AyahVerse at AbjadGenEval Shared Task: Monolingual Precision and Cross-Lingual Analysis in Perso-Arabic AI Detection

Fizza Nawaz*, Ibad-ur-Rehman Rashid*, Uswa Abid, Junaid Hussain
Government Post Graduate College, Mansehra,
Affiliated with Hazara University, Pakistan
fizza.nawaz@gcm.edu.pk, ibad@gcm.edu.pk,
uswa7august@gmail.com, junaidbce@gmail.com

Abstract

This paper presents our submission to the AbjadGenEval shared task on AI-generated text detection in Arabic and Urdu. To address the challenges of morphologically rich and low-resource environments, we developed a composite framework leveraging monolingual specialists (AraBERTv2, CAMeLBERT-DA) and multilingual transformers. Our system achieved robust in-domain performance with Test F1-scores of 0.75 for Arabic and 0.86 for Urdu. Methodologically, we tested both raw and normalized text to distinguish whether models detect based on semantic content or on surface artifacts such as punctuation and formatting patterns. Furthermore, our cross-lingual investigations reveal directional performance differences, where Urdu-trained models achieve 0.75 F1 on Arabic, while Arabic-trained models achieve only 0.61 F1 on Urdu. Despite this difference, both directions maintained notably high recall for the machine class, indicating that the model learns cross-lingual machine detection patterns across the Perso-Arabic script. Finally, transfer performance collapsed when internal layers were frozen, demonstrating that full fine-tuning is essential for cross-lingual detection. However, the observed performance differences may partly reflect data imbalance rather than purely linguistic factors.

1 Introduction

The rapid adoption of Large Language Models (LLMs) has significantly increased the availability of automatically generated text, raising concerns about content authenticity and information integrity. As LLM-generated content becomes increasingly fluent, distinguishing machine-generated text from human writing has become a critical challenge, particularly for low-resource languages.

In this paper, we present our submission to the AbjadGenEval shared task on AI-generated text detection for languages using Arabic script (Ezzini et al., 2026), organized as part of the AraGenEval shared tasks (Abudalfa et al., 2025). The shared task focuses on binary classification of human-written versus machine-generated text in Arabic and Urdu, two linguistically distinct but script-sharing languages.

In this work, we contribute: (1) a detection system that evaluates both raw and normalized text; (2) a comparative analysis between MSA-specific and dialect-aware models; and (3) an investigation into cross-lingual transfer comparing unconstrained fine-tuning against layer-freezing ablation to determine the necessity of deep feature adaptation. While we investigate cross-lingual transfer, we note that our analysis is constrained by imbalanced training data sizes (Arabic: 9K, Urdu: 20K), which may contribute to observed performance differences. Our results show that while machine-generated patterns transfer across the Perso-Arabic script family, complex human writing patterns do not transfer. Our code is available at Github.[1]

2 Background

2.1 Related Work

Early detection methods utilized statistical metrics like perplexity via tools such as GLTR (Gehrmann et al., 2019), but the state-of-the-art has shifted toward supervised fine-tuning of transformers. In high-resource settings, RoBERTa-based detectors (Solaiman et al., 2019) are standard. However, morphologically rich languages like Arabic require specialized encoders such as AraBERT (Antoun et al., 2020) to effectively process their complex orthographical features. Recent work emphasizes that LLMs often struggle to replicate specific human patterns such as diacritics and Tatweel,

[1] https://github.com/FizzaNawaz-167/Ayahverse_AbjadGenEval_Sharedtask/

Proceedings of the 2nd Workshop on NLP for Languages Using Arabic Script (AbjadNLP 2026), pages 497–505
March 28, 2026. ©2026 Association for Computational Linguistics

making preprocessing a critical component of Arabic AI forensics (Alshammari and Elleithy, 2024).

Low-Resource Urdu Context: Research on Urdu AI detection remains sparse compared to Arabic. While the HLU dataset (Ali et al., 2025) provides a foundation for Urdu paragraph and sentence-level classification, baseline performance using multilingual models like mBERT and XLM-R shows a significant gap compared to high-resource languages. Frequent code-switching and informal online text further complicate detection (Ammar et al., 2025).

Cross-Lingual Forensics: Cross-lingual transfer via XLM-R has proven effective for semantic tasks like NER and Sentiment Analysis by aligning embedding spaces (Conneau et al., 2020). However, the transferability of generative artifacts remains underexplored. Recent multi-domain detection initiatives (Lamsiyah et al., 2025) have highlighted the complexity of identifying AI-generated content across diverse linguistic and topical contexts. However, It is currently unknown whether the structural statistical signatures of LLMs transcend linguistic boundaries within the Perso-Arabic script family, where languages share a script but differ vastly in syntax. Our work addresses this gap by investigating if "machineness" can be detected cross-lingually between Arabic and Urdu.

2.2 Task Setup

The AbjadGenEval shared task involves binary classification to distinguish human-written from AI-generated content to build a robust detection system across various news genres (politics, technology, sports). The system takes raw text as input and predicts a binary label: *Human or Machine.*

2.3 Dataset

The dataset (Ezzini et al., 2026) comprises human-written content from verified news platforms and AI-generated text, with statistics detailed in Table 1. The `Arabic` dataset is low-resource and high-quality, while the `Urdu` dataset is larger. For out-of-domain evaluation, we tested on 2000 scientific abstracts, 50% human-written and 50% generated by `AL-LaM` from the KFUPM-JRCAI dataset (Al-Shaibani and Ahmed, 2025).

2.4 Tracks

We participated in both tracks, Arabic and Urdu, of this task.

Table 1: Dataset statistics

Language	**Content Source**	**Train**	**Test**	**Characteristics**
Arabic	Verified News	9,289	200	Low-resource, clean
Urdu	News & Web Scrape	20,776	2,630	Large-scale, noisy
Arabic (OOD)	Scientific Abstracts	–	*2,000*	Out-of-Domain (KFUPM)

3 System Overview

Our detection system (Figure 1) employs complementary preprocessing pipelines, domain-aligned transformer models, and cross-lingual transfer experiments. Below we detail each component's design rationale.

3.1 Feature Isolation Strategy

Our system addresses a fundamental question: Do detectors rely primarily on semantic content or on surface-level artifacts like punctuation and formatting?

We implement complementary pipelines to isolate these factors:

1. **Artifact-Preserving Pipelines (B & D)**: Retain all surface features (punctuation, diacritics, formatting) that may contain LLM-specific stylistic fingerprints.

2. **Semantic-Normalized Pipelines (A & C)**: We normalize the text by removing punctuation and standardizing orthography, forcing models to rely on linguistic content.

By comparing performance across these pipelines (detailed in Appendix B), we determine whether detection relies on deep semantic understanding versus surface artifact memorization.

3.2 Model Selection

1. Monolingual Specialist Systems: For the official leaderboard, we prioritized domain-specific models for fine-tuning including AraBERTv2, CAMeLBERT-DA, ArabicBERT, and also a multilingual model (mBERT) for Urdu, chosen for its superior cross-lingual alignment capabilities compared to available monolingual baselines (See Appendix C for details).

2. Cross-Lingual Transfer System: To analyze transferability across the Perso-Arabic script family, we employed XLM-R. Its shared vocabulary enables testing whether machine-generated statistical patterns transfer across languages sharing the Arabic script. Additionally, we evaluated mBERT specifically for the Urdu-to-Arabic direction to compare the effectiveness of cross-lingual transfer between different multilingual models.

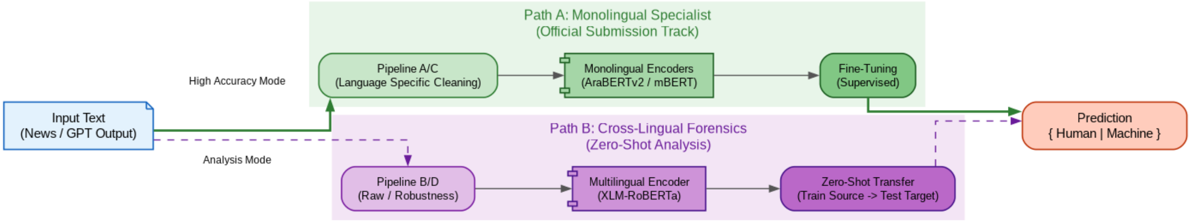


Figure 1: **System overview**

3.3 Cross-Lingual Investigation

Our cross-lingual analysis addresses two practical questions:

1. **Representation Sufficiency**: Does freezing encoder layers (preserving pre-trained multilingual features) maintain cross-lingual detection capability?

2. **Data Scaling Effects**: Given the 2:1 data imbalance (Urdu: 20K, Arabic: 9K), do cross-lingual transfer performance differences stem from dataset size or script/language factors?

These inform whether full fine-tuning is necessary and how data requirements affect multilingual deployment.

4 Experimental Setup

4.1 Data Preparation

We utilized the official and external datasets. To ensure robust evaluation, we trained every model on official training set, and validated on an external dataset (Al-Shaibani and Ahmed, 2025).

4.2 Preprocessing Configurations

To isolate the features contributing to detection, we compared two strategies per language:

Arabic: We contrasted `Pipeline A` (Semantic Normalization) against `Pipeline B` (Artifact Preservation). Following our previous (Rashid and Hashir Khalil, 2025) configurations, Pipeline A unifies orthography and strips punctuation, forcing reliance on linguistic content, while Pipeline B preserves punctuation to detect stylometric fingerprints.

Urdu: We compared `Pipeline C` (Semantic Normalization), which strips punctuation, emojis, and non-Urdu characters, against `Pipeline D` (Artifact Preservation), which retains all artifacts.

Details for each pipeline is given in Appendix B)

4.3 Manifold Visualization

To analyze feature space organization for cross-lingual transfer, we employed t-SNE visualizations of [CLS] token embeddings (as shown in Figure 4) across three conditions:

Untrained Baseline: Embeddings from an untrained model showing no pre-existing class separation.

Cross-Lingual Transfer: Clustering behavior of a model trained on the source language and tested on unseen target language data.

Frozen Ablation: Embedding collapse when internal layers are frozen during training, illustrating the failure of static representations (as shown in Figure 5).

4.4 Implementation Details

All models were implemented using HuggingFace Transformers in Google Colab. We used AdamW optimizer with learning rate 2×10^{-5}, batch size 32, and trained for 2-6 epochs based on validation performance. For the cross-lingual frozen-layer ablation, we froze the bottom six encoder layers while fine-tuning only the top six layers and classification head. Results with different configurations are shown in Table 2 and details are shown in Table 4

5 Results and Discussion

Leaderboard Results: Our models achieved near-perfect training performance (0.99 F1) but faced a generalization gap on the official test set: Arabic detection scored 0.75 F1 utilizing AraBERTv2 (submitted under name AyahVerse, ranking 6th out of 12 participants), while Urdu detection achieved 0.87 F1 utilizing mBERT (submitted under the name Ibad-ur-Rehman, ranking 6th out of 7 participants).

Table 2: Performance summary across development, test, and external datasets.

Track	Model	Dev F1	Test F1
Official Evaluation			
Arabic	AraBERTv2 (Pipeline B)	0.99	**0.75**
Arabic	CAMeLBERT-DA	0.97	0.64
Urdu	mBERT (Pipeline D)	1.00	**0.87**
External Dataset (KFUPM)			
Arabic	AraBERTv2 (Pipeline B)	0.99	0.95
Arabic	ArabicBERT (Pipeline A)	0.99	0.90
Cross-Lingual			
Urdu→Arabic	XLM-R	0.99	0.75
Arabic→Urdu	XLM-R	0.95	0.61
Cross-Lingual Frozen Layers			
Urdu→Arabic	XLM-R	0.99	0.52
Arabic→Urdu	XLM-R	0.95	0.58

5.1 Monolingual Performance

Table 4 presents the performance of our monolingual specialists. On the internal development sets, all models achieved near-perfect F1-scores (0.99), indicating that the models easily distinguished between the machine text characteristics and human sources in the training distribution.

However, evaluation on the official test set revealed a significant generalization gap:

- **Arabic:** Performance initially dropped to an F1 of 0.61 (AraBERTv2). After retraining, AraBERTv2 with frozen layers, early stopping, and Pipeline B F1-score improved to 0.75 F1.

- **Urdu:** The mBERT model with Pipeline D achieved a strong F1 of 0.87. This is likely due to the larger training corpus resulting in better generalization.

Models Comparison: AraBERTv2's superior performance stems from domain alignment. It is pre-trained on formal news (MSA), matching the task distribution, whereas CAMeLBERT-DA is pre-trained on social media text, misaligned with our formal MSA news domain. Performance difference is shown in Table 4.

Robustness Check (External Dataset): On the external KFUPM-JRCAI dataset (AL-LaM abstracts), models achieved 0.90-0.95 F1, higher than the official test. Pipeline B (0.95 F1) outperformed Pipeline A (0.90), suggesting punctuation patterns generalize to unseen generators.

5.2 Cross-Lingual Analysis

Experiments reveal a directional performance differences where Urdu → Arabic (XLM-R) achieves 0.75 F1, while the reverse direction achieves a F1 of 0.61 likely due to smaller Arabic dataset.

XLM-R shows high precision in identifying machine-generated text (0.91 Machine Recall), effectively detecting generation artifacts. In contrast, mBERT is more robust to human variation (0.76 Human Recall) but less sensitive to subtle machine patterns (0.48 Machine Recall). This suggests XLM-R's larger capacity captures cross-lingual AI artifact patterns better than mBERT.

Table 3: Zero-shot cross-lingual transfer performance. There is 2:1 training data disparity between Urdu (20k) and Arabic (9k).

Train (Source)	Test (Target)	Macro F1	Recall (Machine)
Urdu (20k)	Arabic	**0.75**	High (0.91)
Arabic (9k)	Urdu	0.61	High (0.71)

5.3 Ablation Study

Artifact vs. Semantic Features: Across all test conditions, artifact-preserving pipelines (B/D) consistently matched or outperformed semantic pipelines (A/C): Arabic external test (0.95 compared to 0.90 F1), Urdu test (0.87 compared to 0.70 F1). This pattern shows surface artifacts provide more robust detection signals than semantic content alone

Layer Freezing Impact: While freezing layers improved monolingual performance stability, it significantly degraded cross-lingual transfer, demonstrating that full fine-tuning is essential. The Urdu→Arabic advantage (0.75 F1) outperforming Arabic→Urdu (0.61 F1) likely reflects data imbalance more than inherent transfer difficulty.

6 Conclusion

Our monolingual models achieved strong in-domain performance (Arabic: 0.75 F1, Urdu: 0.87 F1), with artifact-preserving preprocessing outperforming semantic normalization, confirming that surface features helps in detection. Our cross-lingual results show that cross-lingual AI detection works partially across Perso-Arabic languages. Urdu→Arabic transfer (0.75 F1) works with more data, while Arabic→Urdu (0.61 F1) struggles with less data, suggesting overfitting to language-specific patterns. Freezing layers failed, proving full fine-tuning is needed to learn cross-lingual detection patterns.

The dataset size imbalance limits our ability to isolate true cross-lingual transfer effects from data availability effects. Future work needs balanced data and testing with more Perso-Arabic languages like Persian to separate script from language influences.

References

Shadi Abudalfa, Saad Ezzini, Ahmed Abdelali, Hamza Alami, Abdessamad Benlahbib, Salmane Chafik, Mo El-Haj, Abdelkader El Mahdaouy, Mustafa Jarrar, Salima Lamsiyah, and Hamzah Luqman. 2025. The AraGenEval shared task on Arabic authorship style transfer and AI generated text detection. In *Proceedings of The Third Arabic Natural Language Processing Conference at EMNLP 2025*, Suzhou, China. Association for Computational Linguistics.

Maged S Al-Shaibani and Moataz Ahmed. 2025. The arabic ai fingerprint: Stylometric analysis and detection of large language models text. *arXiv preprint arXiv:2505.23276*.

Iqra Ali, Jesse Atuhurra, Hidetaka Kamigaito, and Taro Watanabe. 2025. HLU: Human vs LLM generated text detection dataset for Urdu at multiple granularities. In *Proceedings of the 31st International Conference on Computational Linguistics*, page 3495–3510, Abu Dhabi, UAE. Association for Computational Linguistics.

Hamed Alshammari and Khaled Elleithy. 2024. Toward robust arabic ai-generated text detection: Tackling diacritics challenges. *Information*, 15(7).

Muhammad Ammar, Hadiya Murad Hadi, and Usman Majeed Butt. 2025. AI-Generated Text Detection in Low-Resource Languages: A Case Study on Urdu. *Preprint*, arXiv:2510.16573.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. AraBERT: Transformer-based model for Arabic language understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, page 9–15, Marseille, France. European Language Resources Association.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, page 8440–8451. Association for Computational Linguistics.

Saad Ezzini, Irfan Ahmed, Salmane Chafik, Shadi Abudalfa, Mo El-Haj, Ahmed Abdelali, Mustafa Jarrar, Nadir Durrani, Hassan Sajjad, and Farah Adeeba. 2026. Abjadgeneval: Abjad ai generated text detection shared task for languages using arabic script at abjadnlp 2026. In *Proceedings of the 2nd Workshop on NLP for Languages Using Arabic Script (AbjadNLP 2026), co-located with the 19th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2026)*, Rabat, Morocco.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander Rush. 2019. GLTR: Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, page 111–116, Florence, Italy. Association for Computational Linguistics.

Salima Lamsiyah, Saad Ezzini, Abdelkader El Mahdaouy, Hamza Alami, Abdessamad Benlahbib, Samir El amrany, Salmane Chafik, and Hicham Hammouchi. 2025. M-DAIGT: A Shared Task on Multi-Domain Detection of AI-Generated Text. In *Proceedings of the Shared Task on Multi-Domain Detection of AI-Generated Text*, page 1–9, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Ibad-ur-Rehman Rashid and Muhammad Hashir Khalil. 2025. AyahVerse at MAHED shared task: Fine-tuning ArabicBERT with preprocessing for hope and hate detection. In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, page 670–676, Suzhou, China. Association for Computational Linguistics.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook

Kim, Sarah Kreps, Miles McCain, Alex Newhouse, Jason Blazakis, Kris McGuffie, and Jasmine Wang. 2019. Release strategies and the social impacts of language models. *Preprint*, arXiv:1908.09203.

A Figures

B Preprocessing Details

We implemented four preprocessing pipelines to isolate whether detection relies on surface artifacts or semantic content.

B.1 Arabic Pipelines

Pipeline A: Semantic Normalization (Baseline) Designed to remove non-semantic artifacts. Steps include:

- **Orthographic Normalization:** Unifying forms of Alif (أ/آ/إ → ا), Yaa (ى → ي), and Ta Marbuta (ة → ه).
- **Diacritic & Tatweel Removal:** Stripping Tashkeel and Kashida to prevent overfitting to auto-diacritization artifacts found in human news text.
- **Punctuation Removal:** Stripping all punctuation to remove stylometric fingerprints.

Pipeline B: Artifact Preservation (PPA) Identical to Pipeline A, but **punctuation is preserved**. This pipeline examines whether the detector relies on statistical punctuation artifacts (e.g., excessive comma usage) common in generative models.

B.2 Urdu Pipelines

Pipeline C: Script Filtering Designed to handle the high noise in the Urdu dataset.

- **Unicode Filtering:** We remove characters outside the Arabic/Urdu block (`0600-06FF`) to eliminate English code-switching.

Pipeline D: Raw Input Bypasses cleaning entirely to feed raw text into the tokenizer. This tests model robustness against real-world noise, including English code-switching and informal internet formatting.

C Monolingual Model Details

- **AraBERTv2:** Selected for its strong alignment with Modern Standard Arabic syntax and news domain pre-training. (Antoun et al., 2020)
- **CAMeLBERT-DA:** Trained on social media text to capture natural human informality, which contrasts with the rigid, perfect Arabic usually produced by AI models.
- **ArabicBERT:** Leveraged for its massive 95GB pre-training corpus to detect subtle generation artifacts.
- **mBERT (Urdu):** Selected for its superior multilingual alignment capabilities, compared to Urdu specific baselines like Roberta-Urdu.

D Tables

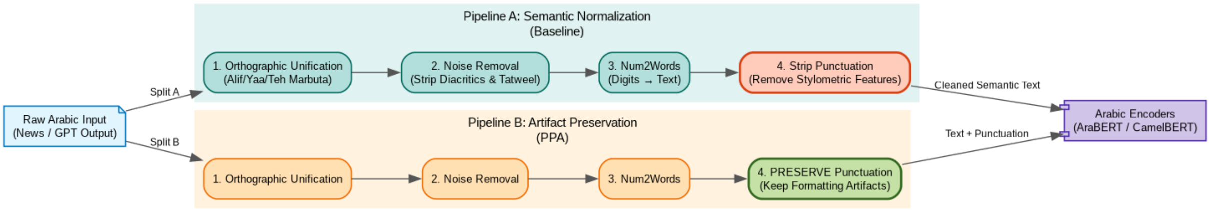


Figure 2: **Preprocessing Pipelines for Arabic**

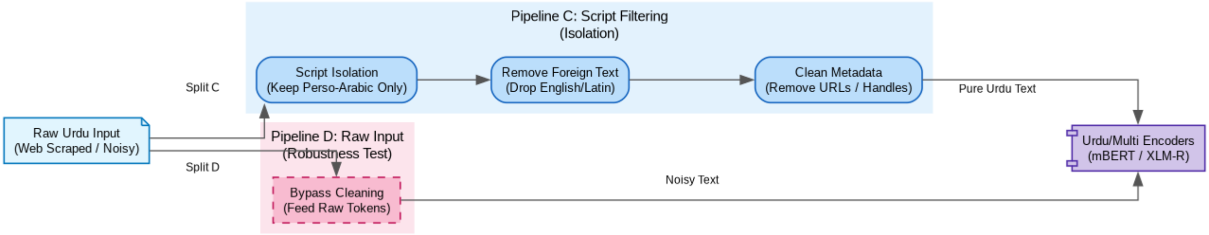
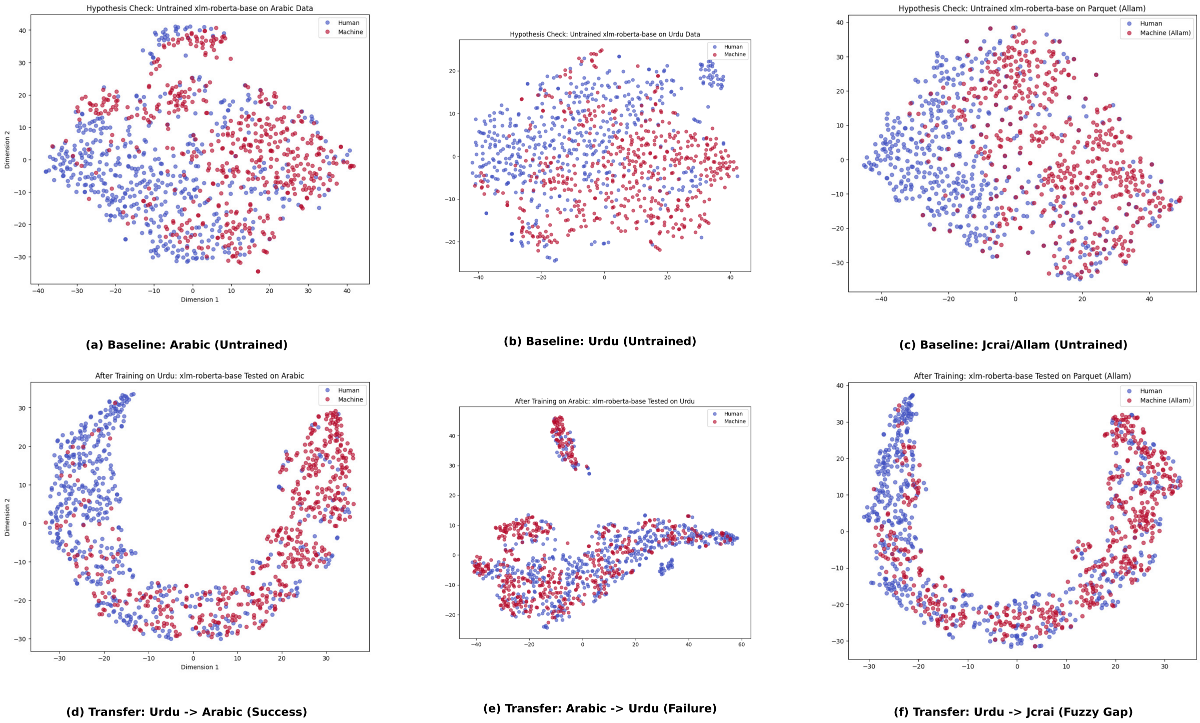


Figure 3: **Preprocessing Pipelines for Urdu**



(a) Baseline: Arabic (Untrained)

(b) Baseline: Urdu (Untrained)

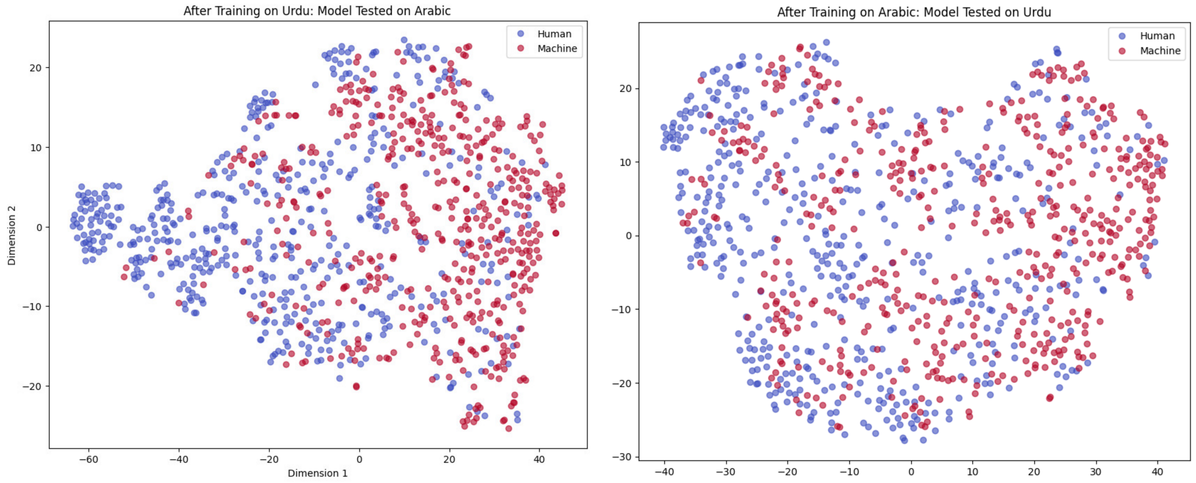
(c) Baseline: Jcrai/Allam (Untrained)

(d) Transfer: Urdu -> Arabic (Success)

(e) Transfer: Arabic -> Urdu (Failure)

(f) Transfer: Urdu -> Jcrai (Fuzzy Gap)

Figure 4: **Cross-lingual transfer visualization using t-SNE on [CLS] embeddings.** Baseline (a-c) shows no pre-existing separation. Transfer results show UrduArabic success (d), Arabic → Urdu Under performing (e), and out-of-domain Urdu→JRCAI partial transfer (f).



(a) Urdu -> Arabic (Frozen Layers)
Result: Manifold Collapse

(b) Arabic -> Urdu (Frozen Layers)
Result: High Entropy (Mixed)

Figure 5: **Impact of freezing internal layers on cross-lingual transfer.** (a) Urdu→Arabic with frozen layers shows manifold collapse, demonstrating that static embeddings cannot maintain transferability. (b) Arabic→Urdu with frozen layers exhibits high entropy mixing, confirming the necessity of deep layer fine-tuning.

Table 4: **Comprehensive performance analysis. Pipeline A:** Semantic Normalization (removes all non-linguistic artifacts). **Pipeline B:** Artifact Preservation (retains punctuation and formatting). **Pipeline C:** Semantic Normalization. **Pipeline D:** Raw Baseline (retains all artifacts). (-) denotes unavailable results.

Dataset	**Model**	**Preproc. Pipeline**	**Macro-F1**	**Accuracy**	**Precision**	**Recall**
		Development Set (Internal) Evaluation				
Arabic (5.3k)	AraBERTv2	Pipeline A	0.99	0.99	0.99	0.99
	AraBERTv2	Pipeline B	0.99	1.00	1.00	1.00
	ArabicBERT	Pipeline A	0.99	1.00	1.00	1.00
	CAMeLBERT-DA(2-epochs)	Pipeline A	0.97	0.98	0.98	0.98
	CAMeLBERT-DA(4-epochs)	Pipeline A	0.98	0.98	0.98	0.98
	CAMeLBERT-DA(6-epochs)	Pipeline A	0.97	0.97	0.98	0.97
Urdu (11.9k)	mBERT	Pipeline C	0.99	0.99	0.99	0.99
	mBERT	Pipeline D	1.00	1.00	1.00	1.00
		Official Test Set (AbjadGenEval)				
Arabic (2k)	AraBERTv2(6-epochs)	Pipeline B	0.61	-	-	-
	AraBERTv2(2-epochs)	Pipeline B	0.70	-	-	-
	AraBERTv2(early-stopping + freezing layers)	Pipeline B	**0.75**	0.72	0.68	0.84
	CAMeLBERT-DA	Pipeline B	0.64	-	-	-
Urdu	mBERT	Pipeline C	0.70	0.70	0.70	0.70
	mBERT	Pipeline D	**0.87**	0.86	0.84	0.89
		External Dataset for Evaluation (KFUPM-JRCAI)				
Arabic	AraBERTv2	Pipeline A	0.90	0.90	0.90	0.90
	AraBERTv2	**Pipeline B**	**0.95**	0.95	0.95	0.95
	ArabicBERT	Pipeline A	0.90	0.90	0.90	0.90
	CAMeLBERT-DA(2-epochs)	Pipeline A	0.79	0.80	0.83	0.80
	CAMeLBERT-DA(4-epochs)	Pipeline A	0.79	0.79	0.83	0.79
	CAMeLBERT-DA(6-epochs)	Pipeline A	0.71	0.73	0.80	0.73

Table 5: **Cross-Lingual Analysis:** Zero-shot transfer results. XLM-R yields High Machine Recall(Recall for machine class), showing **strict sensitivity to machine artifacts**, while mBERT yields High Human Recall, suggesting mBERT is more **robust to human stylistic variations**.

		Class: Human			**Class: Machine**			**Overall**	
Experiment	**Dataset Split**	**Prec**	**Rec**	**F1**	**Prec**	**Rec**	**F1**	**Acc**	**Macro-F1**
Train: Arabic → Test: Urdu									
	Source (Train: Arabic) XLM-R	0.98	0.92	0.95	0.92	0.98	0.95	0.95	0.95
	Target (Test: Urdu) XLM-R	0.66	0.48	0.56	0.59	**0.75**	0.66	0.62	0.61
	Target (Test: Urdu) XLM-R (Freezed Layers during training)	0.66	0.41	0.50	0.57	**0.79**	0.66	0.60	0.58
Train: Urdu → Test: Arabic									
	Target (Test: Arabic) XLM-R	0.87	0.60	0.71	0.69	**0.91**	0.79	0.75	**0.75**
	Target (Test: Arabic) XLM-R (Freezed Layers during training)	**0.92**	0.21	0.34	0.55	**0.98**	0.70	0.59	0.52
	Target (Test: Arabic) mBERT	0.59	**0.76**	0.66	0.66	0.48	0.55	0.62	0.62
	Target (Test: JRCAI) XLM-R	0.89	0.14	0.25	0.53	**0.98**	0.69	0.56	0.47