Generation of Urdu Ghazals using Deep Learning

***Abstract*—Poetry generation is a branch of text generation that demands a high level of linguistic proficiency and creativity. The Urdu language, renowned for its rich poetic tradition, encompasses a diverse collection of poets, each possessing a unique writing style. This research explores automatic generation of Urdu poetry, specifically Urdu ghazals which hold a significant cultural value using natural language processing techniques including deep learning. The dataset used consists of 17,609 couplets from the works of 15 renowned Urdu ghazal poets. We use three different approaches for poetry generation which include (a) the *n*-gram probabilistic model, (b) deep learning models such as LSTM and GRU, and (c) the state-of-the-art GPT-2 model. BLEU scores for each of these were (a) 0.5 for the n-gram model, (b) 0.1 for LSTM and GRU and, (c) 0.6 for the GPT-2 model, which was the highest overall score. In addition to the BLEU score, other metrics such as rhyming score and human evaluation were considered for evaluating the generated poetry. This research is the first to use transformer-based models for Urdu poetry generation, which enables aspiring poets to learn from and be inspired by the rich tradition of Urdu poetry.**

***Index Terms*—urdu poetry generation, natural language processing, deep learning, transformers, computational linguistics**

I. Introduction

Urdu, a language that is widely used in Pakistan and other South Asian countries, has a rich tradition of poetry that dates back to centuries. It encompasses a diverse range of poets, poetic styles, and verse. Urdu poetry is often categorized into structured and unstructured forms, with the former being more popular among poets and readers alike.

The creation of structured Urdu poetry poses several constraints, including meter, rhyme, refrain, and mood, all of which must be considered in crafting a well-formed couplet. In addition to these limitations, the traditional poetic style established by the great poets of the past constitutes a valuable legacy for aspiring poets of the Urdu language.

The field of Natural Language Processing (NLP) has seen significant advancements in recent times, enabling machines to comprehend and generate text with increasing sophistication. Text generation considers both semantic and syntactic aspects of language, while poetry generation is a specialized variant of text generation that takes into account the writing style of a specific poet, the structure of poetry in order to generate poetry in a similar style. It is an intersection of natural language processing and computational creativity.

In this research, we apply advanced NLP techniques and deep learning to generate Urdu poetry in a specified poetic style. Our approach involves training deep learning and transformer-based models on a dataset of Urdu couplets by 15 poets. The eventual goal of this project is to preserve the rich tradition of Urdu poetry and make it accessible to aspiring poets of future generations.

The main contributions of this research are:

1) Computer-generated Urdu poetry: The development of a framework for generating Urdu poetry in the style of well-known poets, providing a foundation for automated Urdu poetry generation.
2) Diverse Dataset and Poetic Styles: Training models on a diverse dataset covering 15 different poetic styles, surpassing previous works that typically focus on fewer styles.
3) Advancement in Urdu Poetry Generation: Addressing the limited research in Urdu poetry generation, particularly the absence of transformer-based models, and showcasing their effectiveness in generating Urdu poetry.
4) Departure from RNN-based Approaches: Moving away from traditional recurrent neural network (RNN) architectures commonly used in Urdu poetry generation, demonstrating the advantages of transformer-based models.

II. Literature Review

With the advent of cutting-edge advancements in Natural Language Processing and computational creativity, the realm of poetry generation has become an area of immense significance. In this section we present various research works on poetry generation in Urdu as well as related languages such as Arabic.

In 2019, a pioneering research [6] utilized deep learning-based NLP techniques for generating Arabic poetry. The study employed a two-step approach, beginning with a phonetic CNN sub-word embedding and keyword extraction/expansion based on semantic similarity. The first verse was generated through the utilization of a Backward and Forward Language Model (B/F-LM) equipped with a Gated Recurrent Unit (GRU) cell, while subsequent verses were generated through the implementation of a HAS2S model. BLEU score for the proposed model was 0.6 surpassing the performance of RNN, GRU, and LSTM based models.

Subsequently in 2020, a research work [7] on Turkish poetry generation in the style of three renowned Ottoman poets of the 16th century - Necati, Mihri Khatun, and Revani was published. The model was trained on a corpus of 9484 couplets attributed to these poets. This work was inspired from an earlier work in 2017 [8] on Hafez - a poetry generator for Persian using RNN. This research proposed a variant of Hafez, named Binari, designed as a Turkish ghazal generator using RNN. The model consisted of a one-layer RNN with GRU units, trained over syllables extracted from a finite state transducer (FST). Both character-level and syllable-level training were

employed, however, the results were not impressive in terms of meaningfulness and grammar.

Despite the scarcity of literature on poetry generation in Urdu, a research work [4] has focused on generating poetry in both Urdu and Hindi through character-level RNNs equipped with LSTM. The study initiated with data through web scraping from sources such as Rekhta.org.The model was trained in three modes - generating *Misra* (one-liners), *Sher* (two-liners), and *Ghazal* (full poems). The results were quite promising, with the model generating acceptable poetry in a ratio of roughly 14 out of 20 in both Urdu and Hindi.

Similarly, another recent work [3] for text generation in Hindi uses RNNs and LSTMs for text classification, translation, and recognition. The training and validation data consists of 5409 and 2318 lines of poetry in Hindi. In this work, four models were experimented with for generating poetry in Hindi: Bidirectional LSTM Generator (BLG), Bidirectional LSTM Generator with Self-Attention (BLG-SA), Bidirectional LSTM-Conv Generator (BLCG), and Bidirectional LSTM-Conv Generator with Self-Attention (BLCG-SA). Results showed that by using RNNs and LSTMs together with CNNs, sequences can be learned with lesser training and time.

In another recent work [2], authors aimed to develop a machine learning model for generating Arabic poetry. They compared the performance of three different models: Long Short-Term Memory (LSTM), Markov-LSTM, and Pre-Trained Generative pretrained transformer-2 (GPT-2). Results showed that GPT-2 model performed the best in terms of generating fluent and relevant outputs, while the LSTM model performed the worst. The study highlights the potential of using pre-trained models for generating poetry in Arabic

In 2022, a study [1] fine-tuned GPT-2 on a corpus of classical Arabic poems and evaluated the model's ability to generate poems in terms of fluency and creativity. The results showed that the model was capable of producing coherent and fluent poems, however, the creativity of the generated poems was limited.

The various works discussed throughout this overview are presented in Table I. The majority of research in this domain has been conducted in other languages such as English, Turkish, and Arabic, with NLP in Urdu being largely restricted to machine translation. However, there is a pressing need to leverage the power of NLP and neural networks to generate Urdu poetry and keep alive its rich and enduring tradition of poetry.

TABLE I: Literature Review Summary

Year	Corpus Language	Model
[1] Jan 2022	Arabic	GPT-2
[2]Sep 2021	Arabic	Character level LSTM and Markov-LSTM and GPT-2
[3] Aug 2021	Hindi	BLG-SA and BLG and BLCG-SA and BLCG
[4]Jul 2021	Urdu and Hindi	Character-level RNN with LSTM
[6]Oct 2019	Arabic	Phonetic CNN embedding with B/F-LM-GRU+HAS2S
[7]Jul 2020	Turkish	Character level RNN and Syllable level RNN

III. Dataset and Methodology

In this section, we present our proposed framework for Urdu poetry generation, which includes the dataset, models, and experimentation. We commence by discussing the corpus and dataset in Section III-A; subsequently we present three primary approaches for poetry generation as follows: Section III-B focuses on probabilistic NLP techniques, Section III-C delves into Traditional Deep Learning models, and Section III-D showcases state-of-the-art natural language processing models.

A. Data Set

We have used an open source corpus having 17,609 Urdu couplets belonging to about 15 notable Urdu poets. The corpus was curated in another study from Rekhta, publicly available here The dataset is a csv file with Urdu couplets and labels. The labels are integers from 1-15 with each integer representing a poet. Table II presents a breakdown of the corpus taken from [5].

TABLE II: Number of Couplets for each poet in Corpus [8]

No	Poet/Shayar	Couplet Count
1	Ahmed Faraz	926
2	Zafar Iqbal	1104
3	Qateel Shifai	780
4	Parveen Shakir	593
5	Nida Fazli	474
6	Faiz Ahmad Faiz	504
7	Jaun Elia	1470
8	Muneer Niyazi	523
9	Allama Iqbal	797
10	Riyaz Khairabadi	1700
11	Haider Ali Atish	1330
12	Siraj Aurangabadi	860
13	Mir Taqi Mir	2971
14	Nazeer Akbar Abadi	1643
15	Mirza Ghalib	1934

B. Probabilistic Model

We initiated our experiments using a probabilistic model, specifically the n-gram model, with n set to 3 for the task of Urdu couplet generation. The n-gram model is a probabilistic model commonly used in natural language processing (NLP) to predict the likelihood of a sequence of words based on the frequency of their co-occurrence. Figure 1 gives a block diagram of steps in our experimentation for the n-gram model.

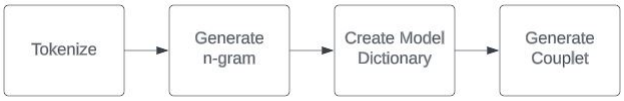

Fig. 1: Block diagram for the n-gram Model

Pre-processing: In the pre-processing stage, we normalized the dataset and tokenized the text using a regular expression tokenizer (RegeXtokenizer). The tokenizer split the text into individual tokens, such as words or phrases. The tokenized text was then divided into a training set (80% of the data) and a test set (20% of the data) for model development and evaluation.

Model: The n-grams were calculated for the training data. Each n-gram represented a sequence of n words in the text. We built a probabilistic model by estimating the probability of each n-gram occurring in the text based on its frequency in the training data.

Finally, the trained model was used to generate Urdu couplets (*sher*) with each line representing a *misra* of the couplet. The model utilized the estimated probabilities of n-grams to generate coherent and meaningful couplets. The results of this experimentation are presented in Sections IV and V.

C. Traditional Deep Learning Models

We conducted experiments using two traditional deep learning models, namely Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), for the task of Urdu couplet generation. LSTM and GRU are types of recurrent neural networks (RNNs) commonly used for sequence prediction and language modeling tasks. Figure 2 provides a block diagram of our experimentation with these traditional deep learning models.

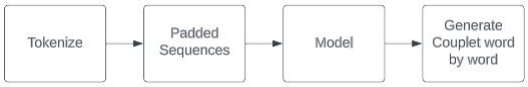

Fig. 2: Block Diagram for the Deep Learning Models (RNNs)

Pre-processing: The pre-processing stage for both RNN models was identical. The dataset was first normalized, tokenized, and converted into padded sequences. This process involved representing each word as a numerical vector and adding padding to ensure that all sequences had the same length. During the experiment, we set the maximum sequence length to 10, as Urdu poetry typically consists of couplets of this length. The data was then divided into a training set (80% of the data) and a test set (20% of the data) for model development and evaluation.

Models: The padded sequences obtained from the pre-processing stage were used to train the traditional deep learning models.

a) LSTM: We trained an LSTM model with 100 LSTM units on the training data. The first layer of the model was an embedding layer that converted the word vectors into a continuous vector space. The final layer used a softmax activation function to select the word with the highest probability. The model was trained for 70 epochs using the Adam optimizer, which is an adaptive learning rate optimization algorithm, with a learning rate of 0.001. The hyper-parameters were chosen with trial and error.

b) GRU: A GRU model with 120 GRU units was trained on the training data. Similar to the LSTM model, the first layer of the GRU model was an embedding layer that converted the word vectors into a continuous vector space. The final layer used a softmax activation function to select the word with the highest probability. The model was trained for 10 epochs using the Adam optimizer, with a learning rate of 0.001. The hyper-parameters were chosen with trial and error.

The trained GRU and LSTM models take a seed sequence (a sequence of words) as input and generate a two-line Urdu couplet (*sher*), with each line representing a *misra* of the couplet, starting with the seed sequence. This generation process involves using the probabilities estimated by the models to predict the most likely words to follow the seed sequence. Subsequently, the models recurrently generate subsequent words based on those predictions until the entire couplet is generated. The results for each experiment will be discussed in Sections IV and V.

D. Generative Transformer-based Model

In our final set of experiments, we utilized the state-of-the-art GPT-2 model (Generative Pre-trained Transformer) for Urdu couplet generation. GPT-2 is a large language model developed by OpenAI and, recognized for its performance in generating natural language text. Figure 3 provides a block diagram of our experimentation with the GPT-2 model.

Fig. 3: Block Diagram for the GPT-2 Model

Pre-processing: To prepare the data for training for the GPT-2 model, we initialized the model with its corresponding tokenizer. The tokenizer played a crucial role in splitting the input text into individual tokens, which were then encoded into numerical representations suitable for the GPT-2 model.

Next, we normalized the dataset used for training and applied the tokenizer to tokenize and encode the dataset. This process transformed the textual data into a format that could be effectively used to train the GPT-2 model. Additionally, a data loader was created to efficiently feed the encoded dataset into the model during training. The data loader facilitated the loading and batching of the training data, optimizing the training process.

Model: We trained the GPT-2 model using the encoded dataset, specifically to generate entire lines of Urdu couplets. The pretrained GPT-2 model was fine-tuned on a dataset of Urdu couplets for approximately 70 epochs. Each epoch represents a complete pass of the training data through the model, allowing the model to learn from the data and improve its performance over time. During training, we employed the Adam optimizer with a learning rate of 0.000002, which determined the step size for parameter updates and optimization. The results of these experiments, including the quality of the generated couplets are discussed in Sections IV and V.

IV. Results

This section presents the results of the aforementioned experiments. As the task was for the poetry generation therefore reliance on a metric like accuracy was not sufficient. To assess the quality of our generated poems we focused mainly on two types of evaluation: metrical evaluation and human evaluation. In metrical evaluation, apart from the validation accuracy shown by models, the generated couplets were evaluated by BLEU score and the rhyming score. On the other hand, in human evaluation, the generated poetry was reviewed by Urdu poetry experts from Habib University. The reviewers then passed remarks about the generated poetry using the knowledge and parameters of urdu poetry, mainly critiquing on Beher [1], Radeef[2], Kafiya[3] and Tashreeh[4].

For the purpose of comparing different models and conducting evaluations, we selected three Urdu seed words and generated couplets using all trained models. These generated couplets were then evaluated using metrical and human evaluation criteria.

A. Probabilistic Model

Table III presents the results of the n-gram model. Metrical and Human Evaluation results are presented with each couplet.

TABLE III: Evaluation Results for n-gram Model

Seed Word	Mu (Urdu for face)	Gham (Urdu for sorrow)	Awaaz (Urdu for Sound)
Couplet	ہو نظارہ سوز پردے میں منہ چھپائے کیوں شیشوں لے سر حضور ساغر جھکا دیے ہیں	ترے غم کے عوض ساری خدائی لے لے جس سے ہمرے کی تھی ہارپا نیکی	کلی ہے نیند کے دوارے جا کے دیکھو یہ آواز اسی خانہ خراب کی سی ہے
BLEU Score	0.5	0.3	0.4
Rhyme Score	0.3	0	0
Expert Evaluation	Sher resembles the style of Mir, Follows beher perfectly along with making sense in terms of language and grammar, has a poetic meaning behind it as well. Radeef and kafiya can make sense if more rhyming couplets follow.	Sher resembles the philosophical style of Iqbal, Does not follow radeef or kafiya, but does follow baher, may need some context to make sense in terms of language and poetic meaning, but grammar is ok.	Sher resembles the style of Jaun Elia, but Mir also has a very similar sher, although it doesn't rhyme within the sher

[1]meter or rhythm of a ghazal, which is a form of Urdu poetry consisting of rhyming couplets. Each couplet in a ghazal has a specific number of syllables and a specific pattern of stress and un-stress, which determines its beher. Most common beher in Urdu poetry is the "matla", which consists of two lines of equal length and the same beher

[2]refers to the rhyme or the last word(s) of a couplet

[3]a rhyming pattern that occurs at the end of each line in a couplet or a poem

[4]refer to the explanation or interpretation of a verse or couplet

B. LSTM

The LSTM model was able to reduce the training loss to 3.7 in 70 epochs with an accuracy of 28%. The loss curves for LSTM are shown in Figure 4, and the evaluation results are presented in Table IV.

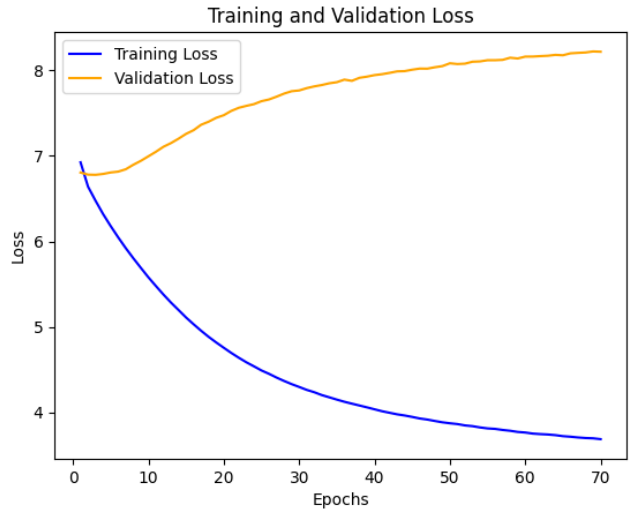

Fig. 4: Loss Curves for LSTM

TABLE IV: Evaluation Results for LSTM Model

Seed Word	Mu (Urdu for face)	Gham (Urdu for sorrow)	Awaaz (Urdu for Sound)
Couplet	منہ لا لطف کہاں سفر دل پسند ذرا نشیں نہ فائدہ نظر کے آفتاب میں ہو دیکھا اک	غم ہے تیری بات بس پر زعم دیوار ہے کی گماں میں تھے خرابات قفل شمشاد جگروں ہوئے	آواز تو نہیں ہے ہوئے نہ سیر آنکھیں تو یقیں خبر چھوڑے میں ہم پہنچائیں ہوں جہاں ہوں
BLEU Score	0.135	0.138	0.131
Rhyme Score	0	0.3	0.2
Expert Evaluation	A meaning overall cannot be extracted here as well, but some exhibits of grammatical sense include phrases like 'dil pasand', it contains meaningful connections like between words 'aftaab' and 'aag' but they dont connect well altogether to form a sher.	There is a rhyme between two misra's, but the words fail to make sense, an overall meaning cannot be extracted again, but there is still some connection and grammatical sense seen like between the neighboring words.	There is no consistent flow in the sher overall, but the words have connection to their surrounding words. A meaning cannot be extracted from the sher even though the words ending both misra's rhyme.

C. GRU

The GRU model was able to reduce the loss to 3.5 in 70 epochs with an accuracy of 32%, which is better than LSTM. The loss curves for GRU are shown in Figure 5, and the evaluation results are presented in Table V.

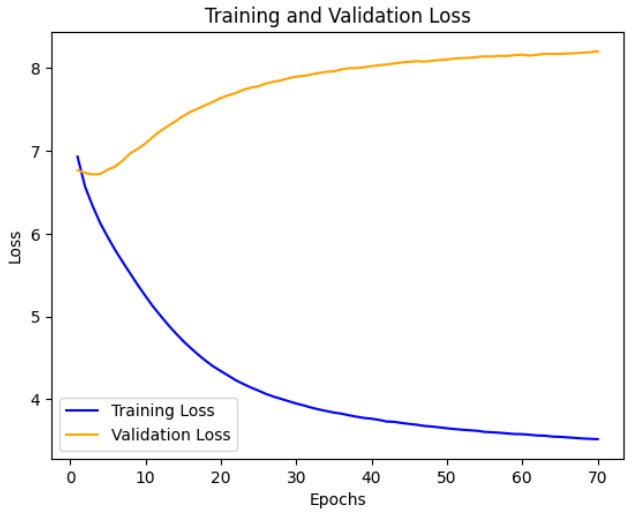

Fig. 5: Loss Curve for GRU

TABLE V: Evaluation Results for GRU Model

Seed Word	Mu (Urdu for face)	Gham (Urdu for sorrow)	Awaaz (Urdu for Sound)
Couplet	منہ قماش تجلی اب نے دم کاتب کھلے کو کے انداز و خفتہ ہے سیکھی آتش قدرت ماہ	غم دست رنگیں ہے عشق دیدہ گردی بڑی کن ناز جل گئی ہے یہ میڑنے بلا کی	آواز زندہ واجب جو ہیں کہ پرواز ہے پھرے دست افسوس ہے عناد ہے مت دیباز میں وہ
BLEU Score	0.141	0.132	0.133
Rhyme Score	0	0	0
Expert Evaluation	The individual connections make most sense in this couplet, as there can also be a logical meaning extracted from different parts of both misra's as visible by the link between 'muh' and 'tajalli'.	Again, there are some sensible connections between words, the model seems to understand some different parts of sentences and makes partial grammatical sense as seen in phrase 'naar jalgai'. There is a mention of Mir so it tries to replicate Mir's style here.	There is a touch of sensibility and connection between, however they don't make a sensible sentence when combined together. We can see that the model is trying to replicate Iqbal's style here, visible by the word 'parwaaz'.

D. Generative Transformer based Model

GPT-2 was able to reduce the loss to 1 in 70 epochs, which is better than both LSTM and GRU. The loss curve for GPT-2 is shown in Figure 6, and the evaluation results are presented in Table VI

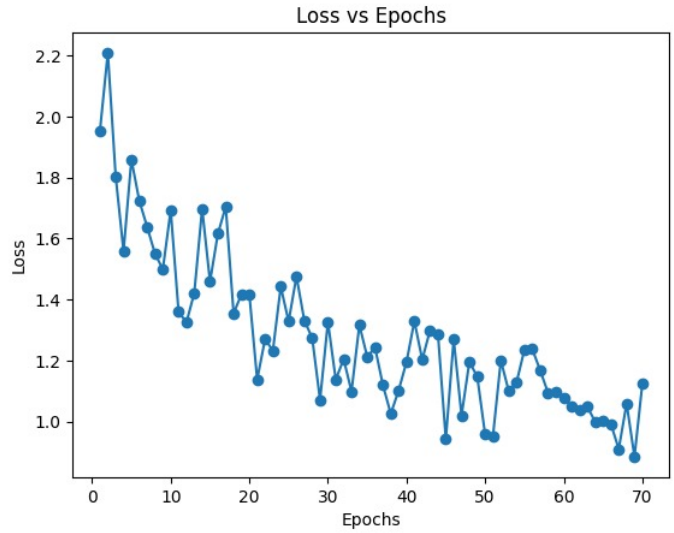

Fig. 6: Loss Curve for GPT-2

TABLE VI: Evaluation Results for GPT-2 Model

Seed Word	Mu (Urdu for face)	Gham (Urdu for sorrow)	Awaaz (Urdu for Sound)
Couplet	منہ دکھا کی جانی ہے یادلے یادا اسا انہ تو تو خو خالی چلے خونہ چلے خونہ خالی خالی خال	غم کی جو کہاں ہے بھی انی بھی انی ی بھ بولا بولا بھی ی دلا بلا دنوگی گی تو ہم	آواز مرے گیا جب آہاں تھے خون جو تھہلے خون خالی میرے تو بھی ترے خرون
BLEU Score	0.48	0.2	0.6
Rhyme Score	0.3	0.3	0.34
Expert Evaluation	A slight hint of sensibility as the model successfully linked two words: 'Muh dikha", but overall it still does not look like a sher	Includes more words and the repetition is reduced. However the couplet does not make grammatical or linguistic sense	Two to three instances of linguistic and grammatical sense in the very beginning of the couplet but the model fails to connect them all to make a sensible sentence, the only parameter seems to be correctly working in the model so far is the consistency of length in misra's.

V. Discussion

The results highlight the strengths and limitations of the different models used in the experiments. The n-gram model, as shown in Table III with a BLEU score of 0.5, generated the best sher in terms of objective evaluation metrics. However, it lacked context and meaningfulness, indicating the limitation of the n-gram model in capturing the diversity and richness of language. Furthermore, human evaluation revealed that the couplets generated by the n-gram model were highly similar to the training data, which emphasizes the fact that the simplistic method of n-gram is unable to learn the relationships between the words themselves, and hence unable to exploit that knowledge to create something new, albeit erroneous.

On the other hand, the neural networks, LSTM and GRU, exhibited lower BLEU and rhyming scores compared to the n-gram model, as depicted in Tables IV and Table V. However, human evaluation indicated that these models attempted to establish sensible connections and grammatical sense, as evident in phrases like "naar jalgai" in Table IV. Unlike the n-gram model, they generated unique Urdu phrases and contexts while maintaining some similarity to the training data.

The loss curves, as shown in Figures 4 and 5, demonstrated that both LSTM and GRU models suffered from over-fitting and achieved a relatively low accuracy of approximately 28% and 32% respectively. This indicates the need for further optimization and regularization techniques to improve their performance.

The introduction of the state-of-the-art GPT-2 model opened up new possibilities for poetry generation. As shown in Table VI, GPT-2 exhibited impressive BLEU scores, reaching a high of approximately 0.6 for the couplet generated with the word seed "awaaz". However, human evaluations revealed that the generated couplets lacked ghazal patterns such as qafiya.

Despite this limitation, it was remarkable to observe that within just 70 epochs, the GPT-2 model was able to establish connections based on word context. For instance, in the first couplet, it associated "mouth" with "sight", generating the phrase "mun dikha". This demonstrated a certain level of sensibility, better utilization of Urdu words, and adherence to beher as compared to the other experimented models. Moreover, the training loss of GPT-2 was much lesser at the end of experimentation as compared to any other model.

Considering these initial results, it is expected that with further training on higher epochs, the model will enhance its understanding and linguistic sensibility. This holds promise for the future refinement and improvement of the GPT-2 model in generating high-quality Urdu couplets.

VI. Conclusion and Future Work

Despite being a challenging and intriguing research area, the task of Urdu poetry generation using natural language processing and deep learning has received limited attention. In this study, we aimed to fill this gap by exploring the application of deep learning techniques in generating Urdu couplets. Our approach involved training and testing various models using a dataset comprising 17,609 couplets from 15 renowned poets. We conducted both statistical and human evaluations on the generated couplets to assess their quality.

Our findings state that GPT-2, a finetuned transformer-based model, outperformed other models in terms of BLEU score (a metric for evaluating text similarity) and the sensibility of the generated couplets. Traditional deep learning models such as LSTM and GRU exhibited lower accuracy scores and produced redundant couplets. However, the n-gram model demonstrated the highest performance in terms of rhyming score.

For future work, further refinement through the utilization of alternative Urdu embeddings can significantly improve the quality of the generated couplets. Moreover, the potential for generating more meaningful couplets that closely adhere to traditional poetic styles can be unlocked by extending the training duration of the GPT model and implementing enhanced embedding techniques.

References

[1] Beheitt, Mohamed El Ghaly, and Moez Ben Haj Hmida. "Automatic Arabic Poem Generation with GPT-2." In ICAART (2), pp. 366-374. 2022.

[2] Asmaa Hakami, Raneem Alqarni, Mahila Almutairi, and Areej Alhothali. "Arabic Poems Generation Using LSTM, Markov-LSTM and Pretrained GPT-2 Models." Computer Science & Information Technology (CS & IT), pp. 139-147. 2021.

[3] Ankit Kumar. "Bidirectional LSTM Networks for Poetry Generation in Hindi.", International Journal of Innovative Science and Research Technology Volume 6, pp. 885-888. 2021.

[4] Mukhtar, Shakeeb AM, and Pushkar S. Joglekar. "Urdu & Hindi Poetry Generation using Neural Networks." arXiv preprint arXiv:2107.14587, 2021.

[5] I. Siddiqui, F. Rubab, H. Siddiqui and A. Samad, "Poet Attribution of Urdu Ghazals using Deep Learning," *2023 3rd International Conference on Artificial Intelligence (ICAI)*, Islamabad, Pakistan, 2023, pp. 196-203

[6] Sameerah Talafha and Banafsheh Rekabdar. "Arabic poem generation incorporating deep learning and phonetic cnnsubword embedding models." International Journal of Robotic Computing, pages 64–91, 2019.

[7] Galip. "Binari: A poetry generation system for ghazals.", Department of Computer Engineering Faculty of Engineering Bogaziçi University, 2020.

[8] Ghazvininejad, Marjan, Xing Shi, Jay Priyadarshi, and Kevin Knight. "Hafez: an interactive poetry generation system." In Proceedings of ACL 2017, System Demonstrations, pp. 43-48. 2017.