Volume 9, Issue 9, September– 2024 International Journal of Innovative Science and Research Technology
ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24SEP1105

An Efficient Transformer-Based System for Text-Based Video Segment Retrieval Using FAISS

Sai Vivek Reddy Gurram[1]
Independent Researcher

Abstract:- An efficient system for text-based video segment retrieval is presented, leveraging transformer-based embeddings and the FAISS library for similarity search. The sys- tem enables users to perform real-time, scalable searches over video datasets by converting video segments into combined text and image embeddings. Key components include video segmentation, speech-to-text transcription using Wav2Vec 2.0, frame extraction, embedding generation using Vision Transformers and Sentence Transformers, and efficient similarity search using FAISS. Experimental results demonstrate the system's applicability in media archives, education, and content discovery, even when applied to a small dataset.

I. INTRODUCTION

The exponential growth of video content across various platforms necessitates efficient methods for indexing and retrieving relevant segments based on textual queries. Traditional methods often rely on metadata or manual annotations, which are neither scalable nor efficient. Recent advances in transformer-based models for natural language processing and computer vision offer new avenues for automating this process.

In this paper, an integrated system is proposed that utilizes state-of-the-art transformer models to generate embeddings for both the visual and textual content of video segments. By indexing these embeddings using Facebook AI Similarity Search (FAISS), fast and accurate retrieval of video segments in response to textual queries is enabled. The contributions of this work are:

A comprehensive pipeline for video segmentation, transcription, frame extraction, and embedding gen- eration.

An efficient method for combining text and image embeddings to create a rich representation of video segments.

The application of FAISS for scalable and real-time similarity search over video datasets.

II. RELATED WORK

➢ *Video Retrieval Systems*

Previous work in video retrieval has focused on keyword- based search and content-based retrieval using low-level features like color histograms and motion vectors [1]. These methods often lack the semantic understanding required for accurate retrieval.

➢ *Transformer Models in NLP and CV*

Transformers have revolutionized NLP and computer vision tasks. Models like BERT [2] and Vision Transformers [3] have achieved state-of-the-art results in various applications. Wav2Vec 2.0 [4] has shown significant improvements in speech recognition tasks, making it suitable for automatic transcription.

➢ *Similarity Search with FAISS*

FAISS [5] is a library for efficient similarity search and clustering of dense vectors. It has been used extensively for large-scale information retrieval tasks due to its speed and scalability.

III. METHODOLOGY

The system comprises several components that work in tandem to enable efficient video segment retrieval.

➢ *Video Segmentation*

- Videos are divided into fixed-length segments to create manageable units for processing and retrieval.
- Video Loading: Videos are loaded using MoviePy [6], providing access to metadata and content.
- Duration Calculation: The total duration is computed to determine the number of segments.
- Segmentation Loop: The video is iterated over, extracting 30-second clips.
- Subclipping: Each segment is saved as an individual MP4 file using standard codecs.

➢ *Speech-to-Text Transcription*

The audio from each video segment is converted into text transcripts.

- Model Used: Wav2Vec 2.0 ("facebook/wav2vec2- large-960h") [4].
- Processor: Wav2Vec2Processor handles audio pre-processing and decoding.
- Audio Loading: Librosa [7] loads and resamples audio to 16 kHz, as required by the model.

➢ *Frame Extraction*

Key frames are extracted from each video segment to capture visual information.

- Frame Sampling: Six equally spaced frames per segment are extracted.
- Timestamp Calculation: Using np. linspace, timestamps are generated excluding the start and end to avoid redundancy.
- Frame Extraction: Frames are captured using MoviePy's get frame method.
- Saving Frames: Extracted frames are saved as JPEG images using Pillow [8].

➢ *Image Embedding Generation*

Embeddings for each extracted frame are generated using a Vision Transformer.

- Model Used: ViT-B/16 pre-trained on ImageNet [3].
- Preprocessing: Frames are resized and normalized.
- Embedding Extraction: Each frame is passed through the ViT model to obtain a 768-dimensional embedding.
- Aggregation: Embeddings from all frames in a segment are averaged to form a single image embedding.

➢ *Text Embedding Generation*

Embeddings for the transcripts of each video segment are generated.

- Model Used: SentenceTransformer's "all-MiniLM- L6-v2" [9].
- Embedding Extraction: Transcripts are converted into embeddings of size 384.

➢ *Combining Embeddings*

To create a multimodal representation, text and image embeddings are combined.

- Concatenation: The 384-dimensional text embed- ding and the 768-dimensional image embedding are concatenated to form a 1,152-dimensional vector.
- Alignment: Embeddings are ensured to correspond to the correct video segments.

➢ *Indexing with FAISS*

The combined embeddings are indexed using FAISS for efficient similarity search.

- Normalization: Embeddings are L2-normalized.
- Index Type: IndexFlatIP is used for inner product (cosine similarity) search.
- Indexing: All embeddings are added to the FAISS index.

➢ *Similarity Search*

The retrieval process involves querying the FAISS index with a user-provided text query.

- Query Embedding: The query is embedded using the same Sentence Transformer model.
- Normalization: The query embedding is L2- normalized and zero-padded to match the combined embedding size.
- Similarity Search: FAISS retrieves the top-k most similar video segments based on cosine similarity.

➢ *Experiments and Results*

- *Dataset*

The system was evaluated on a dataset comprising 10 minutes of video content, consisting of a single video clip segmented into multiple parts. Although the dataset is small, it serves as a proof of concept for the system's functionality.

➢ *Evaluation Metrics*

Functionality Verification: Ensuring each component of the system works as intended.

- Retrieval Examples: Demonstrating retrieval results for sample queries.
- Performance Metrics: Measuring processing time for indexing and retrieval.

IV. RESULTS

Functionality: All components—including video segmentation, transcription, frame extraction, embedding generation, and similarity search—functioned correctly.

➢ *Retrieval Examples:*

Sample queries returned relevant video segments, indicating that the system can effectively match textual queries to video content.

➢ Processing Time:

- Indexing Time: The small dataset allowed for rapid indexing, taking approximately 2 minutes.
- Query Response Time: Average query response time was under 0.1 seconds, demonstrating real-time performance.

V. DISCUSSION

Due to the limited size of the dataset, quantitative metrics like Mean Average Precision (mAP) are less meaningful. However, the successful implementation and operation of the system on this dataset validate the approach. The system can be scaled to larger datasets for more comprehensive evaluations in future work.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24SEP1105

VI. CONCLUSION

An efficient system for text-based video segment retrieval was introduced, leveraging transformer-based embeddings and FAISS for similarity search. By combining text and image embeddings, a richer representation of video content is captured, leading to effective retrieval performance. The system operates in real-time and demonstrates potential applicability in media archives, education, and content discovery. Future work includes scaling the system to larger datasets and conducting extensive evaluations.

REFERENCES

[1]. C. G. M. Snoek, M. Worring, and A. W. M. Smeulders. Early versus late fusion in semantic video analysis. In Proceedings of the 13th Annual ACM International Conference on Multimedia, pages 399–402, Singapore, 2005.

[2]. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 4171–4186, Minneapolis, MN, 2019.

[3]. Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In International Conference on Learning Representations, 2021.

[4]. Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self- supervised learning of speech representations. Ad- vances in Neural Information Processing Systems, 33:12449–12460, 2020.

[5]. Jeff Johnson, Matthijs Douze, and Herv´e J´egou. Billion-scale similarity search with gpus. IEEE Trans- actions on Big Data, 7(3):535–547, 2019.

[6]. F. Zulko. MoviePy: Video editing with Python, 2015. Zenodo.

[7]. Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Ni- eto. librosa: Audio and music signal analysis in python. In Proceedings of the 14th Python in Science Conference, pages 18–25, Austin, TX, 2015.

[8]. Clark. Pillow (PIL Fork) Documentation, 2015. Python Imaging Library (PIL).

[9]. Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, pages 3982– 3992, Hong Kong, China, 2019.

APPENDIX

➢ *Implementation Details*

- Hardware: Experiments were conducted on a ma- chine with an Intel Core i7 CPU and 16 GB RAM.
- Software: Python 3.8, PyTorch 1.9.0, FAISS 1.7.0.
- Libraries: Transformers, SentenceTransformers, MoviePy, Librosa, NumPy.

➢ *Parameters*

- Video Segment Length: 30 seconds.
- Frames per Segment: 6.
- Embedding Dimensions:
- Text: 384.
- Image: 768.
- Combined: 1,152.