AntPivot: A Multi-modal Framework for Livestream Highlight Segmentation

Anonymous ACL submission

Abstract

Livestreaming has become a prominent medium for sharing real-time content, including gaming, sports events, financial investment, and various other forms of live entertainment. However, livestreams can be lengthy, often spanning several hours, making it time-consuming and challenging for users to find the most interesting and engaging moments within the content. In this work, we formulate the definition of *Livestream Highlight Segmentation* and propose the first direct Livestream Highlight Segmentation model *AntPivot* which alleviates the challenges of multi-modal fusion, long duration, and sparse highlights. Specifically, 1) to accelerate the highlight segmentation research in the domain of insurance and fortune, we release a fully-annotated dataset *AntHighlight*; 2) we introduce a multi-modal fusion module to encode the raw data into the unified representation and model their temporal relations to capture clues in a chunked attention mechanism; 3) we propose dynamic-programming decoding to optimize the detection of highlight clips by searching for optimal decision sequences. The extensive experiments demonstrate that AntPivot outperforms text-only models and achieves state-of-the-art results. Ablation Studies further validate the effectiveness of our methods. All the codes and data will be released publicly with the camera-ready version.[1]

1 Introduction

With the explosive growth of transmission speed and storage capacity on the Internet, an increasing amount of information with different levels of importance and usefulness is interwoven into various data flows. Meanwhile, there is also an irreversible tendency that people's available time is becoming more and more fragmented. As a result, users rarely have enough time or attention to separate the valuable information from the other useless part. For the sake of efficiency and convenience in the interaction, it's essential to extract key information from unprocessed data to help users get what they need with little effort. Under this requirement and circumstance, researchers try to design automatic algorithms to segment salient or highlight parts in different kinds of data.

In recent years, significant research (Yu et al., 2019; Su et al., 2020; Zhang et al., 2021; Yu et al., 2022) has introduced joint modeling of vision and language. MCAN (Yu et al., 2019) proposes the deep modular co-attention networks between vision and language, which performs the cross-modal alignment by concisely maximizing the cross-attention. After that, there is a consensus (Wang et al., 2022b,a; Bao et al., 2022) to utilize a cross-attention mechanism to bridge different modalities. VL-BERT (Su et al., 2020) introduce modality-aligned representations for generic vision-language understanding with the MLM paradigm. Despite these advances, there remain significant obstacles to designing multi-modal networks due to differences between modalities, and modeling livestream inputs. A multi-modal highlight segmentation model may open up a host of practical applications: locating highlights to provide users with personalized recommendations, highlight tracking, or expressing information.

Despite the benefits of multi-modal approaches, several challenges remain, including (1) contrary to other kinds of videos, livestreams are usually extremely long in duration, varying from dozens of minutes to several hours, (2) a mass of noise and useless information, such as slips of the tongue, greetings and chit-chats, which harms the performance of methods to a large extent, and (3) there always exist topic shifts and gaps in the expressions of livestreamers, resulting in low coherence and cohesion of corpus.

To this end, we first formulate the task of

[1] * Equal contribution. † Corresponding author.

Livestream Highlight Segmentation as the segmentation and importance evaluation on the temporal dimension of livestreams. Considering there is no benchmark dataset available in this area, a bunch of livestream records in the domain of insurance and fortune are collected from the platform supported by AliPay to construct a new dataset called *AntHighlight* to facilitate this task. To provide an elementary solution to accomplish the goal stated previously, we construct a novel architecture to extract and analyze the semantic information comprehensively and select highlight fragments from the untrimmed livestreams efficiently. Specifically, we first encode the raw data in different views and combine them into the inputs of our model. Afterward, we utilize a novel chunked attention module, named *Pivot Transformer*, to capture temporal dependencies and integrate representations from different semantic levels. Finally, a series of confidences and probabilities are calculated to determine the prediction results in a dynamic-programming manner.

In conclusion, the main contributions of this paper can be summarized in the following aspects:

- We formulate the task of *Livestream Highlight Segmentation* and inject the training objectives and dynamic programming decoding to solve this problem.
- We release the first fully-annotated livestream highlight segmentation benchmark dataset *AntHighlight*.
- Through introducing multimodal fusion and pivot transformer, we propose the first direct livestream highlight segmentation model *AntPivot*, which alleviates the problem of long durations, topic shifts, and sparse highlights.
- Experimental results on the AntHighlight demonstrate that our model outperforms the baselines and achieves state-of-the-art performances.

2 Related Work

Text / Scene Segmentation The task of text segmentation is to split documents or discourse into individual parts. In the early stage, researchers tried to apply some lexicon-based (Hearst, 1997; Choi, 2000) and statistics-based approaches (Utiyama and Isahara, 2001; Eisenstein, 2009) to tackle this problem. Afterward, some efficient neural modules for sequence modeling, such as CRF (Wang et al., 2018), PointerNetwork (Li et al., 2018) and BERT (Lukasik et al., 2020), were also employed to boost better performance and generalization. Similarly, there also exist valuable discussions about the splitting of videos composed of complex scenes. Among them, early works (Rasheed and Shah, 2003; Chasanis et al., 2009) tried to utilize low-level features and carefully design heuristic methods. To explore supervised-learning strategies, some researchers constructed a variety of new datasets based on documentaries (Baraldi et al., 2015), short films (Rotman et al., 2017), long movies (Rao et al., 2020) etc. Different from text/scene segmentation, livestream highlight segmentation needs to model and fusion multimodal inputs and filter out the useless fragments to obtain highlights.

Proposal Generation Given an untrimmed video, the goal of action proposal generation is to ascertain a set of temporal boundaries with high probability or confidence to contain action instances. Current prevailing approaches can be mainly divided into two categories, namely anchor-based methods (Gao et al., 2017; Yang et al., 2021) and boundary-based ones (Tan et al., 2021; Su et al., 2021). The former first define a group of hand-crafted proposal pre-definitions and choose candidates from them in a ranking-based manner, while the latter will directly locate the possible action boundaries in a classification or regression way. Compared with this task, livestream highlight extraction mainly focuses on the comprehension and understanding of streamers' speech data, which have a higher semantic gap and lower redundancy than video data. Besides, the model in our scenario should not generate overlapping proposals, which is allowed and sometimes necessary in the task of action proposal generation.

Multimodal Learning It has been an increasing interest in multimodal modeling on language-visual (Lei et al., 2021; Bao et al., 2022; Yu et al., 2022) and audio-visual (Shi et al., 2022; Huang et al., 2023). BeiT-v3 (Wang et al., 2022a) proposes to take images in a foreign language with a more fine-grained cross-modal mask-and-reconstruction process, sharing partial parameters. Clip-Bert (Lei et al., 2021) employs sparse sampling to enable affordable end-to-end learning for video-and-language tasks. In the domain of audio-visual learning, AV-Hubert (Shi et al., 2022) introduces a self-supervised representation learning

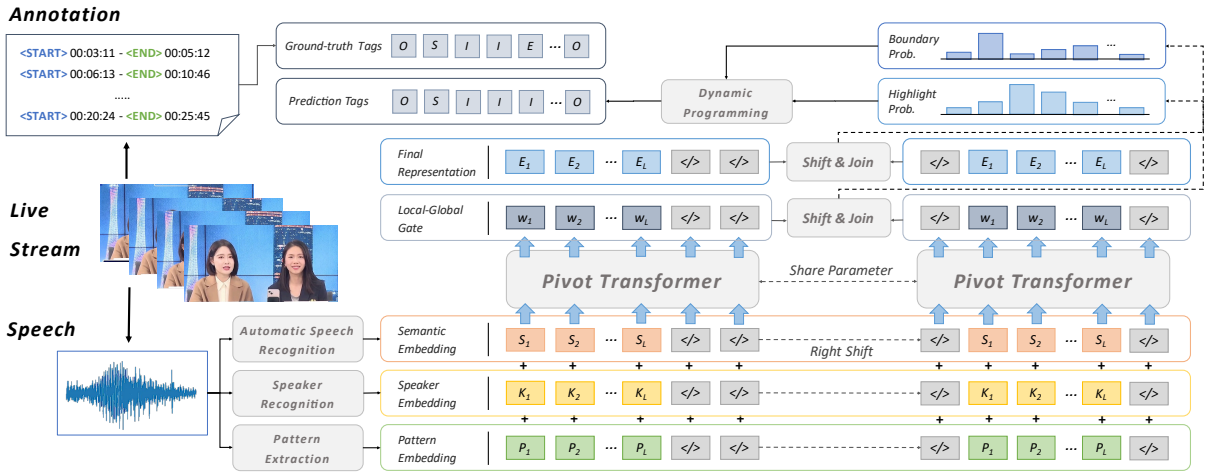


Figure 1: Overall diagram of our proposed *AntPivot* architecture. The </> annotation represents zero-pad in the sequence, and the overall calculations are all conducted in the sentence level.

framework for audio-visual speech. Subsequently, (Huang et al., 2023) leverage the success of AV-Hubert and propose the cross-modal distillation to reduce the data scarcity of visual data. Despite these advances, most research in multimodal learning has focused on two modalities, and may not directly contribute to video, speech, and text modeling.

3 Method

3.1 Problem Definition

Given a long and unprocessed livestream record $\mathbf{R}$, the task of livestream highlight extraction aims to retrieve all proposals for highlight topics and discussions. To be specific, the livestream record can be annotated as $\mathbf{R} = (\mathbf{V}, \mathbf{A})$, where $\mathbf{V}$ and $\mathbf{A}$ represents visual and audio data respectively. And our goal is to construct a proper model to generate a series of proposals covering the most valuable parts of the entire livestream, which can be given as

$$P = \{(s_1, e_1), (s_2, e_2), \ldots, (s_p, e_p)\} \quad (1)$$

where p is the number of proposals and (s_i, e_i) is the start and end timestamp for i-th proposal satisfying

$$s_i < e_i < s_{i+1} < e_{i+1}, \forall i \in \{1, 2, \ldots, p-1\}.$$

3.2 Overall Architecture

Figure 1 demonstrates the overall architecture and calculation procedure of our proposed method. We introduce a data-to-representation modeling procedure to project data to a final representation. And then, we apply the pivot transformer to help us model the context information and estimate highlight scores for each utterance, and we will use the final representations to calculate the confidences of utterances to be boundaries. Finally, the boundary confidences and highlight scores will be aggregated and used as a reference in the dynamic-programming calculation.

3.3 Data-to-Representation Modeling

We propose a novel multimodal modeling scheme designed to transform data across various modalities into the final embedding. Because of the lack of a unified multimodal network to model different modalities, We first get the different modality embeddings and then fuse them. The details of embedding are as follows.

Semantic Embedding We transform the speech data into the corresponding transcripts via an automatic speech recognition module. Afterward, a pre-trained language model is employed to squeeze every sentence of transcripts into a single embedding, which can be annotated as $\mathbf{S} = \{\mathbf{s}_i\}_{i=1}^{L}$ where L is the number of transcripts.

Speaker Embedding We introduce the speaker information to alleviate the problem of topic shifts. In practice, we turn to the solution proposed in (Wang et al., 2021) for effective speaker verification and then project the identification label into speaker embeddings, given by $\mathbf{K} = \{\mathbf{k}_i\}_{i=1}^{L}$.

Pattern Embedding Considering that most streamers tend to switch their mood, stress, or pitch of voice when talking about something important or valuable to arouse the audience's attention and interest, we downsample the mel-frequency spectrum

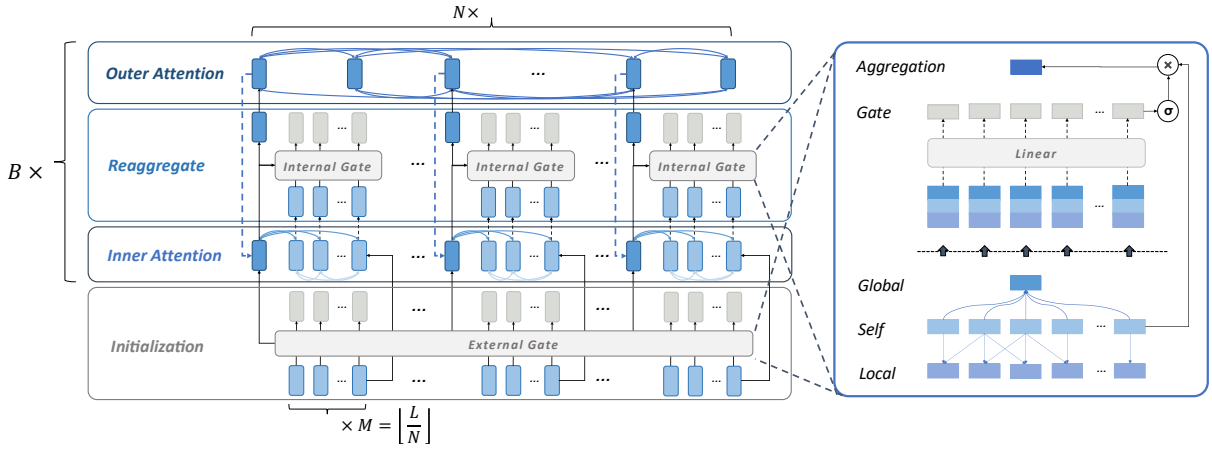


Figure 2: Detailed structure of our proposed *Pivot Transformer* (left part) and gating mechanism (right part). The **blue dashed line** represents dataflow between adjacent iterations.

of speech pieces into a fixed length and concatenate them sequentially to seek out useful temporal patterns within every utterance, which is annotated as $\mathbf{P} = \{\mathbf{p}_i\}_{i=1}^{L}$.

Video Embedding To obtain visual information such as livestream popularity, attention, and charts, a pre-trained 3D CNNs model is adopted to map the video fragments into video embeddings, which is annotated as $\mathbf{V} = \{\mathbf{V}_i\}_{i=1}^{L}$.

To alleviate the embedding dimension mismatch among different embeddings, we add the modality adaptor consisting of a multilayer perceptron and then include a modality dropout to mask the full features of one modality before fusing audio and visual inputs. After that, we compose them in an addition-based manner. Specifically, the final embeddings which are fed into our model can be given as $\mathbf{E} = \{\mathbf{e}_i\}_{i=1}^{L}$, where $\mathbf{e}_i = \mathbf{s}_i + \mathbf{p}_i + \mathbf{k}_i + \mathbf{v}_i$ and all these embeddings are projected into the space of $\mathbb{R}^d$. [2]

3.4 Pivot Transformer

Due to the large computation cost of global attention mechanism, it can be unbearable to utilize a vanilla transformer to deal with livestreams with a long duration under some constraints of devices. Additionally, there also exist massive topic shifts and irrelevant information, which makes the denoising and purification of information significant. In consideration of this, we devise a novel *Pivot Transformer* to alleviate the challenges of long durations, sparse highlights, and topic shifts.

Initialization In the first stage, we employ a bi-directional gated recurrent unit proposed by (Cho et al., 2014) to generate the initial individual and global representations, given by

$$\mathbf{g}, [\tilde{\mathbf{e}}_1, \cdots, \tilde{\mathbf{e}}_L] = \textit{Bi-GRU}([\mathbf{e}_1, \cdots, \mathbf{e}_L]), \quad (2)$$

where $\mathbf{g} \in \mathbb{R}^d$ and $\tilde{\mathbf{e}}_i \in \mathbb{R}^d$ are the final state and output for the i-th step respectively. After that, the input features will be divided into N multiple consecutive chunks with the length of $M = \lfloor \frac{L}{N} \rfloor$, and the rearranged sequence of utterances can be ordered as

$$\mathbf{r}_{i,j}^0 = \tilde{\mathbf{e}}_{(i-1)\times M+j}, \quad 1 \le i \le N, 1 \le j \le M, \quad (3)$$

where $\mathbf{r}_{i,j}^0 \in \mathbb{R}^d$ represents the j-th element in the i-th chunk. And then, an *External Gate* module (will be described below) is applied to generate a group of weights and dynamically aggregate elements into higher-level features which are called *pivots* in this paper. The calculation can be formulated by

$$\mathbf{w}_i^0, \mathbf{t}_i^0 = \textit{ExternalGate}(\mathbf{r}_{i,1}^0, \cdots, \mathbf{r}_{i,M}^0), \quad (4)$$

where $\mathbf{w}_i^0 \in \mathbb{R}^M$ and $\mathbf{t}_i^0 \in \mathbb{R}^d$ represent the gating weights and initial pivot of i-th chunk respectively.

Update Mechanism Given the initialized pivots and elements, we try to exploit a chunked attention mechanism to incorporate context information efficiently. To be specific, in the l-th loop of interaction, a multi-head attention mechanism proposed in (Vaswani et al., 2017) will be first adopted within every chunk to integrate local information, formulated as

$$[\tilde{\mathbf{t}}_i^l, \mathbf{r}_{i,1}^l, \cdots, \mathbf{r}_{i,M}^l] = \textit{MHA}([\mathbf{t}_i^{l-1}, \mathbf{r}_{i,1}^{l-1}, \cdots, \mathbf{r}_{i,M}^{l-1}]), \quad (5)$$

where $\textit{MHA}(\cdots)$ stands for the standard multi-head attention calculation. Subsequently, the pivot

[2] The dimension of representations keeps the same unless specified in the following sections.

features will be reaggregated using an *Internal Gate* module (will be described below), given by

$$\mathbf{w}_i^l, \hat{\mathbf{t}}_i^l = \textit{InternalGate}(\tilde{\mathbf{t}}_i^l, \mathbf{r}_{i,1}^l, \cdots, \mathbf{r}_{i,M}^l), \quad (6)$$

After this operation, the pivot representations can be treated as a reasonable and refined compression in a local range. To further capture the global context, another attention computation will be carried on the sequence of pivots, given as

$$[\mathbf{t}_1^l, \cdots, \mathbf{t}_N^l] = \textit{MHA}([(\hat{\mathbf{t}}_1^l + \tilde{\mathbf{t}}_1^l), \cdots, (\hat{\mathbf{t}}_N^l + \tilde{\mathbf{t}}_N^l)]) \quad (7)$$

In the next step, the updated pivots will be fed into the $(l+1)$-th loop to pass information to the elements within every chunk and the overall procedure mentioned above will be repeated for a total of B times. In this iterative calculation, the pivots actually act as a vital role to interact between local and global ranges. After the updating iterations, we first flatten the gating weights and updated representations as described below to proceed the final prediction.

$$\hat{w}_{(i-1)\times M+j}^l = (\mathbf{w}_i^l)_j, \quad \hat{\mathbf{e}}_{(i-1)\times M+j} = \mathbf{r}_{i,j}^B, \quad (8)$$

After this, we calculate the average value of gating weights in different layers as the highlight confidences, and predict the boundary probabilities using the final representations, which can be given by

$$h_i = \frac{1}{B+1}\sum_{j=0}^{B}\sigma(\hat{w}_i^j), \quad b_i = \mathrm{MLP}(\hat{\mathbf{e}}_i), \quad (9)$$

where $\mathrm{MLP}(\cdot)$ is a multi-layer perceptron module and $\sigma(\cdot)$ is the sigmoid function with $\sigma(\mathbf{x}) = \frac{1}{1+e^{-\mathbf{x}}}$.

It's worth mentioning that we copy the original sequence, shift the elements $M/2$ to the right and repeat the above operations in practice as shown in the right part of Figure 1, which is designed to keep the range of local receptive field at $3M/2$ to prevent the influence caused by absolute element positions. And all the final predictions will be shifted back and calculate the average results with the normal sequence.

Local-Global Gate In this paragraph, we will introduce the gating mechanism used in the structure of pivot transformer, which is designed to estimate the importance of every utterance to predict the highlight score and ensure the essential information to be squeezed into the corresponding pivots. In the calculation, we consider the local context, global information and individual representation synthetically. As depicted in the framed region of Figure 2, a group of global and local features will be generated and concatenated with the original sequence and then the gating weights will be produced via a multi-layer perceptron and be utilized to form the aggregation features, given as:

$$w_i = \mathrm{MLP}([\mathbf{g}; \mathbf{l}_i; \mathbf{r}_i]), \quad \mathbf{t} = \sum_i \frac{e^{w_i}}{\sum_j e^{w_j}}\mathbf{r}_i, \quad (10)$$

where $[;]$ is the concatenation operator and $\mathbf{g}, \mathbf{l}_i, \mathbf{r}_i$ stand for the global, local and individual representations corresponding to the i-th element respectively.

According to the position of the gating unit in the structure, we further customize different schema for the *external* and *internal* ones. In the former ones, we utilize a 1D-convolution operator with a receptive field of $M/2$ to capture local information and take the final state of Bi-GRU as the global information. And for the latter, we directly treat the pivot features in the previous step as the local representations and calculate the global one by averaging all the elements in the sequence.

Complexity Analysis and Comparison Given the description of operations in this architecture, we analyze and compare the theoretical complexity of pivot transformer and the vanilla one in this part. By decomposing the original computation into a two-step chunked mechanism, the cost of $\mathcal{O}(L^2 d)$ in the attention layer can be reduced to $\mathcal{O}(N \cdot M^2 d + N^2 d) = \mathcal{O}((\frac{L^2}{N} + N^2)d)$. The complexity achieves the optimum of $\mathcal{O}(L^{\frac{4}{3}}d)$ with $N = \Theta(L^{\frac{2}{3}})$, but in practice we set $N = M = \sqrt{L}$ for better efficiency in parallel computing.

3.5 Training Loss

Two Objectives have been used to optimize the AntPivot model.

Boundary Loss It is used to guide an accurate boundary recognition for the highlight clips, which can be formulated as:

$$\mathcal{L}_b = -\sum_{i=1}^{L}(\bar{b}_i \log(b_i) + (1-\bar{b}_i)\log(1-b_i)), \quad (11)$$

Highlight Loss It's still difficult for the model to address this problem without any extra knowledge because there are still no adequate clues for the selection and filter of highlight parts. Therefore, we further apply the item of *Highlight Loss* to help the model discriminate highlight segments from others, given by

$$\mathcal{L}_h = -\sum_{i=1}^{L}(\bar{h}_i \log(h_i) + (1-\bar{h}_i)\log(1-h_i)), \tag{12}$$

where $\bar{b}_i \in \{0,1\}$ and $\bar{h}_i \in \{0,1\}$ are the boundary and highlight indicators for the ground-truth annotation of i-th utterance respectively.

Finally, the overall loss function in the training process can be composed in a weighted way, given as

$$\mathcal{L} = \mathcal{L}_b + \lambda \mathcal{L}_h, \tag{13}$$

where λ is the hyper-parameter to balance these two parts.

3.6 Dynamic-programming Decoding

Given the highlight confidences $\{h_i\}_{i=1}^{L}$ and the boundary probabilities $\{b_i\}_{i=1}^{L}$, our goal is to ascertain an optimal prediction sequence to maximize the accumulative score on the final decision path. To make it clear, we first categorize all the possible states of the i-th utterance into four types:(1) the start boundary of the proposal; (2) the middle position of the proposal; (3) the end boundary of the proposal; (4) not contained in any proposal.

And then, we use $f_{i,j}$ to represent the maximal score accumulated to the i-th element in the j-th state listed above. Therefore, the state transition equation can be designed as

$$f_{i,j} = \begin{cases} \max(f_{i-1,2}, f_{i-1,4}) + b_i & j = 1 \\ \max(f_{i-1,1}, f_{i-1,3}) + b_i & j = 2 \\ \max(f_{i-1,1}, f_{i-1,3}) + h_i\bar{b}_i & j = 3 \\ \max(f_{i-1,2}, f_{i-1,4}) + \bar{h}_i\bar{b}_i & j = 4 \end{cases} \tag{14}$$

where $\bar{h}_i$ and $\bar{b}_i$ respectively denote $1-h_i$ and $1-b_i$, the initial states are set as $f_{1,1} = b_1$, $f_{1,2} = f_{1,3} = -\infty$ and $f_{1,4} = (1-h_1)(1-b_1)$ because only the *start* and *out* state are legal for the first utterance. Given the states and transition equations defined above, we can make the predictions in a dynamic-programming way, and record all the decisions related to the optimal result along the sequence.

Afterward, we backtrack the optimal decision path from the better states in $f_{L,3}$ or $f_{L,4}$ to recover the entire sequence of choices and predictions. In this way, we can explore all valid combinations of decisions holistically and generate a stable and reliable result. Finally, we will map the sentence-level predictions to the corresponding timestamps so as to generate results in seconds.

4 Experiment

4.1 Metrics

To evaluate the effectiveness of models objectively and automatically, we adopt two criteria widely used in the related fields, namely **Average Precision** and **Boundary F1 Score**.

Average Precision The task of livestream highlight detection aims to generate proposals to cover the target highlight parts tightly. Therefore, the quality of predictions is determined by the overlap with the ground-truth intervals. Following the conventional protocol in the area of action segmentation, we use Average Precision with tIoU thresholds {0.5, 0.6, 0.7, 0.8, 0.9} to measure the performance.

Boundary F1 Score In addition to the IoU-based metrics, the evaluation of boundary classification should get concerned as well, because an accurate boundary prediction can not only boost the overall precision but also greatly reduce the expense of manual modification and revision to the final results. Considering the application scenarios in reality, we directly desert the intervals within 10 seconds and treat the predictions with a minimal difference of fewer than 5 seconds from ground-truth boundaries as correct ones. Under this circumstance, every predicted boundary will match at most one ground-truth one in the metric calculation. Take the evaluation of start timestamps as an example, the F1 score can be calculated as

$$\tilde{p} = \sum_{i=1}^{p} \mathbb{I}(\min_{j \in \{1,\cdots,\bar{p}\}}(|s_i - \bar{s}_j|) < 5), \tag{15}$$

$$F_1 = \frac{2 \times Prec \times Rec}{Prec + Rec} = \frac{2\tilde{p}}{(\bar{p}+p)}, \tag{16}$$

where $\mathbb{I}(\cdot)$ is the indicator function, and $\{\bar{s}_i, \bar{e}_i\}_{i=1}^{\bar{p}}$ and $\{s_i, e_i\}_{i=1}^{p}$ stand for the ground-truth proposals and predicted ones, respectively.

4.2 Implementation Details

Model Setting Considering the maximal sequence length reaches about 900, we set $N = \sqrt{L} = 30$ as mentioned in the previous section.

Method	Average Precision					F1	
	0.5	0.6	0.7	0.8	0.9	Start	End
Sent-Bert	66.6	57.0	49.2	41.7	33.2	42.7	46.0
XlNet	67.1	58.4	50.6	43.1	34.3	41.3	45.1
Roberta	67.3	58.4	50.9	43.7	36.1	44.1	47.9
XL-Transformer	71.5	61.9	54.6	46.5	38.9	45.0	49.7
Longformer	71.6	61.0	54.7	46.6	39.4	**45.7**	49.3
AntPivot	**72.7**	**63.4**	**55.7**	**48.0**	**39.5**	45.6	**50.2**
- Pivot	70.7	61.2	53.9	45.5	37.3	47.1	48.2
- Shift	71.9	61.7	54.3	45.8	38.4	45.9	46.8
- MultiModal Fusion	70.6	61.2	53.5	45.7	36.5	43.8	48.3

Table 1: Comparison with baselines and ablation studies on the AntHighlight dataset. The best results are given in bold.**bold**.

This setting is also available for the situation where $L > 900$, and the practical complexity will increase accordingly. The dimension d used in our model is set as 256. For all transformer-based architecture, the number of heads is 4 and sandwich layernorm mechanism (i.e. both pre-LN and post-LN are utilized) is adopted. In the pivot transformer, B is set as 3 and the number of attention layer stacked in every stage is 2. As for the transformers and GRU in the experiment part, the number of layers is set as 6 to keep consistency. Apart from this, the attention length of Transformer-XL and the attention window size of Longformer are all set as $N = \sqrt{L} = 30$.

Optimization and Inference All the experiments are conducted on one piece of Tesla P100. In the training procedure, we employ AdamW optimizer proposed by (Loshchilov and Hutter, 2019) with warmup strategy (Vaswani et al., 2017) and cosine annealing learning (Loshchilov and Hutter, 2016). The maximal learning rate of is set as 3e-4 and the weight decay is fixed at 5e-5. To prevent overfitting, a dropout strategy with $p = 0.4$ is applied in the structure. The training will last for 20 epochs and we select the checkpoint with the best performance on the validation dataset. And in the inference stage, all the scores will be pre-processed via a min-max normalization to guarantee a stable prediction. The thresholds in *Simple* strategy are set as $t_b = 0.25$ and $t_h = 0.7$.

4.3 Model Performance

In this study, we conduct a comprehensive comparison with other systems, including 1) Sentence-BertForSequenceClassfication (Reimers and Gurevych, 2020), abbreviated as Sent-Bert; 2) XlNetForSequenceClassfication (Yang et al., 2020), abbreviated as XlNet; 3) RobertaForSequenceClassfication (Xu, 2021), abbreviated as Roberta; 4) Longformer (Beltagy et al., 2020), replacing the pivot transformer with longformer; 5) Transformer-XL (Dai et al., 2019), replacing the pivot transformer with transformer-xl. The results, compiled and presented in Table 1, provide valuable insights into the effectiveness of our approach:

(1) Our model surpassed all sentence classification baselines across all metric scores. This shows the superiority of our proposed AntPivot for livestream highlight segmentation compared to traditional sentence classification models; (2) The comparison with different model structures demonstrates that the long-term memory indeed makes a difference in this task, which can be easily captured and maintained by Transformer-based architecture. Moreover, the distant information can be further denoised and compressed by our proposed mechanism, resulting in better overall performance. (3) As the tIoU thresholds increase, a distinct degradation could be witnessed in average precision.

4.4 Preliminary Analyses

In this section, we will conduct some experiments to compare and analyze the performances with model inputs and prediction strategies.

Analysis on Modal Inputs Table 2 demonstrates the performance difference between multiple settings of modal input combinations. The capital letters "S" / "K" / "P"/ "V" represent the usage of semantic / speaker / pattern embeddings / video embeddings, respectively. The results are compiled and presented in Table 2, and we have the following observations:(1) Compared with the pattern embedding, the speaker information improves the performance of livestream highlight segmenta-

Method	Average Precision					F1	
	0.5	0.6	0.7	0.8	0.9	Start	End
Modal Inputs Analyses							
S	70.6	61.2	53.5	45.7	36.5	43.8	48.3
S+V	70.9	61.4	53.7	45.5	36.9	44.4	48.5
S+P	71.0	61.6	54.1	45.3	36.4	44.9	48.8
S+K	71.0	61.9	54.6	46.5	38.3	45.5	49.7
Prediction Strategy Analyses							
Simple	70.8	60.0	52.6	42.6	34.9	37.8	43.2
Greedy	65.5	58.1	51.2	44.4	37.1	43.8	45.7
Ours	**72.7**	**63.4**	**55.7**	**48.0**	**39.5**	**45.6**	**50.2**

Table 2: Preliminary analyses on modal inputs and prediction strategy. The best results are given in **bold**.

tion across almost all metric values, especially in IOU@0.5 and F1-End. We assume that the speaker information is helpful to resolve the frequent topic shifts in livestreams; (2) Integrating all types of information, rather than just one or two, yields the best performance in livestream highlight segmentation, as evidenced by significant improvements across all metrics. Notably, the tIoU-0.9 value increased from 36.5 to 39.5, highlighting the importance of all four types of information and validating the effectiveness of our proposed data processing schema.

Analysis on Model Structure We investigate the inner structure of AntPivot. As shown in Table 1, we remove the shifting process, multimodal fusion, and pivot mechanism (i.e. only the attention computation inside each chunk is conducted) to verify their effects. Without the interaction between pivot elements, the global information cannot get transmitted and utilized in the calculation, thus hindering the model from understanding the entire content comprehensively. Besides, there exists an apparent margin on the boundary F1 score in the absence of a shifting procedure, which infers this operation can alleviate the impact brought by the absolute position of elements and enhance the discriminative ability in the local range. The removal of multimodal fusion hurts the performance of AntPivot, which shows the effectiveness of this module.

Analysis on Prediction Strategy To assess the effect of different prediction approaches in the inference stage, we further develop two other baseline strategies to compare with the dynamic-programming method.

- **Simple**: we directly pick out all boundary candidates with the constraint of threshold t_b and select all proposals with an average highlight score greater than t_h.
- **Greedy**: We convert this task into a multi-class problem in this setting and make pairs between positions predicted as "start" and "end" categories.

From Table 2, our proposed strategy behaves best among them, and the *Simple* one is much inferior to the others. The reason can be inferred intuitively that the *Simple* strategy disrupts the order of precedence, thus impeding the model from distinguishing the boundary type (i.e. start or end), and the assignment of threshold restricts the generalization of this setting. As for the *Greedy* approach, it actually ignores the influence of relative differences in confidences and probabilities, resulting in a coarse and inaccurate result.

5 Conclusion

In this paper, we propose a novel livestream highlight segmentation task to promote the development of livestream in various fields. To accelerate the development of the research community in livestream highlight segmentation, we collect and release the first publicly accessile dataset for livestream highlight segmentation called AntHightlight. To address the challenges that live stream presents, such as extreme durations, large topic shifts, and much irrelevant information, we develop a chunked attention mechanism and gating strategy to efficiently integrate information, and design a dynamic programming strategy to generate final predictions. The comprehensive experiments demonstrate the practicality of this contributed dataset and the effectiveness of our proposed method and strategy.

References

Hangbo Bao, Wenhui Wang, Li Dong, Qiang Liu, Owais Khan Mohammed, Kriti Aggarwal, Subhojit Som, and Furu Wei. 2022. Vlmo: Unified vision-language pre-training with mixture-of-modality-experts.

Lorenzo Baraldi, Costantino Grana, and Rita Cucchiara. 2015. A deep siamese network for scene detection in broadcast videos. *Proceedings of the 23rd ACM international conference on Multimedia*.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *ArXiv*, abs/2004.05150.

Vasileios Chasanis, Aristidis Likas, and Nikolas P. Galatsanos. 2009. Scene detection in videos using shot clustering and sequence alignment. *IEEE Transactions on Multimedia*, 11:89–100.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *EMNLP*.

Freddy Y. Y. Choi. 2000. Advances in domain independent linear text segmentation. In *ANLP*.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *ArXiv*, abs/1901.02860.

Jacob Eisenstein. 2009. Hierarchical text segmentation from multi-scale lexical cohesion. In *NAACL*.

J. Gao, Zhenheng Yang, Chen Sun, Kan Chen, and Ramakant Nevatia. 2017. Turn tap: Temporal unit regression network for temporal action proposals. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3648–3656.

Marti A. Hearst. 1997. Text tiling: Segmenting text into multi-paragraph subtopic passages. *Comput. Linguistics*, 23:33–64.

Rongjie Huang, Huadai Liu, Xize Cheng, Yi Ren, Linjun Li, Zhenhui Ye, Jinzheng He, Lichao Zhang, Jinglin Liu, Xiang Yin, and Zhou Zhao. 2023. Avtranspeech: Audio-visual robust speech-to-speech translation.

Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L. Berg, Mohit Bansal, and Jingjing Liu. 2021. Less is more: Clipbert for video-and-language learning via sparse sampling.

J. Li, Aixin Sun, and Shafiq R. Joty. 2018. Segbot: A generic neural text segmentation model with pointer network. In *IJCAI*.

Ilya Loshchilov and Frank Hutter. 2016. Sgdr: Stochastic gradient descent with restarts. *ArXiv*, abs/1608.03983.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *ICLR*.

Michal Lukasik, Boris Dadachev, Gonccalo Simoes, and Kishore Papineni. 2020. Text segmentation by cross segment attention. *ArXiv*, abs/2004.14535.

Anyi Rao, Linning Xu, Yu Xiong, Guodong Xu, Qingqiu Huang, Bolei Zhou, and Dahua Lin. 2020. A local-to-global approach to multi-modal movie scene segmentation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10143–10152.

Zeeshan Rasheed and Mubarak Shah. 2003. Scene detection in hollywood movies and tv shows. *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, 2:II–343.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Daniel Rotman, Dror Porat, and Gal Ashour. 2017. Optimal sequential grouping for robust video scene detection using multiple modalities. *Int. J. Semantic Comput.*, 11:193–208.

Bowen Shi, Wei-Ning Hsu, Kushal Lakhotia, and Abdelrahman Mohamed. 2022. Learning audio-visual speech representation by masked multimodal cluster prediction.

Haisheng Su, Weihao Gan, Wei Wu, Junjie Yan, and Y. Qiao. 2021. Bsn++: Complementary boundary regressor with scale-balanced relation modeling for temporal action proposal generation. In *AAAI*.

Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. 2020. Vl-bert: Pre-training of generic visual-linguistic representations.

Jing Tan, Jiaqi Tang, Limin Wang, and Gangshan Wu. 2021. Relaxed transformer decoders for direct action proposal generation. *ArXiv*, abs/2102.01894.

Masao Utiyama and Hitoshi Isahara. 2001. A statistical model for domain-independent text segmentation. In *ACL*.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *ArXiv*, abs/1706.03762.

Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, and Furu Wei. 2022a. Image as a foreign language: Beit pretraining for all vision and vision-language tasks.

Yizhong Wang, Sujian Li, and Jingfeng Yang. 2018. Toward fast and accurate neural discourse segmentation. In *EMNLP*.

Zhiming Wang, Furong Xu, Kaisheng Yao, Yuan Cheng, Tao Xiong, and Huijia Zhu. 2021. Antvoice neural speaker embedding system for ffsvc 2020. *Interspeech 2021*.

Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. 2022b. Simvlm: Simple visual language model pretraining with weak supervision.

Zhuo Xu. 2021. Roberta-wwm-ext fine-tuning for chinese text classification.

Haosen Yang, Wenhao Wu, Lining Wang, Sheng Jin, Boyang Xia, Hongxun Yao, and Hujie Huang. 2021. Temporal action proposal generation with background constraint.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2020. Xlnet: Generalized autoregressive pretraining for language understanding.

Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. 2022. Coca: Contrastive captioners are image-text foundation models.

Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. 2019. Deep modular co-attention networks for visual question answering.

Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. 2021. Vinvl: Revisiting visual representations in vision-language models.

Shiliang Zhang, Zhifu Gao, Haoneng Luo, Ming Lei, Jie Gao, Zhijie Yan, and Lei Xie. 2020. Streaming chunk-aware multihead attention for online end-to-end speech recognition. *ArXiv*, abs/2006.01712.

A The AntHighlight Dataset

A.1 Overview

As a novel task, Livestream Highlight Detection lacks a proper and available dataset to serve as a benchmark. In light of this, we construct the *AntHighlight* dataset by collecting a series of livestream records and annotating all the boundary timestamps for the highlight segments within them. It consists of 3,256 livestream records focused on the theme of funds and wealth from the platform supported by Alipay. AntHighlight contains almost 2649 hours of videos recorded in a real livestream environment with human-annotated highlight labels. Each video is divided into numerous clips according to sentences, and each clip is labeled with whether it is a highlight or not. The major features of AntHighlight include:

- Open source. A lack of data could hinder the construction of livestream highlight detection systems, so we release our corpus to accelerate research in the community.
- Authenticity. To address the issue of extracting highlights in real livestream scenearios, we collect livesreams from Alipay platform.
- High quality. Strict labeling rules and labeling process ensure the quality of the label, and we further ensure the quality of the label through manual review.

A.2 Data Collection and Labeling

Collection Procedure To gather a set of recorded live videos related to financial topics, we collected Chinese live streaming data from the Alipay platform for three weeks, which encompassed solo live broadcasts, two-person live broadcasts, and live broadcasts with more than two individuals. We then generate transcripts using LC-SAN-M, which is pretrained on a 2000 hours Mandarin ASR task he finetune on a 60 hours Mandarin corpus. In addition, we have hired 10 annotators to watch the complete live recordings and formulated strict annotation standards and processes to complete the data annotation.

Labeling Procedure During the data annotation stage, we first use common phrases such as "直播正式开始", "欢迎大家来到直播间", "各位直播间的伙伴们，大家好" to determine the start position of the live broadcast. Next, we identify the start and end positions of the topics(i.e. hightlights), which are mainly categorized into two situations: (1)When there is a switch in speakers (which we determine based on the appearance of spk1 and spk2 in the transcripts), we consider it as the end position of the current speaker if spk2 talks about another subject. (2)When there is no change in speakers, we rely on our custom annotation rules, such as defining topic transitions, identifying connecting words, and using common phrases, to determine the start and end positions of a topic. In this way, we ensure that all text is accurately annotated. Finally, we conduct manual verfication to ensure the high quality of the annotation results.

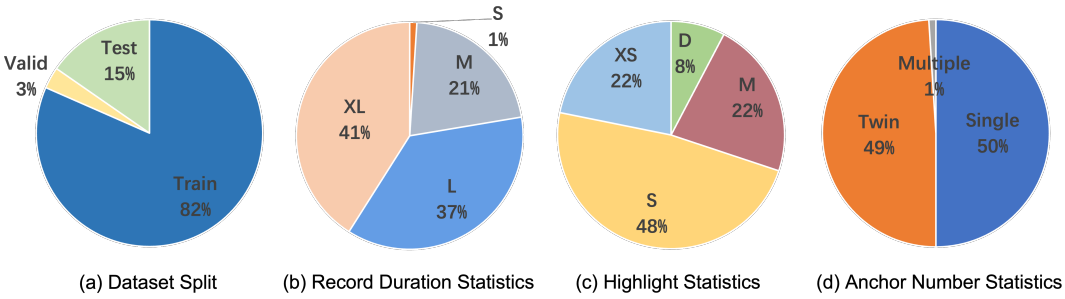
A.3 Statistical Analysis

After the data collection and labeling procedure, we split the dataset and conduct the statistical evaluation.

Dataset Split For the purpose of training and evaluation, we randomly divide the annotated data into three subsets, including training, validation, and test dataset with the size of 2656, 100, and 500 respectively, as shown in Figure 3(a).

Record Duration Statistics We classify the videos in our dataset according to their duration to better understand their distribution. Specifically, we custom videos lasting less than 15 minutes as short, those spanning between 15 to 30 minutes as medium-length, those between 30 minutes to one hour as long, and those exceeding one hour as extremely long. As shown in Figure 3(b), over 78% of the videos in our dataset are longer than half an hour, with less than 1% of videos lasting less than 15 minutes. On average, the videos in our dataset last approximately 49 minutes. As a result, the presence of extremely long duration pose a challenge for the livestream highlight detection task.

Highlight Statistics We also classify the videos based on the proportion of highlight time to total talk time to calculate the hot spot time proportion. Videos with a proportion below 5% are defined as extremely sparse, those between 5% and 10% as sparse, those between 10% and 15% as moderate, and those above 20% as dense. Figure 3(c) shows that real-time live streaming videos contain a large amount of redundant and irrelevant information, with only 8% of videos having a highlight time proportion exceeding 20%. This poses a challenge for extracting key information in live stream highlight



(a) Dataset Split (b) Record Duration Statistics (c) Highlight Statistics (d) Anchor Number Statistics

Figure 3: The statitics of *AntHighlight*. S, M, L, and XL in the record duration statistics respectively represent the short video, middle video, long video, and extra long video. XS, S, M, and D represent the extent of highlights in the entire record as extra sparse, sparse, moderate, and dense, respectively.

detection tasks.

Anchor Number As shown in Figure 3(d), we have count the number of anchors among livestreams and observed that the dataset is primarily composed of one and two speakers. However, the 49% proportion of dual anchors still pose a considerable problem of topic shifting for our task.

B Implementation Details

B.1 Data Processing

Semantic Embedding Any off-the-shelf automatic speech recognition module and language model pretrained on Chinese corpus can be used to produce semantic embeddings. In our experiments, we employ the *LC-SAN-M* proposed by (Zhang et al., 2020) to generate transcripts, which is pretrained on a 20000-hour Mandarin ASR task and finetuned in a 60-hour Mandarin corpus, and we extract the representations corresponding to the [CLS] token predicted by *Sentence-BERT* introduced in (Reimers and Gurevych, 2020) as the sentence embeddings. At the last step, we project the initial 768-d features into 256-d ones using a multi-layer perceptron.

Speaker Embedding To produce speaker embeddings, we first adopt the approach proposed by (Wang et al., 2021) to generate speaker labels for every utterance. And then, we use a 256-d lookup-table to project identification labels into continuous embeddings.

Pattern Embedding In this part, we first generate the 128-d logarithm mel-filterbanks from every utterance and downsample them into a fixed length of 8, resulting in the representations with the size of $(L, 8, 128)$. Afterwards, we concatenate them into single vectors with the length of 1024 and project them into the 256-d subspace via a multi-layer perceptron.