Teacher-Student Distillation for Real-Time Multi-Modal Video Captioning and Semantic Transition Modeling

Kelsey Knowlson*, Matthew Rackley†
Department of Computer Science, University of New Mexico, Albuquerque, NM

***Abstract*—This project leverages multi-modal machine learning models to create accurate, real-time caption generation for videos by using GPU accelerated computations and pretrained models. The BLIP-2 model serves as the primary mechanism for caption generation from raw video frames. To capture temporal shifts in frames, a teacher-student framework is employed with a GPT model acting as the teacher and a BART model as the student. This structure allows the detection of content changes, such as new objects or backgrounds, so that transitional sentences can be generated to bridge the changes. The final caption is created by feeding the sequence of frame-level captions and their respective transitional captions to a GPT model, which summarizes and composes a succinct analysis of the video content. Compared to results from the isolated BLIP-2 model, the modifications in the framework improve cosine similarity scores of the generated text to that of the labels by about 14.55% and increase the fluency and coherency of generated text, measured by calculating perplexity, by 47.82%. Real-time caption generation was achieved by using GPU accelerated computation, decreasing the total processing time to 71.47% of the length of the video.**

I. Introduction

Multi-modal machine learning is a field of machine learning that has been growing in popularity due to an increasing demand for entertainment utilizing technology such as augmented and virtual reality, chat-bots and virtual assistants employing speech and visual inputs, and for video captioning and audio descriptions. [25] The scope of this paper is limited to multi-modal applications related to video captioning and summarization, specifically as it pertains to accessibility applications.

Video captioning is a key component to accessibility for individuals with hearing or sight impairments, whether through written or audio captions. The motivation behind this project is to examine methods of creating accurate video descriptions in real time. Using multi-modal machine learning techniques can not only enhance an important specialty of machine learning, but can be a vital component of developing practical and inexpensive solutions to promote inclusivity in the real-world. The aim of the research is to improve and optimize current multi-modal techniques to create accurate and succinct video descriptions generated in real-time while utilizing pre-trained models such as Bootstrapping Language-Image Pre-training (BLIP), 3D Convolutional Neural Networks (3D CNNs) such as Inflated 3D ConvNet (I3D), Bidirectional and Auto-Regressive Transformers (BART), and Generative Pre-Trained Transformers (GPT) models. The research will explore methods of combining pre-trained models, using teacher-student models to fine-tune outputs, leveraging GPU power to enhance speed and simulate real-time generation, and optimizing parameters of models.

II. Methodology

This research leverages pre-trained models due to the impracticality of training models from scratch on large data sets due to time and computational constraints. BLIP-2, BART, DistilBART, and GPT models are the primary models that will be employed for video evaluation and text generation. Additional models were examined and tested to determine the best model to produce the optimal outcomes.

A. Models

BLIP [26] [17] [16] is a pre-trained machine learning model that uses dual-encoder and multi-modal techniques to identify images and generate descriptions of their content. It is similar to the CLIP model in that it uses contrastive learning, but differs in that it is able to generate text instead of merely matching text-image pairs. It utilizes the following techniques: Image-Test Contrastive Loss (ITC) to promote similarity between images with similar content and distance between images with dissimilar content, Image-Test Matching Loss (ITM) to identify correlations between the visual and textual inputs with binary classification, Language Modeling Loss (LM) to generate text descriptions of inputted images by minimizing cross-entropy loss. [9] The BLIP-2 model is an advancement of BLIP improving accuracy and generalization

ability. It incorporates a vision encoder, as does BLIP, but it couples it with a pre-trained frozen Large Language Model (LLM) both making it more modular, effective and accurate in text generation.[6]

In addition to BLIP, Kosmos-2 [21], a Multi-modal Large Language Model (MLLM) was examined for its potential as the primary image-to-text model due to its image identifying and text generation qualities. Built on a transformer-based framework, Kosmos-2 has expanded upon Kosmos-1 by introducing text grounding features, which is the ability to link specific aspects of generated text to a location in the image and allows this model to specify where in the image elements exist in the frame. Kosmos-2 allows a prompt to be entered upon calling the model, which can help focus the scope of the text generation. [7] [22]

Lastly, a Generative Image-to-Text Transformer (GIT) was considered in this project, since its primary intended task is to be used for image and video captioning. The model uses a single image encoder and text decoder, coupled with a more extensive training dataset than most models, potentially providing more accurate classifications. [31]

This project experimented with incorporating 3D CNNs for action detection in videos to augment the results of the BLIP-2 descriptions. The primary models considered were the I3D model, ResNet-3D (R3D), and Extreme 3D (X3D). These models are designed to analyze videos and determine specific actions in image sequences. Unlike traditional two-dimensional CNNs, 3D CNN models are enhanced to handle spatial and temporal relationships in image sequences, which makes this type of model an ideal choice for video captioning. They accomplish this by adding an additional depth element into their kernel which allows them to learn temporal aspects of the images, unlike 2D CNNs which cannot draw a relationship between two frames. [10]

Multiview Vision Transformers (MViT) were also considered as a method to enhance the detail and accuracy of captions. Although 3D CNNs can capture temporal relationships, their focus lies on short-term dependencies as they cannot retain or learn temporally distant relationships. MViT models improve upon this short-coming by using self-attention mechanisms to retain long-term data and expand upon the temporal learning capabilities of 3D CNNS. Specifically, the MViT-v2 model was utilized for its improved long-range dependencies as compared to the standard MViT. [18]

Finally, GPT models were utilized in experimentation. GPT models are an ubiquitous machine learning model that is used to generate fluent text by applying the use of transformers in their architecture. Due to the self-attention mechanisms of transformers, the GPT model can learn the context of a word regardless of its location within a sentence, which leads to a better understanding of languages and more reliable text generation. In this research, the specific pre-trained GPT models used ranged from GPT-2 to GPT-4 for testing and experimentation both due to accuracy and accessibility.

For summary tasks, BART, DistilBART, and the aforementioned GPT models 3.5 and 4 were utilized. Both BART and DistilBART are sequence-to-sequence models that utilize autoencoders with a bi-directional encoder and an autoregressive decoder. DistilBART differs from BART in that it is a more lightweight version of the model due to minimizing parameters, which unfortunately slightly impacts the accuracy negatively. Due to the project goal of real-time caption generation, both models were examined to find the balance between speed and accuracy. [8] [30]

B. Video Frame Splicing

Short videos are spliced into frames using two separate methods to accommodate both the mechanics of BLIP-2 and 3D CNN models. The 3D CNN models focus on determining actions in videos by processing temporally linear frames from fixed-length sequences. Due to this, frames are taken in sequential windows of 16 to 64 frames per window.
In contrast, the BLIP-2 model is designed to identify and describe stagnant image scenes, establishing the need to create a sampling of still frames at set intervals from the videos to pass to the BLIP-2 model.
In the testing model that involves the use of 3D CNNs, the still frames will correlate with the last frame in a 3D CNN clip sequence for consistency in testing. Otherwise, the frames taken for the BLIP-2 model video will be taken at the intervals of the frame per second rate of the video. The specific method used to obtain the video frames for BLIP-2 and the combined testing of BLIP-2 with action identifying models is the VideoCapture method from the cv2 Python library. In solitary testing of the 3D CNN frame sequences the read_video method is used from torchvision as it tends to be faster but is not as malleable for single frame selection, making it more suited for tasks involving frame sequences.

C. Description Finalization

Once the final raw caption strings are generated from each video, they are passed to a final LLM model consisting of one of the following: BART [15] , DistilBART [29], FLAN-T5, GPT-3.5, or GPT-4. The purpose of this final processing is to summarize and rephrase the text into a single coherent description of the video that will serve as the final predicted caption.

D. Experimental Structures

Experimentation was conducted to determine if the inclusion of 3D CNN models or MViT models for action detection should be incorporated into the final structure to influence BLIP-2's output. The testing was conducted by inputting sequential sequences of 16 frames into the action models, which accept short clips instead of individual frames. The last frame for each clip was then run through BLILP-2. The results from both models were concatenated into a final description for each time step such that the string would hold the structure:

```
"Action:<3D CNN action classification>.
Description:<BLIP image description>"
```

In the case of a video of a man running on a path with a dog, the raw output from the frame might appear as "Action: Walking dog. Description: A man is on a path with red flowers." The list of concatenated captions were then fed into a GPT model to appropriately summarize into a single coherent video description. The concept was to enhance the overall accuracy of the captions by using BLIP-2 to fill in the details of the background and surroundings while a 3D CNN or MViT model determines the actions being conducted by the subject of the video. The action detection was not intended to replace the BLIP-2 model but instead to augment the overall process.

III. Model Development

The development of a robust model is a key component to the success of the project. An emphasis was placed early on the construction of it's core architecture to directly address the primary challenge: enabling efficient, coherent, and non-redundant video captioning. Many existing pre-trained caption models are remarkably repetitive. Often reiterating similar tokens across frame windows with little to no semantic change. This repetitiveness increases token generation and inflates computational overhead. The reduction of which is critical to an online focused real-time application such as our target research.

Formulating the following question will help guide the thought process:

```
Can we efficiently detect semantically
meaningful changes in a video stream and
describe transitions in a way that is both
narratively coherent and computationally
lightweight?
```

This problem draws from narrative theory. [20] According to which states that a written story derives its momentum not from static, sanitary descriptions of a scene. But from it's transitions. [19] As such, an effective online captioning system must capture the progression of elements between those moments in time. [11] [1] [2]

This framing naturally lends itself to a temporal formalization of the problem: identifying meaningful transitions between moments in time and expressing them in language. We define this as follows:

Temporal Caption Transition Formalization

Let a video be represented as a sequence of frames:

$$V = \{f_1, f_2, \ldots, f_T\}$$

Each frame f_t is passed through a vision-language model (e.g., BLIP) to produce a caption $c_t \in \mathcal{C}$, resulting in a sequence of natural language descriptions:

$$\mathcal{C} = \{c_1, c_2, \ldots, c_n\}$$

We define a transition function:

$$\Delta_T : (c_i, c_{i+1}) \mapsto s_i$$

Here, $s_i \in \mathcal{S}$ is a concise, narratively coherent sentence capturing the semantic evolution between c_i and c_{i+1}. The goal of the proposed model is to learn this mapping efficiently and reliably.

From a modeling viewpoint, this strategy involves sequence modeling and time series analysis. While object-tracking and action recognition have well defined and trained models. Much less attention has been given to generating NL transitions that bridge captions of adjacent video frame windows. The model addresses this gap by framing the video as a sequence of discrete, captioned frames or segments and learning to generate concise transitional sentences between these pairs.

For our initial scope, we constrained inputs to short video clips from a stock footage website. Each of which contain a stable scene. This allows us to focus the development of the model to it's ability to learn inter-caption relationships and generate meaningful inter-scene transition statements without an increasingly complex scene.

A. Identification of Key Tasks

From the above section we can identify key tasks for our goal and the advancement and refinement of a newly generated model:

1) **Maintain near to complete real time processing:** Maintaining a streamlined process and model pipeline that is not overly large and prioritizes efficiency is key to allowing the model to run onboard, online and with maximum accessibility.
2) **Reliability in vision inference:** Utilizing a model that already has little to no hallucination is key due to downstream tasks that rely on previous inputs to achieve good results.
3) **Achieve the newly desired token reduction/transitional statements:** Through prompt engineering or pre-trained models. Achieving this behavior on a real-time model would be ideal.
4) **Accuracy:** The model should clearly define what's occurring in the frame window with a high probability of success barring hallucinations that affect the downstream task.

B. Pretrained Models

Various pretrained models were tested for the final model pipeline due to limited resources and scope. The following is a breakdown of the various models and in which areas of the pipeline they were utilized:

TABLE I: Models Evaluated by Pipeline Stage

Stage	Models
Frame Window Inference	BLIP, CLIP, I3D, R3D, X3D, MViT-v2, Kosmos-2, GIT
Summarizer Tasking	BART, DistilBART, GPT-2, TinyLLaMA, FLAN-T5, Mode
Transition Generator	GPT-2, TinyLLaMA, FLAN-T5, Mode BridgeBART

C. Prompt Engineering

While several of these models such as BART and DistilBART are specialized in summarization tasks, they are also data greedy. Supplying merely two sentences to generate a semantic transition between the two embeddings proved fraught with challenges. Not only were hallucinations observed, but the lack of termination lead to numerous incoherent statements where it's pre-trained nature was apparent.
Provided is an example:

```
The man in a blue mask is shopping at a
market. One day a man in a blue mask and
gloves is buying a bunch of blue onions.
Soon it turns out a woman is also buying blue
onions.
```

Methods of using more advanced LLM/GPT models such as tinyllama [33], FLAN-T5 [5], GPT-2 [23], GPT-3.5/GPT-4 [3] were employed to examine if instruction tuning could provide the correct behavior without using excessive limited resources.

Various methods of instruction tuning were necessary to accommodate more advanced models that received larger prompts and have improved contextual understanding. The prompt that generated the most reliably successful generalized summaries was as follows:

```
You are a summarizer. Follow these steps
carefully.
1. Read the following scene descriptions
exactly as written.
2. Identify only the nouns and adjectives
present in the descriptions.
3. Do NOT add anything that is not mentioned
in the text.
4. Combine the descriptions into one clear
sentence.
```

However, with limited resources, even the prompted LLM models proved ineffective in generalizing. In addition, textual hallucination continued to be a common issue, leading to the decision to combine both.

D. Teacher-Student Model

A new construction of a Teacher-Student model [28] was needed to allow for the advanced prompting of a powerful LLM to be distilled [14] into a lightweight, token enforced model to prevent excessive output negatively impacting downstream summarization or real-time output.

1) Building The Dataset: One of the more difficult challenges of the project, was the lack of an existing dataset specifically built to train using transitional sentences designed to bridge semantically similar sentence pairs. Unfortunately,

this entailed the manual creation of a high-quality dataset that could bootstrap the Teacher Model in this process.

As shown in Figure 1, the data used for caption pairs was the Microsoft Research Video Description Corpus [12] [4]. This segment of clips consisted of approximately 2,000 short clips of various degrees of quality and captions. These contiguous pairs of grouped clips were run through the BLIP-2 model. After pre-processing was complete, a transformer model employed to cluster and filter similar captions to create a dataset of approximately 4,000 context rich caption pairs.

About 200 captions were hand classified with a transitional sentences to bootstrap the model's generalization for the next stage.

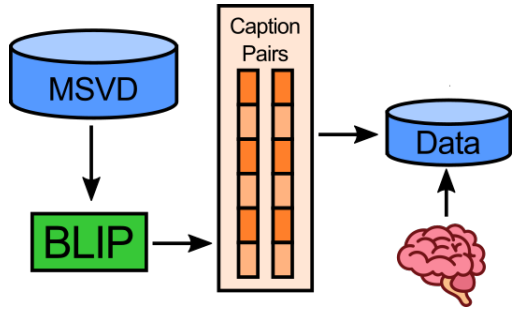

Fig. 1: A small snippet of the full Teacher-Student distillation pipeline as seen in Figure 22. MSVD clips from Sarthak [12] were passed through a basic BLIP-2 model to obtain caption pairs that are directly adjacent to each other in the timeseries video stream. This data was then condensed and compiled into a clean dataset that was then hand-classified by humans to provide the necessary information for bootstrapping the teacher model.

2) Human-in-the-loop Training: Several cycles of Human-in-the-loop training were performed on a random subset of the caption pairs [32], allowing the human to provide additional feedback in terms of prompt engineering, fine tuning, and additional hand-classification. [24] This process is seen in Figure 2, which shows the cycle of evaluation, feedback and training on an advanced GPT model.

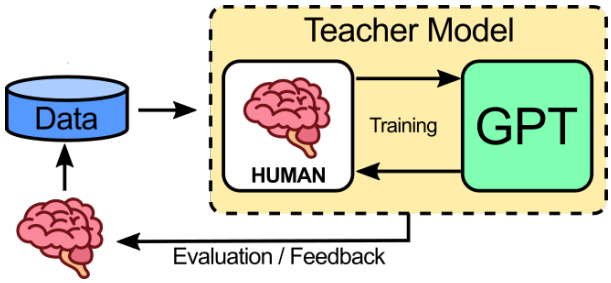

Fig. 2: A focused segment of the full Teacher-Student distillation pipeline, as seen in Figure 22. At this stage, a large-scale GPT model was utilized with prompt engineering to assist in labeling the *MSVD Caption Pair* dataset, as seen in Figure 1. The process uses iterative evaluation to reshape the data and GPT output.

3) Distillation of Knowledge: As shown in Figure 3, the outputs of the Teacher model, which are high-quality transitional sentences, are used in an offline setting to fine-tune a lighter weight BART model. This distilled model is intended to replicate the behavior of the more capable GPT-based Teacher, but within a more efficient architecture suitable for real-time inference.

To formalize this process, we consider the distillation as a knowledge transfer problem, where the goal is to minimize the divergence between the outputs of the Teacher and Student models on the same input pairs.

Distillation Objective

Let T be the Teacher model (e.g., GPT-4), and S be the Student model (e.g., BART). Each training sample consists of an input pair $x = (s_k, s_{k+1})$, representing successive semantic summaries, and a target transition sentence $y = t_k$.
The distillation loss is computed as:

$$\min_{\phi} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\mathrm{KL}\left(T(x) \,\|\, S_\phi(x) \right) \right]$$

Here, ϕ denotes the learnable parameters of the Student model, and KL is the Kullback-Leibler divergence between the output distributions of Teacher and Student.

This process enables the Student model to internalize the mapping from the summary pairs to the narrative transitions, while remaining computationally efficient. A final schematic of this distillation pipeline is shown in Figure 22.

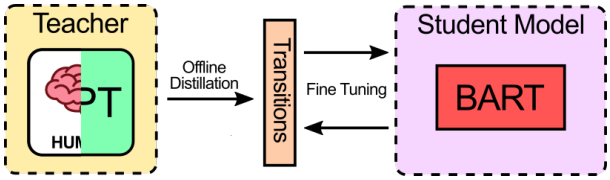

Fig. 3: A segment of the full Teacher-Student distillation pipeline, as seen in Figure 22. The outputs of the Teacher model are used to fine-tune a light weight BART-based Student, transferring the ability to generate semantic transitions.

E. Full Model Construction

This section provides a description of the final model pipeline being evaluated for this research.

1) Video Parsing and Captioning: As shown in Figure 4, a video scene is parsed into frames and passed through the generic Bootstrapped Language-Image Pre-training (BLIP) model to acquire captions in a time series. Dynamic frame

window size is designed for video parsing, although default is one frame per second to be sent to BLIP.

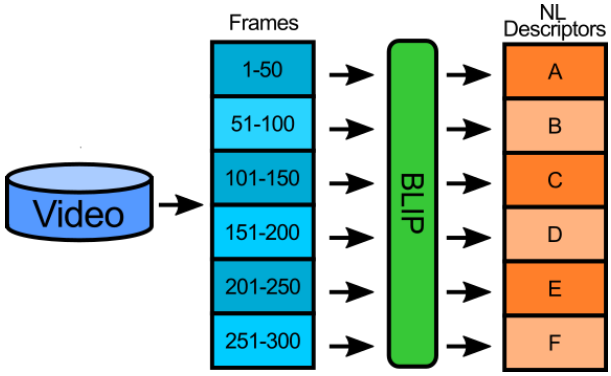

Fig. 4: A section of the full pipeline from Figure 23. A video is separated into frame windows, which are passed through BLIP to obtain natural language descriptors.

2) Clustering Protocol: The clustering stage of the pipeline condenses semantically similar captions into groups. As shown in Figure 5, the stream of natural language captions generated from the BLIP-2 model are passed into a transformer (typically MiniLM or MPNET) to condense its sentiment. Next, the embeddings are clustered in relation to their cosine similarity [27] threshold denoted λ. The stream of clusters is then output along with it's timeseries data as an additional embedding for forward models and downstream tasks.

Clustering Formalization

Let $\mathcal{C} = \{c_1, c_2, \ldots, c_T\}$ be a sequence of captions generated from video frames and let $E = \{e_1, e_2, \ldots, e_T\}$ be their corresponding embeddings. Cosine similarity is computed between any pair:

$$\text{sim}(e_i, e_j) = \cos(\theta) = \frac{e_i \cdot e_j}{\|e_i\|\|e_j\|}$$

Two captions c_i, c_j are grouped if:

$$\text{sim}(e_i, e_j) \geq \lambda, \quad \lambda \in [0, 1]$$

Clusters are then formed as:

$$\mathcal{G} = \{G_1, G_2, \ldots, G_K\}, \quad G_k \subseteq \mathcal{C}$$

such that all $c_i \in G_k$ are mutually semantically similar.

The resulting clusters are time-ordered and passed forward with metadata, including timestamp and cluster index, to aid in tracking semantic flow over time.

3) Summarizing and Transformation: The clustered groups are passed through one of the summarizer models proposed in Table I. These summarizers are pre-trained to produce approximate semantic captions, as long as sufficient context is provided. The results showed that taking the *mode* of

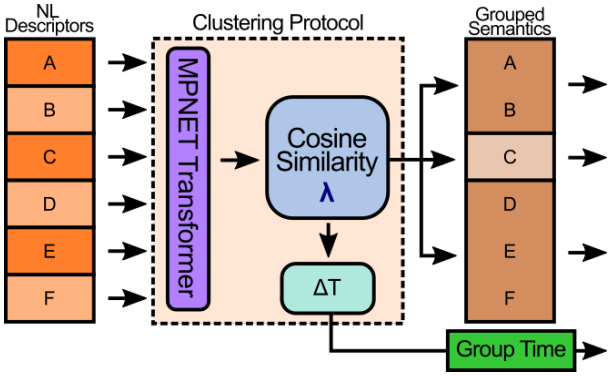

Fig. 5: A partition of the full pipeline from Figure 23. Natural language descriptors are transformed into embeddings and clustered based on cosine similarity using a threshold λ. These clusters are grouped with their associated timeseries metadata for downstream processing.

the grouped captions, provided a sufficiently tight similarity threshold, was often a competitive strategy for group-level summarization when improved performance was needed for heavier models downstream.

The following is a formal definition of this operation:

Cluster Summarization

Let $G_k \subseteq \mathcal{C}$ be a cluster of semantically similar captions. The summarization function is defined as:

$$\Sigma : G_k \mapsto s_k$$

where $s_k \in \mathcal{S}$ is the representative sentence summarizing the semantic content of G_k.
Alternatively, the mode of the set can be used:

$$s_k = \arg\max_{c \in G_k} \text{freq}(c)$$

The summary sentences s_k are next paired in temporal order and passed into the Teacher-Student distilled model *BridgeBART*, seen in Figure 6. To better understand the task this model performs, it can be formalize in terms of sequence transformation and supervised training.

BridgeBART Formalization

Let $s_k, s_{k+1} \in \mathcal{S}$ be successive cluster summaries, and let $t_k \in \mathcal{T}$ denote a sentence capturing the transition between them. BridgeBART is trained to model the sequence-to-sequence transformation:

$$\mathcal{B}(s_k, s_{k+1}) \rightarrow t_k$$

The training objective is to minimize the cross-entropy loss over a dataset $\mathcal{D}$:

$$\min_{\theta} \sum_{(s_k, s_{k+1}, t_k) \in \mathcal{D}} \mathcal{L}(\mathcal{B}_\theta(s_k, s_{k+1}), t_k)$$

where θ are the learnable parameters of BridgeBART, and $\mathcal{L}$ is the loss function enforcing alignment between generated and target transitions.

If the distillation and fine-tuning of *BridgeBART* is successful, it should produce simplified real-time outputs that represent the semantic change between two scene-level captions. An example of story flow can be seen in Figure 17, where *BridgeBART* attempts to describe the directed edges between each embedding.

Finally, the model has the option to directly output these transitions for real-time streaming applications or pass them to a downstream LLM for higher-order narrative evaluation and inference.

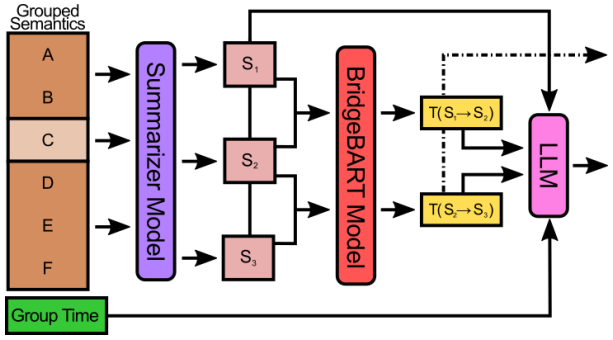

Fig. 6: A portion of the full pipeline from Figure 23. The grouped semantic clusters are passed through a summarizer model to produce the representative embedding vector of the set. This is then passed as pairs through the distilled BridgeBART model which outputs context-rich transitional sentences. At this point, it can be passed to the LLM or output directly for real-time streaming.

IV. Testing

Testing was conducted to identify the most practical models for the pipeline and to determine the optimal stage to introduce them into the structure. Image captioning models were first examined to assess the most appropriate pre-trained model to use as the primary source of image-to-text generation. It should be noted that CLIP was included despite being a contrastive image classification model due to its recognition as a highly accurate image classifier. BLIP was also considered but was ultimately not included in the testing due to the higher success rate and improved functioning of its successor BLIP-2.

Image captioning models were tested by capturing one still frame per every 16 frames in a video. Captioning was generated for each of these frames, creating an outcome pool from which the *mode* description was selected as the final caption. A cosine similarity score compared to the actual label was determined by converting the actual and predicted values into embedding with the MPNET-v2.

Image Captioning Model Testing Formalization

Let $\mathcal{V}$ be the video dataset of N videos. $\forall v_i \in \mathcal{V}$ a set of frames is created as $\mathcal{F}_i$. $\forall f_{ij} \in \mathcal{F}_i$ a caption is generated as $C(f_{ij})$. The final caption for v_i is determined by:

$$\hat{\mathcal{C}} = mode(\{C(f_{i1}), C(f_{i2}), ...C(f_{ij})\})$$

The average cosine similarity score denoted by function $\mathcal{S}()$ is determined as follows:

$$\hat{\mathcal{S}} = \frac{1}{N} \sum_{i=1}^{N} S(\hat{C}(v_i))$$

If $\hat{\mathcal{S}} \geq \lambda$, where λ is a threshold value for cosine similarity, the caption is considered a positive classification.

A. Image Captioning Model Testing

The accuracy of each image captioning model was independently assessed using sklearn metrics. Figure 7 shows the result of comparative accuracy testing for the CLIP, BLIP-2, GIT, and Kosmos-2 models. As can be seen, the

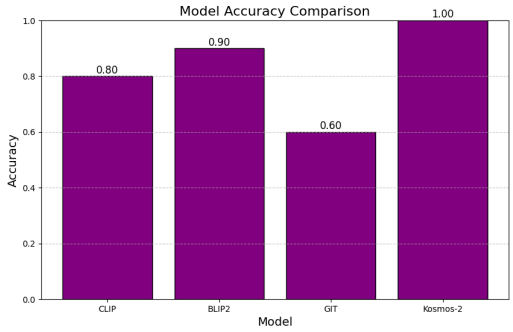

Fig. 7: Accuracy testing of models using sklearn metrics. While Kosmos-2 generated the most detailed and accurate text descriptions, the model was far to slow to be used in a real-time setting, which encouraged the use of BLIP-2 as the primary model.

BLIP-2 model does not achieve the highest accuracy, as it is surpassed by Kosmos-2. However, the BLIP-2 model was still ultimately chosen as the primary image-to-text generator method due to its speed compared to Kosmos-2 and its ranking as the second highest accuracy in image descriptions.

Figure 8 shows the average cosine similarity of the models, for which the range is [-1,1] with 1 as a perfect match and -1 is a perfect opposite. BLIP-2 is the highest ranking, suggesting that despite more instances of the descriptions being correct in the Kosmos-2 model, the overall correctness of the accurate BLIP-2 captions was higher. To illustrate this point, take the following example:

BLIP-2:"a person walking down a dirt path with red flowers"
Kosmos-2:"The image features a woman walking down a dirt path surrounded by a field of red flowers. She is surrounded by the flowers, which are spread across the field. The woman appears to be alone in the scene, as she is the only person visible"
Actual Label:"A dog walks down path with red flowers and a man follows."

While both pass the cosine similarity threshold to be considered correct, BLIP-2 captures more context and has fewer hallucinations making the evaluation more correct.

This result reinforced the decision to use BLIP as the primary model. The fluency and readability of the generated

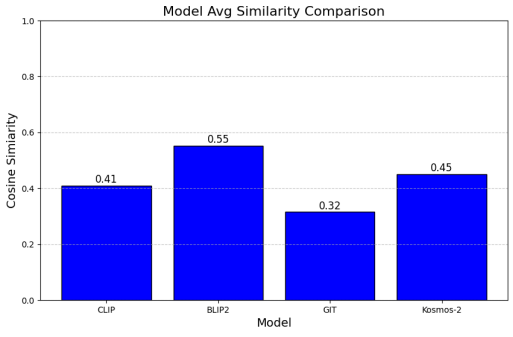

Fig. 8: Averaged cosine similarities using sentence transformers were used to examine the similarity in semantic meaning between actual labels and those generated. Due to the succinctness of the BLIP generated text, the model was able to outperform even Kosmos-2 further emphasizing its value.

text were evaluated using a perplexity score [13], which is a common metric used to evaluate the predictability of the next word in a sequence. It is calculated using the exponential of the cross-entropy loss of a model, with higher perplexity scores indicating greater unpredictability and less grammatically coherent sentence structures. Low perplexity scores indicate more common and fluid sentence structures, which can be an indication that the text is more natural sounding and logical.

To generate perplexity scores, the final captions were tokenized and fed into a pre-trained GPT-2 model to evaluate the loss between the predicted next word and the actual. The exponent of the loss was calculated to produce the final score, which was then averaged across the captions for the full dataset for each model. Figure 9 illustrates that BLIP-2 actually generated the most coherent labels for the images with Kosmos-2 following closely behind. In observation, Kosmos-2 created more detailed and precise descriptions than BLIP-2, but it also experienced more hallucinations and clipped, run-off sentences which is likely what increased its final score. CLIP is not included as it is not a generative model so perplexity scores do not apply.

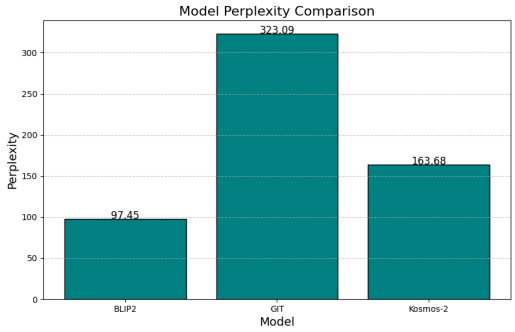

Fig. 9: Fluency and coherency of the generated text in the final summary is evaluated by using perplexity scores from a GPT model. The scores indicate the ease of predicting the next word in a sequence. Higher scores typically signify more random and incoherent text, while lower scores are likely more clear and readable. The BLIP model outperforms all others in readability.

B. 3D CNN and MViT Model Evaluation

All 3D CNN and MViT models used in this project were evaluated independently under a variety of parameter conditions such as with and without normalization, with oversampling of frames while keeping the sequence intact, segmenting the video by adjusting the start and stop times to gather frames, selecting the middle 10 seconds exclusively of a video, and including oversampling and segmentation together. The concept behind the video cropping was to avoid transitional moments in the video under the assumption that majority of frames illustrating the action would occur towards the middle of the video. Figure 10 illustrates the results of a grid search of start and end times of the MViT-v2 model specifically, confirming that the model performs best when shorter segments from the middle portion of the video is

used. The other models showed similar results, but MViT-v2 was the most successful and illustrative.

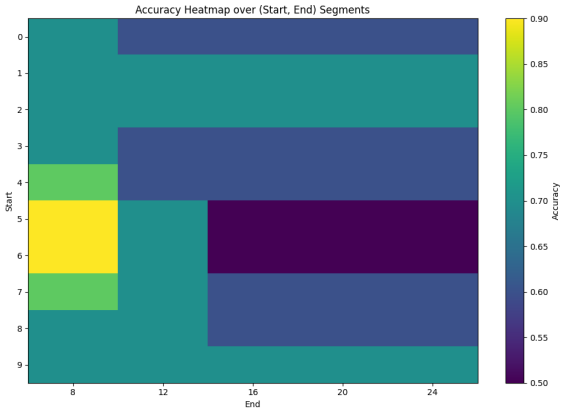

Fig. 10: Averaged accuracy testing of MViT-v2 model over start and stop values ranging from 0-9 and 8-24 respectively of video segmentation testing. Graph illustrates that with smaller segmentation, high accuracy values can be reached.

Using the best parameter setting and outcome for each model, the overall metrics were compared. Figure 11 shows the results of the accuracy testing using sklearn metrics. Unsurprisingly, the MViT-v2 model outperforms the 3D CNN models, likely due to its advancements over 3D CNNs like self-attention or multiscale token hierarchy which both give it the ability to learn longer range temporal relationships as well as improved details and semantic information.

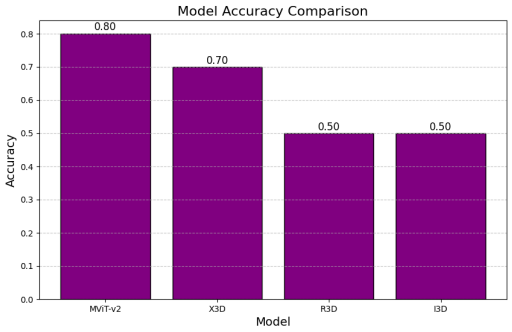

Fig. 11: Accuracy testing of 3D CNN and MViT models using sklearn metrics. MViT performs best, while I3D and R3D perform worst.

Figure 12 contains a graph with the cosine similarity comparison of the MViT and 3D CNN models. Again, the models follow a similar ranking as seen in the analysis of the accuracy. As these are not generative models and include only action states and not details, the expectation is that a tighter bound would be seen on the difference between accuracy and cosine similarity.

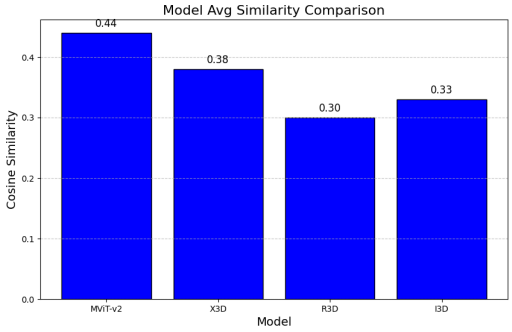

Fig. 12: Cosine similarity of the predicted action labels of 3D CNN and MViT models as compared to the actual labels.

C. BLIP-2 Augmented with 3D CNN

The model that augmented BLIP-2 with MViT-v2, the most successful action model tested, was evaluated and compared to the original BLIP-2 method to determine if improvements were found. Unfortunately, the testing yielded a decrease in accuracy instead of an increase. The primary issue was that the success of the combined model seemed to be bounded by the individual model with the lowest accuracy. As the MViT-v2 model underperformed compared to the BLIP-2 model in individual testing, the combined model's accuracy was reduced to more closely resemble that of the MViT-v2 score.

During observation, it was noted that for instances in which paired classifications were contextually the similar, the cosine similarity remained fairly similar to what would be seen in an isolated BLIP-2 test. However, in cases in which paired classifications were contextually different, more often the MViT-v2 model produced the incorrect classification, thereby reducing the overall efficiency of the overall model. Due to these results, action classifiers were not included in the final model construction.

V. Results

The results of the final structure of the project yielded an improvement in both cosine similarity and readability of the final generation.

A. Accuracy and Perplexity Analysis

Utilizing an identical cosine similarity threshold for accuracy and the same testing set of videos and captions as used in the singular model evaluations, the proposed pipeline structure was tested using BART, DistilBART, FLAN-T5, GPT-3.5, and GPT-4 as possible summarizer models.

Figure 13 provides the results of the summarizer testing in the final pipeline structure. As can be seen, all models except BART produce highly accurate results. While the final results may not capture all elements from the videos, they

succeed in passing the allowed cosine similarity threshold to be considered a correct classification. As compared to the individual model testing they exceed or meet all other accuracy scores. As the proposed pipeline structure uses BLIP-2 for image captioning, the comparison to the individual BLIP-2 model can be made, meaning the structural changes created a 10% increase in accuracy from 90% to 100%. However, its important to note that the bounds used to determine a successful classification were kept very loose, aiming to achieve a *general* description of the video, but allowing a broad range of divergence in the description as compared to the actual caption. Using a stricter bound would likely show a significant decrease in overall accuracy. Regardless of the allowed deviation, the proposed pipeline structure still showed overall improvements over the standard model.

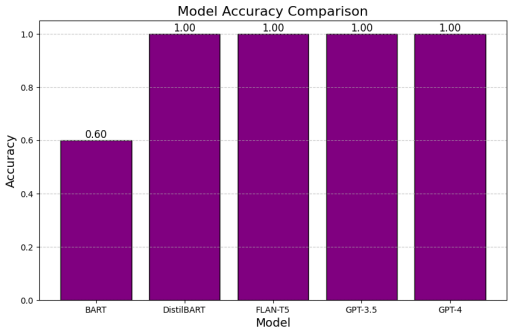

Fig. 13: Accuracy testing of summarizer models in the final pipeline structure.

Figure 14 presents the averaged cosine similarity scores for each of the summarizer models, offering a clearer comparison of performance improvements between the proposed model and the standard BLIP-2 model. In the single model testing, BLILP-2 achieved the highest average cosine similarity score with a score of 0.55. The proposed structure improved upon this baseline score across summarizers with the exception of BART. The most significant improvement is found using GPT-3.5 with a score of 0.63, representing a 14.55% increase in average cosine similarity, very similar to the 10% increase observed in accuracy.

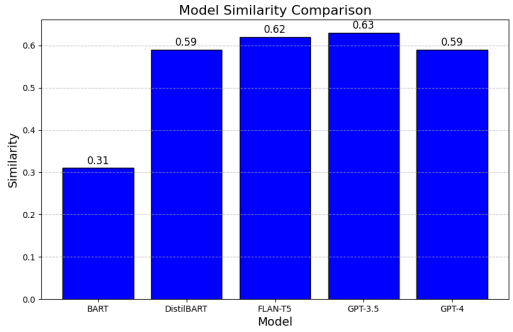

Fig. 14: Cosine Similarity score of final results of summarized text. The use of similarity scores to detect scene changes resulted in about a 14.55% increase in overall semantic similarity using GPT-3.5.

Figure 15 illustrates the result of the perplexity testing on across summary models in the final structure. As compared to the individual BLIP-2 score of 97.45, the GPT-3.5 summary model is 50.85, which is a 47.82% decrease in the perplexity of the generated text. The GPT-3.5 model was chosen as a comparison as it was the most accurate summary model.

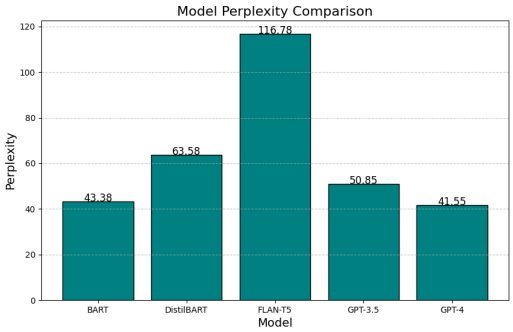

Fig. 15: Perplexity score of final summarized captions. (Lower is better.) The use of a LLM to create a final, succinct summary decreased the perplexity by 47.82%, making the captions more fluid and natural.

B. Transitional Generations of Caption Pairs

The transitional sentence generation with BridgeBart was largely successful, though still occasionally suffered from mild hallucinations, such as the example given in Table II. This table showcases a selection of caption pairs that had significant semantic changes and their generated transitional sentences. In most the transitions are accurate and able to seamlessly conjoin the two captions. In the last example, a gender change occurred from a woman to a man, but was otherwise correct.

An example of a frame pair that would trigger a transitional sentence is seen in Figure 16. While the transition is not perfect due to the initial mislabeling of a dog as a cat, it accurately generates a sentence that manages to capture the essence of the transition.

TABLE II: Examples of BridgeBart semantic transitions between two given caption pairs and using those contexts to generate a reduced yet accurate caption. Note that some still fail and require additional feedback and correction in fine tuning

Caption Pair	Generated Transition
A man is jumping off a cliff into a river *A dog jumping into a body of water*	The dog joins the man.
A young boy wearing a green shirt *A young boy is sitting in a chair*	The boy sits in a chair.
The girl in the spider suit *A woman talking to another woman*	The woman is talking to a man.

(a) Initial Caption: a black cat walking down a dirt path. 1

(b) Initial Caption: a person walking down a dirt road with a dog.

Fig. 16: Example of scene change that triggers a transitional sentence to be generated. The sentence generated was "A person joins the black cat with a dog."

Figure 25 displays the distinction between similar captions illustrated with t-SNE where a high λ of 0.85 is used and the clusters are clearly defined. In contrast, Figure 18 shows the effect of a reduced λ of 0.61 in which the most dominant caption has begun to absorb a similar caption. These graphs show the sensitivity of the threshold value in determining a transition scene in which a transition sentence needs to be generated.

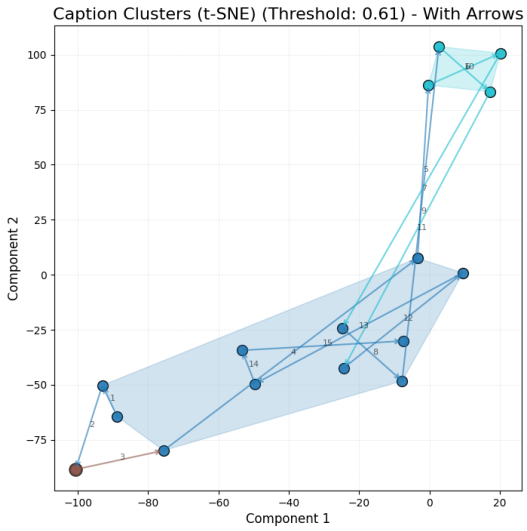

Fig. 17: With $\lambda = 0.61$, the narratively similar captions are still separate however the decreased sensitivity has removed important narrative captions and clustered it with the more dominant expression. The legend can be seen in Figure 24.

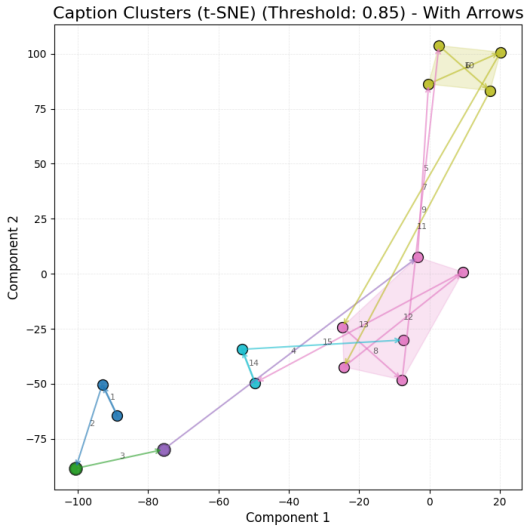

Fig. 18: T-SNE plot of a series of caption embeddings with $\lambda = 0.85$. Notice how sensitive it is to change such that the there are multiple edges connecting two convex hulls suggesting these two are narratively similar. The legend can be seen in Figure 25.

C. Real-Time

Real-time processing was explored under various conditions, including parallelizing video processing, experimenting with multiple input methods, utilizing lightweight models, and making structural changes. However, the only method that yielded a significant improvement was utilizing GPU accelerated computation. By incorporating a GPU, the average

frame processing time across the dataset was reduced from 2.73 seconds with a CPU to 0.24 seconds, resulting in a decrease of 91.35%. Similarly, the overall processing speed decreased by 90.71% from an average of 42.24 seconds to 3.92 seconds. The system specifications of the device the results were discovered on can be viewed on Table III.

TABLE III: System Specifications

Component	Specification
Operating System	Windows 11 Pro (Version 22H2)
Processor	AMD Ryzen 7 3700X 8-Core @ 3.59GHz
Memory (RAM)	32 GB DDR4
Graphics Card	NVIDIA RTX 2060 Super
Storage	1 TB NVMe SSD
System Type	64-bit Operating System, x64-based Processor

In Figure 19, a comparison of total processing times between CPU and GPU computation is presented. The reduction in caption generation time demonstrates that the real-time processing goal was not only met, but exceeded. Given that the average video length used in speed testing was 13.74 seconds, the GPU accelerated caption generation was 71.47% faster than the videos themselves.

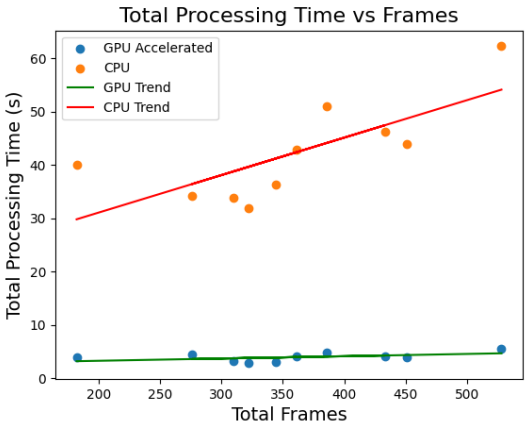

Fig. 19: Comparison of overall computation time between CPU and GPU utilization.

Due to the relatively short duration of the videos used during th testing, the average frame processing time was plotted against the total number of frames per video, as shown in Figure 21. The trend illustrates a negative correlation between average frame processing time and total frame count, which is likely attributed to fixed initial costs, such as loading the video and allocating memory. The fixed cost overhead will have a greater impact on shorter videos and improve the efficiency of processing larger videos, indicating that the current model should be suited to handle longer videos. However, this does not imply that the model will generate captions faster for longer videos, simply that it will handle them with more efficiency on a frame-per-frame basis.

Figure 21 demonstrates a positive correlation between the total time elapsed during processing and frame count. Fortunately, the increase in processing time for longer videos remains roughly linear, indicating that the increase in overall processing time will be predictable and scalable. Taken together, the results suggest that the GPU accelerated model will be adequately be able to generate captions for videos of varying length in real time.

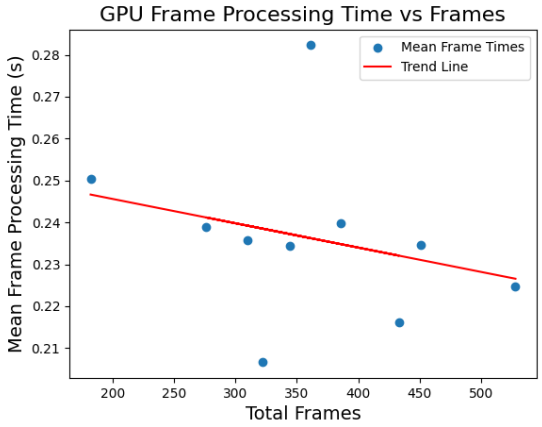

Fig. 20: GPU mean frame processing time plotted against total frames in the video.

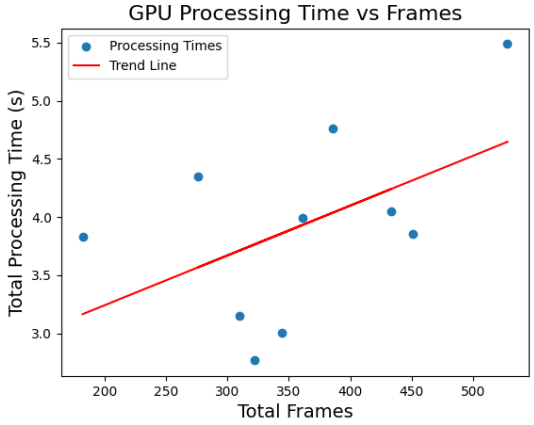

Fig. 21: GPU total processing time plotted against total frames in the video.

VI. Conclusion

This project evaluated viability of existing models for use in video captioning tasks, ultimately identifying BLIP-2 as the most optimal model for real-time caption applications. Through structural experimentation, the project determined that the combined use of image-capturing models with 3D CNNs did not lead to improved accuracy or perplexity. Alternatively, a pipeline structure was proposed utilizing the mode

of frame-by-frame captions, semantic analysis using cosine similarity scores, and a teacher-student framework to generate transition statements for scenes changes. The revised structure was shown to enhance captioning abilities in accuracy by 14.55% using the cosine similarity metric and perplexity by 47.82% as compared to any singular model, proving an effective and beneficial framework. Additionally, leveraging GPUs proved an effective method to allow for real-time applications, decreasing the processing time to an average of 71.47% of the length of the video, combined with the use of light-weight models that reduce storage and computational usage.

VII. Future Work

The following are items of future work proposed to enhance this model:

- Dynamic frame window resizing to adjust for faster paced scenes versus slower ones.
- Dynamically adjust the Summary Model based on the size of the dataset. Since several scenes can be caption rich or caption starved, it should be feasible to design an online deterministic function to adjust accordingly to models that are more data hungry when it may benefit the accuracy of the overall model.
- Incorporate audio into the evaluation to help influence the final captioning.
- Use a more robust teacher model to generate cleaner pseudo-labeled datasets that can then be used for the student/offline distillation process.

VIII. Team Overview

The responsibilities for the project were divided as follows:

Name	Primary Focus	Responsibilities
Kelsey Knowlson	Experimental Testing and Evaluation	• Model benchmarking and parameter tuning • Video frame splicing experiments • Cosine similarity and perplexity evaluations • Combined model testing (BLIP-2 with 3D CNNs) • Visualization and analysis
Matthew Rackley	Model Architecture and Development	• Pipeline and model design • Teacher-Student distillation • Clustering and summarization protocols • BridgeBART fine-tuning • Prompt engineering and dataset creation

TABLE IV: Team roles and task divisions for project development and testing.

IX. Appendix

```
A dog is a dog is running on a dog. An Is A
dog is always is a run.A dog.An Is Is A Dog
is a Dog is is is Is Is Is.A Dog is A Dog Is
```

```
A tin man is holding a book in his hands.
He is holding a page in his book. The book
is open and he is holding the book with his
hands.
```

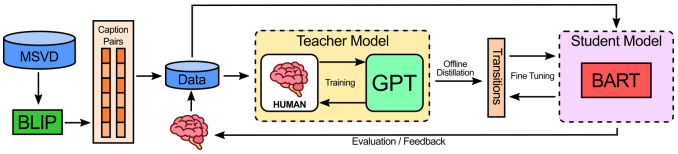

Fig. 22: Overview of the offline distillation pipeline for transition generation. Caption pairs are extracted from MSVD using BLIP and optionally reviewed by humans. A teacher model (GPT) generates transition sentences, which are used as pseudo-labels to fine-tune a student model (BART). The student model learns to generate semantic transitions from caption pairs, enabling efficient inference during downstream video summarization.

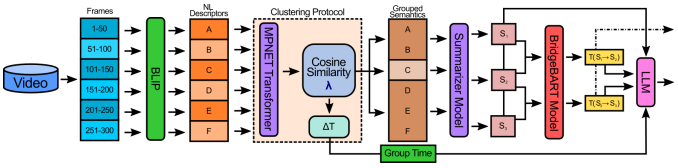

Fig. 23: Diagram of the caption grouping and summarization pipeline. Intermediate narrative transitions produced by the fine tuned BridgeBART can be used directly as real-time output or further refined by an LLM to generate full-scene storytelling. Condensed contextual tokens enable an online method for streaming or sequential processing. A breakdown can be shown in Figures 4, 5 and 6.

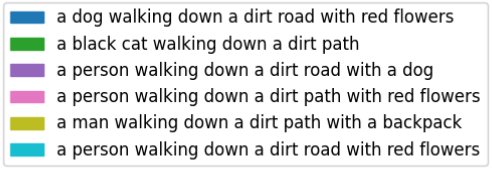

Fig. 24: The legend for the TSNE plot of a story caption. With $\lambda = 85$.

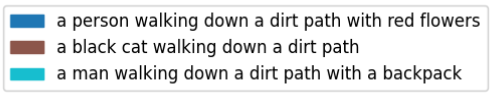

Fig. 25: The legend for the TSNE plot of a story caption. With $\lambda = 61$.

TABLE V: Mathematical Symbol Glossary

Symbol	Meaning
f_t	Frame at time t
c_t	Caption for frame f_t
e_t	Embedding of caption c_t
$\mathcal{C}$	Set of all captions
$\mathcal{G}$	Set of semantic clusters
G_k	Cluster of semantically similar captions
s_k	Summary sentence for cluster G_k
t_k	Transition sentence between s_k and s_{k+1}
λ	Cosine similarity threshold for clustering
Δ_T	Transition generation function
$\mathcal{B}$	BridgeBART transition generator
T, S	Teacher and Student models respectively

References

[1] Peter Abell. *The Syntax of Social Life: The Theory and Method of Comparative Narratives*. Oxford, UK: Clarendon Press, 1987. ISBN: 9780198272717.

[2] Lee Roy Beach. *The Theory of Narrative Thought*. Newcastle upon Tyne, UK: Cambridge Scholars Publishing, 2022. ISBN: 9781527581623.

[3] Tom B. Brown et al. "Language Models are Few-Shot Learners". In: *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*. 2020.

[4] David L. Chen and William B. Dolan. "Collecting highly parallel data for paraphrase evaluation". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. 2011, pp. 190–200.

[5] Hyung Won Chung, Le Hou, and Shayne *et al.* Longpre. "Scaling Instruction-Finetuned Language Models". In: *arXiv preprint arXiv:2210.11416* (2022).

[6] Hugging Face. *BLIP-2 - Hugging Face Transformers Documentation*. Accessed: 2025-04-13. 2024. URL: https://huggingface.co/docs/transformers/main/model_doc/blip-2.

[7] Hugging Face. *Kosmos-2 - Hugging Face Transformers Documentation*. https://huggingface.co/docs/transformers/main/en/model_doc/kosmos-2. Accessed: 2025-04-13. 2024.

[8] GeeksforGeeks. *BART Model for Text Auto Completion in NLP*. https://www.geeksforgeeks.org/bart-model-for-text-auto-completion-in-nlp/. Accessed: 2025-04-13. 2023.

[9] GeeksforGeeks. *Understanding BLIP — A HuggingFace Model*. Accessed: 2025-03-01. 2023. URL: https://www.geeksforgeeks.org/understanding-blip-a-huggingface-model/.

[10] GeeksforGeeks. *Video Classification with a 3D Convolutional Neural Network*. Accessed: 2025-03-01. 2023. URL: https://www.geeksforgeeks.org/video-classification-with-a-3d-convolutional-neural-network/.

[11] David Herman et al. *Narrative Theory: Core Concepts and Critical Debates*. Columbus, OH: The Ohio State University Press, 2012. ISBN: 9780814211861.

[12] Sarthak Jain. *MSVD clips*. https://www.kaggle.com/datasets/sarthakjain004/msvd-clips. Accessed: 2025-04-14. 2024.

[13] Frederick Jelinek et al. "Perplexity – a measure of the difficulty of speech recognition tasks". In: *Proceedings of the 94th Meeting of the Acoustical Society of America* (1977).

[14] Xiaoqi Jiao et al. "TinyBERT: Distilling BERT for Natural Language Understanding". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2020, pp. 4163–4174.

[15] Mike Lewis et al. "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*. 2020, pp. 7871–7880.

[16] Junnan Li et al. "BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models". In: *Proceedings of the 40th International Conference on Machine Learning (ICML)*. PMLR, 2023, pp. 19730–19742.

[17] Junnan Li et al. "BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation". In: *Proceedings of the 39th International Conference on Machine Learning (ICML)*. PMLR, 2022, pp. 12888–12900.

[18] Yanghao Li et al. "MViTv2: Improved Multiscale Vision Transformers for Classification and Detection". In: *arXiv preprint arXiv:2112.01526* (2021). Accessed: 2025-04-13. URL: https://arxiv.org/abs/2112.01526.

[19] Joseph P. Magliano and Jeffrey M. Zacks. "The Impact of Continuity Editing in Narrative Film on Event Segmentation". In: *Cognitive Science* 35.8 (2011), pp. 1489–1517. DOI: 10.1111/j.1551-6709.2011.01202.x.

[20] Scott McCloud. *Understanding Comics: The Invisible Art*. HarperCollins, 1993.

[21] Zhiliang Peng et al. "Kosmos-2: Grounding Multimodal Large Language Models to the World". In: *arXiv preprint arXiv:2306.14824* (2023).

[22] Zhiliang Peng et al. "Kosmos-2: Grounding Multimodal Large Language Models to the World". In: *arXiv preprint arXiv:2306.14824* (2023). URL: https://arxiv.org/abs/2306.14824.

[23] Alec Radford et al. "Language Models are Unsupervised Multitask Learners". In: *OpenAI Technical Report* (2019).

[24] Alexander Ratner et al. "Snorkel: Rapid training data creation with weak supervision". In: *Proceedings of the VLDB Endowment*. Vol. 11. 3. 2017, pp. 269–282.

[25] Antonio M. Rinaldi, Cristiano Russo, and Cristian Tommasino. "Automatic image captioning combining natural language processing and deep neural networks". In: *Results in Engineering* 18 (2023), p. 101107. ISSN: 2590-1230. DOI: https://doi.org/10.1016/j.rineng.2023.101107. URL: https://www.sciencedirect.com/science/article/pii/S2590123023002347.

[26] Michael S Ryoo et al. *BLIP-3-Video: You Only Need 32 Tokens to Represent a Video Even in VLMs*. 2025. URL: https://openreview.net/forum?id=CKYsXi0dOV.

[27] Gerard Salton, A. Wong, and C. S. Yang. "A Vector Space Model for Automatic Indexing". In: *Communications of the ACM* 18.11 (1975), pp. 613–620.

[28] Victor Sanh et al. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter". In: *NeurIPS Workshop on Energy Efficient Machine Learning and Cognitive Computing*. 2019.

[29] Sam Shleifer and Alexander M. Rush. "Pre-trained Summarization Distillation". In: *arXiv preprint arXiv:2010.13002* (2020).

[30] Aadya Singh. *Comparing LLMs for Text Summarization and Question Answering*. https://www.analyticsvidhya.com/blog/2024/11/text-summarization-and-question-answering/. Accessed: 2025-04-13. 2024.

[31] Jianfeng Wang et al. "GIT: A Generative Image-to-Text Transformer for Vision and Language". In: *arXiv preprint arXiv:2205.14100* (2022). URL: https://arxiv.org/abs/2205.14100.

[32] Tongshuang Wu and Daniel S. Weld. "A Survey of Human-in-the-loop Machine Learning". In: *ACM Computing Surveys (CSUR)* 55.4 (2022), pp. 1–39.

[33] Peiyuan Zhang et al. “TinyLlama: An Open-Source Small Language Model”. In: *arXiv preprint arXiv:2401.02385* (2024).