Video Dialog

Joana Wang • Miguel Domingos • João Zarcos

[60225] [60431] [60183]

Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa

April 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.

3 Evaluation

3.1 Dataset description

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

After filtering the videos, the group created index mappings for use in the notebook. 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 the full semantic meaning of each video.

3.2 Results analysis

During this 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 the 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 showing 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.

4 Appendix

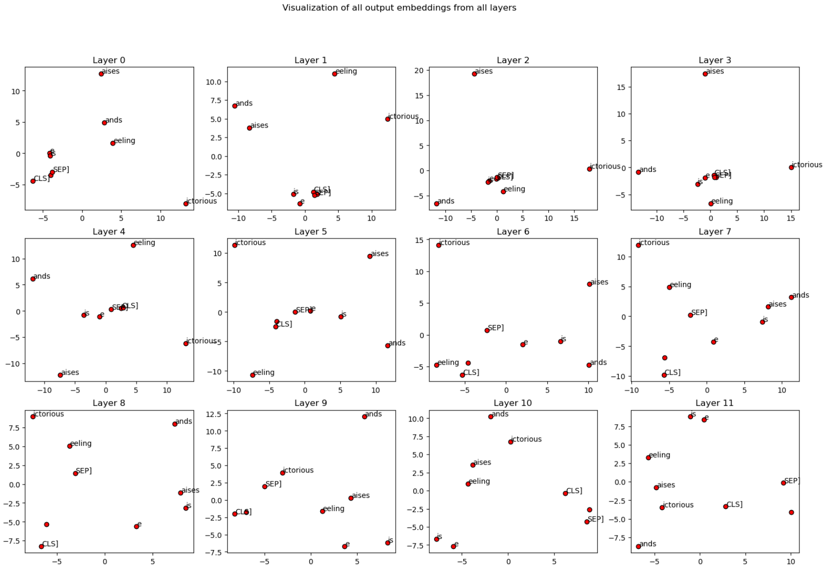

Fig. 1. Visualization of the contextual word embeddings from layer 0 to layer 11

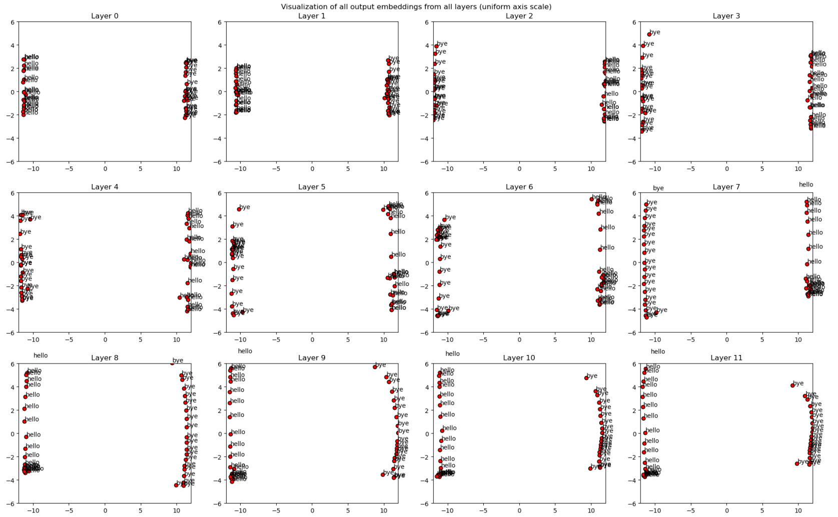

Fig. 2. Visualization of the embeddings and the distance across all tokens formed by 2 sentences, each one with the same word repeated 20 times, from layer 0 to layer 11

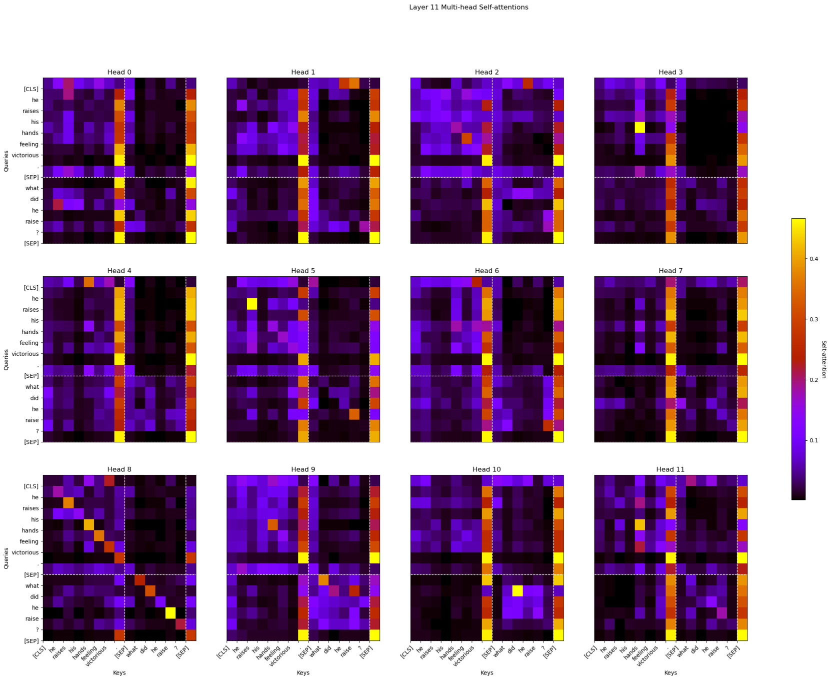

Fig. 3. Multi-head self-attentions analysis from layer 0 to layer 11 - cross-encoder

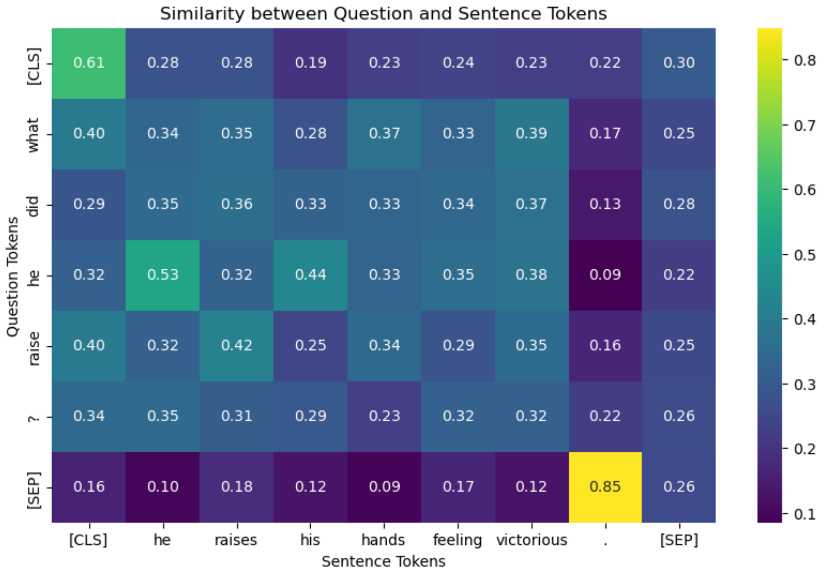

Fig. 4. Similarity comparison between question and answer tokens - dual-encoder

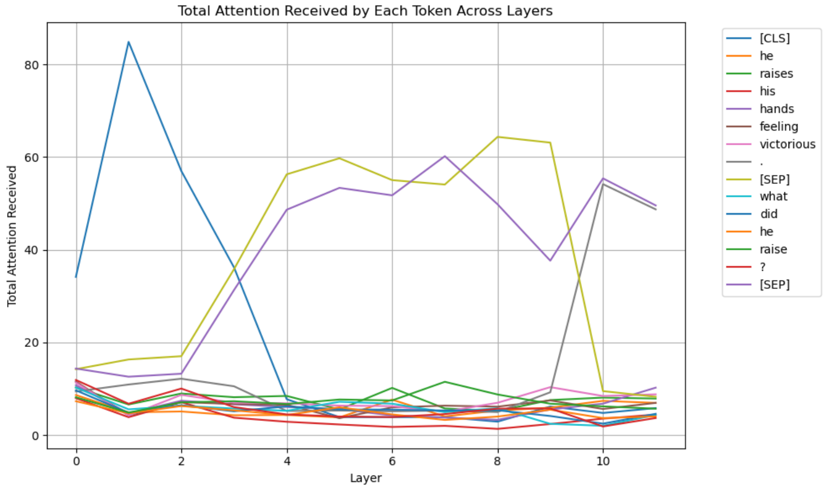

Fig. 5. Visualization of the attention that each token receives on each layer

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