

國立臺灣大學管理學院資訊管理系

碩士論文

Department of Information Management

College of Management

National Taiwan University

Master Thesis



以雙流注意力機制模型擷取直播影片精華

Two-Stream Attention Model for Highlight Extraction

羅良瑋

Liang-Wei Lo

指導教授：陳建錦 博士

Advisor: Chien-Chin Chen, Ph.D.

中華民國 110 年 7 月

July 2021



國立臺灣大學碩士學位論文  
口試委員會審定書

以雙流注意力機制模型擷取直播影片精華

Two-Stream Attention Model for Highlight  
Extraction

本論文係羅良瑋君（學號 R08725024）在國立臺灣大學資訊管理學系、所完成之碩士學位論文，於民國 110 年 7 月 5 日承下列考試委員審查通過及口試及格，特此證明

口試委員：

陳建錦

張詠淳

陳孟彰

所 長：

陳建錦


## ii. 誌謝

首先，我想感謝親切的建錦老師，在研究的路途上給予我們許多幫助和教導，也不時關心我們生活的概況，讓我們在碩士這兩年的學習歷程中收穫良多，最後在老師完善的指導下也順利了完成碩士的論文，很感謝老師這兩年來辛苦的付出！

此外，我也很謝謝同一屆的三位同學柏霖、芝妘、思仔，不管在課堂上還是論文研究上你們都讓我學習許多，也總是受到許多幫助，像是論文進度卡關時的想法交流，又或是準備面試一起努力刷題，都讓我收益良多，很慶幸碩士班研究所期間有三位厲害的同學相伴，讓我的碩士班旅途更加精采豐富。

最後我想謝謝我的母親、父親還有姊姊，碩士班的兩年期間給我許多支持，在不如意時也總是給予我許多鼓勵，也因為有你們一路上的陪伴，我才能夠順利的完成碩士班，謝謝你們一直以來的付出！

### iii. 中文摘要



近年來隨著談話型的串流影片越來越普及，直播平台漸漸的成為人們吸收新資訊的另一個管道。然而，談話型的直播影片通常較為冗長，使得大部分的觀眾無法全程參與直播，為了吸引觀眾加入直播串流影片甚至進一步成為訂閱者，提供精華片段對直播主和直播平台而言就變得格外重要。近年來有許多影片精華擷取相關的研究，其中多數研究使用影像上的資訊作為特徵再進一步擷取影片精華片段，然而這樣的方式並不適用於談話型的直播影片，原因在於談話型直播影片的精華與影像畫面並沒有直接相關，而是與直播主的言談以及觀眾的反應有關。在此篇論文中，我們使用了直播主的言談以及觀眾的留言作為模型輸入，提出了針對談話型直播影片精華擷取的模型，並進一步利用了位置的特徵增強和專注力機制強化特徵向量。此外，我們也透過自調節權重網路給予兩個文字分流預測分數不同的權重增強模型的表現。實驗證明我們的方法在現實生活的資料集上，表現比起近年提出的幾個知名的精華擷取模型來得更好。

#### **iv. Abstract**

As more and more conversation-oriented streaming videos are available, streaming platforms have gradually taken the place of traditional media for people to access information. Nevertheless, conversation-oriented streaming videos are often lengthy, which makes people reluctant to attend to the whole video. Highlight extraction has thus become necessary for streamers and platform providers to attract people and to watch their videos to become subscribers. Previous highlight extraction methods analyzed visual features of videos and were unable to deal with conversation-oriented streaming videos whose highlights are related to streamer discourses and viewer responses. In this research, we investigate highlight extraction on conversation-oriented streaming videos. Instead of evaluating visual features, the proposed highlight extraction method simultaneously examines textual streams of streamer discourses and viewer messages to conduct highlight extraction. The two techniques of position enrichment and message attention are developed to distill meaningful embeddings of the two textual streams. Also, a self-adaptive weighting scheme is deployed to effectively leverage the embeddings for highlight extraction. Experiments based on real world streaming data demonstrate that the two textual streams, self-adaptive weighting scheme, position enrichment, and message attention are useful to extract highlights of conversation-oriented streaming videos. Moreover, the extraction results are superior to those derived by well-known deep learning-based highlight extraction methods.

# Table of Contents



口試委員會審定書.....	i
誌謝.....	ii
中文摘要.....	iii
Abstract.....	iv
1 Introduction.....	1
2 Related work.....	5
2.1 Supervised Highlight Extraction.....	6
2.2 Unsupervised Highlight Extraction.....	7
2.3 Video Highlight Extraction using Textual Information.....	9
3 Methodology.....	12
3.1 Video Preprocessing and Discourse Segmentation .....	13
3.2 Streamer Discourse Embedding and Position Enrichment.....	15
3.3 Viewer Message Embedding and Attention.....	16
3.4 Highlight Extraction and Self-Adaptive Weighting Scheme.....	19
3.5 Model Training and Highlight Extraction Loss.....	20
4 Experiment.....	22
4.1 Evaluation Dataset and Metrics .....	22
4.2 Effect of System Components.....	25
4.3 Comparison with Other Highlight Extraction Methods.....	29
5 Conclusion.....	33
Reference.....	35



## 1. Introduction

The popularity of the streaming industry has in recent years skyrocketed due to the swift progress of Internet technologies. With more and more streaming platforms like Twitch and YouTube springing up, watching streaming videos is not just a trend but has become a daily activity for the younger generation. The COVID-19 pandemic has further intensified the surge of online streaming. Lockdowns and social distancing measures have transformed the people's lifestyles and increased the demand for stay-at-home entertainment in general and streaming in particular. According to Grand View Research surveys<sup>1</sup>, 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 for benefiting both streamers and platform providers. For platform providers, their revenues are normally based on paid advertising such that the more viewers a platform has, the more profit the platform 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 eye of audience has become a practical issue.

Live streaming consists of visual and audio contents, and emphasizes interactions between streamers and viewers. To attract viewers, streamers are encouraged to design vivid and engaging content. Streaming content can further be archived as videos for public access to help the channel later earn subscriptions. However, live streaming recorded videos are often so lengthy that viewers are reluctant to watch them. In order to increase channel exposure and attract new subscribers and viewers,

---

\* Corresponding author

<sup>1</sup> <https://www.grandviewresearch.com/industry-analysis/video-streaming-market>

many streamers have started to provide streaming highlights by using video editing tools or cooperating with third-party studios. Twitch officials, for instance, offer streamers streaming markers for highlighting content. 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 save human labor, there is an urgent need for effective highlight extraction methods that automatically compose highlights of streaming videos.

Video highlight extraction is an active multimedia research topic. Essentially, highlights are the most attractive video sections that are short enough to capture the gist of a video and its key points (Han et al., 2019; Xiong et al., 2019; Zhang et al., 2020). Due to the advance of artificial intelligence and the availability of image pre-trained models, many recent deep learning approaches (Yao et al., 2016; Fu et al., 2017; Han et al., 2019; Xiong et al., 2019) have been developed to extract highlights from videos. The approaches normally divide a video into sequential segments from which representative features such as image patterns and audio characteristics are derived to discover highlights. For instance, Yao et al. (2016) employed the well-know AlexNet pre-trained model (Krizhevsky et al., 2012) and a 3D deep convolutional neural network (Tran et al., 2015) to extract spatial and temporal image features from fixed-size video segments. The features were evaluated by a fully connected network to predict a score indicating the highlight probability of a given segment. While these studies are effective to detect video highlights, most of them focus on gaming or out-door videos. This is because the highlights of these action-oriented videos generally involve visual or audio effects whose features can be successfully distilled by deep neural networks to enhance the highlight extraction procedure. It is worth noting that a great portion of live streaming videos is now conversation-oriented in that streamers literally share personal experience or introduce the recent buzz or trending memes. In fact, the top watched streaming category on Twitch in 2020 was Just Chatting, which collected about 1,600k conversation-oriented streamer channels and received more than 19



billion watching hours<sup>2</sup>. 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 might not be able to identify the representative features of highlight sections, and therefore fail to distinguish highlights from this type of streaming video. In our work, we proposed our novel highlight extraction model to address following questions:

*RQ1: How to effectively extract highlight from conversation-oriented streaming videos?*

*RQ2: How to improve model performance by utilizing streaming video properties based on textual information?*

In this paper, we study the highlight extraction of conversation-oriented streaming videos. To the best of our knowledge, this is the first work that explores properties of conversation-oriented streaming videos for effective highlight extractions. Instead of investigating visual and audio effects, we examine streamer conversation and viewer feedback. The proposed method considers two textual streams for automatic highlight extraction. One is the streamer conversation, and the other is the viewer messages posted in the chat rooms that reflect viewer feedback in streamer-viewer interactions. The 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 to a streaming video can be 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. We noticed that viewer messages are sometimes distracted and not every message is informative. Hence, an attention mechanism is 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

---

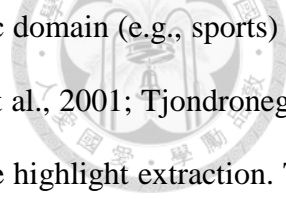
<sup>2</sup> [https://sullygnome.com/game/Just\\_Chatting](https://sullygnome.com/game/Just_Chatting)

information are fed into multilayer perceptrons to predict a highlight score. Segments with a high score are selected to construct the video highlight. This paper makes following contribution:

1. To the best of our knowledge, this is the first study that extracting highlight from streaming videos by simultaneously using streamer discourses and audience messages.
2. By further taking natural properties of streaming video into account, we further improve model performance by using position enrichment and attention mechanism.
3. Our method outperforms many well-known highlight extraction methods based on real world streaming videos.

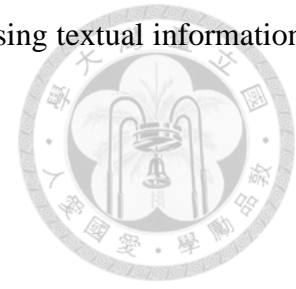
The remainder of this paper is organized as follows. Section 3 provides a review of related works. In Section 4, we detail the proposed highlight extraction method, and then in Section 5 we evaluate the system's performance. Section 6 summarizes our conclusions.

## 2. Related Work



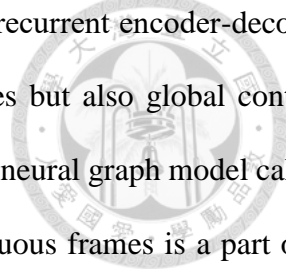
In the past, methods of video highlight extraction would focus on a specific domain (e.g., sports) and relied on extraction rules or heuristics defined by domain experts (Nepal et al., 2001; Tjondronegoro et al., 2004). For instance, Nepal et al. (2001) investigated basketball game highlight extraction. The authors compiled a set of extraction rules that simultaneously scan image and acoustic patterns such as the appearance of scoreboards or the loudness of crowd cheer 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 (Rui et al., 2000; Otsuka et al., 2005; Zhang et al., 2006; Lee et al., 2012; Sun et al., 2014;). In (Rui et al., 2000), 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 (e.g., support vector machines (Hearst, 1998)) to classify excited commentator speeches which are then probabilistically fused with detected ball-hitting sounds to calculate the highlight probability of a candidate clip. Lee et al. (2012) 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 whom the camera wearer interacted with. Labeled highlights were provided to train a regression function that estimates 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 convolution networks and long-short term memory (Hochreiter & Schmidhuber, 1997) architectures. These improvements have enhanced image feature engineering and therefore have strengthened highlight extraction results. Below, we categorize recent deep learning

studies as supervised and unsupervised, and then review research works using textual information of live streaming videos for highlight extraction.



### *2.1 Supervised Highlight Extraction*

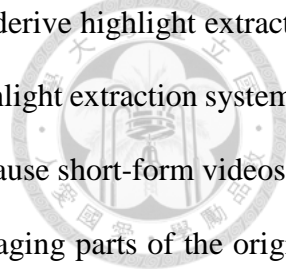
Yao et al. (2016) employed techniques of pairwise learning to detect highlights of first-person videos and developed a ranking-based highlight extraction method named TS-DCNN. Differing from traditional methods that analyze individual video segments, the method learns a highlight scoring function according 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 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. (2018) further improve the model by utilizing mechanism on both temporal and spatial stream. In temporal stream, the attention mechanism let model focus on frames that are worth watching. Similar in the spatial stream, the attention mechanism guides the highlight extraction model to generate useful feature from specific regions from a frame and neglects some unimportant regions. Although video contexts can be meaningful as a way to identify important video segments, most highlight extraction methods neglect context information and evaluate video segments independently. Wei et al. (2018) addressed this segment-independent problem by means of a sequence-to-sequence highlight extraction model. The designed encoder sequentially receives the feature vectors of video frames and generates 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



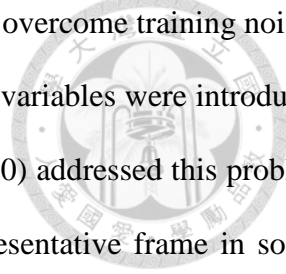
the highlight segment, and the segment's highlight confidence score. The recurrent encoder-decoder mechanism enables the method to capture not only local segment features but also global context information. In (Zhang et al., 2020), the authors developed an object-aware neural graph model called VH-GNN that detects video objects to determine whether a clip of continuous frames is a part of a video highlight. This method first applies two pre-trained models, namely Region Proposal Network (RPN) (Ren et al., 2016) and RoIAlign (He et al., 2017), 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. Rather than only seize important feature from continuous frame, Rochan et al., (2020) address the highlight extraction problem with side information – user history. The proposed model consists of two sub-networks, a highlight detection network and a history encoder network. The highlight detection network is utilized to give each segment a score to show how possible it should be categorized as highlight. The history encoder network is responsible for generate user preference style based on user history information. In the testing phase, the segment will be first fed into highlight detection model and produce representative features from segment visual pattern. These representative features are following interacted with the user history information in temporal-adaptive instance normalization layer. In temporal-adaptive normalization layer, they will alter the previous encoding frame features based on user history and thus finally generate user-specific video highlight.

## 2.2 Unsupervised 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,



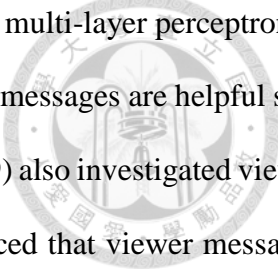
the unsupervised approach makes use of logical assumptions to implicitly derive highlight extraction models. Yang et al. (2015) constructed a domain-specific (e.g., surfing) highlight extraction system by assuming that videos less than four minutes long are highlights. This is because short-form videos 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 are submitted to crawl short-form videos from the web. The videos will be segmented into snippets which are 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 are regarded as highlights because their features are consistent with those of the short-form videos. Note that the above assumption brings noises (i.e., non-highlight short-form videos) into the model training. To lessen the impact of noisy data, the authors enhanced the autoencoder with the shrinking exponential loss. Ringer and Nicolaou (2018) also employed autoencoders to identify video highlights in an unsupervised manner. In contrast to the last method, the authors treated video frames with a high reconstruction loss as highlights. This is because the authors focused on video game streaming whose game videos are normally lengthy and share similar backgrounds. The authors thus assumed highlights are anomalies in videos and are associated with a 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 an anomaly and are thus categorized as highlights. Contrary to the above methods that identify highlights by means of frame or segment losses, Xiong et al. (2019) developed a model that explicitly predicts a highlight score for a video segment. To save 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 noises, e.g., a long-form video might be classified as a highlight, a group of latent variables were introduced to induce whether the pairwise score ranking is valid. Rani & Kumar (2020) addressed this problem with an unsupervised model framework which focused on selecting representative frame in social media videos. First, they utilized different measurement for visual feature such as correlation between two frame and a colour histogram difference measurement to generate overall feature fusion score for each frame. Frames with higher cumulative feature value means that there will be a significant difference between current frame and previous frame. Also, a defined threshold was utilized to validate whether this frame was useful. Next, they clustered these useful candidates' key frames by using Kohonen Self Organization Map. In clustering phase, they first extract HSV histogram as input for Kohonen SOM model for each frame. The Kohonen SOM model will learn the distribution of input key frames and output clusters containing similar key frames. Finally, the author calculated the Euclidean distance between each pair of frames in the same cluster and select the pair with maximum Euclidean distance to be most representative frames in each cluster.

### *2.3 Video 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. (2017) 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 (He et al., 2016) was employed to the segment frames to extract important visual features. At the same time, the viewer messages posted within the segment were 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 embeddings 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. (2019) 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 builds a language model which produces word embeddings for messages. Also, a bidirectional Gated-Recurrent Unit (biGRU) architecture that sequentially processes viewer messages is implemented to encode viewer messages with context information. In (Wang et al., 2020), they proposed a time-sync comments-based popularity prediction model to discover attractive segment in videos. The model was designed to solve two main task which are predicting segment popularity and audience emotion. The video pattern and audience messages are utilized as features to solve two main tasks. To extract useful pattern from video, they propose a long-short term memory-based encoder model to generate useful embedding to every segment. However, in the other hand, for a newly released video, there will not be any comments before the videos was published and thus cannot use message information as input. To address this problem, they utilized language transfer model to generate representative comment from video segment. While training the language transfer model, a video segment embedding will be paired with true comment and an uncorrelated comment and the model will update parameters based on the similarity of comment text embedding and video feature embedding. The language transfer model thus can generate correlated messages embedding given video segment after training. Later, they combined two output embeddings from the last layer of pretrained language transfer model and video encoder model as final embedding. This final embedding will be fed into fully connected layer and make classification or regression.

To sum up, video highlight extraction is an active and challenging research topic. Previous studies used to focus on action-oriented videos (e.g., sports and video games) and they rely on visual features



distilled through deep convolutional neural networks. While these studies demonstrate remarkable highlight extraction performance, they might not be effective for conversation-oriented streaming videos because visual features or patterns do not constitute the main video highlights. In this work, rather using convolutional networks and visual features, we propose a two-stream neural network architecture that examines streamer discourses and viewer messages. The two types of textual information can reflect the intentions of both streamers and viewers, and are therefore helpful for extracting meaningful highlights from conversation-oriented streaming videos.

### 3. Proposed System

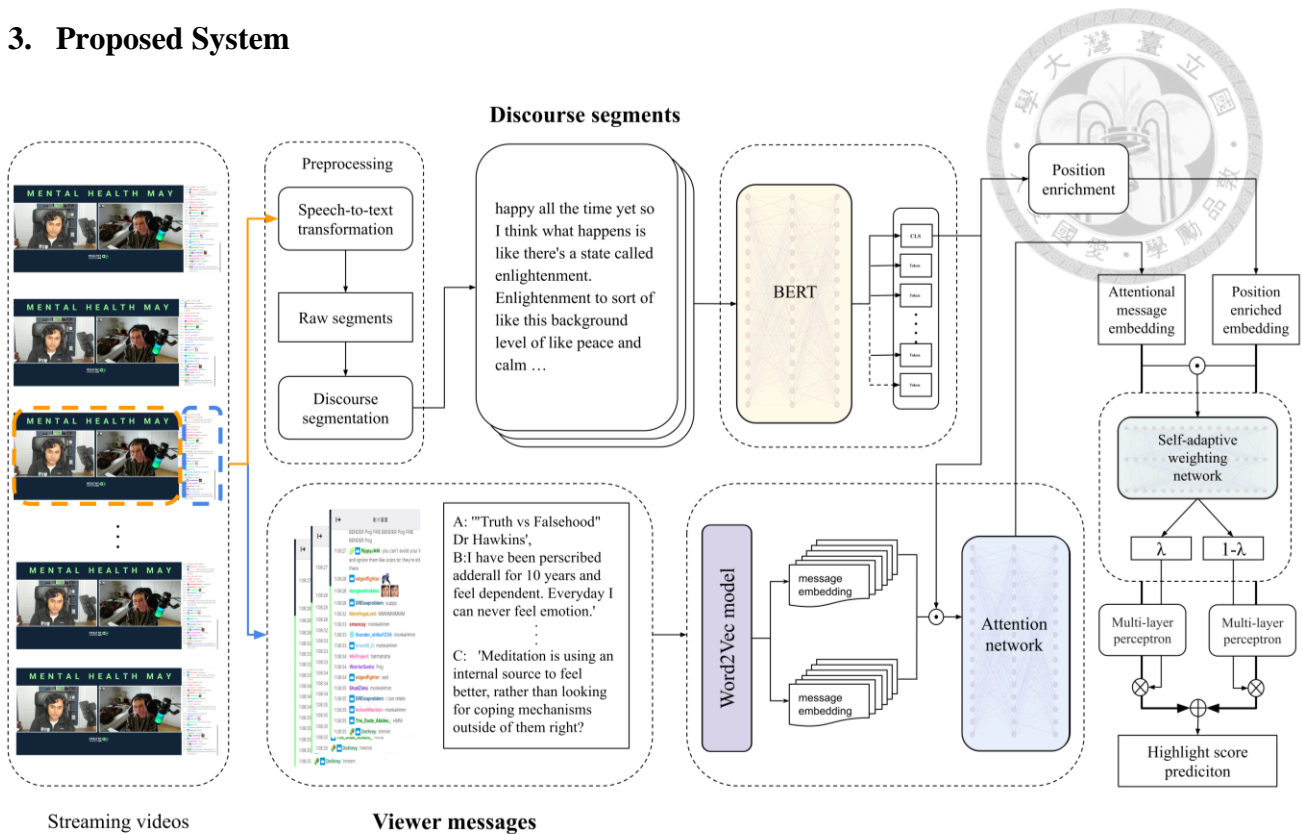
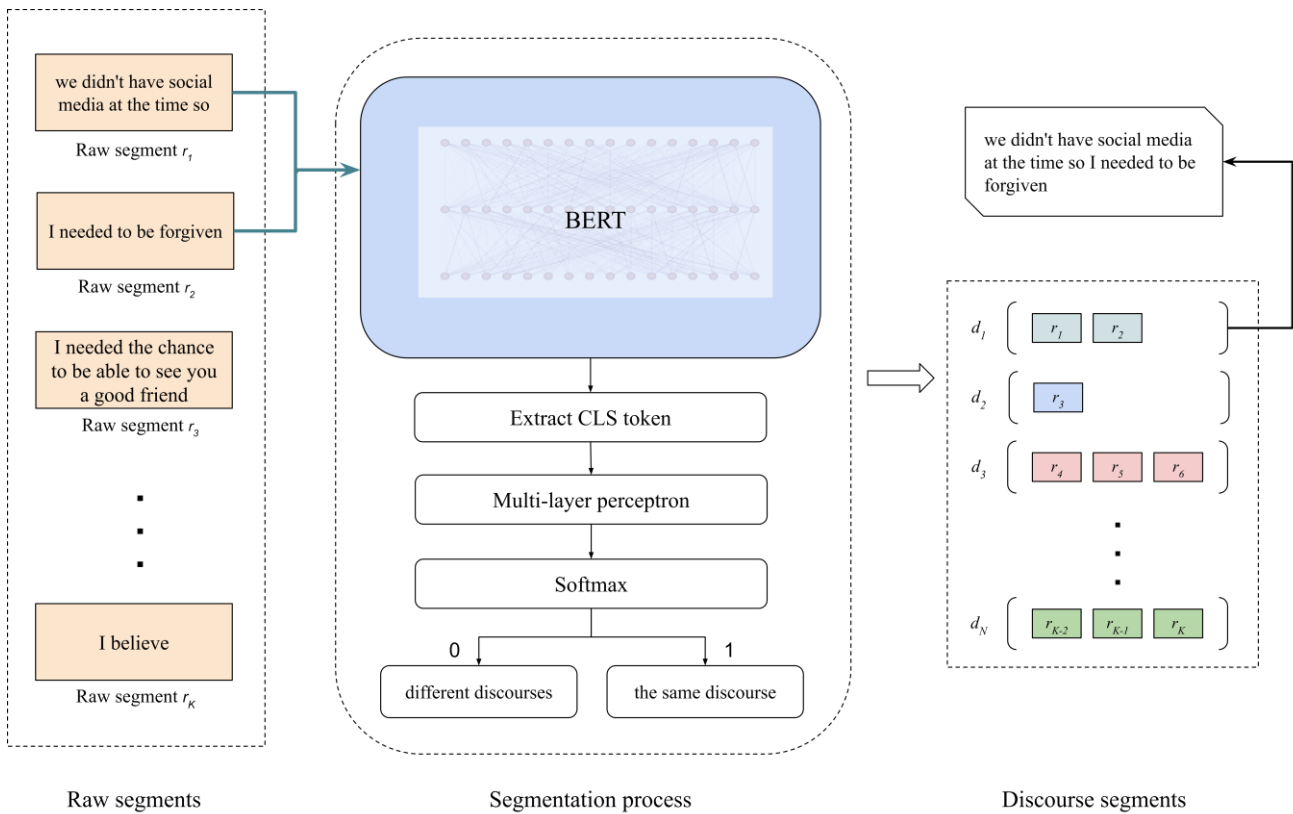


Fig. 1. The system architecture of our proposed model.

In this section, we present the 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 in accordance with the discourses of streamer conversation. Then, the state-of-the-art language model BERT (Devlin et al., 2019) is applied to the spoken sentences of the streamer within a discourse segment to derive the semantic embedding of the streamer discourse. In addition, 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 (Duprez et al., 2015), our proposed method 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

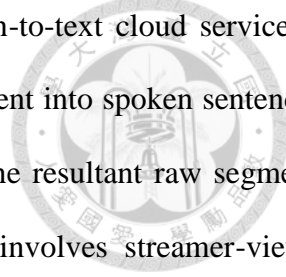
the intention of the messages as a message embedding. Lastly, the embeddings of the streamer discourse and the viewer messages are respectively fed into a multi-layer perceptron and are leveraged by a self-adaptive weighting scheme to predict a highlight score of the segment. Segments with a high score are selected to construct the video highlight. We detail the highlight extraction method in the following sections.

### 3.1 Video Preprocessing and Discourse Segmentation



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

As mentioned in the related work section, methods of highlight extraction normally divide a video into a sequence of segments. Since our highlight extraction is based on the textual information of streaming videos, instead of decomposing a video into image frames, 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*  $\{r_1, r_2, \dots, r_K\}$ . In other words, acoustic silence forms



the delimiter of two consecutive raw segments. Next, the Google speech-to-text cloud service<sup>3</sup> is employed to convert the audio conversation of the streamer in a raw segment into spoken sentences. We observed that the silence-based segmentation is so error-prone that the resultant raw segments hardly 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. Merging adjacent raw segments is closely related to the next sentence prediction task (Devlin et al., 2019) 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 is narratively following the other sentence. Because a discourse segment is constituted 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*  $\{d_1, d_2, \dots, d_N\}$  by merging the adjacent raw segments that belong to the same discourse.

---

<sup>3</sup> <https://cloud.google.com/speech-to-text>

### 3.2 Streamer Discourse Embedding and Position Enrichment

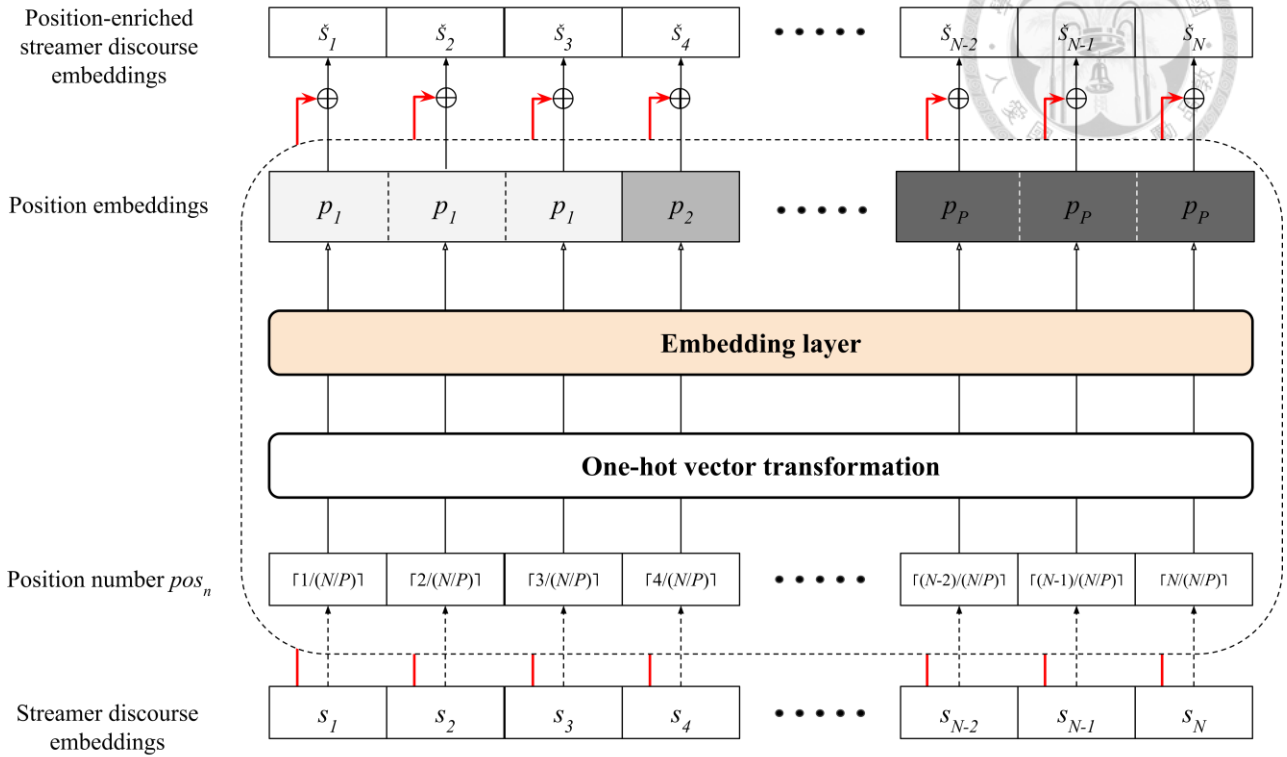


Fig. 3. The position enrichment mechanism.

Having created discourse segments, BERT again is applied to the spoken contents of the segments to obtain a series of *streamer discourse embeddings*  $\{s_1, s_2, \dots, s_N\}$ . Note that in the segmentation step, the input to BERT is a pair of adjacent raw segments because the segmentation task is for detecting 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 video highlights. 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 (Tas & Kiyani, 2007)

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 (Erkan & Radev, 2004) and latent topics (Ozsoy et al., 2011), have been explored. Notably, position-related features, such as the order of a sentence 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 (Shen et al., 2007; Tas & Kiyani, 2007; Yeh et al., 2005). 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:

$$pos_n = \lceil n / (N/P) \rceil, \quad (1)$$

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 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  $\check{s}_n = s_n + p_{pos_n}$ .

### 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 rooms reveal their responses regarding streamer discourses. As highlights are supposed to interest viewers, we consider viewer messages as an important source of highlight extraction. Like

streamer discourse embeddings, it might be assumed that we can simply 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 hence are meaningful; unfortunately, most pre-trained language models, e.g., BERT, are built against formal text like wiki pages or books. They therefore cannot recognize these casual tokens and fail to discover the intention of viewers. To resolve this difficulty, we derive embedding vectors of viewer messages by means of the skip-gram of Word2Vec (Mikolov et al., 2013), 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  $\{w_1, w_2, \dots, w_T\}$ , the skip-gram approach estimates the embedding vector of each unique token by maximizing the following sum of log probabilities:

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

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 at predicting the surrounding tokens (i.e.,  $w_{t+h}$ 's) based on a given token  $w_t$ . The prediction probability is calculated as follows:

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

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  $msg_i = \{w_1, w_2, \dots, w_l\}$  which contains a series of tokens, we average the embedding vectors of the tokens to obtain the message embedding vector  $m_i$  as follows:

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

Regarding the diversity of viewer opinions, popular streaming always attracts a lot of viewers and thus attracts a huge number of viewer messages. For instance, in our experiment dataset, each evaluated streaming video contains around 17,000 messages. The message contents are so diverse that not every message is crucial to highlight extraction, so in order to effectively utilize viewer messages, we designed an attention mechanism that weights viewer messages in terms of streamer discourses and the highlight extraction task.

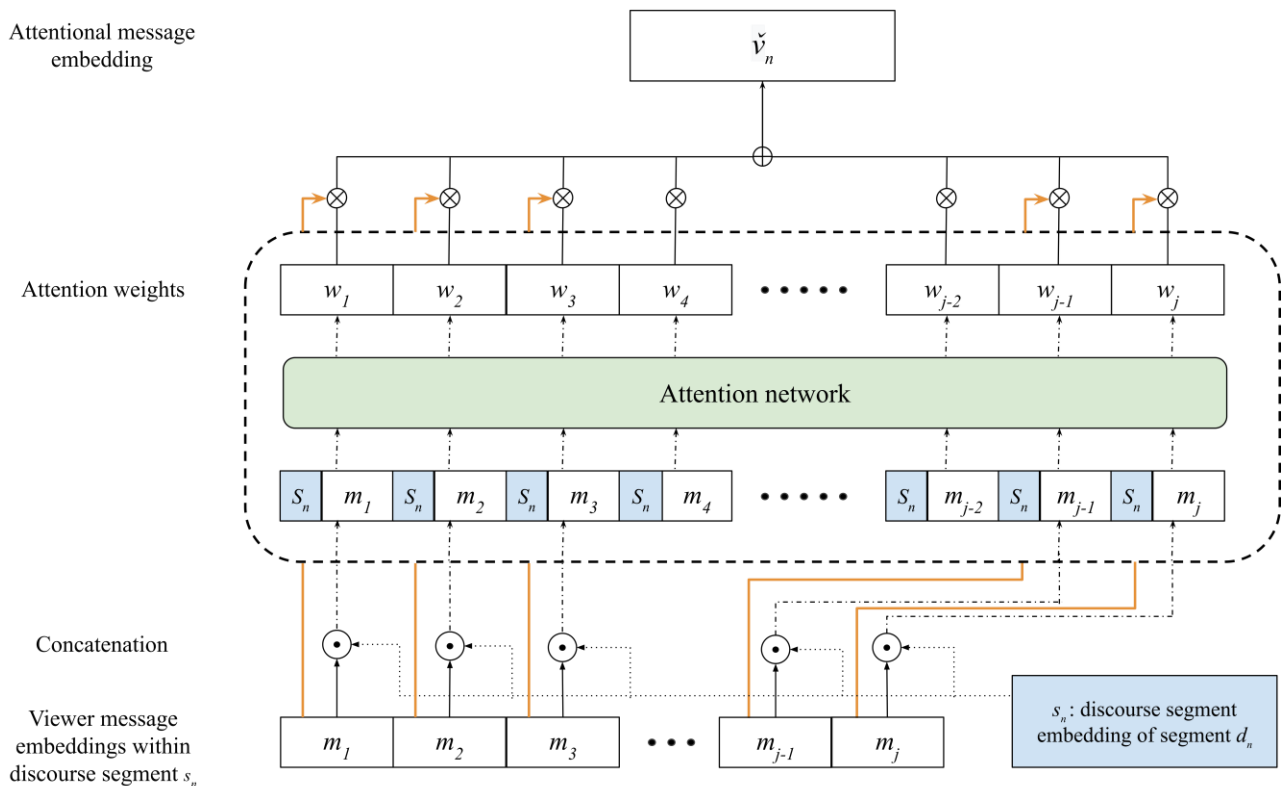


Fig. 4. The viewer message attention mechanism.

Bahdanau et al. (2015) exhibited the bottleneck problem of machine translation. Methods of machine translation are normally based on the encoder-decoder architecture (Bahdanau et al., 2015) 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, which leads to the bottleneck of machine translation. To solve this problem, the authors suggested using attention techniques to customize the weights of hidden states when emitting each output token. To tackle the huge amount of diverse viewer messages, our attention mechanism shown in Figure 4 captures the intention of the viewers in a discourse segment by means of the following equation:

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

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.

### 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 highlight. A baseline approach to integrate the two embeddings for the probability prediction is to respectively feed the embeddings into a multi-layer perceptron 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 develop a *self-adaptive weighting scheme* that customizes  $\lambda$  in accordance with the given  $\check{s}_n$  and  $\check{v}_n$ . As shown in Figure 1, the weighting scheme first concatenates the two embeddings and feeds the aggregated embeddings into a self-adaptive weighting layer that learns to leverage the two embeddings. By doing so, the scale  $\lambda$  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  $Q = [\langle d_1, y_1 \rangle, \langle d_2, y_2 \rangle, \dots, \langle d_L, y_L \rangle]$  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:

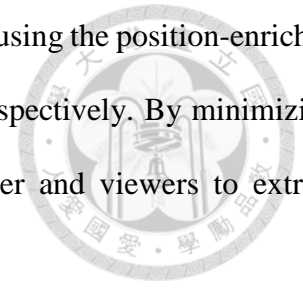
$$HE_{Loss}(Q) = \frac{1}{L} \sum_{l=1}^L (\lambda_l * S_{Loss}(\langle d_l, y_l \rangle) + (1 - \lambda_l) * M_{Loss}(\langle d_l, y_l \rangle)), \quad (6)$$

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 (Brink et al., 1996).

$$S_{Loss}(\langle d_l, y_l \rangle) = -[y_l \cdot \log \hat{y}_l^s + (1 - y_l) \cdot \log(1 - \hat{y}_l^s)] \quad (7)$$

$$M_{Loss}(\langle d_l, y_l \rangle) = -[y_l \cdot \log \hat{y}_l^m + (1 - y_l) \cdot \log(1 - \hat{y}_l^m)], \quad (8)$$

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 and the attentional message embedding, respectively. By minimizing  $HE_{Loss}$ , our model parameters are guided to distill intentions of 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 parameters on the highlight extraction performance. Finally, we compare the proposed method with well-known highlight extraction methods.

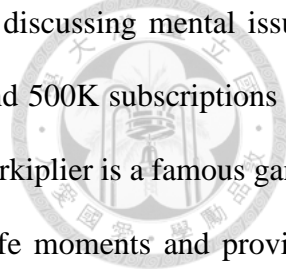
### 4.1 Evaluation Dataset and Metrics

**Table 1**

The statistics of the evaluated videos.

Number of testing videos	44
Length of testing videos (sec.)	379,546
Number of streamer discourse segments	8,269
Number of viewer messages	777,379
Number of message tokens	2,693,026
Length of the labeled highlights (sec.)	52,861

Video highlight extraction is an active research topic, and hence several studies have released video datasets with labeled highlights. However, as mentioned in the introduction section, most of the 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. 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 44 streaming videos categorized in TALK SHOWS & PODCASTS from two famous streamers for evaluations. Streamer @HealthyGamer\_GG is an addiction psychiatrist. His streaming



videos were designed to help gamers overcome their game addiction by discussing mental issues. According to the official statistics of Twitch<sup>4</sup>, this streamer now has around 500K subscriptions and has accumulated around 5M views of his channel. The other streamer @Markiplier is a famous gamer and influencer with more than 2M subscriptions. This streamer shares life moments and provides thought-provoking perspectives in his streaming and the total views of his channel is 12.8M. We crawled all the viewer messages of the testing videos from chat rooms to evaluate the effect of viewer intentions on highlight extraction. The total length of the evaluated videos is 379,546 seconds (105.4 hours) with each streaming video being about 2.4 hours long and having around 17,600 messages. To compile the ground truth of the videos, we invited five experts who are heavy streaming users and are familiar with Twitch. Before they labeled the highlights of the videos, an orientation was held to ensure the quality and usability of the labeled data. In short, the experts were asked to first watch the videos thoroughly. Then, they had to think about the video content for a while before labeling highlight sections. Also, in order to ensure that the labeled highlights were concise and solid, the highlights were restricted to no longer than 15% of the video length. The labeling was time-consuming, resulting in the highlight editing hours for each video taking around 3 times the video length. Detailed statistics of the evaluated videos are listed in Table 1.

We adopted the conventional 5-fold cross validation (Wong, 2015) to obtain performance results. Specifically, we evenly divided the streaming videos into 5 disjoint 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 in accordance with 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

---

<sup>4</sup> [https://twitchtracker.com/healthygamer\\_gg/statistics](https://twitchtracker.com/healthygamer_gg/statistics), the statistics of @HealthyGamer\_GG  
<https://twitchtracker.com/markiplier/statistics>, the statistics of @Markiplier

were compared with the ground truth to report the precision@ $K$ , recall@ $K$ , and F1@ $K$  defined as follows:

$$\text{precision@}K = \frac{|highlight_{predicted} \cap highlight_{ground\_truth}|}{|highlight_{predicted}|} \quad (9)$$

$$\text{recall@}K = \frac{|highlight_{predicted} \cap highlight_{ground\_truth}|}{|highlight_{ground\_truth}|} \quad (10)$$

$$\text{F1@}K = \frac{2 * \text{precision@}K * \text{recall@}K}{\text{precision@}K + \text{recall@}K}, \quad (11)$$

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 the frequency used to judge the superiority of prediction systems. Note that PyTorch<sup>5</sup>, a well-known deep learning library, was adopted to implement our highlight extraction method. The optimizer we used to learn network parameters was AdamW (Loshchilov & Hutter, 2019) 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 (Mikolov et al., 2013). 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.

---

<sup>5</sup> <https://pytorch.org/>

**Table 2**

The system hyper-parameter settings.

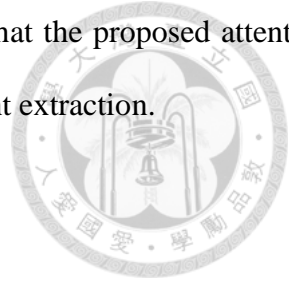
Learning optimizer	AdamW
Learning rate	1e-5
Batch size	8
Dropout rate	0.2
Dim. of streamer discourse embeddings	768
Dim. of position embeddings	768
Dim. of viewer message embeddings	300
$P$	5



#### 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. Then, we examine the proposed position and attention mechanisms. Finally, we compare the performance with and without using the self-adaptive weighting scheme. As shown in Table 3, the performance scores of our method are not very high. The  $\text{precision@10}$  is only around 0.2. This is because the evaluated streaming videos are very long and the expert-labeled highlights are only 14% of the evaluated videos. Identifying these highlight sections is thus very challenging. Nevertheless, our method still outperforms many well-known 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 on streamer discourses (i.e., the scores of Streamer Discourse Embeddings). This is because live streaming normally attracts a large number of viewers, and for this reason, there is such a large diversity in viewer messages that the extraction model based on viewer messages is likely to be distracted. It is worth noting that by using the attention mechanism, both the precision and recall scores improved with the

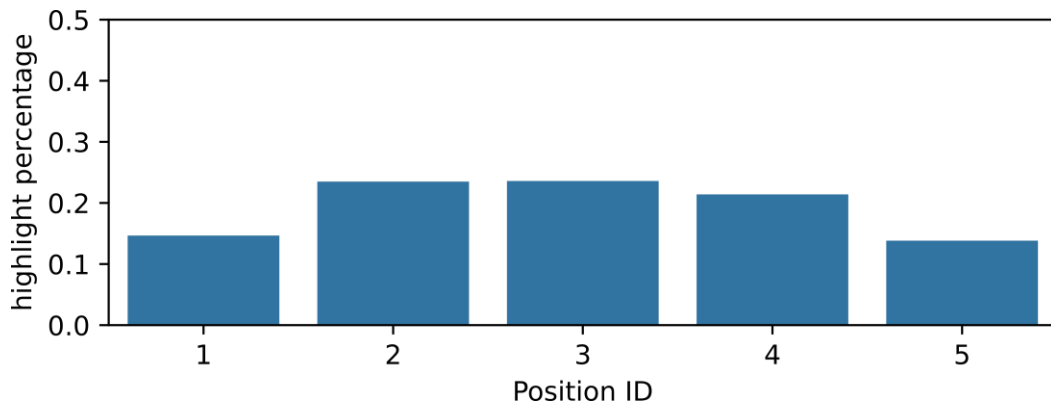
F1 score increasing from 0.1508 to 0.1685. This improvement indicates that the proposed attention mechanism successfully distills viewer messages and is helpful for highlight extraction.



**Table 3**

Highlight extraction performance of system components.

	precision@10	recall@10	F1@10
Streamer Discourse Embeddings	0.1888	0.1402	0.1609
Position-Enriched Streamer Discourse Embeddings	0.1996	0.1486	0.1704
Message Embeddings	0.1782	0.1308	0.1508
Attentional Message Embeddings	0.1985	0.1463	0.1685
<b>Our proposed method</b>	<b>0.2080</b>	<b>0.1537</b>	<b>0.1768</b>

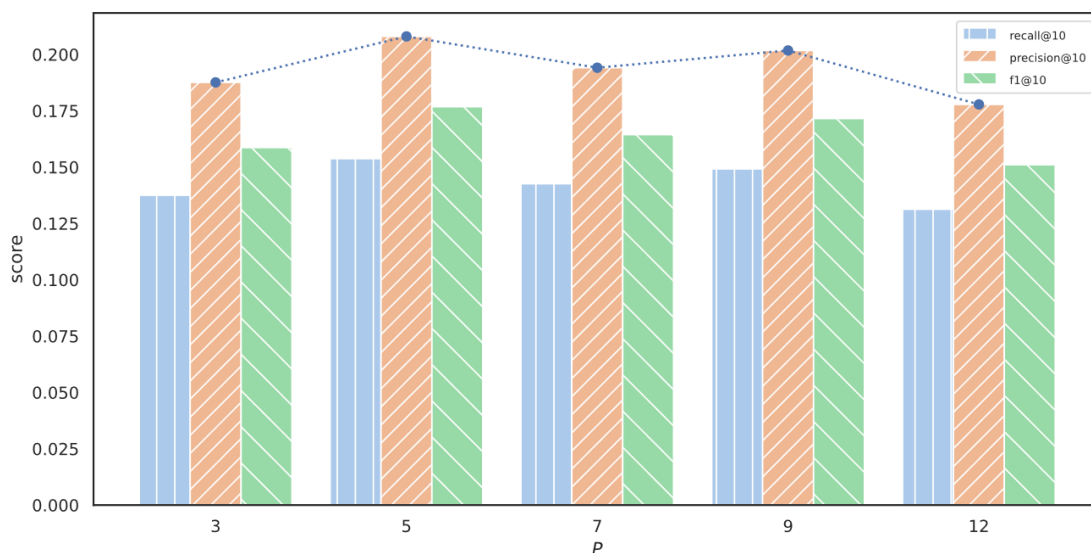
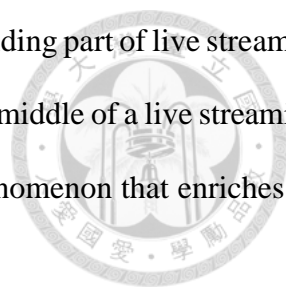


**Fig. 5.** The highlight distribution over different positions.

The results based on streamer discourses also improved when incorporating the streamer discourse embeddings with the position embeddings. Figure 5 shows the distribution of the expert-labeled highlights across positions. Differing from the phenomenon mentioned in Section 3.2 of the text summary frequently occurring at the beginning or ending of the summarized text, the labeled highlights often occur in the middle of a video. This is probably because viewers generally do not join live streaming 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 live streaming is not usually very climactic. As a result of this, highlights often occur in the middle of a live streaming. The designed position embeddings successfully capture this positional phenomenon that enriches the streamer discourse embedding in terms of highlight extraction.



**Fig. 6.** The effect of the parameter  $P$ .

As a further demonstration of position embedding, Figure 6 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 setting  $P$  at 5 produces a good highlight extraction performance, we used the 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. Considering that the two textual embeddings convey intentions of two important roles in information communication, i.e., information delivers and information receivers who both generally hold different perspectives regarding the discussed topic due to communication noise (Solomon et al., 2008), using the two embeddings together improves the highlight extraction results.

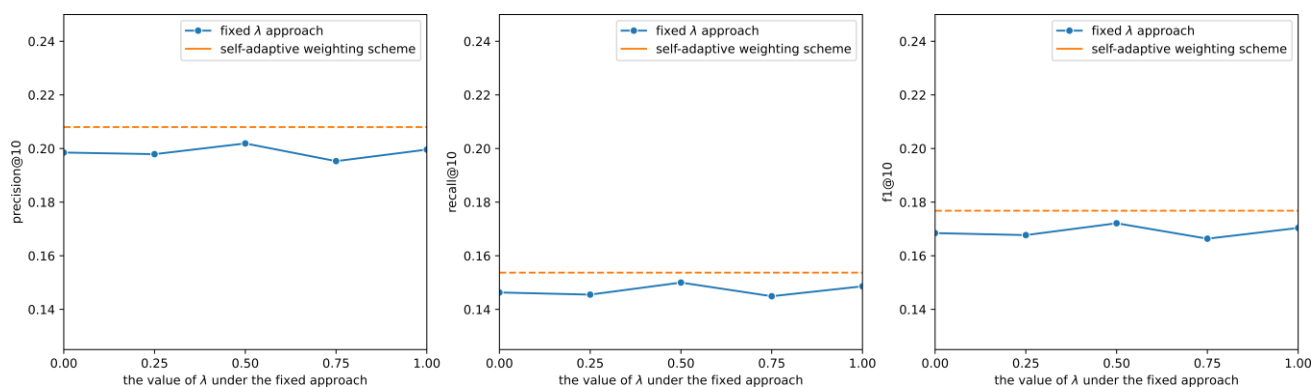


Fig. 7. Comparisons with and without the self-adaptive weighting scheme.

In addition to the two textual embeddings and parameter  $P$ , we also investigated the effect of our self-adaptive weighting scheme. Figure 7 compares the performances of our method with and without self-adaptive weighting. When 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 7, the performance scores of the fixed- $\lambda$  approach are fluctuating and there is no obvious tendency of  $\lambda$  in favor of highlight extractions. In other words, 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 exciting viewer feedback and interesting streamer discourse. As a fixed  $\lambda$  cannot customize individual weights of streamer discourses and viewer messages, the results are inferior. By contrast, the self-adaptive weighting is superior because it is capable of calibrating  $\lambda$  in accordance with the streamer and viewer embeddings.

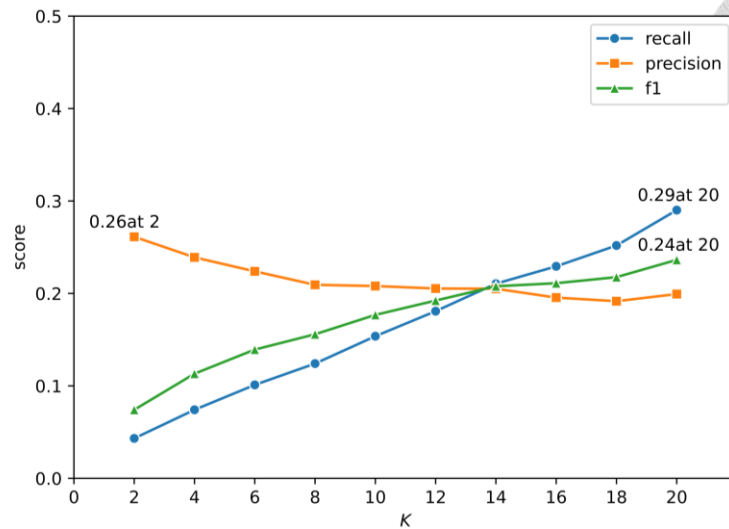


Fig. 8. The performances of the extracted highlights under different  $K$  percentage.

Finally, we examine 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 8, 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, as 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 0.26 precision at  $K = 2$ ) indicate that the top segments we estimated correspond well with the expert-labeled highlights, thus indicating that our method is promising in constructing short streaming highlights.

#### 4.3 Comparison with Other Highlight Extraction Methods

The above experiments thoroughly evaluated our system components. Next, we compare our proposed method with five well-known video highlight extraction methods introduced in the related work section, namely, V-CNN-LSTM (Fu et al., 2017), Joint-lv-LSTM (Fu et al., 2017), TS-DCNN (Yao et al., 2016), biGRU-DNN (Han et al., 2019), and VH-GNN (Zhang et al., 2020). We selected the methods for comparisons because these recent deep-learning based methods have demonstrated extraordinary performance on video highlight extraction. Among these methods, V-CNN-LSTM,

Joint-lv-LSTM, and biGRU-DNN are specific to live streaming videos, and the latter two further 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. By 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 reveals the benefit of textual information for streaming highlight extraction. To ensure fair comparisons, all the methods were implemented by using public packages and the hyper-parameters were set as suggested in the original papers. The same 5-fold cross validation was conducted to obtain their highlight extraction performances.

**Table 4**

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

	precision@10	recall@10	F1@10
V-CNN-LSTM	0.1319	0.0949	0.1104
Joint-lv-LSTM	0.1399	0.1006	0.1170
TS-DCNN	0.1204	0.0867	0.1008
biGRU-DNN	0.1635	0.1176	0.1368
VH-GNN	0.1530	0.1102	0.1281
Our proposed method	<b>0.2080</b>	<b>0.1537</b>	<b>0.1768</b>

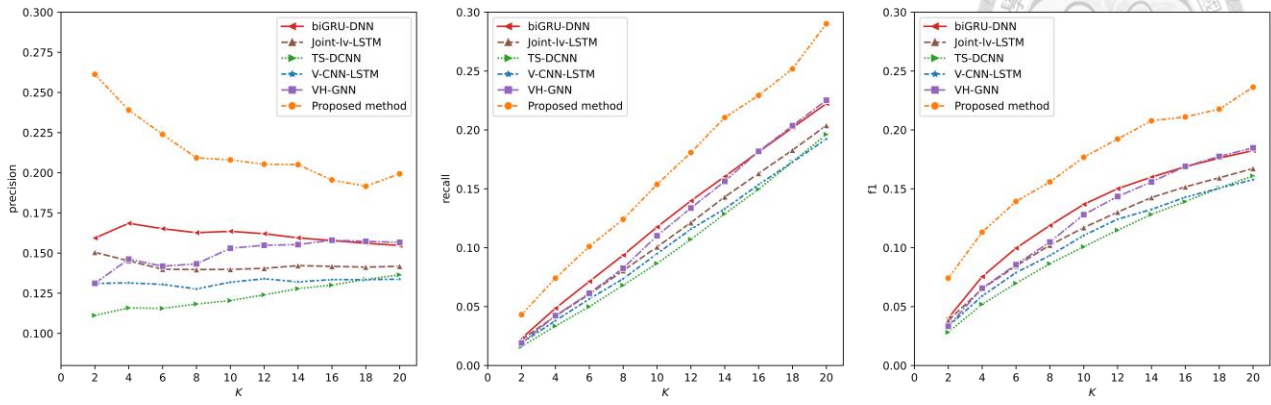


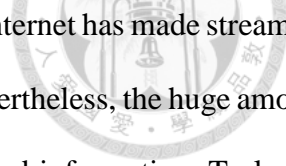
Fig. 9. The performances of the compared methods under different settings of  $K$ .

Table 4 lists the performance scores of the compared methods under  $K = 10$  and Figure 9 shows the performances under different settings of  $K$ . We expected the compared methods would produce comparable results owing to their superior performances reported in the respective papers. Contrary to our expectation, the compared methods are 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. When dealing with the evaluated streaming videos which are conversation-oriented, the methods are ineffective since visual effects are not indicative of the video highlights. V-CNN-LSTM, Joint-lv-LSTM, TD-DCNN, and VH-GNN are inferior because they rely 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.

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  $K$ , biGRU-DNN performs much better than the other compared methods do. This is because biGRU-DNN is also a textual-based method that 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 the approaches are based on view messages, the outperformance of Attentional Message Embeddings again demonstrates the advantage of our attention mechanism in distilling viewer messages relevant to highlight extraction. 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 are superior to Joint-lv-LSTM.

## 5. Conclusion



The abundance of conversation-oriented streaming videos available on the Internet has made streaming platforms a treasure-house of information for people of all walks of life. Nevertheless, the huge amount of streaming videos also overwhelm people when searching for their desired information. To lessen the information overload problem, and also to increase channel exposure and subscriptions, it is important to provide highlights for users. In this paper, we have developed our model which automatically extracts highlights from conversation-oriented streaming videos. The results of our literature review indicate that our research is the first study investigating conversation-oriented streaming video highlight extraction. Differing from previous highlight extraction methods which mostly focus on action-oriented videos and heavily rely on visual features, our proposed model simultaneously examines streamer discourses and viewer messages, and enhances the embeddings of the textual information by means of the designed position embeddings and the message attention mechanism. Experiments based on real world streaming data show that our model outperforms several state-of-the-art highlight extraction methods.


Our performance scores show that discovering highlights of conversation-oriented streaming videos is challenging and the results have great room for improvement. Since conversation-oriented streaming highlight extraction has not yet been well-addressed in the literature and since this type of videos is currently the most popular type, our study serves to encourage future research. In the future, we will keep enhancing the embeddings distilled from streamer discourses and viewer messages. Recent language models have started investigating BERT enhancement in terms of scalability and training efficiency. To better comprehend viewer messages distracted by informal tokens such as social buzzwords and slang, and also to process long streamer discourses, we will enhance our embedding process with advanced language models. In this study, we encode streamer discourses and viewer messages independently. To better capture steamer-viewer interactions, and so as to effectively


identify the most attractive part of a video, in the future, we will also investigate encoding architectures to concurrently process these types of textual information. Finally, to produce better highlight extraction results, we will explore more side information such the sentence intensity of viewer messages and stream discourse within a segment, which indirectly reflect the emotion of information senders and receivers.

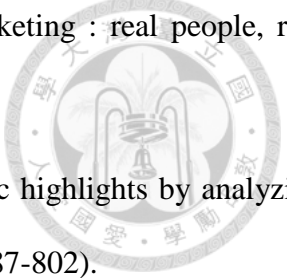


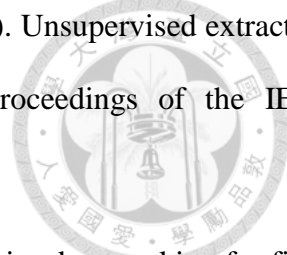
## References

- Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. Paper presented at 3rd International Conference on Learning Representations, ICLR 2015, San Diego, United States.
- Brink, A. D., & Pendock, N. E. (1996). Minimum cross-entropy threshold selection. *Pattern recognition*, 29(1), 179-188.
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). 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* (pp. 4171-4186).
- Duprez, C., Christophe, V., Rimé, B., Congard, A., & Antoine, P. (2015). Motives for the social sharing of an emotional experience. *Journal of Social and Personal Relationships*, 32, 757-787.
- Erkan, G., & Radev, D. R. (2004). Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22, 457-479.
- Fu, C.-Y., Lee, J., Bansal, M., & Berg, A. (2017). Video highlight prediction using audience chat reactions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (pp. 972-978).
- Han, H.-K., Huang, Y.-C., & Chen, C. C. (2019). 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* (pp. 75-81).
- He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 2961-2969).

- 
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 770-778).
  - Hearst, M. A. (1998). Support vector machines. *IEEE Intelligent Systems*, 13, 18–28.
  - Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9, 1735-1780.
  - Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, 1097-1105.
  - Jiao, Y., Li, Z., Huang, S., Yang, X., Liu, B., & Zhang, T. (2018). Three-dimensional attention-based deep ranking model for video highlight detection. *IEEE Transactions on Multimedia*, 20(10), 2693-2705.
  - Lee, Y. J., Ghosh, J., & Grauman, K. (2012). Discovering important people and objects for egocentric video summarization. In 2012 IEEE Conference on Computer Vision and Pattern Recognition (pp. 1346-1353).
  - Loshchilov, I., & Hutter, F. (2019). Decoupled weight decay regularization. Paper presented at 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA.
  - Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.
  - Nepal, S., Srinivasan, U., & Reynolds, G. (2001). Automatic detection of 'Goal' segments in basketball videos. In Proceedings of the ninth ACM International Conference on Multimedia (pp. 261-269).

- 
- Otsuka, I., Nakane, K., Divakaran, A., Hatanaka, K., & Ogawa, M. (2005). A highlight scene detection and video summarization system using audio feature for a personal video recorder. *IEEE Transactions on Consumer Electronics*, 51, 112-116.
  - Ozsoy, M. G., Alpaslan, F. N., & Cicekli, I. (2011). Text summarization using latent semantic analysis. *Journal of Information Science*, 37, 405-417.
  - Rani, S., & Kumar, M. (2020). Social media video summarization using multi-Visual features and Kohnen's Self Organizing Map. *Information Processing & Management*, 57(3), 102190.
  - Ren, S., He, K., Girshick, R., & Sun, J. (2016). Faster r-cnn: towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39, 1137-1149.
  - Ringer, C., & Nicolaou, M. A. (2018). Deep unsupervised multi-view detection of video game stream highlights. In *Proceedings of the 13th International Conference on the Foundations of Digital Games* (pp. 1-6).
  - Rochan, M., Reddy, M. K. K., Ye, L., & Wang, Y. (2020, August). Adaptive video highlight detection by learning from user history. In *European Conference on Computer Vision* (pp. 261-278). Springer, Cham.
  - Rui, Y., Gupta, A., & Acero, A. (2000). Automatically extracting highlights for TV baseball programs. In *Proceedings of the eighth ACM International Conference on Multimedia* (pp. 105-115).
  - Shen, D., Sun, J.-T., Li, H., Yang, Q., & Chen, Z. (2007). Document summarization using conditional random fields. In *Proceedings of the 20th International Joint Conference on Artificial intelligence, IJCAI* (Vol. 7, pp. 2862-2867).

- 
- Solomon, M. R., Marshall, G. W., & Stuart, E. W. (2008). *Marketing : real people, real choices*. Upper Saddle River (N.J.): Pearson/Prentice Hall.
  - Sun, M., Farhadi, A., & Seitz, S. (2014). Ranking domain-specific highlights by analyzing edited videos. In *European Conference on Computer Vision* (pp. 787-802).
  - Tas, O., & Kiyani, F. (2007). A survey automatic text summarization. *PressAcademia Procedia*, 5, 205-213.
  - Tjondronegoro, D., Chen, Y.-P. P., & Pham, B. (2004). Highlights for more complete sports video summarization. *IEEE multimedia*, 11, 22-37.
  - Tran, D., Bourdev, L., Fergus, R., Torresani, L., & Paluri, M. (2015). Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 4489-4497).
  - Wang, Z., Zhou, J., Ma, J., Li, J., Ai, J., & Yang, Y. (2020). Discovering attractive segments in the user-generated video streams. *Information Processing & Management*, 57(1), 102130.
  - Wei, Z., Wang, B., Hoai, M., Zhang, J., Lin, Z., Shen, X., Měch, R., & Samaras, D. (2018). Sequence-to-segments networks for segment detection. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems* (pp. 3511-3520).
  - Wong, T. T. (2015). Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation. *Pattern Recognition*, 48(9), 2839-2846.
  - Xiong, B., Kalantidis, Y., Ghadiyaram, D., & Grauman, K. (2019). Less is more: learning highlight detection from video duration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 1258-1267).

- 
- Yang, H., Wang, B., Lin, S., Wipf, D., Guo, M., & Guo, B. (2015). Unsupervised extraction of video highlights via robust recurrent auto-encoders. In Proceedings of the IEEE International Conference on Computer Vision (pp. 4633-4641).
  - Yao, T., Mei, T., & Rui, Y. (2016). Highlight detection with pairwise deep ranking for first-person video summarization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 982-990).
  - Yeh, J.-Y., Ke, H.-R., Yang, W.-P., & Meng, I.-H. (2005). Text summarization using a trainable summarizer and latent semantic analysis. *Information Processing & Management*, 41, 75-95.
  - Zhang, B., Dou, W., & Chen, L. (2006). Combining short and long term audio features for TV sports highlight detection. In European Conference on Information Retrieval (pp. 472-475).
  - Zhang, Y., Gao, J., Yang, X., Liu, C., Li, Y., & Xu, C. (2020). 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 (Vol. 34, pp. 12902-12909).