Enhancing Video Moment Retrieval from Text Queries: A GPT-4 Based Approach

Yu Hau (Howard) Chen
howardyh.chen@mail.utoronto.ca

ABSTRACT
This paper presents a novel approach to Video Moment Retrieval (VMR) from Text Queries using Multimodal Large Language Models (MLLMs), specifically GPT-4. Our method strategically selects key frames from videos and generates textual descriptions without requiring extensive training or task-specific models. We demonstrate the viability of this approach through empirical evaluation on the MSR-VTT dataset, comparing it with state-of-the-art benchmarks. The results indicate potential, despite not outperforming top methods. This research contributes to the growing body of work in efficient and accessible VMR methodologies.

INTRODUCTION
Video Moment Retrieval (VMR) from Text Queries represents a significant advancement in multimedia research, aiming to isolate specific segments within videos based on textual descriptions. This technology allows users to search for and retrieve video moments using natural language, making it easier to find specific content in large video datasets. Current approaches in VMR from Text Queries predominantly leverage deep learning techniques for understanding and correlating text queries with video segments [8], [15], [6], [4]. This process typically involves training sophisticated deep neural networks to master tasks such as generating cross-modal embeddings and performing precise temporal localization, thereby bridging the semantic gap between textual descriptions and video content.

Multimodal Large Language Models (MLLMs) have extended the capabilities of traditional text-based models by incorporating an understanding of visual data. This integration allows MLLMs to perform tasks that require simultaneous interpretation of text and images or videos. For instance, in the realm of image description, models like and GPT-4 [14] and LLaVA [11] have demonstrated the ability to generate relevant natural language from images. In the context of VMR, the ability of MLLMs to convert video frames into descriptive text offers a novel avenue for enhancing retrieval processes. Such models could analyze video content and generate text that accurately describes specific moments, facilitating more effective matching with text queries.

In our research, we propose a novel approach utilizing Multimodal Large Language Models (MLLMs) for Video Moment Retrieval. Our method hinges on the strategic selection of key frames from a video, from which MLLMs generate detailed textual descriptions given proper prompting strategy. These descriptions capture not only the content of individual frames but also piece together the overarching narrative or theme of the entire video. Subsequently, this process culminates in the creation of a comprehensive text document for each video. To facilitate retrieval with text queries, we employ semantic search techniques to retrieve the top k most relevant documents. This methodology offers an effective solution for video retrieval, achieving competitive results in a zero-shot context. Importantly, it circumvents the need for task-specific modeling or extensive training, showcasing the potential of MLLMs in simplifying the VMR process.

In the remainder of this paper, we first review the related work to contextualize our approach. We then describe our method in detail, and conclude with the presentation and discussion of our results.

RELATED WORK

Video Moment Retrieval (VMR) from Text Queries
In the domain of Video Moment Retrieval (VMR) from Text Queries, the primary objective is to locate and retrieve video segments that are contextually aligned with a given textual query. Contemporary methods predominantly utilize deep learning models to encode both videos and text into a shared embedding space [20]. Another notable strategy is the dual encoding strategy to enhance the retrieval process [5]. However, a common challenge in these methodologies is the dependency on extensive paired video-text datasets for training, which can be resource-intensive, as outlined by [13]. Our approach diverges from this norm by leveraging pre-trained Multimodal Large Language Models (MLLMs), which obviates the need for model training.

Multimodal Large Language Models
Multimodal Large Language Models (MLLMs) have gained prominence for their ability to process both text and visual information. An example of this is LLaVA [11], which combines a large language model with a visual encoder to handle multimodal data. Similarly, GPT-4 [14] represents a significant advancement in MLLMs, known for its effective integration of text and image processing. In our research, we utilize GPT-4's capabilities via the OpenAI API to analyze video content, leveraging its ability to understand and generate information from both text and visuals.

Information Retrieval
The field of Information Retrieval has seen advancements through the integration of specialized tools and technologies. LangChain [2], for instance, has been recognized in recent

studies for its effectiveness in building language model-based applications, enhancing query processing capabilities. In the realm of embeddings, we utilized OpenAI's text-embedding-ada-002, noted for its ability to transform textual data into semantically rich vector representations. The use of Facebook AI Similarity Search (FAISS) [9] has also been a topic of interest, particularly for its efficiency in managing and querying large vector datasets. Furthermore, the adoption of top k Similarity Search (cosine similarity) helps to retrieve the most relevant documents by measuring vector similarity. These technologies collectively form a foundation upon which our research builds.

METHOD

Figure 1 displays a high-level diagram of our approach. Consider a database of videos, for each video, we first extract frames that contain distinct content. After that, we generate textual description of these frames as a collection so that documents representing the video is generated. Then, we embed all these documents through OpenAI Embedding, and store the embedding vectors using FAISS. Finally, given a text query, we retrieve the top k most relevant document. Below, we describe each component in more detail.

Frames Extraction

The initial stage for processing each video in our database involves extracting frames. A key step involves efficiently handling the repetition of frames often found in video content. To address this, we integrated the content-aware scene detection algorithm from the PySceneDetect package [1]. This tool is utilized for identifying jump cuts within the video, which are indicative of significant scene transitions. By detecting these transitions, PySceneDetect enables us to isolate and extract frames that are adjacent to these cuts, ensuring that the frames selected for further analysis are those most likely to contain distinct content. As the result, we obtained a sequence of frames from the video.

Frames Description

After extracting a sequence of frames from each video, the next step involves generating descriptive content for these frames. To accomplish this, we utilize GPT-4 Vision [14]. We specifically designed a Chain-of-Thought prompt [18] to guide the model's analysis, ensuring a contextually rich interpretation of the video content. The prompt is as follows:

"

Review these multiple images, which represent frames from a scene, and use a chain-of-thought approach to infer the scene's overall theme or story.

- Start by briefly noting any common elements or recurring themes across the images.
- Then, consider what these elements or themes suggest about the video's content. Are there any patterns or consistent messages?
- Finally, based on these observations, synthesize a summary of the video's likely narrative or main topic. Focus on the overarching story the images collectively convey.

"

This structured approach enables GPT-4 Vision to analyze each frame not only individually but also in the context of the sequence, allowing for a more comprehensive understanding of the video's storyline or theme. This provides us a textual description for the video.

Information Retrieval

After obtaining a document for each video, the goal is to retrieve the most relevant document representing a video in response to a text query. To generate embeddings for each documents, we utilize OpenAI's text-embedding-ada-002, leveraging its ability to transform text into high-dimensional vectors that meaningfully capture semantics. Note that the text query is embedded the same way. Efficient management and querying of these embeddings are handled by FAISS (Facebook AI Similarity Search) [9], an optimized library for similarity search and clustering of dense vectors. This is crucial for effectively processing large-scale datasets. The retrieval mechanism is driven by a similarity search that employs cosine similarity. Finally, we return the top k most relevant video-based document.

RESULT

In the following section, we detail the outcomes of our research. We outline the datasets employed, the metrics used to assess performance, and present our findings. These results are contextualized against current state-of-the-art benchmarks

Dataset

The dataset we used for evaluation is the MST-VTT dataset (Microsoft Research - Video to Text) [19]. This dataset is used widely for developing and evaluating algorithms for tasks such as video captioning, video content description, and multimodal learning, where the goal is to bridge the semantic gap between video content and natural language. To evaluate against the state-of-the-art benchmarks, we use 100 video sample of the MSR-VTT 1kA, which is a subset of MSR-BTT used specifically for evaluating video retrieval algorithms. This subset is utilized in the research community to test and benchmark the state-of-the-art methods in video retrieval based on textual queries.

Evaluation Metrics

Our evaluation employs the metric of Recall @ k, which measures the proportion of times the correct video is found among the top k results for a given query. We present our findings using Recall @ 1, Recall @ 5, and Recall @ 10 to indicate the performance of our system at various levels of retrieval granularity.

Comparison

Table 1 shows how our method compares with other video retrieval models on the MSR-VTT leaderboard [16]. Although our results do not surpass the top-performing methods, but still offers a solid starting point. The results show the potential of

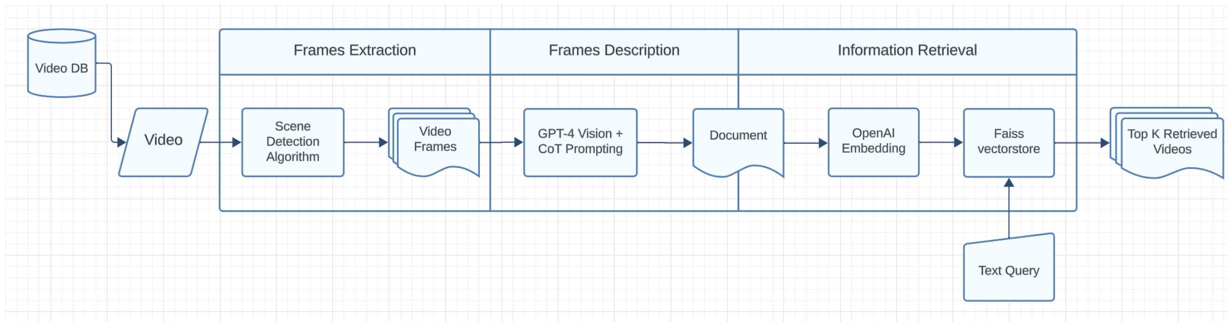

Figure 1. System Flowchart

using Multimodal Large Language Models in Video Moment Retrieval. Notably, our method falls short in Recall @ 10, which may suggest that our model is better at identifying highly relevant videos than at retrieving a wider range of relevant results.

CONCLUSION AND FUTURE WORKS

In conclusion, our study introduces a novel approach to Video Moment Retrieval using Multimodal Large Language Models, particularly leveraging GPT-4's advanced capabilities. Despite not achieving state-of-the-art result, our methodology establishes a robust baseline without the need for specialized model training, demonstrating the efficacy of MLLMs in interpreting and summarizing video content. While our recall rates indicate room for improvement, particularly in broader searches, these findings point to a promising direction for future work:

- Integrating closed captions and exploring the use of audio data to complement the visual information, as dialogues and sounds can offer substantial clues about video content.
- Developing more cost-effective strategies, recognizing the current financial constraints posed by GPT-4 Vision, to make our approach more accessible.
- Experimenting with various prompting strategies to refine the efficiency and accuracy of frame descriptions generated by MLLMs.
- Expanding our evaluation to include a broader range of video retrieval benchmarks, enhancing the robustness and generalizability of our findings.

REFERENCES

[1] 2023. PySceneDetect: Video Scene Cut Detection and Analysis Tool. `https://github.com/Breakthrough/PySceneDetect`. (2023).

[2] Harrison Chase. 2022. LangChain. (Oct. 2022). `https://github.com/langchain-ai/langchain`

[3] Sihan Chen, Xingjian He, Longteng Guo, Xinxin Zhu, Weining Wang, Jinhui Tang, and Jing Liu. 2023. VALOR: Vision-Audio-Language Omni-Perception Pretraining Model and Dataset. (2023).

[4] Ran Cui, Tianwen Qian, Pai Peng, Elena Daskalaki, Jingjing Chen, Xiaowei Guo, Huyang Sun, and Yu-Gang Jiang. 2022. Video Moment Retrieval from Text Queries via Single Frame Annotation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22)*. ACM. DOI:`http://dx.doi.org/10.1145/3477495.3532078`

[5] Jianfeng Dong, Xirong Li, Chaoxi Xu, Shouling Ji, Yuan He, Gang Yang, and Xun Wang. 2019. Dual Encoding for Zero-Example Video Retrieval. (2019).

[6] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. 2017. TALL: Temporal Activity Localization via Language Query. *CoRR* abs/1705.02101 (2017). `http://arxiv.org/abs/1705.02101`

[7] Xingjian He, Sihan Chen, Fan Ma, Zhicheng Huang, Xiaojie Jin, Zikang Liu, Dongmei Fu, Yi Yang, Jing Liu, and Jiashi Feng. 2023. VLAB: Enhancing Video Language Pre-training by Feature Adapting and Blending. (2023).

[8] Ronghang Hu, Huazhe Xu, Marcus Rohrbach, Jiashi Feng, Kate Saenko, and Trevor Darrell. 2015. Natural Language Object Retrieval. *CoRR* abs/1511.04164 (2015). `http://arxiv.org/abs/1511.04164`

[9] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data* 7, 3 (2019), 535–547.

[10] Kunchang Li, Yali Wang, Yizhuo Li, Yi Wang, Yinan He, Limin Wang, and Yu Qiao. 2023. Unmasked Teacher: Towards Training-Efficient Video Foundation Models. (2023).

[11] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual Instruction Tuning. (2023).

[12] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. 2021. CLIP4Clip: An Empirical Study of CLIP for End to End Video Clip Retrieval. (2021).

[13] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. 2020. End-to-End Learning of Visual Representations from Uncurated Instructional Videos. (2020).

Model	Recall @ 1	Recall @ 5	Recall @ 10
VALOR [3]	59.9	83.5	89.6
UMT-L [10]	58.8	81.0	87.1
VLAB [7]	55.1	78.8	87.6
OmniVL [17]	47.8	74.2	83.8
CLIP4Clip-seqTransf [12]	44.5	71.4	81.6
Our Result	56.0	75.0	80.0

Table 1. Comparison of model performance on Recall @ k.

[14] OpenAI, :, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mo Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2023. GPT-4 Technical Report. (2023).

[15] Yingwei Pan, Tao Mei, Ting Yao, Houqiang Li, and Yong Rui. 2015. Jointly Modeling Embedding and Translation to Bridge Video and Language. (2015).

[16] Papers With Code. State of the Art Video Retrieval on MSR-VTT. `https://paperswithcode.com/sota/video-retrieval-on-msr-vtt`. (????). Accessed: 2024-01-10.

[17] Junke Wang, Dongdong Chen, Zuxuan Wu, Chong Luo, Luowei Zhou, Yucheng Zhao, Yujia Xie, Ce Liu, Yu-Gang Jiang, and Lu Yuan. 2022. OmniVL:One Foundation Model for Image-Language and Video-Language Tasks. (2022).

[18] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and

Denny Zhou. 2023. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. (2023).

[19] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. MSR-VTT: A Large Video Description Dataset for Bridging Video and Language. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 5288–5296. DOI: http://dx.doi.org/10.1109/CVPR.2016.571

[20] Ran Xu, Caiming Xiong, Wei Chen, and Jason Corso. 2015. Jointly Modeling Deep Video and Compositional Text to Bridge Vision and Language in a Unified Framework. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 29. AAAI, 2346–2352. DOI: http://dx.doi.org/10.1609/AAAI.V29I1.9512