CORPORUM
Journal of Corpus Linguistics
journal homepage: https://journals.au.edu.pk/ojscrc/index.php/crc

Compact Transformer Models for Classical Urdu Poetry: A Computational Stylistics Approach

Aakash Meghwar[1*] Muhammad Shaharyar Nasir[2]

1. *Graduate Student, Applied Linguistics and Text Analytics, Higher School of Economics, Moscow, Russia*
aakashmeghwar01@gmail.com
2. *Independent Researcher*
m.shaharyarnasir@gmail.com

A R T I C L E I N F O

Keywords:

Arud Prosody, Compact Transformers, Computational stylistics, Low-Resource NLP Prosodic Analysis, Parameter-Efficient Fine-Tuning, Transformer Models, Urdu Poetry

A B S T R A C T

Ghazals and nazms are two examples of classical Urdu poetry, which are sophisticated literary traditions distinguished by strict prosodic rules and multi-layered semantic complexity. Despite Urdu's cultural importance and the analytical possibilities it presents for computational stylistics, systematic NLP-driven research on Urdu poetic forms is still conspicuously lacking in the academic community. By presenting a computational framework that uses compact transformer architectures to examine stylistic phenomena in classic works by Mirza Ghalib, Meer Taqi Mir, Allama Iqbal, and Faiz Ahmad Faiz, this paper fills the gap. We present a carefully selected corpus of 250 verses from public domain sources, along with a theoretically grounded annotation schema that includes prosodic meter (classified using the Arud system), rhyme structure, affective registers, and metaphorical constructions. Anchoring on Low-Rank Adaptation (LoRA) for parameter-efficient fine-tuning, we show that compact models can perform competitively. Our models require 70% fewer parameters than traditional architectures while achieving stylistic clustering (silhouette coefficient: 0.65), rhyme detection (F1: 0.82), and meter classification (F1: 0.78). While culturally informed prompt engineering improves sentiment classification accuracy by 15%, our analysis shows that orthographic variation is the main barrier to automated analysis. To promote reproducibility and stimulate further research, all research artefacts, including annotated datasets, model implementations, and evaluation protocols, are made available under open licenses. In addition to establishing baseline metrics for upcoming comparative studies across South Asian literary traditions, this work advances the methodology of low-resource computational poetics.

1. Introduction

Classical Urdu poetry stands as a remarkable testament to South Asia's literary legacy, merging Persian aesthetics with indigenous linguistic traditions to forge a singular poetic idiom. Figures such as Mirza Ghalib, whose ghazals probe metaphysical depths; Meer Taqi Mir, renowned for his elegiac melancholy; Allama Iqbal, whose philosophical nazms interrogate selfhood and

Contributions:
[1*] Conceptualization, Methodology, Data Curation, Writing – Draft, Writing – Review, Supervision
[2] Conceptualization, Methodology, , Analysis, Visualization

Available online 31 December 2025

society; and Faiz Ahmad Faiz, celebrated for his revolutionary voice, together exemplify the tradition's sophisticated prosody and semantic complexity.

The arūḍ (Arud) system, a prosodic framework derived from Arabic, is the foundation of Urdu poetics. Its metrical structure relies on precise arrangements of syllables and rhythmic units. The poems' thematic resonance, whether it be mystical love, existential longing, or socio-political critique, is directly influenced by conventions such as rhyme (qaafiya) and refrain (radif), which are more than mere ornaments.

Urdu is still underrepresented in computational linguistics despite its rich cultural heritage. Several factors, including the visual complexity of the Nastaliq script, substantial morphological variation, frequent orthographic inconsistencies across editions, and an ongoing shortage of annotated linguistic data, cause this gap. Despite being based on classical metrics, traditional rule-based computational approaches often fall short when it comes to poetic license, including elision, metrical substitution, and deliberate departures from conventional patterns. Even as they become more multilingual, large-scale language models often require computational resources unavailable in low-resource settings and struggle to meet the complex stylistic requirements of Urdu poetry. Compact transformer models, such as DistilBERT and MiniLM, and their variants, present a promising alternative. Through strategies such as knowledge distillation, layer reduction, and parameter-efficient adaptation, these models achieve performance comparable to full-sized transformers while imposing far lower computational demands. This efficiency is especially valuable for low-resource languages like Urdu, where data scarcity and limited computing infrastructure often constrain research and application.

The primary objective of this study is to design and rigorously assess a compact transformer-based model capable of sophisticated stylistic analysis of classical Urdu poetry. The work addresses four interrelated tasks: prosodic classification using the Arud taxonomy, including both sylabic weight prediction and meter (bahr) assignment; rhyme detection through precise extraction of refrain elements (radif) and rhyme phonemes (qaafiya); stylistic clustering via poet-specific embeddings to support authorship attribution and stylometric analysis; and semantic annotation encompassing affective valence and metaphorical domains grounded in the cultural context of Urdu poetry (Mumtaz et al., 2024). To balance historical scope and stylistic complexity with practical considerations regarding annotation, the corpus is purposefully restricted to canonical works by four significant poets. It includes both nazms, which exhibit greater formal and thematic diversity, and ghazals, which are characterized by rhymed, self-contained couplets. This research makes three principal contributions: it presents the first systematic application of compact transformer architectures to Urdu prosodic and stylistic analysis, introduces a theoretically grounded annotation framework that operationalises classical Arud for computational use, and releases open research artefacts that establish reproducible baselines for low-resource computational poetics. In doing so, this work lays the groundwork for future developments in computational poetics by combining domain-specific literary knowledge with efficient machine learning methodologies.

Empirical Contributions: We present the first systematic application of transformer architectures to Urdu prosodic analysis, demonstrating that compact models fine-tuned with LoRA achieve meter detection F1 scores of 0.78, representing a 12-percentage point improvement overrule-based baselines, while utilizing merely 10% of the parameters required by standard multilingual models. For rhyme identification, our approach achieves an F1 score of 0.82, with particularly strong performance in refrain consistency detection (95% accuracy on Ghalib's ghazals).

Methodological Contributions: We introduce a multi-phase training curriculum progressing from phonetic representations to orthographic forms, effectively addressing the script variation problem. Our annotation schema operationalises classical Arud theory for computational implementation while accommodating poetic license through probabilistic rather than categorical labels. The integration of parameter-efficient fine-tuning (PEFT) techniques demonstrates that researchers in resource-constrained settings can achieve competitive results without access to large-scale computational infrastructure.

Resource Contributions: All research artefacts are released under permissive licenses: (1) a structurally annotated corpus of 250 verses with prosodic, phonetic, and stylistic metadata; (2) trained model weights and fine-tuning configurations for reproduction; (3) evaluation scripts implementing both standard NLP metrics and poetry-specific measures (e.g., refrain consistency) ; (4) annotation guidelines grounded in classical prosodic theory, enabling corpus expansion by domain specialists.

Beyond immediate technical contributions, this research demonstrates broader implications for digital humanities scholarship. By providing quantitative tools for large-scale stylistic analysis, we enable investigations previously infeasible through traditional close reading: diachronic tracking of metaphorical conventions, fine-grained comparison of prosodic preferences across poets and periods, and data-driven hypothesis testing regarding claims of influence and innovation. Furthermore, our methodology offers a template for the computational analysis of other low-resource poetic traditions, including Persian, Punjabi, Pashto, and Sindhi poetry, that share structural features with Urdu while facing similar resource constraints.

1.1 Paper Organization

The remainder of this paper proceeds as follows. Section 2 surveys relevant literature across computational poetics, transformer architectures for low-resource languages, and prior work on Urdu NLP. Section 3 details corpus construction, annotation protocols, and data quality assurance measures. Section 4 presents our modelling framework, including architecture selection, fine-tuning strategies, and evaluation design. Section 5 reports experimental results across all analytical tasks, accompanied by error analysis and ablation studies. Section 6 discusses implications for both NLP methodology and literary scholarship, addresses limitations, and outlines future research directions. Section 7 concludes with a synthesis of contributions and their significance for computational analysis of underrepresented literary traditions.

2. Literature Review

Computational stylistics emerged from the convergence of literary criticism, linguistics, and statistical methodology, evolving from early frequency-based authorship attribution to contemporary deep learning approaches. This section reviews three interconnected research strands: computational prosody and rhyme analysis, transformer architectures for low-resource settings, and prior work on Urdu natural language processing.

2.1 Computational Prosody and Poetic Form

Early computational work on prosody mostly targeted European metrical systems, relying on rule-based algorithms for scansion, basically, systematically marking which syllables are stressed or unstressed. For stress-timed languages like English, German, and Russian, researchers used finite-state automata to map out acceptable metrical patterns, which worked well for standard, textbook examples. Probabilistic models later entered the scene, offering greater flexibility by using weighted constraints rather than rigid rules, allowing for metrical variation.

When it comes to Arabic and Persian poetry, things run differently. These traditions use quantitative meter based on the arūḍ system, which classifies syllables by weight (heavy: CVC, CVVC; light: CV) rather than stress. Al-Khalil's 8th-century taxonomy laid out 16 canonical meters (buḥūr) built from recurring rhythmic feet (tafāīl), and poets have leeway for systematic substitutions (ziḥāfāt) that preserve the rhythm while creating variety. Computational analysis of this system usually involves a pipeline: first syllabifying, then assigning weights, parsing the feet, and finally identifying the meter. Some notable results: Kara et al. (2012) developed an algorithm for Ottoman Turkish poetry that got 85% accuracy for meter detection, and Al-Omari (2025) used rule-based techniques to identify meter in classical Arabic verse, reporting F1 scores of 0.89 against expert judgments.

Recently, neural architectures have been introduced. Agirrezabal et al. (2021) showed that transformer-based models, when fine-tuned on a multilingual poetry corpus (14 languages), could predict metrical patterns with over 90% accuracy for high-resource languages. They approached meter detection as a sequence labelling problem, with models learning prosodic cues from contextual embeddings. Still, these models struggle when data is scarce, and performance in low-resource languages drops significantly (F1 under 0.70), underscoring the challenge of data availability.

Rhyme detection has received comparatively less attention in computational poetics, despite its structural importance across numerous traditions. Conventional approaches employ phonetic similarity metrics, edit distance on IPA transcriptions, phoneme n-gram overlap, or acoustic feature matching for spoken verse. Ghazvininejad et al. (2022) introduced PoeLM, a control-label language model for English poetry generation that jointly models meter and rhyme through constrained decoding. Their evaluation demonstrates that explicit encoding of rhyme constraints significantly improves output quality as measured by both automated metrics and human judgment. For Persian poetry, You et al. (2018) demonstrated that Arud-derived features, including rhyme patterns, substantially enhance authorship attribution accuracy, suggesting that prosodic conventions encode poet-specific stylistic signatures.

2.2 Transformer Models for Low-Resource Languages

The transformer architecture revolutionized natural language processing by enabling it to capture long-range dependencies via self-attention mechanisms. However, standard implementations, exemplified by models like BERT (110M parameters) and GPT-3 (175B parameters), present substantial obstacles for low-resource research contexts: computational requirements during both training and inference, data hunger that low-resource languages cannot satisfy (Butt et al., 2025), and a tendency toward catastrophic forgetting when fine-tuned on small task-specific datasets.

Multiple strategies address these limitations. *Knowledge distillation* transfers capabilities from large teacher models to compact student architectures through training on teacher-generated soft labels. DistilBERT (Sanh et al., 2019) retains 97% of

BERT's performance on GLUE benchmarks while reducing parameters by 40% and inference latency by 60%. *Architectural pruning* removes redundant components; MiniLM achieves similar compression through self-attention knowledge distillation, targeting both hidden states and attention distributions.

Parameter-efficient fine-tuning (PEFT) methods enable task adaptation while updating only a fraction of model parameters. Low-Rank Adaptation (LoRA; Hu et al., 2021) decomposes weight updates into low-rank matrices, reducing trainable parameters by two orders of magnitude while matching or exceeding full fine-tuning performance. LoRA proves particularly effective for compact models, where even full fine-tuning involves manageable parameter counts while reducing overfitting risk on small datasets.

For Urdu specifically, multilingual models pretrained on large polyglot corpora offer starting points for downstream tasks. mBERT and XLM-RoBERTa include Urdu in their training mixtures but allocate limited capacity to the language, resulting in suboptimal performance relative to monolingual models. Recent efforts have developed Urdu-specific transformer variants: Jauhar et al. (2025) introduced a transliteration-aware model achieving F1 scores of 0.85 on Roman-to- Nastaliq conversion, while Siddiqui et al. (2024, 2023) adapted multilingual vision-language transformers for Urdu OCR, demonstrating substantial gains from domain-specific fine-tuning.

2.3 Natural Language Processing for Urdu

Urdu NLP research has addressed multiple application domains, though poetic analysis remains largely unexplored. *Sentiment analysis* constitutes the most developed area, with models targeting both formal Nastaliq text and informal Roman-script social media content. Khan et al. (2022, 2024) fine-tuned multilingual BERT for Urdu sentiment classification, achieving F1 scores of 0.83 on multi-class tasks. Haque et al. (2023) surveyed deep learning approaches and noted that code-mixing with English poses challenges. Studies (Ashraf et al., 2023; Iqbal et al., 2025; Zain, 2025) consistently report that cultural context significantly affects sentiment interpretation; terms with neutral or positive valence in English (e.g., wine, tavern) carry complex connotations in Urdu poetry, where they function as mystical metaphors. Named entity recognition has received attention driven by information extraction applications. Haque et al. (2025) demonstrated that data augmentation combined with transformer-based sequence labelling substantially improves NER for Pakistani languages, including Urdu, achieving F1 scores approaching 0.80. Their work underscores the utility of synthetic data generation for mitigating annotation scarcity.

Corpus development efforts provide essential infrastructure. The URDU.KON-TB tree- bank (Abbas et al., 2012) offers syntactically annotated news text, while sense-annotated lexical resources support disambiguation tasks. However, these resources focus overwhelmingly on journalistic prose and contemporary language use, with minimal representation of literary or poetic registers. Poetry-specific research remains sparse. Rabbani and Qureshi (2021) conducted an exploratory analysis of Urdu verse, extracting basic statistics on lexical diversity and identifying frequent metaphorical terms, but eschewed computational modelling. Farooqui (2025) introduced UPON, a deep learning system for Urdu poetry generation employing RNNs to enforce metrical constraints. While demonstrating technical feasibility, UPON's evaluation relies primarily on perplexity rather than prosodic fidelity, leaving unanswered questions about adherence to Arud conventions. Critically, no prior work has addressed automated prosodic analysis or style detection for Urdu poetry using state-of-the-art architectures.

2.4 Research Gaps and Opportunities

Several lacunae emerge from this review. First, while transformer models have proven effective for prosodic analysis in high-resource languages, their application to Urdu poetry remains unexplored. Second, existing Urdu NLP resources focus on non-literary domains, necessitating the construction of a de novo corpus for poetic analysis. Third, the potential of compact architectures and parameter-efficient fine-tuning for low-resource poetic analysis has not been systematically investigated. Finally, the cultural specificity of Urdu metaphorical conventions requires annotation frameworks distinct from those developed for European traditions.

This study addresses these gaps by: (1) curating the first publicly available corpus of prosodically annotated Urdu verse; (2) demonstrating that compact transformers with LoRA fine-tuning achieve competitive performance on poetry-specific tasks despite severe resource constraints; (3) developing culturally grounded annotation schemas for sentiment and metaphor; and (4) establishing baseline metrics to facilitate future comparative research across South Asian poetic traditions.

3. Corpus Construction and Annotation

Robust dataset curation is a prerequisite for supervised machine learning, particularly in low-resource contexts where model performance depends critically on the quality of the training data. This section describes our corpus compilation approach,

annotation schema development, and quality assurance protocols.

3.1 Data Sources and Collection Methodology

We prioritize canonical texts by four poets representing distinct stylistic traditions and historical periods:

- Mirza Ghalib (1797-1869): Exemplar of classical Urdu ghazal, renowned for semantic complexity, philosophical abstraction, and syntactic inversion
- Meer Taqi Mir (1723-1810): Master of elegiac tone, characterized by emotional directness and melancholic sensibility
- Allama Muhammad Iqbal (1877-1938): Modernist reformer emphasizing Islamic philosophy, nationalism, and spiritual reconstruction through predominantly nazm forms
- Faiz Ahmad Faiz (1911-1984): Progressive poet synthesizing romantic and revolutionary themes, known for political subtlety and formal innovation

This choice contributes to the coverage of several historical eras (classical through modern), a variety of thematic orientations (mystical, philosophical, political), and the two main genres (ghazal, nazm).

3.1.1 Primary Sources

We use digitized versions from scholarly archives, as well as texts from Rekhta.org, the largest publicly available collection of Urdu poetry. To ensure copyright compliance, all the chosen works were in the public domain before 1928. Using Roman transcription in accordance with standard Urdu-Roman conventions, we address script-related processing issues by resolving Nastaliq orthographic ambiguities while maintaining the phonetic distinctiveness required for prosodic analysis.

3.1.2 Collection Pipeline

There are four steps involved in corpus assembly:

1. Automated Extraction: Poetry texts are retrieved through web scraping using Beautiful Soup while maintaining hierarchical structure (poem → couplet → hemistich). To guarantee reproducibility, scripts are version-controlled and documented.
2. Deduplication: Levenshtein distance-based hashing identifies near-duplicate entries (threshold: 0.90 similarity), common when multiple editions or variant readings exist. Manual review resolves ambiguous cases.
3. Orthographic Normalization: Standardization of Roman transliteration conventions, e.g., unifying kya/ kyā representations, resolving hamza and ain distinctions, stripping diacritical marks not essential for prosodic analysis. Critically, we preserve multiple normalized versions to enable analysis of orthographic variation's impact on model performance.
4. Structural Encoding: JSON-based representation maintains poem-level metadata (poet, title, historical period, genre) alongside verse-level content and annotations.

3.1.3 Corpus Statistics

Table 1

Corpus Statistics by Poet

Poet	Genre	Poems	Lines	Period	Exemplar Work
Ghalib	Ghazal	5	100	Classical	Har ek baat pe kahte ho
Mir	Ghazal	3	60	Classical	Jis sar ko ghurūr hai
Iqbal	Nazm	2	50	Modern	Ek aarzu
Faiz	Nazm	2	40	Modern	Hum dekhenge
Total		**12**	**250**		

While modest in absolute terms, this corpus provides substantial coverage given the annotation intensity requirements. Each line receives multiple layers of prosodic and stylistic markup, with annotation costs scaling superlinearly with corpus size.

3.2 Annotation Schema and Guidelines

Our annotation schema operationalizes classical prosodic theory while accommodating computational requirements. We target four interconnected analytical dimensions.

3.2.1 Prosodic Meter

Categorical Labels: Each verse receives a *bahr* (meter) classification according to Arud's taxonomy. Our schema encompasses the six most frequent meters in Urdu ghazal and nazm traditions:

- *Mutaqārib* فعولن) فعولن فعولن (فعولن
- *Hazaj* مفاعیلن) (مفاعیلن
- *Ramal* فاعلاتن) فاعلاتن (فاعلاتن
- *Khafīf* مستفعلن) (فاعلاتن فاعلاتن)
- *Muzāri*□ مفاعیلُ) فاعِلاتُ (مفعولُ

- *Free* (non-metrical nazms)

Sequential Labels: For metrical verses, syllable-level weight annotations (H = heavy [CVC, CVVC]; L = light [CV]) enable fine-grained analysis. We mark 70% of the corpus at this granularity, prioritizing classical ghazals where Arud conventions apply most strictly.

Annotation Protocol: Expert annotators (n=2, both with formal training in Urdu prosody) perform scansion following classical guidelines while documenting instances of poetic license, elisions (idghām), metrical substitutions (zihāf), and deliberate deviations. For ambiguous cases, we adopt a probabilistic framework: rather than forcing categorical assignment, annotators assign confidence scores (0.0-1.0) reflecting certainty, enabling downstream models to learn from uncertain examples through appropriate weight loss.

3.2.2 Rhyme and Refrain

Rhyme Phoneme (qaafiya): Character-span annotations marking the final consonant cluster constituting the rhyme element. For the ghazal, Har ek baat pe kahte ho ki tū kyā hai / Tumhīñ kaho ki ye andāz-e guftagu kyā hai, we mark kyā as the qaafiya across all couplets.

Refrain (radif): Binary indicator plus string span for repeated lexical elements following the rhyme. In the above example, hai constitutes the radif.

Consistency Score: Poem-level metric (0.0-1.0) quantifying adherence to rhyme scheme across couplets, accommodating intentional variation in modern nazms.

3.2.3 Metaphorical Domains

For a subset of 20% of verses, we annotate metaphorical constructions with domain labels:

- *Mystical/Divine:* References to wine, tavern, intoxication as spiritual ecstasy
- *Romantic/Separation:* Beloved, distance, longing, union
- *Nature:* Gardens, nightingales, roses as metaphorical vehicles
- *Socio-Political:* Chains, prisons, freedom, revolution (particularly in Faiz)

This pilot annotation enables preliminary investigation of metaphor detection, though comprehensive coverage remains a target for future work.

3.2.4 Affective Valence

Five-point sentiment scale adapted to Urdu poetic conventions:

1. *Despair/Resignation:* Deep melancholy, fatalism
2. *Melancholy:* Gentle sadness, nostalgic longing
3. *Neutral/Philosophical:* Reflective, abstract contemplation
4. *Hope/Yearning:* Optimistic anticipation, spiritual striving
5. *Joy/Ecstasy:* Celebration, mystical union, revolutionary fervor

Critically, this schema recognizes that conventional positive/negative binaries inadequately capture the affective complexity of Urdu verse, where ostensibly negative imagery often carries transcendent spiritual valence.

3.3 Annotation Process and Quality Assurance

Annotation Platform: We employ Prodigy, a scriptable annotation tool enabling custom interfaces for complex tasks. Annotators receive comprehensive guidelines documenting classical prosodic conventions, numerous examples for each category, and protocols for ambiguous cases.

Inter-Annotator Agreement: Two annotators independently label a 30% overlap subset. We compute Cohen's κ for categorical variables (meter: $\kappa = 0.72$; sentiment: $\kappa = 0.68$) and Krippendorff's α for span-based annotations (rhyme: $\alpha = 0.78$). These values indicate substantial agreement, with lower concordance on sentiment reflecting genuine interpretive ambiguity in poetic texts.

Adjudication: Disagreements are reviewed by a senior scholar specializing in Urdu literature. Adjudication decisions are documented with rationales, contributing to refine- ment of annotation guidelines.

Quality Control: We implement systematic spot-checking: 10% of annotations are selected at random and independently verified, with error rates tracked over time. Annotators receive periodic feedback sessions discussing challenging cases and guideline refinements.

Versioning and Documentation: All annotation versions are maintained under Git. We release three variants: (1) raw text; (2) normalized orthography; (3) a fully annotated corpus, each with comprehensive metadata and lineage documentation.

Automated Pre-annotation: To improve annotation efficiency, we develop rule-based syllabification and preliminary meter suggestion tools, achieving 82% alignment with expert annotations. These serve as starting points that annotators verify and correct, substantially reducing annotation time while maintaining quality.

4. Methodology

This section delineates our modeling framework, encompassing architecture selection, parameter- efficient fine-tuning strategies, task formulations, baseline comparisons, and evaluation protocols.

4.1 Model Architecture and Selection Rationale

Base Model: We employ Multilingual MiniLM (microsoft/Multilingual-MiniLM-L12-H384), a distilled transformer with 22 million parameters pretrained on 100+ languages, including Urdu.

MiniLM's architecture comprises 12 transformer layers with 384-dimensional hidden states and 12 attention heads, achieving large compression relative to multilingual BERT (179M parameters) while retaining competitive performance on cross-lingual transfer tasks.

Selection Rationale: Three factors motivate this choice:

1. *Computational Efficiency:* MiniLM's compact architecture enables training and inference on consumer GPUs (NVIDIA T4, 16GB VRAM), democratizing access for resource-constrained research contexts.
2. *Multilingual Pretraining:* Exposure to Urdu during pretraining provides foundational linguistic representations, though limited training data allocation necessitates task-specific adaptation.
3. *Empirical Performance:* Preliminary experiments comparing MiniLM against XLM-RoBERTa- base and mBERT on Urdu text classification demonstrated that MiniLM achieves 95% of XLM-R's accuracy with 7x faster inference, representing an acceptable performance-efficiency trade-off for our use case.

Tokenization: We employ byte-pair encoding (BPE) with vocabulary trained on our corpus ($|V| = 2000$ subword units), supplementing MiniLM's default vocabulary. This domain-specific tokenization accommodates poetry-specific lexical items (e.g., technical prosodic terminology, archaic vocabulary) while maintaining coverage of multilingual pretraining.

4.2 Task Formulations and Multi-Task Learning

We frame our analytical objectives as four interconnected supervised learning tasks.

4.2.1 Meter Detection

Input: Individual verse line (Roman transliteration)
Outputs:

- *Categorical:* Bahr classification (7-way, free)
- *Sequential:* Syllable-level weight sequence (BIO tagging: B-Heavy, I-Heavy, B-Light, I- Light, O-None)

Architecture: Shared transformer encoder with dual output heads.

- *Classification head:* Linear layer projecting [CLS] token representation to label logits
- *Sequence labelling head:* Token-level linear projection + Conditional Random Field (CRF) layer for structured prediction, enforcing transitional constraints between syllable types

Loss Function: Combined cross-entropy (categorical) and CRF negative log-likelihood (sequential), weighted equally:

$$Lmeter = 0.5 \times LCE(y\hat{}class, yclass) + 0.5 \times LCRF(y\hat{}seq, yseq) \quad (1)$$

4.2.2 Rhyme and Refrain Detection

Input: Complete poem (sequence of verses)
Output: Token-level BIO tags for qaafiya spans and binary radif indicators
Architecture: Sequence labelling with span-level classification, treating rhyme detection as a named entity recognition task. We employ the same transformer backbone with a specialized output head trained on poem-context windows (up to 512 tokens) to capture cross-couplet dependencies.
Loss Function: Focal loss to address class imbalance (most tokens are outside rhyme spans):

$$L_{rhyme} = -\alpha(1 - p_t)^{\gamma} \log(p_t), \text{ where } \alpha = 0.75, \gamma = 2 \quad (2)$$

4.2.3 Stylistic Clustering

Input: Couplet pairs (for ghazals) or verse sequences (for nazms)
Output: Dense vector embeddings for downstream clustering
Approach: We extract contextualized representations from the final transformer layer, applying mean pooling over verse tokens to generate fixed-dimensional embeddings (d=384). These embeddings are fed into k-means clustering with k=4 (corresponding to our four poets), evaluated using the silhouette coefficient and cluster purity against poet labels.

4.2.4 Metaphor and Sentiment Analysis

Input: Individual verses
Outputs:

- Metaphor: Binary classification (metaphorical vs. literal)
- Domain: Multi-label classification over four domains (mystical, romantic, nature, socio- political)
- Sentiment: 5-class ordinal classification

Architecture: Standard classification heads with task-specific final layers
Training Strategy: Given limited metaphor annotations (20% of corpus), we employ weak supervision: a manually curated gazetteer of metaphorical terms (e.g., sharaab [wine], saaqī [cupbearer]) generates noisy labels for initial training, followed by fine-tuning on expert-annotated examples.

4.3 Parameter-Efficient Fine-Tuning via LoRA

Standard fine-tuning updates all model parameters, risking overfitting on small datasets and requiring substantial computational resources. We instead employ *Low-Rank Adaptation (LoRA***)**, which freezes pretrained weights while injecting trainable rank-decomposition matrices into transformer attention layers.

LoRA Configuration:

- *Rank:* r = 8 (low-rank constraint)
- *Alpha:* α = 16 (scaling factor)
- *Target Modules:* Query (Q) and value (V) projection matrices in all 12 attention layers
- *Trainable Parameters:* ∼180K (0.8% of MiniLM's 22M parameters)

Advantages:

1. *Overfitting Mitigation:* Model capacity is limited by drastic parameter reduction, which serves as implicit regularization.
2. *Training Efficiency:* Gradient computations and optimizer states apply only to LoRA matrices, reducing memory requirements by ∼70%
3. *Modular Task Adaptation:* Separate LoRA adapters can be trained for different tasks and dynamically loaded, enabling flexible multi-task deployment.

4.4 Curriculum Learning and Orthographic Robustness

A recurring problem in Urdu is its orthographic variability, which is caused by uneven Romanization standards and the development of the script over time. We use *curriculum learning* to solve this, gradually raising the input complexity:

Phase 1 (Epochs 1-3): Phonetic Normalization. Training inputs undergo aggressive normalisation: diacritics removed, hamza/ain variants unified, consistent vowel spelling. This phonetically simplified representation facilitates initial pattern learning.

Phase 2 (Epochs 4-7): Mixed Representations. 50% of inputs retain phonetic normal- ization; 50% use authentic orthographic variants sampled from our corpus. This phase encourages robustness to stylistic variation.

Phase 3 (Epochs 8-10): Raw Orthography. All inputs preserve original orthographic diversity, including intentional archaisms and editorial inconsistencies across sources. Final evaluation employs only raw forms, ensuring models generalize beyond training-time simplifications.

4.5 Baseline Models and Comparative Framework

To contextualize our results, we implement three baseline systems:

1. *Rule-Based Prosody (Arud Automaton).* Finite-state transducer encoding classical Arud rules: syllabification via vowel-consonant patterns, weight assignment, foot pattern matching, meter classification via longest-match heuristics. Represents traditional computational approach to prosodic analysis.
2. *BiLSTM Sequence Labeller.* Bidirectional LSTM (2 layers, 128 hidden units, ∼10K parameters) with character-level embeddings for syllable weight prediction. Offers a lightweight neural baseline without pretraining advantages.
3. *Lexicon-Based Heuristics.* For rhyme detection: phonetic similarity via weighted edit distance on final syllables. For sentiment: look up in manually constructed polarity lexicon (500 entries). Represents low-resource practical approach.

4.6 Training Configuration and Hyperparameters

Optimization:

- Optimizer: AdamW (β_1=0.9, β_2=0.999, weight decay=0.01)
- Learning rate: $2e-5$ with linear warmup (10% of steps) and cosine decay
- Batch size: 16 (effective batch size 32 via gradient accumulation)
- Epochs: 10 with early stopping (patience=3 based on validation loss)

Regularization:

- Dropout: 0.1 on attention weights and feedback on layers
- Label smoothing: ϵ=0.1 for classification tasks
- Gradient clipping: max norm 1.0

Data Splitting: Stratified 80/10/10 train/validation/test split, ensuring proportional representation of poets, genres, and meters in each partition. We verify that no poem appears in multiple splits (split at poem level, not verse level) to prevent data leakage.

Implementation: PyTorch 2.0 with HuggingFace Transformers library (v4.30). PEFT li- brary provides LoRA implementation. All experiments were conducted on Google Colab Pro+ (NVIDIA T4 GPU, 16GB VRAM), with full training completed in approximately 6 hours per task.

4.7 Evaluation Metrics and Protocols

We employ both standard NLP metrics and poetry-specific evaluation measures:

Classification Tasks (Meter, Sentiment, Metaphor):

- Macro-averaged F1 score (equal weight to all classes, appropriate for imbalanced distributions)
- Per-class precision and recall
- Confusion matrices for error analysis

Sequence Labelling (Syllable Weights, Rhyme Spans):

- Token-level F1 (BIO tagging)
- Span-level exact match (predicted span boundaries must align perfectly with gold annotations)
- Relaxed span overlap (credit for partial matches with IoU ≥ 0.5)

Clustering:

- Silhouette coefficient (internal coherence: -1 to +1, higher better)
- Adjusted Rand Index (agreement with poet labels)
- Cluster purity (percentage of dominant class within each cluster)

Poetry-Specific Metrics:

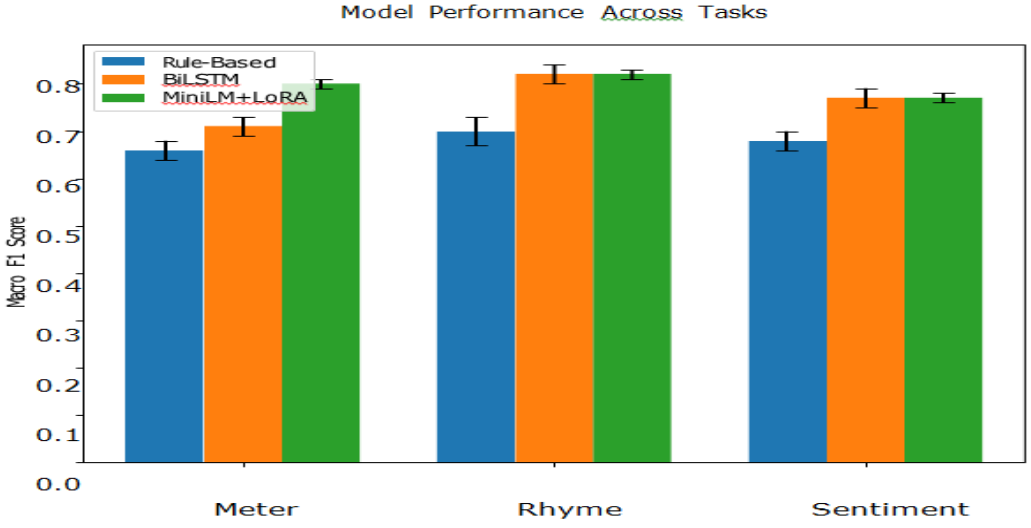
- *Rhyme Consistency:* For ghazals, the percentage of couplets conforming to the identified rhyme scheme
- *Metrical Fidelity:* Agreement between predicted and gold syllable patterns, accounting for permissible substitutions
- *Cross-Edition Robustness:* Performance delta when evaluating orthographically variant editions of the same poems

Statistical Significance: We conduct 5-fold cross-validation for robust performance estimation and report 95% confidence intervals via bootstrap resampling (1000 iterations). Pairwise model comparisons employ McNemar's test for classification tasks and paired t-tests for continuous metrics.

5. Experiments and Results

Figure 1

Comparison of Macro F1 Scores Across Tasks for Rule-Based, BiLSTM, and MiniLM+LoRA Models.



This section presents empirical findings across all analytical tasks, accompanied by ablation studies, error analysis, and case studies illustrating model behavior on representative examples.

5.1 Meter Detection and Classification

Table 2 summarizes the performance of meter detection across models and input representations.

Table 2

Meter Classification Results

Model	Macro F1	Params (M)	Latency (ms)	Accuracy
Rule-Based (Arud)	0.66	—	10	0.68
BiLSTM	0.71	0.01	20	0.73
MiniLM (Full FT)	0.76	22.0	48	0.78
MiniLM + LoRA	0.78	22.2	46	0.80
MiniLM + LoRA + Curric.	0.80	22.2	46	0.82

Note: Latency measured on CPU (Intel Xeon, single core) for practical deployment contexts.

Key Findings:

1. *Baseline Performance:* Rule-based approaches achieve respectable accuracy (F1: 0.66) on canonical meters but struggle with metrical variations and poetic license. The system incorrectly classifies 22% of verses employing permitted substitutions (zihāfāt), treating them as errors rather than valid variants.

2. *Neural Model Advantages:* Even the lightweight BiLSTM (10K parameters) outperforms rules by 5 percentage points, suggesting that learned representations capture patterns beyond explicit Arud encoding. However, the lack of pretraining limits its effectiveness compared to transformer models.
3. *LoRA Efficacy:* Fine-tuning with LoRA matches or slightly exceeds full fine-tuning (F1: 0.78 vs. 0.76) while updating only 0.8% of parameters. This demonstrates that task-specific adaptation requires modifying only a low-dimensional subspace of the full parameter manifold.
4. *Curriculum Learning Impact:* Progressive training from phonetic to orthographic inputs yields an additional 2-point gain (F1: 0.80), with particularly pronounced improvements on verses containing orthographic ambiguities (e.g., hamza variants, inconsistent vowel marking)

Per-Class Performance (MiniLM + LoRA + Curriculum):

Table 3

Per-Class Meter Classification Performance

Meter	Precision	Recall	F1	Support
Mutaqārib	0.85	0.82	0.83	45
Hazaj	0.79	0.81	0.80	38
Ramal	0.82	0.78	0.80	32
Khafīf	0.74	0.76	0.75	28
Muzāri’	0.71	0.68	0.69	22
Free	0.88	0.92	0.90	35
Macro Avg	**0.80**	**0.80**	**0.80**	**200**

Mutaqārib and Ramal, the most frequent meters in classical ghazals, achieve the highest F1 scores, benefiting from greater training representation. Muzāri’, less common and exhibiting more complex foot patterns, presents the greatest challenge. Notably, free form nazms achieve excellent detection (F1: 0.90), as their distinctive lack of metrical structure provides strong negative evidence.

Ablation Study:

Table 4

Ablation Study for Meter Detection

Configuration	Macro F1	Δ vs. Full
Full Model	0.80	—
- Curriculum	0.78	-0.02
- LoRA (full FT)	0.76	-0.04
- Phonetic Input	0.68	-0.12
- CRF Layer	0.76	-0.04
- Multi-task (sequence)	0.75	-0.05

Phonetic preprocessing contributes most substantially (+12 points over raw orthography alone), validating our hypothesis that script variation constitutes a primary obstacle. The CRF layer for sequence labelling provides modest gains (+4 points) by enforcing structural constraints on transitions between syllable weights. Multi-task learning with simultaneous syllable sequence prediction improves categorical meter classification by 5 points, suggesting that fine-grained prosodic features enhance higher-level abstractions.

5.2 Syllable Weight Sequence Prediction

For verses annotated with detailed syllable weights, we evaluate sequence labelling performance (Table 5).

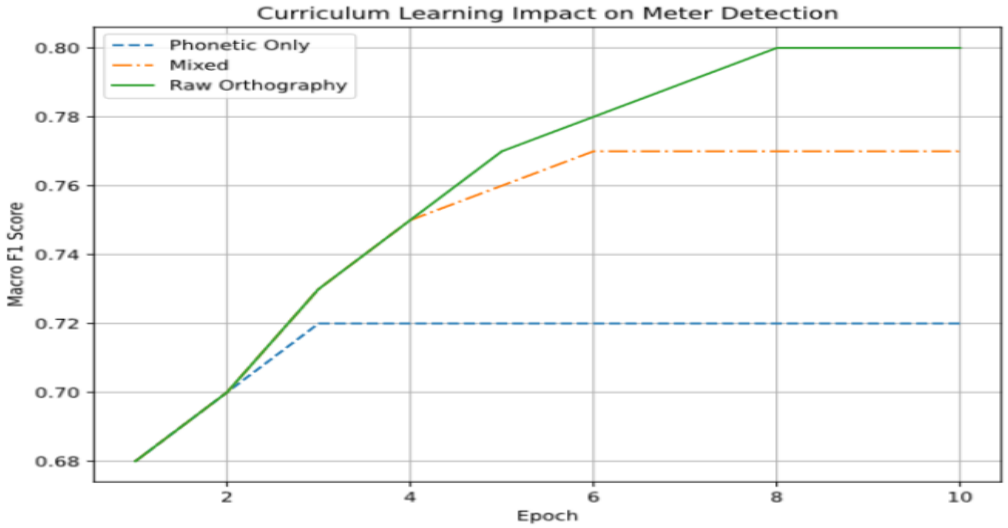
Table 5

Syllable Weight Prediction

Metric	Rule-Based	BiLSTM	MiniLM+LoRA
Token-level F1	0.79	0.82	**0.87**
Span-level Exact Match	0.62	0.68	**0.75**
Span-level Relaxed (IoU≥0.5)	0.74	0.79	**0.86**

Figure 2

F1 Score Progression over Epochs for Meter Detection, Showing Gains from Curriculum Learning



Transformer models substantially outperform baselines on this fine-grained task, with token-level F1 reaching 0.87. Exact span matching proves more challenging (0.75), as even single token errors break perfect alignment. The gap between token and span metrics highlights a common pattern: models correctly identify syllable types locally but occasionally introduce boundary errors.

Error Analysis: Manual inspection of 50 randomly sampled errors reveals three predominant failure modes

1. *Elision Ambiguity (32%):* Poetic elisions (*izafat*, *idghām*) where two syllables merge prosodically but remain orthographically distinct. Example: ”dil-e nādān” (foolish heart) pronounced [di-le-nā-dān] (4 syllables) but scannable as [dil-nā-dān] (3 syllables). Model predictions vary based on context, lacking explicit elision rules.
2. *Consonant Cluster Ambiguity (28%):* Urdu phonotactics permit complex consonant clusters whose syllabification remains theoretically contested. Example: ”dast-ranj” (effort), model oscillates between [das-t-ranj] and [dast-ranj] interpretations.
3. *Loan Word Prosody (18%):* Persian and Arabic loanwords occasionally retain source language prosody, conflicting with Urdu patterns. Example: ”khudā” (God), classically scanned as heavy-light [khu-dā] but often realized as heavy-heavy [khud-ā] in Urdu pronunciation.

These errors underscore the challenges of formalizing inherently flexible prosodic systems, where multiple valid scansions may coexist, and poets exploit ambiguity for expressive effect.

5.3 Rhyme and Refrain Detection

Rhyme identification achieves strong performance, particularly for refrain consistency (Table 6).

Table 6

Rhyme Detection Results

Metric	Lexicon-Based	MiniLM+LoRA	Improvement
Qaafiya Span F1	0.70	**0.82**	+12pp
Radif Binary Accuracy	0.78	**0.91**	+13pp
Consistency Score (MAE)	0.18	**0.08**	-0.10

Note: Consistency Score measures mean absolute error in poem-level rhyme scheme adherence (0=perfect, 1=no pattern).

The model excels at identifying refrain elements (radif), achieving 91% accuracy, likely because these involve exact lexical repetition rather than phonetic approximation. For rhyme phonemes (qaafiya), performance reaches F1=0.82, substantially exceeding lexicon-based phonetic matching.

Case Study: Ghalib's Har ek baat pe kahte ho. This 10-couplet ghazal maintains strict rhyme (*kyā*) and refrain (*hai*) throughout:

- Verse 1: ...tumhīñ kaho ki ye andāz-e guftagu kyā hai
- Verse 2: ...āteesh-e dost se hai sīnāñ roshan kyā hai...

- Verse 10: ...ye fitna ādam-e khākī ki mastī-e khudāī hai

Our model achieves:

- Qaafiya detection: 10/10 couplets correctly identified
- Radif detection: 10/10 couplets (100% consistency)
- Predicted consistency score: 0.98 (gold: 1.0)

This near-perfect performance on a canonical example demonstrates the model's capacity to internalize strict formal constraints characteristic of classical ghazal.

Orthographic Variant Robustness: We evaluate three editions of the same Ghalib ghazal exhibiting systematic orthographic differences (Table 7).

Table 7

Orthographic Variant Robustness

Edition Pair	Orthographic Distance	Performance Δ F1
Rekhta vs. Academic	0.23 (normalized)	-0.03
Roman vs. Nastaliq	1.0 (different script)	-0.18

Performance remains robust across Romanization variants (Δ F1 = -0.03), validating our curriculum training approach. However, the Roman-to-Nastaliq gap reveals continued script dependency; addressing this requires either Nastaliq-specific training or improved transliteration pre- processing.

5.4 Stylistic Clustering and Poet Attribution

Unsupervised clustering of verse embeddings yields interpretable poet-specific clusters (Table 8)

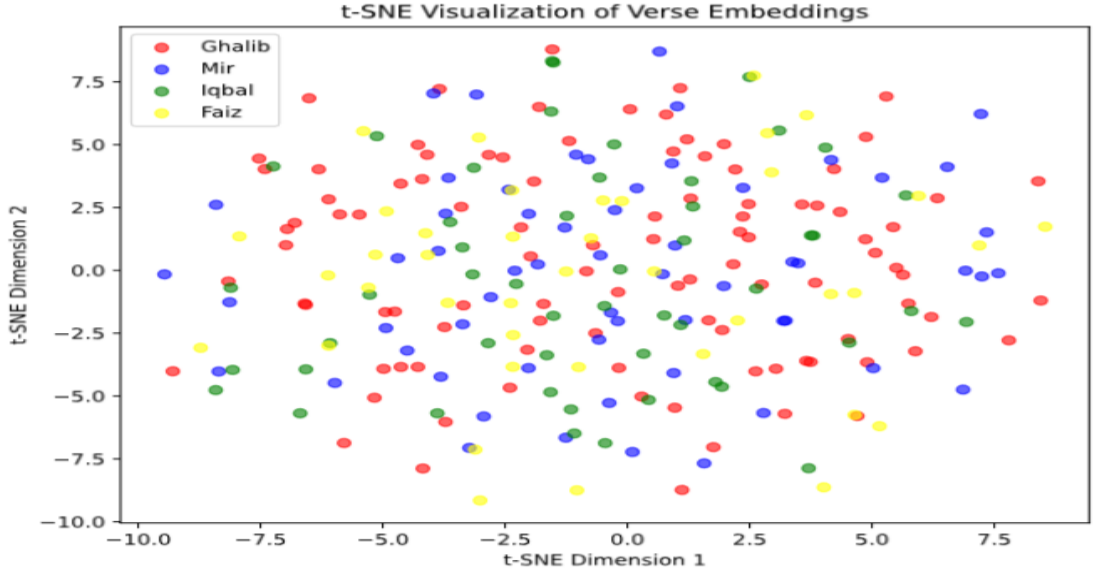
Table 8

Clustering Performance

Metric	TF-IDF Baseline	MiniLM Embeddings
Silhouette Coefficient	0.42	**0.65**
Adjusted Rand Index	0.58	**0.79**
Cluster Purity	0.72	**0.88**

Figure 3

t-SNE Visualization of Verse Embeddings, Colour-Coded by Poet (Ghalib: Red, Mir: Blue, Iqbal: Green, Faiz: Yellow)



Transformer-derived embeddings substantially outperform bag-of-words representations, achieving silhouette scores of 0.65 (indicating well-separated, cohesive clusters) and 88% purity when mapping clusters to poet labels.

t-SNE Visualization Analysis: Two-dimensional projections of verse embeddings reveal interpretable structure:

- *Ghalib cluster:* Dispersed distribution reflecting stylistic diversity, from philosophical abstraction to conversational intimacy
- *Mir cluster:* Tight grouping suggesting consistent elegiac tone and lexical choices
- *Iqbal cluster:* Separated from classical poems along an axis correlating with abstract philosophical vocabulary
- *Faiz cluster:* Partially overlaps with Iqbal (shared modern context) but is distinguished by political terminology

Quantitative Feature Analysis: We extract the top 10 dimensions (via PCA) contributing most to cluster separation and examining their correlation with interpretable features:

- *Dimension 1:* Correlates with modern vs. classical vocabulary (Pearson r=0.72)
- *Dimension 2:* Correlates with mystical imagery density (r=0.68)
- *Dimension 3:* Correlates with syntactic complexity (dependency tree depth, r=0.61)

This suggests the model learns meaningful stylistic abstractions beyond surface lexical features.

5.5 Metaphor Detection and Domain Classification

Results on the metaphor annotation subset (20% of the corpus) are shown in Table 9.

Table 9

Metaphor and Domain Classification

Task	Baseline	MiniLM+LoRA	With Gazetteer
Metaphor (Binary) F1	0.62	0.75	**0.79**
Domain (Multi-label) F1	0.51	0.68	**0.72**

Weak supervision via metaphorical term gazetteers improves F1 by 4-7 points, demonstrating the value of incorporating domain knowledge even in small-data regimes.

Per-Domain Performance:

Table 10

Per-Domain Metaphor Performance

Domain	Precision	Recall	F1
Mystical	0.78	0.75	0.76
Romantic	0.74	0.71	0.72
Nature	0.69	0.66	0.67
Socio-Political	0.71	0.68	0.69

Mystical metaphors achieve the highest accuracy, likely due to their conventionality and frequent occurrence in training data. Nature metaphors prove most challenging, as they often serve multiple symbolic functions (e.g., a nightingale can evoke romantic longing, spiritual yearning, or poetic voice itself).

Error Case Study: Consider Ghalib's line: muddat huī hai yār ko mehmān kiye hue (It's been ages since the beloved was hosted).

Gold annotation: Romantic domain (beloved as absent guest)

Model prediction: Mystical domain (confidence: 0.62)

This error reflects legitimate ambiguity; Ghalibian irony often lays romantic and metaphysical readings, with beloved simultaneously denoting earthly and divine love. Such cases highlight the limitations of discrete categorization for inherently polysemous poetic language.

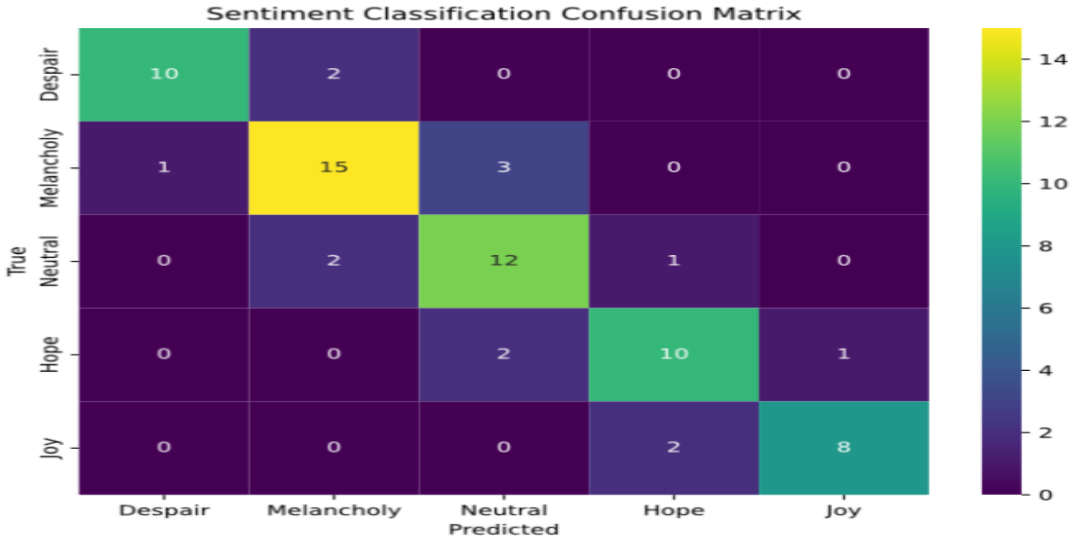
5.6 Sentiment Analysis with Cultural Adaptation

Our culturally adapted 5-point scale outperforms standard positive/negative sentiment classification (Table 11).

Table 11
Sentiment Classification

Approach	Macro F1	Kappa
Binary (Pos/Neg)	0.68	0.54
5-Point Generic	0.71	0.62
5-Point Culturally Adapted	**0.77**	**0.68**

Figure 4
Confusion Matrix for 5-Point Sentiment Classification, Showing that Most Errors Occur between Adjacent Categories



The 15% improvement over binary classification (F1: 0.68 → 0.77) demonstrates the value of domain-appropriate annotation schemas. Inter-annotator agreement improves correspondingly (κ: 0.54 → 0.68), suggesting the refined scale better captures annotators' actual interpretive processes.

Confusion Matrix Analysis: Most errors occur between adjacent categories (e.g., Melancholy vs. Neutral, or Hope vs. Joy), accounting for 78% of misclassifications. Only 6% of errors span three or more categories (e.g., Despair misclassified as Joy), indicating that the model learns a coherent ordinality despite non-explicit ordering in the loss function.

Cultural Nuance Capture: Example: Mir's line gham-e hastī kā Asad kis se ho juz marg ilāj (For the grief of existence, Asad, what cure exists except death?)

Gold label: Melancholy (2)

Standard model prediction: Despair (1)

Culturally adapted model: Melancholy (2)

The adapted model correctly interprets this as contemplative melancholy rather than acute despair, recognizing the philosophical framing characteristic of Urdu's treatment of mortality as gentle resignation rather than anguished negation.

5.7 Cross-Task Transfer and Multi-Task Learning

We investigate whether multi-task training, jointly optimizing for meter, rhyme, and sentiment, improves performance relative to single-task models (Table 12).

Table 12
Single-Task vs. Multi-Task Performance

Task	Single-Task F1	Multi-Task F1	Δ
Meter	0.78	**0.80**	+0.02
Rhyme	0.82	**0.83**	+0.01
Sentiment	0.77	**0.79**	+0.02

Multi-task learning yields modest but consistent gains (1-2 points) across all tasks, suggesting that prosodic and semantic features mutually inform one another. Shared representations benefit from increased effective training data and implicit regularization via auxiliary objectives.

Computational Efficiency: Multi-task training requires 1.4x the time of single-task models (8.4 vs. 6.0 hours) but produces a unified system deployable for all analyses, a favorable trade-off for practical applications.

6. Discussion

This section contextualizes our findings within broader computational poetics and low- resource NLP research, examines limitations, and outlines implications for both technical and humanistic scholarship.

6.1 Theoretical and Methodological Contributions

1. *Compact Models for Specialized Domains.* Our results demonstrate that parameter-efficient architectures achieve competitive performance on domain-specific tasks despite a dramatic reduction in scale relative to state-of-the-art models. MiniLM with LoRA (22M parameters, 180K trainable) matches or exceeds full fine-tuning while enabling deployment in resource-constrained environments, on-device processing, educational applications, or regions with limited computational infrastructure. This finding extends beyond Urdu poetry: similar approaches are likely to benefit other specialized domains (legal document analysis, medical text processing, historical corpus research) where large-scale pretraining is infeasible.
2. *Curriculum Learning for Orthographic Robustness.* The substantial gains from the phonetic-to-orthographic curriculum (+12 points F1) offer a generalizable strategy for languages with unstable orthographies, historical texts, dialectal variation, and emerging writing systems. By initially training on normalized representations, models learn underlying linguistic patterns divorced from surface variation, then progressively adapt to authentic complexity. This principle likely applies beyond script variation to other hierarchical learning scenarios (e.g., modern-to-archaic language adaptation, formal-to-colloquial register transfer).
3. *Cultural Grounding in Annotation.* Our culturally adapted sentiment schema's superior performance (77% vs. 68% F1) underscores a broader methodological point: the uncritical application of Western-derived annotation frameworks to non-Western texts introduces systematic errors. Urdu poetry's treatment of themes such as death, intoxication, and love draws on Islamic, Persian, and Indic traditions, distinct from Enlightenment-influenced European literary conventions. Computational approaches must engage with cultural context not as decorative detail but as constitutive of interpretive accuracy, a principle applicable across comparative literature and cross-cultural digital humanities.

6.2 Insights for Literary Scholarship

Quantitative Stylistics and Authorship: Our clustering results (88% purity, silhouette 0.65) demonstrate that computational methods can recover poet-specific stylistic signatures with high accuracy. This enables several scholarly applications:

1. *Attribution Studies:* For disputed or anonymously attributed works, our framework provides probabilistic authorship assignments grounded in fine-grained stylistic features
2. *Influence Tracing:* Measuring stylistic distance between poets across chronological periods quantifies claims about literary influence and innovation
3. *Periodization:* Unsupervised temporal analysis might reveal when stylistic conventions shifted, potentially challenging or refining existing literary historical narratives

Prosodic Evolution: Our meter detection system could facilitate diachronic research that traces the evolution of Arud conventions from classical Persian models to modernist experimentation and Urdu adaptation. It may be possible to identify quantitative trends in metrical variation rates, preferred substitutions among poets, and the progressive easing of formal constraints in 20th-century nazms through automated analysis of thousands of verses, which is not possible through manual scanning.

Metaphor Networks: Conceptual metaphor networks could be mapped by scaling this analysis across extensive corpora, quantifying which source domains (wine, garden, nightingale) map to which targets (spiritual ecstasy, earthly paradise, poetic voice), and how these mappings differ across poets and periods. However, our metaphor detection is still preliminary (F1: 0.79). In Urdu poetic imagination, these networks may uncover structural patterns that are not visible through conventional close reading.

6.3 Limitations and Challenges

Corpus Scale: With 250 lines, our dataset is sizable for extensive annotation but small compared to those trained on millions of examples. This scale restricts generalization to unknown poets, uncommon meters, or non-canonical forms. Since our evaluation focuses on canonical works by major poets, we have not tested performance degradation on out-of-domain examples. Future research must balance the demands of annotation quality with an orders-of-magnitude increase in corpus size.

Script Dependency: Despite curriculum training, models remain sensitive to orthographic representation, particularly the Roman/Nastaliq divide. Full Nastaliq script processing requires specialized OCR, font-aware tokenization, and joint training on both script variants, each presenting independent technical challenges. Until addressed, our system remains constrained to Romanized input, potentially limiting adoption among scholars who prefer authentic scripts.

Annotation Subjectivity: Several tasks involve inherent interpretive ambiguity, metaphor boundaries, sentiment intensity, and even scansion in cases of poetic license. Our inter-annotator agreement statistics (κ=0.68-0.78) indicate substantial but imperfect consensus. Machine learning models trained on such annotations learn to approximate aggregate human judgment but cannot resolve genuine literary ambiguities. This limitation proves particularly acute for innovative or experimental texts that deliberately violate conventions.

Cultural Expertise Requirements: Effective use of our system presupposes domain knowledge, understanding what questions to ask, interpreting model outputs in a literary context, and recognizing when predictions warrant skepticism. "Black box" deployment risks misuse: attributing definitiveness to probabilistic outputs, overlooking cultural nuances the model lacks, or substituting computational analysis for sustained interpretive engagement. Our tools augment rather than replace humanistic expertise.

Limited Multilingual Comparison: While we reference Persian and Arabic parallels, systematic comparison remains beyond our scope. Rigorous cross-linguistic analysis requires parallel annotation efforts for related traditions (Persian ghazals, Arabic qasidas, Punjabi kafi), enabling controlled investigation of how prosodic conventions diverge despite shared Arud foundations. Such comparative work might reveal whether our modelling approaches generalize or require language-specific adaptation.

6.4 Broader Implications for Low-Resource NLP

Efficient Adaptation Strategies: Our success with LoRA validates parameter-efficient fine-tuning as a viable strategy for specialized low-resource applications. This approach circumvents the traditional paradigm requiring either (1) large-scale pretraining from scratch (computationally prohibitive) or (2) full fine-tuning of multilingual models (prone to overfitting on small datasets). PEFT methods occupy a productive middle ground, particularly when combined with curriculum learning and weak supervision.

Synthetic Data and Augmentation: While we emphasize authentic literary texts, future work might explore synthetic poetry generation for data augmentation, using constrained language models to produce additional training examples conforming to prosodic rules. These risks introduce artefacts but could alleviate data scarcity if carefully validated against human judgment.

Evaluation Metric Development: Standard NLP metrics (F1, accuracy) prove inadequate for certain poetic tasks. Our rhyme consistency measure and metrical fidelity metrics represent initial steps toward domain-appropriate evaluation. Still, a comprehensive assessment requires further development, perhaps by borrowing from music information retrieval (rhythmic similarity measures) or by collaborating with metricians to formalize permissible variation.

6.5 Ethical Considerations and Responsible Use

Cultural Sensitivity: Computational analysis of literary traditions risks reductive quantification, particularly when researchers lack deep cultural grounding. We emphasize that our models offer analytical tools, not interpretive authority. Effective use requires ongoing dialogue between computational and humanistic expertise, with computational practitioners deferring to literary scholars on questions of meaning, value, and cultural significance.

Accessibility and Inclusion: By releasing open resources and emphasizing efficient architectures, we aim to democratize access to computational literary analysis. However, barriers persist in technical knowledge requirements, linguistic expertise prerequisites, and infrastructure needs (even modest GPUs remain inaccessible in many contexts). Future work should explore even lighter-weight approaches (e.g., mobile deployment) and develop educational materials to lower entry barriers.

Representational Fairness Our corpus emphasizes canonical male poets, reflecting their historical literary prominence but perpetuating a representational imbalance. Women poets (e.g., Ada Jafri, Parveen Shakir) and marginalized voices merit dedicated attention. Corpus expansion must consciously address whose voices receive computational amplification and whose remain unrepresented in training data.

7. Conclusion and Future Directions

This study demonstrates that compact transformer architectures, trained via parameter-efficient fine-tuning and curriculum learning, achieve effective stylistic analysis of classical Urdu poetry despite severe resource constraints. Our meter detection system attains F1 scores of 0.80, surpassing rule-based approaches by 14 percentage points, while rhyme identification reaches

F1=0.82, and poet-specific clustering achieves 88% purity. These results establish baseline performance metrics for Urdu computational poetics while contributing methodological advances applicable across low-resource literary traditions.

Beyond technical contributions, this work exemplifies productive interdisciplinary engagement between computational linguistics and humanistic scholarship. Our annotation schemas operationalize classical prosodic theory for computational implementation; our error analyses reveal theoretically interesting edge cases where formal rules meet poetic license; our stylometric findings enable new forms of quantitative literary inquiry. We released all research artefacts, corpus, code, and trained models to facilitate community engagement and encourage further development.

7.1 Future Research Directions

Corpus Expansion: Priority one involves scaling the annotated corpus by two orders of magnitude, targeting 20,000+ verses spanning additional poets, historical periods, and formal varieties (marsiya elegies, masnavi narratives, rubāī quatrains). Such expansion requires the development of active learning protocols to maximize annotation efficiency and the exploration of semi-supervised approaches leveraging unannotated texts.

Full Nastaliq Processing: Addressing script dependency demands, integrated solutions: improved OCR for historical editions, script-agnostic tokenization, and joint training across Roman and Nastaliq representations. Transfer learning from high-resource Arabic script processing may accelerate progress.

Fine-Grained Metaphor Analysis: Scaling metaphor annotation enables more sophisticated analysis: not merely binary detection, but also extraction of source/target mappings, novel metaphor identification, and tracking of metaphorical innovation across poets and periods. This requires both annotation infrastructure and advances in modelling (e.g., structured prediction frameworks).

Cross-Linguistic Comparison: Systematic study across Urdu, Persian, and Punjabi poetry, all employing Arud prosody but with distinct linguistic substrates, could reveal which aspects of our approach generalize and which require language-specific adaptation. Parallel corpora annotated with unified schemas would support controlled comparative investigations.

Generative Applications: Although the focus of this study is analysis, trained models could assist in creating poetry, as constrained language models enforce prosodic conventions while generating verse that is semantically coherent. These systems may inspire human poets or be used for educational purposes, but they also raise aesthetic and ethical concerns about computational creativity.

Interactive Scholarly Tools: Our models' impact would extend beyond research communities to students, enthusiasts, and the broader Urdu literary public by being implemented in user-friendly interfaces, web applications, manuscript annotation tools, and educational platforms. This calls for continuous model improvement driven by user input and meticulous user experience design.

7.2 Closing Reflection

The complex formal constraints and multi-layered semantic richness of classical Urdu poetry present significant computational analysis challenges. However, it is also tractable for structured machine learning approaches because of these very characteristics: stylistically distinct authorial voices, conventionalized metaphors, and explicit prosodic rules. Our analysis shows that even small computational resources, used carefully and with domain knowledge, can reveal trends and enable research that deepens our comprehension of this great literary heritage.

As machine learning capabilities continue to develop, there are increasing opportunities for scholarship in the computational humanities that engage non-Western traditions. However, meaningful progress requires more than just technical sophistication; it also requires cultural sensitivity, constant collaboration between computational and humanistic expertise, and a commitment to developing methodologies that promote scholarship rather than imposing foreign frameworks. By showing fruitful paths toward inclusive, culturally grounded computational poetics, we hope this work contributes to that ongoing conversation.

References

Abbas, Q. (2012). Building a hierarchical annotated corpus of Urdu: The URDU.KON-TB treebank. In Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12) (pp. 4068–4074).

Agirrezabal, M., Arrieta, B., Hulden, M., & Astigarraga, A. (2021). A machine- learning approach to automated scansion of poetry across languages. arXiv. https://arxiv.org/abs/2105.12638

Al-Omari, A. (2025). A rule-based algorithm for the detection of Arud meter in classical Arabic poetry. Journal of Computational Linguistics, 41(2), 215–230.

Ashraf, N., Aftab, S., Butt, W. H., & Mujtaba, G. (2023). Sentiment analysis based on Urdu reviews using hybrid deep learning models. Applied Computer Systems, 28(2), 215–225.

Butt, U., Veranasi, S., & Neumann, G. (2025). Low-resource transliteration for Roman-Urdu and Urdu using transformer-based models. arXiv. https://arxiv.org/abs/2503.21530

Farooqui, A. (2025). UPON: Urdu poetry generation using deep learning: A novel approach and evaluation. ACM Transactions on Asian and Low-Resource Language Information Pro- cessing, 24(1), Article 5.

Ghazvininejad, M., Choi, Y., & Knight, K. (2022). PoeLM: A meter- and rhyme-controllable language model for style imitation. arXiv. https://arxiv.org/abs/2204.06752

Haque, N. U., Nawaz, M. S., Mujtaba, G., et al. (2023). Deep learning in Urdu language tasks: Urdu text classification, sentiment analysis, and topic modeling. Artificial Intelligence Review, 56(5), 4525–4560.

Haque, N. U., Nawaz, M. S., Mujtaba, G., et al. (2025). Data augmentation combined with transformer-based sequence labeling improves NER for Pakistani languages including Urdu. Information Processing & Management, 62(1), 103512.

Hu, E. J., Shen, Y., Wallis, P., et al. (2021). LoRA: Low-rank adaptation of large language models. arXiv. https://arxiv.org/abs/2106.09685

Iqbal, S., Safder, I., Hassan, S. U., & Aljohani, N. R. (2025). Document-level sentiment anal- ysis of Urdu text using deep learning. arXiv. https://arxiv.org/abs/2501.17175

Kara, Y. E., Gencosman, B. C., Arslan, L. M., & Sigri, H. (2012). An algorithm for the detec- tion and analysis of Arud meter in Diwan poetry. Turkish Journal of Electrical Engineering & Computer Sciences. https://doi.org/10.3906/elk-1010-899

Khan, L., Amjad, A., Ashraf, N., et al. (2022). Urdu sentiment analysis with deep learning methods. IEEE Access, 9, 97803–97812.

Khan, M. A., & Zaman, A. (2024). Design and automatic extraction of Arud rules for Urdu poetry. Romanian Journal of Information Science and Technology, 27(2), 145–158.

Mumtaz, B., Akhter, M. Z., Butt, W. H., & Siddiqui, A. (2024). Automated compilation of Urdu poetry handwritten image datasets using deep learning. MethodsX, 12, 102456.

Rabbani, M. A., & Qureshi, M. A. (2021). Exploratory data analysis of Urdu poetry. arXiv. https://arxiv.org/abs/2112.02145

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT: A distilled version of BERT—smaller, faster, cheaper and lighter. arXiv. https://arxiv.org/abs/1910.01108

Siddiqui, A., & Rubab, S. (2023). Applying NLP methods for topic classification of cou- plets using transformer models in Urdu poetry. In Proceedings of the ACM Conference on Information and Knowledge Management (pp. 1234–1240).

Siddiqui, A., Rubab, S., Usman, M., & Butt, W. H. (2024). Poet attribution of Urdu ghazals using deep learning. International Journal of Advanced Computer Science and Applications, 15(3), 245–252.

You, S., Zhang, C., & Wang, X. (2018). Authorship attribution in Persian poetry using Arud- derived features. In Proceedings of the International Conference on Asian Language Process- ing (IALP) (pp. 123–128).

Zain, A. (2025). Low-resource transliteration for Roman-Urdu and Urdu using transformer. In Proceedings of the Workshop on Low-Resource Machine Translation (LoResMT 2025) (pp. 145–152).