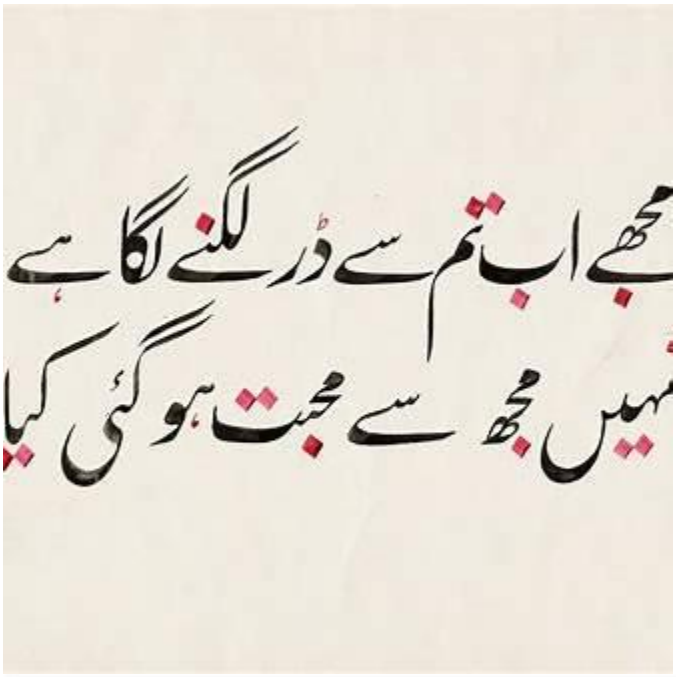

SEO package (Project 4)

Primary SEO title (recommended):
ShayarAI: Transformer-Based Urdu Shayari Generator Trained for Classical Style, Rhyme, and Meter

Alternate titles:

- Building ShayarAI: A GPT-Style Urdu Poetry Generator with Prosody-Aware Evaluation
- From Rekhta-Scale Poetry Corpora to AI-Generated Shayari: An End-to-End NLP Pipeline
- Urdu Creative AI Done Right: Style-Controlled Shayari Generation with Transformers

URL slug: `shayarai-urdu-shayari-generator-transformer`

Meta description (155–160 chars):
MuFaw AI Research Lab built ShayarAI, a Transformer-based Urdu shayari generator with poet-style control and prosody-aware evaluation.

Target keywords:

- Primary: *Urdu shayari generator AI, Urdu poetry generator Transformer, GPT Urdu text generation*
- Secondary: *Nastaliq tokenization, Urdu prosody behr taqti, style-controlled text generation, digital humanities Urdu NLP*

ShayarAI: Cultural-Tech NLP for Urdu Shayari Generation

MuFaw AI Research Lab | Creative AI + Digital Humanities

Urdu poetry is not just "text generation." It's a structured literary tradition shaped by **script (Nastaliq), sound patterns (behr), rhyme conventions, and cultural nuance**. Nastaliq is widely treated as the standard writing style for Urdu, which immediately makes preprocessing/tokenization non-trivial in modern NLP pipelines.

ShayarAI is MuFaw AI Research Lab's end-to-end system that learns from large-scale classical Urdu poetry sources (including public repositories like Rekhta) to generate **original couplets** that are stylistically aligned to a selected poet voice—while keeping the output explicitly labeled as **AI-generated** (not authentic poetry by the poet).

The problem: Urdu creative AI tools are scarce (and usually shallow)

Most “poetry generators” ignore the hard parts:

- Urdu’s right-to-left script and Nastaliq conventions
- Prosody: meter (behr) and scansion (taqti)
- Style control across poets and eras
- Usable tooling for educators, researchers, and creators

Digital humanities research repeatedly highlights that low-resource, right-to-left languages like Urdu remain underserved in computational tooling.

What we built

ShayarAI combines:

1. **Data pipeline** to collect and structure Urdu poetry into `[Poet, Verse]` training pairs
2. **Urdu-aware tokenization** suitable for Transformer training
3. **Transformer-based generation model** (GPT-2-like) fine-tuned for Urdu poetic patterns
4. **Style + topic control** via prompts/conditioning fields
5. **Prosody-aware evaluation hooks** (meter/rhyme checks as signals, not absolute “truth”)
6. A deployable demo: **FastAPI backend + Streamlit UI**, with optional hosting on Hugging Face Spaces

---

Data and cultural grounding (without overclaiming rights)

Rekhta is a major Urdu literature platform aimed at disseminating Urdu poetry broadly, including rendering content in multiple scripts beyond Urdu.
ShayarAI’s dataset pipeline is designed to work with **publicly available poetry text** while keeping a strict boundary around:

- licensing/permissions,
- source attribution policies,
- and “no impersonation” usage (the system does *style inspiration*, not identity claims).

How ShayarAI works (architecture)

Web ingestion → clean dataset → tokenizer → fine-tune → generate → evaluate → serve

1. **Web ingestion** (public poetry pages)
2. **Urdu-specific cleaning** (diacritics handling, punctuation normalization, duplicate removal)
3. **Structured dataset creation**: `[Poet, Couplets]`
4. **Tokenization (Urdu BPE)** for stable subword modeling
    - GPT-style models commonly rely on Byte Pair Encoding variants; GPT-2 tokenization is byte-level BPE, which helps avoid unknown characters.
5. **Model training / fine-tuning** using a GPT-2-like Transformer
    - GPT-2 is a Transformer language model trained to predict the next token and can generate coherent continuations from prompts.
6. **Inference API** (prompt + poet-style tag + topic)
7. **Evaluation signals** (meter/rhyme heuristics + scansion tools)

Prosody and "meter-aware" evaluation (what makes it not a toy)

In Urdu prosody, **bahr** is a meter pattern used in Urdu (and related) poetic traditions.
Taqti is scansion—breaking verses into rhythmic units to verify patterns.

Rekhta itself provides a "Taqti" tool that describes scansion as identifying rhythmic patterns and checking bahr compliance.
ShayarAI uses this idea pragmatically: meter/rhyme checks act as **evaluation metrics** and **filters**, not as a guarantee that every generated couplet is "classically perfect."

Real-world use cases (technical + non-technical)

1) Education and literature learning

- Generate practice couplets for teaching poetic devices, vocabulary, and stylistic differences between poets
- Use meter checks (taqti) to teach rhythm patterns rather than memorizing rules blindly

2) Digital humanities and Urdu NLP research

Computational analysis of Urdu poetry is an active research area (e.g., stylistics and sentiment analysis on large ghazal corpora).
ShayarAI supports:

- dataset creation + normalization,

- generation experiments,
- and controlled comparisons across poet styles.

3) Cultural organizations and preservation tooling

Rekhta Foundation's stated mission is preservation and promotion of Urdu language, literature, and culture.
ShayarAI fits as a "creative exploration layer" on top of preservation efforts—helping audiences interact with the tradition in modern formats.

4) Content creators (with guardrails)

- Generate draft couplets for captions, scripts, and themes
- Explicit labeling + "AI-generated" watermarking to prevent misattribution/impersonation

---

Tech stack

- **Modeling:** GPT-2-like Transformer fine-tuned with PyTorch
- **Tokenization:** Urdu-aware BPE workflow
- **Data:** BeautifulSoup, pandas, regex cleaning
- **Backend:** FastAPI
- **Demo UI:** Streamlit
- **Optional retrieval:** FAISS similarity search (prompt grounding / nearest-neighbor inspiration)
- **Hosting:** Hugging Face Spaces for demo deployment

Responsible deployment notes (what clients ask first)

- Public-domain/publicly accessible content pipelines must still respect **source policies and licensing**.
- Generated output must be **clearly labeled as AI-generated**, especially when style-conditioning references famous poets.
- No "this is by Ghalib/Faiz" claims; only "inspired-by / style-conditioned."

FAQ (SEO-friendly)

Is ShayarAI copying existing poetry?
The goal is original generation from learned patterns, but robust safeguards matter: deduplication checks, similarity search, and clear labeling reduce misattribution risk.

Why GPT-style Transformers for poetry?
Transformer LMs like GPT-2 are designed for next-token prediction and can adapt to prompt style, making them practical for controlled text generation tasks.

How do you handle Urdu script complexity?
By Urdu-aware preprocessing and tokenization, designed for Nastaliq-based Urdu text conventions.

Can it enforce meter (behr) perfectly?
No. Prosody checks act as evaluation and filtering signals; perfect classical compliance requires deeper constraints and/or human review.

CTA (MuFaw AI Research Lab)

Try the demo and generate your own Urdu shayari with AI.
If you want this trained on your curated corpus or constrained to specific meters/topics, request a pilot build.