Video Dialog

Joana Wang • Miguel Domingos • João Zarcos
[60225] [60431] [60183]

Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa

May 16, 2025

Table of Contents

Index

1 Introduction

1.1 Project Overview

This project aims to build an intelligent system capable of searching and retrieving specific moments in videos based on queries in natural language. Users can input questions or describe scenes, and the system will analyze the video content to return the most relevant segments.

This system enables precise moment retrieval in videos through natural language queries. It processes videos by segmenting them into meaningful clips and extracting multimodal embeddings that capture textual content. When users submit queries, the system converts them into the same embedding space and performs similarity matching against indexed video segments or compares the query with the indexed video segments based on similarity and frequency of words. The architecture leverages transformer-based models for semantic understanding and OpenSearch for efficient retrieval, combining deep learning with information retrieval techniques for accurate results.

1.2 Video Structure

The video content can be divided and organized into three levels of detail, depending on the respective task. Alternatively, the video can be treated as a complete unit to be indexed and analyzed.

- **Video moments**: capture scenes that are semantically coherent. A video moment may contain different video shots.
- **Video shots**: are visually coherent frames that were captured with one single camera shot.
- **Video keyframes**: are specific frames that are good representations of a temporal range of similar frames.

2 Algorithms and Implementation

2.1 Embedding Representations

In the initial phase of our system for video moment retrieval, we focus on developing a semantic embedding space that enables efficient search and retrieval capabilities. Embeddings serve as the foundational layer for organizing both video metadata and natural language queries into a unified vector space. This approach is crucial for enabling semantic similarity-based search, surpassing traditional keyword-based methods in terms of flexibility and contextual understanding.

We employ dual-encoder architectures to separately encode video moment captions and textual queries into the same embedding space. These encoders are trained to ensure that semantically related video moments and queries are located close together in the embedding space. For this project, we utilize the ActivityNet Captions dataset, which provides a rich structure of video moments aligned with descriptive captions.

To operationalize this, we compute static text embeddings for all video moment captions and persist them using Pickle. Real-time queries are embedded using the same encoder to ensure semantic alignment. All embeddings are indexed using OpenSearch, configured with k-nearest neighbors (k-NN) support for efficient similarity search.

Compared to standard keyword-based search, semantic embedding search provided higher precision in retrieving contextually relevant moments. While keyword search relies on exact matches or bag-of-words similarity, embedding-based retrieval captures paraphrasing, synonyms, and more complex semantic relationships.

The embeddings are generated using transformer-based encoders, known for their ability to capture high-level semantic representations. In this phase of the project, video indexing is performed at the whole video level, rather than at the level of individual moments or keyframes. This approach simplifies the indexing process and provides a solid foundation for exploring the embedding space and implementing semantic retrieval using OpenSearch.

2.2 Large Vision and Language Models

In the project's second phase, the CLIP model, a type of Vision-Language Model, is adopted to understand and compute the semantic similarity between video keyframes and captions within a shared embedding space.

The CLIP model has a dual encoder architecture:

- **Image encoder**: to generate embedding vectors for video keyframes.
- **Text encoder**: to generate embedding vectors for the video captions.
- Image and text are encoded in the common representation space to compute their similarity.

In our implementation, we utilized the *OpenAI* version of CLIP for image and text processing. Instead of treating the videos at the whole level, we extracted the keyframes for each video for this implementation phase. These keyframes are embedded using CLIP's image encoder and paired with the respective caption embedding, generated by CLIP's text encoder. The resulting vectors are stored in OpenSearch for future retrieval.

In addition to CLIP, we integrate the large vision and language model (LVLM), a machine learning model that can learn and process visual and textual information. This project uses and highlights the Llava model, which can understand and answer questions about visual content (images) and textual data. Llava is a large multimodal model that combines a vision encoder (e.g., CLIP) with a large language model, like Vicuna or LLaMA. It is powerful for processing and responding to multimodal inputs for visual and language tasks, including visual question-answering, caption-related image interpretation, and similar image retrieval. By integrating Llava, the system can react with more complex interactions in a semantically reasonable and explainable way.

3 Evaluation

3.1 Dataset Description

The videos were separated into two datasets: a captions dataset and a video dataset. The group curated and filtered these datasets, selecting the 10 videos with the most moments that also had associated captions and valid URLs.

After filtering the videos, the group created index mappings to use in the notebook during the first phase. Two mappings were established:

- **First mapping**: Contains video title (video ID) and description (captions of the video)
- **Second mapping**: Similar to the first, but with the addition of caption embeddings

Individual caption segments were concatenated into a single string before embedding to capture each video's full semantic meaning.

During the second phase, a single index mapping was created for use in the notebook. This mapping combines information from both captions and keyframes to enable similarity search using vector embeddings. The mapping includes the following fields:

- **video_id**: A unique identifier for each video.
- **frame_timestamp**: The timestamp of the corresponding keyframe in the video.
- **caption**: The caption sentence associated with the video moment.
- **caption_vector**: A 768-dimensional embedding of the caption, stored as a `knn_vector` for similarity search.
- **image_clip_vector**: A 768-dimensional embedding of the keyframe image, also stored as a `knn_vector`.

Both the caption and image embeddings use the HNSW (Hierarchical Navigable Small World) algorithm with the FAISS engine, enabling efficient nearest-neighbor search in high-dimensional space. Parameters for the HNSW method include `ef_construction = 256` and `m = 48`, and the similarity metric used is inner product.

This unified index structure allows for multimodal retrieval by enabling comparisons between textual and visual content within each video.

3.2 Baselines

In the first approach, the baseline evaluation consists of text-only semantic similarity and recurring pre-trained language models like BERT. In this case, captions are embedded into a high-dimensional vector space for semantic similarity, which is assessed by the cosine similarity between tokens. The solution is considered efficient for basic text searching, but lacks in the visual field visualization.

In the second phase, the baseline introduces CLIP, a more advanced pre-trained Vision-Language model that embeds both texts and images into a shared semantic space. This approach enables more complex comparisons and associations between the video captions (texts) and video keyframes (images), allowing cross-modal retrieval.

We assume that a well-trained vision-language model should:

- Assign a higher similarity score to caption-keyframe pairs that describe the same moment.
- Produces an interpretable heatmap that highlights the most relevant part of a part for a given caption.

When we compare the second phase baseline with the first phase, we can note a clear evolution in terms of complexity, accuracy, and effectiveness. Then it improves, allowing for richer video-caption queries and searches.

3.3 Results analysis

3.3.1 Phase 1

During the 1st phase, the group analysed several aspects.

Firstly, the group discussed how the embeddings space organize data and allow for specific search. In terms of Data Organization, in semantic search, each document is transformed into a vector and positioned within a high-dimensional space. Documents with similar meanings are placed close together and, in traditional search, documents are stored based on text terms and frequencies, meaning that the index organizes data around keyword occurrence. In terms of Proximity-Based Search, in the embedding space, search is performed by finding documents that are nearest neighbors to the query vector and, in the traditional search, relevance is determined by keyword overlap and frequency, meaning the system does not consider whether two different words or phrases mean the same thing. In terms of Flexibility in Querying, semantic search enables queries to return documents based on the meaning alignment, even if no keywords match directly, but traditional search requires the exact terms to appear in the document to be

considered relevant. In terms of Dimensional Context, the embedding space captures contextual relationships, allowing it to distinguish between similar words used in different contexts, which is an aspect that traditional indexing does not have.

Secondly, the group discussed these 4 relevant topics:

- Contextual Embeddings
- Positional Embeddings
- Self-Attention
- Interpretability

For the topic *Contextual Embeddings*, to gain insight into how the model encodes input tokens at different layers, the group extracted and visualized the contextual embeddings from each hidden layer of a pre-trained BERT-based model. Given a sentence (caption from the Captions dataset), we tokenized the input using the corresponding tokenizer and passed it through the model with the configuration set to output both hidden states and attention weights. This enabled us to access the hidden representations for each token across all layers. We selected individual layers´ hidden states and projected the high-dimensional embeddings into two dimensions using PCA. Each token was then visualized as a point in a 2D space to inspect how token representations evolve across layers. The group produced a multi-layer visualization: a grid of scatter plots (Fig. 1) displaying the token embeddings across all layers, helping to track how the representation of each token changes. Visualizations reveal that token embeddings evolve from dense, non-contextual representations in the initial layer to well-separated, semantically meaningful vectors in deeper layers. Middle layers show emerging context, while deeper layers encode strong semantic distinctions. Notably, some tokens (e.g., content words) shift significantly across layers, whereas special tokens (e.g., [SEP]) remain relatively stable due to their fixed functional roles.

For the topic *Positional Embeddings*, the group performed a similar process from the previous topic, but, instead of having only one sentence, we had two sentences, being one of them the word "hello" repeated 20 times and the other being the word "bye" repeated 20 times too. We intended to analyse how the context and the position of each word would affect the word´s position on the plot. We can see that, in Fig. 2, the tokens are organized in structured arcs, reflecting their relative positions in the sentence.

For the topic *Self-Attention*, the group examined the self-attention mechanism of a transformer cross-encoder and repeated it with a dual encoder.

- **Cross-encoders** jointly encode the query and the sentence, allowing fine-grained token-level interactions through self-attention. Attention heatmaps (Fig. 3) reveal clear semantic and syntactic patterns, with later layers show-

ing strong cross-segment alignment and functional differentiation across heads. Analysing deeper the heatmaps from Fig. 3, at Layer 11, several attention heads (e.g., Heads 0, 1, 2, 4, 5) exhibit strong cross-segment attention across the [SEP] token, aligning question and sentence tokens. The [CLS] token receives broad attention, acting as an information aggregator, while [SEP] tokens help encode segment boundaries. Some heads (e.g., Head 2) focus on semantic alignment, linking question words like "what" to relevant sentence tokens such as "raises" or "hands," and showing signs of entity resolution. Local, syntax-driven attention diminishes in favour of more abstract, semantic reasoning. Specific heads specialize in tasks such as cross-segment matching or positional tracking, while others appear to be underutilized. These patterns reflect a shift from structural encoding to deep semantic processing, guided by the model's relevance-focused objectives.

- **Dual encoders**, by contrast, encode each sentence independently and compare only the final pooled embeddings. Although this approach sacrifices token-level interpretability and attention analysis, it offers significant advantages in efficiency (ideal for retrieval and semantic similarity tasks). The cosine similarity between token embeddings (Fig. 4) reveals some referential and morphological alignment, offering limited interpretability. Analysing the plot in Fig. 4, pronouns such as "he" in both question and sentence show strong alignment, indicating referential consistency. Verbs like "raise" and "raises" exhibit high similarity, reflecting the model's sensitivity to morphological variations. [CLS] tokens are also highly similar due to their role in capturing sentence-level meaning, while punctuation and structural tokens like [SEP] and "." show lower similarity with semantic tokens but often align with each other. These observations suggest that, despite lacking token-level interactions, dual encoders still encode useful semantic signals that contribute to final sentence-level similarity.

The plot in Fig. 5 illustrates the *Interpretability* aspect, showing the total attention that each token received across the 12 layers of the transformer for the sentence: *"He raises his hands feeling victorious."* Each line in the plot represents a token, where the x-axis corresponds to the transformer layers and the y-axis indicates the total attention received. The value at each layer reflects the amount of attention the token received from all other tokens combined, summed across all attention heads. By analysing the plotted lines, we can make the following observations:

1. The `[CLS]` token gathers global context and receives significant attention in the early layers (layers 0–2). This behaviour aligns with its role as the summary embedding of the entire sentence, often used for tasks such as classification or question-answering.
2. The `[SEP]` token gains attention in mid-late layers, which matches with its function as a "separator" to mark the boundary between question and context and manage segment transitions.

3. The plot stands out two context tokens, "hands" and "victorious", that get increased attention in further layers, peaking around 7-9 layers. These two words are considered crucial to answer the implied question, so they're semantically relevant for the model.
4. Some function words (e.g. "did", "he", ".") remain always low attention, reflecting their low semantic contribution.

3.3.2 Phase 2

To capture semantically meaningful associations between video frames and textual descriptions, we adopted a language-vision contrastive learning approach using the CLIP model. CLIP is trained to embed images and text into a shared latent space, such that paired (image, text) inputs are close in that space, while unrelated pairs are far apart. This makes CLIP well-suited for keyframe-level retrieval and alignment tasks in multimodal settings.

In our pipeline, we utilized the pretrained CLIP ViT-L/14 model from HuggingFace Transformers. Video frames were sampled and preprocessed using CLIP's visual encoder, and each moment's visual representation was encoded into a 768-dimensional embedding. Textual queries — in the form of captions or narrations — were tokenized and passed through CLIP's text encoder, producing corresponding language embeddings in the same latent space.

To compute the alignment between a query and a keyframe, we used inner product between embeddings. These similarity scores were then used to rank video segments in order of relevance to the given query. All frame-level visual embeddings were indexed using FAISS for efficient large-scale retrieval.

To better understand the temporal dynamics of contrastive alignment, we visualized the variation in similarity scores across the video timeline for each caption. As shown in Fig. 6, each curve represents the CLIP similarity between a specific caption and video frames. Peaks in these curves indicate moments where the visual content strongly aligns with the corresponding caption, serving as soft indicators for relevant temporal segments. For instance, the caption "Two women enter and do flips" shows distinct peaks around 10 and 90 seconds, suggesting strong alignment with the visual content at those points. In contrast, other captions may exhibit lower or more diffuse similarity values, reflecting weaker or more ambiguous visual grounding. These contrastive similarity curves offer an interpretable means for moment localization and serve as a diagnostic tool to assess caption quality and dataset structure. Captions with localized peaks tend to reflect specific, well-grounded events, while overlapping peaks may indicate redundant or semantically broad descriptions. This visualization aids both in evaluating dataset alignment quality and guiding future refinement efforts.

Moreover, to provide spatial insight into how CLIP interprets visual regions of interest in a given frame with respect to a language query, we utilized attention rollout and gradient-based methods to extract activation maps from the vision transformer. For each video moment, the group selected the keyframe that relates the most with the caption asssociated with that moment. Fig. 7 presents interpretable visual maps for the keyframes gathered earlier, showing the regions that contribute the most to text-image similarity. These attention overlays help reveal the semantic grounding behind the model's retrieval decisions, enhancing transparency and explainability.

The combination of these visualizations provides both temporal and spatial interpretability of the contrastive retrieval process. It demonstrates not only when the video moment is relevant to a query, but also where within each frame the model focuses attention, effectively bridging natural language understanding and visual perception.

To deepen our understanding of how captions align with video content, both temporally and spatially, we present two complementary visualizations: the CLIP similarity heatmap (Fig.8) and the contrastive interpretability matrix (Fig.9).

Figure 8 illustrates the similarity scores between each caption and a selection of keyframes sampled from different timestamps across the video `v_94wjthSzsSQ`. Each cell represents the CLIP similarity between a specific caption and a corresponding frame. Brighter values indicate stronger semantic alignment. This view allows us to compare how each caption performs across different moments in the video. For example, the caption "Two women enter and do flips." exhibits high similarity at both 93s and 8.53s, indicating that the described action occurs at multiple, visually consistent points in time. In contrast, captions with more diffuse or general descriptions produce lower or less localized scores.

Figure 9 complements this analysis with spatial interpretability through attention heatmaps. For each caption–keyframe pair, we visualize the CLIP attention weights over the frame, highlighting the regions most influential in determining the similarity score. These maps reveal which visual cues the model attends to when associating language with vision. For instance, the caption "The jump up in the air and spin." consistently activates regions corresponding to human motion, demonstrating focused visual grounding. Conversely, captions with weaker alignment display sparse or inconsistent activations, suggesting less visual-semantic coherence.

Together, these visualizations offer a rich interpretation of CLIP-based alignment, showing not only when a caption corresponds to a video moment, but also what parts of the image contribute most to that alignment. This interpretability is essential for validating caption quality, diagnosing dataset ambiguity, and guiding future improvements in multimodal retrieval and localization.

4 Idea for Phase 3

In the third phase of this project, we focus on the development of a specialized chatbot designed to interact with and retrieve information from the WISDOM subject materials. The core objective of this phase is to create an intelligent system capable of understanding and simplifying complex educational content derived from lecture materials, primarily distributed in .pptx format.

The chatbot's knowledge base is constructed by extracting both textual and visual information from PowerPoint presentations. Although PDF files were initially considered for content ingestion, .pptx files were ultimately selected due to their superior performance in preserving structural integrity and enabling more efficient parsing of content. This choice ensures a more reliable and consistent data extraction pipeline.

For embedding the extracted data, we utilize OpenAI's CLIP/BLIP model. CLIP/BLIP enables multimodal embedding by jointly processing text and images, thereby capturing the contextual and visual nuances of the educational materials. The generated embeddings are indexed and stored using OpenSearch, an open-source search and analytics engine that supports efficient semantic search capabilities.

User queries are processed by searching the OpenSearch index to retrieve the most relevant content based on semantic similarity. Retrieved information is then passed to a language model, which not only returns an accurate answer but also simplifies the content to ensure better comprehension for students. We are also going to design the application to be able to make summaries of the several contents of the course.

We are thinking of using an LLM like Llama 3.2:3B to generate the responses to the users.

5 Appendix

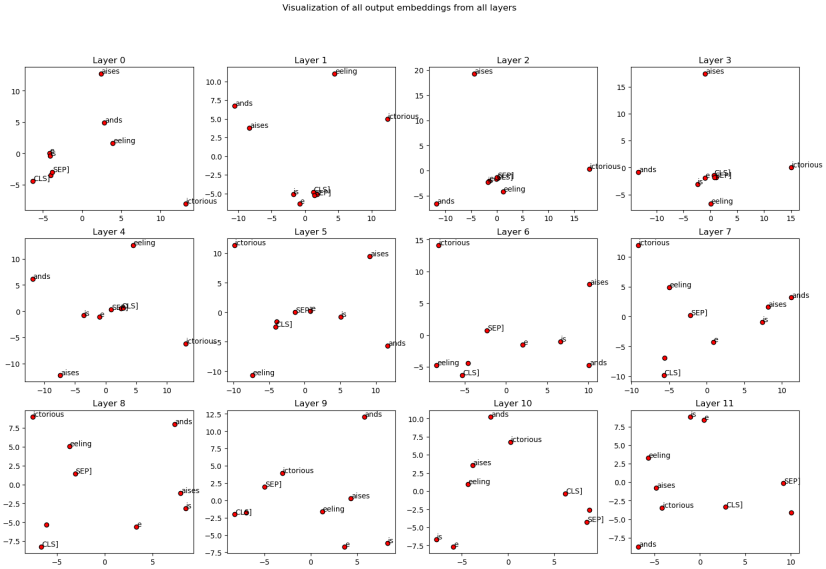

Fig. 1. Visualization of the contextual word embeddings from layer 0 to layer 11

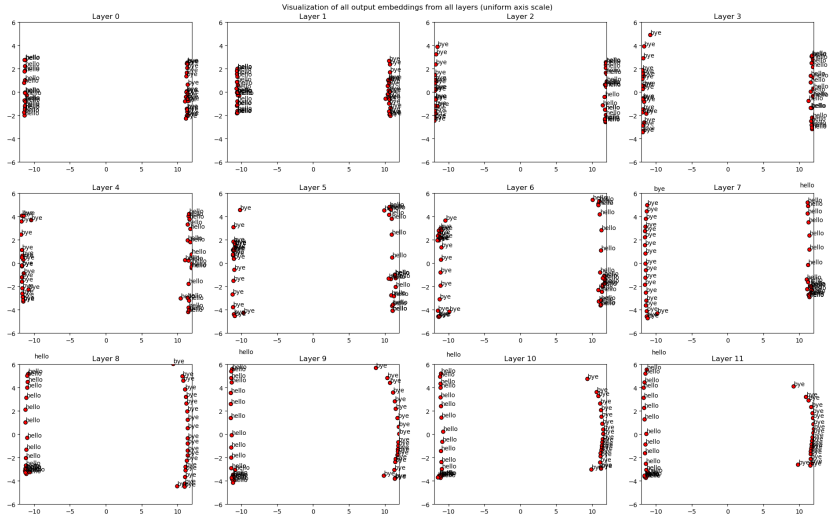

Fig. 2. Visualization of the embeddings and the distance across all tokens formed by 2 sentences, each one with the same word repeated 20 times, from layer 0 to layer 11

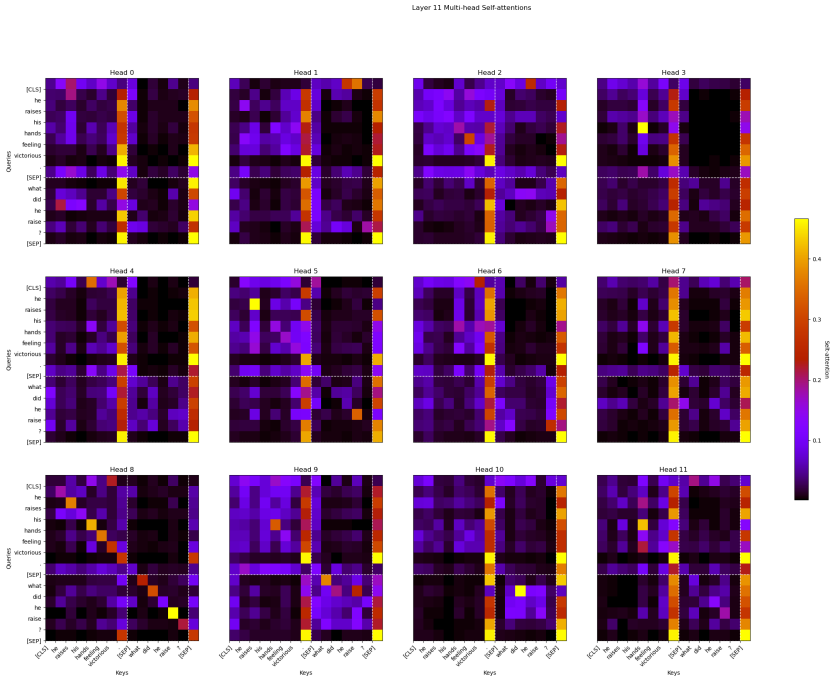

Fig. 3. Multi-head self-attentions analysis from layer 0 to layer 11 - cross-encoder

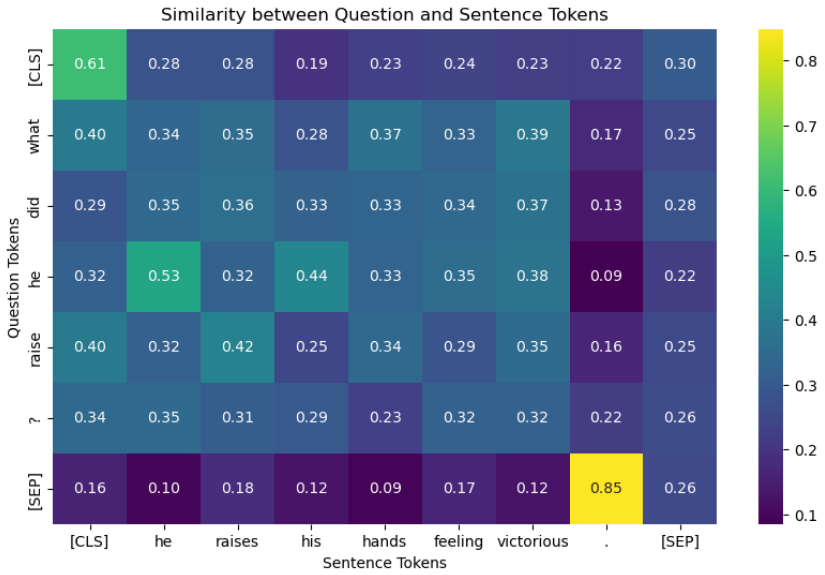

Fig. 4. Similarity comparison between question and answer tokens - dual-encoder

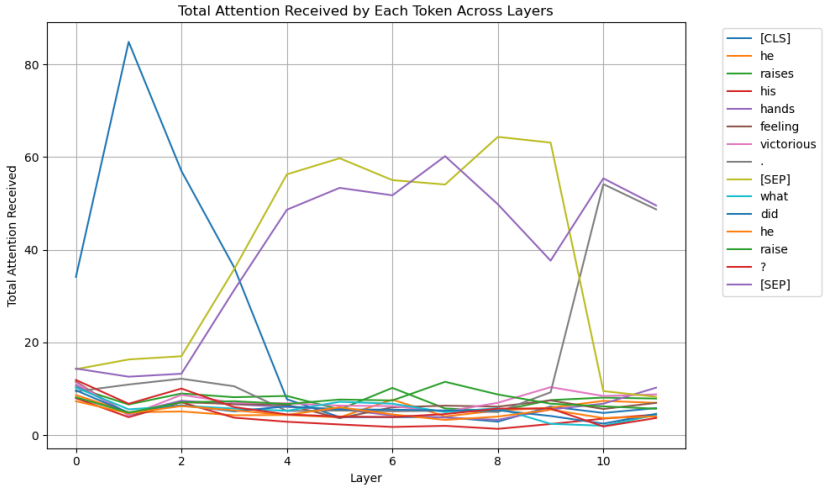

Fig. 5. Visualization of the attention that each token receives on each layer

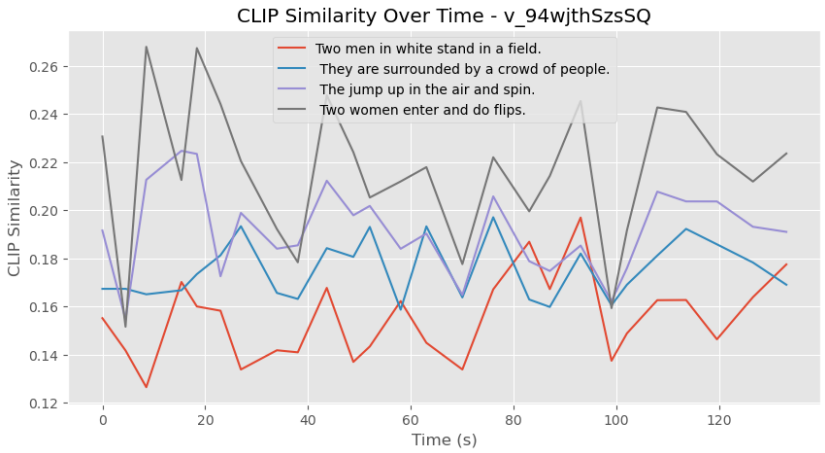

Fig. 6. Visualization of each caption curve variation during the entire video - v_94wjthSzsSQ

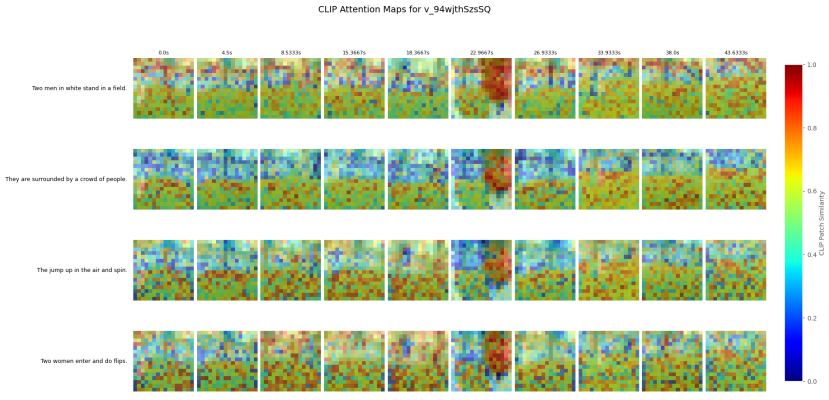

Fig. 7. Visualization of the interpretable visual maps for all moments of the video v_94wjthSzsSQ

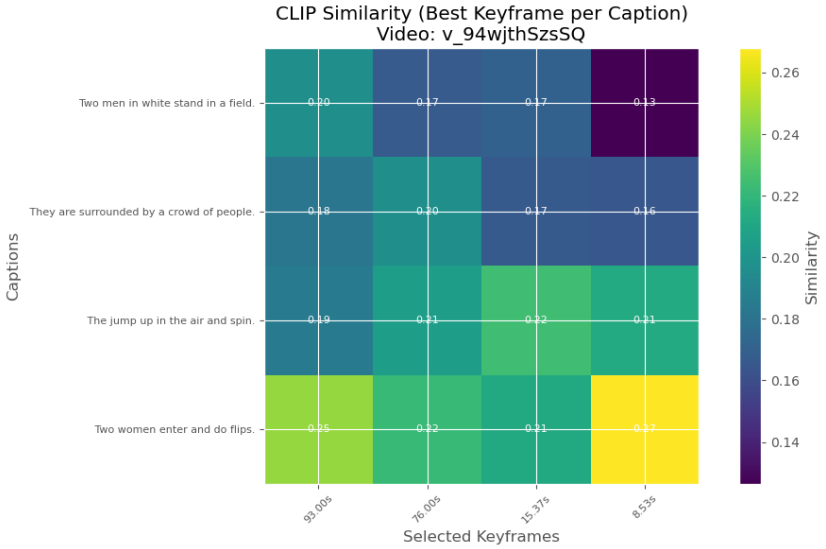

Fig. 8. Contrastive similarity matrix between keyframes and all captions using CLIP for video v_94wjthSzsSQ. Each cell (i, j) represents the similarity score between the keyframe of moment i and the caption of moment j

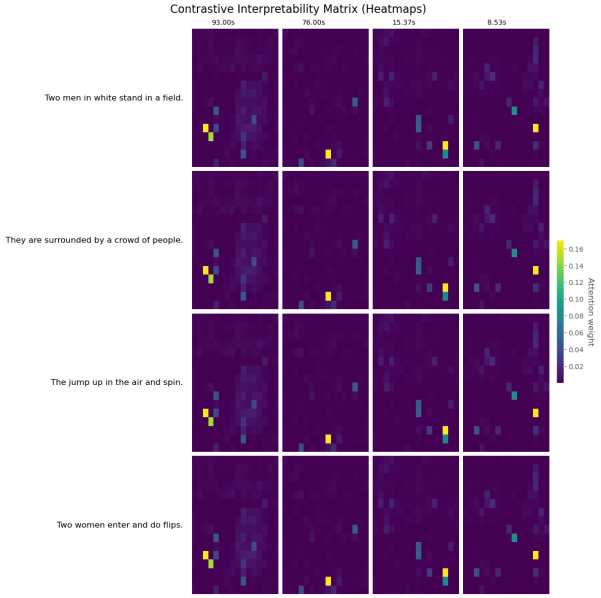

Fig. 9. Interpretable visual heatmap showing the attention between each caption and every frame for video v_94wjthSzsSQ. Each cell (i, j) reflects the visual alignment between the caption of moment i and the frame of moment j

References

1. OpenSearch Project. *OpenSearch documentation*. Retrieved April 15, 2025, from https://opensearch.org/docs/latest/
2. DeepSeek. *DeepSeek: Open-source LLM and search projects*. Retrieved April 15, 2025, from https://www.deepseek.com
3. OpenAI. (2024). *ChatGPT (GPT-4) [Large language model]*. Retrieved April 15, 2025, from https://chat.openai.com
4. Python Software Foundation. *pickle — Python object serialization*. Retrieved April 15, 2025, from https://docs.python.org/3/library/pickle.html
5. Docker Inc. *Docker [Software platform]*. Retrieved April 15, 2025, from https://www.docker.com
6. Transformers *Transformers [Library of pretrained natural language processing]*. Retrieved April 15, 2025, from https://huggingface.co/docs/transformers/en/indexs
7. CLIP *CLIP [Contrastive Language-Image Pre-Training]*. Retrieved April 15, 2025, from https://arxiv.org/abs/2103.00020 and https://github.com/openai/CLIP
8. Llava *Llava [Large Multimodal Model]*. Retrieved April 15, 2025, from https://llava-vl.github.io/