the reference sequence.

D.2 Phonemic Evaluation Metrics

Orthographic similarity does not necessarily imply phonetic similarity. To evaluate pronunciation preservation, we convert both reference and predicted transliterations into International Phonetic Alphabet (IPA) representations using Epitran [Mortensen et al., 2018].

Given a reference phoneme sequence P_r and predicted phoneme sequence P_p, we compute Phoneme Error Rate (PER) as normalized Levenshtein distance:

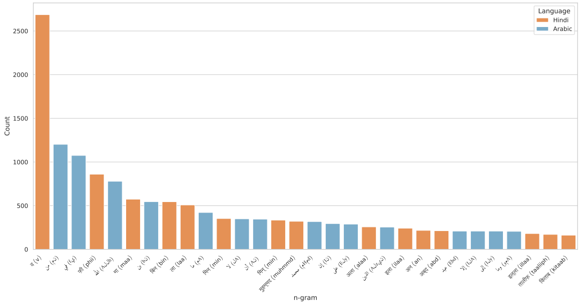


Figure 7: **Top 15 Overall N-Grams.** Joint frequency analysis of common tokens across the merged corpus.

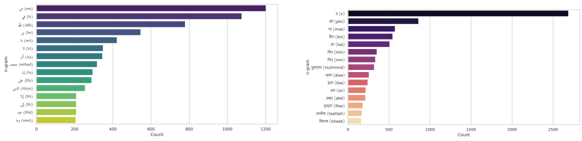


Figure 8: **Top Unigrams.** Comparison of the top 15 Arabic source (left) and Hindi target (right) unigrams.

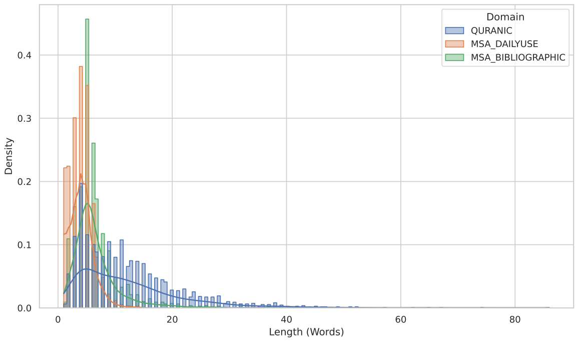


Figure 9: **Arabic Word Lengths.** Domain-wise distribution of sentence lengths measured in words for the Arabic source text.

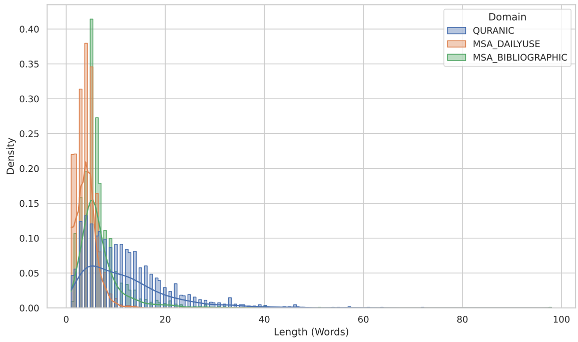


Figure 10: **Hindi Word Lengths.** Domain-wise distribution of sentence lengths measured in words for the Hindi target text.

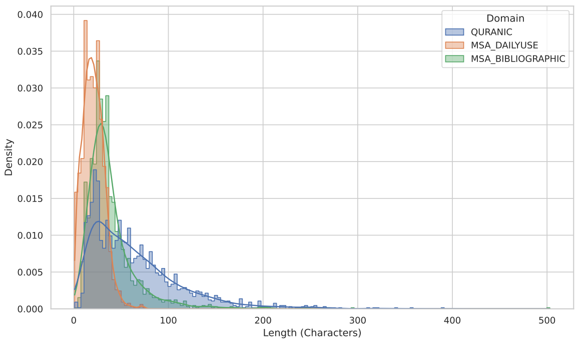


Figure 11: **Arabic Character Lengths.** Character-level sequence length distributions across different text domains.

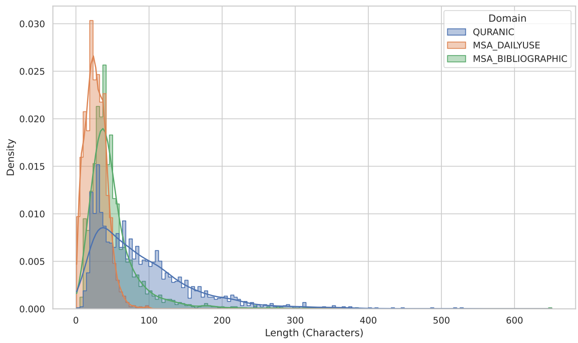


Figure 12: **Hindi Character Lengths.** Character-level sequence length distributions highlighting script expansion in Hindi.

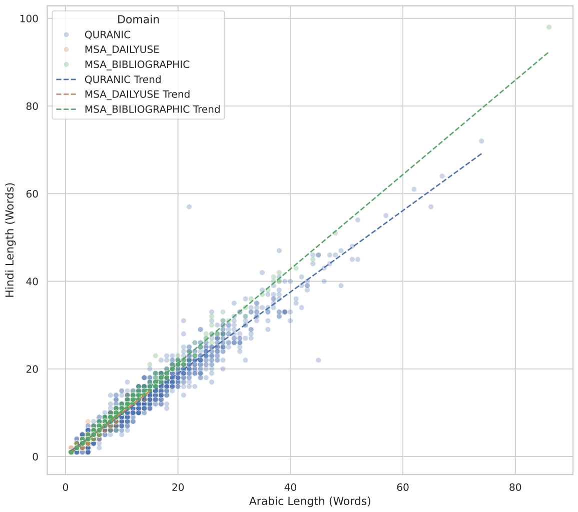


Figure 13: **Word Length Correlation.** Cross-lingual sequence length correlation in terms of word counts across domains.

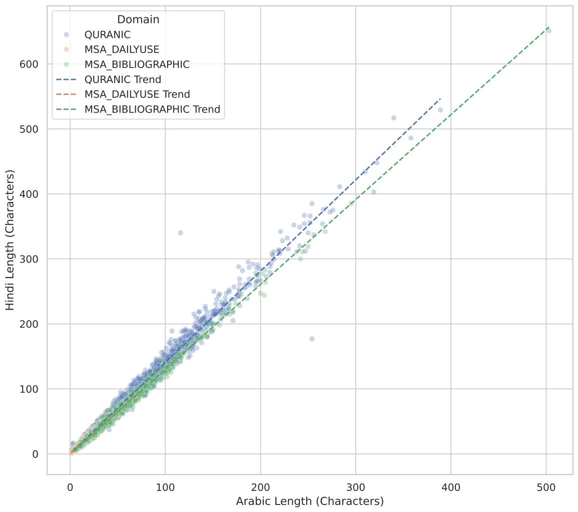


Figure 14: **Character Length Correlation.** Cross-lingual character-level length correlation confirming transliteration parity.

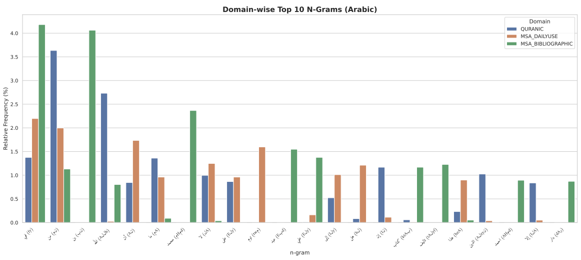


Figure 15: **Domain-wise Arabic N-Grams.** Vocabulary frequency profile comparison of Arabic tokens across domains.

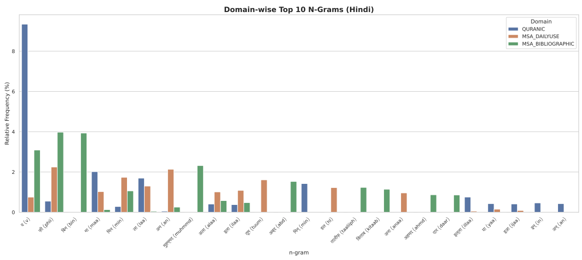


Figure 16: **Domain-wise Hindi N-Grams.** Vocabulary frequency profile comparison of Hindi tokens across domains.

Table 8: Representative examples of corpus-cleaning operations.

Category	Reference	Prediction	Divergence
Orthographic	मिन्हा,	मिन्हा	punctuation deletion
Vocalic	आख़िर	आख़र	vowel-marker deletion
Syllabic	जल्सा	जलसा	halant deletion
Consonantal	तामल	यामल	consonant substitution

Table 9: Linguistic profile of the $\mathcal{AH}$-*Translit* Corpus.

Domain	Lang.	Pairs	Total Words	TTR	Avg. Char	Avg. Word
Quranic	Arabic	6251	76816	0.3288	63.99	12.24
	Hindi	6251	73776	0.4894	89.94	11.78
Daily-Use	Arabic	16833	67466	0.4390	20.94	4.01
	Hindi	16833	67680	0.4449	27.38	4.02
Bibliographic	Arabic	21633	145651	0.3180	39.15	6.73
	Hindi	21633	150213	0.3170	50.71	6.94
Equal-Mix	Arabic	30000	229929	0.2822	41.26	7.66
	Hindi	30000	227147	0.3508	55.99	7.57

Table 10: Training hyperparameters for the primary architectures.

Group	Hyperparameter	GRU-Seq2Seq	Vanilla Transformer
Architecture	Embedding Dimension	128	128
	Hidden Size (Latent Dim)	256	256
	Feedforward Dimension	–	512
	Layers / Attention Heads	3 / 1	3 / 4
Optimization	Optimizer	AdamW	AdamW
	Learning Rate / Weight Decay	3×10^{-4} / 1×10^{-6}	3×10^{-4} / 1×10^{-6}
	Dropout (Enc/Dec/Attn)	0.2 / 0.2 / –	0.2 / 0.2 / 0.1
	Gradient Clipping	1.0	1.0
textitDuration	Epochs (Base Domains)	30	100
	Epochs (Mixture Models)	30	100

Domain	Train	Val.
Quranic	3,825	426
Daily-Use	13,349	1,484
Bibliographic	17,669	1,964

Table 11: Domain-specific train, validation, and test splits used throughout the benchmark.

Mixture	Train	Val.
Equal-Mix	27,000	3,000
Prop-Mix	27,000	3,000

Table 12: Training and validation sizes for mixed-domain generalist models.

$$PER = \frac{\text{Lev}(P_r, P_p)}{\max(|P_r|, |P_p|)} \times 100,$$

where $\text{Lev}(\cdot)$ denotes edit distance.

PER treats all phoneme substitutions equally.

However, not all phoneme confusions are linguistically equivalent. Substituting a phonetically similar sound may have substantially less impact on intelligibility than substituting an articulatorily distant sound.

To capture this distinction, we additionally compute Weighted Phoneme Error Rate (WPER) using PanPhon [Mortensen et al., 2016]. PanPhon represents phonemes as vectors of articulatory features, including voicing, place of articulation, manner of articulation, and nasality. Substitution costs are therefore proportional to phonetic distance rather than being uniformly assigned, allowing WPER to provide a linguistically informed estimate of pronunciation degradation.

D.3 Layer-Specific Survival Metrics

Aggregate performance metrics quantify overall transliteration quality but do not reveal which linguistic components are preserved or degraded. To facilitate fine-grained error analysis, we define layer-specific survival metrics over the linguistic taxonomy introduced in Section 4.2.

For a given linguistic layer L, let U_L denote the number of reference units assigned to that layer and E_L denote the corresponding alignment errors. The survival rate is computed as:

$$SR_L = \max\left(0, 1 - \frac{E_L}{U_L}\right) \times 100,$$

with $SR_L = 100$ when $U_L = E_L = 0$, and $SR_L = 0$ when $U_L = 0$ but $E_L > 0$.

This formulation yields four diagnostic metrics:

- **Consonant Survival Rate (CSR):** Measures preservation of lexical consonantal structure.
- **Vowel Survival Rate (VSR):** Measures preservation of vocalic and inflectional information.
- **Syllabic Survival Rate (SSR):** Measures preservation of syllabic and phonotactic structure.
- **Orthographic Survival Rate (OSR):** Measures preservation of orthographic and stylistic markers.

These metrics are used throughout the analysis sections to identify systematic degradation patterns across domains, linguistic categories, and model families.

D.4 Aggregation

Unless otherwise stated, all reported metrics are macro-averaged across evaluation instances:

$$M_{\text{macro}} = \frac{1}{N}\sum_{i=1}^{N} M_i.$$

Macro-averaging assigns equal weight to every evaluation instance regardless of sequence length and therefore reflects average per-word transliteration quality.

For selected analyses, we additionally report micro-averaged CER:

$$CER_{\text{micro}} = \frac{\sum_i (S_i + D_i + I_i)}{\sum_i |R_i|} \times 100.$$

Unlike macro-averaging, micro-averaging weights examples according to sequence length and provides a corpus-level estimate of transliteration error.

E Linguistic Divergence Framework Details

The Linguistic Divergence Framework (LDF) attributes character-level transliteration errors to linguistically meaningful categories of divergence between Arabic and Hindi orthography. While attribution is performed at the character level, the four categories span progressively deeper layers of linguistic representation:

Orthographic $\rightarrow$ Vocalic $\rightarrow$ Syllabic $\rightarrow$ Consonantal.

These categories are associated with surface form, pronunciation, syllable structure, and lexical identity, respectively. This appendix provides detailed category definitions, attribution rules, and illustrative examples.

E.1 Category Definitions

Orthographic. Formatting and script-conventional symbols such as punctuation, whitespace, hyphenation, and other non-lexical orthographic markers that typically preserve pronunciation and lexical identity.

Vocalic. Independent vowels, dependent vowel markers (*matras*), and nasalization markers that primarily influence vowel realization and pronunciation while often preserving lexical identity.

Syllabic. Symbols that modify syllable structure and consonant clustering, most notably the *halant* (्), thereby affecting phonotactic realization.

Consonantal. Lexical consonants whose insertions, deletions, or substitutions frequently affect lexical distinction and may alter word identity.

E.2 Error Attribution Procedure

For each prediction-reference pair, we compute a minimum-edit-distance alignment and extract all insertions, deletions, and substitutions. Each edit operation is attributed to the affected Devanagari character and assigned to exactly one divergence category according to Table 4. Because every character belongs to a unique category, category assignment is deterministic given an alignment.

Formally, let

$$E = \{e_1, e_2, \ldots, e_n\}$$

denote the set of edit operations extracted from an aligned prediction-reference pair. Each edit is mapped to a divergence category through

$$f : E \rightarrow C,$$

where

$$C = \{\text{Orthographic}, \text{Vocalic}, \text{Syllabic}, \text{Consonantal}\}.$$

Aggregating category assignments across examples yields category-specific error counts used throughout our analyses.

E.3 Interpreting Divergence Profiles

The Linguistic Divergence Framework decomposes CER into linguistically meaningful sources of error rather than defining a new evaluation metric. While CER measures the total number of character-level edits, the LDF identifies the linguistic representation layer affected by each edit.

Consequently, two systems with similar CER values may exhibit markedly different divergence profiles. For example, a model dominated by Vocalic divergences indicates difficulties in vowel realization, whereas one dominated by Consonantal divergences suggests failures that more directly affect lexical distinction and word identity. The framework therefore complements aggregate evaluation metrics by revealing the linguistic sources of transliteration degradation and enabling more interpretable comparisons across domains, architectures, and training settings.

F Extended Results

F.1 Mixture Architecture Performance Comparison

Table 13 summarizes the head-to-head performance of the mixture regimens.

F.2 Qualitative Results

F.3 Silent Failures Detected

Across 60,000 evaluated samples, rigorous validation isolated 6 silent failures. These were entirely localized to the recurrent baseline.

F.4 Global Model Performance Leaderboard

The global leaderboard decisively places the attention-driven Vanilla Transformer above the recurrent GRU baseline. Table 15 represents top-tier aggregated performance.

F.5 Domain Generalization Heatmaps

We visualize cross-domain model generalization behavior via CER and BLEU metric distributions across train-test configurations.

F.6 Side-by-Side Character Transformation Analysis (Top 15)

F.7 Phoneme and Error Type Distributions

We examine the correlation between phonetic alignments (PER) and overall character error distributions.

F.8 Character Confusion Heatmaps (Vanilla Transformer)

We map character-level errors (deletions, insertions, and substitutions) specifically for the best-performing Vanilla Transformer model.

F.9 Error Rate vs. Sequence Length

We analyze how Character Error Rate scales as sentence lengths increase across all three evaluation domains.

F.10 Phonetic Error Flow Analysis

We trace phonetic transformations from source Arabic characters to target Hindi characters through edit operations to categorize errors.

Table 13: **Comprehensive Evaluation Results**. We report exact percentages for Character Error Rate (CER) and Word Error Rate (WER) across all benchmark domains.

		Quranic		Daily-Use		Bibliographic	
Model Variant	**Arch.**	**CER**	**WER**	**CER**	**WER**	**CER**	**WER**
`Equal-Mix`	VanillaT	13.48	53.12	5.44	21.21	3.87	15.62
	GRU	16.54	59.22	8.41	33.91	5.93	24.10
`Prop-Mix`	VanillaT	14.37	54.34	5.07	20.22	3.28	13.52
	GRU	15.98	58.32	6.39	25.54	3.81	16.15

Table 14: Qualitative cross-domain transliteration analysis. We highlight correct mappings in blue and errors or hallucinations in red. Specialist models (`Quran-only`, `MSA-only`, `Bib-only`) consistently hallucinate out-of-domain stylistic artifacts, while the `Equal-Mix` model successfully generalizes across text types.

Model	**Quranic**	**Daily-Use**	**Bibliographic**
Source (Arabic)	ذكرى وما كنّا ظالمين `zikrā va mā kunnā zālimīna`	لا أريدك أن تنساني. `lā urīduka ana tansānī`	بيروت: أنطون سعادة، 1938 `bairūta: aṁtūna sa`^{c}`ādah, 1938`
Target (Hindi)	ज़िकरा व मा कुन्ना ज़ालिमीन `zikrā va mā kunnā zālimīna`	ला उरीदुक अन तन्सानी। `lā urīduka ana tansānī`	बैरूत : अंतून सआदह, 1938 `bairūta : aṁtūna sa`^{c}`ādah, 1938`
Quranic-only	ज़िकरा व मा कुन्ना ज़ालिमीन	ला अरीदुक अन् तन्सानी	बैरूत व अन्तून सआदतिन् गुल्क़न् ज़ाल
Daily-only	ज़क्किरा व मा कुन्ना ज़ालिमीन	ला उरीदुक अन तन्सानी।	बैरूत 1 अन्तून सआदा, 113
Bib-only	ज़िकरा व मा कुना ज़ालमीन	ला उरीदिक अन तिन्सानी	बैरूत : अंतून सआदह, 1938
Equal-Mix	ज़िकरा व मा कुन्ना ज़ालिमीन	ला उरीदुक अन तन्सानी।	बैरूत : अंतून सआदह, 1938
Gemma 4 26B	ज़िकरा व मा कुन्ना ज़ालिमीन	ला उरीदुका अन तन्सानी।	बैरूत : अंतून सआदह, 1938

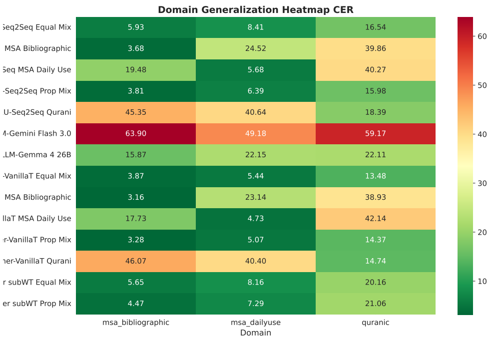


Figure 17: **Domain Generalization Heatmap (CER).** Model performance generalization across all domains measured by Character Error Rate (CER).

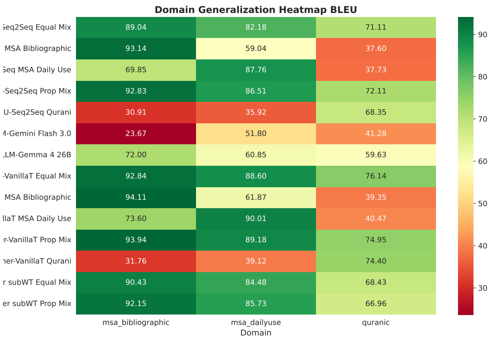


Figure 18: **Domain Generalization Heatmap (BLEU).** Cross-domain architectural robustness measured by BLEU score.

Figure 19: **PER Statistical Analysis Heatmap.** Phoneme Error Rate (PER) distribution variance highlighting critical model errors across domains.

Figure 20: **Model-wise Error Type Distribution.** Frequency distribution of phonetic insertions, deletions, and substitutions.

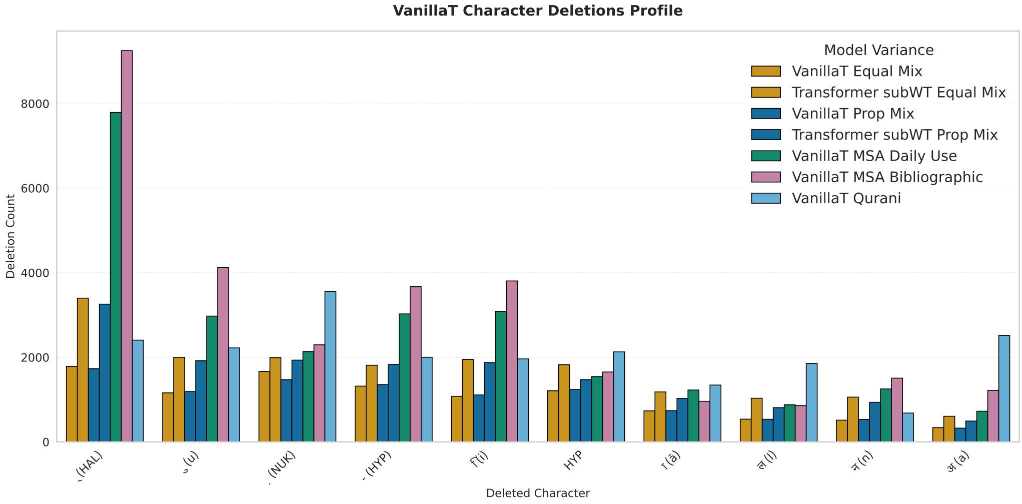


Figure 21: **Top 15 Character Deletions (Vanilla Transformer).** Heatmap illustrating Devanagari characters frequently omitted by the model.

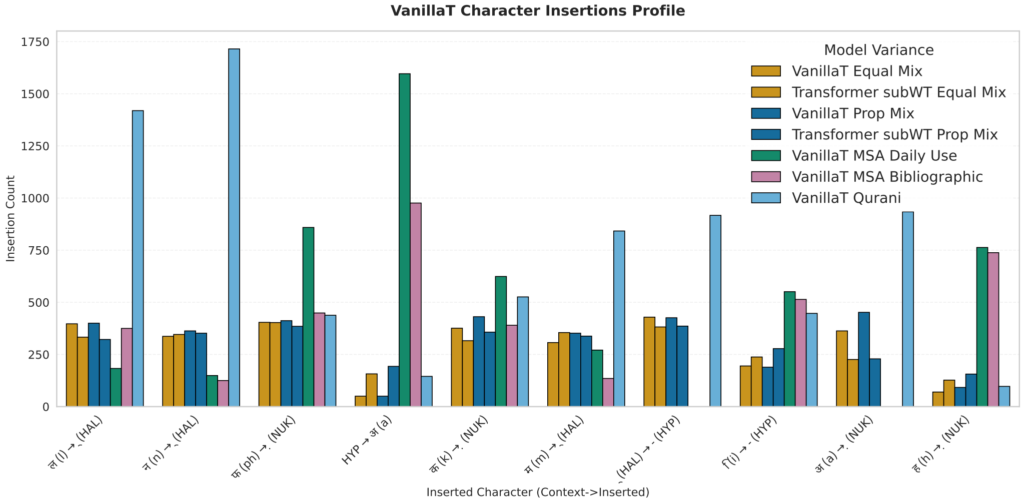


Figure 22: **Top 15 Character Insertions (Vanilla Transformer).** Heatmap detailing spurious characters hallucinated or inserted during transliteration.

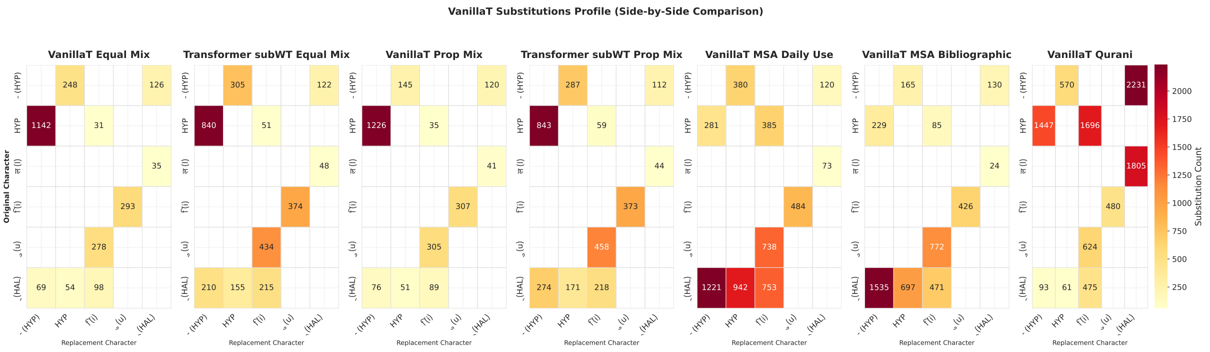
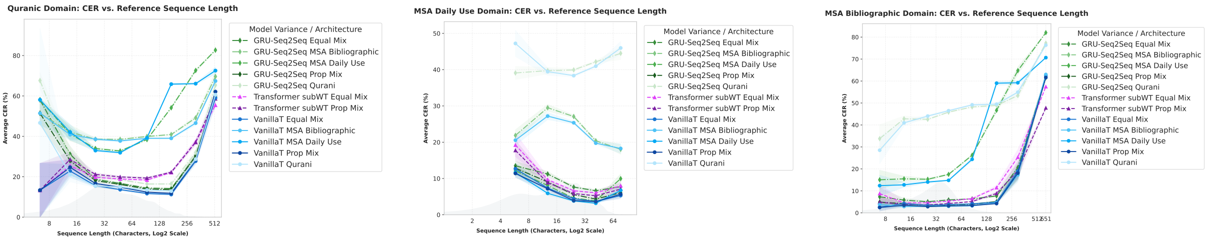


Figure 23: **Top 15 Character Substitutions (Vanilla Transformer).** Heatmap mapping the most common consonant/vowel script confusion patterns.



(a) Quranic Domain

(b) Daily-Use Domain

(c) Bibliographic Domain

Figure 24: **Error Rate vs. Sequence Length.** Scatter plots mapping individual sequence Character Error Rates (CER) against sequence lengths across base domains.

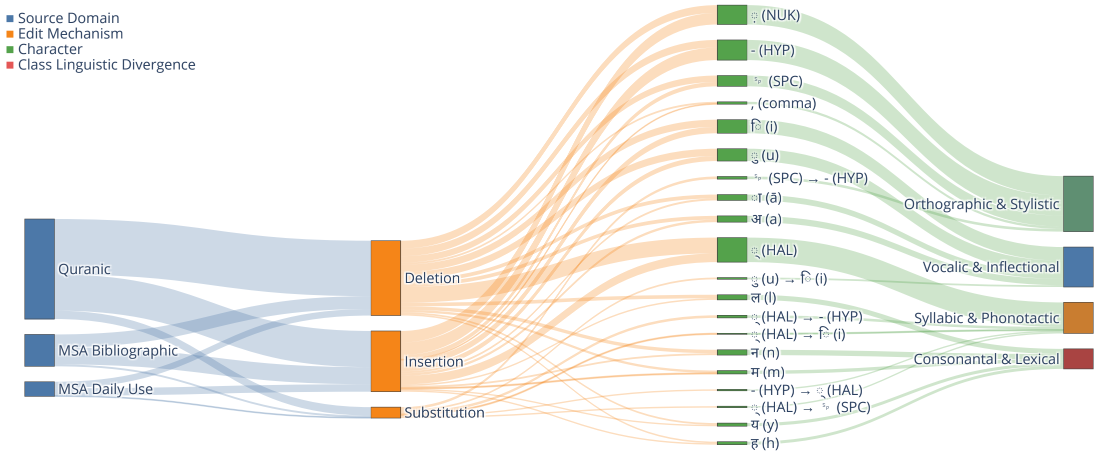


Figure 25: **Global Phonetic Error Flow.** Mapping of transliteration transformations from source text properties to downstream linguistic divergence categories.

Table 15: **Global Performance Leaderboard Overview.** Summary of aggregated models showing character mapping efficiency.

Architecture	Model Variance	Domain	CER (%)	WER (%)	BLEU
VanillaT	VanillaT Prop Mix	Bibliographic	3.28	13.52	62.4
VanillaT	VanillaT Equal Mix	Bibliographic	3.87	15.62	59.8
VanillaT	VanillaT Daily-Use	Daily-Use	4.90	19.34	54.1
VanillaT	VanillaT Prop Mix	Daily-Use	5.07	20.22	52.8
VanillaT	VanillaT Equal Mix	Daily-Use	5.44	21.21	51.5
GRU-Seq2Seq	GRU-Seq2Seq Prop Mix	Bibliographic	3.81	16.15	57.6
GRU-Seq2Seq	GRU-Seq2Seq Equal Mix	Bibliographic	5.93	24.10	49.3
GRU-Seq2Seq	GRU-Seq2Seq Prop Mix	Daily-Use	6.39	25.54	47.9
GRU-Seq2Seq	GRU-Seq2Seq Equal Mix	Daily-Use	8.41	33.91	41.2

Table 16: **Character-Level Error Taxonomy (Top 15 Transformations).** We use the symbolic placeholder (○) to denote combining diacritics/vowels and explicit labels for non-printing characters: **[SPC]** for Space and **[HYP]** for Hyphen. This reveals the "Vocalic Decay" where models struggle with diacritical clusters.

Substitutions (Mapping)				Deletions (Char → ∅)				Insertions (Context → Char)			
Rk	Mapping	Count	%	Rk	Char	Count	%	Rk	Context	Ins	%
1	[SPC] → [HYP]	7980	1.05	1	○ [halant]	49359	6.49	1	la _	○ [hal]	0.75
2	○ [u] → ○ [i]	6619	0.87	2	○ [u-matra]	26154	3.44	2	na _	○ [hal]	0.68
3	○ [hal] → [HYP]	5932	0.78	3	○ [i-matra]	25365	3.34	3	[SPC] _	[a]	0.64
4	[HYP] → ○ [hal]	5721	0.75	4	[HYP]	24261	3.19	4	pha _	○ [nuq]	0.62
5	○ [i] → ○ [u]	4539	0.60	5	○ [nuqta]	23030	3.03	5	ka _	○ [nuq]	0.56
6	[SPC] → ○ [i]	4033	0.53	6	[SPC]	20828	2.74	6	ma _	○ [hal]	0.48
7	la → ○ [hal]	3930	0.52	7	○ [aa-matra]	12104	1.59	7	○ [i] _	[HYP]	0.46
8	○ [hal] → ○ [i]	3686	0.48	8	la (consonant)	12034	1.58	8	la _	○ [i]	0.45
9	○ [aa] → ○ [i]	3643	0.48	9	na (consonant)	11707	1.54	9	ha _	○ [nuq]	0.44
10	[a] → la	3566	0.47	10	[a] (vowel)	11357	1.49	10	○ [hal] _	[HYP]	0.39
11	○ [i] → ○ [ee]	3523	0.46	11	ma (consonant)	7863	1.03	11	[a] _	○ [nuq]	0.38
12	○ [hal] → [SPC]	3079	0.41	12	ya (consonant)	6074	0.80	12	ma _	○ [u]	0.38
13	ja → za	3006	0.40	13	, (comma)	5152	0.68	13	○ [nuq] _	○ [i]	0.38
14	[a] → ○ [hal]	2969	0.39	14	ta (consonant)	4843	0.64	14	ba _	○ [i]	0.35
15	[SPC] → ○ [u]	2934	0.39	15	ha (consonant)	4786	0.63	15	○ [nuq] _	○ [hal]	0.33

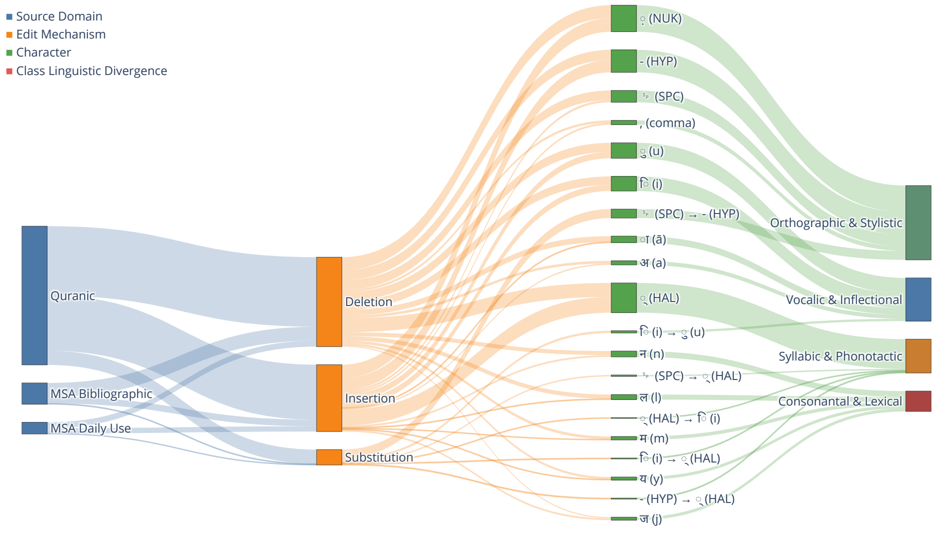


Figure 26: **Vanilla Transformer (Equal-Mix) Error Flow.** Distribution and propagation of character-level error categories for the Vanilla Transformer trained under the Equal-Mix setting.