Multimodal RAG for Story-to-Image and Image-to-Story

DS635 – MACHINE LEARNING SYSTEM ENGINEERING

TEAM MEMBERS

202418041 : PALAK JAIN

202418051 : SHEETAL JAIN

1. Introduction

Traditional machine learning systems usually operate on a single type of data, such as text-only models for language processing or image-only models for computer vision. However, real-world human understanding does not operate in isolated modalities. Humans naturally connect what they see with what they read, hear, or imagine. This project is motivated by the need to bridge this gap between different data modalities.

The core objective of this project is to build a Multimodal Retrieval-Augmented Generation (RAG) system that can reason across text (stories) and images. Specifically, the system supports: first, retrieving the most relevant image when given a detailed story or paragraph; and second, generating a detailed narrative story when given an image.

Rather than training large models from scratch, this project focuses on understanding how pre-trained transformer-based models can be composed together in a pipeline to achieve sophisticated behavior. The emphasis is on interpretability, modularity, and conceptual clarity rather than brute-force scale.

2. Scope and Intentional Focus of the Project

During development, multiple multimodal functionalities were explored, including text-to-image retrieval, image-to-caption retrieval, text-to-story generation, and image-to-story generation. However, this project deliberately focuses on only two tightly coupled tasks.

The first task is Story or Text to Image retrieval. In this task, the system accepts a full narrative story or paragraph as input and retrieves the most semantically relevant image from the dataset. This demonstrates the system's ability to understand long-form textual meaning rather than just keywords.

The second task is Image to Story generation. In this task, the system takes an image as input and generates a detailed, multi-sentence story that describes the scene, actions, emotions, and progression over time. This task demonstrates the integration of retrieval and generation using a RAG pipeline.

3. Dataset Description: Flickr8k (In Depth)

The Flickr8k dataset is a well-known benchmark dataset in the field of multimodal learning, particularly for tasks involving image and text alignment. It consists of approximately 8,000 real-world photographs collected from Flickr, each paired with five independent description written by human annotators.

Each image in the dataset depicts everyday scenes such as people playing, children at home, animals in outdoor environments, and social interactions. The descriptions are not rigid labels but natural language descriptions, which means they vary in wording, structure, and focus. This diversity is extremely valuable because it teaches models that the same visual scene can be described in multiple valid ways.

In this project, the dataset serves two purposes. First, the images are used as candidates for retrieval when a story is provided as input. Second, the description act as grounding context for story generation, ensuring that generated narratives remain faithful to visual content rather than hallucinating unrelated details.

4. High-Level System Architecture

The overall system architecture follows a modular design in which each component has a clearly defined role. This modularity is crucial for understanding, debugging, and extending the system. The architecture can be conceptually divided into three layers: the embedding layer, the retrieval layer, and the generation layer.

4.1 Embedding Layer: CLIP

The embedding layer converts raw inputs such as stories and images into numerical representations known as embeddings. Embeddings capture semantic meaning in a fixed-length vector space, allowing different data modalities to be compared mathematically.

This project uses CLIP (Contrastive Language–Image Pretraining), which maps both text and images into a shared 512-dimensional embedding space. CLIP consists of a text encoder and an image encoder, ensuring that semantically related images and textual descriptions lie close to each other in the vector space. Using CLIP across the system guarantees consistent semantic alignment between stories and images.

4.2 Retrieval Layer: Semantic Similarity Search

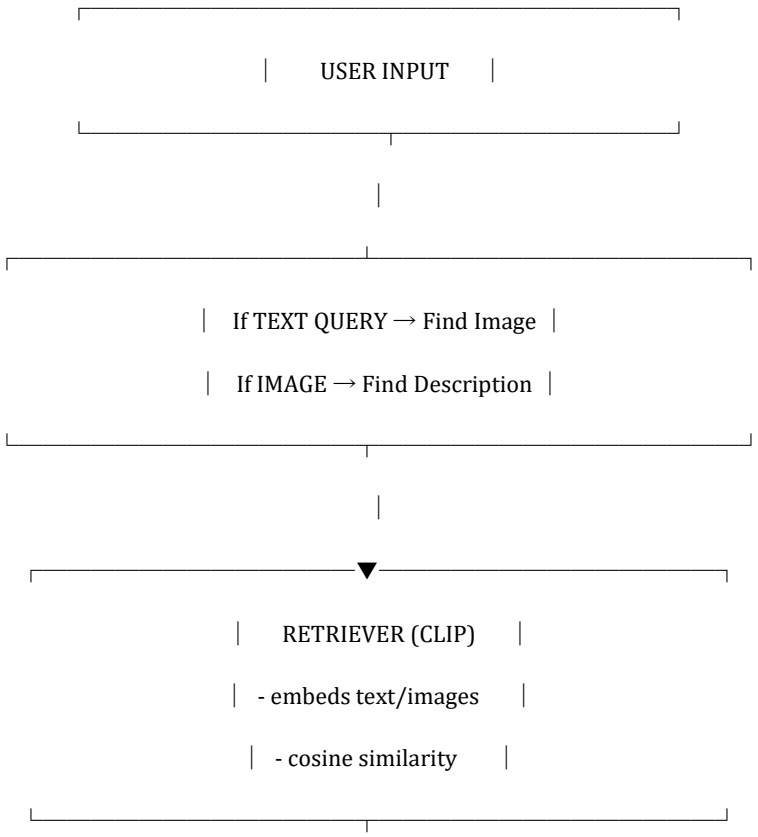
The retrieval layer identifies the most relevant items in the dataset by comparing embeddings. Retrieval in this system is entirely semantic rather than keyword-based. Cosine similarity is used to measure how closely two embeddings align in the vector space.

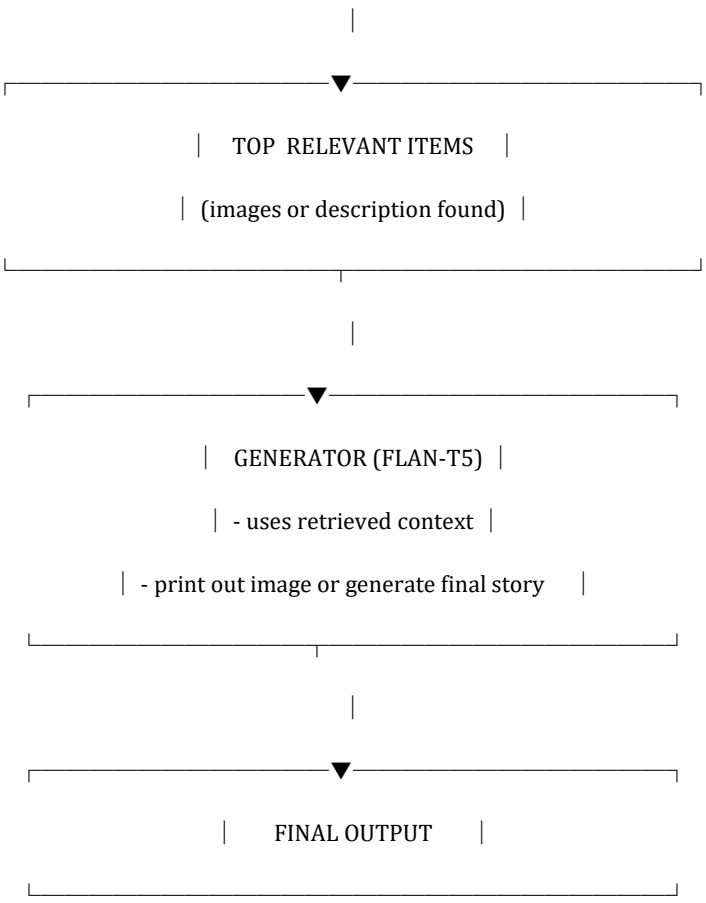
For story-to-image retrieval, the story embedding is compared with precomputed image embeddings to retrieve the most relevant image. For image-to-story generation, the image embedding is compared with caption embeddings to retrieve captions that best describe the image. This retrieval step provides accurate and grounded context for downstream processing.

4.3 Generation Layer: Retrieval-Augmented Story Generation

The generation layer produces natural language stories using a Retrieval-Augmented Generation (RAG) approach. Instead of generating stories directly from images, the system first retrieves relevant captions that act as grounding context.

Story generation is performed using the FLAN-T5 model, guided by a carefully designed prompt. The prompt includes retrieved captions, narrative instructions, few-shot examples, and length constraints. This design ensures that generated stories remain factually consistent with the image while exhibiting narrative structure and coherence. Separating retrieval and generation allows story quality to be improved through prompt engineering without retraining the model.





5. Methodology:

Story to Image Retrieval

The Story/Text to Image retrieval process is designed to allow the system to identify the most visually relevant image when given a full narrative paragraph or story as input. Unlike traditional search systems that rely on keyword matching or manually assigned labels, this approach aims to capture the overall semantic meaning of the story and align it with visual content.

The process begins when the user provides a story or descriptive paragraph. This input text may contain multiple sentences, emotions, actions, and contextual details. The system does not attempt to parse or analyze individual words explicitly. Instead, it treats the entire story as a single semantic unit whose meaning must be embedded into a numerical representation.

The first technical step involves tokenization using CLIP's tokenizer. Tokenization converts the raw text into a sequence of tokens that the model can process. CLIP imposes a strict maximum token length of 77 tokens.

Once tokenized, the story is passed through CLIP's text encoder, which transforms the sequence of tokens into a 512-dimensional embedding vector. This vector acts as a compressed semantic representation of the entire story. Crucially, this embedding is not a bag of keywords; rather, it encodes relationships, actions, and contextual meaning in a way that allows comparison with image embeddings.

In parallel, all images in the dataset have already been processed offline using CLIP's image encoder. Each image is converted into its own 512-dimensional embedding and stored on disk. Precomputing these embeddings is an important efficiency optimization, as it avoids repeatedly encoding images during retrieval and allows the system to scale to larger datasets.

The retrieval step itself is performed using cosine similarity. Cosine similarity measures how closely aligned two vectors are in the embedding space by computing the cosine of the angle between them. A higher cosine similarity score indicates stronger semantic alignment. In this context, it measures how closely the meaning of the input story aligns with the visual content of each image.

The system computes the cosine similarity between the story embedding and every image embedding in the dataset. The image with the highest similarity score is selected as the most relevant result. This image is then presented to the user as the output of the story-to-image retrieval process.

This methodology enables the system to retrieve images based on meaning rather than keywords. For example, even if the story does not explicitly mention words like "couch" or "living room," the system can still retrieve an indoor image of a child playing if the overall narrative context aligns semantically with such scenes. This demonstrates the power of embedding-based retrieval in multimodal systems.

Image to Story Generation using RAG

Image to Story generation is implemented using a Retrieval-Augmented Generation (RAG) framework. The fundamental motivation behind using RAG is to address a key limitation of standalone language models: when generating text without external context, language models may hallucinate details or produce content that is only loosely related to the input.

In a RAG-based approach, generation is explicitly grounded in retrieved information. This grounding ensures that the generated story remains faithful to the visual content of the image rather than relying solely on the internal knowledge of the language model.

The process begins when the user provides an image as input. The image is first processed using CLIP's image encoder, which converts the visual content into a 512-dimensional embedding vector. This vector captures high-level semantic information about the image, such as objects present, actions taking place, and the overall scene.

Once the image embedding is obtained, the system performs a retrieval step to identify relevant textual descriptions. Specifically, all captions associated with images in the dataset are converted into CLIP text embeddings. These caption embeddings represent textual descriptions of visual scenes and are therefore well-suited to act as grounding context.

Using cosine similarity, the image embedding is compared against all caption embeddings. The captions with the highest similarity scores are selected as the most semantically aligned with the input image. These retrieved captions serve as a factual and descriptive summary of what is happening in the image.

The retrieved captions are not directly presented to the user. Instead, they are used internally to construct a prompt for the generation model. This prompt is a carefully designed textual input that combines three critical elements: grounding information, narrative guidance, and structural constraints.

Grounding information comes from the retrieved captions, which ensure that the story is visually consistent with the image. Narrative guidance is provided through explicit instructions that ask the model to write a story rather than a caption. Structural constraints, such as minimum word length and multi-paragraph requirements, are included to prevent short or overly concise outputs.

The generation itself is performed using FLAN-T5, an instruction-tuned language model. FLAN-T5 is particularly suitable for this task because it responds well to explicit instructions and examples provided in the prompt. Although it is a relatively lightweight model, its behavior can be significantly influenced through careful prompt engineering.

During generation, decoding parameters such as minimum length, temperature, and nucleus sampling are applied. These parameters control how conservative or creative the model is, as well as how long the generated output must be. By enforcing a minimum length and encouraging diversity through sampling, the system produces stories that are longer, more expressive, and more narrative in nature.

The final output is a multi-sentence story that describes not only what is visible in the image, but also the implied actions, emotions, and progression of events. Importantly, because the story is grounded in retrieved captions, it maintains a strong connection to the visual content and avoids introducing unrelated or implausible details.

This RAG-based methodology highlights a key principle of modern multimodal AI systems: retrieval enhances generation. By separating the tasks of understanding (retrieval) and expression (generation), the system achieves both relevance and coherence in its outputs.

7. Prompt Engineering and Controlled Story Length

Prompt engineering plays a critical role in the quality of generated stories. Initial experiments resulted in short, caption-like summerizations outputs because the language model interpreted the task conservatively.

To overcome this limitation, multiple techniques were applied simultaneously. These include providing few-shot examples of narrative stories, explicitly instructing the model to write multiple paragraphs, and enforcing a minimum generation length using decoding parameters.

By combining strong narrative prompts with decoding constraints such as minimum length, temperature, and nucleus sampling, the system reliably generates stories of maximum 100 words that exhibit narrative flow.

8. End-to-End Workflow Explanation

A. Story/Text → Image Retrieval Workflow

1. The workflow begins when a user provides a narrative paragraph or story as input through the user interface (app.py). Unlike keyword-based queries, the input can be a full descriptive story.

2. The input story is first processed using the CLIP text tokenizer, which converts the text into tokens suitable for the model. If the story exceeds CLIP's maximum token limit, it is truncated to ensure compatibility.

3. The tokenized story is passed through CLIP's text encoder, which converts the input into a 512-dimensional semantic embedding representing the overall meaning of the story.

4. In parallel, embeddings for all dataset images have already been computed and stored offline using the build_index.py script. These embeddings are generated using CLIP's image encoder and saved to disk to avoid repeated computation.

5. The story embedding is compared against all precomputed image embeddings using cosine similarity, which measures semantic alignment in the embedding space.

6. The image whose embedding has the highest similarity score with the story embedding is retrieved as the most relevant result.

7. The retrieved image path is returned to the user interface (app.py), where the image is loaded and displayed to the user.

This workflow enables semantic image retrieval, allowing the system to retrieve images based on the meaning of a story rather than exact word matches.

B. Image → Story Generation Workflow

1. The workflow begins when a user provides an image path through the user interface (app.py). The user may also control the desired story length using a slider.
2. The input image is processed by CLIP's image encoder, which converts the visual content into a 512-dimensional image embedding that captures the semantic features of the image.
3. Captions associated with the dataset images are preprocessed and stored using the prepare_flickr.py script, which organizes image paths and caption data into a structured corpus file.
4. During generation, caption embeddings are either loaded from disk or computed on the fly and compared with the image embedding using cosine similarity.
5. The captions with the highest similarity scores are retrieved and treated as grounding context that describes the visual content of the input image.
6. These retrieved captions are inserted into a structured prompt along with explicit narrative instructions, few-shot examples, and minimum length constraints.
7. The constructed prompt is passed to the FLAN-T5 language model, which generates a multi-sentence, story-like narrative that reflects both the visual content and the imposed storytelling constraints.
8. The generated story is returned to the user interface (app.py) and displayed alongside the input image.

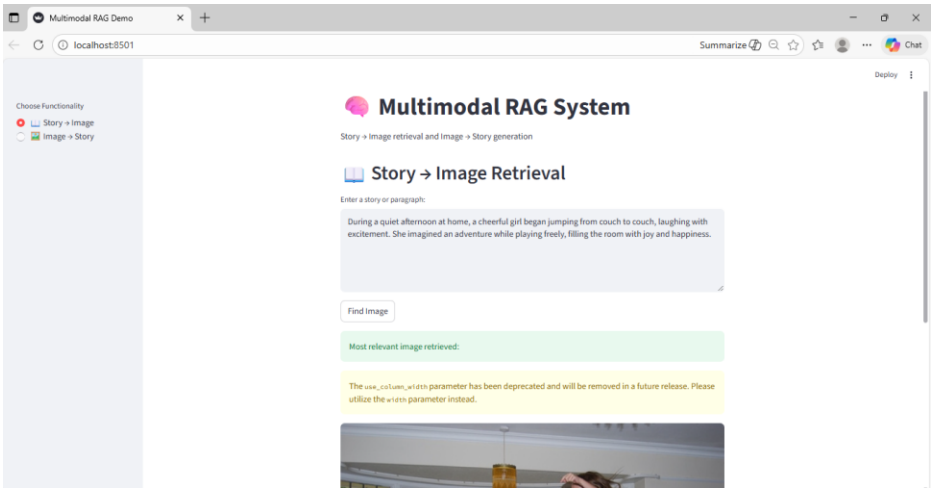
This workflow follows a Retrieval-Augmented Generation (RAG) approach, ensuring that the generated story remains grounded in real visual descriptions rather than relying solely on the language model's internal knowledge.

9. Results and Qualitative Analysis

Experimental results show that the system successfully retrieves semantically appropriate images for long narrative inputs, even when explicit keywords are not present.

Similarly, the Image to Story module produces coherent, grounded stories that reflect the visual content of the input image. The use of retrieval grounding significantly reduces hallucination and improves relevance.

Story to Image :



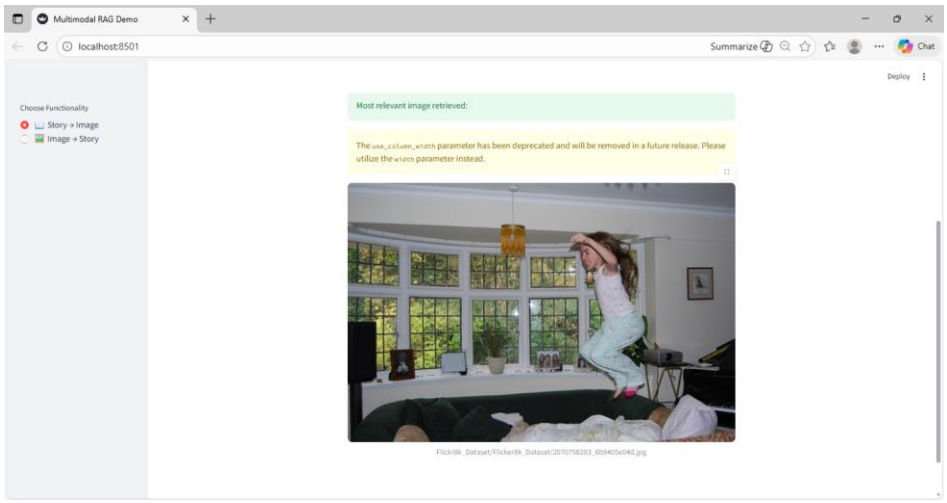
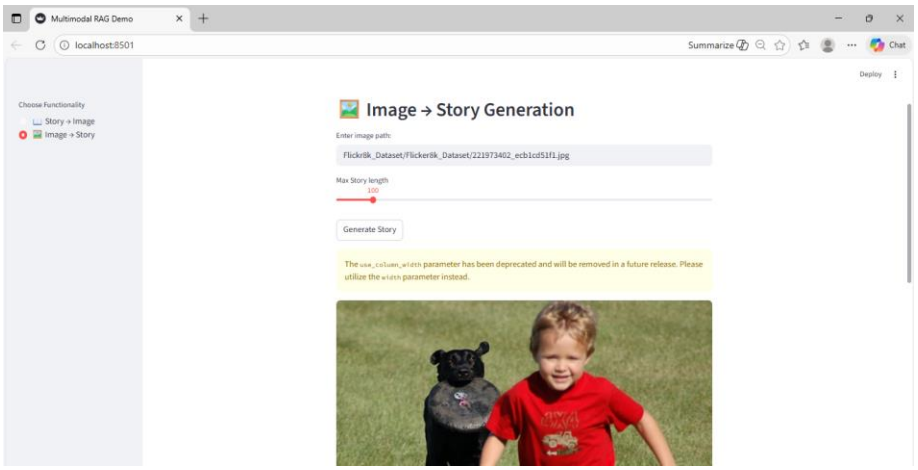
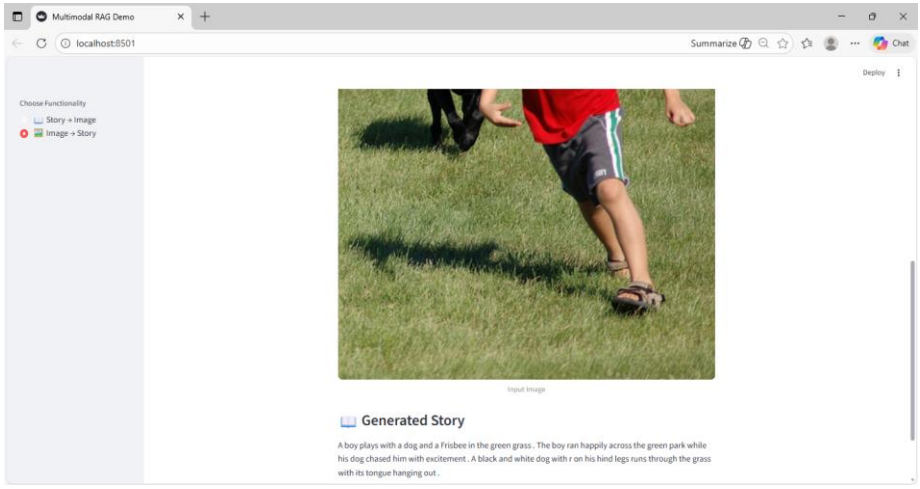


Image to Story:





10. Limitations and Design Trade-offs

While the system performs well within its intended scope, it has inherent limitations. The use of FLAN-T5-base limits the richness of storytelling compared to larger generative models. Additionally, truncation of stories to meet CLIP's token limit may discard some contextual information.

These trade-offs were intentionally accepted to maintain CPU compatibility and conceptual clarity.

11. Conclusion and Learning Outcomes

This project demonstrates how multimodal understanding can be achieved by composing pre-trained models into a Retrieval-Augmented Generation pipeline. By focusing on Story to Image retrieval and Image to Story generation, the system highlights the power of embeddings, similarity search, and prompt-controlled generation.