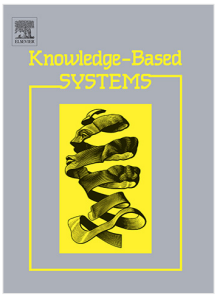
# Journal Pre-proof




COHETS: A highlight extraction method using textual streams of streaming videos

Chien Chin Chen, Liang-Wei Lo, Sheng-Jie Lin

PII: S0950-7051(22)01093-0
DOI: https://doi.org/10.1016/j.knosys.2022.110000
Reference: KNOSYS 110000

To appear in: *Knowledge-Based Systems*

Received date : 29 January 2022
Revised date : 7 October 2022
Accepted date : 10 October 2022

Please cite this article as: C.C. Chen, L.-W. Lo and S.-J. Lin, COHETS: A highlight extraction method using textual streams of streaming videos, *Knowledge-Based Systems* (2022), doi: https://doi.org/10.1016/j.knosys.2022.110000.

# COHETS: A Highlight Extraction Method using Textual Streams of Streaming Videos

*Chien Chin Chen**

*Liang-Wei Lo*

*Sheng-Jie Lin*

*{patonchen, r08725024, d11725003}@ntu.edu.tw*
*Department of Information Management*
*National Taiwan University*
*No.1, Sec. 4, Roosevelt Rd., Taipei City 106, Taiwan (R.O.C.)*

## Abstract

As more and more conversation-oriented streaming videos become available, streaming platforms have gradually taken the place of traditional media for people to access information. Still, conversation-oriented streaming videos are often lengthy and people are reluctant to view the whole video. In this research, we investigated highlight extraction on conversation-oriented streaming videos. Previous highlight extraction methods analyzed visual features of videos and are therefore unable to deal with conservation-oriented streaming videos whose highlights are related to streamer discourses and viewer responses. For this reason, the proposed highlight extraction method called COHETS does not evaluate visual features but rather simultaneously examines textual streams of streamer discourses and viewer messages to extract meaningful highlights. Experiments based on real world streaming data demonstrate that streamer discourses and viewer responses via their feedback messages are useful for extracting highlights of conversation-oriented streaming videos. Also, our designed position enrichment and message attention techniques effectively distill the embeddings of the two textual streams and lead to extraction results that are superior to those of state-of-the-art deep learning-based highlight extraction methods and extraction-based text summarization methods.

*Keywords:* supervised learning, natural language processing, video highlight extraction

## 1. Introduction

The streaming industry has in recent years skyrocketed due to the swift progress of Internet technologies. As streaming platforms like Twitch and YouTube become more popular, watching streaming videos is not just a trend but a daily activity for the younger generations. The COVID-19 pandemic further intensified the surge of online streaming. Lockdowns and social distancing measures transformed people's lifestyles and increased the demand for stay-at-home entertainment in general and streaming in particular. According to Grand View Research surveys[1], in March 2020 at the outset of the pandemic, the viewership of Twitch increased by 31%. The global market sizes of the streaming industry in 2019 and 2020 were $42.6 billion and $50.1 billion, respectively, and the market growth rate (compound annual growth rate (CAGR)) from 2021 to 2028 is estimated to be 21.0%. The promising market size and the huge viewer population have stimulated novel business models to benefit both streamers and platform providers. For platform providers, their revenues are normally based on paid advertising in that the more viewers a platform has, the more profit it gets; for streamers, profit mainly comes from viewer donations and product placement, which are also based on the viewer numbers. Since a lot of platforms and streamers engage in this competitive business, how to catch and keep the attention of the audience has become a practical concern in the streaming industry [1].

Live streaming consists of visual and audio content and emphasizes the interaction between streamers and viewers. To attract viewers, streamers design vivid and engaging content, which can be further archived as videos on the channel for public consumption, which can in turn help the channel gain more followers and subscriptions. However, live streaming recorded videos are often so lengthy that viewers are reluctant to watch them. In order to increase channel content and attract new subscribers and viewers, many streamers have started to provide streaming highlights by using video editing tools or cooperating with third-party studios. Twitch officials, for instance, provide streamers streaming markers for highlighting content. However, even with the help of tools, highlighting and editing are still time-consuming because streamers or studios need to review the whole streaming video. To reduce the need for human labor in this

* Corresponding author

[1] https://www.grandviewresearch.com/industry-analysis/video-streaming-market

burgeoning industry, there is an urgent need for effective highlight extraction methods that automatically identify highlights of streaming videos.

Extracting video highlights is an active research topic. Essentially, highlights are the most attractive video sections that are short enough to capture the gist of a video [2-4]. Due to advances in artificial intelligence and the availability of image pre-trained models, many recent deep learning approaches [2-6] have been developed to extract highlights from videos. These approaches normally divide a video into sequential segments and derive representative features, such as visual patterns and audio characteristics, to discover highlights. For instance, Yao et al. [5] employed the well-known AlexNet pre-trained model [7] and a 3D deep convolutional neural network [8] to extract spatial and temporal image features from fixed-size video segments. These features were then evaluated by a fully connected network to predict a score indicating the highlight probability of a given segment. While these studies are effective for detecting video highlights, most of them focus on gaming or out-door videos because the highlights of action-oriented videos generally involve visual or audio effects whose features can be successfully identified by deep neural networks to enhance the highlight extraction procedure. It is worth noting that many live streaming videos are now conversation-oriented, such as streamers who share personal experiences, or comment on trending memes, or share the latest buzz in their niche. In fact, the top most-watched streaming category on Twitch in 2021 was Just Chatting, which collected about 2,188k conversation-oriented streamer channels and was watched for more than 3 billion hours[2]. For this kind of streaming, streamers entertain their viewers not by producing exciting visual effects, but by comprehensively conversing and interacting with viewers. As there are few visual and audio effects, existing highlight extraction methods are generally unable to identify the representative features of highlight sections, and thus fail to distinguish highlights from this type of streaming video.

In this paper, we focus on extracting highlights of conversation-oriented streaming videos. To the best of our knowledge, this is the first work that explores the properties of conversation-oriented streaming videos for effective highlight extraction. Instead of analyzing visual and audio effects, we examine streamer conversations and viewer feedback. The proposed method is called COHETS (Conversation-Oriented Highlight

[2] https://sullygnome.com/game/Just_Chatting

Extraction using Textual Streams) and integrates two textual streams to automatically extract highlights. The first stream is the streamer conversation, and the other is the viewer messages posted in the chat boxes that reflect viewer feedback in streamer-viewer interactions. Our method first decomposes a streaming video into a sequence of discourse segments and encodes the streamer conversation within a segment into a streamer discourse embedding. As the relative position of a segment in a streaming video serves as a hint for highlight extraction, position embeddings are developed to enrich the streamer discourse embedding. At the same time, the viewer messages posted within a segment are encoded to form a viewer message embedding. However, since we noticed that viewer messages are sometimes off-topic and not always informative, an attention mechanism was designed to weight viewer messages when aggregating the viewer message embedding. Finally, the attentional message embedding together with the streamer discourse embedding enriched by the position information are fed into multilayer perceptrons and are leveraged by a self-adaptive weighting scheme to predict a highlight score. Segments with a high score are then selected to construct the video highlight.

Because no studies have ever investigated highlight extraction on conversation-oriented streaming videos, this paper is unique to examine the following three research questions:

**RQ1** – Can streamer discourses and viewer messages be used to effectively extract the highlights of conversation-oriented streaming videos? In particular, can COHETS outperform state-of-the-art highlight extraction methods which are normally based on visual pattern analysis?

**RQ2** – Which textual stream (streamer or viewer) is more valuable to conduct conversation-oriented streaming video highlight extraction? Can one stream dominate the other one in terms of highlight extraction?

**RQ3** – How do the designed position embeddings, message attention, and self-adaptive weighting affect the highlight extraction results?

To answer these questions, we collected a large amount of data on real world streaming videos. The results of experiments based on these data show that COHETS outperforms many state-of-the-art highlight extraction methods and text summarization methods. Also, the two textual streams and the designed components are useful for deriving highlights of conversation-oriented streaming videos in terms of highlight extraction

precision, recall, and F1 values.

The remainder of this paper is organized as follows. Section 2 reviews recent highlight extraction and text summarization works using deep learning technologies. After that, we detail the proposed highlight extraction method, and then in Section 4 we evaluate the system's performance. Section 5 discusses the research questions and the corresponding implications, and finally Section 6 summarizes our conclusions.

## 2. Related Work

In the past, methods of video highlight extraction would focus on a specific domain, such as sports, and would rely on extraction rules or heuristics defined by domain experts [9,10]. For instance, in Nepal et al.'s [9] study on basketball game highlight extraction, they compiled a set of extraction rules that simultaneously scans image and acoustic patterns, such as the appearance of scoreboards or the loudness of crowd cheers, to detect highlights of basketball scoring. While expert-defined rules and heuristics are effective, creating them is not easy. Therefore, in order to facilitate video highlight extraction, many studies started using machine learning techniques to automatically learn associations between visual-audio features and video highlights [11-15]. In [11], the authors studied highlight extraction in TV baseball games and decomposed a baseball game video into a set of candidate clips. The authors employed various supervised machine learning algorithms, such as support vector machines [16], to classify excited commentator speech which were then probabilistically fused with detected sounds of a bat hitting the ball to calculate the highlight probability of a candidate clip. Lee et al. [14] presented a highlight extraction method for egocentric videos which recorded activities from the first-person view through a wearable camera. Instead of directly analyzing video frames and visual features, the authors measured the importance of recorded objects and people with whom the camera wearer interacted. Labeled highlights were provided to train a regression function that estimated a highlight score of a video segment by considering object importance features, such as the distance of an object to the hands of the camera wearer and the frequency of the object occurrence.

More recently, deep learning has become the major methodology for video highlight extraction due to advances in deep convolutional networks and long-short

term memory (LSTM) architectures [17]. These improvements have enhanced image feature engineering and have therefore strengthened highlight extraction results. Below, we categorize recent deep learning studies as supervised and unsupervised, and then review the research works using textual information of live streaming videos for highlight extraction. Moreover, because the designed highlight extraction method analyzes texts to identify important discourse segments in streaming videos, our research is related to extraction-based text summarization whose goal is to extract representative text units (e.g., sentences) from the original text. At the end of this section, we review recent studies on extraction-based text summarization.

### *2.1 Supervised Deep Learning Highlight Extraction*

Yao et al. [5] employed techniques of pairwise learning to detect highlights of first-person videos and developed a ranking-based highlight extraction method named TS-DCNN. This method differs from traditional ones that analyze individual video segments in that it learns a highlight scoring function according to pairs of video segments. Each pair consists of one highlight segment and one non-highlight segment, and the scoring function aims to maximize the score difference between the segment pair, thereby effectively differentiating the highlight segment from the non-highlight one. Two convolutional neural networks, based on AlexNet and the C3D neural network, respectively capture the significant visual features of a video frame and the temporal dynamics of the features across frames. The networks enable the learning of vital image features related to highlights and their transformation in continuous frames. Finally, videos are summarized by skimming the non-highlight segments at a high-speed rate. Jiao et al. [18] also examined visual and temporal features of video frames for effective highlight extraction. They further adopted attention mechanisms to help the extraction model focus on (i) image regions meaningful to highlight extraction, and (ii) a series of frames that are worth watching. Although video contexts can be meaningful as a way to identify attractive video segments, most highlight extraction methods neglect context information and evaluate video segments independently. Wei et al. [19] addressed this segment-independent problem by means of a sequence-to-sequence highlight extraction model. They designed an encoder to sequentially receive the feature vectors of video frames and to generate a list of hidden states. The segment

detection unit functions as a decoder that considers both the encoder hidden states and the previous decoder state to output three highlight indicators: the starting position of a highlight segment, the ending position of the highlight segment, and the confidence score of the segment's highlight. The recurrent encoder-decoder mechanism enables this method to capture not only local segment features but also global context information.

The mvsDGCN system [20] was the first graph-based deep learning video summarization method. Given a set of videos (e.g., outdoor videos relating to the Great Wall of China), the method first divides the videos into a set of video shots. A graph is then constructed by measuring the pairwise similarities of the shots such that a weighted link is established if two shots are similar. The authors applied the graph convolution network (GCN) [21] to the weighted graph to derive node embeddings in accordance with neighbor relationships. Rather than using a fixed graph, the structure of the graph is dynamic during the convolution learning. Specifically, the graph used in the ($l$+1)th layer of graph convolution is based on the node embeddings output by layer $l$. The learned node embeddings are then classified to select meaningful shots for multi-video summarization. In [4], the authors developed an object-aware neural graph model called VH-GNN that detects video objects to determine whether a clip of continuous frames is part of a video highlight. This method first applies two pre-trained models, namely the Region Proposal Network (RPN) [22] and RoIAlign [23], to a video frame to detect object box boundaries and then generates features for video objects. The objects subsequently form the nodes of a spatial graph that distills object features through an attention and message passing mechanism. The distilled object features of all frames in a clip are leveraged by a temporal graph to predict a highlight score for the clip. And in order to enhance the highlight extraction results, a multi-stage loss function including a highlight classification loss and a ranking loss was implemented to optimize the two graph networks.

In addition to visual features of video frames, Rochan et al. [24] also considered the browsing history of individual users to construct personalized video highlights. The highlight extraction model consists of one history encoder and one CNN-based network. The history encoder examines the browsing history of a user to induce the preference style of the user. The visual features of video frames extracted by the CNN-

based network along with the preference style are fed into a temporal-adaptive normalization layer to discover attractive video segments relevant to user preferences. To diversify the video summaries, Li et al. [25] developed a global diverse attention that weights a frame by considering its dissimilarity to all the other frames in a video. The authors first adopted a pre-trained model (e.g., GoogleNet) to extract visual features of video frames, which are evaluated by the global diverse attention to increase the weight of a frame dissimilar to the other frames. The weighted feature vector of a frame is fed into an embedding layer and then a regression layer. The regression layer determines the importance of the frame while the embedded layer combined with a determinantal point process removes redundant key frames for a diversity of video summaries. Zhu et al.'s [26] hierarchical attention approach for video summarization is like many previous works by first extracting the visual features of a frame by means of a pre-trained CNN model. To enrich the frame-level features, the hierarchical attention model consists of an intra-block attention and an inter-block attention. The intra-block attention aggregates the features of a series of frames within a video block whose features are then leveraged by the inter-block attention to derive the video-level feature. These multiscale features are then fed into a regression network to predict the important score of a frame.

### *2.2 Unsupervised Deep Learning Highlight Extraction*

One challenge of the above supervised highlight extraction methods is the preparation of training highlights because highlight labeling is usually labor-intensive. To remove this time-consuming task, the unsupervised approach makes use of logical assumptions to implicitly derive highlight extraction models. Yang et al. [27] constructed a highlight extraction system for specific domains, like surfing, by assuming that videos under four minutes are highlights. This is because short-form videos are likely to be edited by the video owners and are the most exciting and engaging parts of the original videos. In the training phase, domain-specific keywords were submitted to crawl short-form videos from the web. The videos were then segmented into snippets and fed into the 3D convolutional neural network to extract representative visual features. Finally, a recurrent autoencoder with LSTM cells was trained to implicitly identify highlight segments. Segments with a low reconstruction loss were regarded as highlights because

their features are consistent with those of the short-form videos. Note that the above assumption brings noise (i.e., non-highlight short-form videos) into the model training. To lessen the impact of noisy data, the authors enhanced the autoencoder with a shrinking exponential loss. Ringer and Nicolaou [28] also employed autoencoders to identify video highlights in an unsupervised manner. In contrast to the previous method, the authors treated video frames with high reconstruction loss as highlights because they focused on video game streaming videos which are normally lengthy and share similar backgrounds. Ringer and Nicolaou thus assumed highlights are anomalies in videos and are associated with high reconstruction loss. In addition to applying autoencoders to the video frames of streamers and games, the authors also evaluated game audio whose reconstruction loss is based on the short-term Fourier transform and principal component analysis. The reconstruction losses of these three components are summed together such that frames whose losses are above a pre-defined threshold are considered as anomalies and are thus categorized as highlights. Lan and Ye [29] investigated LSTM and autoencoders for unsupervised video summarization as well. To enhance the video summarization performance, the variational autoencoder-based summarizer was incorporated into a generative adversarial network in which the adversarial discriminator strives to discriminate the original videos from the videos reconstructed from the summarizer.

In contrast to these methods that identify highlights by means of frame or segment losses, Xiong et al. [3] developed a model that explicitly predicts a highlight score for a video segment. To minimize the effort of preparing training data, the authors also assumed that short-form videos are highlights and adopted techniques of pairwise learning to train a prediction model. Each training pair consists of one short-form video and one long-form video, and the objective of the model is to maximize the highlight scores for the short videos while minimizing the scores for long videos. To overcome training noise, like when a long-form video is classified as a highlight, a group of latent variables was introduced to determine whether the pairwise score ranking is valid. Rani and Kumar [30] adopted an unsupervised clustering approach to identify the key frames of a video. The selected key frames function as the thumbnails of social media videos. Similar to Ringer and Nicolaou's assumption, a frame is regarded as important if its visual features (e.g., the color histogram) are significantly different to those of the previous frame. A self-organization map clustering method grouped important frames

into clusters automatically, and for each cluster, Euclidean distances of the frames in terms of the visual features were computed to identify a pair of frames with a very large distance. These frame pairs were selected to represent the highlight of the given video.

### *2.3 Highlight Extraction using Textual Information*

In addition to visual features, some recent studies have started using textual information for video highlight extraction. Fu et al. [6] focused on online game streaming highlight extraction and developed Joint-lv-LSTM, which is based on their CNN-RNN method called V-CNN-LSTM. Joint-lv-LSTM examines both video frames, which are the only input of V-CNN-LSTM, and viewer messages. In the preprocessing phase, streaming videos are first sliced into frames which are concatenated as segments with a sliding window approach. Next, the ResNet-34 model [31] is employed to the segment frames to extract important visual features. At the same time, the viewer messages posted within the segment are concatenated and fed into a character-level LSTM to produce an embedding that represents the viewer intention in that segment. Finally, the visual features of the segment frames in combination with the message embedding are fed into a multi-layer perceptron to predict a highlight score for that segment. The evaluations show that viewer messages are helpful side information to enhance streaming video highlight extraction. Han et al. [2] also investigated viewer messages for online game streaming highlight extraction. The authors noticed that viewer messages for live streaming normally contain a lot of Internet slang and emoticons. To better comprehend the view messages, the designed highlight extraction model biGRU-DNN built a language model which produces word embeddings for messages. Also, a bidirectional Gated-Recurrent Unit (biGRU) architecture that sequentially processes viewer messages was implemented to encode viewer messages with context information. To highlight streaming videos with only a few audience messages, Wang et al. [32] presented a language transfer model that infers plausible message embeddings from visual segment features. The authors prepared a set of streaming videos with audience messages to train the language transfer model. An LSTM-based encoder was first employed to derive the visual embedding of a segment from video frames, through which the transfer model learns to discriminate a message posted within the segment from a random message. The outputs of the model's last layer represent the message

embedding which is conjoined with the visual embedding to predict the attractive segments of a streaming video.

### *2.4 Extraction-based Text Summarization*

Extraction-based summarization methods normally explore the semantic, sentiment, and syntactic information of a text to identify representative text units, such as sentences, for summaries. For instance, the ELSA method [33] uses Singular Value Decomposition (SVD), an effective latent semantic analysis approach based on matrix factorization, to discover the semantic concepts of documents. Unlike more common summarization approaches that manage terms independently, ELSA employs a frequent itemset mining algorithm to group terms that frequently co-occur. The discovered concepts are enhanced by using co-occurring terms and sentences that are related to the major concepts are then selected to construct document summaries. Srivastava et al. [34] also explored semantic concepts of text for effective text summarization. The authors employed Latent Dirichlet Allocation (LDA), a probabilistic latent semantic model, to assign a topic label to each sentence. In this model, sentences of the same topic are vectorized by averaging the Word2Vec vectors [35] of their tokens, and then these sentences are clustered to select representative sentences as summaries. To better comprehend the semantics of short sentences, Mohd et al. [36] expanded sentence contents by means of pre-trained language models. The expansion replaces the tokens of a sentence with a set of semantically-related words obtained by Word2Vec. Then, a clustering procedure is applied to the expanded sentences, called big vectors, to discover the meaningful topics of a document. The designed summarization method computes the weights of the terms in big vectors, which are combined with syntactic statistics, like the counts of nouns, verbs, and proper nouns, to select the important sentences of document topics.

Mutlu et al. [37] modeled summary extraction as a binary classification task and used deep learning techniques to predict if a sentence is representative. Their designed network architecture consists of one LSTM layer connected with two fully-connected network layers that sequentially process semantic embeddings of sentence tokens. The authors further ensembled the semantic embeddings with 42 hand-crafted sentence features to improve their summarization results. Jing et al. [38] adopted GCNs to

convolute sentence embeddings through word embeddings. The summarization model considers the semantic similarity and the syntactic dependency of words in a sentence to polish the adjacent matrices of GCNs. The convoluted sentence embeddings, along with the document embedding that is iteratively convoluted through the sentence embeddings of a document are used to predict important scores of sentences for text summarization. Hu et al. [39] examined sentiment information of texts to summarize hotel reviews. To identify various aspects of the reviews, the authors clustered review sentences according to the association of nouns and the sentiment of adjectives in sentences. A sentence's information value was derived through the position, length, and count of cue phases in the sentence. Further, the sentence's authority and credibility were respectively measured by the number of "helpful" votes received by the review author and the stability of the author's ratings. For each review aspect (cluster), sentences that were informative and written by reliable authors were selected to construct comprehensive review summaries.

To sum up, video highlight extraction is an active and challenging research topic. Many previous studies focused on action-oriented videos (e.g., for sports and video games) and they relied on visual features distilled through deep convolutional neural networks. While these studies demonstrate remarkable highlight extraction performance, they are not necessarily effective for conversation-oriented streaming videos because visual features or patterns do not constitute the main video highlights. While extraction-based text summarization investigates semantic, syntactic, and even sentiment features of a text, the analyzed text units (e.g., sentences) are so short that they may affect the extraction performance of the methods when dealing with long discourses of streamers and lengthy viewer messages. In this work, we therefore do not use convolutional networks and visual features, but instead propose a two-stream neural network architecture that examines streamer discourses and viewer messages. Since these two types of textual information can reflect the intentions of both streamers and viewers, they are helpful for extracting meaningful highlights from conversation-oriented streaming videos.

## 3. COHETS System

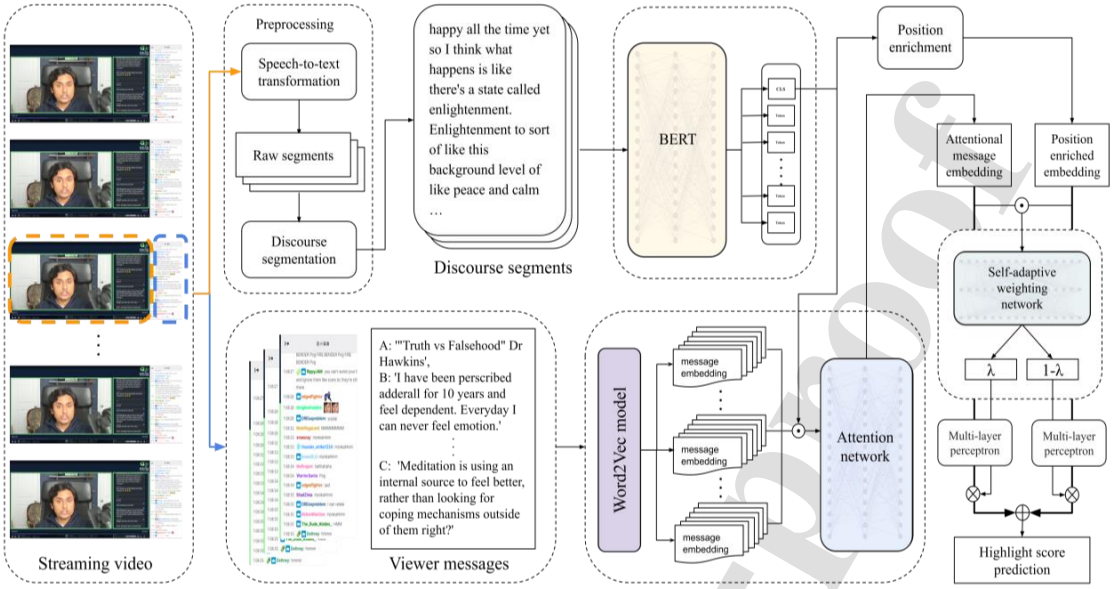



**Fig. 1.** The system architecture of COHETS.

In this section, we present the COHETS system which extracts highlights from conversation-oriented streaming videos by simultaneously examining the two textual streams of streamer conversation and viewer messages. Figure 1 shows our system architecture. For a streaming video, the preprocessing stage first partitions it into a series of segments according to the discourses of the streamer conversation. Then, the state-of-the-art language model BERT [40] is applied to the spoken sentences of the streamer within a discourse segment to derive the semantic embedding of the streamer discourse. And to enhance the highlight extraction results, we present a position enrichment mechanism that enriches the streamer discourse embedding by considering the position of a discourse segment in the video. Because streaming highlights are supposed to resonate with viewers from not only the streamer conversation but also the interactivity between viewers and streamers [41], COHETS also evaluates viewer messages posted during a live streaming to represent the response and engagement of viewers to the streamer-viewer interactions. We collected all the viewer messages posted within a discourse segment, and an attention mechanism was designed to aggregate the intention of the messages as a message embedding. Lastly, the embeddings of the streamer discourse and the viewer messages are respectively fed into a multi-layer perceptron and are leveraged by a self-adaptive weighting scheme to

predict a highlight score of the segment. Segments with a high score are selected to construct the video highlight. We detail the highlight extraction method in the following sections.

### *3.1 Video Preprocessing and Discourse Segmentation*

As mentioned in the related work section, methods of highlight extraction normally divide a video into a sequence of image frame segments. We do not do this because our highlight extraction is based on the textual information of streaming videos. Instead, we examine streamer conversation to discover *discourse segments*, each of which consists of a list of spoken sentences that stand for a coherent dialogue. Given a streaming video, we first measure the variation of acoustic intensity to partition the video into a series of *raw segments* $\{r_1, r_2, \ldots, r_K\}$. In other words, acoustic silence forms the delimiter of two consecutive raw segments. Next, the Google speech-to-text cloud service[3] is employed to convert the audio conversation of the streamer in a raw segment into spoken sentences. However, we observed that the silence-based segmentation is so error-prone that the resultant raw segments barely comprise meaningful dialogue. This is because live streaming involves streamer-viewer interactions: when streaming, streamers often stop to digest the feedback (i.e., messages) from the viewers, which fragments the discourse with unexpected silences. To have our method process coherent segments, we merge adjacent raw segments if they share the same discourse.

[3] https://cloud.google.com/speech-to-text

Merging adjacent raw segments is closely related to the next sentence prediction task [40] of BERT that guides BERT to distill the semantics of words and sentences by differentiating the relation of two given sentences. In the next sentence prediction task, BERT is asked to conduct a binary classification that judges whether one given sentence narratively follows the other sentence. Because a discourse segment consists of pairs of adjacent raw segments belonging to the same dialogue, the same classification approach is implemented to form the discourse segments of a streaming video. As shown in Figure 2, given the spoken sentences of two adjacent raw segments $r_k$ and $r_{k+1}$, we first derive their CLS embedding through BERT. The CLS embedding is a special contextual embedding that BERT uses to represent the given text. The embedding is then fed into a multi-layer perceptron attached by a softmax function to output the probability that the two raw segments are part of the same dialogue. We sequentially apply the classification to all pairs of adjacent raw segments and produce a sequence of *discourse segments* $\{d_1, d_2, ..., d_N\}$ by merging the adjacent raw segments that belong to the same discourse.

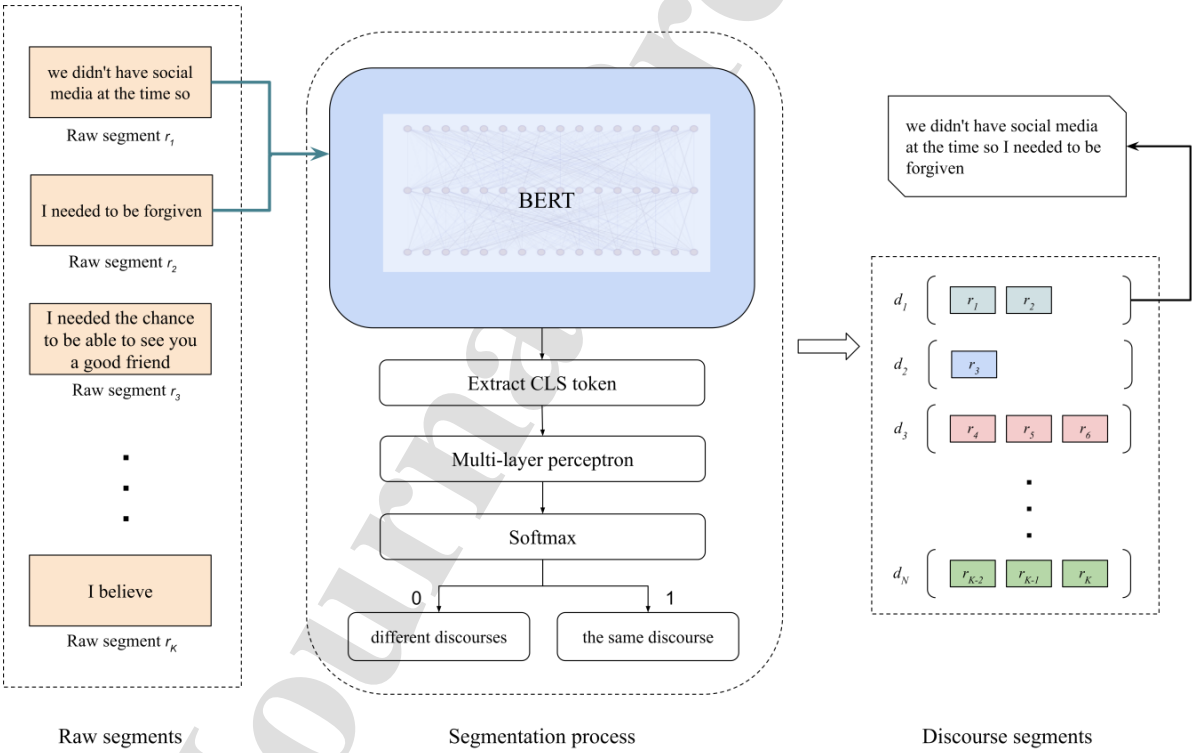



**Fig. 2.** The process of discourse segmentation.

*3.2 Streamer Discourse Embedding and Position Enrichment*

Having created discourse segments, BERT is again applied to the spoken contents of the segments to obtain a series of *streamer discourse embeddings* $\{s_1, s_2, ..., s_N\}$. Note that in the segmentation step, the input to BERT is a pair of adjacent raw segments because the aim of the segmentation task is to detect discourse boundaries. Here, all the spoken sentences of a discourse segment are fed into BERT to obtain the CLS embedding that represents the semantics of streamer discourse in a discourse segment. Intuitively, we could integrate BERT with a downstream classification task that receives the streamer discourse embedding of a segment and calculates the probability that the segment is a part of a video highlight. However, we noticed that the relative position of a segment to a streaming video is a clue for highlight extraction. In particular, the first-half segments are better indicators than the ending segments. This is because a live streaming is normally too lengthy for most viewers to finish watching. Streamers are thus motivated to show their best content early in order to impress viewers and retain them for the whole streaming. In a sense, the task of extracting highlights from streaming videos in terms of streamer discourses is similar to the extraction-based text summarization [42] that selects representative text units (e.g., sentences) from the original text to construct a summary. To identify representative text units for summaries, many features, such as text similarities [43] and latent topics [44], have been explored. Notably, position-related features, such as the order of sentences in a document, have been validated as playing key roles in effective text summarization because the beginning and ending text units tend to capture the gist of a text [45, 46]. Based on these findings, the proposed position enrichment mechanism incorporates position information of segments into our highlight extraction process.

Specifically, position embeddings are developed to enrich the streamer discourse embeddings of a video. We partition a video into $P$ positions, and a discourse segment $d_n$ is aligned with a position number $pos_n$ by using the following equation:

$$pos_n = \left\lceil {}^{n}/_{(N/P)} \right\rceil, \quad (1)$$

where $N$ is the number of discourse segments and $pos_n$ is a positive integer within [1, $P$]. To derive position embeddings, our position enrichment mechanism first represents each position by the one-hot encoding. As shown in Figure 3, the one-hot vector of

position $pos_n$ is passed through a linear embedding layer having 768 outputs. This output size is identical to the length of a streamer discourse embedding based on the BERT-based pre-trained model. The outputs as a whole, denoted as $p_{pos_n}$, are regarded as the position embedding of position $pos_n$, which is combined with the streamer discourse embedding to obtain the *position-enriched streamer discourse embedding* $\check{s}_n$ and $\check{s}_n = s_n + p_{pos_n}$.

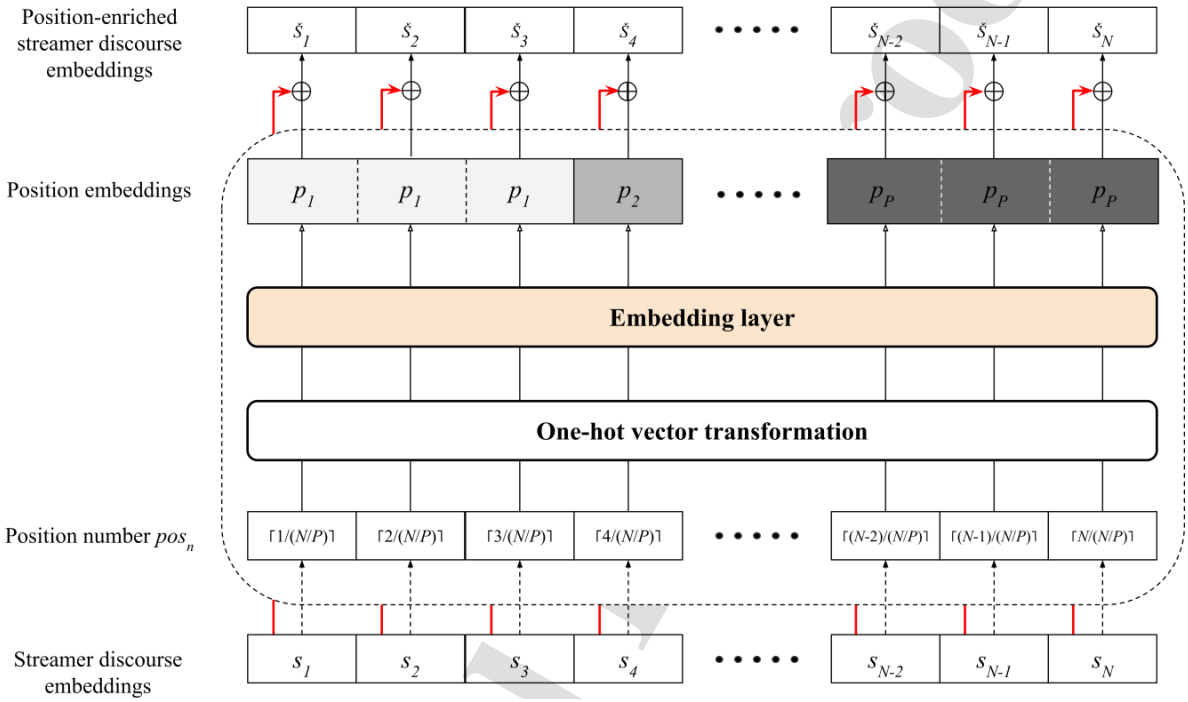



**Fig. 3.** The position enrichment mechanism.

### *3.3 Viewer Message Embedding and Attention*

One unique trait of live streaming is the streamer-viewer interaction in that the messages posted by viewers in chat boxes reveal their reactions to streamer discourses. As highlights are supposed to interest viewers, we take viewer messages as an important source of highlight extraction. As with streamer discourse embeddings, it might be assumed that we can apply a pre-trained language model to viewer messages and encode the intention of viewers as an embedding vector. However, two obstacles make this approach impossible: (i) casual language usage, and (ii) diverse viewer opinions. Viewer messages often contain Internet slang, emoticons, or acronyms, which normally express the emotion of viewers and are thus meaningful; unfortunately, most pre-

trained language models are built against formal texts and genres, like wiki pages or books. They therefore cannot recognize these casual tokens and fail to discover the intention of the viewers. To resolve this difficulty, we derive embedding vectors of viewer messages by means of the skip-gram of Word2Vec [35], a well-known word embedding model that exhibits an extraordinary ability to model word semantics by producing similar embedding vectors for close words. Given a series of textual tokens $\{w_1, w_2, \ldots, w_T\}$, the skip-gram approach estimates the embedding vector of each unique token by maximizing the following sum of log probabilities:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-H\le h\le H, j\neq 0} \log p(w_{t+h}|w_t), \tag{2}$$

where $T$ is the number of the textual tokens, and $H$ is the window size of surrounding tokens used to estimate word embeddings. Basically, the model aims to predict the surrounding tokens (i.e., $w_{t+h}$'s) based on a given token $w_t$. The prediction probability is calculated as follows:

$$p(w_{t+h}|w_t) = \frac{exp(e_{w_t} \cdot e_{w_{t+h}})}{\sum_{v\in V} exp(e_{w_t} \cdot e_v)}, \tag{3}$$

where $e_{w_t}$ is the embedding vector of token $w_t$ and $V$ is the set of unique tokens. The function *exp* returns the exponential of the vector inner product. By using the skip-gram approach, we derive embeddings of both normal message tokens (i.e., words) and casual message tokens. Then, for each message posted by a viewer $msg_i = \{w_1, w_2, \ldots, w_l\}$ which contains a series of tokens, we average the embedding vectors of the tokens to obtain the message embedding vector $m_i$ as follows:

$$m_i = \frac{1}{l}\sum_{t=1}^{l} e_{w_t}. \tag{4}$$

It should be noted that in [2] we used biGRU to encode viewer messages. When we developed COHETS and tried biGRU to extract embeddings from viewer messages, we found that the highlight extraction performances were good and comparable to those based on the above mean-pooling strategy. However, the current design of COHETS already contains three learnable components (i.e., the position embedding, message attention network, and self-adaptive weighting network), and adding one more learnable component (i.e., biGRU) would not improve system performance. This is why

we decided to only use the mean-pooling strategy and keep COHETS as simple and efficient as possible.

Regarding the diversity of viewer opinions, popular streaming attracts a lot of viewers and a huge number of viewer messages. For instance, in our experiment dataset, each evaluated streaming video contains around 18,000 messages. The message contents are so diverse that not every message is crucial to highlight extraction, so in order to effectively use viewer messages, we designed an attention mechanism that weights viewer messages in terms of streamer discourses and the highlight extraction task. Bahdanau et al. [47] described the bottleneck problem of machine translation: machine translation methods are normally based on the encoder-decoder architecture such that the encoder aggregates all coding information (called hidden states) of the input tokens to construct one context vector used by the decoder to emit output tokens; however, when dealing with a long input, the single context vector is unable to carry all the hidden states, and this leads to the bottleneck of machine translation. To solve this problem, the authors suggest using attention techniques to customize the weights of hidden states when emitting each output token. In a similar way, we tackle the enormous amount of diverse viewer messages with our attention mechanism (Figure 4) that captures the intention of the viewers in a discourse segment by means of the following equation:

$$\check{v}_n = \sum_{m_i \in M_n} (W_{msg} \cdot [s_n \odot m_i])\, m_i, \tag{5}$$

where $\check{v}_n$ is the *attentional message embedding* under discourse segment $d_n$, $M_n$ denotes the set of Word2Vec message embeddings within discourse $d_n$, and $W_{msg}$ is a vector standing for the learning parameter of our attention mechanism. To weight the individual viewer messages, our attention mechanism first concatenates the embeddings of $s_n$ and $m_i$. The concatenated vector then passes through the attention layer parameterized by $W_{msg}$, which correlates $s_n$ and $m_i$ with the highlight extraction task in order to calculate the weight of $m_i$. Finally, the attentional message embedding $\check{v}_n$ is the sum of all message embeddings calibrated by their attention weights.

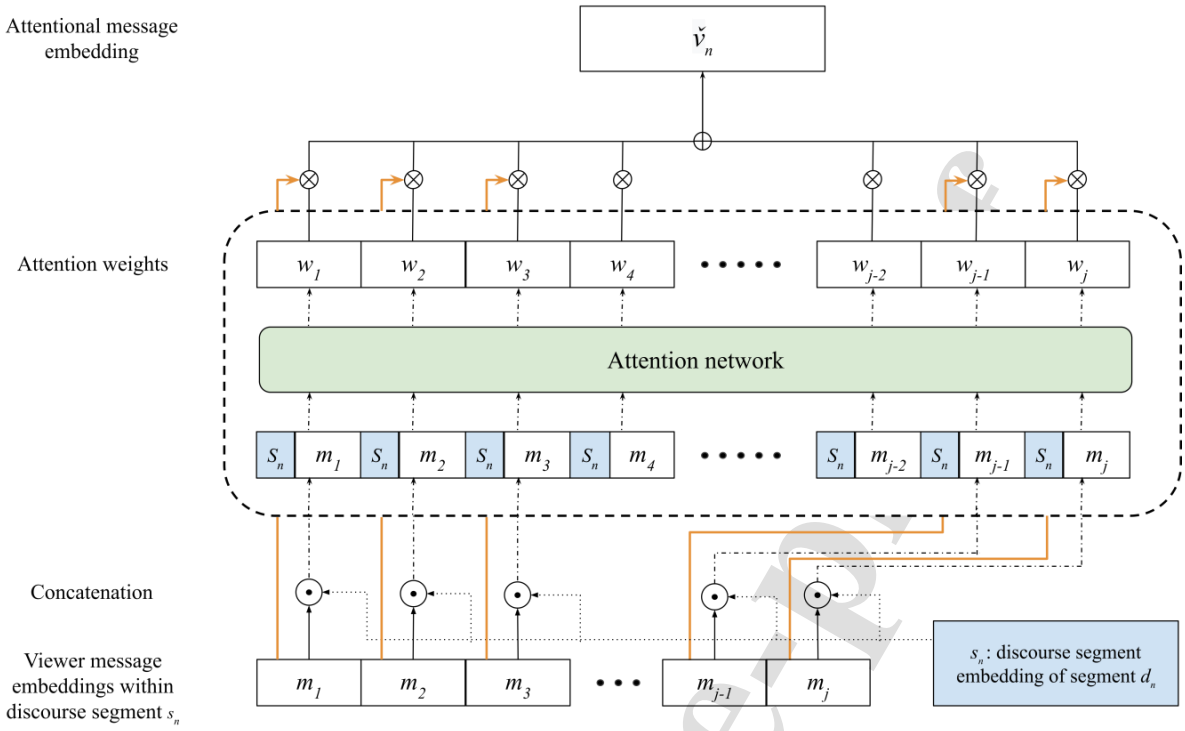



**Fig. 4.** The viewer message attention mechanism.

### *3.4 Highlight Extraction and Self-Adaptive Weighting Scheme*

For each discourse segment $d_n$, we consider both its position-enriched streamer discourse embedding $\check{s}_n$ and the attentional message embedding $\check{v}_n$ to estimate the probability that the segment is a part of a highlight. A baseline approach to integrate the two embeddings for the probability prediction feeds each embedding into a multi-layer perceptron (MLP) that outputs a highlight score between 0 and 1, and averages the two prediction scores by means of a weighting scale $\lambda$ whose range is [0, 1]. The scale $\lambda$ calibrates the contribution of the two textual embeddings when constructing highlights. For instance, by setting $\lambda = 0.75$, we fix the influence of streamer discourses to be three times larger than viewer messages. Rather than setting a fixed $\lambda$ for all segments, we developed a *self-adaptive weighting scheme* that customizes $\lambda$ in accordance with the given $\check{s}_n$ and $\check{v}_n$, and predicts the highlight probability $\hat{y}_n$ of $d_n$ as follows:

$$h_l^s = \phi(W_l^s h_{l-1}^s + b_l^s) \quad (6)$$

$$\hat{y}_n^s = \sigma(W_L^s h_{L-1}^s + b_L^s) \quad (7)$$

$$h_l^m = \phi(W_l^m h_{l-1}^m + b_l^m) \quad (8)$$

$$\hat{y}_n^m = \sigma(W_L^m h_{L-1}^m + b_L^m) \quad (9)$$

$$\lambda_n = \sigma(W_{self} \cdot [\check{s}_n \odot \check{v}_n]) \quad (10)$$

$$\hat{y}_n = \lambda_n \hat{y}_n^s + (1 - \lambda_n)\hat{y}_n^m, \quad (11)$$

where $L$ denotes the number of MLP layers. The symbols $h_l^s$ and $h_l^m$ are the outputs of layers $l$ of the streamer discourse embedding MLP and the viewer message embedding MLP, respectively, and they are the iterated inputs of the $l$+1th layers. The inputs of the two MLPs (i.e., $h_0^s$ and $h_0^m$) are $\check{s}_n$ and $\check{v}_n$. The activation function $\phi$ here is Relu, and $\sigma$ is the sigmoid function that turns the outputs $\hat{y}_n^s$ and $\hat{y}_n^m$ of the two MLPs into probabilities. Variables $W_l^s$, $W_l^m$, $b_l^s$, and $b_l^m$ are weight matrices and biases, and they are our model parameters. Equation 10 calculates the self-adaptive weight $\lambda_n$ in that the sigmoid function $\sigma$ restricts $\lambda_n$ to the range [0, 1]. As shown in Figure 1, the weighting scheme first concatenates the two embeddings and feeds the aggregated embedding into a self-adaptive weighting layer parameterized by the model parameter $W_{self}$ that learns to leverage the two embeddings. By doing so, the scale $\lambda_n$ is adaptive to the content of the two embeddings, i.e., the intentions of the streamer and the viewers. In the experiment section, we examine the effect of our self-adaptive weighting scheme.

### *3.5 Model Training and Highlight Extraction Loss*

Here, we introduce the loss function that allows us to minimize the error of highlight extraction in order to acquire appropriate model parameters during the training stage. To train our highlight extraction method, we collected a number of streaming videos. Let $Q$ = [<$d_1$, $y_1$>, <$d_2$, $y_2$>, ..., <$d_L$, $y_L$>] be a set of training instances in which $d_l$ is a discourse segment decomposed from the training videos. Symbol $y_l$ is $d_l$'s label, and it is 1 if the segment is a part of a highlight; otherwise, it is 0. Our highlight extraction loss $HE_{Loss}$ is defined as follows:

$$HE_{Loss}(Q) = \frac{1}{L}\sum_{l=1}^{L}(\lambda_l * S_{Loss}(< d_l, y_l >) + (1-\lambda_l) * M_{Loss}(< d_l, y_l >)), \quad (12)$$

where $S_{Loss}$ and $M_{Loss}$ denote the extraction losses caused by using position-enriched streamer discourse embeddings and attentional message embeddings, respectively; and $\lambda_l$ is the self-adaptive weight of training segment $d_l$. In this study, we measure $S_{Loss}$ and $M_{Loss}$ in terms of the binary cross entropy [48].

$$S_{Loss}(< d_l, y_l >) = -[y_l \cdot \log \hat{y}_l^s + (1-y_l) \cdot \log(1-\hat{y}_l^s)] \quad (13)$$

$$M_{Loss}(< d_l, y_l >) = -[y_l \cdot \log \hat{y}_l^m + (1-y_l) \cdot \log(1-\hat{y}_l^m)], \quad (14)$$

where $\hat{y}_l^s$ and $\hat{y}_l^m$ are the highlight probabilities of segment $d_l$ estimated by using the position-enriched streamer discourse embedding (i.e., Equation 7) and the attentional message embedding (i.e., Equation 9), respectively. By minimizing $HE_{Loss}$, our model parameters are guided to distill the intentions of the streamer and viewers to extract meaningful streaming highlights.

## 4. Experiment

In this section, we first introduce the evaluation dataset, performance metrics, and evaluation procedure. Then, we verify the effects of the two textual embeddings and system components on the highlight extraction performance. Finally, we compare the proposed method with state-of-the-art highlight extraction methods and extraction-based text summarization methods.

### *4.1 Evaluation Dataset and Metrics*

**Table 1**

The statistics of our dataset and four well-known video datasets.

| | Ours | YouTube | SumMe | TVSum | VTW* |
|---|---|---|---|---|---|
| Video type | c.o. | a.o. | most a.o. | a.o. | a.o. |
| Number of videos | 65 | 712 | 25 | 50 | 4,000 |
| Length of videos (min.) | 9,128 | 1,430 | 66 | 210 | 6,000 |
| Length of the labeled highlights (min.) | 1,355 | 274.4 | 5.45 | 31.5 | 440 |
| Number of viewer messages | 1,174,589 | n.a. | n.a. | n.a. | n.a. |
| Number of message tokens | 4,406,777 | n.a. | n.a. | n.a. | n.a. |

c.o./a.o.: conversation-oriented/action-oriented. *: Although the dataset contains 18,100 videos, only 4,000 videos (1.5 minutes duration on average) have labeled highlights (3.3 seconds duration on average). Since the four other datasets have no viewer messages, their statistics related to messages are not available (n.a.).

Video highlight extraction is an active research topic, and because of this, several studies have released video datasets with labeled highlights. However, as mentioned in the related work section, most of these studies focus on action-oriented videos whose datasets are related to gaming or out-door activities. Since we were unable to find any public datasets for conversation-oriented streaming videos, we compiled a dataset to evaluate the proposed method. Table 1 details the statistics of the evaluated videos. To distinguish the uniqueness and necessity of our dataset, Table 1 also compares our dataset with four well-known video datasets, including the YouTube dataset [15], SumMe dataset [49], TVSum dataset [50] and VTW dataset [51]. Compared with these four well-known datasets, the size (i.e., the length of videos) of our data is significantly larger. Note that the compared datasets have no viewer messages and their videos are action-oriented. So far, our dataset is the only dataset specific for conversation-oriented streaming video highlight extraction. To promote this new research topic and to attract attention from highlight extraction researchers, the dataset has been released[4] and is also being continuously extended with more labeled streaming videos.

We collected streaming videos from Twitch, one of the largest streaming platforms that provides a diverse range of categories of streaming channels and videos. Here, we selected for evaluation 65 streaming videos categorized under "Talk Shows & Podcasts" from two famous streamers. Streamer @HealthyGamer_GG is an addiction psychiatrist whose streaming videos are designed to help gamers overcome their game addiction by discussing mental issues. According to official Twitch statistics[5], this streamer now has around 500K subscriptions and has accumulated around 5M views on his channel. The other streamer @Markiplier is a famous influencer with more than 2M subscriptions. This streamer shares life moments and provides thought-provoking perspectives in his streaming; his channel has a total of 12.8M views. We crawled all the viewer messages of the testing videos from chat boxes to evaluate the effect of viewer intentions on highlight extraction. The total length of the evaluated videos is 9,128 minutes (152.1 hours) with each streaming video being about 2.34 hours long, and having around 18,600 messages and 163 streamer discourse segments. Figures 5 and 6 show the distributions of the video lengths and the segment durations. Almost all

[4] https://github.com/lopeterlo/Conversation-oriented-dataset

[5] https://twitchtracker.com/healthygamer_gg/statistics, the statistics of @HealthyGamer_GG
https://twitchtracker.com/markiplier/statistics, the statistics of @Markiplier

the videos range from 1.5 hours to 3 hours in length, and a great portion of the segments are less than one minute. The short segment length is because streamers seldom talk continuously in order to make their streaming lively and vivid.

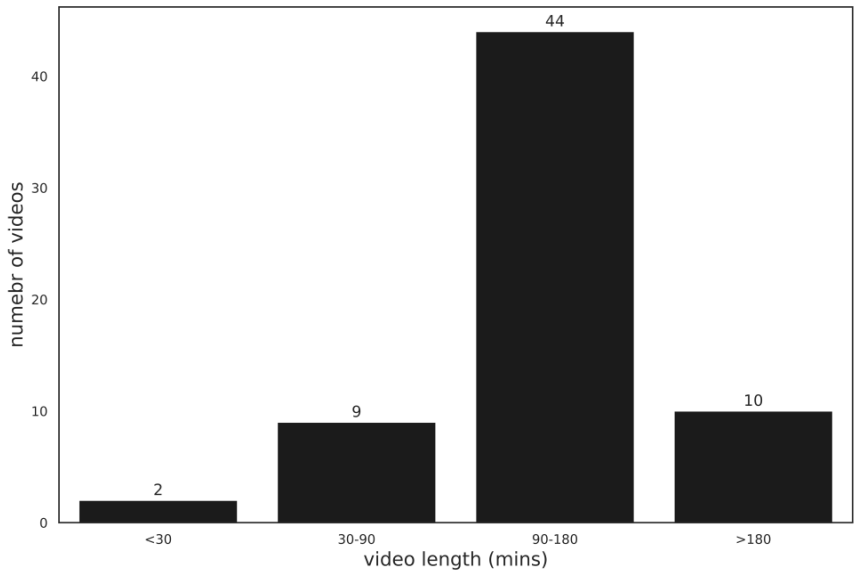



**Fig. 5.** The length distribution of the evaluated videos.

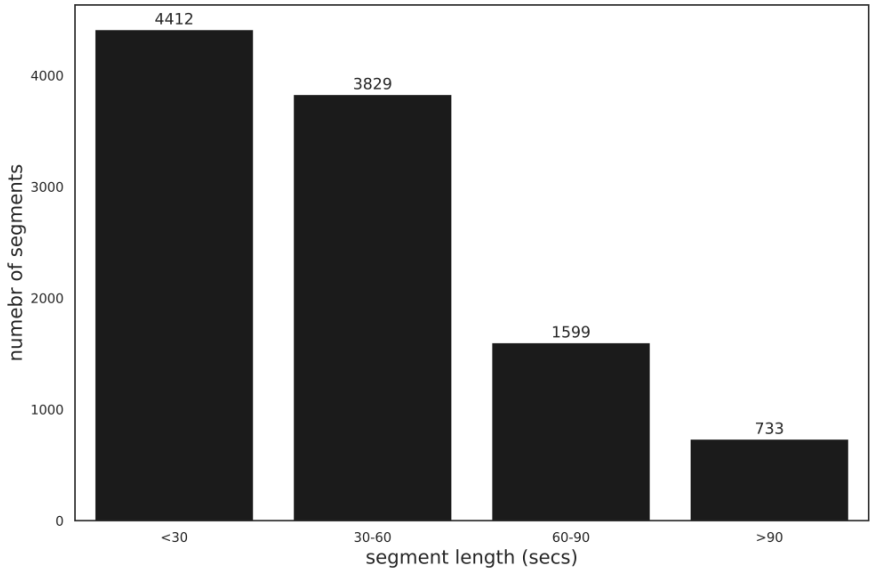



**Fig. 6.** The distribution of segment durations.

Before annotating the data, we reviewed the literature to compile three key guidelines for creating trustworthy datasets.

1. Diverse annotators: As described in [15, 50, 52], data annotation is normally conducted by many domain experts to minimize the biases that happen if data is annotated by only a few people. Therefore, we invited six experts who are heavy streaming users to annotate the highlights of the collected videos.
2. Two-stage annotation: To ensure the correctness of data annotation, many studies (e.g., [53, 54]) adopt a double check mechanism in which the data annotated by experts are validated by supervisors. In our study, data were validated by the experts and the second author of this study who together reviewed the annotated highlights.
3. Multiple checkpoints: In addition to the above guidelines we learned from the relevant literature, we set up checkpoints to remove last-minute annotations. We also periodically checked the progress of the highlight annotations. Slow or fast annotators were reminded to maintain highlight labeling progress and quality.

Before the experts annotated the highlights of the videos, an orientation was held to ensure the quality and usability of the annotated data. The experts were asked to first watch the videos thoroughly. Then, they had to think about the video content for a while before annotating the highlight sections. Moreover, in order to ensure that the annotated highlights were concise and clear, the highlights were restricted to no longer than 15% of the video length. The annotation was time-consuming and resulted in a lengthy highlight editing process: each video took around 5 times the video length. Also, to help clarify our dataset, the reasons for highlight annotations were provided, e.g., this section is a highlight because of the excited audience responses. This auxiliary information can encourage future research not only on highlight extraction but also for different disciplines to investigate the important factors that makes a streaming video popular.

We adopted the conventional 5-fold cross validation [55] to obtain performance results. Specifically, we evenly divided the streaming videos into 5 disjointed subsets and evaluated our highlight extraction performance in 5 runs. Each run selected one subset of the videos for testing and trained the extraction model by using the remaining 4 subsets. For each testing video, we ranked all its discourse segments according to

their highlight probability scores. The top segments whose length reached *K*% of the video length were predicted as the video highlight. The predicted highlights of all five runs were compared with the ground truth to report the precision@*K*, recall@*K*, and F1@*K*, which are the major highlight extraction evaluation metrics adopted in many studies (e.g., [2, 6, 9, 10, 25, 50]) to measure system performances. The definitions of the metrics are as follows:

$$\text{precision@}K = \frac{\left|highlight_{predicted} \cap highlight_{ground\_truth}\right|}{\left|highlight_{predicted}\right|} \quad (15)$$

$$\text{recall@}K = \frac{\left|highlight_{predicted} \cap highlight_{ground\_truth}\right|}{\left|highlight_{ground\_truth}\right|} \quad (16)$$

$$\text{F1@}K = \frac{2 * \text{precision@}K * \text{recall@}K}{\text{precision@}K + \text{recall@}K}, \quad (17)$$

where $highlight_{predicted}$ and $highlight_{ground\_truth}$ stand for the predicted highlights and the ground truth, respectively. The absolute values measure their lengths in the unit of seconds. The precision@*K* measures the percentage of the predicted highlights that coincide with the ground truth. The recall@*K* reports the percentage of the ground truth that are predicted as highlights. The F1@*K* is the harmonic mean of the precision and recall scores, and is frequently used to judge the superiority of prediction systems. Note that PyTorch[6], a well-known deep learning library, was adopted to implement our highlight extraction method. The optimizer we used to learn network parameters was AdamW [56] with a 1e-5 learning rate. At the same time, to prevent overfitting, we inserted dropout layers in our networks with a dropout rate of 0.2. The dimension of our streamer discourse embeddings was 768 because the embeddings were based on the BERT base pre-trained model whose embedding length is 768. The dimension of the Word2Vec-based attentional message embeddings was 300, as suggested in [35]. For position embeddings, the parameter *P* was set at 5, which means we divided every video into 5 position parts. In the next section, we examine the highlight extraction performance under different settings of *P*. Table 2 lists the settings of the system hyper-parameters.

[6] https://pytorch.org/

**Table 2**

The system hyper-parameter settings.

| | |
|---|---|
| Learning optimizer | AdamW |
| Learning rate | 1e-5 |
| Batch size | 8 |
| Dropout rate | 0.2 |
| Dim. of streamer discourse embeddings | 768 |
| Dim. of position embeddings | 768 |
| Dim. of viewer message embeddings | 300 |
| *P* | 5 |

**Table 3**

Highlight extraction performance of system components.

| | precision@10 | recall@10 | F1@10 |
|---|---|---|---|
| Streamer Discourse Embeddings | 0.1919 | 0.1436 | 0.1644 |
| Position-Enriched Streamer Discourse Embeddings | 0.2139 | 0.1598 | 0.1829 |
| Message Embeddings | 0.1860 | 0.1385 | 0.1588 |
| Attentional Message Embeddings | 0.1982 | 0.1471 | 0.1689 |
| COHETS | **0.2350** | **0.1744** | **0.2002** |

*4.2 Effect of System Components*

Here, we evaluate our system components. We first investigate the influence of streamer discourses and viewer messages on highlight extraction, and then examine the proposed position and attention mechanisms. Finally, we compare the performance with and without the self-adaptive weighting scheme. Table 3 shows the performance results on our dataset. Although the scores of our method are not very high, with the precison@10 being only 0.235, this is because the evaluated streaming videos are very long and the expert-labeled highlights consist of only 14% of the evaluated videos, which means that identifying highlight sections is very challenging. Nevertheless, our method still outperforms many state-of-the-art highlight extraction methods, as shown in the next section. The highlight extraction results based on viewer messages (i.e., the scores of Message Embeddings) are relatively inferior to those based on streamer discourses (i.e., the scores of Streamer Discourse Embeddings). This is because live streaming normally attracts a large number of viewers with different interests, which leads to a wide content range of viewer messages that can distract the extraction model based on viewer messages. It is worth noting that by using the attention mechanism, both the precision and recall scores improved with the F1 score increasing from 0.1588

to 0.1689. This improvement indicates that the proposed attention mechanism successfully distills viewer messages and is helpful for highlight extraction.

The results based on streamer discourses also improved when incorporating the streamer discourse embeddings with the position embeddings. This outcome validates the observation that the relative position of a segment to a streaming video is a good clue for highlight extraction because streamers generally optimize the show flow of live streaming to attract as large an audience as possible. As a further demonstration of position embedding, Figure 7 illustrates the effect of the parameter $P$, which determines the granularity of the position embedding. The figure shows that a large $P$ (i.e., $P$ = 12) deteriorates the highlight extraction performance, and this is because a large $P$ partitions a video so much that the resultant position embeddings are too specific to discover highlights distributed over consecutive positions. Conversely, a small $P$ (i.e., $P$ = 3) cannot distinguish possible highlight positions, which also affects highlight extraction performance. As the setting of $P$ = 5 produces a good highlight extraction performance, we used this setting in the following experiments. By using both the attentional message embeddings and the position-enriched streamer discourse embeddings, our method achieves the best results. In other words, both textual streams of streamer discourses and viewer messages need to be considered together to enhance the highlight extraction of streaming videos.

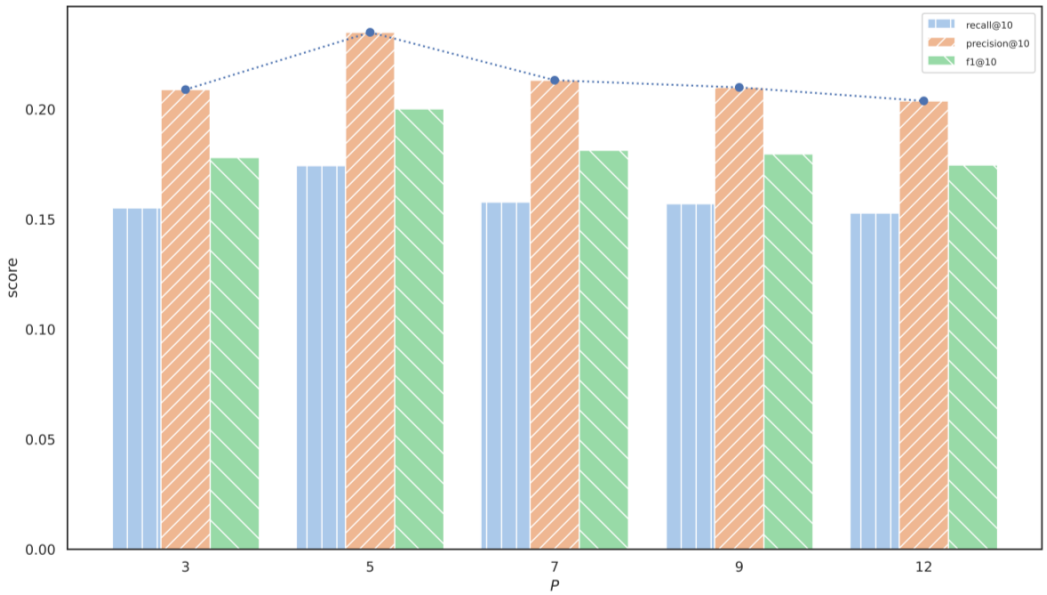



**Fig. 7.** The effect of the parameter $P$.

In addition to the two textual embeddings and parameter $P$, we also investigated the effect of our self-adaptive weighting scheme. Figure 8 compares the performances of our method with and without self-adaptive weighting. Without the self-adaptive weighting, $\lambda$ (i.e., the scale used to average the prediction scores) is a fixed value against all the evaluated segments. In Figure 8, the performance scores of the fixed-$\lambda$ approach fluctuate and there is no obvious tendency of $\lambda$ in favor of highlight extractions. That is, for some segments, streamer discourses are valuable because of the engaging speech of the streamers. Sometimes, viewer messages are informative since they point out a segment is pleasing, and a certain number of highlight segments involve both excited viewer feedback and interesting streamer discourse. As a fixed $\lambda$ cannot customize the individual weights of streamer discourses and viewer messages, the results are inferior. In contrast, self-adaptive weighting is superior because it is capable of calibrating $\lambda$ in accordance with the streamer and viewer embeddings.

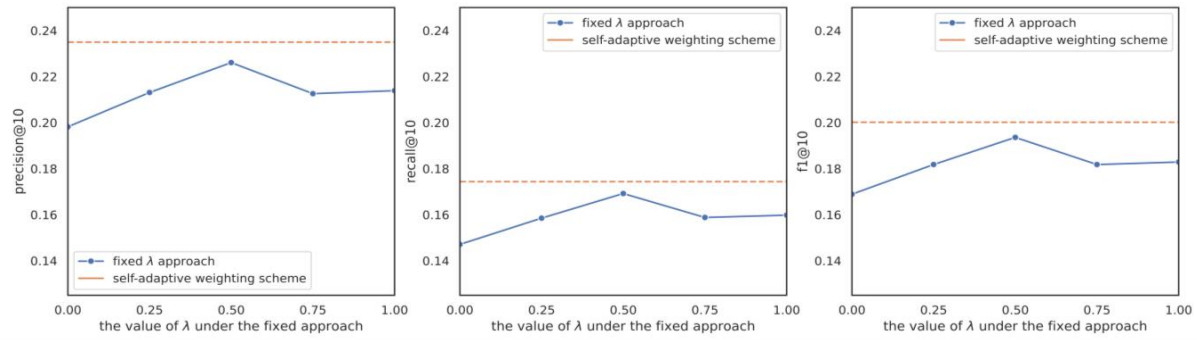



**Fig. 8.** Comparisons with and without the self-adaptive weighting scheme.

To better understand the influence of segment durations, Figure 9 illustrates the distributions of highlight prediction scores against segment durations. Compared with long segments, short segments (i.e., segments less than 30 seconds) tend to have a low highlight prediction score. This is because short segments normally involve casual streamer discourses and viewer messages that our method would not consider as highlights. It is interesting to see that segments with medium and long lengths have similar score distributions. In other words, our method does not simply prefer long segments. A segment will be predicted as a highlight segment as long as its streamer discourses or viewer messages are meaningful.

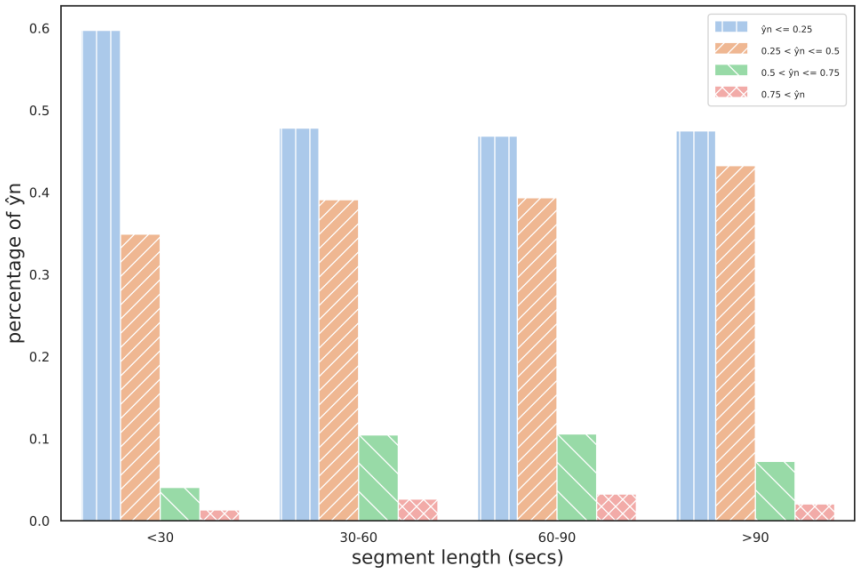



**Fig. 9.** The influence of segment durations on highlight prediction scores.

Figure 10 further examines the performance scores with respect to different video lengths. For short videos (i.e., the videos whose lengths are less than 30 minutes), COHETS extracts the highlights successfully. This is because short videos are generally topic-focused, and so it is relatively easy to identify their highlight segments. By contrast, lengthy videos contain many discourse segments and discovering the highlight segments is more challenging. Nevertheless, COHETS can be considered robust since the performances on medium and long videos are close.

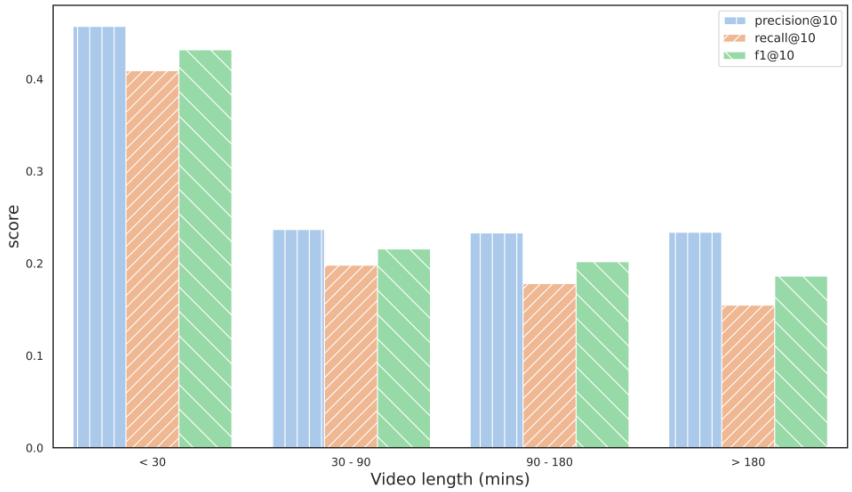



**Fig. 10.** The influence of video lengths on highlight extraction performances.

Finally, we examined the quality of the extracted highlights under different settings of *K*, i.e., the percentage of the top-scoring segments selected for constructing highlights. As shown in Figure 11, the recall values are positively proportional to *K*. This is because recall is a non-decreasing metric, so the recall value increases as more segments are added into the highlights. However, since the selection of the highlight segments is based on the ranking of their highlight scores, a large *K* includes low-score segments that affect the precision of our method. The high precision scores of small *K*'s (e.g., the 0.29 precision at $K = 1$) indicate that the top segments we estimated correspond well with the expert-labeled highlights, thus showing the potential of our method in constructing short streaming highlights.

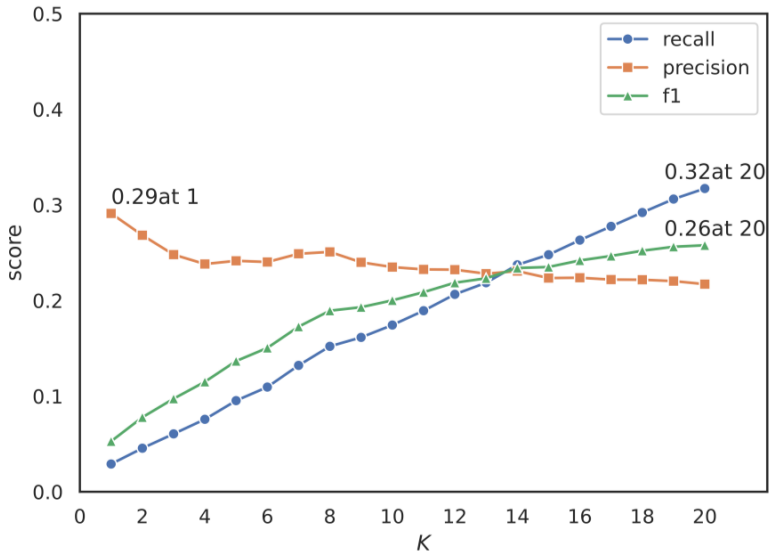



**Fig. 11.** The performances of the extracted highlights under different *K* percentages.

### *4.3 Comparison with Other Highlight Extraction and Text Summarization Methods*

The above experiments thoroughly evaluated our system components. Next, we compare COHETS with five state-of-the-art video highlight extraction methods (TS-DCNN [5], V-CNN-LSTM [6], Joint-lv-LSTM [6], biGRU-DNN [2], and VH-GNN [4]), and five extraction-based text summarization methods (the LDA-based method [34], big-vector based method [36], semantic-LSTM based method [37], forward selection method, and backward selection method). We selected the video highlight extraction methods for comparison because they are recent deep-learning-based methods that have been presented in prestigious research venues. Moreover, they have demonstrated extraordinary performance on video highlight extraction, especially VH-GNN which adopts advanced graph neural networks. Among these methods, V-CNN-LSTM, Joint-lv-LSTM, and biGRU-DNN are specific to live streaming videos, while the latter two also use viewer messages to extract highlights. Strictly speaking, only biGRU-DNN and our method are purely text-based approaches insofar they both examine textual information of streaming videos for highlight extraction. In contrast, V-CNN-LSTM, Joint-lv-LSTM, TS-DCNN, and VH-GNN employ sophisticated models (e.g., AlexNet or ResNet) to extract graphical or visual features from video frames to extract highlights. A comparison of these methods clearly reveals the benefit of textual information for streaming highlight extraction.

The text summarization methods were selected for comparison because their task is to extract representative text units from the given text, which is related to our highlight extraction process. Moreover, the methods rely on advanced NLP techniques (e.g., word vectors of language models, latent semantic analysis, and deep neural networks) to explore the semantic and syntactic information of text. Comparing these summarization methods thus reveals the effects of advanced NLP techniques when dealing with streaming videos. Note that both the LDA-based method and the big-vector based method use pre-trained language models to determine the semantics of words. The LDA-based method further explores the latent topics of text units, i.e., discourse segments in our experiment, by means of Latent Dirichlet Allocation. In addition to semantic information, the big-vector based method investigates syntactic information (e.g., noun, verb, and proper noun counts) of the given text to discover representative text units. The semantic-LSTM based method employs pre-trained language models to derive semantic information of text units too. The word vectors of a text unit are then input into an LSTM layer connected with two fully-connected network layers to predict whether or not the text unit is a highlight. The forward and backward selection methods are classic summarization baselines [57, 58] and generate highlights by respectively extracting the initial and end segments of a video. Also, we examined a random selection baseline which randomly selected discourse segments as highlights. To ensure fair comparisons, all the methods were implemented using public packages and the hyper-parameters were set as suggested in the methods' original papers. The same 5-fold cross validation was conducted to obtain their highlight extraction performances.

Table 4 lists the performance scores of the compared methods under $K = 10$, and Figure 12 shows the performances under different settings of $K$. We expected the five video highlight extraction methods would produce comparable results in light of the superior performances reported in their respective papers. However, the methods were inferior. As mentioned in the related work section, current highlight extraction methods mostly focus on action-oriented videos whose highlights generally involve rich visual effects. Therefore, when dealing with the evaluated streaming videos which are conversation-oriented, these methods are ineffective since visual effects are not indicative of the video highlights. In particular, V-CNN-LSTM, Joint-lv-LSTM, TD-DCNN, and VH-GNN were inferior because they rely on sophisticated network

architectures to distill visual patterns for highlight extraction. Instead of discovering visual patterns, our method examines the textual information of these conversation-oriented videos and produces better highlight extraction results. Perhaps surprisingly, the random-selection baseline is not worse than some of the highlight extraction methods. But as mentioned above, these methods extract highlights by analyzing visual effects which are mostly absent in the evaluated conversation-oriented streaming videos. These methods have no real clues and are thus forced to randomly select highlight segments. As a result of this, their performance scores are close to those of the random-selection baseline.

**Table 4**

The performance scores of the compared methods under $K$=10.

| | precision@10 | recall@10 | F1@10 |
|---|---|---|---|
| V-CNN-LSTM | 0.1289 | 0.0929 | 0.1080 |
| Joint-lv-LSTM | 0.1326 | 0.0956 | 0.1110 |
| TS-DCNN | 0.1202 | 0.0857 | 0.1000 |
| biGRU-DNN | 0.1749 | 0.1180 | 0.1409 |
| VH-GNN | 0.1453 | 0.1045 | 0.1216 |
| LDA-based method | 0.1677 | 0.1061 | 0.1299 |
| Big-Vector based method | 0.1246 | 0.0826 | 0.0994 |
| Semantic-LSTM based method | 0.1688 | 0.1076 | 0.1314 |
| Forward-Selection | 0.0807 | 0.0514 | 0.0628 |
| Backward-Selection | 0.0906 | 0.0492 | 0.0637 |
| Random-Selection | 0.1323 | 0.0980 | 0.1125 |
| COHETS | **0.2350** | **0.1744** | **0.2002** |

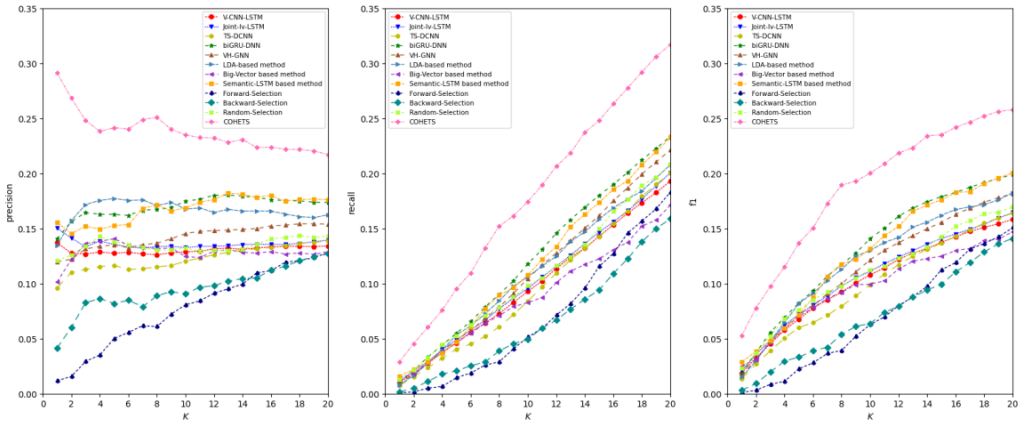



**Fig. 12.** The performances of the compared methods under different settings of $K$.

The value of the textual information in identifying conversation-oriented streaming highlights can also be validated by the experiment results of biGRU-DNN and Joint-

lv-LSTM. Under a small *K*, the precision scores of the two methods are much better than those of the other video highlight extraction methods. The biGRU-DNN method is also text-based and examines viewer messages to extract streaming video highlights. However, as the method neglects streamer discourses, its performance is inferior to ours. It is interesting to note that the performance scores of biGRU-DNN are lower than those of the Attentional Message Embeddings reported in Table 3. Since both approaches are based on viewer messages, our superiorly performing Attentional Message Embeddings again demonstrates the advantage of our attention mechanism in distilling viewer messages relevant for extracting highlights. By exploring viewer messages, Joint-lv-LSTM enhances V-CNN-LSTM in terms of precision, recall, and F1. Nevertheless, Joint-lv-LSTM processes viewer messages character by character which means that the resultant message embeddings overlook word meaning. In contrast, biGRU-DNN and our method embed viewer messages in terms of word vectors and therefore outperform Joint-lv-LSTM.

The performance scores of the LDA-based and semantic-LSTM based summarization methods are higher than those of the purely visual-pattern based highlight extraction methods (i.e., V-CNN-LSTM, TS-DCNN, VH-GNN). The results again validate the use of textual information and the effect of NLP techniques when extracting the highlights of conversation-oriented streaming videos. Nevertheless, the compared summarization methods are still inferior to our method. The reason for their ineffectiveness has to do with the characteristic differences between streaming video highlight extraction and extraction-based text summarization. First, text summarization normally assumes the inputs are well-written texts. However, when dealing with streaming texts, especially viewer messages, language models built against formal texts and genres are incapable of capturing the gist of the casual texts, and this affects extraction performance. Further, the text units evaluated by text summarization methods are often short (e.g., one or a few sentences), whereas a discourse segment often consists of multiple spoken sentences of streamers along with dozens of viewer messages. The text would be too large and unwieldy for the summarization methods to work out the semantics of a segment. We remedied the above issues by designing the message attention mechanism and by building a language model upon viewer messages. This is why COHETS produces a better highlight extraction performance. Note that both the forward and backward selection methods are ineffective. In Section 5, we

present the distribution of the expert-labeled highlights across positions (Figure 14), which reveals that streamers tend not to present their best content either at the beginning or end of a live streaming. For this reason, the forward and backward text summarization baselines are inferior.

In sum, the comparison with the state-of-the-art text summarization methods shows that extracting highlights from conversation-oriented streaming videos is non-trivial. By considering the characteristics of live streaming, typically associated with a massive amount of textual information and casual language usage, COHETS enhances current NLP techniques with the designed message attention mechanism, self-adaptive weighting scheme, and position embedding, thus achieving good highlight extraction performance.

*4.4 Case Study*

The merits of COHETS can be seen in the following highlight extraction result from the testing video H_28 "Talking with Asmongold! | MENTAL HEALTH MAY !nonprofit." We use this result as a case study. Figure 13 shows that the highlights extracted by COHETS coincide with two highlight sections labeled by the experts. Note that the extracted highlights are normally short video fragments. This is because a highlight extraction unit of COHETS is a discourse segment. In practice, these fragments can be concatenated if they are temporally close in order to provide better user experience. The experts labeled the time period [00:56:11-01:05:19] as a highlight, during which the invited guest of the live stream talked about not having a goal in life. Many viewers posted emotional responses at that moment by typing a lot of encouraging messages, like "Nothing wrong in retiring when you are not motivated anymore" and "we can help fix that." Because COHETS examines viewer messages, we successfully detected this highlight section. After this, streamer @HealthyGamer_GG, who is an addiction psychiatrist, interacted with the guest and showed him how to face and overcome his frustration. Their interactions (starting from 01:15:33 to 01:18:30) were the core of this live stream and this highlight was also detected by COHETS by analyzing streamer discourses.

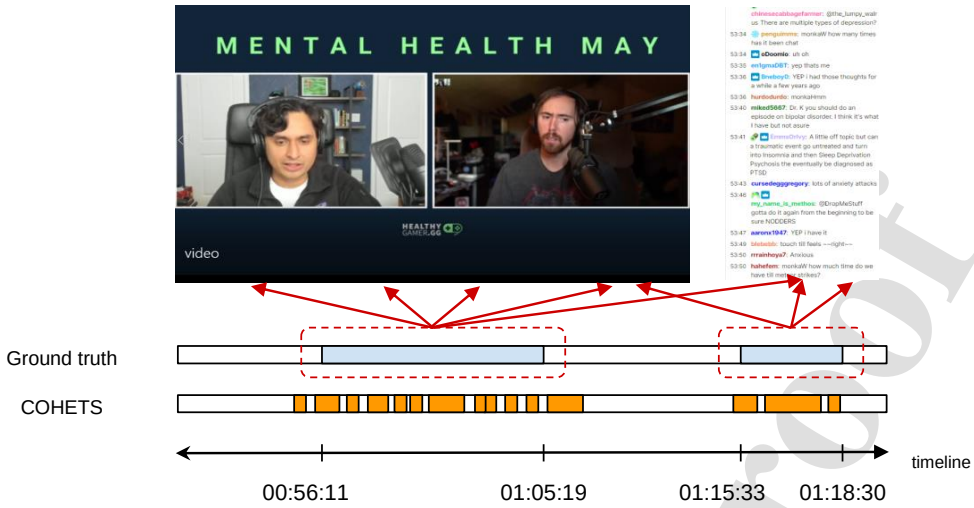



**Fig. 13.** The case study based on "H_28 - Talking with Asmongold! | MENTAL HEALTH MAY !nonprofit."

Here, the extraction results of the compared highlight extraction methods are not presented because these visual-feature based methods have difficulty identifying these two highlight sections. The case study once again validates COHETS as superior for distinguishing highlights from conversation-oriented videos and shows that existing highlight extraction methods cannot handle this type of streaming videos.

## 5. Discussion and Implications

In this section, we discuss the experiment results and answer the three research questions mentioned in the Introduction.

**RQ1** – Can streamer discourses and viewer messages be used to effectively extract the highlights of conversation-oriented streaming videos? In particular, can COHETS outperform state-of-the-art highlight extraction methods which are normally based on visual pattern analysis?

The evaluations show that COHETS surpasses the state-of-the-art methods. The comparison indicates that the information of the two textual streams (streamer discourses and viewer messages) are more useful than graphical or visual patterns of video frames for extracting highlights in conversation-oriented streaming videos. Still, the precision, recall, and F1 values of COHETS are not very high. This demonstrates the challenge and difficulty of extracting highlights from conversation-oriented streaming videos. While the textual streams are informative, the contents of

conversation-oriented streaming videos are often wildly diverse and rarely repeated. The learned model always encounters unexpected discourses and viewer feedback that lowers the highlight extraction performance. In contrast, action-oriented videos (e.g., gaming videos) often share similar scenes and pictures that makes extracting highlights comparatively simple. COHETS's performances also reveal there is substantial room for improvement in the area of extracting highlights for conversation-oriented streaming videos using textual information. As more and more live streaming videos are now conversation-oriented, we have released the evaluation dataset and continue to enlarge it for future investigations of this important and practical problem.

**RQ2** – Which textual stream (streamer or viewer) is more valuable to conduct conversation-oriented streaming video highlight extraction? Can one stream dominate the other one in terms of highlight extraction?

Our evaluations show that streamer discourses are more valuable than viewer messages. This is because the information of viewer messages are often random and diverse and contain a variety of opinions from different audiences. Casual tokens of viewer messages also affect the highlight extraction outcome. On the other hand, streamer discourses tend to be relatively focused, which, as a consequence, improves highlight extraction results. While the performance results based on streamer discourses are better than those based on viewer messages, this does not mean viewer messages should be ignored. In fact, our method achieves the best performance by using both textual streams. We need to consider that the two textual streams convey intentions of two important roles in information communication, i.e., information deliverers and information receivers who both generally hold different perspectives regarding the discussed topic due to communication noise [59]. Using these two streaming embeddings together unquestionably improves the highlight extraction results.

**RQ3** – How do the designed position embeddings, message attention, and self-adaptive weighting affect the highlight extraction results?

Our experiments show that the designed components are helpful for the highlight extraction task. Here, we detail the effect of the position embeddings. Figure 14 illustrates the distribution of the expert-labeled highlights across positions. Also, the distributions of the predicted highlights with and without position embeddings are shown to demonstrate the effect of position embeddings. In contrast to the common text

practice of providing summary information at the beginning or end of a text [45, 46], the labeled highlights in our dataset often occur at the two-five and three-five positions of a video. This is probably because viewers generally do not join a live stream at the very beginning, which forces streamers to postpone showing their best content till later. Also, as viewers do not always stay tuned for the whole duration, the ending part of a live stream is not usually very climactic. So, it is reasonable that highlights often occur at the two-five and three-five positions of a live stream. Comparing the two highlight prediction distributions, the percentage of the predicted highlights occurring in the second and third positions boosts significantly if streamer discourse embeddings are enriched by position embeddings. This result validates the idea that positional information in streaming plays an important role in our highlight extraction process. Also, the designed position embeddings successfully capture the positional phenomenon of expert-labeled highlights that enhances the streamer discourse embedding in terms of highlight extraction.

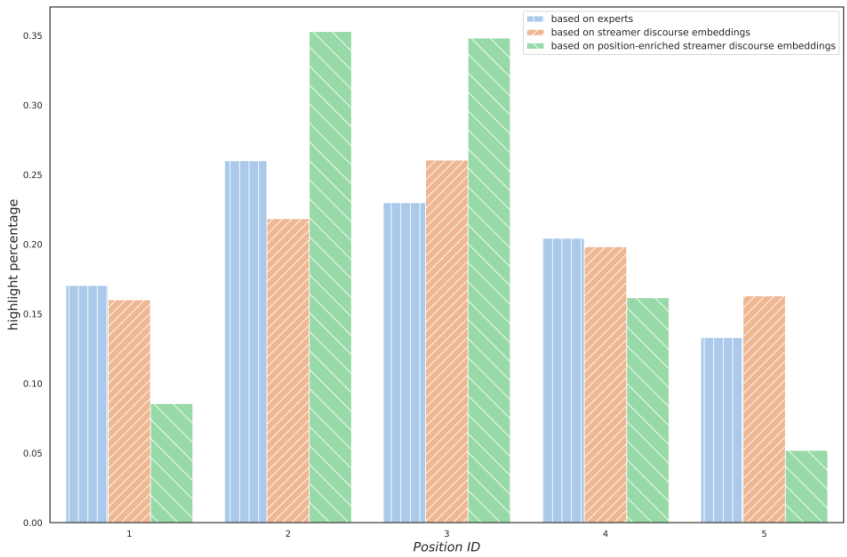



**Fig. 14.** The highlight distributions over different positions.

Regarding theoretical implications, our literature survey reveals that research on extracting highlights from conversation-oriented streaming videos has not been well-addressed. Our study thus contributes to this new research topic. Also, differing from existing highlight extraction methods which are normally based on visual image

features, our proposed model focuses on examining the textual information of streaming videos. Our findings suggest that for conversation-oriented streaming videos, textual information is more valuable than visual features of video frames for extracting highlights. This is a new perspective for streaming video highlight extraction. Further, to make streamer discourses and viewer messages work, we designed additional embedding enrichment techniques (i.e., the position embeddings and the message attention mechanism) and a self-adaptive weighting scheme, which proved crucial. Finally, as a preliminary study of conversation-oriented streaming video highlight extraction, the experiment dataset which contains a large quantity of labeled streaming videos has been publicly released to encourage more study of this new and important research topic.

As for practical implications, our research benefits streaming platform providers and streamers in different ways. First, the system architecture of COHETS is straightforward. Difficulties of the system implementation such as informal language usage and discourse segmentation, are pinpointed to facilitate model implementation and practice. Moreover, as conversation-oriented streaming videos are currently the top watched streaming category, COHETS and the extracted highlights can enable streamers and platform providers to attract even larger audiences which can in turn increase both popularity and revenue.

## 6. Conclusions and Future Work

The abundance of conversation-oriented streaming videos available on the Internet has turned streaming platforms into treasure-houses of useful and interesting information for people of all walks of life. Unfortunately, the massive numbers of streaming videos also overwhelm people searching for specific information. To reduce the burden of information overload, and also to increase channel exposure and subscriptions, it would be very useful to provide highlights for users. In this paper, we have developed COHETS to automatically extract highlights from conversation-oriented streaming videos. The results of our literature review indicate that our research is the first study to investigate conversation-oriented streaming video highlight extraction. Differing from previous highlight extraction methods that mostly focus on action-oriented videos and that heavily rely on visual features, COHETS simultaneously examines streamer

discourses and viewer messages. Our method further enhances the embeddings of the textual information by incorporating our proposed position embeddings and message attention mechanism. Experiments based on real world streaming data show that COHETS outperforms several state-of-the-art highlight extraction and text summarization methods.

Nonetheless, our performance scores indicate that discovering highlights of conversation-oriented streaming videos is challenging and the results suggest room for improvement. Since conversation-oriented streaming highlight extraction remains under-addressed in the literature and since this type of video is currently the most popular type of streaming video, we hope our study will stimulate future research in this area. In the future, we will keep enhancing the embeddings distilled from streamer dicourses and viewer messages, especially in light of recent language models that have started investigating BERT enhancement in terms of scalability and training efficiency. We will therefore enhance our embedding process with advanced language models in order to process long streamer discourses and also to better comprehend how viewer messages are distracted by informal tokens such as social buzzwords and slang. Moreover, in this study, we encode streamer discourses and viewer messages independently, but in the future we aim to better capture streamer-viewer interactions and more effectively identify the most attractive part of a video by also investigating encoding archtectures to concurrently process these types of textual information. Finally, to produce better highlight extraction results, we will explore more side information such as the sentence intensity of viewer messages and stream discourse within a segment, which can indirectly reflect the emotional state of information senders and receivers.

**Acknowledgments**

This research was supported by the Ministry of Science and Technology of Taiwan under grant MOST 109-2410-H-002-071-MY2 and the National Science and Technology Council under grant NSTC 111-2410-H-002-052

**References**

[1] Zhang, S., Liu, H., He, J., Han, S., & Du, X., A Deep Bi-directional Prediction Model for Live Streaming Recommendation, Information Processing & Management, 58(2) (2021) 102453. https://doi.org/10.1016/j.ipm.2020.102453

[2] Han, H.-K., Huang, Y.-C., & Chen, C. C., A Deep Learning Model for Extracting Live Streaming Video Highlights using Audience Messages, in Proceedings of the 2019 2nd Artificial Intelligence and Cloud Computing Conference, (2019) 75-81. https://doi.org/10.1145/3375959.3375965

[3] Xiong, B., Kalantidis, Y., Ghadiyaram, D., & Grauman, K., Less is More: Learning Highlight Detection from Video Duration, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, (2019). 1258-1267. https://doi.org/10.1109/CVPR.2019.00135

[4] Zhang, Y., Gao, J., Yang, X., Liu, C., Li, Y., & Xu, C., Find Objects and Focus on Highlights: Mining Object Semantics for Video Highlight Detection via Graph Neural Networks, in Proceedings of the AAAI Conference on Artificial Intelligence, (2020) 12902-12909. https://doi.org/10.1609/aaai.v34i07.6988

[5] Yao, T., Mei, T., & Rui, Y., Highlight Detection with Pairwise Deep Ranking for First-Person Video Summarization, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (2016) 982-990. https://doi.org/10.1109/CVPR.2016.112

[6] Fu, C.-Y., Lee, J., Bansal, M., & Berg, A., Video Highlight Prediction using Audience Chat Reactions, in Proceedings of the Conference on Empirical Methods in Natural Language Processing, (2017) 972-978. http://dx.doi.org/10.18653/v1/D17-1102

[7] Krizhevsky, A., Sutskever, I., & Hinton, G. E., ImageNet Classification with Deep Convolutional Neural Networks, Advances in Neural Information Processing Systems, 25 (2012) 1097-1105.

[8] Tran, D., Bourdev, L., Fergus, R., Torresani, L., & Paluri, M., Learning Spatiotemporal Features with 3D Convolutional Networks, in Proceedings of the IEEE International Conference on Computer Vision, (2015) 4489-4497. https://doi.org/10.1109/ICCV.2015.510

[9] Nepal, S., Srinivasan, U., & Reynolds, G., Automatic detection of 'Goal' segments in basketball videos, in Proceedings of the ninth ACM International Conference on Multimedia, (2001) 261-269. https://doi.org/10.1145/500141.500181

[10] Tjondronegoro, D., Chen, Y.-P. P., & Pham, B., Integrating Highlights for More Complete Sports Video Summarization, IEEE multimedia, (2004) 11 22-37. https://doi.org/10.1109/MMUL.2004.28

[11] Rui, Y., Gupta, A., & Acero, A., Automatically Extracting Highlights for TV Baseball Programs, in Proceedings of the eighth ACM International Conference on Multimedia, (2000) 105-115. https://doi.org/10.1145/354384.354443

[12] Otsuka, I., Nakane, K., Divakaran, A., Hatanaka, K., & Ogawa, M., A Highlight Scene Detection and Video Summarization System using Audio Feature for a Personal Video Recorder, in Proceedings of International Conference on Consumer Electronics, (2005) 223-224. https://doi.org/10.1109/ICCE.2005.1429798

[13] Zhang, B., Dou, W., & Chen, L., Combining Short and Long Term Audio Features for TV Sports Highlight Detection, in Proceedings of the 28th European Conference on Advances in Information Retrieval, (2006) 472-475. https://doi.org/10.1007/11735106_44

[14] Lee, Y. J., Ghosh, J., & Grauman, K., Discovering Important People and Objects for Egocentric Video Summarization, in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, (2012) 1346-1353. http://dx.doi.org/10.1109/CVPR.2012.6247820

[15] Sun, M., Farhadi, A., & Seitz, S., Ranking Domain-specific Highlights by Analyzing Edited Videos., in Proceedings of European Conference on

Computer Vision, (2014) 787-802. https://doi.org/10.1007/978-3-319-10590-1_51

[16] Hearst, M.A., Dumais, S.T., Osuna, E., Platt, J., & Scholkopf, B., Support Vector Machines. IEEE Intelligent Systems and their Applications, (1998) 13 18–28. https://doi.org/10.1109/5254.708428

[17] Hochreiter, S., & Schmidhuber, J., Long Short-Term Memory. Neural Computation, (1997) 9 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

[18] Jiao, Y., Li, Z., Huang, S., Yang, X., Liu, B., & Zhang, T., Three-Dimensional Attention-Based Deep Ranking Model for Video Highlight Detection. IEEE Transactions on Multimedia, (2018) 20(10) 2693-2705. https://doi.org/10.1109/TMM.2018.2815998

[19] Wei, Z., Wang, B., Hoai, M., Zhang, J., Shen, X., Lin, Z., Měch, R., & Samaras, D., Sequence-to-Segments Networks for Detecting Segments in Videos, IEEE Transactions on Pattern Analysis and Machine Intelligence, (2021) 43(3) 1009-1021. https://doi.org/10.1109/TPAMI.2019.2940225

[20] Wu, J., Zhong, S-h., & Liu, Y., Dynamic Graph Convolutional Network for Multi-video Summarization, Pattern Recognition, (2020) 107 107382. https://doi.org/10.1016/j.patcog.2020.107382

[21] Kipf, T.N., & Welling, M., Semi-Supervised Classification with Graph Convolutional Networks, in Proceedings of the 5th International Conference on Learning Representations (ICLR), (2017), 1-14.

[22] Ren, S., He, K., Girshick, R., & Sun, J., Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, IEEE Transactions on Pattern Analysis and Machine Intelligence, (2017) 39 1137-1149. https://doi.org/10.1109/TPAMI.2016.2577031

[23] He, K., Gkioxari, G., Dollár, P., & Girshick, R., Mask R-CNN, in Proceedings of the IEEE International Conference on Computer Vision, (2017) 2961-2969. https://doi.org/10.1109/ICCV.2017.322

[24] Rochan, M., Reddy, M. K. K., Ye, L., & Wang, Y., Adaptive Video Highlight Detection by Learning from User History, in Proceedings of European Conference on Computer Vision, (2020) 261-278. http://dx.doi.org/10.1007/978-3-030-58589-1_16

[25] Li, P., Ye, Q., Zhang, L., Yuan, L., Xu, X., & Shao, L., Exploring Global Diverse Attention via Pairwise Temporal Relation for Video Summarization, Pattern Recognition, (2021) 111 107677. https://doi.org/10.1016/j.patcog.2020.107677

[26] Zhu, W., Lu, J., Han Y., & Zhou, J., Learning Multiscale Hierarchical Attention for Video Summarization. Pattern Recognition, (2022) 122 108312. https://doi.org/10.1016/j.patcog.2021.108312

[27] Yang, H., Wang, B., Lin, S., Wipf, D., Guo, M., & Guo, B., Unsupervised Extraction of Video Highlights via Robust Recurrent Auto-encoders, in Proceedings of the IEEE International Conference on Computer Vision, (2015) 4633-4641. http://dx.doi.org/10.1109/ICCV.2015.526

[28] Ringer, C., & Nicolaou, M.A., Deep Unsupervised Multi-View Detection of Video Game Stream Highlights, in Proceedings of the 13th International Conference on the Foundations of Digital Games, (2018) 1-6. https://doi.org/10.1145/3235765.3235781

[29] Lan, L., & Ye, C., Recurrent Generative Adversarial Networks for Unsupervised WCE Video Summarization, Knowledge-Based Systems, (2021) 222, 106971. https://doi.org/10.1016/j.knosys.2021.106971

[30] Rani, S., & Kumar, M., Social Media Video Summarization using Multi-Visual Features and Kohnen's Self Organizing Map. Information Processing & Management, (2020) 57(3) 102190. https://doi.org/10.1016/j.ipm.2019.102190

[31] He, K., Zhang, X., Ren, S., & Sun, J., Deep Residual Learning for Image Recognition, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (2016) 770-778. https://doi.org/10.1109/CVPR.2016.90

[32] Wang, Z., Zhou, J., Ma, J., Li, J., Ai, J., & Yang, Y., Discovering Attractive Segments in the User-generated Video Streams. Information Processing & Management, (2020) 57(1) 102130. https://doi.org/10.1016/j.ipm.2019.102130

[33] Cagliero, L., Garza, P., & Baralis, E., ELSA: A Multilingual Document Summarization Algorithm based on Frequent Itemsets and Latent Semantic Analysis, ACM Transactions on Information Systems, (2019) 37(2), Article No.: 21. https://doi.org/10.1145/3298987

[34] Srivastava, R., Singh, P., Rana, K.P.S., & Kumar, V., A Topic Modeled Unsupervised Approach to Single Document Extractive Text Summarization, Knowledge-Based Systems, (2022) 246, 108636. https://doi.org/10.1016/j.knosys.2022.108636

[35] Mikolov, T., Chen, K., Corrado, G., & Dean, J., Efficient Estimation of Word Representations in Vector Space, (2013) available at http://arxiv.org/abs/1301.3781

[36] Mohd, M., Jan, R., Shah, M., Text Document Summarization using Word Embedding, Expert Systems with Applications, (2020) 143, 112958. https://doi.org/10.1016/j.eswa.2019.112958

[37] Mutlu, B., Sezer, E.A., Akcayol, M.A., Candidate Sentence Selection for Extractive Text Summarization, Information Processing & Management, (2020) 57(6), 102359. https://doi.org/10.1016/j.ipm.2020.102359

[38] Jing, B., You, Z., Yang, T., Fan W., & Tong H., Multiplex Graph Neural Network for Extractive Text Summarization, in Proceedings of the Conference on Empirical Methods in Natural Language Processing, (2021)133–139. https://doi.org/10.18653/v1/2021.emnlp-main.11

[39] Hu, Y.-H., Chen, Y.-L., Chou, H.-L., Opinion Mining from Online Hotel Reviews – A Text Summarization Approach, Information Processing & Management, (2017) 53, 436–449. https://doi.org/10.1016/j.ipm.2016.12.002

[40] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K., BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, in Proceedings of the 2019 Conference of the North American Chapter of the

Association for Computational Linguistics, (2019) 4171-4186. http://dx.doi.org/10.18653/v1/N19-1423

[41] Duprez, C., Christophe, V., Rimé, B., Congard, A., & Antoine, P., Motives for the Social Sharing of an Emotional Experience. Journal of Social and Personal Relationships, (2015) 32(6) 757-787. https://doi.org/10.1177%2F0265407514548393

[42] Mao, X., Huang, S., Shen, L., Li, R., & Yang, H., Single Document Summarization using the Information from Documents with the Same Topic, Knowledge-Based Systems, (2021) 228 107265. https://doi.org/10.1016/j.knosys.2021.107265

[43] Erkan, G., & Radev, D.R., LexRank: Graph-based Lexical Centrality as Salience in Text Summarization, Journal of Artificial Intelligence Research, (2004) 22 457-479. https://doi.org/10.1613/jair.1523

[44] Ozsoy, M.G., Alpaslan, F.N., & Cicekli, I., Text Summarization using Latent Semantic Analysis, Journal of Information Science, (2011) 37(4) 405-417. https://doi.org/10.1177/0165551511408848

[45] Shen, D., Sun, J.-T., Li, H., Yang, Q., & Chen, Z., Document Summarization using Conditional Random Fields, in Proceedings of the 20th International Joint Conference on Artificial intelligence (IJCAI), (2007) 2862-2867.

[46] Nasar, Z., Jaffry, S. W., & Malik, M. K., Textual Keyword Extraction and Summarization: State-of-the-Art. Information Processing & Management, (2019) 56(6) 102088. https://doi.org/10.1016/j.ipm.2019.102088

[47] Bahdanau, D., Cho, K., & Bengio, Y., Neural Machine Translation by Jointly Learning to Align and Translate, in 3rd International Conference on Learning Representations (ICLR), (2015). https://doi.org/10.48550/arXiv.1409.0473

[48] Zhang, Z., & Sabuncu, M. R., Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels, in Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS), (2018) 8792–8802.

[49] Gygli, M., Grabner, H., Riemenschneider, H., & Van Gool, L., Creating Summaries from User Videos, in Proceedings of the European Conference on Computer Vision, (2014) 505-520. https://doi.org/10.1007/978-3-319-10584-0_33

[50] Song, Y., Vallmitjana, J., Stent, A., & Jaimes, A., TVSum: Summarizing Web Videos using Titles, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (2015) 5179-5187. https://doi.org/10.1109/CVPR.2015.7299154

[51] Zeng, K. H., Chen, T. H., Niebles, J. C., & Sun, M., Title Generation for User Generated Videos, in Proceedings of the European Conference on Computer Vision, (2016) 609-625. http://dx.doi.org/10.1007/978-3-319-46475-6_38

[52] Potapov, D., Douze, M., Harchaoui, Z., & Schmid, C., Category-Specific Video Summarization, in Proceedings of the European Conference on Computer Vision, (2014) 540-555. https://doi.org/10.1007/978-3-319-10599-4_35

[53] Yeung, S., Russakovsky, O., Jin, N., Andriluka, M., Mori G., & Fei-Fei, L., Every Moment Counts: Dense Detailed Labeling of Actions in Complex Videos, International Journal of Computer Vision, (2018) 126 375–389. https://doi.org/10.1007/s11263-017-1013-y

[54] Corona, K., Osterdahl, K., Collins, R., & Hoogs, A., MEVA: A Large-Scale Multiview, Multimodal Video Dataset for Activity Detection, in Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV), (2021) 1059-1067. https://doi.org/10.1109/WACV48630.2021.00110

[55] Manning, C.D., Raghavan, P., & Schutze, H., Introduction to Information Retrieval, (2008), Cambridge Univ. Press.

[56] Loshchilov, I., & Hutter, F., Decoupled Weight Decay Regularization. in 7th International Conference on Learning Representations (ICLR), (2019). https://doi.org/10.48550/arXiv.1711.05101

[57] Chen, C. C. & Chen, M. C., TSCAN: A Content Anatomy Approach to Temporal Topic Summarization, IEEE Transactions on Knowledge and Data Engineering, (2012) 24, 170-183. https://doi.org/10.1109/TKDE.2010.228

[58] Nenkova, A., Automatic Text Summarization of Newswire: Lessons Learned from the Document Understanding Conference, in Proceedings of the 20th National Conference on Artificial Intelligence (AAAI), (2005) 1436-1441. https://doi.org/10.7916/D83B67K6

[59] Solomon, M. R., Marshall, G. W., & Stuart, E. W., Marketing: Real People, Real Choices. (2020) 10th Edition, Upper Saddle River (N.J.): Pearson/Prentice Hall.

**Declaration of interests**

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: