Visual Story Compression: Towards Automatic Summarization of Comics

Ankur Sinha, M B Ashish, Nitin Rajesh, Nitish Mahapatre

April 22, 2025

Abstract

Comics represent a unique multimodal medium where visual and textual elements are intertwined in a sequential narrative. This project focuses on the task of generating abstractive summaries for comic strips by modeling the interplay between comic panels (images) and dialogue (text). To achieve this, we propose a system that first extracts visual features from the comic panels using a ResNet-50, and retrieves the textual content via Optical Character Recognition (OCR). These multimodal inputs are then processed through a BART-style encoder-decoder transformer model, designed to capture narrative structure and coherence. The resulting system is capable of producing concise, coherent summaries that reflect both the visual and textual storyline of the input comics.

1 Introduction

Multimodal text summarization is an active and challenging area of research within natural language processing. It involves generating meaningful summaries by integrating information from multiple modalities—typically text, images, audio, or video. Unlike unimodal summarization tasks that focus solely on textual input, multimodal summarization demands the ability to interpret and fuse diverse data types into a coherent, unified output.

In *Abstractive Summarization* Rather than extracting and rearranging parts of the input, abstractive methods aim to generate novel sentences that capture the essential meaning of the source content. This requires deeper semantic understanding and generation capabilities, often relying on advanced neural architectures such as encoder-decoder transformer models.

Comics present a particularly interesting use case for multimodal summarization. As a narrative medium, comics intertwine visual elements (comic panels) with textual content (dialogue or narration), arranged in a carefully crafted sequential layout. The storyline emerges not just from the images or the text alone, but from the interaction between the two over time. A successful summarization system must therefore be capable of interpreting the visual progression of panels alongside the corresponding textual dialogue, recognizing narrative structures and temporal flow.

In this project, we explore a model architecture that captures this unique structure. Visual features are extracted from comic panels using a ResNet-50 convolutional neural network, while Optical Character Recognition (OCR) is employed to extract textual

elements. These are then processed using a BART-style encoder-decoder transformer, enabling the generation of abstractive summaries that reflect the multimodal and narrative richness of comic strips.

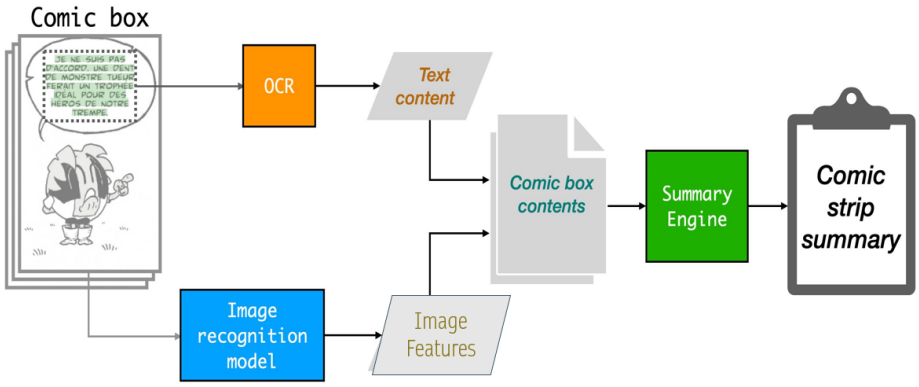

Figure 1: Commic Summarization

2 Related Work

A notable advancement in multimodal understanding of comics is presented by Vivoli et al. [2], who introduce a novel Multimodal Large Language Model (Multimodal-LLM) tailored for the comics text-cloze task. This task involves selecting the appropriate text to complete a comic panel, given its neighboring panels. The authors propose a domain-adapted ResNet-50 visual encoder, fine-tuned using SimCLR, which achieves performance comparable to more complex models but with significantly fewer parameters. Additionally, they enhance model input quality by releasing new OCR annotations for the dataset, leading to further improvements in task performance.

3 Methodology

3.1 Methodology 1

3.1.1 Architecture

. The architecture of our comic summarizer combines several state-of-the-art techniques in computer vision and natural language processing. We leverage a 6x6 BART-style encoder-decoder model to generate abstractive summaries. This architecture is designed to handle multimodal input, where both image and text data must be processed and integrated in a coherent manner.

The input to the model consists of two main components: comic images and text. The comic images are processed through a ResNet-50 convolutional neural network, which acts as a feature extractor. ResNet-50 captures high-level visual features, which are taken from the last layer of the convolution network, enabling the model to understand the content of each panel in terms of both objects and context.

Simultaneously, the textual content from the comic is extracted using Optical Character Recognition (OCR), which converts the dialogue and other textual elements from

the image into machine-readable text. This OCR text is then tokenized using the BART tokenizer, which prepares the text for the next stages of processing.

The extracted visual features from the ResNet-50 and the tokenized OCR text are then embedded together. This fused representation of image and text is passed sequentially to the encoder of the BART-style model. The encoder processes the combined representation, learning to capture the contextual relationships between the sequential panels and dialogue. Finally, the decoder generates the abstractive summary by attending to both the visual and textual features, producing a concise and coherent output.

This approach allows the model to effectively utilize both modalities in generating summaries that are contextually aware of both the narrative and visual elements of the comic.

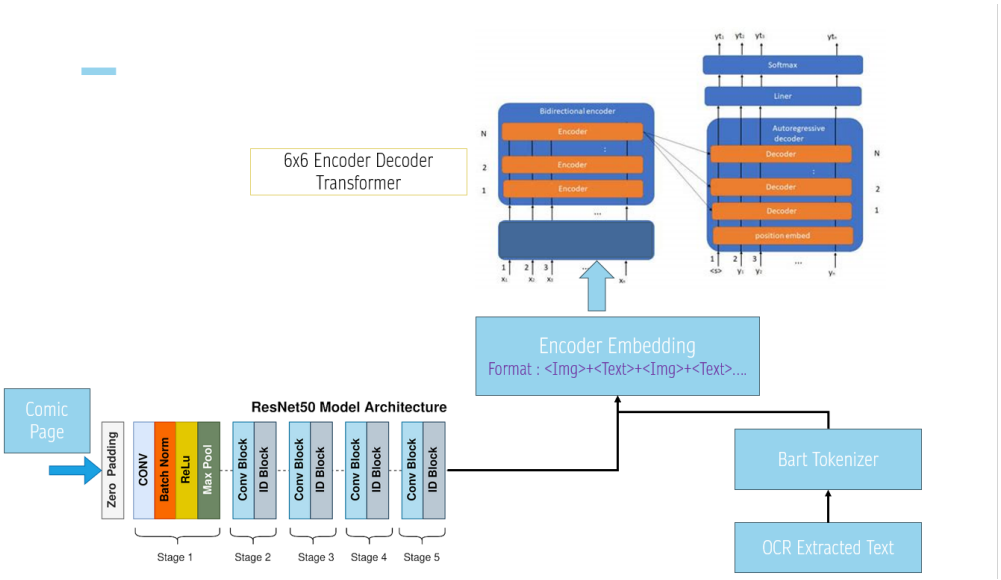

Figure 2: Architecture

3.1.2 Transformer PreTraining

Our model was pretrained using two distinct tasks to enhance its performance in the multimodal summarization task: Masked Language Modeling (MLM) and Summarization Pretraining. Each of these tasks was designed to equip the model with the necessary skills to handle both textual and multimodal data.

Masked Language Modeling (MLM) The first pretraining task was Masked Language Modeling (MLM), where 15-30% of the tokens in the input text were randomly masked. The model was then trained to predict the missing tokens, essentially learning to denoise the input sequence. This task encourages the model to understand the underlying structure and context of the text, which is essential for generating coherent and contextually accurate summaries.

The following datasets were used for MLM pretraining:

- **BookCorpus** - A large collection of books.
- **ROC Stories** - A dataset of short stories designed for narrative generation tasks.
- **nampdn-ai/mini-en** - A smaller English language corpus.

- **WikiBooks2048** - A dataset derived from Wikipedia books.

XSum Summarization Pretraining The second pretraining task involved summarization, specifically on the XSum dataset. This task is designed to help the model learn abstractive summarization techniques by training on news articles and their corresponding single-sentence summaries. The model was trained to condense lengthy textual content into concise summaries, learning how to extract key information and represent it succinctly.

Training Details For both pretraining tasks, we used a batch size of 32, gradient accumulation of 6-8 steps , which allowed the model to efficiently process data while maintaining sufficient memory usage. During pretraining, we also implemented ***Distillation learning***, using a teacher-student framework. The pretrained **bart-large** model was used as the teacher for the MLM task, while the **bart-large-xsum** model served as the teacher for the summarization task. Distillation learning allowed our model to leverage the knowledge of these large models, improving its ability to perform both language modeling and summarization tasks.

The combination of these pretraining tasks helped the model acquire strong foundational knowledge in both language understanding and summarization, making it well-suited for the multimodal comic summarization task.

3.1.3 MultiModal Abstractive Summarization FineTuning

For fine-tuning our model, we used the **VITA/Comic-9K** dataset, which consists of comic books and corresponding summaries. The dataset includes approximately 1792 unique comic stories, containing a total of 44,500 images paired with 1792 textual summaries. These images represent a rich set of comic panels, and the summaries provide a compact representation of the narrative across the panels.

Fine-tuning the model on this dataset enabled it to learn the complex relationships between images and text within the context of comics, refining its ability to generate accurate and coherent abstractive summaries. By utilizing both the visual and textual data, the model becomes capable of understanding the sequential nature of comics and producing summaries that integrate both modalities effectively.

3.1.4 Text extraction

Text extraction was carried out through OCR using the **pytesseract** library. Following this, text is processed by the following steps:

- OCR extraction of all identified characters
- Sentence tokenization using **nltk** library
- Word tokenization of each sentence
- Building lemmatized word index
- Removing words that are not present in corpus

With the following steps, it was possible to extract an array of sentences from each page in the comic.

3.2 Image captioning

Although our model takes a multi-modal approach, we make use of separate image captioning module to deal with the limited dataset and training requirements. Image captioning involves generating a short descriptive sentences for a given image. In this project, we build an end-to-end image captioning system using a convolutional neural network (CNN) for image feature extraction and a gated recurrent unit (GRU) for sequence modeling.

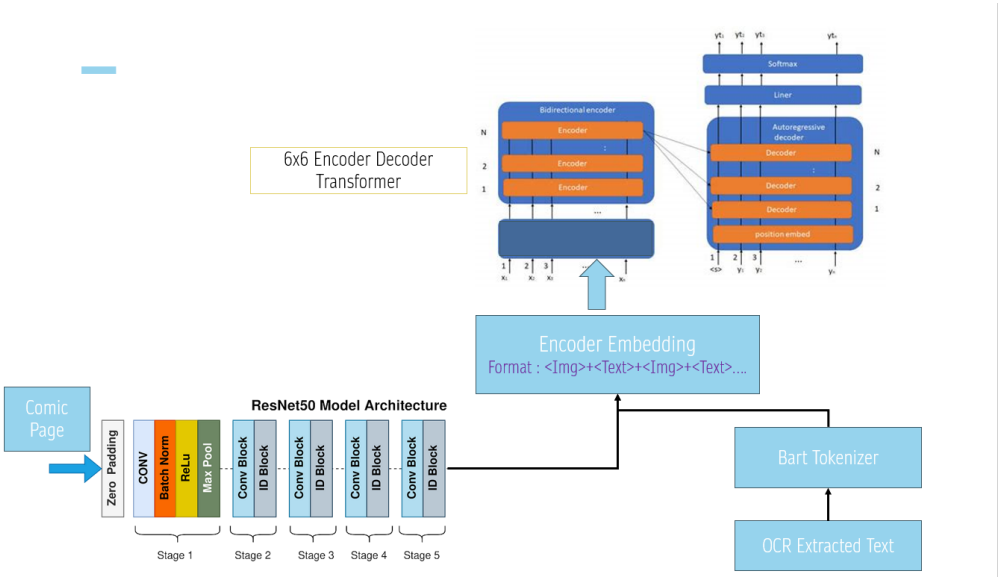

Figure 3: Architecture

3.2.1 Dataset

We used the Flickr8k dataset, which consists of 8,000 images, each annotated with five different captions. This dataset is a standard benchmark for evaluating image captioning models.

Each data sample includes:

- An image (JPEG format)
- Around five textual captions describing the image

3.2.2 Encoder: ResNet-50

We use a pre-trained ResNet-50 model to extract visual features from the input images. Specifically, we remove the final classification layer and use the output of the penultimate fully connected layer as the image feature vector. This results in a 2048-dimensional vector which is then projected down to the GRU hidden dimension (e.g., 100) using a linear layer.

3.2.3 Decoder: GRU

The decoder is a GRU-based RNN that generates captions word by word. The main motivation for using GRU is the relatively low resource requirements compared to training a Transformer, or even LSTM network. This fits the tight resource constraint requirements.

The model uses pre-trained GloVe embeddings to represent words in a semantic space. The decoder takes as input the embedding of the previous word and the hidden state, and outputs the probability distribution over the vocabulary:

3.2.4 Caption Generation

During inference, the encoder processes an image to produce the initial hidden state for the GRU. A start-of-sequence token () is used to begin the generation. The decoder then iteratively predicts the next word until it emits the end-of-sequence token () or reaches a maximum length.

[language=Python, caption=Caption Generation Loop] for $_inrange(max_length)$: $outputs, hidden = decoder(input_token, hidden)next_token = outputs.argmax()if next_token == eos_token : breakcaption.append(next_token)input_token = next_token$

Training The model was trained using cross-entropy loss between the predicted word distributions and the ground-truth words in the caption. We used the Adam optimizer with a learning rate of 1e-3. To prevent memory issues, gradients were computed using loss.backward() and optimizers updated the weights with optimizer.step().

We froze the embedding weights and encoder parameters to reduce computational load and training time. Padding and truncation were handled during preprocessing to ensure all caption sequences were the same length.

3.2.5 Preprocessing and Embeddings

We used the NLTK tokenizer to tokenize captions and created a vocabulary with special tokens SOS, EOS, PAD, and UNK. GloVe embeddings (400k words, 50d vectors) were used for initializing the embedding layer. Out-of-vocabulary words were mapped to ¡UKN¿.

To reduce resource requirements, the top 30k words were shortlisted.

3.3 Conclusion

This project demonstrates an effective pipeline for image captioning using ResNet-50 and GRU. The use of pre-trained models (ResNet and GloVe) helps in transferring learned knowledge and reduces training time. Despite the simplicity, the model performs reasonably well and can be extended to more complex architectures like attention-based transformers.

4 Results

4.1 Method-1 Result

The model generated summaries that sounded accurate in relation to the input comic panels and dialogue, typically limited to 1–2 lines. However, a recurring issue observed was the repetition of character mentions within the output.

The following parameters were derived from training the image captioning model:

- Resnet50 encoder scaling down to 224x224 res images.
- GRU decoder generating sentences up to 32 tokens in length.

- Trained model of 18.3MB in size

Sentences were of moderately sound grammer, albeit with significant repetition. Eg: *a man in a red shirt and a black shirt and a black hat and a woman in a black shirt and a woman in a black shirt*

5 Contributions

Member	Contributions
Nitin	Image captioning with GRU, text OCR
Ashish	Training and Creating custom BART-like model
Nitish	Data pipeline
Ankur	Multi-modal fine tuning

Table 1: Contribution Table

References

[1] Mike Lewis, Yinhan Liu, Nick Tu, Varun Gupta, and Luke Zettlemoyer, *BART: Denoising Sequence-to-Sequence Pretraining for Natural Language Generation, Translation, and Comprehension*, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2019), 2019, pp. 7871-7880.

[2] Emanuele Vivoli, Joan Lafuente Baeza, Ernest Valveny Llobet, Dimosthenis Karatzas, *Multimodal Transformer for Comics Text-Cloze*, arXiv:2403.03719, 2024.

[3] Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, James Allen, *A Corpus and Cloze Evaluation for Deeper Understanding of Commonsense Stories*, Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT), 2016, pp. 839-849.