Multimedia Systems (2024) 30:285
https://doi.org/10.1007/s00530-024-01479-6

REGULAR PAPER

HMTV: hierarchical multimodal transformer for video highlight query on baseball

Qiaoyun Zhang[1] · Chih-Yung Chang[2] · Ming-Yang Su[3] · Hsiang-Chuan Chang[4] · Diptendu Sinha Roy[5]

Received: 13 December 2023 / Accepted: 3 September 2024

Abstract
With the increasing popularity of watching baseball videos, there is a growing desire among fans to enjoy the highlights of these videos. However, the extraction of the highlights from lengthy baseball videos faces a significant challenge due to its time-consuming and labor-intensive nature. To address this challenge, this paper proposes a novel mechanism, called Hierarchical Multimodal Transformer for Video query (HMTV). The proposed HMTV incorporates a two-phase involving Coarse-Grained clipping for candidate videos and Fine-Grained identification for highlights. In the Coarse-Grained phase, a pitching detection model is employed to extract relevant candidate videos from baseball videos, encompassing the features of pitch deliveries and pitching. In the Fine-Grained phase, Transformer encoder and pre-trained Bidirectional Encoder Representations from Transformers (BERT) are utilized to capture relationship features between frames of candidate videos and words from users' questions, respectively. These relationship features are then fed into the Video Query (VideoQ) model, implemented by the Text Video Attention (TVA). The VideoQ model identifies the start and end positions of the highlights mentioned in the query within the candidate videos. Simulation results demonstrate that the proposed HMTV significantly improves accuracy of highlights identification in terms of precision, recall, and F1-score.

Keywords Hierarchical multimodal Transformer · BERT · Highlight query

1 Introduction

Baseball, as a widely followed sport globally, captures the attention of fans who often seek to relive highlights such as strikeouts, catches, or home runs. However, extracting these highlights from each extensive baseball video, which contains several hours, to meet the specific demands of fans can be a time-consuming and labor-intensive task. To address this challenge, video summarization and Video Question Answer (VideoQA) technologies have attracted a large amount of attention. For example, as illustrated in Fig. 1, given a query "The 'Strikeout' highlight," this paper focuses on accurately identifying the start and end frames of the desired highlight that corresponds to the query. The system then returns the localized video segment to the user. This approach enables users to precisely pinpoint and retrieve

Communicated by Xun Yang.

✉ Chih-Yung Chang
cychang@mail.tku.edu.tw

Qiaoyun Zhang
zqyun@chzu.edu.cn

Ming-Yang Su
minysu@mail.mcu.edu.tw

Hsiang-Chuan Chang
149190@o365.tku.edu.tw

Diptendu Sinha Roy
diptendu.sr@nitm.ac.in

[1] School of Computer and Information Engineering, Chuzhou University, Chuzhou 239000, China

[2] Department of Computer Science and Information Engineering, Tamkang University, New Taipei 25137, Taiwan

[3] Department of Computer Science and Information Engineering, Ming Chuan University, Taoyuan City 333, Taiwan

[4] Department of Transportation Management, Tamkang University, New Taipei 25137, Taiwan

[5] Department of Computer Science and Engineering, National Institute of Technology, Shillong 793003, India

Published online: 23 September 2024

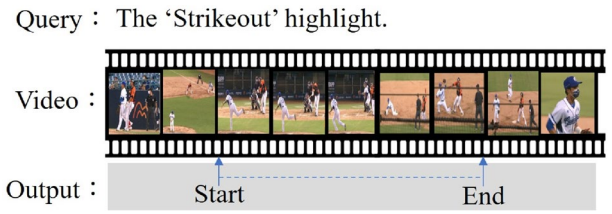


Fig. 1 Illustration of video highlight query on Baseball

specific segments of interest within lengthy videos, significantly reducing the time and effort needed for manual searching.

In recent years, there has been many studies falling in the video summarization category [1, 2, 22, 23]. These studies can be further categorized into unsupervised and supervised methods. The unsupervised methods employed different techniques such as clustering [3, 4], reinforcement learning [5] and adversarial learning [6, 7], to achieve unsupervised video summarization. Despite the remarkable performance achieved by unsupervised methods, they cannot learn from manually created summaries.

To address this problem, supervised learning mechanisms have been proposed [8–10]. These studies explicitly aimed to acquire summarization capabilities based on human-annotated labels. They exhibited superior performance compared to unsupervised learning since they can better capture the high-level semantic knowledge labeled by humans in generating summaries. However, video summarization is a compressed representation of the original video. It may lead to information loss, especially concerning certain details as well as context, and lack of textual information when using video summarization to capture highlights in a baseball video.

Recently, some other researchers have shown significant interest in VideoQA technology, which requires the model to automatically answer questions based on the given videos [19–21, 47–49]. Most existing studies [24, 25] have proposed and validated to tackle short videos with simple and clear semantics. However, VideoQA on long-term videos has rarely been explored. To address this problem, studies [27, 28] extracted question-related long-term videos and predicted answers. However, these methods lacked consideration for Fine-Grained video representation based on the interested behavior and were required to comprehend the content of the videos and provide corresponding answers based on the given questions, rather than directly displaying the videos.

To address the above-mentioned limitations, this paper develops a novel mechanism, called HMTV, which integrates a Coarse-Grained clipping for candidate videos and Fine-Grained identification for highlights. In the Coarse-Grained phase, a pitching detection model is employed to rapidly extract relevant candidate video segments from several hours of raw baseball video. This initial filtering step significantly reduces the computational load for subsequent processing. In the Fine-Grained phase, a Transformer encoder is utilized to capture the temporal relationships between frames in the candidate video segments. Meanwhile, a pre-trained language model, BERT is leveraged to analyze the semantic relationships between words in each user's question. The extracted visual and textual features are then jointly fed into the VideoQ model, which automatically identifies the precise start and end positions of the requested highlights within the candidate segments. The proposed VideoQ, implemented by the TVA, predicts whether the candidate video contains the highlights mentioned in the question and identifies the start and end positions of those highlights by both textual and visual features. The main contributions of the paper are summarized as follows:

(1) Utilizing pitch detection model for efficient analysis of long videos: The proposed HMTV is specifically designed for lengthy and structurally complex sports videos, partitioning videos into fixed-length, non-overlapping segments. It utilizes pitching detection model to accurately identify candidate videos containing potential highlight. It is capable of rapidly extracting relevant segments from hours-long video content and conducting in-depth analysis, significantly reducing computational costs, and enhancing the efficiency and accuracy of highlight extraction.

(2) Integration of BERT and Transformer encoder models for capturing semantic analysis in queries and temporal relationships in video frames, respectively: This paper utilizes a pre-trained BERT model to deeply analyze the semantic relationships in user queries and employs a Transformer encoder to capture the temporal relationships between frames in candidate videos. This approach improves the understanding of contextual information and the accuracy of recognition. The proposed HMTV not only effectively enhances the accuracy of queries but also significantly improves the efficiency of video analysis, especially in accurately locating highlight in long videos.

(3) Adopting VideoQ model for identifying highlights: The proposed VideoQ model employs TVA to integrate user's question and the given candidate video, aiming to enhance the comprehension of the video. Additionally, it further utilizes Multi-Layer Perceptron (MLP) to capture relationships between frames, leading to a more comprehensive understanding of temporal information in candidate videos. The proposed VideoQ model effectively distinguishes subtle differences between similar behaviors, enhancing the model's sensitivity to behavior details and improving the accuracy of highlight identification.

The remainder of the paper is organized as follows. Section 2 discusses and compares previous relevant work. Section 3 describes the assumptions and problem descriptions. Section 4 presents the detailed proposed HMTV mechanism. Section 5 provides the performance study. The summary and future work are discussed in Sect. 6.

2 Related work

In this section, the related literature concerning video summarization and VideoQA are reviewed.

2.1 Video summarization

Video summarization approaches have been divided into unsupervised and supervised based on the perspective of the learning model.

Unsupervised methods utilized various techniques, including clustering [3, 4], reinforcement learning [5] and adversarial learning [6, 7], to accomplish video summarization without the need for labeled data. Rabbouch et al. [3] proposed a pattern recognition system that employed an unsupervised clustering algorithm with the objective of detecting, counting and recognizing a few dynamic objects crossing a roadway. Zhao et al. [4] proposed an unsupervised video summarization method by motion-based frame selection and clustering validity indexes to determine the optimal representatives of the original video. These approaches primarily utilized visual features at a low level and information related to motion. Although they have demonstrated satisfactory performance, they faced challenges in effectively handling videos with complex scene clutters and varying illumination conditions. To address these challenges, Zhou et al. [5] introduced a reinforcement learning-based framework for training a deep summarization network (DSN) designed for video summarization. The DSN predicted the probability of each video frame. It then utilized these probability distributions to select frames and generate video summarizations. However, its performance was constrained by the challenge of minimizing information loss between the summarized and original videos. Yuan et al. [7] proposed an unsupervised video summarization by developing a cycle-consistent adversarial Long Short-Term Memory (LSTM) architecture to effectively reduce the information loss in the summary video. However, they were unable to learn from manually created summaries.

To overcome this limitation, supervised approaches [8–18, 36] have shown significant advancements, leveraging summaries crafted by human-annotated labels. Li et al. [8] presented a summarization framework for both edited and raw videos. The framework designed four models to capture key properties, including importance, representativeness, diversity, and storyness. It built a comprehensive score function, where the weights for the four models are learned in a supervised manner for both edited and raw videos. Sharghi et al. [9] enhanced supervised video summarization with sequential determinantal point processes, addressing exposure bias and incorporating user-input for summary length. However, nonlinear relationships may be overlooked in complex data. To address this limitation, studies [12–18] employed deep learning-based approaches to extract video summarization. For example, Ji et al. [13] proposed attentive encoder-decoder networks for video summarization framework, which employed a bidirectional LSTM for encoding contextual information and two attention-based LSTM networks in the decoder. Zhu et al. [14] proposed a multiscale hierarchical attention method, leveraging both short-range and long-range temporal representations. This approach integrates analysis at the frame, block, and video levels, extending into a two-stream framework to effectively incorporate both appearance and motion information. Liu et al. [36] combined traditional machine learning and deep neural networks to distinguish content sections from non-content sections across various online sports streaming. Furthermore, Zhao et al. [15] introduced a hierarchical structure with a graph convolution network (GCN), capturing global dependencies. However, these methods seldom considered the inter-frame correlations, resulting in the loss of information, especially in specific details and context, during video summarization for baseball highlights.

To address these problems, other researchers [37–43] have adopted Transformer models for various video analysis and summarization tasks, as Transformer models have significantly influenced the field of video processing. Dosovitskiy et al. [37] introduced the Vision Transformer (ViT), a pioneering model that adapts the Transformer architecture, originally designed for natural language processing, for image processing tasks. The ViT distinguished itself thanks to numerous groundbreaking features. The ViT leveraged the Transformer architecture's self-attention mechanism, which enabled ViT to aggregate information from different patches based on their relevance to each other, considering the global context of the entire image. To further consider the spatial–temporal correlations between frames, Ahn et al. [38] utilized a spatial–temporal cross attention Transformer for human action recognition. Hsu et al. [39] and Zhao et al. [40] proposed a spatial–temporal video Transformer, which learned the spatial and temporal correlations from the full set of video summaries to achieve state-of-the-art performance. However, these models might perform poorly in handling long videos and highly dynamic scenes due to their reliance on strong correlations between consecutive frames.

Liang et al. [41] implemented a fusion of Convolution Neural Network (CNN) and Transformer along with adaptive frame scheduling to enhance video semantic segmentation. Meanwhile, Pang et al. [42] developed the Skeleton-Transformer (SkT) for predicting future pose elements in video frames and assessing anomalies by comparing the predicted pose elements with their expected values. However, the SkT overlooked the topological connections among key points. To overcome this challenge, Cheng et al. [43] combined the capabilities of Transformer and GCN for human pose estimation in the analysis of sports technique. The Transformer analyzed the sequential relationships among feature maps, while the GCN was tasked with modeling the topological structure among key points, thus ensuring precise positioning and effective learning of feature representations. However, the skeletons of multiple individuals might overlap or intermingle in crowded environments, making it difficult for the model to accurately distinguish the behaviors of everyone. Additionally, the angle of video capture could also affect the accurate recognition of skeletons, especially when the camera position resulted in partial obstructions or captured from the side, which might lead to misunderstandings or loss of critical behavioral information.

Different from previous studies, the proposed HMTV initially utilizes a pre-trained BERT model to analyze the relationships between words in user queries, ensuring a deeper understanding of the context of each question. Then, it employs the Transformer encoder to capture temporal relationships between frames in candidate videos, enhancing its ability to discern subtle changes and patterns over time. Finally, it integrates the VideoQ model with extracted question and visual features to precisely predict the start and end positions of baseball highlights without overlooking details. The proposed HMTV ensures that no critical details are overlooked, thereby significantly improving the accuracy of the highlight detection.

2.2 VideoQA

Recently, VideoQA has captured public attention, involving models that automatically answer questions based on provided videos [19–28]. Chu et al. [21] and Xue et al. [22] introduced temporal attentions on question and video, aiming to learn effective joint question-video representation. However, they overlooked the spatial information within video frames. Zha et al. [23] proposed a spatiotemporal-textual co-attention network to learn effective question-video representation for video question answering. Zhao et al. [24] proposed a multi-stream hierarchical attention context network for multi-turn video question answering, modeling sequential conversation context and incorporating spatial–temporal attention. This facilitated effective reasoning and precise answer prediction but was limited to short video clips.

To address the challenges of long-term video analysis, Yu et al. [27] proposed compositional attention networks with two-stream fusion for long VideoQA. Their approach employed a two-stream encoder, while the decoder incorporated a compositional attention module to integrate features. Similarly, Wang et al. [28] proposed a matching-guided attention model, which jointly extracted question-related long video snippets and predicted answers in a unified framework. Although these models have achieved remarkable results, they primarily focused on general video content understanding and answering questions rather than providing detailed video representations centered on specific behaviors. Consequently, these models fell short of directly highlighting key moments within the videos.

Additionally, to better capture complex spatiotemporal relationships, several studies [50–52] have explored significant advancements in video modeling using Transformer-based models. For example, Yang et al. [50] proposed a hierarchical visual Transformer network to address domain generalization issues in visual tasks, mitigating domain shift phenomena. Han et al. [51] introduced a Bi-branch complementary network for text-video retrieval, which bridged the gap between text and video modalities by combining local spatiotemporal relationships and global temporal information. Li et al. [52] developed the redundancy-aware Transformer for VideoQA. Therefore, this paper leverages Transformer models to capture spatiotemporal information, enhancing the overall performance.

To address the above limitations, the proposed HMTV employs a pitching detection model to extract fixed-length candidate videos from lengthy baseball videos, capturing the specific behavior of pitching. It employs VideoQ and MLP models to predict the start and end positions of the highlights mentioned in the query within the candidate videos. The proposed HMTV effectively identifies subtle differences in behaviors, enhancing sensitivity to details and improving the accuracy of highlight identification.

3 Notations, assumptions and problem descriptions

This section introduces the notations, assumptions, problem descriptions and objective of our study.

3.1 Notations and assumptions

Assume that a baseball video consists of m continuous frames, represented by $\Phi = (f_1, f_2, ..., f_m)$. Let $f_i \in \Phi$

denote the i-th frame. Let $\widehat{H} = (\widehat{H}_1, \widehat{H}_2, ..., \widehat{H}_h)$ and $H = (H_1, H_2, ..., H_h)$ denote a set of predicted and labeled h highlights extracted from the videoΦ, respectively. Let η_j denote the Intersection over the Union (IoU) of the j-th highlight. The value of η_j can be calculated by the Exp. (1).

$$\eta_j = \frac{H_j \cap \hat{H}_j}{H_j \cup \hat{H}_j}, \tag{1}$$

where $\widehat{H}_j \in \widehat{H}$ and $H_j \in H$ corresponds to specific predicted and labeled highlights, respectively.

3.2 Problem descriptions

Let $TP_j^{\mathcal{M}}$ denote true positive of the j-th highlight predicted by mechanism $\mathcal{M}$. The value of $TP_j^{\mathcal{M}}$ can be derived by Eq. (2).

$$TP_j^{\mathcal{M}} = \begin{cases} 1, & \eta_j \geq \eta^{TP}, \\ 0, & otherwise, \end{cases} \tag{2}$$

where η^{TP} denote the true positive threshold of IoU.

Let $FP_j^{\mathcal{M}}$ and $FN_j^{\mathcal{M}}$ denote false positive and false negative of the j-th highlight predicted by mechanism $\mathcal{M}$, respectively. The value of $FP_j^{\mathcal{M}}$ and $FN_j^{\mathcal{M}}$ can be derived by Eqs. (3) and (4), respectively.

$$FP_j^{\mathcal{M}} = \begin{cases} 1, & \frac{(H_j \cup \hat{H}_j) \setminus H_j}{H_j \cup \hat{H}_j} \geq \eta^{FP} \\ 0, & otherwise \end{cases}, \tag{3}$$

$$FN_j^{\mathcal{M}} = \begin{cases} 1, & \frac{(H_j \cup \hat{H}_j) \setminus \hat{H}_j}{H_j \cup \hat{H}_j} \geq \eta^{FN} \\ 0, & otherwise \end{cases}, \tag{4}$$

where η^{FP} and η^{FN} denote the false positive and false negative threshold of IoU, respectively.

Let $TP^{\mathcal{M}}$, $FP^{\mathcal{M}}$ and $FN^{\mathcal{M}}$ denote the true positive, false positive, and false negative of h highlights in Φ applying $\mathcal{M}$, respectively. The value of $TP^{\mathcal{M}}$, $FP^{\mathcal{M}}$ and $FN^{\mathcal{M}}$ can be derived by Eqs. (5), (6) and (7), respectively.

$$TP^{\mathcal{M}} = \sum_{j=1}^{h} TP_j^{\mathcal{M}}, \tag{5}$$

$$TP^{\mathcal{M}} = \sum_{j=1}^{h} FP_j^{\mathcal{M}}, \tag{6}$$

$$FN^{\mathcal{M}} = \sum_{j=1}^{h} FN_j^{\mathcal{M}}. \tag{7}$$

Let $Pre^{\mathcal{M}}$ and $Recall^{\mathcal{M}}$ denote the precision and recall by applying mechanism $\mathcal{M}$, respectively. The values of $Pre^{\mathcal{M}}$ and $Recall^{\mathcal{M}}$ can be calculated by Eqs. (8) and (9), respectively.

$$Pre^{\mathcal{M}} = \frac{TP^{\mathcal{M}}}{TP^{\mathcal{M}} + FP^{\mathcal{M}}}, \tag{8}$$

$$Recall^{\mathcal{M}} = \frac{TP^{\mathcal{M}}}{TP^{\mathcal{M}} + FN^{\mathcal{M}}}. \tag{9}$$

Let $F1^{\mathcal{M}}$ denote $F1$-score by applying mechanism $\mathcal{M}$. The value of $F1^{\mathcal{M}}$ can be calculated by Eq. (10).

$$F1^{\mathcal{M}} = \frac{2 * Pre^{\mathcal{M}} * Recall^{\mathcal{M}}}{Pre^{\mathcal{M}} + Recall^{\mathcal{M}}}. \tag{10}$$

3.3 Objective

Assume that there is a set of potential mechanisms, denoted as Ω, which are employed to identify the highlights. The main goal of this paper is to find the most effective mechanism $\mathbb{M}$, which satisfies the Eq. (11).

3.3.1 Objective function

$$\mathbb{M} = \underset{M \in \Omega}{argmax} F1^{\mathcal{M}}. \tag{11}$$

To accomplish the mentioned objective function, it is crucial to satisfy the following constraints. Let β_i denote whether the frame f_i is a highlight. That is

$$\beta_i = \begin{cases} 1, & if\ f_i\ is\ a\ highlight, \\ 0, & otherwise. \end{cases}$$

Let C denote the set of categories and C_k denotes the category of highlight H_j. Let γ_k denote whether the category C_k is in C. That is

$$\gamma_k = \begin{cases} 1, & if\ C_k \in C, \\ 0, & otherwise. \end{cases}$$

The following continuous spatial constraint, which is described in Eq. (12), ensures that there is at least one highlight in the video Φ.

(1) Continuous spatial constraint

$$\exists V_j = (f_a, f_b) \subseteq \Phi \text{such that} \quad \prod_{i=a}^{b} \beta_i = 1, \tag{12}$$

where f_a and f_b $(a \leq b)$ denote the frames in Φ.

The following category constraint, as shown in Eq. (13), guarantees that each highlight $H_j \in H$ extracted from video Φ belongs to some specific category $C_k \in C$.

(2) Category constraint

$$\prod_{k=1}^{h} \gamma_k = 1. \tag{13}$$

4 The proposed HMTV mechanism

This paper proposes a highlight identification mechanism, called HMTV, which aims to identify highlights moments in baseball videos, including strikeout, successful catching, and home run instances. As shown in Fig. 2, the proposed HMTV consists of two phases: *Coarse-Grained clipping for candidate videos* and *Fine-Grained identifying for highlights*. A complete and interesting highlight typically involves a pitcher's pitching. Consequently, in the initial phase, the proposed HIM adopts a pitching detection model to extract candidate videos, isolating segments with the potential for containing highlights. The second phase utilizes a VideoQ model to precisely identify the starting and ending points of the highlights in the input videos that correspond to the questions posed by the users.

4.1 Coarse-grained clipping for candidate videos

In this phase, the main objective is to extract relevant segments from a given baseball video, aiming to create a candidate video that potentially contains highlights. It consists of three steps: input video partitioning for draft videos, generating numerous questions, and pitching detection for candidate videos. The following provides the details of each step.

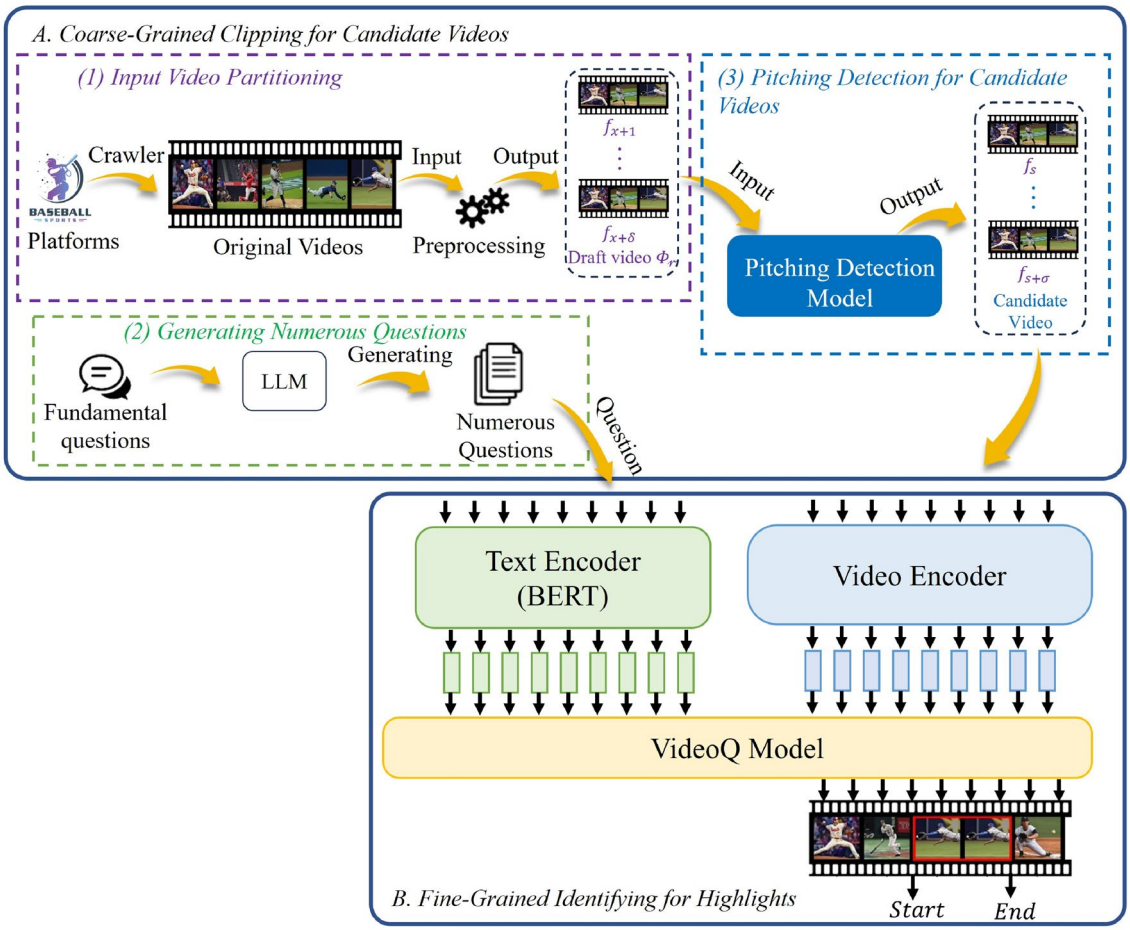


Fig. 2 The architecture of the proposed HMTV

4.1.1 Input video partitioning

Due to certain limitations on the data input length of the model, this step partitions the baseball video into several non-overlapped videos with a fix length. Each segment represents a draft video. Recall that a baseball video consists of m continuous frames, represented by $\Phi = (f_1, f_2, ..., f_m)$. Let δ denote the fixed duration of clipping in frames. The proposed HIM partitions the video Φ frame by frame with δ frames as a unit, generating numerous draft videos. If there are fewer than δ frames, padding is applied to fill the gap. Let α denote the number of draft videos. The value of n can be denoted by Exp. (14).

$$n = \lceil \frac{m}{\delta} \rceil. \tag{14}$$

Let $\Phi = (\Phi_1, \Phi_2, ..., \Phi_n)$ denote the n partitioned videos. Each $\Phi_\varkappa \in \Phi$ contains δ frames and let$\Phi_\varkappa = (\Phi_\varkappa^{start} = f_{x+1}, \ldots, \Phi_\varkappa^{end} = f_{x+\delta})$. The value of f_{x+j} can be derived from Exp. (15).

$$f_{x+j} = f_{(\varkappa-1)*\delta+j}. \tag{15}$$

4.1.2 Generating numerous questions

The input to the proposed VideoQ model consists of two parts: a user's question, which can indicate the highlight of interests and a video. Through these two inputs, our model will output the start and end positions of the highlight in the video. In the baseball video, subtitles are important features that can be used to better understand the contents of the video. The proposed VideoQ model searches for highlights that contain catching, strikeouts, and home runs.

To enhance both the diversity and volume of training data, this paper employs a Large Language Model Llama [53] to automatically generate many questions with equivalent meanings. Specifically, the Llama ensures the relevance of the generated questions to the target content. During the generation process, the proposed HMTV applies a maximum length constraint of θ words to each question. The Llama is prompted with a set of initial "seed" questions, which are manually crafted to cover the key aspects of the video content. From these seeds, the Llama generates paraphrased questions that maintain the same semantic meaning but vary in phrasing and structure. This process significantly expands the training dataset and improves the model's ability to generalize to different question formulations. Let $Q = \{q_1, q_2, ..., q_\}\}$ denote the original collected questions, where $\}$ denotes the number of collected questions. Each question q_i can generate ρ new questions through LLM. That is

$$q_i = \{q_{i,1}, q_{i,2}, \ldots, q_{i,\rho}\}, \tag{16}$$

where all questions in q_i have equivalent meanings. The Q can be denoted by Exp. (17) which can be used as a training set.

$$Q = \{q_{i,j} | q_{i,j} \in q_i\}(1 \le i \le \}, 1 \le j \le \rho). \tag{17}$$

4.1.3 Pitching detection for candidate videos

This step employs a specialized pitching detection model to identify potential highlights within the draft videos. This model is designed to recognize instances of pitching in baseball games, including pitch deliveries and pitching,

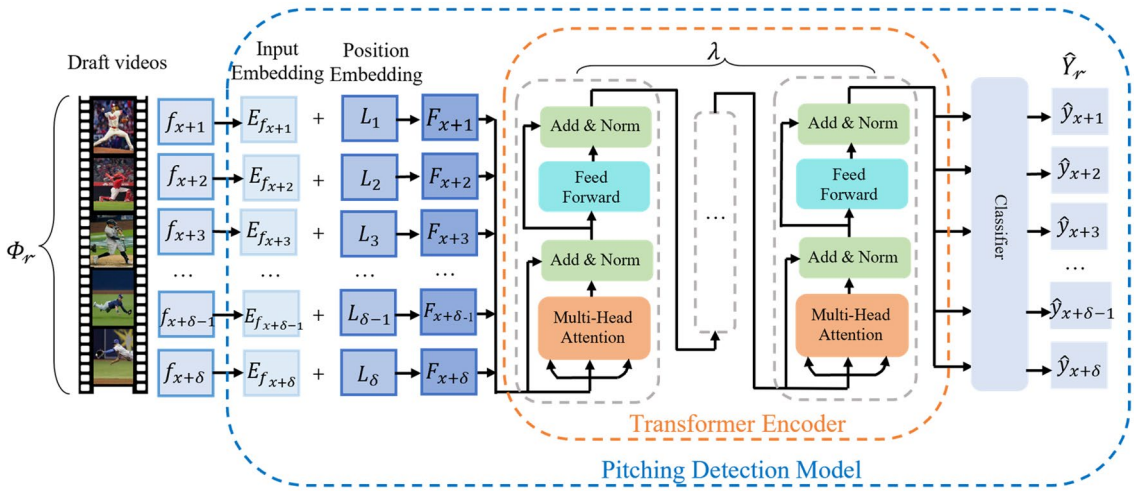


Fig. 3 The architecture of the pitching detection model

often associated with highlight-worthy moments. By analyzing each draft video, the model can precisely identify segments that are likely to feature pitching actions and label the initiation time of the pitching. These identified segments are subsequently extracted and compiled into candidate videos, representing potential highlight-worthy moments within the baseball video.

As shown in Fig. 3, each draft video Φ_s is presented as a sequence of δ frames, denoted by $(f_{x+1}, \ldots, f_{x+\delta})$. That is,

$$\Phi_s = \left(f_{x+1}, \ldots, f_{x+\delta}\right). \tag{18}$$

The Input Embedding and Position Embedding are applied to the inputΦ_s. Let F_{x+i} denote the embedding vector of the frame f_{x+i} $(1 \le k \le \alpha, 1 \le i \le \delta)$. Furthermore, let $F_s = (F_{x+1}, F_{x+2}, \ldots, F_{x+\delta})$ denote the input matrix of Φ_s. This matrix F_s is then fed into the Transformer Encoder model, which is composed of several identical blocks. Let λ denote the number of identical blocks. Each blocks contains a multi-head attention module followed by a feed forward module. A dot-product attention mechanism is applied on the d_{model} dimensional queries, keys, and values for each head. Let $\hbar$ denote the number of heads. The queries, keys, and values are grouped into matrices $Q_{s,j}$, $K_{s,j}$, and $V_{s,j}$ for $head_{s,j}$ $(1 \le j \le \hbar)$, respectively. The dimensions of matrices $Q_{s,j}$,$K_{s,j}$, and $V_{s,j}$ are d_k, d_k and d_v, respectively. The weights matrices of the queries, keys, and values of $head_{s,j}$ are denoted as $W^Q_{s,j} \in \mathbb{R}^{d_k \times d_{model}}$,$W^K_{s,j} \in \mathbb{R}^{d_k \times d_{model}}$, and $W^V_{s,j} \in \mathbb{R}^{d_v \times d_{model}}$, respectively. The value of $Q_{k,j}$,$K_{k,j}$, and $V_{k,j}$ can be calculated by Exps. (19)~(21), respectively.

$$Q_{s,j} = F_s \times W^Q_{s,j}, \tag{19}$$

$$K_{s,j} = F_s \times W^K_{s,j}, \tag{20}$$

$$V_{s,j} = F_s \times W^V_{s,j}. \tag{21}$$

Let $\alpha_{s,j}$ denote the output of the self-attention in $head_{s,j}$. The value of $\alpha_{s,j}$ can be derived from Exp. (22).

$$\alpha_{s,j} = Softmax\left(\frac{Q_{s,j}K_{s,j}^{\ T}}{\sqrt{d_k}}\right) * V_{s,j}. \tag{22}$$

Furthermore, there are $\hbar$ self-attention functions $head_{s,j}$ for $1 \le j \le \hbar$. These functions should be concatenated, aiming to jointly consider input information from various representation subspaces at distinct positions. Let H^M denote the output matrix of the multi-head self-attention at first identical block. That is

$$H^M = Concat\left(\alpha_{s,1}, \alpha_{s,2}, \ldots, \alpha_{s,\hbar}\right) * W^H_s, \tag{23}$$

where $W^H_s \in \mathbb{R}^{\hbar d_v \times d_{model}}$ denotes the weighted matrix of the output H^M. In addition to the multi-head self-attention, a fully connected feed-forward network (FFN) is applied to the output of the multi-head self-attention in each identical block. The output of FFN then adds the matrix H^M to generate the final output of the identical block, which is denoted by H. That is

$$H = \mathrm{FFN}\left(H^M\right) + H^M. \tag{24}$$

Finally, the output H is fed into the next identical block. These steps, from Exps. (19) to (24) are repeated until a total of λ identical blocks have been processed. The output of the Transformer Encoder model is then fed into a classifier. This classifier utilizes the output vector of each frame to predict whether the frame is related to pitching.

Till now, the design of the pitching detection model is presented. The next step is to present the training process. Before training, the video should be partitioned into several draft videos each with a fix δ frames. Each draft video will be treated as input data for the model. The frames that related to the pitching behavior should be labeled by 1 while the others should be labeled by 0. Consider the s-th input draft video $\Phi_s = \left(f_{x+1}, \ldots, f_{x+\delta}\right)$. Let $\widehat{Y}_s = (\widehat{y}_{x+1}, \ldots, \widehat{y}_{x+\delta})$ denote the prediction output of the classifier for input Φ_s. Each value $\widehat{y}_{x+i} \in \widehat{Y}_s$ is a probability value and satisfies the condition $0 \le p_{x+i} \le 1$. Let $Y_s = (y_{x+1}, \ldots, y_{x+\delta})$ be the label of input Φ_s. As a pitching behavior comprises multiple consecutive frames relevant to pitching, it is essential to examine a constraint ensuring that there are at least ω values of $y_{x+i} \in Y_s$ that equal to 1. A valid label should satisfy the following constraint. That is

$$\sum_{i=1}^{\delta} y_{x+i} \ge \omega. \tag{25}$$

Let L_p denote loss function of the pitching detection model for all $\Phi_s \in \Phi$. The binary cross entropy can be applied to calculate each specific loss for each predicted $\widehat{y}_{x+i}$, as compared with the corresponding label y_{x+i}. Exp. (26) presents the loss function of the pitching detection model.

$$L_p\left(y_{x+i}, \widehat{y}_{x+i}\right) = -\frac{1}{\delta}\sum_{i=1}^{\delta}(y_{x+i} log\widehat{y}_{x+i} + (1 - y_{x+i})(1 - log\widehat{y}_{x+i})). \tag{26}$$

During the inference phase, let ∂ denote the threshold. If the predicted output $\widehat{y}_{x+i}$ is greater than ∂, it is considered as 1. That is

$$\hat{y}_{x+i} = \begin{cases} 1, \hat{y}_{x+i} > \partial, \\ 0, \hat{y}_{x+i} \le \partial. \end{cases} \tag{27}$$

Let $\hat{\varphi}_{\circledast}$ denote the prediction of the pitching detection model. The value of $\hat{\varphi}_{\circledast}$ can be computed by Exp. (28).

$$\hat{\varphi}_{\circledast} = \begin{cases} 1, \sum_{i=1}^{\delta} \hat{y}_{x+i} \ge \omega, \\ 0, \textit{otherwise}. \end{cases} \tag{28}$$

If $\hat{\varphi}_{\circledast}$ is equal to 1, it is indicated that the input $\Phi_{\circledast}$ contains a pitching behavior. To obtain the candidate video, let $p_{x+\vartheta} \in P_{\circledast}(1 \le \vartheta \le \delta)$ represent the frame where the first occurrence of 1 is detected. This frame will be considered as the starting frame of the pitching behavior and is denoted as f_s. That is

$$f_s = f_{x+\vartheta} = f_{(\circledast-1)*\delta+\vartheta}. \tag{29}$$

Let $\mathbb{C}_{temp}$ denote the candidate video. It starts at frame f_s and ends with the frame δ frames after f_s in Φ. It can be derived from Exp. (30).

$$\mathbb{C}_{temp} = \left(f_s, \ldots, f_{s+\delta}\right). \tag{30}$$

Let $\mathbb{C}$ denote a set of candidate videos. The initial value of $\mathbb{C}$ is $\varnothing$. The value of $\mathbb{C}$ can be derived from Exp. (31).

$$\mathbb{C} = \mathbb{C} \cup \left\{\mathbb{C}_{temp}\right\}. \tag{31}$$

Till now, all candidate videos that contain the pitching behavior have been collected. The next phase aims to further identify the candidate videos by applying the Fine-Grained approach.

4.2 Fine-grained identification for highlights

This phase aims to utilize a QA model to identify the highlights from the candidate set $\mathbb{C}$. The input consists of two parts: the first one is the text, representing the query to find a specific highlight in a candidate video. For example, "Please help me find the catching highlight." The second part is a candidate video. This QA model will have the capability to automatically identify the start and end positions of the highlight. The QA model comprises three modules: Text Encoder, Video Encoder and VideoQ. The Text Encoder is responsible for establishing connections between words of each question using BERT. The Video Encoder's role is to capture the relationships between frames of each candidate video. The VideoQ, which is implemented by Transformer Encoder, is used to predict whether the candidate video contains the highlights mentioned in the question and to identify the start and end positions of those highlights by combining text and video features.

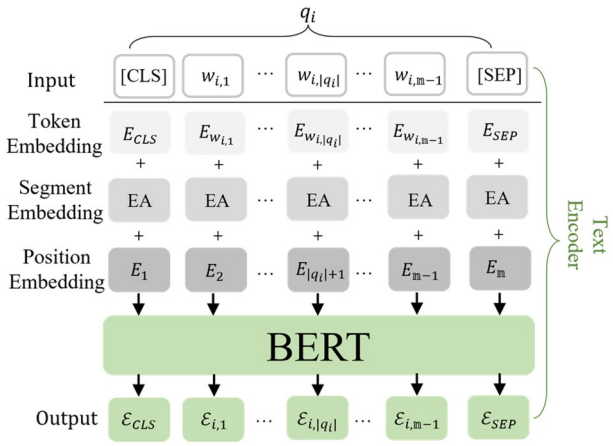


Fig. 4 The process of the Text Encoder model

4.2.1 Text encoder module

This model aims to analyze the relationships between words in each question of users based on pre-trained language model BERT. As shown in Fig. 4, let $q_i \in Q$ denote the i-th input query. It is divided into $|q_i|$ words using Chinese Knowledge and Information Processing Toolkit (CKIP) [46]. That is

$$q_i = \left(w_{i,1}, \ldots, w_{i,|q_i|}\right), \tag{32}$$

where $w_{i,j}(1 \le j \le |q_i|)$ corresponds to the j^{th} word after splitting the question q_i using CKIP. The token, segment, and position embeddings are applied to the input q_i. The [CLS] and [SEP] denote the first and delimiter tokens of each question in BERT. Let $d_{\mathscr{E}}$ and $\mathbb{m}$ denote the dimension and input length of the BERT, respectively. Recall that $\}$ and ρ denote the numbers of collected questions and new questions generated by LLM for each collected question, respectively. The value of $\mathbb{m}$ can be derived from Exp. (33)

$$\mathbb{m} = 2 + \max_{\forall i} |q_i| (1 \le i \le \rho * g). \tag{33}$$

Let $\mathcal{E}_i$ denote the text encoder of the input question q_i after processing by the BERT model. It denotes the relationships between individual words within question q_i. That is

$$\mathcal{E}_i = BERT\left(q_i\right) = \left(\mathcal{E}_{CLS}, \mathcal{E}_{i,1}, \ldots, \mathcal{E}_{i,\mathbb{m}-1}, \mathcal{E}_{SEP}\right) \in \mathbb{R}^{\mathbb{m} \times d_{\mathscr{E}}}, \tag{34}$$

where $\mathcal{E}_{i,j} \in \mathbb{R}^{d_{\mathscr{E}}}$ represents the hidden states of the BERT.

4.2.2 Video encoder module

This module aims to capture the relationships between frames of the candidate video using the Transformer encoder. Assume that the candidate videos $\mathbb{C} = \{\mathbb{C}_1, \ldots, \mathbb{C}_\gamma\}$ consist of γ videos. Each candidate video $\mathbb{C}_j \in \mathbb{C}$ is segmented into δ frames. Each frame is embedded as d_ℓ dimensional vector and is then processed through the Transformer encoder, which is like the pitching detection process described in Exps. (18) ~ (24). Let $\mathcal{F}_j = (\mathcal{F}_{j+1}, \ldots, \mathcal{F}_{j+\delta}) \in \mathbb{R}^{\delta \times d_\ell}$ denote the output of Transformer encoder which can capture the frame relationships in $\mathbb{C}_j$.

4.2.3 VideoQ module

This module aims to predict whether the candidate video contains the highlights mentioned in the question and pinpoint the start and end points of the highlights using text video attention. As shown in Fig. 5, the VideoQ model takes the two types of encodings, $\mathcal{E}_i$ and $\mathcal{F}_j$, which are generated by the Text Encoder and Video Encoder, respectively, as the inputs.

The VideoQ model employs the Text Video Attention (TVA), a multi-head self-attention, to facilitate the integration of information between $\mathcal{E}_i$ and $\mathcal{F}_j$. The objective is to incorporate question-related information into the video.

The VideoQ model consists of ξ layers of TVA, followed by a global average pooling (GAP) operation and Multilayer Perceptron (MLP) networks for the predictions of start and end locations of the highlight. In the TVA, $\mathcal{E}_i$ and $\mathcal{F}_j$ are two inputs of layer normalization (LN), aiming to keep their values within a consistent range. This helps prevent significant differences between vectors that might lead to bias in calculating self-attention scores in the model. As shown in the right side of Fig. 5, the TVA consists of LA, multi-head transformer attention, LN as well as MLP layers. Let $\mathcal{E}'_i$ and $\mathcal{F}'_j$ denote the output of LN. They are then fed into a multi-head transformer attention. The TVA applies a dot-product attention on queries, keys, and values with the d'_{model} dimensional for each head. Let $\mathcal{L}$ denote the number of heads. Each $head_{i,j,\ell}$ denotes the ℓ^{th} head integrating information from question $\mathcal{E}_i$ into candidate video $\mathcal{F}_j$.

Let $Q_{i,j,\ell}$ be utilized to capture attention from video $\mathcal{F}'_j$ to question $\mathcal{E}'_i$, denoting the queries matrix extracted from the $\mathcal{F}'_j$ for $head_{i,j,\ell}$ $(1 \le \ell \le \mathcal{L})$. Similarly, $K_{i,j,\ell}$ is responsible for extracting attention from question $\mathcal{E}'_i$ to video $\mathcal{F}'_j$, representing the keys matrix from $\mathcal{E}'_i$ for $head_{i,j,\ell}$.

Additionally, $V_{i,j,\ell}$ incorporates the information of question $\mathcal{E}'_i$ into the video $\mathcal{F}'_j$, denoting the values matrix from $\mathcal{E}'_i$ for $head_{i,j,\ell}$. The dimensions of matrices $Q_{i,j,\ell}$,

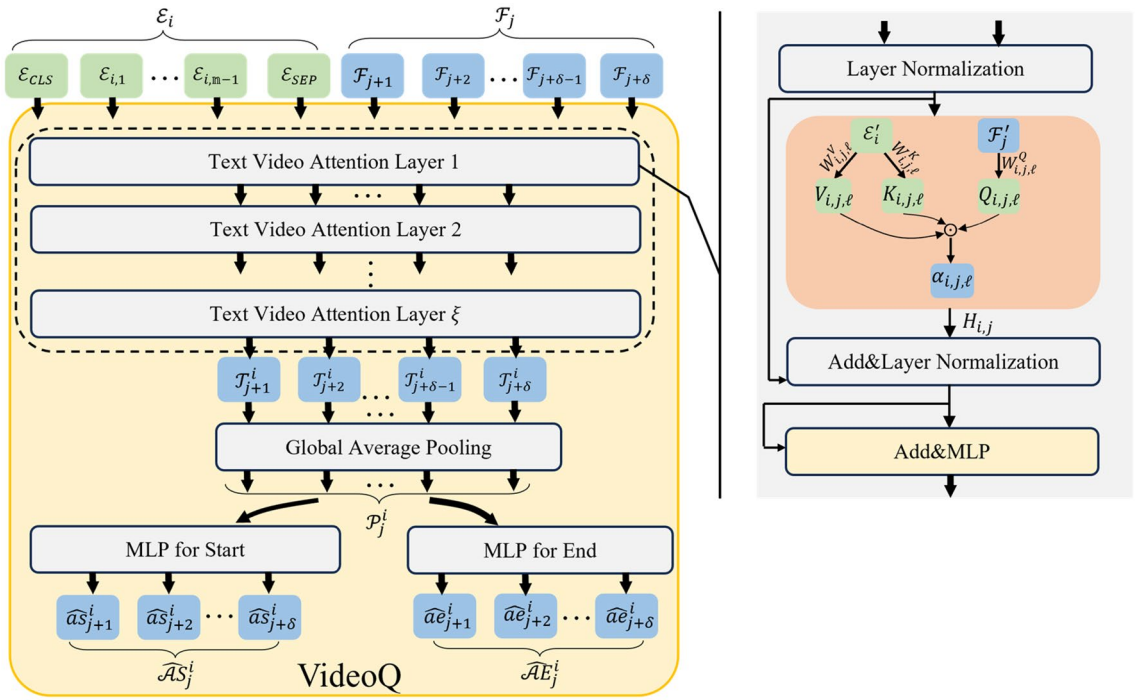


Fig. 5 The process of the VideoQ model

$K_{i,j,\ell}$, and $V_{i,j,\ell}$ are d'_k, d'_k and d'_v, respectively. The weights matrices of the queries, keys, and values of $head_{i,j,\ell}$ are denoted as $W^Q_{i,j,\ell} \in \mathbb{R}^{d'_k \times d'_{model}}$, $W^K_{i,j,\ell} \in \mathbb{R}^{d'_k \times d'_{model}}$, and $W^V_{i,j,\ell} \in \mathbb{R}^{d'_v \times d'_{model}}$, respectively. The value of $Q_{i,j,\ell}$, $K_{i,j,\ell}$, and $V_{i,j,\ell}$ can be calculated by Exps. (35) ~ (37), respectively.

$$Q_{i,j,\ell} = \mathcal{F}'_j \times W^Q_{i,j,\ell}, \tag{35}$$

$$K_{i,j,\ell} = \mathcal{E}'_i \times W^K_{i,j,\ell}, \tag{36}$$

$$V_{i,j,\ell} = \mathcal{E}'_i \times W^V_{i,j,\ell}. \tag{37}$$

Let $\alpha_{i,j,\ell}$ denote the output of the self-attention in $head_{i,j,\ell}$. The value of $\alpha_{i,j,\ell}$ can be derived from Exp. (38).

$$\alpha_{i,j,\ell} = softmax\left(\frac{Q_{i,j,\ell} {K_{i,j,\ell}}^T}{\sqrt{d'_k}}\right) * V_{i,j,\ell}. \tag{38}$$

Furthermore, assume that there are $\mathcal{L}$ self-attention functions $head_{i,j,\ell}$ for $1 \le \ell \le \mathcal{L}$. These functions should be concatenated, aiming to jointly consider input information from various representation subspaces at distinct positions. Let $H_{i,j}$ denote the output matrix of the multi-head self-attention. That is

$$H_{i,j} = concat\left(\alpha_{i,j,1}, \alpha_{i,j,2}, \ldots, \alpha_{i,j,\mathcal{L}}\right) * W^H_{i,j}, \tag{39}$$

where $W^H_{i,j} \in \mathbb{R}^{\mathcal{L}d'_v \times d'_{model}}$ denotes the weighted matrix of the output $H_{i,j}$. In addition to the multi-head self-attention, the $H_{i,j}$ will be further processed by LN and a MLP network. The MLP applies a nonlinear transformation to each vector. In the multi-head self-attention, only question features are fused into the candidate video, but the temporal relationships between candidate video frames are not extracted. MLP is responsible for capturing the temporal relationships between video vectors and incorporating them into each vector.

Let $\mathcal{T}^i_j = (\mathcal{T}^i_{j+1}, \ldots, \mathcal{T}^i_{j+\delta})$ denote the output resulting from ξ layers of TVA. It corresponds to the video $\mathcal{F}_j$ encoding after incorporating question $\mathcal{E}_i$ and includes the understanding of question for each frame as well as the relationships between frames. Subsequently, the video encoding $\mathcal{T}^i_j$ undergoes GAP operation, reducing its dimension from $\mathbb{R}^{\delta \times d'_{model}}$ to $\mathbb{R}^{\delta}$, denoted by $\mathcal{P}^i_j \in \mathbb{R}^{\delta}$. Finally, the $\mathcal{P}^i_j$ is input of the Multi-Class Single-Label classification networks. As shown in Fig. 5, the two MLP networks are used to predict the start and end point of the highlight, represented as $\widehat{\mathcal{AS}}^i_j = (\hat{as}^i_{j+1}, \ldots, \hat{as}^i_{j+\delta})$ and $\widehat{\mathcal{AE}}^i_j = (\hat{ae}^i_{j+1}, \ldots, \hat{ae}^i_{j+\delta})$, respectively. The values of $\hat{as}^i_{j+x} \in \widehat{\mathcal{AS}}^i_j$ and $\hat{ae}^i_{j+x} \in \widehat{\mathcal{AE}}^i_j$ are probability values which are between 0 and 1.

Let $\mathcal{AS}^i_j = (as^i_{j+1}, \ldots, as^i_{j+\delta})$ and $\mathcal{AE}^i_j = (ae^i_{j+1}, \ldots, ae^i_{j+\delta})$ be the labels of start and end points of highlight in video $\mathcal{F}_j$ associated with the question $\mathcal{E}_i$. The values of $as^i_{j+x} \in \mathcal{AS}^i_j$ and $ae^i_{j+x} \in \mathcal{AE}^i_j$ are either 0 or 1. As the start and end points of the highlight may comprise multiple frames, it is essential to examine a constraint ensuring that there are at least π values that equal to 1. A valid label should satisfy the following constraint. That is

$$\begin{aligned} &\sum_{x=1}^{\delta} as^i_{j+x} \ge \pi, \\ &\sum_{x=1}^{\delta} ae^i_{j+x} \ge \pi. \end{aligned} \tag{40}$$

Let L_{φ} denote the loss function of the VideoQ for all $\mathbb{C}_j \in \mathbb{C}$ $(1 \le j \le \gamma)$. It is calculated using categorical cross-entropy to combine the losses of both classifiers and refine the VideoQ model, for the predicted $\widehat{\mathcal{AS}}^i_j$ and $\widehat{\mathcal{AE}}^i_j$, as compared with the corresponding labels $\mathcal{AS}^i_j$ and $\mathcal{AE}^i_j$. Exp. (41) presents the loss function of the VideoQ model.

$$\begin{aligned} &L_{\varphi}\left(\mathcal{AS}^i_j, \mathcal{AE}^i_j, \widehat{\mathcal{AS}}^i_j, \widehat{\mathcal{AE}}^i_j\right) \\ &= -\frac{1}{\delta}\sum_{x=1}^{\delta} as^i_{j+x} log\hat{as}^i_{j+x} - \frac{1}{\delta}\sum_{x=1}^{\delta} ae^i_{j+x} log\hat{ae}^i_{j+x}. \end{aligned} \tag{41}$$

During the inference phase, let ρ^S and ρ^E denote the thresholds of start and end points of highlight, respectively. If the predicted outputs $\hat{as}^i_{j+x}$ and $\hat{ae}^i_{j+x}$ are greater than ρ^S and ρ^E, respectively, they are considered as 1. That is

$$\begin{aligned} \hat{as}^i_{j+x} &= \begin{cases} 1, & \hat{as}^i_{j+x} \ge \rho^S, \\ 0, & otherwise, \end{cases} \\ \hat{ae}^i_{j+x} &= \begin{cases} 1, & \hat{ae}^i_{j+x} \ge \rho^E, \\ 0, & otherwise. \end{cases} \end{aligned} \tag{42}$$

The input candidate video $\mathbb{C}_j$ is considered as containing a highlight $\hat{H}_{temp}$, the output $\widehat{\mathcal{AS}}^i_j$ and $\widehat{\mathcal{AE}}^i_j$ should be satisfied the Exp. (43).

$$\left(\sum_{x=1}^{\delta} \hat{as}^i_{j+x} \ge \pi\right) \& \left(\sum_{x=1}^{\delta} \hat{ae}^i_{j+x} \ge \pi\right) = 1. \tag{43}$$

In case that Exp. (43) holds, it indicates that the input $\mathbb{C}_j$ contains H_{temp} mentioned in question q_i. Let

$\mathcal{F}_{j+\not}\in\mathcal{F}_j(1\le\not\le\delta)$ be the starting frames that satisfy the following conditions.

$$\sum_{k=1}^{\not-1}\hat{as}^i_{j+k}=0 \text{ and } \hat{as}^i_{j+\not}=1. \tag{44}$$

Let $\mathcal{F}_{j+\not}\in\mathcal{F}_j(1\le\not\le\delta)$ represent the last end frame that satisfy the following conditions.

$$\hat{ae}^i_{j+\not}=1 \text{ and } \sum_{k=\not+1}^{\delta}\hat{ae}^i_{j+k}=0. \tag{45}$$

The $\hat{H}_{temp}$ starts at frame $\mathcal{F}_{j+\not}$ and ends with the frame $\mathcal{F}_{j+\not}$. It can be derived from Exp. (46).

$$\hat{H}_{temp}=\left(\mathcal{F}_{j+\not},\ldots,\mathcal{F}_{j+\not}\right), \tag{46}$$

where the $\mathcal{F}_{j+\not}$ and $\mathcal{F}_{j+\not}$ can be obtained by the Exp. (47).

$$\begin{aligned}\mathcal{F}_{j+\not}&=f_{j+\not}=f_{(j-1)*\delta+\not},\\ \mathcal{F}_{j+\not}&=f_{j+\not}=f_{(j-1)*\delta+\not}.\end{aligned} \tag{47}$$

Recall that $\hat{H}$ denotes a predicted set of highlights. The initial value of $\hat{H}$ is $\varnothing$. The set $\hat{H}$ can be derived from Exp. (48).

$$\hat{H}=\hat{H}\cup\left\{\hat{H}_{temp}\right\}. \tag{48}$$

5 Performance study

This section aims to evaluate the performance improvement of the proposed HMTV against the existing related studies, including ViT [37], PGL-SUM [29] and CLIP-It [30]. The ViT model utilized Transformer encoder for processing video independently. The PGL-SUM incorporated global and local multi-head attention mechanisms, capturing frame dependencies at various granularities to generate a video summary. However, both the ViT and PGL-SUM focus on visual information, neglecting the other modalities, such as text and audio. The CLIP-It generated video summaries based on user-defined natural language queries or system-generated descriptions. It employed a Language-Guided Attention head to calculate a unified representation of frame and language embedding. Additionally, a Frame-Scoring Transformer assigns scores to each frame using the fused representations. However, the CLIP-It failed to consider the importance of temporal information between frames in the videos. In contrast, the proposed HMTV employs TVA to integrate the features of the question and video, and MLP to capture the temporal relationships between frames. This mechanism aims to identify the specific highlight mentioned in the question from the video. The following presents the parameters setting, datasets and simulation results.

Table 1 Parameters setting

Parameters	Value
Video input size	(300, 160, 90, 3) ~ (300, 320, 180, 3)
Patch size	(1, 160, 90) ~ (1, 320, 180)
Embedding dimension $d_{\not}$	512 ~ 1024
Layers	2 ~ 10
Heads	4 ~ 16
Max length of the BERT	32
LN epsilon	1e − 6
Learning rate	1e − 4 ~ 3e − 4

5.1 Parameters setting and datasets

Table 1 lists the parameters setting in the experiments. The video size is set ranging from (300, 160, 90, 3) to (300, 320, 180, 3) while the patch size is varied ranging from (1, 160, 90) to (1, 320, 180). The embedding dimension $d_{\not}$ is varied ranging from 512 to 1024. The layers and heads of Transformer are varied from 2 to 10, and 4 to 16, respectively. The maximal length of the BERT is set to 32. The LN epsilon is set to 1e-6 and the learning rate is varied from 1e − 4 to 3e − 4.

To evaluate the performance of the proposed HMTV, the experiments are conducted based on several datasets, including self-collected, Sum Me, TV Sum, MLB-YouTube (MLB), NTU-RGB + D 120 (NTU) and Toyota Smart Home (TSH). The self-collected dataset, collected from YouTube and ELTA, comprises three categories of baseball

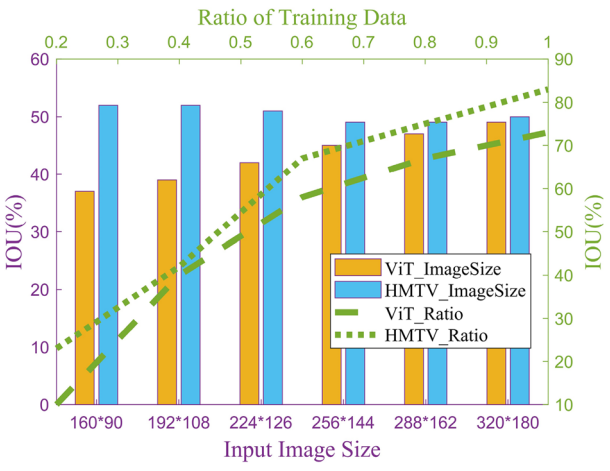


Fig. 6 The IOU comparison of the proposed HMTV with the ViT by varying ratio of training data and input image size

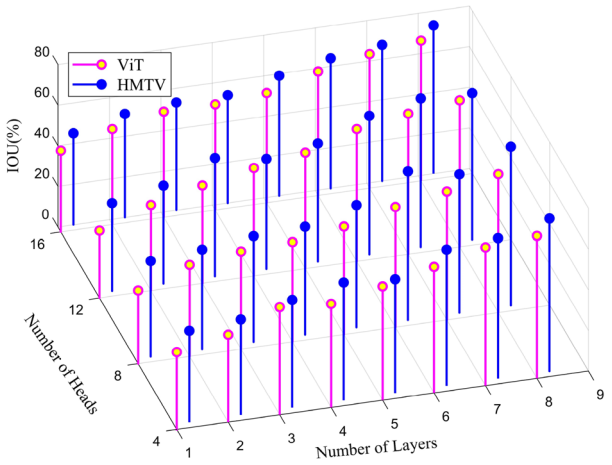


Fig. 7 The IOU comparison of the proposed HMTV model with the ViT model by varying the numbers of heads and layers

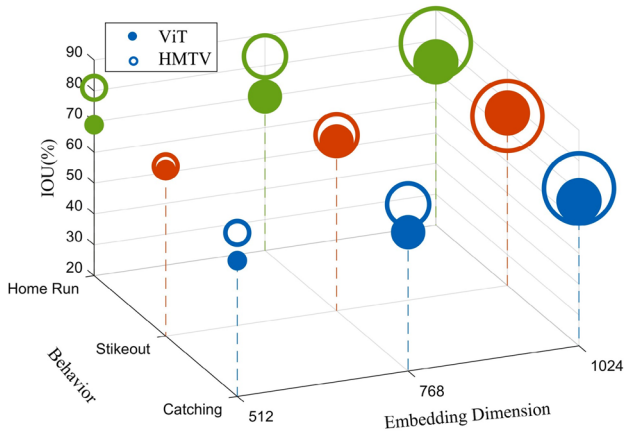


Fig. 8 The IOU comparison of the proposed HMTV with the ViT model by varying behavior and embedding dimension

highlights: 41 catching videos, 72 strikeout videos, and 76 home run videos. The experiments generate 30 high-quality questions, and subsequently replaced the keywords with terms corresponding to various key highlights. These questions serve as textual inputs for training.

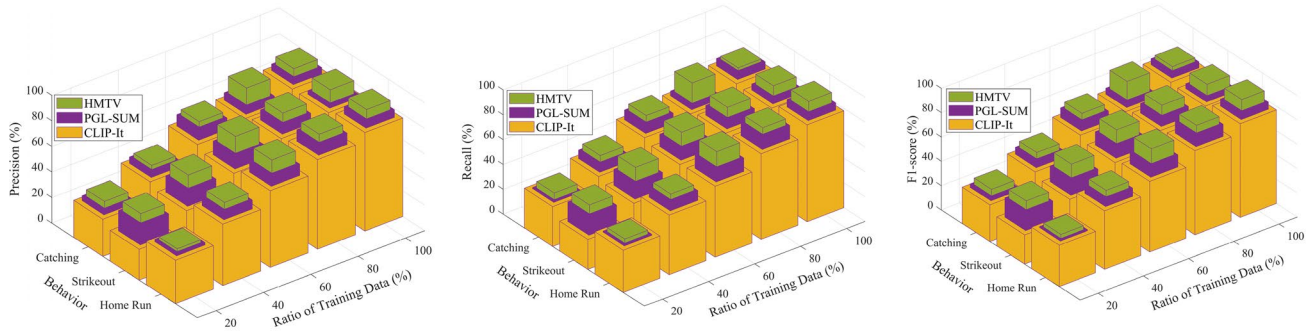
5.2 Simulation results

Figure 6 compares the Intersection over Union (IOU) of the proposed HMTV and the ViT model across varying input image sizes and training data ratios. The input image size varies from 160*90 to 320*180, and the training data ratio varies from 0.2 to 1. As shown in Fig. 5, the IOU of all the compared mechanisms increases with the ratio of training data. The reason is that a larger ratio of training data provides a more comprehensive set of samples for adjusting the model parameters to improve the recognition. As a result, the proposed HMTV outperforms ViT model, leveraging TVA to focus on questions and videos.

As shown in Fig. 6, the IOU of the ViT model increases with the input image size. This occurs because that the larger image size provides more details and information, facilitating more accurate recognition and classification by the model. It is observed that the IOU of the HMTV is more evenly distributed. This occurs because the use of TVA, which integrates question information into video vectors, adopting Transformer to analyze the relationship between questions and videos during training. This effective fusion of features reduces the model's dependence on input image size.

Figure 7 further compares the IOU of the proposed HMTV and the ViT, by varying the numbers of heads and layers in the Transformer. The pairing of layers and heads is curial. The numbers of layers and headers are varied from 1 to 8, and 4 to 16, respectively. As shown in Fig. 7, the IOU generally increases with the number of heads and layers in most cases. The proposed HMTV outperforms ViT when adopting the same heads and layers. The reason is that it adopts Coarse-Grained clipping for removing redundant videos and Fine-Grained identification of highlights by incorporating questions to enhance the recognition performance. For the ViT model, the IOU achieves the maximal when the number of heads and layers are set to 12 and 8, respectively. The proposed HMTV achieves its peak IOU when the number of heads and layers are set to 8 and 8, respectively. It achieves high performance with fewer number of heads, effectively minimizing computation time.

Figure 8 compares the IOU and training time of the proposed HMTV and ViT models by varying behaviors and embedding dimension. The size of the circle denotes the training duration. The behaviors include catching, strikeout, and home run. The embedding dimensions are set to 512, 768, or 1024. As shown in Fig. 8, The IOU of ViT model increases with embedding dimension for all behaviors. The reason is that the model is capable of learning more complex and detailed features with the increasing of embedding dimension. This can effectively enhance behavior recognition performance. The IOU of the proposed HMTV increases when the embedding dimension is smaller than 768 and decreases when the embedding dimension is longer than 768. It outperforms the ViT because of the additional input of questions. However, the training duration will be increased as higher-dimensional features require more iterations for the model to converge. The proposed HMTV has more training duration than the ViT under the same embedding dimension. The reason is that the additional input



(a) Precision (b) Recall (c) F1-score

Fig. 9 Comparison of HMTV with the related works PGL-SUM and CLIP-It by varying behavior and ratio of training data

Table 2 The ablation study of the proposed HMTV on the self-collected dataset

Modules	F1-Score
Transformer encoder	0.431
+ Pitching detection model	0586
+ BERT	0.696
+ VideoQ	0.753

Table 3 Comparing F1-score of different mechanisms on TVSum and SumMe

Mechanism	SumMe	TVSum
Dsnet [31]	0.502	0.621
CLIP-It [30]	0.525	0.63
PGL-SUM [29]	0.556	0.61
ViT [37]	0.493	0.562
STVT [39]	0.518	0.648
HMTV	**0.559**	**0.648**

Table 4 Comparing F1-score of different mechanisms on MLB, NTU and TSH

Mechanism	MLB	NTU	TSH
IDT [32]	0.273	–	0.237
I3D [33]	0.301	–	0.451
STA [34]	–	0.946	0.503
NPL [35]	0.427	0.937	0.546
ViT [37]	0.453	0.873	0.528
DRDN [44]	–	0.929	–
HMTV	**0.519**	**0.951**	**0.573**

question results in an increase in the number of parameters, consequently leading to an extended training duration.

Figure 9a–c compare the proposed HMTV with the related studies PGL-SUM and CLIP-It in terms of precision, recall, and F1-score by varying behavior and ratio of training data, respectively. The ratio of training data is varying from 0.2 to 1 and the behaviors include catching, strikeout, and home run. As shown in Fig. 9, all mechanisms have the common trend that the precision, recall as well as F1-score are increasing with the training data ratio for all behaviors. The reason is that a higher ratio of training data provides more training samples to fine-tune the model parameters and enhance recognition performance. The proposed HMTV outperforms the PGL-SUM and CLIP-It mechanisms in all cases. This occurs because that the proposed HMTV integrates both video and question information, capturing features in a more comprehensive manner.

As shown in Table 2, an ablation study is conducted on self-collected dataset to evaluate each module's effectiveness in the proposed HMTV, which includes the Transformer encoder, pitching detection model, BERT, and VideoQ, using F1-Score as the metric. The baseline, only utilizing a Transformer encoder for video features extraction, achieves an F1-score of 0.431. Integrating of the pitching detection model, the performance rises to 0.586, indicating that this model can efficiently discard redundant information and focus on key content. By further employing the BERT model to analyze user queries, the F1-score reaches 0.696, demonstrating the benefit of fusing textual features. Finally, combining textual and visual features into the VideoQ model, the performance peaks at 0.753. These findings underscore the importance of textual and visual fusion for enhancing the model's efficiency and understanding in this task.

The above experiments are conducted on self-collected dataset from YouTube and ELTA. Tables 3 compares the proposed HMTV against existing studies, including Dsnet [31], CLIP-It [30], PGL-SUM [29], ViT [37], and STVT [39] using SumMe and TVSum datasets. Table 4 further compares the F1-score of the proposed HMTV against

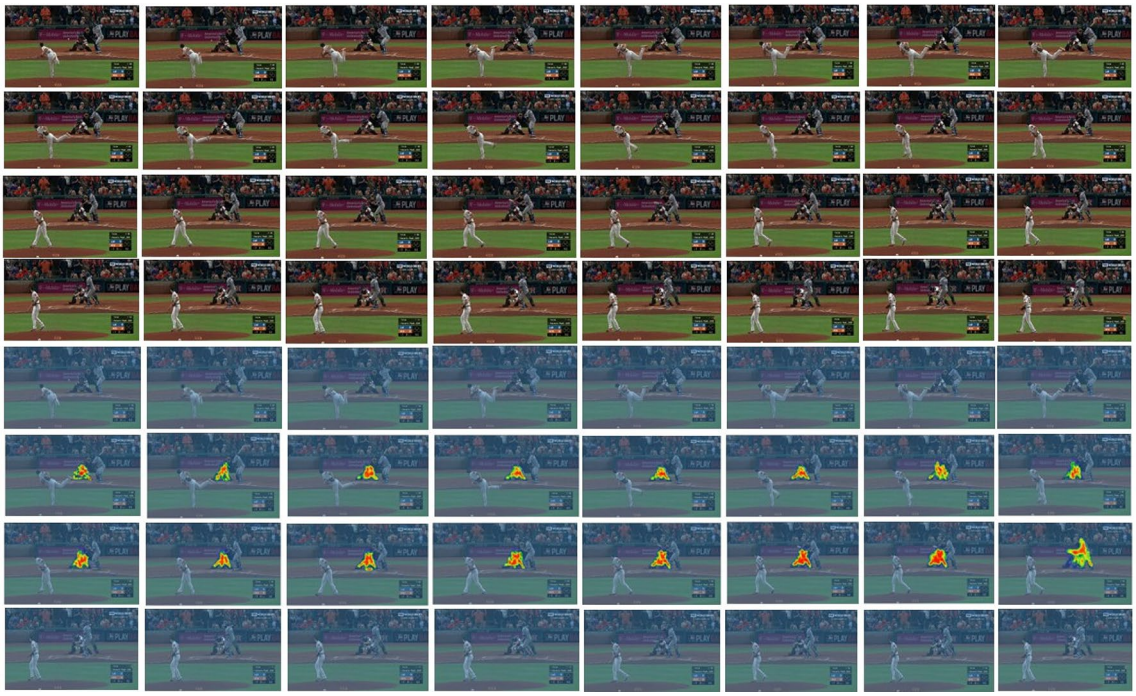


Fig. 10 The visualization of the prediction results for a baseball segment based on CAM heatmaps, highlighting key temporal frames of catching behavior

existing studies, including IDT [32], I3D [33], STA [34], NPL [35], ViT [37] and DRDN [44] based on datasets, including MLB, NTU and TSH. In comparison, the proposed HMTV outperforms all the other mechanisms. This occurs because that the proposed HMTV utilizes a Transformer encoder to capture the temporal relationships between frames and employs BERT to analyze the semantic relationships between words in each user's question. The extracted visual and textual features are then jointly fed into the VideoQ model, which automatically identifies the precise start and end positions of the requested highlights within the video. This allows the model to capture more comprehensive features and better understand the content by integrating both visual and semantic information.

Figure 10 presents a visualization of the prediction results for a baseball segment based on CAM heatmaps [45], facilitated by the proposed HMTV. The upper part of the image displays the original video frames from the baseball segment, while the lower part shows the heatmap generated by the proposed HMTV using CAM. This illustration clearly demonstrates how the proposed HMTV effectively captures the key temporal features of the catching highlight, focusing on the athlete's movements for precise and timely detection. As the athlete moves to catch the ball, the highlighted areas in the heatmap move correspondingly, clearly outlining the athlete's trajectory. The heatmaps at these time steps accurately focus on the changes in the athlete's movements,

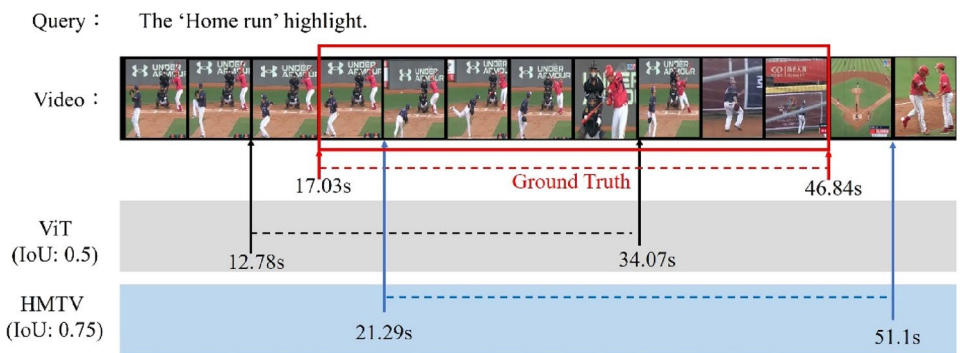


Fig. 11 The 'Home run' highlight in the video (Duration: 55.36 s)

showcasing the proposed HMTV special ability to capture temporal features.

Figure 11 illustrates the qualitative results of the proposed HMTV, compared to a baseline model, ViT, in retrieving the segment corresponding to the query "The 'Home run' highlight" from a video clip with a duration of 55.36 s. The ground truth segment, shown in red, spans from 17.03 to 46.84 s of the video. The ViT predicts the segment from 12.78 to 34.07 s, capturing a part of the relevant event but failing to include the complete highlight. The IoU score for ViT in this case is 0.5. On the other hand, the proposed HMTV demonstrates superior performance by predicting the segment from 21.29 to 51.1 s, which covers most of the ground truth segment. The IoU score for the proposed HMTV in this case is 0.75, indicating a more accurate retrieval of the target moment. The reason is that the proposed HMTV first utilizes the Pitching Detect model to accurately identify the start of the highlight. Then, it combines this with the VideoQA model, which integrates both video content and query information, enabling the model to capture the entire relevant segment more effectively.

6 Conclusions

This paper proposes a novel highlight identification mechanism, called HMTV, which can effectively extract the highlights mentioned in uses' questions from lengthy baseball videos. The proposed HMTV integrates a hierarchical multimodal Transformer, including a pitching detection model, text encoder, video encoder and VideoQ. Initially, the pitching detection model employs a Transformer encoder to capture pitching behavior. This can significantly reduce the complexity of computation and identify the starting positions of candidate segments. Then, the text and video encoders employ Transformer encoder and pre-trained BERT to extract features from user' questions and candidate videos, respectively. Finally, the VideoQ model employs TVA to identify the start and end positions of highlights corresponding to users' questions within the candidate videos. Simulation results demonstrate that the proposed HMTV outperforms existing models, highlighting its effectiveness in highlights identification tasks.

Future work will further explore the integration of additional multimodal information, such as audio and subtitles, to achieve a comprehensive understanding and extraction of video highlights. Through the fusion of various perceptual modalities, a more holistic comprehension of videos can be achieved, aiming to enhance the effectiveness of highlights identification.

Acknowledgements This work was supported in part by the Scientific Research Project of Chuzhou University under Grant No. 2022XJYB05, Higher Education Research Program Project under Grant No. 2022AH010067, Anhui Outstanding Youth Fund Project under Grant No. 2022AH030109, and Science and Technology Plan Project in Chuzhou Grant No. 2021ZD016.

Author contributions Qiaoyun Zhang: conceptualization, methodology, formal analysis, writing-original-draft. Chih-Yung Chang: conceptualization, methodology, formal analysis, writing-original-draft. Ming-Yang Su: writing-review-editing. Hsiang-Chuan Chang: writing-review-editing. Diptendu Sinha Roy: writing-review-editing, supervision.

Data availability The datasets generated during the current study are not publicly available but are available from the corresponding author.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Gao, J., Yang, X., et al.: Unsupervised video summarization via relation-aware assignment learning. IEEE Trans. Multimedia **23**, 3203–3214 (2020)
2. Jung, Y., Cho, D., et al.: Discriminative feature learning for unsupervised video summarization. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 8537–8544 (2019)
3. Rabbouch, H., Saâdaoui, F., Mraihi, R.: Unsupervised video summarization using cluster analysis for automatic vehicles counting and recognizing. Neurocomputing **260**, 157–173 (2017)
4. Zhao, Y., et al.: Unsupervised video summarization via clustering validity index. Multimed. Tools Appl. **79**, 33417–33430 (2020)
5. Zhou, K., Qiao, Y., and Xiang, T.: Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 7582–7589 (2018)
6. Mahasseni, B., Michael, L., and Sinisa, T.: Unsupervised video summarization with adversarial lstm networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 202–211 (2017)
7. Yuan, L., et al.: Unsupervised video summarization with cycle-consistent adversarial LSTM networks. IEEE Trans. Multimed. **22**(10), 2711–2722 (2020)
8. Li, X., Zhao, B., Lu, X.: A general framework for edited video and raw video summarization. IEEE Trans. Image Process. **26**(8), 3652–3664 (2017)
9. Sharghi, A., Borji, A., et al.: Improving sequential determinantal point processes for supervised video summarization. In: Proceedings of the European Conference on Computer Vision, pp. 517–533 (2018)
10. Wei, H., Ni, B., et al.: Video summarization via semantic attended networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 216–223 (2018)
11. Rochan, M., and Wang, Y.: Video summarization by learning from unpaired data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7902–7911 (2019)
12. Muhammad, K., Hussain, T., et al.: Cost-effective video summarization using deep CNN with hierarchical weighted fusion for IoT surveillance networks. IEEE Internet Things J. **7**(5), 4455–4463 (2020)

13. Ji, Z., Xiong, K., et al.: Video summarization with attention-based encoder–decoder networks. IEEE Trans. Circ. Syst. Video Technol. **30**(6), 1709–1717 (2020)
14. Zhu, W., Lu, J., et al.: Learning multiscale hierarchical attention for video summarization. Pattern Recogn. **122**, 108312 (2022)
15. Zhao, B., Li, H., et al.: Reconstructive sequence-graph network for video summarization. IEEE Trans. Pattern Anal. Mach. Intell. **44**(5), 2793–2801 (2022)
16. Ji, Z., Jiao, F., et al.: Deep attentive and semantic preserving video summarization. Neurocomputing **405**, 200–207 (2020)
17. Huang, S., Li, X., et al.: User-ranking video summarization with multi-stage spatio–temporal representation. IEEE Trans. Image Process. **28**(6), 2654–2664 (2019)
18. Yang, Z., Garcia, N., et al.: A comparative study of language transformers for video question answering. Neurocomputing **445**, 121–133 (2021)
19. Gu, M., Zhao, Z., et al.: Graph-based multi-interaction network for video question answering. IEEE Trans. Image Process. **30**, 2758–2770 (2021)
20. Yin, C., Tang, J., et al.: Memory augmented deep recurrent neural network for video question answering. IEEE Trans. Neural Netw. Learn. Syst. **31**(9), 3159–3167 (2020)
21. Chu, W., Xue, H., et al.: The forgettable-watcher model for video question answering. Neurocomputing **314**, 386–393 (2018)
22. Xue, H., Zhao, Z., Cai, D.: Unifying the video and question attentions for open-ended video question answering. IEEE Trans. Image Process. **26**(12), 5656–5666 (2017)
23. Zha, Z.J., Liu, J., et al.: Spatiotemporal-textual co-attention network for video question answering. ACM Trans. Multimed. Comput. Commun. Appl. **15**(2s), 1–18 (2019)
24. Zhao, B., Li, X., Lu, X.: Property-constrained dual learning for video summarization. IEEE Trans. Neural Netw. Learn. Syst. **31**(10), 3989–4000 (2020)
25. Jin, W., Zhao, Z., et al.: Adaptive spatio-temporal graph enhanced vision-language representation for video QA. IEEE Trans. Image Process. **30**, 5477–5489 (2021)
26. Liu, Y., Zhang, X., et al.: Cross-attentional spatio-temporal semantic graph networks for video question answering. IEEE Trans. Image Process. **31**, 1684–1696 (2022)
27. Yu, T., Yu, J., et al.: Compositional attention networks with two-stream fusion for video question answering. IEEE Trans. Image Process. **29**, 1204–1218 (2019)
28. Wang, W., Huang, Y., Wang, L.: Long video question answering: a matching-guided attention model. Pattern Recogn. **102**, 107248 (2020)
29. Apostolidis, E., Balaouras, G., et al.: Combining global and local attention with positional encoding for video summarization. In: IEEE International Symposium on Multimedia, pp. 226–234 (2021)
30. Narasimhan, M., Rohrbach, A., Darrell, T.: Clip-it! Language-guided video summarization. Adv. Neural. Inf. Process. Syst. **34**, 13988–14000 (2021)
31. Zhu, W., Lu, J., et al.: Dsnet: a flexible detect-to-summarize network for video summarization. IEEE Trans. Image Process. **30**, 948–962 (2020)
32. Wang, H., and Schmid, C.: Action recognition with improved trajectories. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3551–3558 (2013)
33. Carreira, J., and Zisserman, A.: Quo vadis, action recognition? A new model and the kinetics dataset. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6299–6308 (2017)
34. Das, S., Dai, R., et al.: Toyota smarthome: Real-world activities of daily living. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 833–842 (2019)
35. Piergiovanni, A. J., and Ryoo, M. S.: Recognizing actions in videos from unseen viewpoints. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4124–4132 (2021)
36. Liu, Z., et al.: Detecting content segments from online sports streaming events: challenges and solutions. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 6414–6423 (2024)
37. Dosovitskiy, A., et al.: An image is worth 16x16 words: transformers for image recognition at scale. In: International Conference on Learning Representations, pp. 1–21 (2021)
38. Ahn, D., Kim, S., et al.: Star-transformer: a spatio-temporal cross attention transformer for human action recognition. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 3330–3339 (2023)
39. Hsu, T.C., Liao, Y.S., Huang, C.R.: Video summarization with spatiotemporal vision transformer. IEEE Trans. Image Process. **32**, 3013–3026 (2023)
40. Zhao, C., Wang, C., et al.: ISTVT: interpretable spatial-temporal video transformer for deepfake detection. IEEE Trans. Inf. Forensics Secur. **18**, 1335–1348 (2023)
41. Liang, Z., Dong, W., Zhang, B.: A dual-branch hybrid network of CNN and transformer with adaptive keyframe scheduling for video semantic segmentation. Multimed. Syst. **30**, 67 (2024)
42. Pang, W., He, Q., Li, Y.: Predicting skeleton trajectories using a skeleton-transformer for video anomaly detection. Multimed. Syst. **28**(4), 1481–1494 (2022)
43. Cheng, H., et al.: Joint graph convolution networks and transformer for human pose estimation in sports technique analysis. J. King Saud Univ.-Comput. Inf. Sci. **35**(10), 101819 (2023)
44. Liu, W., Zhong, X., et al.: Dual-recommendation disentanglement network for view fuzz in action recognition. IEEE Trans. Image Process. **32**, 2719–2733 (2023)
45. Feng, Z., et al.: VS-CAM: vertex semantic class activation mapping to interpret vision graph neural network. Neurocomputing **533**, 104–115 (2023)
46. Ma, W.Y., Chen, K.: Design of CKIP Chinese word segmentation system. Chin. Oriental Languages Inf. Process. Soc. **14**(3), 235–249 (2005)
47. Shen, X., Yang, X., et al.: Semantics-enriched cross-modal alignment for complex-query video moment retrieval. In: Proceedings of the 31st ACM International Conference on Multimedia, pp. 4109–4118 (2023).
48. Yang, X., Wang, S., et al.: Video moment retrieval with cross-modal neural architecture search. IEEE Trans. Image Process. **31**, 1204–1216 (2022)
49. Li, J., Li, K., et al.: Dual-path temporal map optimization for make-up temporal video grounding. Multimed. Syst. **30**, 140 (2024)
50. Yang, X., Chang, T., et al.: Learning Hierarchical Visual Transformation for Domain Generalizable Visual Matching and Recognition. International Journal of Computer Vision, 1–27 (2024).
51. Han, N., Chen, J., et al.: BiC-Net: learning efficient spatio-temporal relation for text-video retrieval. ACM Trans. Multimed. Comput. Commun. Appl. **20**(3), 1–21 (2023)
52. Li, Y., Yang, X., et al.: Redundancy-aware transformer for video question answering. In: Proceedings of the 31st ACM International Conference on Multimedia, pp. 3172–3180 (2023).
53. Touvron, H., Lavril, T., et al.: LLaMA: open and efficient foundation language models. Preprint at arXiv https://doi.org/10.48550/arXiv.2302.13971 (2023).