Generative AI for Video-to-Text Summarization

Ethan Villalovoz
Washington State University
Pullman, WA 99163
ethan.villalovoz@wsu.edu

Fernando Medina
Washington State University
Pullman, WA 99163
fernando.medina@wsu.edu

Luke Flock
Washington State University
Pullman, WA 99163
luke.flock@wsu.edu

Tomer Zangi
Washington State University
Pullman, WA 99163
tomer.zangi@wsu.edu

Abstract

The exponential growth of online video content has created an urgent need for automated summarization techniques to extract meaningful insights from instructional videos. This study explores the potential of generative AI models, specifically FLAN-T5 and BART, for video-to-text summarization. Leveraging the HowTo100M dataset, we developed a preprocessing pipeline to align video captions with task descriptions and fine-tuned state-of-the-art transformer-based models for summarization tasks. Quantitative evaluation using ROUGE and BLEU metrics revealed challenges in achieving high precision and recall, with FLAN-T5 excelling in precision and BART producing verbose but contextually aligned outputs. The study also highlights limitations, including dataset quality, computational constraints, and the complexities of multimodal integration. These insights underscore the need for improved datasets, optimized training strategies, and scalable architectures to advance video summarization. Our findings provide a foundation for future generative AI and instructional content analysis work.

1 Introduction

1.1 Background and Motivation

With the exponential growth of online video content, extracting meaningful insights and summaries from videos has become increasingly important. Instructional videos, such as tutorials and how-to guides, often contain dense information that can be difficult to navigate and consume efficiently. Generative AI promises to transform this landscape by automatically summarizing such videos into concise, text-based descriptions, making the content more accessible and searchable.

Recent advancements in generative models, such as FLAN-T5 and BART, have demonstrated significant potential for text generation tasks. However, leveraging these models for video-to-text summarization remains an open challenge due to the complexity of aligning video data, textual captions, and task descriptions. Additionally, integrating multimodal learning methods, which combine visual, audio, and textual features, poses further computational and architectural challenges.

1.2 Problem Statement

Instructional videos often lack concise and structured summaries that users can quickly reference. While generative AI has proven effective in text-based summarization, applying these methods to video data is non-trivial due to:

Conference on Neural Information Processing Systems (NeurIPS).

- The unstructured nature of video captions.
- The need to align captions with task descriptions accurately.
- Computational constraints, particularly for multimodal integration.

Without robust solutions, users are left with the time-consuming task of manually extracting information from long video content. This problem is especially significant for large datasets like HowTo100M, where scalable solutions are essential for meaningful analysis.

1.3 Objectives and Scope

The primary objective of this project is to develop a generative AI framework for video-to-text summarization, focusing on instructional videos. Specifically, we aim to:

1. Preprocess and align video captions and task descriptions from the HowTo100M dataset.
2. Fine-tune state-of-the-art language models, such as FLAN-T5 and BART, for text generation.
3. Evaluate model performance using standard metrics like ROUGE and BLEU.
4. Explore the integration of multimodal learning to combine visual and textual features (though this step was eventually scrapped due to technical constraints).

This project's scope is limited to using video captions as input data for text summarization. While multimodal learning was considered, computational and dimensionality challenges led to a more focused exploration of text-based methods. This project serves as a foundation for future work in video summarization and highlights both the potential and limitations of generative AI in this domain.

2 Related Work

2.1 Video-to-Text Summarization

Video-to-text summarization is a challenging task that has garnered significant attention in recent years. Traditional approaches have relied on extracting handcrafted features from video frames, such as motion descriptors and object detection outputs, and aligning them with text descriptions using statistical methods. However, these methods often needed help capturing instructional videos' semantic richness and contextual nuances.

Recent advancements have focused on leveraging deep learning models to automate feature extraction and improve semantic understanding. For example, methods such as VideoBERT have utilized pre-trained transformer architectures to model the temporal relationships in videos and align them with corresponding textual descriptions. Despite these improvements, these models are computationally intensive and require large-scale annotated datasets, making them less scalable.

2.2 Generative AI for Text Summarization

Generative AI models have demonstrated remarkable success in text summarization tasks. Pre-trained language models, such as BART and FLAN-T5, excel at generating coherent and contextually accurate summaries. These models leverage transformer-based architectures to perform sequence-to-sequence learning, enabling them to generate concise outputs from complex inputs.

While these models have been widely applied to text-based summarization, their application to video data still needs to be improved. One notable challenge is the alignment of unstructured video captions with meaningful task descriptions.

2.3 Our Approach

Our approach builds on the strengths of pre-trained generative AI models, specifically FLAN-T5 and BART, to address the limitations of existing methods. By focusing on the textual modality (video captions), we avoid the computational complexity associated with multimodal learning while still capturing the semantic richness of the instructional content.

In addition, we introduce a streamlined data preprocessing pipeline to align video captions with task descriptions, leveraging the HowTo100M dataset. This pipeline enables us to train and evaluate generative models efficiently, providing a foundation for future work integrating multimodal features. Unlike prior efforts that rely on large-scale annotated datasets, our approach emphasizes scalability and adaptability to diverse video content.

2.4 Summary of Contributions

Compared to previous work, our contributions are threefold:

- Demonstrated the feasibility of using generative AI models for video-to-text summarization on instructional video datasets.
- Developed a scalable data preprocessing and alignment framework for the HowTo100M dataset.
- Identified and documented the challenges of multimodal integration, paving the way for future research.

3 Methodology

3.1 Data Preprocessing and Preparation

The first step in our approach involved preparing the instructional video data for use in generative text summarization. The HowTo100M dataset served as the primary source for our experiments, containing video metadata, task descriptions, and captions.

3.1.1 Dataset Overview

The dataset includes:

- **HowTo100M_v1.csv**: Contains metadata for each video, such as video IDs, categories, and task IDs.
- **task_ids.csv**: Maps task IDs to corresponding task descriptions.
- **caption.json**: Provides video captions aligned with their respective timestamps.

3.1.2 Preprocessing Pipeline

Our data preprocessing pipeline consisted of the following steps:

1. **Loading Metadata and Mapping Descriptions:** Metadata from `HowTo100M_v1.csv` was merged with task descriptions from `task_ids.csv` using task IDs as keys.
2. **Parsing Captions:** Captions from `caption.json` were aligned with video metadata by matching video IDs, and any malformed or missing entries were filtered out.
3. **Cleaning and Structuring:** Redundant columns were removed, and the cleaned data was saved as `preprocessed_dataset.csv` for subsequent use.

This step outputs a structured dataset containing aligned video IDs, task descriptions, and processed captions, which form the basis for training and evaluating text generation models.

3.2 What is Fine-Tuning?

Fine-tuning is a process of adapting pre-trained models to specific tasks or domains by training them on additional, task-specific datasets. Instead of training a model from scratch, which requires vast amounts of data and computational resources, fine-tuning leverages the knowledge already encoded in a pre-trained model to achieve faster and more effective results.

3.2.1 Why Fine-Tuning is Effective

Pre-trained models, such as FLAN-T5 and BART, are trained on diverse datasets covering various tasks. These models learn general-purpose features and language representations. Fine-tuning adjusts the model's weights to align with the specific requirements of the target task (e.g., video-to-text summarization) while preserving the foundational knowledge.

3.2.2 Steps in Fine-Tuning

1. **Tokenization:** Convert input and output text into tokenized representations that the model can process.
2. **Training:** Use the task-specific dataset to train the model further, optimizing for the desired task's objectives.
3. **Validation and Testing:** Evaluate the model's performance on separate validation and test datasets to monitor overfitting and generalization.

3.3 Text Generation Models and Their Implementation

3.3.1 FLAN-T5 Fine-Tuning

FLAN-T5 is a transformer-based sequence-to-sequence model specifically designed for fine-tuning task-specific instructions. Our fine-tuning process included:

- **Input and Output Pairing:** Captions served as the input text, while task descriptions acted as the target output.
- **Data Splitting:** The preprocessed dataset was split into training (80%), validation (10%), and testing (10%) subsets.
- **Training Loop:** The model was trained for three epochs using cross-entropy loss. A multi-GPU setup was employed for computational efficiency.
- **Evaluation:** The model's performance was evaluated using ROUGE and BLEU metrics, with intermediate checkpoints saved for analysis.

3.3.2 BART Fine-Tuning

BART, another transformer-based model, is known for its robustness in text generation tasks. The fine-tuning process for BART followed a similar structure to FLAN-T5:

- **Tokenization:** Captions and task descriptions were tokenized using the BART tokenizer, with padding and truncation applied to ensure consistent input lengths.
- **Model Training:** The model was trained using the same split dataset, with three epochs and a learning rate optimized for sequence-to-sequence tasks.
- **Comparison:** Results from BART were compared against FLAN-T5, highlighting performance differences in generating task summaries.

3.4 Challenges in Multimodal Learning

An extension of our project aimed to integrate multimodal features, such as visual and textual data, to enhance the summarization process. However, this step was ultimately scrapped due to practical challenges:

- **GPU Memory Limitations:** Despite utilizing 4 NVIDIA RTX A6000 GPUs, the computational demands of multimodal fusion exceeded hardware capabilities.
- **Dimensionality Issues:** Extracted features from video frames had mismatched dimensions, making alignment with textual data non-trivial.
- **Time Constraints:** The timeline for this project did not allow for extensive debugging and optimization of multimodal architectures.

3.5 Summary of Methodology

Our methodology emphasizes a streamlined approach to video-to-text summarization by leveraging pre-trained generative AI models. We provide a foundation for further exploration in this domain through meticulous data preparation, effective fine-tuning strategies, and a focus on practical challenges.

4 Dataset

4.1 Dataset Overview

For this project, we utilized the HowTo100M dataset, a large-scale instructional video dataset containing over 1.2 million video-caption pairs sourced from YouTube. The dataset is particularly suited for video-to-text summarization tasks due to its rich metadata, diverse instructional categories, and detailed captions aligned with video timestamps.

The dataset comprises the following key components:

- **HowTo100M_v1.csv**: Contains metadata such as video IDs, categories, and task IDs for each instructional video.
- **task_ids.csv**: Maps task IDs to task descriptions derived from WikiHow, offering concise textual summaries for various instructional tasks.
- **caption.json**: Provides video captions as time-aligned text segments detailing the spoken content in the videos.

4.2 Data Cleaning and Alignment

Given the raw nature of the HowTo100M dataset, several preprocessing steps were required to align and clean the data for our experiments. The cleaning and alignment process was as follows:

4.2.1 Metadata Alignment

- **Task Description Mapping:** The `task_ids.csv` file was used to map task IDs in `HowTo100M_v1.csv` to their respective descriptions. This step ensured each video was associated with a clear instructional goal.
- **Column Filtering:** Only relevant columns (e.g., `video_id`, `category_1`, `category_2`, `task_description`) were retained, reducing noise in the dataset.

4.2.2 Caption Parsing and Cleaning

- **Parsing Captions:** Captions from `caption.json` were loaded and mapped to their corresponding video IDs in the metadata file. Only video IDs present in both files were retained.
- **Text Cleaning:** Common preprocessing techniques were applied to captions, including:
  - Removing stop words and punctuation.
  - Converting text to lowercase for consistency.
  - Handling missing or malformed captions by filtering them out.

4.2.3 Final Dataset Preparation

After cleaning and aligning the data:

- The processed dataset was saved as `preprocessed_dataset.csv`, containing columns for video IDs, task descriptions, categories, and processed captions.
- A subset of the dataset was extracted for this project, prioritizing completeness and alignment between captions and task descriptions.

4.3 Dataset Splits

The preprocessed dataset was divided into three subsets for training, validation, and testing, ensuring a balanced distribution across instructional categories:

- **Training Set (80%):** Used to fine-tune the generative models. This set contained most of the data, ensuring sufficient diversity for model learning.
- **Validation Set (10%):** Used to monitor the model's performance during training and prevent overfitting. This set helped adjust hyperparameters and evaluate generalization.
- **Testing Set (10%):** Reserved exclusively for final model evaluation. This set provided an unbiased estimate of the model's performance on unseen data.

4.3.1 Dataset Statistics

- **Original Dataset Size:** 1.2 million video-caption pairs.
- **Subset Used:** 0.2% of the original dataset was sampled for this project due to computational constraints, resulting in approximately 24,000 samples.
- **Distribution Across Categories:** Ensured representation of diverse instructional categories in all three splits.

4.4 Summary

The HowTo100M dataset provided a comprehensive foundation for video-to-text summarization experiments. Through rigorous cleaning, alignment, and splitting processes, the preprocessed dataset served as a reliable input for training and evaluating the FLAN-T5 and BART models. The data preparation pipeline ensured the quality and consistency required for generating meaningful results.

5 Experimental Setup

5.1 Hardware Specifications

The experiments were conducted on a high-performance workstation with the following specifications:

- **Processor:** AMD Ryzen Threadripper PRO 5995WX with 64 cores and 128 threads, operating at a base frequency of 2.7 GHz and a boost frequency of 4.5 GHz.
- **Memory:** 503 GiB of RAM, ensuring sufficient capacity for large-scale datasets and model training.
- **Storage:**
  - NVMe SSD: 3.5 TB, used for system operations and caching.
  - HDD: 87.6 TB distributed across six 14.6 TB drives designated for large dataset storage.
- **GPUs:** 4 NVIDIA RTX A6000 GPUs, each with 48 GB of memory, operating on CUDA version 12.2. These GPUs provided high computational throughput for training large models.

This configuration enabled efficient preprocessing, model fine-tuning, and evaluation across multiple experiments.

5.2 Software Environment

The software stack used for implementing and training the models included:

- **Operating System:** Ubuntu 22.04 LTS.
- **Programming Language:** Python 3.8.
- **Deep Learning Libraries:**
  - `transformers`: For leveraging pre-trained models like FLAN-T5 and BART.
  - `torch` (PyTorch): For model training, dataset handling, and GPU acceleration.

- `datasets`: For computing evaluation metrics like ROUGE and BLEU.

- **Data Processing Tools:**
  - `pandas`: For data cleaning and organization.
  - `json`: For parsing and processing video captions.
  - `yt-dlp`: For downloading YouTube videos from the HowTo100M dataset.
- **Visualization Tools:**
  - `matplotlib` and `seaborn`: For plotting model performance and dataset statistics.
- **Version Control:** GitHub was used for collaborative development and maintaining version history.

5.3 Model Hyperparameters

The key hyperparameters for model fine-tuning were as follows:

- **Batch Size:** 16.
- **Learning Rate:** 5×10^{-5} with the AdamW optimizer.
- **Number of Epochs:** 3.
- **Maximum Sequence Length:** 512 tokens for both input and output.
- **Loss Function:** Cross-entropy loss.

5.4 Training Workflow

The training process was structured as follows:

1. **Data Loading:** Preprocessed data was divided into training, validation, and testing sets.
2. **Fine-Tuning:** Pre-trained models were fine-tuned on the training set, using captions as input and task descriptions as the target output.
3. **Validation:** Model performance was evaluated on the validation set at the end of each epoch.
4. **Checkpointing:** Checkpoints were saved after each epoch for recovery and further experimentation.
5. **Testing:** The fine-tuned models were evaluated on the unseen test set to measure generalization.

5.5 Challenges Encountered

The following challenges were encountered during experimentation:

- **GPU Memory Constraints:** Despite high-memory GPUs, multimodal learning faced dimensionality issues, leading to the scrapping of this step.
- **Dataset Complexity:** Aligning captions with task descriptions required extensive preprocessing and validation.

5.6 Summary

The experimental setup provided a robust environment for conducting large-scale generative AI experiments, ensuring optimal resource utilization and reproducibility.

6 Results and Evaluation

This section comprehensively evaluates the models employed in our generative AI project, comparing FLAN-T5 and BART. It presents quantitative and qualitative analyses, highlighting the strengths and limitations of each approach.

6.1 Quantitative Results

6.1.1 FLAN-T5 Performance

The results indicate modest performance in text summarization, with low BLEU scores and limited n-gram precision beyond unigrams. FLAN-T5 generated outputs that diverged significantly from the ground truth, as evidenced by the BLEU and ROUGE scores.

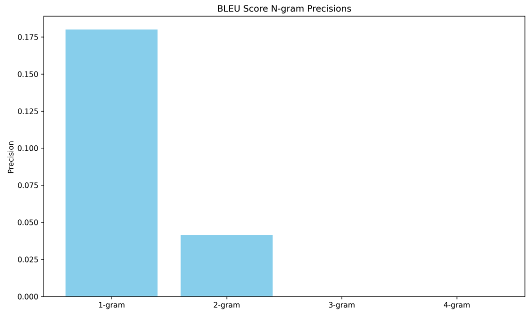

Figure 1: BLEU N-gram Precision Plot FLAN-T5 Model

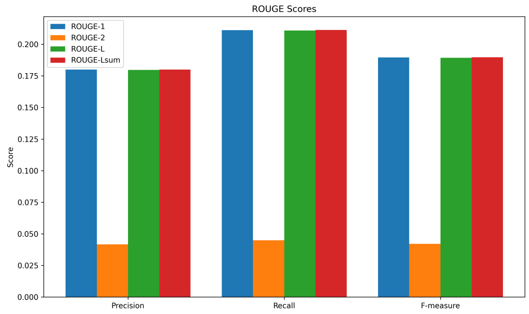

Figure 2: ROUGE Scores Plot FLAN-T5 Model

6.1.2 BART Performance

BART demonstrated relatively better recall but lower precision, suggesting verbose output that partially aligns with the reference summaries. The BLEU score remained low, emphasizing the gap between generated and target summaries.

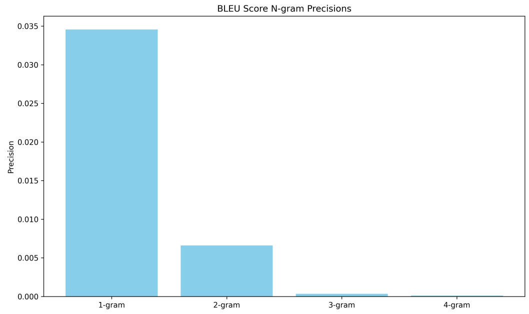

Figure 3: BLEU N-gram Precision Plot BART Model

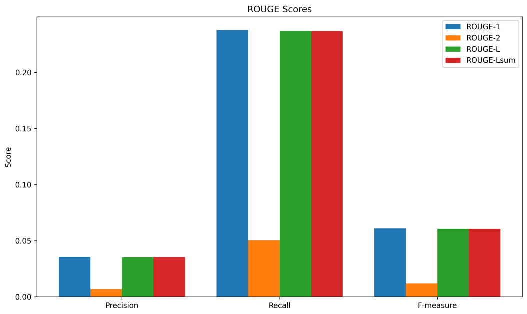

Figure 4: ROUGE Scores Plot BART Model

6.2 Qualitative Analysis

Below are sample outputs from both models:

FLAN-T5 Generated Summary: “Make a Grass Root Beer” **Reference Summaries:**

- “Make Crawfish Bisque”
- “Measure and Record Vital Signs During First Aid”

BART Generated Summary: “Make a Duct Tape Bracelet out of a T-Shaped Plastic Bottle (Basement Backsplash Method) With a Plastic Bag and a Coffee Mug and a Plastic Retaining Container (Beverly Method)” **Reference Summaries:**

- “Make Crawfish Bisque”
- “Measure and Record Vital Signs During First Aid”

The generated summaries often contained repetitive or irrelevant information, highlighting challenges in alignment with the task’s objectives.

6.3 Challenges in Performance

Both models exhibited limitations, including:

1. Poor handling of domain-specific language.
2. A lack of generalization for unseen tasks.
3. Difficulty aligning generated summaries with context.

Future work aims to explore improved training techniques and datasets for enhanced alignment and relevance in summarization tasks.

7 Discussion

7.1 Analysis of Results

The evaluation of FLAN-T5 and BART models revealed distinct performance characteristics. FLAN-T5 demonstrated higher precision in 1-gram predictions, suggesting a better understanding of basic sentence structures. However, both models struggled to capture higher-order relationships, as evident in low BLEU and ROUGE-2 scores. BART, on the other hand, achieved better recall, indicating it generated more verbose outputs that partially aligned with the ground truth but at the expense of precision.

7.2 Key Insights and Observations

- **Limited Generalization:** Both models failed to generalize effectively to unseen tasks, as seen in their inability to produce diverse summaries closely matching the references.

- **Verbose Outputs:** BART's outputs were overly verbose, often deviating from the succinctness required for effective summarization.
- **Challenges with Domain-Specific Content:** Both models struggled with domain-specific tasks, such as accurately summarizing complex or unfamiliar instructional videos.

7.3 Limitations of the Current Approach

- **Dataset Quality:** The dataset lacked sufficient alignment between video content and textual annotations, leading to suboptimal learning.
- **Model Capacity:** Both models faced challenges in learning effective summarization strategies due to limited task-specific fine-tuning.
- **Resource Constraints:** GPU memory limitations restricted experimentation with larger models and multimodal architectures.

8 Future Work

Further fine-tuning using high-quality, task-specific datasets is essential for improving model performance and enhancing the model's understanding of instructional content. Additionally, data augmentation strategies can improve robustness and enable better generalization across diverse tasks and scenarios.

Addressing GPU memory constraints is another critical area of focus. Access to higher-performance GPUs or implementing distributed training setups would allow experimentation with larger models and more complex architectures. Moreover, applying model optimization techniques, such as pruning and quantization, can significantly reduce computational overhead and make training more resource-efficient.

Finally, revisiting the multimodal integration step with optimized pipelines and dimensionality reduction techniques can help manage diverse input modalities effectively. These improvements are crucial for advancing the models' capabilities in handling complex tasks that require both visual and textual understanding.

9 Conclusion

This project explored generative AI techniques for video-to-text summarization using FLAN-T5 and BART models. Quantitative results highlighted the limitations of current models, particularly in domain-specific tasks, while qualitative analysis underscored challenges in alignment and coherence.

The findings emphasize the need for improved datasets, task-specific fine-tuning, and resource-efficient multimodal approaches. These insights pave the way for future advancements in generative AI for instructional and domain-specific applications.

10 Appendix

10.1 Code Snippets and Configurations

The project code is available at [GitHub Link].

10.2 Description of the Scrapped Multimodal Step

The multimodal step, which aimed to integrate video features with text generation, was scrapped due to challenges in handling high-dimensional inputs and GPU memory constraints. Future work will address these limitations by employing dimensionality reduction techniques and scalable training pipelines. To make up for the work, in the original proposal, we were only going to do FLAN-T5 for text generation. We decided to include BART to compare models so that we could still have interesting comparisons within our work.