Cold Start Problem For Automated Live Video Comments

Hao Wu
ADAPT Centre
School of Engineering
Trinity College Dublin
Dublin, Ireland
hao.wu@adaptcentre.ie

François Pitié
ADAPT Centre
School of Engineering
Trinity College Dublin
Dublin, Ireland
pitief@tcd.ie

Gareth J. F. Jones
ADAPT Centre
School of Computing
Dublin City University
Dublin, Ireland
Gareth.Jones@dcu.ie

Abstract

Live video comments, or "danmu", are an emerging feature on Asian online video platforms. Danmu are time-synchronous comments that are overlaid on a video playback. These comments uniquely enrich the experience and engagement of their users, and have become a determining factor in the popularity of videos on these platforms. Similar to the "cold start problem" in recommender systems, a video will only start to attract attention when sufficient danmu comments have been posted on it. We study this video cold start problem and examine how new comments can be generated automatically on less-commented videos. We propose to predict danmu comments to promote user engagement, by exploiting a multi-modal combination of the video visual content, subtitles, audio signals, and any surrounding comments (when they exist). Our method fuses these multiple modalities in a transformer network which is then trained for different comment density scenarios. We evaluate our proposed system through both a retrieval based evaluation method, as well as human judgement. Results show that our proposed system improves significantly over state-of-the-art methods.

1 Introduction

Live video comments, or "danmu", is an emerging feature of video sharing platforms such as Bilibili and Nicovideo, which has been adopted by hundreds of millions of users in Asia. Danmu comments are a time-synchronous commentary subtitle system that displays user comments as streams of moving subtitles overlaid on the video playback screen (see Fig. 1). Danmu comments have become a key feature of these video platforms. So much so, that videos with many danmu comments stand a higher chance of being recommended or searched, and naturally attract more viewers.

This new form of media consumption comes with a vast amount of annotated video data and

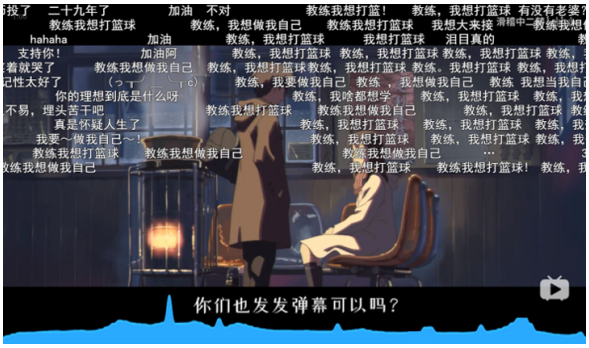

Figure 1: A video frame from bilibili.com with damnu comments overlaid. The lower part of the image shows danmu comment distribution over the video. The subtitle says: "could you publish some danmu?" and the viewers are responding with a *damnu burst*.

opens the path to multiple new research strands for video technologies, including automated highlighting, summarization and conversational engagement. The main focus of the research literature (see Section 2) has so far been on the automatic generation of danmu comments (Lv et al., 2019; Ma et al., 2019; Weiying et al., 2020). In particular, Shuming et al. (Ma et al., 2019) recently proposed in "Livebot", a new benchmark with a baseline unified transformer architecture to automatically generate new danmu comments from existing danmu comments and video content. This literature has mostly focused on the analysis of videos that already have many comments. This is however probably not the most critical scenario for automated danmu generation as these videos are already popular. Also, it is easier in these cases to exploit the numerous nearby comments to generate new comments. Similar to the "cold start problem" in recommender systems, the real issue faced by content creators is that videos need many danmu comments to start attracting traffic.

In this paper we propose to solve this "video cold start problem" by a method that can generate danmu comments on videos which have zero, few,

Proceedings of the Third Workshop on Multimodal Artificial Intelligence, pages 54–62

or many comments. We propose a multi-density cold video transformer (MCVT) that can leverage multi-modal signals including surrounding comments, video frames, but also subtitles and audio signals in an end-to-end neural network (see Section 4). The key idea is then to approach the task globally and train the network for different comment density scenarios (see Section 5). To achieve this, we collect the publishing timestamps of comments from the video platform and look at the sequence of the comment publishing times (see section 3). This allows us to consider different snapshots of a video's commenting lifetime (ie. when the video was freshly uploaded with no comments, then when it had a few comments, and later with many comments). This information has not been exploited in existing work described in the literature, but we show that it can be used effectively in training of danmu generation.

We evaluate our system in Section 6 through both a retrieval based evaluation method and human judgement. Results show that our system is able to produce comments that are close to the quality of human comments. The key contributions of this paper are as follow:

- We are the first to investigate the cold video problem for automated creation of danmu for videos which enables us to create comments for freshly uploaded videos.
- We expand a publicly available danmu video dataset (Ma et al., 2019) by doubling its size and enriching multi-modal features from video embedded subtitles.
- We propose a multi-density cold video transformer (MCVT) architecture and training framework which can generate high quality comments with different comment density and outperforms state-of-the-art method.

To make our work fully reproducible, both the source codes and the dataset used have been made public available. [1]

2 Related Work

In this section we introduce existing work on automated danmu generation, detection of video highlights based both on manually contributed danmu and atomated analysis of video content, and automated creation of descriptive captions for videos.

[1]https://github.com/fireflyHunter/Cold-Video-Danmu-Generation

2.1 Danmu Generation

The earliest work in danmu content generation was based on a generative adversarial model, where the video frames are directly mapped into the comments textual space (Lv et al., 2019). This method, however, does not exploit existing nearby comments. Ma et al. (2019) proposed *LiveBot* which combines both visual and textual contexts in an encoding phase with a Transformer architecture. They also proposed evaluation metrics and released a publicly accessible training set. This work has served as a benchmark for the most recent approaches (Zhang et al., 2020; Chaoqun et al., 2020; Weiying et al., 2020). In previous work, we reworked the baseline implementationof *LiveBot* to address several shortcomings in both the original dataset and implementation (Wu et al., 2020).

We note that *LiveBot*, and its successors, are trained on densely commented videos, and use all available comments to make predictions. Thus, they do not consider what will in practice be the more useful setting for automatec danmu creation of videos with few or no comments, which we refer to as the *cold start* scenario. Also, they do not make use of all of the attributes of the comments. In particular, the *publishing time* of the comments is not included in the training set. This means that the causality between comments is lost and that the target comments could potentially predate the proposed contextual comments. Also, these methods do not consider *where* to publish in the video timeline.

2.2 Highlight Detection

Video highlights could provide pointers for comment generation, some prior work has tried to predict popular segments in videos. Video highlights, as they are called, can be identified by looking at the current distribution of published danmu comments (see plot in Fig. 1). This is the idea exploited in (Xu et al., 2017), where a personalised frame-level recommendation is based on the analysis of published comments. More relevant to the cold start problem is highlight prediction solely from video content, as proposed in (Zheng et al., 2020) using a bi-directional Long-short Term Memory (LSTM) architecture.

2.3 Video Captioning

Related to our application is the task of video captioning, which aims to generate descriptive sen-

Statistics	Training	Dev	Test	Total
#Videos	4,272	200	200	4672
#danmu	2,549,340	123,646	116,374	2,789,360
avg. duration (s)	217	222	216	217
avg. #danmu/s	2.75	2.78	2.69	2.75

Table 1: Training, development and test sets statistics.

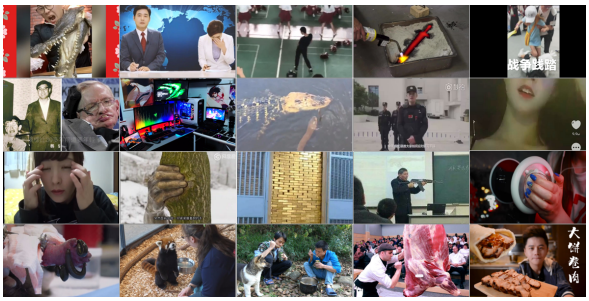

Figure 2: Examples of frames from collected videos. The video content features events from daily life.

tences of a video sequence. Current architectures for this usually follow an encoder-decoder pattern. In the encoder, the sequence of video frames is embedded by a CNN (Subhashini et al., 2014) or RNN (Nitish et al., 2015). The decoder, typically an LSTM, generates captions from the contextual output of the encoder. Techniques like reinforcement learning (Xin et al., 2018), contextual-aware video captioning (Spencer et al., 2018) and semantic attention model (Gan et al., 2017) have also been explored by researchers in this field. What emerges from the recent literature is that the Transformer architecture, as proposed in *Livebot*, has become the state-of-the-art approach for multi-modal text generation applications and thus we adopt this as the baseline for our application.

3 Task Overview

In this section we define our danmu creation task, introduce the dataset used in our work and outline the video content extraction methods used in this investigation.

3.1 Task Definition

To address the cold start problem we aim to be able to generate high quality comments given videos with different comment densities. In order to handle different danmu density scenarios of the cold start problem, we first sort the existing comments $\mathbf{C}$ for a video by their publication time and only keep a subset $\mathbf{C}_p$ consisting of a percentage p of the earliest comments of the video. This strategy is enforced to reconstruct video danmu comments in different phases of their lifetime. Then we define our task as follows: given a video $\mathbf{V} = \{\mathbf{s_0}, \ldots, \mathbf{s_L}\}$ (following accepted convention $\mathbf{V}$ is split into segments of one second duration), the generation module is asked to generate a target comment $\boldsymbol{y}$ using comments from $\mathbf{C}_p$ and the k previous seconds of the video clip $s_{[i-k,i]}$.

3.2 Dataset

For our investigation, we constructed a large-scale dataset with 4,672 videos and 2,789,360 danmu comments, which is publicly available [2]. Part of the data (2,322 videos and 857,993 comments) comes from the publicly available automatic danmu generation *Response to Livebot* dataset (Wu et al., 2020). As our task aims to generate comments for videos with low comment densities compared to a general comment creation list, the size of the suitable training data is reduced significantly during the reconstruction of the cold start scenarios . We thus added another 2,350 videos from the same danmu video website (bilibili.com) to the dataset. The Livebot dataset is mainly themed around natural life, to keep it consistent, the appended videos were selected by having a web crawler pick the 100 most popular "Daily Life" category videos of the recent three days everyday for two months. Fig. 2 presents a small subset of the video frames in this dataset. We scale up the data split in previous work (Ma et al., 2019) (2161 / 100 / 100) and have 4272 / 200 / 200 videos in the training / development / test sets, respectively. Table 1 shows damnu statistics for the dataset.

A key contribution of our paper is that we take into account the publication timestamp of each of the danmu comments. The training data for a particular level p, percentage of existing manual comments preserved, is defined as follows. Each target comment for the training set is randomly sampled from the original comment set $\mathbf{C}$ and the corresponding comment's context is defined as the 5 nearest comments from $\mathbf{C}_p$ that precede the target danmu in the video timeline. This follows the observation made in Livebot (Ma et al., 2019), that the semantic and textual similarity of comments

[2] github.com/fireflyHunter/Cold-Video-Danmu-Generation

is correlated to their timeline proximity and that the danmu context should be limited to the 5 nearest comments. We also add a *causality* constraint by applying the constraint that the comments must have been published before the target danmu in natural time.

We sample the training data for $p = 0\%, 5\%, 30\%, 50\%, 70\%$ and 100%, to form a combined training set of 4,800,145 pairs of target comment/context comments. Target comments can be sampled multiple times for different contexts.

For the 200 videos of the test set, we focus on the video highlights by only selecting 1879 comments in the most frequently commented moments in the video timeline. To study the system performance under different comment densities, we build one test set for each of the proposed values of p.

3.3 Video Information Extraction

We further augment the complete danmu commenting dataset multi-modally by extracting the audio and the subtitle information in addition to the visual and textual comment information. We believe that these additional features will help with the cold start problem.

Visual & Audio Signals. We follow standard practice by sampling one video frame per second of video. The frame from the i-th second of the video is denoted as f_i. The audio soundtrack is extracted from a video and uniformly re-sampled using a 16kHz standard.

Subtitles. We observe that human created danmu comments frequently respond to speech in the video. Fig. 1 shows an example of it: viewers are asked in the subtitles, to post danmu comments. This motivates us to transcribe the speech from the videos. Instead of using speech recognition, we opt to use optical character recognition (OCR). We found that the quality of transcripts produced by speech recognition tools was by comparison of poor quality. While most of the videos on the platform embed speech subtitles that OCR tools can accurately identify. Lastly, captions also display non-speech information which could be exploited. For OCR, we use the open-source Tesseract (Kay, 2007) OCR engine on the lower half of the sampled video frames.

Note that only 109 videos out of 4672 videos contained zero recognisable text and each video contains an average of 13.97 unique subtitles (see Fig. 3).

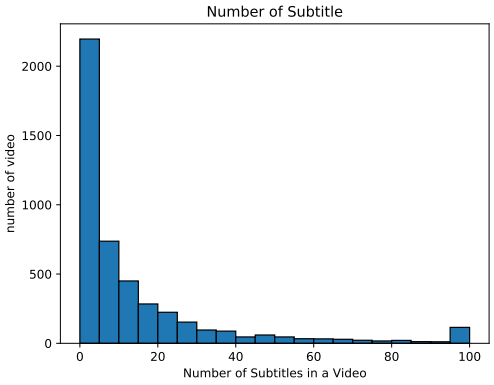

Figure 3: Histogram of the number of subtitles in the videos. For most of the videos there are less than 20 unique subtitles.

4 Network Architecture

Our proposed model, presented in Fig 4, applies standard Transformer modules with an encoder-decoder architecture. During the encoding stage, visual, audio and text features are first encoded respectively, then three transformer modules are used to fuse the information for the three modalities recursively. In the decoder, the target comment is decoded through a transformer layer with multiple multi-head attention modules that attend to three encoded multi-modal representations respectively.

4.1 Video Encoder

As in (Ma et al., 2019), video frames are encoded through a pre-trained 18-layer ResNet. We take the output from the last pooling layer of ResNet as visual feature, the frame vector of the i-th second of the video is denoted as $v_i \in \mathbb{R}^{n_{18}}$, where $n_{18} = 512$ is the size of the resulting ResNet18 features. The frame vectors in the video clip are combined as $\hat{\boldsymbol{v}}_{\boldsymbol{i}} = \{v_{i-k}, \ldots, v_i\}$.

4.2 Audio Encoder

For the audio signal, we use 20-dimensional mel-frequency cepstral coefficients (MFCCs) and another 20-dimensional MFCCs derivatives as audio frame features (Di Gangi et al., 2019). These are extracted with a Hanning window of 40 ms length and 32 ms hop size. We include all audio frames as the audio input, hence we sample 32 audio vectors for each second of the audio. The audio information at time point i is denoted as a_i^j, where j is the j-th audio frame vector in the window analysis at time i. A GRU module (Chaoqun et al., 2020) is applied to recursively encode the input audio sequence. At

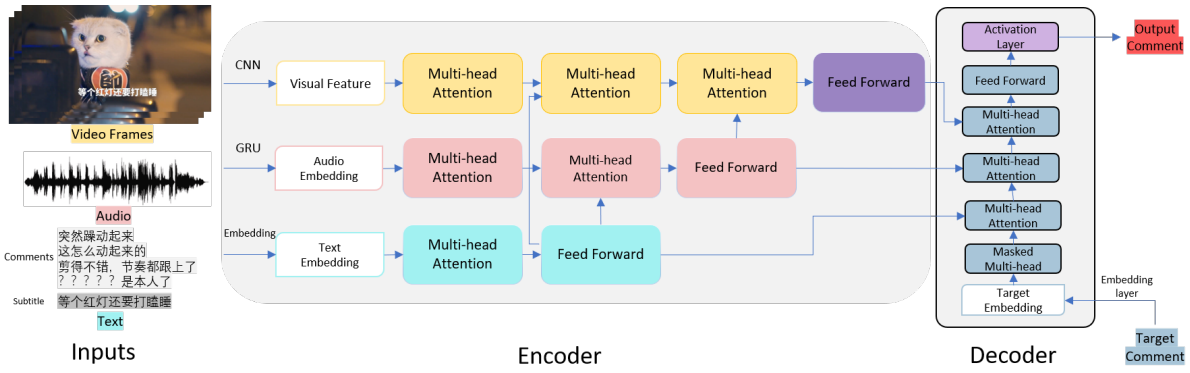

Figure 4: Architecture of the proposed model.

each stage, the current hidden state h_i^j is calculated based on the last hidden state h_i^{j-1} and the current input audio frame vector a_i^j. The sequence of hidden states h_i^j for all audio frames is concatenated into an audio encoder output $\hat{\boldsymbol{a}}_{\boldsymbol{i}} \in \mathbb{R}^{n_a \times 512}$, where $n_a = 32 \times k$ is the number of audio frames in the analysis window and 512 the dimension of the hidden state.

4.3 Text Encoder

Contextual comments are concatenated with a special delimiter token T_d inbetween each comment and then combined with the unique subtitles from the analyzed k second window. As opposed to Livebot (Ma et al., 2019), where there are always 5 context comments, in our cold start scenario we sometimes have less than 5 and even 0 comments. In the extreme case we use a special token T_n with an empty comment field to show that no context comments are available.

All unique subtitles within analysis window $s_{[i-k,i]}$ are also concatenated with the same delimiter token. Finally, we form the text input by combining comment sequence and subtitle sequence with T_d.

We remove the punctuation and segment words using Jieba (an open-sourced Chinese text segmentation tool). Each word of text input is then passed to an embedding layer of size $d \times |V|$, where d is the dimension of the word embedding and $|V|$ is the size of the vocabulary. After embedding, the text input for analysis window $s_{[i-k,i]}$, is now represented as $\hat{\boldsymbol{e}}_{\boldsymbol{i}} \in \mathbb{R}^{n \times d}$.

4.4 Fusion of Modalities

Following the success of the Transformer architecture in multi-modal processing (Ma et al., 2019; Chaoqun et al., 2020), we adopt a multi-unit Transformer module to recursively learn and combine representations from all three modalities. The Transformer unit first encodes the text input $\hat{\boldsymbol{e}}_{\boldsymbol{i}}$ into a transitional hidden state H_e. Then, a second transformer unit combines H_e and the input audio with two multi-head attention modules, the first one attending to $\hat{\boldsymbol{a}}_{\boldsymbol{i}}$ and the second one attending to H_e. Finally, another unit with three multi-head attention modules is used to summarise the video clip representation H_{vae}.

4.5 Decoder

In the model decoder, the output comment is generated through a transformer layer with 4 multi-head attention modules that attend to the target comment $\boldsymbol{y}$, text hidden state H_e, visual hidden state H_{ae} and audio hidden state H_{vae} respectively. Then the probability of output comment is produced with an softmax layer on top of the decoder output.

5 Network Training Regime

5.1 Multi-Density Learning

A key aspect of our method is to consider all the different cold start scenarios together by adopting a multi-task training strategy.

In detail, our training regime is implemented by randomly assigning, at each mini-batch, the percentage p of earlier comments that are kept from a fixed set of values $\{0\%, 5\%, 30\%, 50\%, 70\%, 100\%\}$. Recall that $p = 0\%$ corresponds to the cold start problem, and $p = 100\%$ corresponds to the situation where all other comments are available (such as in Livebot (Ma et al., 2019)). By alternating between these values of p, we are able to train the network for both the cold start and Livebot scenario.

5.2 Training Detail

The video analysis window size k is set to 5 (s). For the text input, we build the vocabulary by selecting the most frequent 50,000 words in the dataset and set the max length of the input text sequence to 50. In the model, the text embedding is of size 512 and is randomly initialized before training. The dimension of the audio's GRU hidden state is set to 512. We apply the same setting for all transformer components used in the network. For each transformer, the hidden state dimension is set to 512, the feed forward network dimension is 2048, the number of heads is 8 and the number of blocks is 6. The loss criterion is cross-entropy. The number of epochs is set to 10, the batch size to 64 and we use the Adam optimizer (Kingma and Ba, 2014) with settings β_1 = 0.9, β_2 = 0.998, weight decay =1×10^{-4}, $\epsilon = 1\times10^{-8}$ and learning rate 1×10^{-4}. All training was done on a Linux server with a single RTX 2080 Ti graphic card, 16 cores Intel(R) Xeon(R) CPU E5-2623 v4 @ 2.60GHz and 256GB RAM. The model is implemented using Pytorch 1.4.0 and Python 3.6. With above settings, it takes around 34 hours to complete the training.

6 Experiments

In this section we report results for our investigation of comment generation. We use the Livebot model (Ma et al., 2019) as a baseline. Specifically, we use the code from (Wu et al., 2020), trained on our full dataset with only video frames and surrounding comments as input. The models proposed in (Chaoqun et al., 2020; Zhang et al., 2020) are very recent and their code is not publicly available yet, so we do not consider these as one of our baseline methods. Other older neural architectures such as LSTM are also not included in this study since it is well established that Transformers are the method of choice for modelling multi-modal signals.

6.1 Evaluation

We note that reference-based metrics for generation tasks like BLEU and ROUGE are not suitable for evaluation of video comments (Das et al., 2017; Ma et al., 2019; Zhang et al., 2020). Hence we follow (Das et al., 2017) and focus on the ability to rank the correct comment originally appearing at this point in the video over other comments taken from the dataset. We evaluate our system through a retrieval based protocol: the model is asked to re-rank a candidate set for each test sample. The comment set for re-ranking is made of 100 comments, including 5 correct groundtruth comments for this point in the video, the 20 most similar comments to the title of the video based on tf-idf score (plausible candidates), the 20 most frequent comments in the dataset and 55 randomly sampled comments.

We report the Recall@k, Precision@k, Mean Rank (MR) and Mean Reciprocal Rank (MRR) as evaluation metrics on this retrieval task. The confidence interval is reported for each of these metrics with confidence level at 95% (for R@k, we use the confidence interval for population proportions).

6.2 Ablation Study

The retrieval task results are reported in Table 3 and Figure 5. In this ablation study, we compare 4 variants of the model.

- **Livebot** (Ma et al., 2019) leverages textual and visual information in a Transformer architecture. It is trained on the extended dataset using the implementation provided in (Wu et al., 2020). The training is done here with p=100%.
- **Livebot-t** applies the same network architecture as textbfLivebot, but is trained with our multi-density training strategy to evaluate the effectiveness of our proposed training regime.
- **MCVT** is the final system proposed in this work, which includes the training regime and the inclusion of the additional audio and subtitle features.
- **MCVT-Zero** is listed to further examine the performance limit in the cold start scenario, i.e. we assume a situation where no comments are present. Thus, we train the MCVT network uniquely on the cold start scenario for $p = 0\%$

The results in Table 3 show that **Livebot-t** outperforms the baseline **Livebot** model in most cases, and thus demonstrates the effectiveness of our training strategy. One exception is found when $p = 100\%$, the **Livebot** model, trained only with densely commented videos, slightly outscores **Livebot-t**, we think this means the information learned from multi-density training strategy produces extra noise when the model only aims to

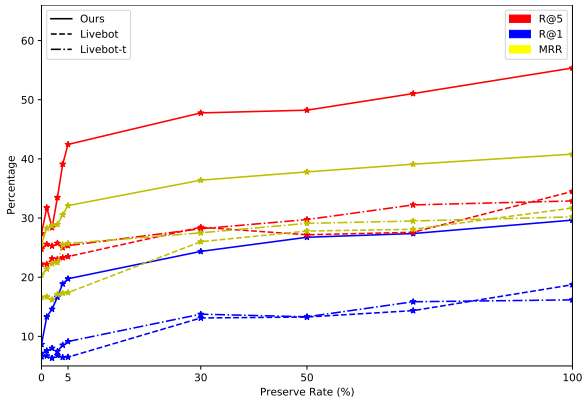

Figure 5: Model performance for R@5, R@1, MRR, at different comment densities p (see Table 3).

generate comments for popular videos. By contrast, from the third and fourth rows of Table 3, we can see that our **MCVT** model has similar performance to **MCVT-zero**, which has been trained specifically for the complete cold start scenario. In this situation, the extra knowledge gained from learning popular videos does not appear to affect the performance in the cold start situation. This comparison between the behaviour of the **Livebot** and **MCVT** systems potentially demonstrates the advantage of our training regime in the case of cold start scenario.

We also see that our model outperforms **Livebot-t** in every scenario, which also supports the idea that integrating the audio signal and subtitle in the generation system can significantly improve the performance of the model.

6.3 Human Evaluation

Additionally, we also use human judgements to obtain a more intuitive and reliable measurement of the generated comments. A subset of 50 videos was randomly sampled from the 200 videos of the test set. Three native Chinese speakers familiar with danmu were asked to rate the quality of the generated comments on three criteria: fluency, relevancy and engagement.

- **Fluency** is intended to measure the language quality of the generated comment.
- **Relevancy** measures the semantic relevancy between the generated comment and the input video and nearby comments.
- **Engagement** should reflect how likely it is that the generated comment will motivate others to respond.

Model	p	Fluency	Relevance	Engagement
MCVT	0%	4.25	3.17	2.76
MCVT	5%	4.33	3.36	2.99
MCVT	50%	4.59	3.78	**3.07**
MCVT	100%	4.47	**3.91**	2.97
Human	-	**4.79**	3.58	3.01

Table 2: Human evaluation on 50 videos from the test set. Each comment is graded between 1 and 5, by 3 reviewers, for their language fluency, relevance to the video content and on how likely they are to provoke other viewers to also comment.

The score for all 3 measurements ranges from 1 (poor) to 5 (excellent). The final score is the average of the scores of the three annotators. The evaluation was conducted on the comments generated by our method for $p \in \{0\%, 5\%, 50\%\}$. For reference, we also evaluate the groundtruth comment set for these videos.

Table 2 reports the results of this human evaluation. We can see that the overall performance of model is almost indistinguishable from real danmu comments. Our relevancy and engagement scores are actually higher when $p \geq 50\%$. The quality of our model degrades slightly for the complete cold start scenario, but the results are still quite close to human comments.

6.4 Case Study

Examples of predicted outputs are shown in Fig. 6. The corresponding video frame shows a groundhog being fed. The subtitle, context comment, generated comments and target comments are reported in the table to the right. We can see that the model generates reasonable comments, which are relevant to the video shot and match the video's positive emotion (e.g. "laugh", "hahaha" and "lol"), even in the case of a complete cold start.

7 Conclusions and Further Development

In this paper we investigate the cold video start problem in automated danmu comment generation. We propose a multi-modal fusion network which includes processing of video frames, already published comments, and also audio and caption text. We train it for different comment density scenarios and perform extensive experiments on an expanded danmu video dataset. Results demonstrate the advantage of our method over the state-of-the-art in solving the cold video start problem.

Table 3: Results of comment generation module, model performance is presented with metrics of R@k, P@k, MRR (higher is better, showed in percentage) and MR (lower is better), p is the percentage of the preserved comments applied in test set.

Model	p	R@1	R@5	R@10	MR	MRR	P@5	P@10
Livebot	0 %	6.56 ± 0.05	22.23 ± 0.22	31.36 ± 0.29	22.15 ± 0.37	16.6 ± 0.48	6.44 ± 0.18	6.58 ± 0.18
Livebot-t	0 %	7.09 ± 0.06	24.78 ± 0.23	37.77 ± 0.36	19.86 ± 0.46	20.4 ± 0.48	6.89 ± 0.18	8.02 ± 0.18
MCVT-zero	0 %	**8.79** ± 0.07	27.25 ± 0.25	45.58 ± 0.44	**18.28** ± 0.33	25.6 ± 0.51	8.45 ± 0.18	**8.85** ± 0.20
MCVT	0 %	8.65 ± 0.07	**27.36** ± 0.25	**47.90** ± 0.44	18.81 ± 0.33	**25.8** ± 0.52	**8.70** ± 0.19	8.68 ± 0.19
Livebot	5 %	6.49 ± 0.05	23.49 ± 0.22	32.88 ± 0.31	21.59 ± 0.34	17.4 ± 0.48	6.15 ± 0.19	6.74 ± 0.18
Livebot-t	5 %	9.13 ± 0.08	25.34 ± 0.23	39.40 ± 0.38	19.51 ± 0.34	25.7 ± 0.48	8.90 ± 0.21	8.59 ± 0.21
MCVT	5 %	**19.74** ± 0.18	**42.44** ± 0.4	**56.70** ± 0.55	**12.90** ± 0.35	**32.1** ± 0.64	**18.75** ± 0.36	**19.11** ± 0.38
Livebot	30 %	13.11 ± 0.13	28.45 ± 0.27	41.50 ± 0.40	19.93 ± 0.37	26.0 ± 0.47	12.88 ± 0.24	11.59 ± 0.24
Livebot-t	30 %	13.75 ± 0.13	28.19 ± 0.27	45.59 ± 0.44	18.71 ± 0.35	27.5 ± 0.48	13.14 ± 0.27	13.07 ± 0.27
MCVT	30 %	**24.36** ± 0.22	**47.77** ± 0.46	**61.38** ± 0.59	**11.87** ± 0.31	**36.4** ± 0.59	**24.85** ± 0.41	**24.15** ± 0.42
Livebot	50 %	13.27 ± 0.12	27.17 ± 0.26	41.98 ± 0.40	20.44 ± 0.37	27.8 ± 0.44	13.37 ± 0.29	13.09 ± 0.27
Livebot-t	50 %	13.31 ± 0.12	29.74 ± 0.29	47.07 ± 0.46	18.39 ± 0.34	29.1 ± 0.51	15.59 ± 0.31	16.23 ± 0.32
MCVT	50 %	**26.75** ± 0.25	**48.23** ± 0.46	**62.57** ± 0.60	**11.23** ± 0.29	**37.8** ± 0.67	**26.17** ± 0.42	**26.89** ± 0.42
Livebot	70 %	14.35 ± 0.14	27.59 ± 0.26	42.09 ± 0.41	19.13 ± 0.36	28.1 ± 0.48	15.15 ± 0.34	14.76 ± 0.34
Livebot-t	70 %	15.85 ± 0.14	32.22 ± 0.31	55.44 ± 0.53	18.11 ± 0.36	29.5 ± 0.48	16.77 ± 0.35	17.01 ± 0.35
MCVT	70 %	**27.38** ± 0.25	**51.04** ± 0.49	**63.21** ± 0.61	**11.10** ± 0.27	**39.1** ± 0.71	**28.25** ± 0.43	**27.65** ± 0.42
Livebot	100 %	18.83 ± 0.16	34.50 ± 0.33	52.17 ± 0.51	17.81 ± 0.36	34.7 ± 0.48	18.88 ± 0.36	18.31 ± 0.36
Livebot-t	100 %	17.17 ± 0.15	32.89 ± 0.31	52.91 ± 0.51	18.09 ± 0.36	33.2 ± 0.48	18.15 ± 0.36	18.11 ± 0.36
MCVT	100 %	**29.65** ± 0.28	**55.36** ± 0.53	**63.90**± 0.62	**10.81** ± 0.29	**40.8** ± 0.65	**29.79** ± 0.43	**29.82** ± 0.43

P	Context comment	Output	Target Comments
0%	-	吃土拨鼠2333333 Eating groundhog lol	兔子有时候也有这种状态 Rabbits sometimes will behave like this 这两只打架受内伤了，要不我们…… These two were injured internally in the fight, should we... 土八鼠听着蛮可爱的 The groundhogsounds cute
5%	都是老鼠，待遇差别真大 They are all rats, the treatment is really different. 这是熊吧？ Is this a bear?	我看老鼠要笑到缺氧 I think the mouse is going to laugh until hypoxia	
30%	老子，在吃饼干！ I'm eating cookies! 这个土拨鼠中暑了活不了了 This groundhog can't live after heatstroke. 都是老鼠，待遇差别真大 They are all rats, the treatment is really different.	我看老鼠要笑到缺氧 I think the mouse is going to laugh until hypoxia	
100%	这个土拨鼠很漂亮的 This groundhog is very beautiful 吃了你们就不给了 You won't give it if I eat it. 土拨鼠哈哈哈 Groundhog hahaha 老子，在吃饼干！ I'm eating cookies! 都是老鼠，待遇差别真大 They are all rats, the treatment is really different.	被土拨鼠洗脑了哈哈哈哈哈哈 Brainwashed by groundhog hahahahahaha	

Figure 6: An example from the test set, left side is the video frame and the subtitle translation of the time point. The table on the right shows the target comments, context comments and the generated comment in different preserve rate p.

Our next research goal is to leverage a highlight detection method in this task to seek to further improve the system performance, since this is expected to reveal areas of likely user interest on the video timeline which could provide pointers for preferred locations for the automated creation of danmu comments.

Acknowledgement

This work was supported by Science Foundation Ireland as part of the ADAPT Centre (Grant 13/RC/2106) (www.adaptcentre.ie) at Trinity College Dublin.

References

D. Chaoqun, C. Lei, M. Shuming, W. Furu, Z. Conghui, and Z. Tiejun. 2020. Multimodal matching transformer for live commenting. In *ECAI*.

Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José MF Moura, Devi Parikh, and Dhruv Batra. 2017. Visual dialog. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 326–335.

Mattia Antonino Di Gangi, Matteo Negri, Roldano Cattoni, Dessi Roberto, and Marco Turchi. 2019. Enhancing transformer for end-to-end speech-to-text translation. In *Machine Translation Summit XVII*, pages 21–31. European Association for Machine Translation.

Z. Gan, C. Gan, X. He, Y. Pu, K. Tran, J. Gao, L. Carin, and D. Li. 2017. Semantic compositional networks for visual captioning. In *CVPR*.

Anthony Kay. 2007. Tesseract: an open-source optical character recognition engine. *Linux Journal*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *ICLR*.

G. Lv, T. Xu, Q. Liu, E. Chen, W. He, M. An, and Z .Chen. 2019. Gossiping the videos: An embedding-based generative adversarial framework for time-sync comments generation. In *Springer*.

S. Ma, L. Cui, D. Dai, F. Wei, and X. Sun. 2019. Livebot: Generating live video comments based on visual and textual contexts. In *AAAI*.

S. Nitish, M. Elman, and S. Ruslan. 2015. Unsupervised learning of video representations using lstms. In *ICML*.

W. Spencer, J. Heng, B. Mohit, C. Shih-Fu, and V. Clare. 2018. Incorporating background knowledge into video description generation. In *EMNLP*.

V. Subhashini, H. Xu, D. Jeff, R. Marcus, M. Raymond, and S. Kate. 2014. Translating videos to natural language using deep recurrent neural networks. *arXiv*.

W. Weiying, C. Jieting, and J. Qin. 2020. Videoic: A video interactive comments dataset and multimodal multitask learning for comments generation. In *ACMMM*.

H. Wu, G. J. F. Jones, and F. Pitié. 2020. Response to livebot: Generating live video comments based on visual and textual contexts. *arXiv*.

W. Xin, C. Wenhu, J. Wu, Y. Wang, and Y. William. 2018. Video captioning via hierarchical reinforcement learning. In *CVPR*.

C. Xu, Z. Yongfeng, A. Qingyao, X. Hongteng, Y. Junchi, and Q. Zheng. 2017. Personalized key frame recommendation. In *SIGIR*, pages 315–324.

Z. Zhang, Z. Yin, S. Ren, X. Li, and S. Li. 2020. Dca: Diversified co-attention towards informative live video commenting. In *CCF NLPCC*.

W. Zheng, Z. Jie, M. Jing, L. Jingjing, A. Jiangbo, and Y. Yang. 2020. Discovering attractive segments in the user-generated video streams. *Information Processing & Management*.