

# A Multimodal AI Framework for Telugu Poetry Understanding and Visual Representation

Allu Sowrya Deepika<sup>1</sup>, Prof. Dhavala Lalitha Bhaskari<sup>2</sup>

<sup>1</sup>M. Tech Student, Department of Computer Science and Systems Engineering, Andhra University College of Engineering, Visakhapatnam, India

<sup>2</sup>Professor, Department of Computer Science and Systems Engineering, Andhra University College of Engineering, Visakhapatnam, India

**Abstract**—Classical Telugu poems (padyams) express the essence in poetic metrical verse, with the use of archaic words and cultural allusion, making them difficult to read by the modern reader and even by general language models. This paper presents the Padyam2Gadyam+ pipeline, which takes a Telugu poem and translates the same into Telugu prose (gadyam) and into an image. Using only 549 poem–meaning–English triples from a set of 599 poem–meaning–English triples, Gemma-2-9B-it is trained with 4-bit Quantized Low-Rank Adaptation (QLoRA), which updates 0.58% of the model's parameters. In Module 2, the English meaning is given as a structured text instruction to be passed to Gemini API as a text-to-image prompt. For Module 3, a prompt is created with Stable Diffusion 1.5 and LoRA adapter. The model is fine-tuned for a test set of 20 poems which were not used in the training set, and a relative BLEU gain of 86.7% is observed compared to the same base model with three shots. The baseline scores for all three measures are all below the 95% confidence interval limits for the fine-tuned model. Qualitative analysis is used for representative examples to analyze prompt and image outputs. The modules are wrapped in a Flask Web App which will provide the English translation and image of a given poem. The results indicate that it is possible to develop an adaptation of a large-scale quantized model for the task of poetry interpretation using a small and carefully selected subset of data.

**Index Terms**—Telugu poetry, low-resource machine translation, QLoRA, Gemma, Stable Diffusion, multimodal AI.

## I. INTRODUCTION

Telugu is an old language which boasts metrical poetry where a poetic verse (padyam) can contain everything like the meaning in a regular meter, the

compound words, the old meanings and the references of epics and puranic stories. A reader who can read only the modern Telugu language - and a reader of any other language - typically require a prose gloss (gadyam) to follow a verse. It's difficult to automatically produce such glosses for two reasons. The resources for modeling Telugu grammar are challenging due to its features: The word form will carry multiple markers and the resources to cover for this are still limited [1, 2]. In addition, the difficulty of reading poetry isn't in figuring out the meaning of a specific word, but the ability to infer the meaning of a line within the context of the poem and then express that meaning in the literal language.

This is not a problem that is cleared by large language models (LLMs). A study has been conducted to analyze, retrieve, generate and support tasks of Telugu verse recently, which can be used as a challenging domain to benchmark LLMs [3]. Task-dependent adaptation is thus required for understanding poetry, and cannot be achieved from general instruction alone. A second issue is that of modality. A normal system of interpreting poems does not go further than the text; a picture of the scene described in a verse can help a poet-interpret who doesn't know the cultural background of the poem. Usually, the text-to-image generation is not associated with the interpretation component of a pipeline.

In this paper, Padyam2Gadyam+ is introduced, which is an end-to-end framework consisting of three modules. In Module 1, using the poem in the Telugu language, and the Telugu meaning of the prose, the English meaning of the poem is retrieved by training small adapter matrices over a frozen and quantized model, Gemma-2-9B-it [4]. A structured instruction

(or prompt) in English is fed into the Gemini API, which in turn outputs a descriptive prompt in text-to-image format. A structured prompt in English is passed to the Gemini API which then outputs a prompt in Text-to-image form. The rendering of the prompt is done with Stable Diffusion 1.5 + LoRA adapter in Module 3. Stable Diffusion is a latent diffusion model, which runs the denoising process in a compressed latent space instead of pixel space [5]. A Flask Web application is used to connect all the modules; once the user submits a poem and its meaning, he/she can get the poem text and the image. The overall architecture is depicted in Fig. 1.

The contributions of this work are: (1) A task-specific QLoRA fine-tuning procedure that maps a Telugu poem and its prose meaning to English prose and only updates 0.58% of the model parameters; (2) a prompt-construction stage to convert the generated meaning to a constrained description of the image; (3) an image-generation stage using Stable Diffusion 1.5 with LoRA; (4) a quantitative comparison of the finetuned model with 3-shot baseline on a fixed test set with bootstrap confidence intervals; and (5) a Flask application for running the three stages as a single workflow.

A summary of the related work is provided in Section II. The problem and the objectives are given in Section III. The framework is explained in Section IV and the experimental setup and implementation is explained in Section V. The results are presented and discussed in Sections VI and VII respectively, the limitations and future work are discussed in Section VIII and finally the paper is concluded in Section IX.

## II. RELATED WORK

### A. Language Models for Indian Languages

Multilingual pretraining has been the main source of language technology for Indian languages. mT5 extends the T5 model to 101 languages by introducing a shared vocabulary [6] while MuRIL incorporates a BERT-style encoder which was trained on the corpus of Indian language and transliterated text [7]. A compact sequence-to-sequence model "IndicBART" was also tested over MT and extreme summarisation tasks and has shown satisfactory performance under low-resource settings with the Indic languages, for which we know that the model is applicable [8]. IndicTrans2 aims to extend the translation to

multilingual languages with all the 22 scheduled languages of India and highlights the need for large parallel corpora and diverse benchmarks and freely available models [9]. The models are mainly trained on prose texts which are from recent times. However, some adaptation is needed to be able to interpret poetry in these models as classical verse differs from such material in vocabulary, and in word order and reference.

### B. Parameter-Efficient Fine-Tuning

Gemma 2 is a family of open decoder-only Transformer models, available in a number of different sizes, including a 9 billion parameter variant [10]. It is costly in terms of memory and time to update all the weights of this size of a model. To lower the cost, LoRA freezes the pretrained weights, and learns a low-rank update for a small number of weight matrices [11]. QLoRA goes an extra mile: the frozen model is represented in 4-bit precision and adapters are trained with the quantized model weights, which is enough to fine-tune large models on a single GPU [4]. This can be used for Telugu poetry as there are only a handful of hundred examples of the task-specific corpus and fine-tuning would be expensive and may lead to overfit.

### C. Prompt Construction and Text-to-Image Generation

There is a family of multimodal models that process text, images, audio and video [12] that can be used to rewrite a text of prose as a visual text. There is a measurable effect of the wording of the text-to-image prompts on the output. To get a grasp of prompt structure, Liu and Chilton conducted an experimental study, and developed rules focusing on the subject and style keywords [13]. To learn how to convert a user's input to a prompt that is more suitable for the image model, Hao et al. proposed Promptist [14].

For the generation side, latent diffusion models compress an image into a lower dimensional latent space and diffusion in the latent space, coupled with the cross-attention of textual information, reduced the computation cost from pixel space to latent space diffusion [5]. This is a design that's followed by Stable Diffusion. Pretrained DPs can be customized with a few simple additions, like Textual Inversion [15] for introducing a new concept from a small number of pictures, and the LoRA family of adapters, which have

been studied and tested systematically for the purpose of customization in the case of Stable Diffusion [16].

#### D. Research Gap

The research on Indian-language generation, converting large models to Indian languages, and text-to-image synthesis has been conducted separately in the past. To the best of our knowledge, there is no such system which does the interpretation of Telugu poems along with prompt construction and generation of image. Padyam2Gadyam+ is a solution to this.

### III. PROBLEM STATEMENT AND OBJECTIVES

The goal of the system is to be presented with a Telugu poem, (a) a literal translation of the prose meaning of the poem into English, (b) an image that captures the essence of the poem or central idea. There are two limitations to the design. First, it has a relatively small number of elements in the corpus (599 triples) and the language model must be adapted without overfitting and with a limited amount of GPU memory. Second,

in order to pass from the concrete visual descriptions to the English meaning there has to be some intermediate step; the English meaning is a narrative or didactic prose, but the text-to-image model is not.

The goals are: (1) to fine tune Gemma-2-9B-it to generate meaning from an English poem–meaning pair; (2) to compare with the few shot version of the same base model using standard reference-based metrics; (3) to include the prompt that contains the facts of the meaning and add a step that appends no unsupported detail; (4) to generate the image from that prompt using Stable Diffusion 1.5 and LoRA; and (5) to integrate the steps into a web application.

### IV. PROPOSED FRAMEWORK

The input of Padyam2Gadyam+ is fed into three consecutive stages as shown in Fig.1. The output of each module forms the input for the next module and modules can be developed, replaced and evaluated independently.

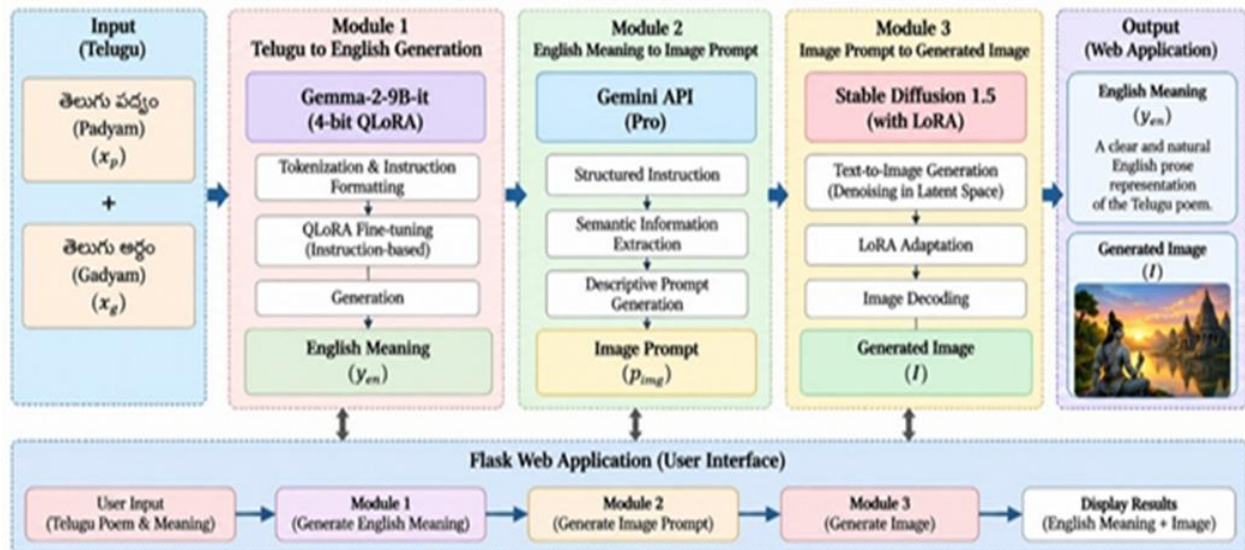


Fig. 1. Architecture of Padyam2Gadyam+: Telugu poem and prose meaning as input, Module 1 (Gemma-2-9B-it with 4-bit QLoRA), Module 2 (Gemini API prompt construction), Module 3 (Stable Diffusion 1.5 with LoRA) and the Flask web application that connects them.

#### A. Problem Formulation

Suppose the Telugu poem content is  $x_p$  and Telugu prose content is  $x_g$  in the same. The framework produces an English meaning  $y_{en}$ , an image prompt  $p_{img}$  and an image  $I$  through three mappings:

$$y_{en} = f_{\theta}(x_p, x_g) \quad (1)$$

$$p_{img} = g_{\varphi}(y_{en}) \quad (2)$$

$$I = h_{\psi}(p_{img}) \quad (3)$$

where  $f_{\theta}$  is the fine-tuned language model,  $g_{\varphi}$  is the prompt-construction step and  $h_{\psi}$  is the image generator. The whole pipeline is the composition  $F = h_{\psi} \circ g_{\varphi} \circ f_{\theta}$ .

### B. Module 1: Telugu Poem to English Meaning

The poem is formulated within a strict rendition of instructions, and requires a simple translation of the poem into English prose. Training and inference are performed using the same template. Suppose that the input to the file is  $xm_{list} = [xp; xg]$  and the English output of  $n$  tokens is  $yen = (y_1, \dots, y_n)$ . The model predicts the target autoregressively, given the input parameters  $x$ , and the target tokens  $y < i$ . The negative log-likelihood of the reference meaning in the model distribution  $p_\theta$ :

$$L(\theta) = -\sum_{i=1}^n \log p_\theta(y_i | y_{<i}, x) \quad (4)$$

The loss calculation is made using only the English target. The ignore index is assigned to the tokens that are used for the instruction and used for the Telugu input, considering the prediction but not providing a gradient. This means that the emphasis is on the English product and not adaptation of the language.

The input precisions are 4 bits NF4 (with 2 bits of double quantization) and the computation is FP16, the weights are fixed during the pretraining stage with 4 bits precision. A low rank update is provided for each adapted projection and the quantized weight matrix  $Wq$  is updated:

$$W' = W_q + \frac{\alpha}{r} BA \quad (5)$$

where  $B \in \mathbb{R}^{d \times r}$  and  $A \in \mathbb{R}^{r \times k}$  are the trainable matrices,  $r$  is the rank and  $\alpha$  is the scaling constant. The update is scaled by 2 ( $r = 16$  and  $\alpha = 32$ ). There are four attention projections  $q\_proj, k\_proj, v\_proj, o\_proj$  and three feed-forward projections  $gate\_proj, up\_proj, down\_proj$  per decoder layer, each with an adapter. This projection has  $r(d + k)$  parameters,  $k$  input and  $d$  output dimensions. This is equivalent to 1,286,144 parameters per layer and for the 42 layers 54,018,048, which is 0.58% of the 9,295,724,032 parameters of the adapted model.

The adapted model is used at inference to produce the meaning in English, in a deterministic decoding fashion, with a maximum of 400 new tokens. This text generating output is then sent to Module 2.

### C. Module 2: English Meaning to Image Prompt

The meaning in English is written for a reader and will typically contain a number of events, names and abstract statements. In a text-to-image model, there is a short description as one scene. This is accomplished

using a structured instruction in Module 2 by calling the Gemini API (Pro).

The instruction is based on the following rules: (1) choose one point from the meaning; (2) include at most three subjects, indicate one for each subject where it is located in the picture; (3) keep the description brief; (4) preserve facts of the meaning and its cultural or mythological context using general language when the meaning does not provide information, do not add anything not substantiated; (5) avoid art-style terms (this aspect is treated separately in Module 3); (6) for verses containing moral or didactic content, limit the number of subjects to one or two. This leads to a new prompt ( $pimg$ ) being passed to Module 3.

### D. Module 3: Image Prompt to Generated Image

This is a LoRA Adapter for using Stable Diffusion 1.5. The first thing that was to be encoded by the text encoder  $\tau$  is the conditioning  $c = \tau(pimg)$ . The generation starts from Gaussian noise in the latent space and gradually filters out the noise. In the denoising network, the network tries to estimate the noise  $\varepsilon\psi = \varepsilon\psi(z_t, t, c)$  given the current latent  $z_t$ , the step  $t$  and the conditioning  $c$ , and one reverse step is

$$z_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left( z_t - \frac{1-\alpha_t}{\sqrt{1-\alpha_t}} \varepsilon\psi \right) + \sigma_t \varepsilon \quad (6)$$

The noise-schedule coefficients are  $\alpha_t$  and  $\sigma_t$ ; the noise scale of the step is given by  $\sigma_t$ , and noise is  $\varepsilon$  is a standard Gaussian distribution. The last step of the decoder  $D$  of the autoencoder projects the final latent  $z_0$  onto the image:

$$I = D(z_0) \quad (7)$$

The LoRA adapter is the same as low rank updates to the denoising network in (5). The output prompt from Module 2 will be free from any style terms, and the adapter will be added here so that the output will appear as desired.

### E. End-to-End Workflow

For every request, the Flask application goes through the following steps:

1. The user provides a Telugu poem and its meaning (Telugu prose) in a web form, this is called Input.
2. Formatting: The two texts are loaded into the instructions template used in training, and tokenized.

3. Developed Gemma-2-9B-it: English meaning generation (Module 1).
4. Prompt Construction: yen sent to the Gemini API and pimg (module 2) received.
5. I is created by image synthesis with Stable Diffusion 1.5 + LoRA,  $p_{img}$  is added.
6. The image I is the one I synthesized with the combination of Stable Diffusion 1.5, LoRA and pimg while the image pimg is the image I fed into the Stable Diffusion model.
7. The interface displays the generated image and the English meaning at the same time, which is named Output.

## V. IMPLEMENTATION AND EXPERIMENTAL SETUP

### A. Dataset

A total of 599 triples are used for training Module 1 using Padyam2Gadyam corpus. Every triple is comprised of a Telugu poem, Telugu prose translation and English prose translation, which is used as a reference.

The training set, the validation set and the test set are randomly split into three subsets of the corpus with random seed number 42 as displayed in Table I. For all tests, the same test set is used. The length of the sequence the maximum length that can be tokenized is 1,024 tokens.

TABLE I. DATASET SPLIT

| Subset     | Triples | Share |
|------------|---------|-------|
| Training   | 549     | 91.7% |
| Validation | 30      | 5.0%  |
| Test       | 20      | 3.3%  |
| Total      | 599     | 100%  |

### B. Training Configuration

The configuration of Module 1 is given in Table II. Training epochs = 2, per device batch size = 1, gradient accumulation = 8, thus effective batch size = 8. For the training set the 549 triples then result in 69 updates to the parameters for each of the epochs and 138 overalls. The paged 8-bit AdamW optimizer is used with a learning rate of  $5 \times 10^{-5}$ , a cosine schedule, a weight decay of 0.01 and warm-up over the first 5% of the updates. To save memory, gradient checkpointing is turned on.

TABLE II. MODULE 1 TRAINING CONFIGURATION

| Parameter                   | Setting                                                       |
|-----------------------------|---------------------------------------------------------------|
| Base model                  | Gemma-2-9B-it                                                 |
| Fine-tuning method          | 4-bit QLoRA                                                   |
| Quantization                | NF4 with double quantization                                  |
| Compute precision           | FP16                                                          |
| LoRA rank / alpha / dropout | 16 / 32 / 0.05                                                |
| LoRA target layers          | q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj |
| Trainable parameters        | 54,018,048 (0.58%)                                            |
| Maximum sequence length     | 1,024 tokens                                                  |
| Epochs                      | 2                                                             |
| Batch size                  | 1 (effective 8 with accumulation)                             |
| Optimizer                   | Paged AdamW (8-bit)                                           |
| Learning rate               | $5 \times 10^{-5}$                                            |
| Scheduler                   | Cosine                                                        |
| Weight decay                | 0.01                                                          |
| Warm-up                     | 5% of the updates                                             |
| Optimization steps          | 138                                                           |
| Random seed                 | 42                                                            |

### C. Baseline

Gemma-2-9B-it is evaluated on the same 20 test triples for the same three-shot setting, but without the use of an adapter. There are three training examples as models in the prompt, followed by the test poem and its meaning. The base system and the test inputs are the same in both systems. The baseline is based on in context examples, while the proposed is based on the trained adapters.

### D. Evaluation of Module 1

English meanings are generated and then compared to the reference meanings with BLEU [17], METEOR [18] and ROUGE-L [19]. The BLEU score is based on n-gram precision score, the METEOR score is based on the word level matching score with stem and synonym level matching, and the ROUGE-L score is based on the longest common subsequence. The scores are all scaled to 0–1. To estimate the 95% confidence intervals, 200 resamples of the test set are used, thus inducing uncertainty due to the small test set size.

### E. Evaluation Protocol for Modules 2 and 3

The evaluation of Modules 2 and 3 is a qualitative evaluation carried out by looking at representative outputs (Section VI-B). The following two modules (Themes 3 and 4) do not have numerical scores. For the sake of completeness, they are listed here, along with their quantitative measures. Prompt fidelity is the cosine similar between the embedding of the English meaning and the embedding of the prompt,

$$S_{sem} = \frac{e_m \cdot e_p}{\|e_m\| \|e_p\|} \quad (8)$$

and by how many of the key elements of the meaning (action, subject, setting and objects) are repeated in the prompt,

$$C_{attr} = \frac{|A \cap A_p|}{|A|} \times 100 \quad (9)$$

When A is a set of attributes in the meaning and A<sub>p</sub> is a set of attributes in the prompt. The cosine similarity of the text embedding of the prompt and the image embedding of the output is used to measure text-image alignment, called CLIP [20]:

$$S_{CLIP} = \frac{E_T(p_{img}) \cdot E_I(I)}{\|E_T(p_{img})\| \|E_I(I)\|} \quad (10)$$

Here E<sub>T</sub> and E<sub>I</sub> are the CLIP text and image encoders. The higher the number, the more similar the prompt is to the image. The measures applied to this, along with subjective ratings from humans on the semantic alignment, visual quality and cultural relevance are pending for the next phase of work.

### F. Web Application

These three modules are provided by a Flask application. It reads the poem and its meaning, loads the modules one after another and prints them on one page, as well as the meaning in English and the image. A model can be replaced without affecting the rest of the application as each module only communicates with the next module through text.

## VI. RESULTS

### A. English Meaning Generation

The 20 test triples in Table III and Fig. 2 are compared with the fine-tuned model. The fine-tuned model reaches BLEU 0.3049, METEOR 0.5850 and ROUGE-L 0.5132, and the baseline reaches 0.1633, 0.4586 and 0.4236. The absolute gains are 0.1416, 0.1264 and 0.0896, which correspond to relative gains of 86.7%, 27.6% and 21.2%.

TABLE III. MODULE 1 RESULTS ON THE FIXED TEST SET (20 TRIPLES)

| Model Measure /             | BLEU        | METEOR      | ROUGE-L     |
|-----------------------------|-------------|-------------|-------------|
| Gemma-2-9B-it (three-shot)  | 0.1633      | 0.4586      | 0.4236      |
| Gemma-2-9B-it (4-bit QLoRA) | 0.3049      | 0.5850      | 0.5132      |
| Absolute gain               | 0.1416      | 0.1264      | 0.0896      |
| Relative gain               | 86.7%       | 27.6%       | 21.2%       |
| 95% CI, QLoRA model         | 0.254–0.354 | 0.528–0.635 | 0.463–0.561 |

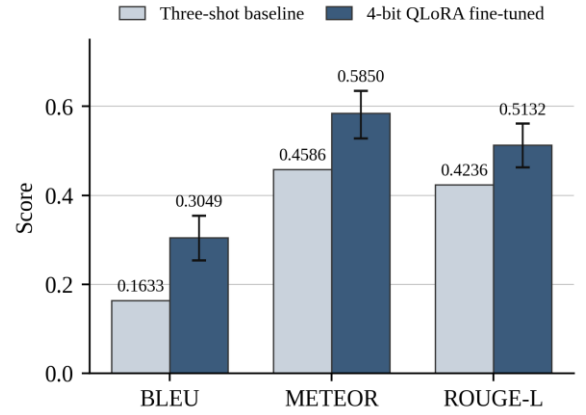


Fig. 2. BLEU, METEOR and ROUGE-L of the three-shot baseline and the QLoRA fine-tuned model on the test set. Error bars show the bootstrap 95% confidence intervals of the fine-tuned model.

The bootstrap 95% confidence intervals of the fine-tuned model are [0.254, 0.354] for BLEU, [0.528, 0.635] for METEOR and [0.463, 0.561] for ROUGE-L. For each metric the baseline is in the bottom part of the interval. In the paired comparison measurement, the same test items were used and an improvement was found in all three measurements, in favor of the fine-tuned model.

In the fine-tuned model, the entities, actions, places and relations mentioned in the Telugu prose meaning are preserved in the outputs generated by the model. The meanings of the examples of Fig. 3 have been preserved in their names of mythological characters (here Trinavarta, Putana and Bana) and events associated with them, and are presented in readable English prose.

### B. Prompt and Image Outputs

There are three full runs of the pipeline in Fig. 3: the English meaning in Module 1; the prompt in Module 2; and the image in Module 3.

It is a salutation acknowledging several deeds of Krishna – its first feature. Don't attempt to show them all in the prompt. It has a village with cows, a peacock feather and pots of butter in it, and it selects a village with these features. The second one is related to the protection of cowherds and gopikas from the storm, the central figure is standing in the middle while the group is under dark clouds. Here there are two deviations that can be seen. The meaning goes before the discus, and the discus follows the prompt and the meaning is to a different episode, so two events are put together in one scene. The third example is a didactic, yoga and peace of mind verse. No story elements;

prompt becomes a sage in an ashram in the woods by a river. The picture is representative of this.

Three observations follow. Firstly, the prompts are brief and specific and maintain the cultural context of the meaning. Secondly, the images include the elements that are mentioned in the prompts and are repeated with the subject matter in the place mentioned in the prompts.

Third, the selection of one moment can lose some of the meaning of verses that include a sequence of events: and as in the second example, different parts of different events can be combined. The observations are of a subjective and qualitative nature. They are based on a limited number of examples, and are not intended to supersede the measures set out in Section V.

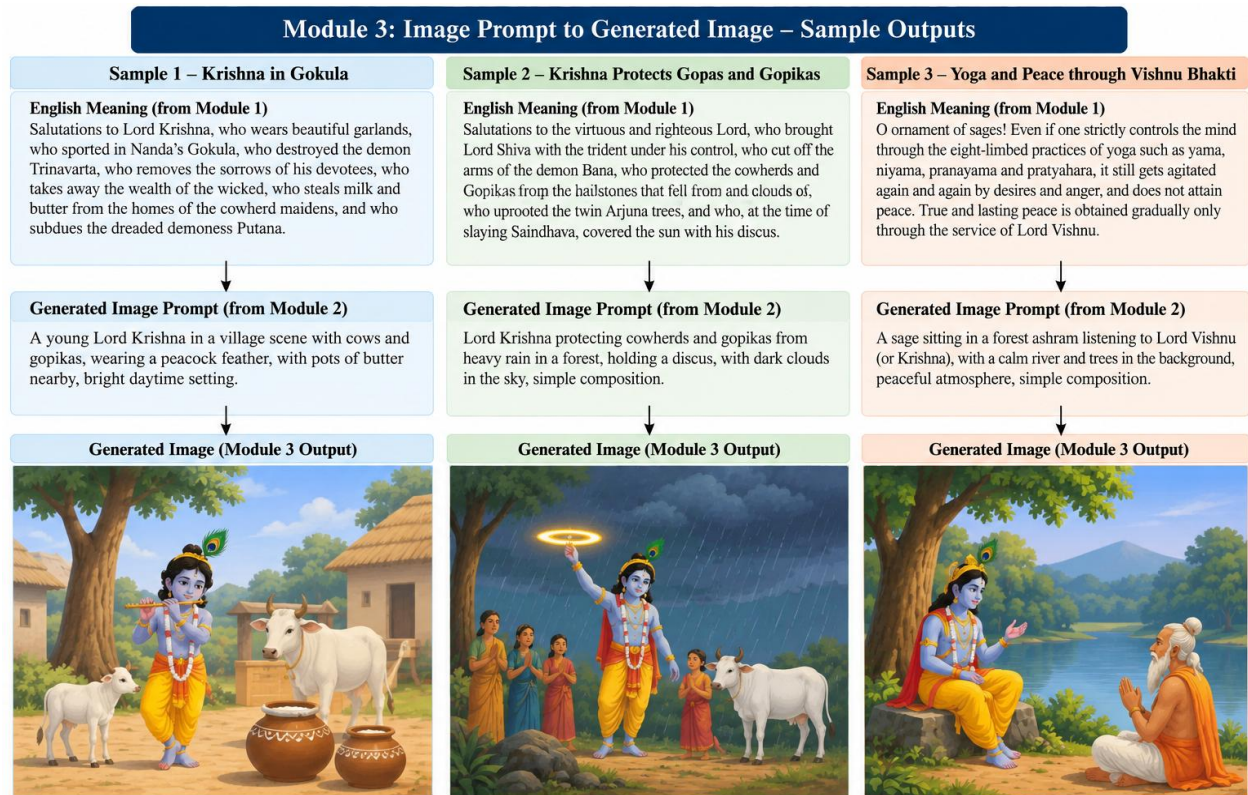


Fig. 3. Three sample runs of the pipeline: English meaning generated by Module 1, image prompt produced by Module 2 and image generated by Module 3C. Integrated Application

With the Flask application the three modules run as a single request. The user provides the poem and how it is understood and receives the English meaning and the image without having to go through the intermediate prompt.

## VII. DISCUSSION

The comparison in Table III focuses on the results of changing the weights, since the base model and the test inputs and metrics are the same in both rows. The

greatest % increase is in BLEU. The n-gram matches are used to compute rewards; the n-grams which generate meaningful rewards are influenced by the rules of the language of the reference meaning, and the adapters are trained on the 549 meaning triplets. The enhancement of METEOR and ROUGE-L is not so remarkable. Probably it is because the baseline already recovers much of the content (because the meaning of the Telugu prose is also included in the input), and these two measures will award credit for content that is correct, but not the same wording.

This was done by training 54 million parameters in a total of 138 updates on a model stored in 4-bit precisions. In a typical case, the amount of data in literary German is relatively small, with only a few hundred examples, or there is a limited amount of GPU memory available, and it is possible to adapt a 9-billion-parameter model to the specific tasks.

It means practically in the modular design, when there are errors. Only the result of the previous stage is used for each stage, so if the English meaning is missing from one stage, it cannot be added later and if a prompt is inserted that is a combination of two events, the image will be the same. The final image is dictated by the poorest part, so the English meaning was given priority and careful consideration.

The completeness of the rules in Module 2 are sacrificed for clarity. With only one moment and 3 subjects co-takes up a description, the image model can generate the description, and without adding style terms, the prompt is virtually a content description. Well, it's a price, because the image represents one part of the poem. It should be used to deepen the understanding of the verse, rather than to read the entire verse.

#### VIII. LIMITATIONS AND FUTURE WORK

There are a number of limitations associated with this study. (2) Test set size: 20 test triples, which gives wide confidence intervals for the quantitative results. Evaluation scope: Module 2 and 3 were only evaluated with examples and the measures in (8) - (10) have not yet been applied with human ratings. This is because the system is not able to understand a poem by itself, and Module 1 is supposed to be the Telugu prose meaning of the poem too. (4) Metrics: BLEU, METEOR and ROUGE-L: metrics based on word overlap, not on the meaning. (5) External dependency:

Module 2 depends on a hosted API, which means that the behaviour of Module 2 may vary with the model version of the hosted API.

The corpus will be extended to other periods and types of poems, and to semantic and human evaluation of all three modules for the prose meaning module, and to other language and diffusion models and other Indian languages.

#### IX. CONCLUSION

In this paper, we proposed the Padyam2Gadyam+ system that converts Telugu poem and its prose translation to its English translation and illustrative image. It was fine-tuned with 0.58% of its parameters using 549 triples with 4-bit QLoRA. The adapted model achieved BLEU 0.3049, METEOR 0.5850 and ROUGE-L 0.5132 on the fixed test set; all of these scores were higher than the three-shot baseline of the same model, and were higher than the lower bounds of the bootstrap confidence intervals. The concept is then mapped to a structured description (as the instructions in Gemini) and then generated using Stable Diffusion 1.5 and LoRA. A Flask app makes the stages into a workflow.

It shows that a small, well-designed dataset is enough to fine-tune a large language model to understand poetry, and that the specialized model can be used to produce images when an explicit text generation prompt construction step is given. The quantitative evaluation of the prompt and image stages should then be evaluated on a larger test set.

#### DECLARATIONS

Funding: No specific external funding was received for this work.

Conflict of Interest: The authors declare no conflict of interest.

Data and Code Availability: The dataset split, training configuration and source code are available from the corresponding author on reasonable request.

Use of Generative AI: Gemma 2, the Gemini API and Stable Diffusion are components of the proposed system and are described in the paper. An AI writing assistant was used to improve the language and

organization of the manuscript. The authors reviewed and verified all content and take full responsibility for it.

## REFERENCES

- [1] P. Dasari, A. Chelpuri, N. Vuppala, M. Marreddy, P. Krishnamurthy, and R. Mamidi, "Transformer-based context aware morphological analyzer for Telugu," in *Proc. Workshop on Speech and Language Technologies for Dravidian Languages (DravidianLangTech)*, 2023.
- [2] S. R. R. Mangamuru, S. P. Karnati, B. K. Sajja, D. Phogat, and P. B., "Enhancing Telugu part-of-speech tagging with deep sequential models and multilingual embeddings," in *Proc. Int. Conf. Natural Language Processing (ICON)*, 2023. [Online]. Available: <https://aclanthology.org/2023.icon-1.77/>
- [3] C. Kranti and S. Vajjala, "MetricalARGS: Studying metrical poetry with LLMs," in *Proc. 8th Workshop on Indian Language Data: Resources and Evaluation*, May 2026, pp. 1–13, doi: 10.63317/4o3nvxw6qws0.
- [4] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "QLoRA: Efficient finetuning of quantized LLMs," in *Adv. Neural Inf. Process. Syst.*, vol. 36, 2023. [Online]. Available: <https://arxiv.org/abs/2305.14314>
- [5] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2022, pp. 10684–10695.
- [6] L. Xue *et al.*, "mT5: A massively multilingual pre-trained text-to-text transformer," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Lang. Technol. (NAACL-HLT)*, 2021, pp. 483–498, doi: 10.18653/v1/2021.naacl-main.41.
- [7] S. Khanuja *et al.*, "MuRIL: Multilingual representations for Indian languages," arXiv:2103.10730, 2021.
- [8] R. Dabre, H. Shrotriya, A. Kunchukuttan, R. Puduppully, M. M. Khapra, and P. Kumar, "IndicBART: A pre-trained model for Indic natural language generation," in *Findings of the Assoc. Comput. Linguistics: ACL 2022*, 2022, pp. 1849–1863, doi: 10.18653/v1/2022.findings-acl.145.
- [9] J. Gala *et al.*, "IndicTrans2: Towards high-quality and accessible machine translation models for all 22 scheduled Indian languages," *Trans. Mach. Learn. Res.*, 2023. [Online]. Available: <https://arxiv.org/abs/2305.16307>
- [10] Gemma Team, "Gemma 2: Improving open language models at a practical size," arXiv:2408.00118, 2024.
- [11] E. J. Hu *et al.*, "LoRA: Low-rank adaptation of large language models," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2022. [Online]. Available: <https://arxiv.org/abs/2106.09685>
- [12] Gemini Team, "Gemini: A family of highly capable multimodal models," arXiv:2312.11805, 2023.
- [13] V. Liu and L. B. Chilton, "Design guidelines for prompt engineering text-to-image generative models," in *Proc. CHI Conf. Human Factors Comput. Syst.*, 2022. [Online]. Available: <https://arxiv.org/abs/2109.06977>
- [14] Y. Hao, Z. Chi, L. Dong, and F. Wei, "Optimizing prompts for text-to-image generation," in *Adv. Neural Inf. Process. Syst.*, vol. 36, 2023. [Online]. Available: <https://arxiv.org/abs/2212.09611>
- [15] R. Gal *et al.*, "An image is worth one word: Personalizing text-to-image generation using textual inversion," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2023. [Online]. Available: <https://arxiv.org/abs/2208.01618>
- [16] S.-Y. Yeh, Y.-G. Hsieh, Z. Gao, B. B. W. Yang, G. Oh, and Y. Gong, "Navigating text-to-image customization: From LyCORIS fine-tuning to model evaluation," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2024. [Online]. Available: <https://arxiv.org/abs/2309.14859>
- [17] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: A method for automatic evaluation of machine translation," in *Proc. 40th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2002, pp. 311–318.
- [18] S. Banerjee and A. Lavie, "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments," in *Proc. ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, 2005, pp. 65–72.

- [19] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Proc. Workshop on Text Summarization Branches Out*, 2004, pp. 74–81.
- [20] A. Radford *et al.*, "Learning transferable visual models from natural language supervision," in *Proc. 38th Int. Conf. Mach. Learn. (ICML)*, 2021, pp. 8748–8763.