AI Storybook Creator Agent

Team Lead:

T.Sandhya Vara Lakshmi [22B21A4307]

Team Members:

K.Pavani [22B21A4316]

K.Murali Rama Raju *[22B21A4385]*

L.Akash [22B21A4377]

P.Purna Satya Naga Pavan Sai [22B21A4353]

Department: *CSE & Artificial Intelligence*

Institution: *Kakinada Institute Of Engineering & Technology*

Academic Year: *2025–26*

Abstract

This project develops an automated storytelling system that converts a user-provided prompt into a complete narrative. It uses preprocessing techniques like cleaning and tokenization to prepare the input for story generation.

The story is generated using a sequence-to-sequence model or transformer model with attention, ensuring coherent and meaningful content. Simultaneously, a diffusion model creates relevant images that match the story's context.

Finally, the system combines the generated story and images into a single exportable PDF. This approach provides an interactive and visually engaging storytelling experience, making content creation faster and more appealing.

Introduction

Storytelling has been a fundamental part of human communication, education, and entertainment. With advancements in Artificial Intelligence (AI), computers are now capable of generating human-like stories and artwork. However, combining story generation and image generation in a single system remains a challenging task.

Importance of the Problem

Traditional storytelling requires manual effort in both writing and illustration. For individuals without artistic or writing expertise, creating engaging stories with visuals can be challenging. By automating story and image generation, this project lowers the barrier to creativity, offering applications in entertainment, education, and creative industries.

Problem Statement

"The project aims to Build a system that takes a user prompt, generates a coherent story, creates matching images using a diffusion model, and combines them into a single exportable PDF, automating the storytelling process with visuals."

Objectives:

1. Build an AI model that takes a user prompt and generates a coherent story.
2. Train the story generator using the WritingPrompts dataset.
3. Use a diffusion model to generate an image corresponding to the generated story.
4. Integrate both components into a single pipeline for end-to-end story + image generation.

Proposed Methodology

The Proposed methodology by using pretrained models involves following steps:

1. **Input Acquisition o**
    a. User enters a short text prompt (e.g., "A robot exploring a magical forest").
2. **Story Generation (NLP Module) o**
    a. Use transformer-based models such as GPT-2 (via Hugging Face transformers) to generate a short narrative.
    b. Apply basic text processing to ensure coherence and readability.
3. **Image Generation (Diffusion Module) o**
    a. Use a pretrained Stable Diffusion model (runwayml/stable diffusion-v1-5) from Hugging Face's diffusers library.
    b. Convert key story segments into image prompts. o Generate and save images corresponding to the story.
4. **Integration & Output o**
    a. Save story as a text file or PDF (using FPDF).
    b. Optionally embed images within the story PDF for a complete illustrated storybook experience.

The proposed methodology by building scratch model involves the following steps:

1. **Data Collection:**

- Use the Hugging Face WritingPrompts dataset or custom prompt-story pairs.
- Each entry has:

    o prompt → input

    o story → target output

2. **Preprocessing:**

- The text is cleaned and split into words or tokens.
- Words are converted into numbers (token IDs).
- Sequences are padded to the same length, and positional information is added so the model knows the order of words.

3. **Model Architecture**:

- Token Embeddings: turn token IDs into vectors.
- Positional Embeddings: add word order info.
- Stacked Transformer Blocks:
  - Self-Attention → finds relationships between words.
  - Feed-Forward Network → processes features.
  - Residual + Normalization → keeps training stable.
- Output Layer: softmax predicts the probability of the next token.

4. **Training:**

- Teacher forcing: use the correct previous word during training.
- Loss function: Cross-Entropy Loss.
- Optimizer: Adam/AdamW (adjust weights).
- Train for several epochs in mini-batches.
- Use gradient clipping to prevent unstable training.

5. **Validation:**

- After each epoch, evaluate on validation set.
- Track loss and perplexity (exp(loss)).
- Save the best model based on validation results.

6. **Testing:**

- Perplexity on test set (lower is better):
  - perplexity = exp(test_loss_per_token)
- Automated metrics (use with caution for creative generation):

    - BLEU / ROUGE: measure n-gram overlap between generated story and reference. Useful for relevance but not creativity.
    - Distinct-n (diversity): fraction of unique n-grams (distinct-1, distinct-2) to detect repetitions.
- Human evaluation:
    - Rate coherence, fluency, relevance to prompt, creativity on sample outputs.
- Ad hoc tests:
    - Generate stories for a fixed set of prompts (held-out). Inspect for hallucination, repetition, or inconsistency across paragraphs.
    - Safety / content checks (if deployed): filter toxic content.

7. Story Generation:

- Encode user prompt into token IDs and generate tokens autoregressively.
- For long stories, use chunked generation with last tokens as context.
- Apply sampling techniques (top-k or nucleus/top-p) for creativity and diversity.

8.Post-Processing:

- Convert generated token IDs back to words.
- Format text into paragraphs for readability.

9. Generation:

- Take a user prompt.
- Output a story.

10. Image Generation:

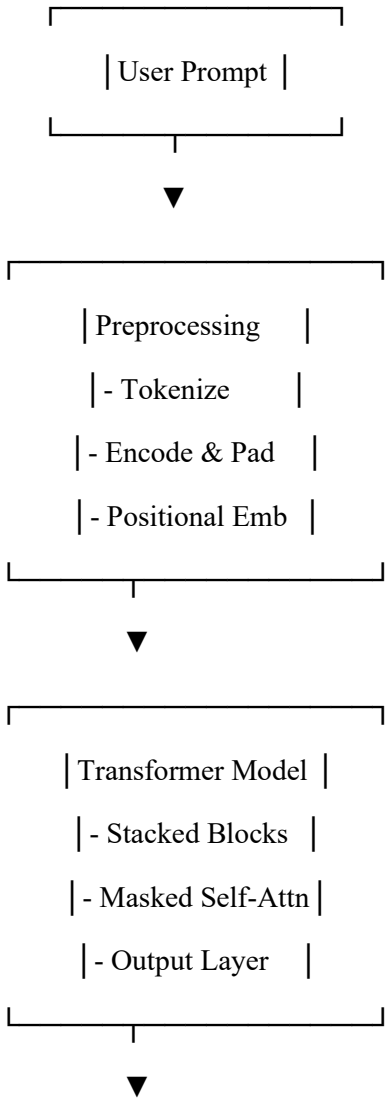
- Use a diffusion model (Stable Diffusion / Hugging Face Diffusers).
- Instead of separate user input, you can feed the first 1–2 sentences of the generated story into the diffusion model.

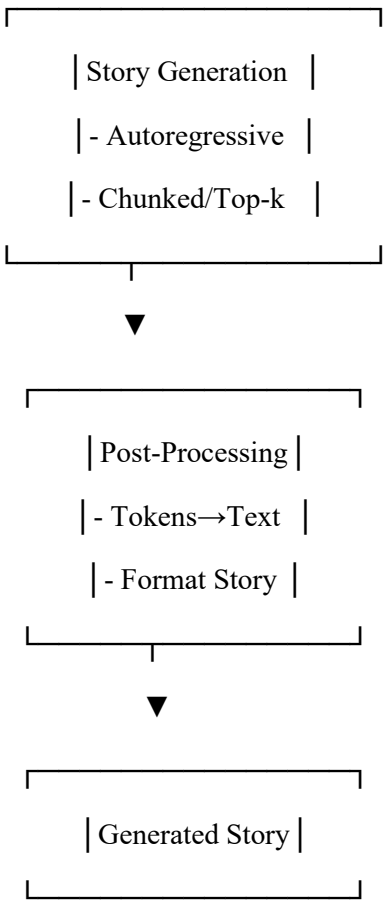
- That way, images match the generated story.

11. Integration:

- User enters: "A magical forest with hidden creatures"
- Model generates story.
- Diffusion model creates matching image.
- Output: Story + Image.

Block Diagram





1. **Expected Working**

2. The user provides a short prompt (e.g., *"A magical forest with hidden creatures"*).
3. The story generator model processes the input and generates a multi-paragraph story.
4. The diffusion model takes part of the story and generates a corresponding image.
5. The system outputs both the story + image for the user.

Conclusion

This project demonstrates the integration of AI text generation and AI image generation into one application. By training on a large-scale dataset like WritingPrompts and using diffusion models for visualization, the system

aims to provide a seamless way of transforming simple prompts into complete creative works. The final product can serve as a storytelling assistant, a learning tool, and an inspiration engine for writers and artists.

References

- Hugging Face Transformers, https://huggingface.co/transformers.
- Diffusers: State-of-the-art diffusion models, https://huggingface.co/docs/diffusers.
- Scikit-learn: Machine Learning in Python, https://scikit-learn.org/.