📝 Daily Work Report – July 17, 2025

Internship at Geniteam Solutions
Intern: Dua Mohyyuddin

✅ Today's Focus:

I explored Transformer-based language models and worked on enhancing a custom poetry generation model using the Bigram Language Model architecture. The work included both **theoretical understanding** and **practical implementation enhancements**.

📚 Key Learnings:

- Understood the **Transformer architecture**, including self-attention, multi-head attention, layer normalization, and positional embeddings.
- Studied how these mechanisms help models understand long-range dependencies and generate coherent sequences.
- Compared **character-level** tokenization (used in our Rumi poetry model) vs. word/subword-level approaches in real-world applications.

🛠️ Enhancements Made to the Existing Model:

1. Temperature Sampling – *Better Text Generation Control*

- Allows dynamic control over the randomness/creativity of generated poetry.
- Added options to sample with low (0.5), medium (1.0), or high (1.5) temperature.
- **Result:** More diversity and user control in poem generation style.

2. Learning Rate Scheduling – *Smarter Training*

- Implemented OneCycleLR to manage learning rate during training.

- Mimics a human learning process: starts slow, speeds up, then refines.
- **Result:** Improved convergence and training stability.

3. Gradient Clipping – *Stable Training*

- Prevented exploding gradients by clipping them to a threshold.
- **Result:** More reliable training even with complex sequences.

4. Rhyme Helper – *Poetry-Specific Utility*

- Created a utility to suggest rhyming words using the last 3 characters.
- Example: “love” → “dove”, “above”, etc.
- **Result:** Enhanced poetic structure and musicality.

5. Gated Linear Units (GLU) – *Smarter Computation*

- Replaced standard feedforward layers with GLU.
- Allows finer control of neuron outputs (like a dimmer switch).
- **Result:** Improved ability to learn nuanced poetic structures.

6. Mixed Precision Training – *Faster + Memory Efficient*

- Enabled automatic mixed precision using PyTorch’s `GradScaler`.
- **Result:** Faster training on GPU with reduced memory usage.

✨ Results from Enhancements:

- The model can now generate poetry with different tones:
    - **Creative mode (temp=1.2)** produces wild, imaginative lines.

    - **Conservative mode (temp=0.7, top_k=40)** gives more structured and traditional poems.

- Rhyme suggestions now help generate verses that follow classical poetic forms.