

LiveBot: Generating Live Video Comments Based on Visual and Textual Contexts

Shuming Ma,^{1,*} Lei Cui,² Damai Dai,¹ Furu Wei,² Xu Sun¹

¹MOE Key Lab of Computational Linguistics, School of EECS, Peking University

²Microsoft Research Asia

{shumingma,daidamai,xusun}@pku.edu.cn
{lecu,fuwei}@microsoft.com

Abstract

We introduce the task of automatic live commenting. Live commenting, which is also called “video barrage”, is an emerging feature on online video sites that allows real-time comments from viewers to fly across the screen like bullets or roll at the right side of the screen. The live comments are a mixture of opinions for the video and the chit chats with other comments. Automatic live commenting requires AI agents to comprehend the videos and interact with human viewers who also make the comments, so it is a good testbed of an AI agent’s ability to deal with both dynamic vision and language. In this work, we construct a large-scale live comment dataset with 2,361 videos and 895,929 live comments. Then, we introduce two neural models to generate live comments based on the visual and textual contexts, which achieve better performance than previous neural baselines such as the sequence-to-sequence model. Finally, we provide a retrieval-based evaluation protocol for automatic live commenting where the model is asked to sort a set of candidate comments based on the log-likelihood score, and evaluated on metrics such as mean-reciprocal-rank. Putting it all together, we demonstrate the first “LiveBot”. The datasets and the codes can be found at <https://github.com/lancopku/livebot>.

Introduction

The comments of videos bring many viewers fun and new ideas. Unfortunately, on many occasions, the videos and the comments are separated, forcing viewers to make a trade-off between the two key elements. To address this problem, some video sites provide a new feature: viewers can put down the comments during watching videos, and the comments will fly across the screen like bullets or roll at the right side of the screen. We show an example of live comments in Figure 2. The video is about drawing a girl, and the viewers share their watching experience and opinions with the live comments, such as “simply hauntingly beautiful”. The live comments make the video more interesting and appealing. Besides, live comments can also better engage viewers and create a direct link among viewers, making their opinions and responses more visible than the average comments in the comment section. These features have a tremendously

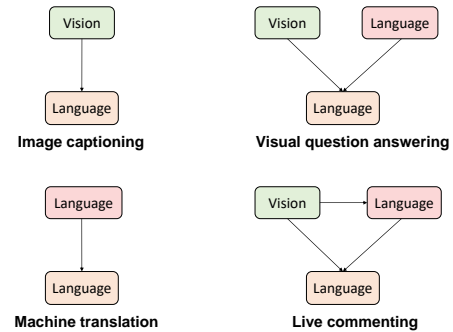


Figure 1: The relationship between vision and language in different tasks.

positive effect on the number of users, video clicks, and usage duration.

Motivated by the advantages of live comments for the videos, we propose a novel task: automatic live commenting. The live comments are a mixture of the opinions for the video and the chit chats with other comments, so the task of living commenting requires AI agents to comprehend the videos and interact with human viewers who also make the comments. Therefore, it is a good testbed of an AI agent’s ability to deal with both dynamic vision and language.

With the rapid progress at the intersection of vision and language, there are some tasks to evaluate an AI’s ability of dealing with both vision and language, including image captioning (Donahue et al. 2017; Fang et al. 2015; Karpathy and Fei-Fei 2017), video description (Rohrbach et al. 2015; Venugopalan et al. 2015a; 2015b), visual question answering (Agrawal, Batra, and Parikh 2016; Antol et al. 2015), and visual dialogue (Das et al. 2017). Live commenting is different from all these tasks. Image captioning is to generate the textual description of an image, and video description aims to assign a description for a video. Both the two tasks only require the machine to see and understand the images or the videos, rather than communicate with the human. Visual question answering and visual dialogue take a significant step towards human-machine interaction. Given an image, the machine should answer questions about the image or conduct multiple rounds of a dialogue with the human. Different from the two tasks, live commenting requires to

*Joint work between Microsoft Research Asia and Peking University

Copyright © 2019, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

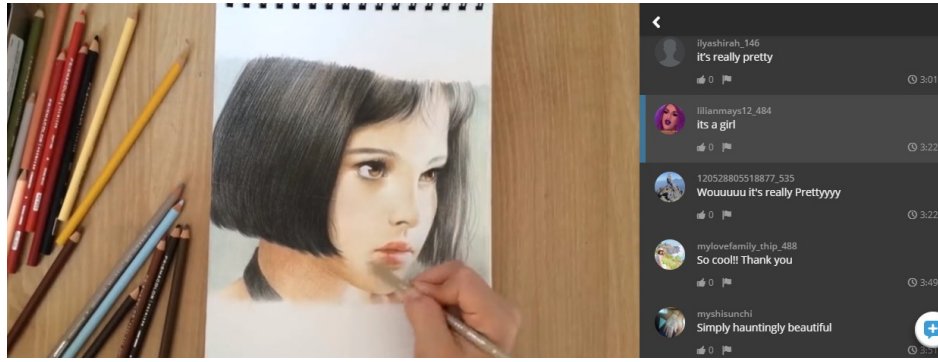


Figure 2: An example of live comments from a video streaming website ViKi.

understand the videos and share the opinions or watching experiences, which is a more challenging task.

A unique challenge of automatic live commenting is the complex dependency between the comments and the videos. First, the live comments are related to both the videos and the surrounding comments, and the surrounding comments also depend on the videos. We summarize the comparison of the dependency between live commenting and other tasks in Figure 1. Second, the live comments are not only conditioned on the corresponding frames that they appear on but also the surrounding frames, because the viewers may comment on either the upcoming video streaming¹ or the past. More specifically, we formulate the live commenting task as: given a video V , one frame in the video f , the time-stamp t of the frame, and the surrounding comments C (if any) and frames I at around the time-stamp, the machine should make a comment relevant to the clips or the other comments near the frame f .

In this work, we build a “LiveBot” to make live comments for the videos. We construct a large-scale live comment dataset with 2,361 videos and 895,929 comments from a popular Chinese video streaming website called Bilibili. In order to model the complex dependency described above, we introduce two neural approaches to generate comments. We also provide a retrieval-based evaluation protocol for live commenting where the model is asked to sort a set of candidate answers based on the log-likelihood score, and evaluated on metrics such as mean-reciprocal-rank. Experimental results show that our model can achieve better performance than the previous neural baselines in both automatic evaluation and human evaluation.

The contributions of the paper are as follow:

- To the best of our knowledge, we are the first to propose the task of automatic live commenting for videos.
- We construct a large-scale live comment dataset with 2,361 videos and 895,929 comments, so that the data-driven approaches are possible for this task.
- We introduce two neural models to jointly encode the visual content and the textual content, which achieve better

¹For example, some viewers will turn back the videos and put down the warning of upcoming surprising scenes.

performance than the previous neural baselines such as the sequence-to-sequence model.

- We provide a retrieval-based evaluation protocol for live commenting where the model is asked to sort a set of candidate answers and evaluated on metrics such as mean-reciprocal-rank.

The Live Comment Dataset

In this section, we introduce our proposed Live Comment Dataset. We first describe how we collect the data and split the dataset. Then we analyze the properties of live comments.

Collection of Videos and Live Comments

Here, we describe the Live Comment Dataset. The videos are collected from a popular Chinese video streaming website called Bilibili. In order to collect the representative videos, we obtain the top representative queries from the search engine, and crawl the top 10 pages of the video search results. The queries cover 19 categories, including pets, sports, animation, food, entertainment, technology and more. We remove the duplicate and short videos, and filter the videos with low quality or few live comments to maintain the data quality. As a result, we have 2,361 videos of high quality in total.

On the video website, the live comments are naturally paired with the videos. For each video, we collect all the live comments appeared in the videos. We also crawl the time-stamps when the comments appear, so that we can determine the background (surrounding frames and comments) of the given comments. We tokenize all comments with the popular python package Jieba. As a result, we have 895,929 live comments paired with the videos and the time-stamps. We also download the audio channels of the videos. We find it intractable to align the segment of audio with the comments. Therefore, we do not segment the audio, and reserve the entire audio for the videos.

Table 3 shows an example of our data. The pictures above are three selected frames to demonstrate a video about feeding cats. The table below includes several selected live comments with the corresponding time-stamps. It shows that the live comments are related to the frames where the comments



Figure 3: A data example of a video paired with selected live comments in the Live Comment Dataset. Above are three selected frames from the videos to demonstrate the content. Below is several selected live comments paired with the time stamps when the comments appear on the screen.

appear. For example, the video describes an orange cat fed with the catmint at 0:48, while the live comment at the same frame is asking “is the orange cat short leg?”. The comment is also related to the surrounding frames. For example, the video introduces three cats playing on the floor at 1:52, while the live comment at 1:56 is saying “So many kittens!”. Moreover, the comment is related to the surrounding comments. For example, the comment at 2:39 asks “whether the catmint works for the tiger”, and the comment at 2:41 responds that “catmint also works for the tiger”.

Dataset Split

To split the dataset into training, development and testing sets, we separate the live comments according to the corresponding videos. The comments from the same videos will not appear solely in the training or testing set to avoid overfitting. We split the data into 2,161, 100 and 100 videos in the training, testing and development sets, respectively. Finally, the numbers of live comments are 818,905, 42,405, and 34,609 in the training, testing, development sets. Table 1 presents some statistics for each part of the dataset.

Data Statistics

Table 2 lists the statistics and comparison among different datasets and tasks. We will release more data in the future. Our Bilibili dataset is among the large-scale dataset in terms of videos (2,361) and sentences (895,929). YouCook (Das et al. 2013), TACos-M-L (Rohrbach et al. 2014) are two popular action description datasets, which focus on the cooking domain. M-VAD (Torabi et al. 2015) and MPII-MD (Rohrbach et al. 2015) are the movie description datasets, while MovieQA (Tapaswi et al. 2016) is a popular movie question answering dataset. MSVD (Chen and Dolan

Statistic	Train	Test	Dev	Total
#Video	2,161	100	100	2,361
#Comment	818,905	42,405	34,609	895,929
#Word	4,418,601	248,399	193,246	4,860,246
Avg. Words	5.39	5.85	5.58	5.42
Duration (hrs)	103.81	5.02	5.01	113.84

Table 1: Statistics information on the training, testing, and development sets.

2011) is a dataset for the task of paraphrasing, and MSR-VTT (Xu et al. 2016) is for video captioning. A major limitation for these datasets is limited domains (i.e. cooking and movie) and small size of data (in terms of videos and sentences). Compared with these datasets, our Bilibili dataset is derived from a wide variety of video categories (19 categories), which can benefit the generalization capability of model learning. In addition, the previous datasets are designed for the tasks of description, question answering, paraphrasing, and captioning, where the patterns between the videos and the language are clear and obvious. Our dataset is for the task of commenting, where the patterns and relationship between the videos and the language are latent and difficult to learn. In summary, our Bilibili dataset represents one of the most comprehensive, diverse, and complex datasets for video-to-text learning.

Analysis of Live Comments

Here, we analyze the live comments in our dataset. We demonstrate some properties of the live comments.

Distribution of Lengths Figure 4 shows the distribution of the lengths for live comments in the training set. We

Dataset	Task	#Video	#Sentence
YouCook	Action Description	88	2,668
TACos-M-L	Action Description	185	52,593
M-VAD	Movie Description	92	52,593
MPII-MD	Movie Description	94	68,375
MovieQA	Question Answering	140	150,000
MSVD	Paraphrasing	1,970	70,028
MSR-VTT	Captioning	7,180	200,000
Bilibili	Commenting	2,361	895,929

Table 2: Comparison of different video-to-text datasets.

Interval	Edit Distance	TF-IDF	Human
0-1s	11.74	0.048	4.3
1-3s	11.79	0.033	4.1
3-5s	12.05	0.028	3.9
5-10s	12.42	0.025	3.1
>10s	12.26	0.015	2.2

Table 3: The average similarity between two comments at different intervals (Edit distance: lower is more similar; Tf-idf: higher is more similar; Human: higher is more similar).

can see that most live comments consist of no more than 5 words or 10 characters. One reason is that 5 Chinese words or 10 Chinese characters have contained enough information to communicate with others. The other reason is that the viewers often make the live comments during watching the videos, so they prefer to make short and quick comments rather than spend much time making long and detailed comments.

Correlation between Neighboring Comments We also validate the correlation between the neighboring comments. For each comment, we select its 20 neighboring comments to form 20 comment pairs. Then, we calculate the sentence similarities of these pairs in terms of three metrics, which are edit distance, tf-idf, and human scoring. For human scoring, we ask three annotators to score the semantic relevance between two comments, and the score ranges from 1 to 5. We group the comment pairs according to their time intervals: 0-1s, 1-3s, 3-5s, 5-10s, and more than 10s. We take the average of the scores of all comment pairs, and the results are summarized in Table 3. It shows that the comments with a larger interval are less similar both literally and semantically. Therefore, it concludes that the neighboring comments have higher correlation than the non-neighboring ones.

Approaches to Live Commenting

A challenge of making live comments is the complex dependency between the comments and the videos. The comments are related to the surrounding comments and video clips, and the surrounding comments also rely on the videos. To model this dependency, we introduce two approaches to generate the comments based on the visual contexts (surrounding frames) and the textual contexts (surrounding comments). The two approaches are based on two popular architectures for text generation: recurrent neural network (RNN) and transformer. We denote two approaches as Fusional RNN

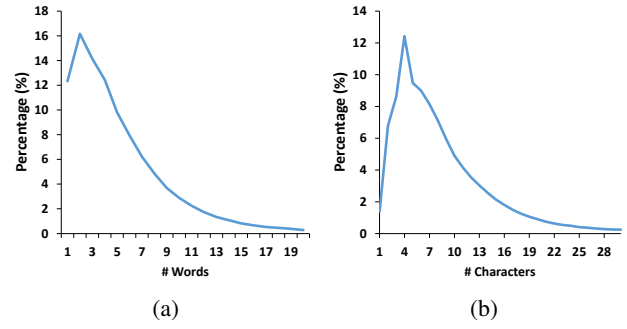


Figure 4: Distribution of lengths for comments in terms of both word-level and character-level.

Model and Unified Transformer Model, respectively.

Problem Formulation

Here, we provide the problem formulation and some notations. Given a video V , a time-stamp t , and the surrounding comments C near the time-stamp (if any), the model should generate a comment y relevant to the clips or the other comments near the time-stamp. Since the video is often long and there are sometimes many comments, it is impossible to take the whole videos and all the comments as input. Therefore, we reserve the nearest m frames² and n comments from the time-stamp t . We denote the m frames as $I = \{I_1, I_2, \dots, I_m\}$, and we concatenate the n comments as $C = \{C_1, C_2, \dots, C_n\}$. The model aims at generating a comment $y = \{y_1, y_2, \dots, y_k\}$, where k is the number of words in the sentence.

Model I: Fusional RNN Model

Figure 5 shows the architecture of the Fusional RNN model. The model is composed of three parts: a video encoder, a text encoder, and a comment decoder. The video encoder encodes m consecutive frames with an LSTM layer on the top of the CNN layer, and the text encoder encodes n surrounding live comments into the vectors with an LSTM layer. Finally, the comment decoder generates the live comment.

Video Encoder In the video encoding part, each frame I_i is first encoded into a vector v_i by a convolution layer. We then use an LSTM layer to encode all the frame vectors into their hidden states h_i :

$$v_i = CNN(I_i) \quad (1)$$

$$h_i = LSTM(v_i, h_{i-1}) \quad (2)$$

Text Encoder In the comment encoding part, each surrounding comment C_i is first encoded into a series of word-level representations, using a word-level LSTM layer:

$$r_i^j = LSTM(C_i^j, r_i^{j-1}) \quad (3)$$

We use the last hidden state $r_i^{L(i)}$ as the representation for C_i denoted as x_i . Then we use a sentence-level LSTM layer

²We set the interval between frames as 1 second.

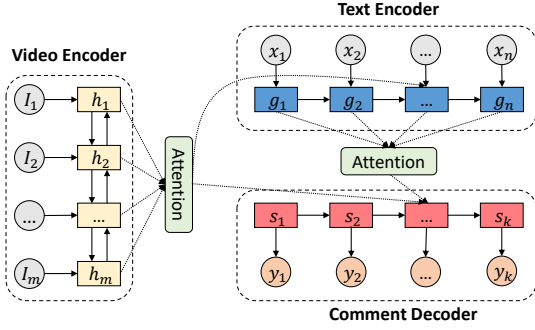


Figure 5: An illustration of Fusional RNN Model.

with the attention mechanism to encode all the comments into sentence-level representation g_i :

$$\hat{g}_i = LSTM(x_i, g_{i-1}) \quad (4)$$

$$g_i = Attention(\hat{g}_i, \mathbf{h}) \quad (5)$$

With the help of attention, the comment representation contains the information from videos.

Comment Decoder The model generates the comment based on both the surrounding comments and frames. Therefore, the probability of generating a sentence is defined as:

$$p(y_0, \dots, y_T | \mathbf{h}, \mathbf{g}) = \prod_{t=1}^T p(y_t | y_0, \dots, y_{t-1}, \mathbf{h}, \mathbf{g}) \quad (6)$$

More specifically, the probability distribution of word w_i is calculated as follows,

$$\hat{s}_i = LSTM(y_{i-1}, s_{i-1}) \quad (7)$$

$$s_i = Attention(\hat{s}_i, \mathbf{h}, \mathbf{g}) \quad (8)$$

$$p(w_i | w_0, \dots, w_{i-1}, \mathbf{h}) = Softmax(Ws_i) \quad (9)$$

Model II: Unified Transformer Model

Different from the hierarchical structure of Fusional RNN model, the unified transformer model uses a linear structure to capture the dependency between the comments and the videos. Similar to Fusional RNN model, the unified transformer model consists of three parts: the video encoder, the text encoder, and the comment decoder. Figure 6 shows the architecture of the unified transformer model. In this part, we omit the details of the inner computation of the transformer block, and refer the readers to the related work (Vaswani et al. 2017).

Video Encoder Similar to Fusional RNN model, the video encoder first encodes each frame I_i into a vector v_i with a convolution layer. Then, it uses a transformer layer to encode all the frame vectors into the final representation h_i :

$$v_i = CNN(I_i) \quad (10)$$

$$h_i = Transformer(v_i, \mathbf{v}) \quad (11)$$

Inside the transformer, each frame's representation v_i attends to a collection of the other representations $\mathbf{v} = \{v_1, v_2, \dots, v_m\}$.

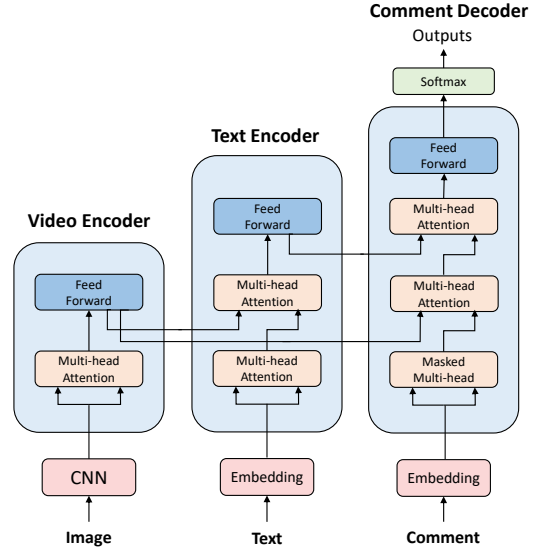


Figure 6: An illustration of Unified Transformer Model.

Text Encoder Different from Fusion RNN model, we concatenate the comments into a word sequence $e = \{e_1, e_2, \dots, e_L\}$ as the input of text encoder, so that each words can contribute to the self-attention component directly. The representation of each words in the comments can be written as:

$$g_i = Transformer(e_i, \mathbf{e}, \mathbf{h}) \quad (12)$$

Inside the text encoder, there are two multi-head attention components, where the first one attends to the text input e and the second attends to the outputs \mathbf{h} of video encoder.

Comment Decoder We use a transformer layer to generate the live comment. The probability of generating a sentence is defined as:

$$p(y_0, \dots, y_T | \mathbf{h}, \mathbf{g}) = \prod_{t=1}^T p(y_t | w_0, \dots, w_{t-1}, \mathbf{h}, \mathbf{g}) \quad (13)$$

More specifically, the probability distribution of word y_i is calculated as:

$$s_i = Transformer(y_i, \mathbf{y}, \mathbf{h}, \mathbf{g}) \quad (14)$$

$$p(y_i | y_0, \dots, y_{i-1}, \mathbf{h}, \mathbf{g}) = Softmax(Ws_i) \quad (15)$$

Inside the comment decoder, there are three multi-head attention components, where the first one attends to the comment input \mathbf{y} and the last two attend to the outputs of video encoder \mathbf{h} and text encoder \mathbf{g} , respectively.

Evaluation Metrics

The comments can be various for a video, and it is intractable to find out all the possible references to be compared with the model outputs. Therefore, most of the reference-based metrics for generation tasks like BLEU and ROUGE are not appropriate to evaluate the comments.

	Model	#I	#C	Recall@1	Recall@5	Recall@10	MR	MRR
Video Only	S2S-I	5	0	4.69	19.93	36.46	21.60	0.1451
	S2S-IC	5	0	5.49	20.71	38.35	20.15	0.1556
	Fusional RNN	5	0	10.05	31.15	48.12	19.53	0.2217
	Unified Transformer	5	0	11.40	32.62	50.47	18.12	0.2311
Comment Only	S2S-C	0	5	9.12	28.05	44.26	19.76	0.2013
	S2S-IC	0	5	10.45	30.91	46.84	18.06	0.2194
	Fusional RNN	0	5	13.15	34.71	52.10	17.51	0.2487
	Unified Transformer	0	5	13.95	34.57	51.57	17.01	0.2513
Both	S2S-IC	5	5	12.89	33.78	50.29	17.05	0.2454
	Fusional RNN	5	5	17.25	37.96	56.10	16.14	0.2710
	Unified Transformer	5	5	18.01	38.12	55.78	16.01	0.2753

Table 4: The performance of the baseline models and the proposed models. (#I: the number of input frames used at the testing stage; #C: the number of input comments used at the testing stage; Recall@k, MRR: higher is better; MR: lower is better)

Model	Fluency	Relevance	Correctness
S2S-IC	4.07	2.23	2.91
Fusion	4.45	2.95	3.34
Transformer	4.31	3.07	3.45
Human	4.82	3.31	4.11

Table 5: Results of human evaluation metrics on the test set (higher is better). All these models are trained and tested give both videos and surrounding comments.

Inspired by the evaluation methods of dialogue models (Das et al. 2017), we formulate the evaluation as a ranking problem. The model is asked to sort a set of candidate comments based on the log-likelihood score. Since the model generates the comments with the highest scores, it is reasonable to discriminate a good model according to its ability to rank the correct comments on the top of the candidates. The candidate comment set consists of the following parts:

- **Correct:** The ground-truth comments of the corresponding videos provided by the human.
- **Plausible:** The 50 most similar comments to the title of the video. We use the title of the video as the query to retrieval the comments that appear in the training set based on the cosine similarity of their tf-idf values. We select the top 30 comments that are not the correct comments as the plausible comments.
- **Popular:** The 20 most popular comments from the dataset. We count the frequency of each comment in the training set, and select the 20 most frequent comments to form the popular comment set. The popular comments are the general and meaningless comments, such as “2333”, “Great”, “hahahaha”, and “Leave a comment”. These comments are dull and do not carry any information, so they are regarded as incorrect comments.
- **Random:** After selecting the correct, plausible, and popular comments, we fill the candidate set with randomly selected comments from the training set so that there are 100 unique comments in the candidate set.

Following the previous work (Das et al. 2017), We measure the rank in terms of the following metrics: **Recall@k**

(the proportion of human comments found in the top-k recommendations), **Mean Rank** (the mean rank of the human comments), **Mean Reciprocal Rank** (the mean reciprocal rank of the human comments).

Experiments

Settings

For both models, the vocabulary is limited to the 30,000 most common words in the training dataset. We use a shared embedding between encoder and decoder and set the word embedding size to 512. For the encoding CNN, we use a pretrained resnet with 18 layers provided by the Pytorch package. For both models, the batch size is 64, and the hidden dimension is 512. We use the Adam (Kingma and Ba 2014) optimization method to train our models. For the hyper-parameters of Adam optimizer, we set the learning rate $\alpha = 0.0003$, two momentum parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$ respectively, and $\epsilon = 1 \times 10^{-8}$.

Baselines

- **S2S-I** (Vinyals et al. 2015) applies the CNN to encode the frames, based on which the decoder generates the target live comment. This model only uses the video as the input.
- **S2S-C** (Sutskever, Vinyals, and Le 2014) applies an LSTM to make use of the surrounding comments, based on which the decoder generates the comments. This model can be regarded as the traditional sequence-to-sequence model, which only uses the surrounding comments as input.
- **S2S-IC** is similar to (Venugopalan et al. 2015a) which makes use of both the visual and textual information. In our implementation, the model has two encoders to encode the images and the comments respectively. Then, we concatenate the outputs of two encoders to decode the output comments with an LSTM decoder.

Results

At the training stage, we train S2S-IC, Fusional RNN, and Unified Transformer with both videos and comments. S2S-I is trained with only videos, while S2S-C is trained with only comments. At the testing stage, we evaluate these models under three settings: video only, comment only, and both

video and comment. Video only means that the model only uses the images as inputs (5 nearest frames), which simulates the case when no surrounding comment is available. Comment only means that the model only takes input of the surrounding comments (5 nearest comments), which simulates the case when the video is of low quality. Both denotes the case when both the videos and the surrounding comments are available for the models (5 nearest frames and comments).

Table 4 summarizes the results of the baseline models and the proposed models under three settings. It shows that our proposed models outperform the baseline models in terms of all evaluation metrics under all settings, which demonstrates the effectiveness of our proposed models. Moreover, it concludes that given both videos and comments the same models can achieve better performance than those with only videos or comments. Finally, the models with only comments as input outperform the models with only videos as input, mainly because the surrounding comments can provide more direct information for making the next comments.

Human Evaluation

The retrieval evaluation protocol evaluates the ability to discriminate the good comments and the bad comments. We also would like to evaluate the ability to generate human-like comments. However, the existing generative evaluation metrics, such as BLEU and ROUGE, are not reliable, because the reference comments can be various. Therefore, we conduct human evaluation to evaluate the outputs.

We evaluate the generated comments in three aspects: **Fluency** is designed to measure whether the generated live comments are fluent setting aside the relevance to videos. **Relevance** is designed to measure the relevance between the generated live comments and the videos. **Correctness** is designed to synthetically measure the confidence that the generated live comments are made by humans in the context of the video. For all of the above three aspects, we stipulate the score to be an integer in $\{1, 2, 3, 4, 5\}$. The higher the better. The scores are evaluated by three human annotators and finally we take the average of three raters as the final result.

We compare our Fusional RNN model and Unified Transformer model with the strong baseline model S2S-IC. All these models are trained and tested give both videos and surrounding comments. As shown in Table 5, our models achieve higher scores over the baseline model in all three degrees, which demonstrates the effectiveness of the proposed models. We also evaluate the reference comments in the test set, which are generated by the human. It shows that the comments from human achieve high fluency and correctness scores. However, the relevance score is lower than the fluency and correctness, mainly because the comments are not always relevant to the videos, but with the surrounding comments. According to the table, it also concludes that the comments from unified transformer are almost near to those of real-world live comments. We use Spearman’s Rank correlation coefficients to evaluate the agreement among the raters. The coefficients between any two raters are all near 0.6 and at an average of 0.63. These high coefficients show that our human evaluation scores are consistent and credible.

Related Work

Inspired by the great success achieved by the sequence-to-sequence learning framework in machine translation (Sutskever, Vinyals, and Le 2014; Cho et al. 2014; Bahdanau, Cho, and Bengio 2014), Vinyals et al. (2015) and Mao et al. (2014) proposed to use a deep convolutional neural network to encode the image and a recurrent neural network to generate the image captions. Xu et al. (2015) further proposed to apply attention mechanism to focus on certain parts of the image when decoding. Using CNN to encode the image while using RNN to decode the description is natural and effective when generating textual descriptions.

One task that is similar to live comment generation is image caption generation, which is an area that has been studied for a long time. Farhadi et al. (2010) tried to generate descriptions of an image by retrieving from a big sentence pool. Kulkarni et al. (2011) proposed to generate descriptions based on the parsing result of the image with a simple language model. These systems are often applied in a pipeline fashion, and the generated description is not creative. More recent work is to use stepwise merging network to improve the performance (Liu et al. 2018).

Another task which is similar to this work is video caption generation. Venugopalan et al. (2015a) proposed to use CNN to extract image features, and use LSTM to encode them and decode a sentence. Similar models (Shetty and Laaksonen 2016; Jin et al. 2016; Ramanishka et al. 2016; Dong et al. 2016; Pasunuru and Bansal 2017; Shen et al. 2017) are also proposed to handle the task of video caption generation. Das et al. (2017) introduce the task of Visual Dialog, which requires an AI agent to answer a question about an image when given the image and a dialogue history. Moreover, we are also inspired by the recent related work of natural language generation models with the text inputs (Ma et al. 2018; Xu et al. 2018).

Conclusions

We propose the tasks of automatic live commenting, and construct a large-scale live comment dataset. We also introduce two neural models to generate the comments which jointly encode the visual contexts and textual contexts. Experimental results show that our models can achieve better performance than the previous neural baselines.

Acknowledgement

We thank the anonymous reviewers for their thoughtful comments. This work was supported in part by National Natural Science Foundation of China (No. 61673028). Xu Sun is the corresponding author of this paper.

References

- Agrawal, A.; Batra, D.; and Parikh, D. 2016. Analyzing the behavior of visual question answering models. In *EMNLP 2016*, 1955–1960.
- Antol, S.; Agrawal, A.; Lu, J.; Mitchell, M.; Batra, D.; Zitnick, C. L.; and Parikh, D. 2015. VQA: visual question answering. In *ICCV 2015*, 2425–2433.

- Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.
- Chen, D., and Dolan, W. B. 2011. Collecting highly parallel data for paraphrase evaluation. In *ACL 2011*, 190–200.
- Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.
- Das, P.; Xu, C.; Doell, R. F.; and Corso, J. J. 2013. A thousand frames in just a few words: Lingual description of videos through latent topics and sparse object stitching. In *CVPR 2013*, 2634–2641.
- Das, A.; Kottur, S.; Gupta, K.; Singh, A.; Yadav, D.; Moura, J. M.; Parikh, D.; and Batra, D. 2017. Visual dialog. In *CVPR 2017*, 1080–1089.
- Donahue, J.; Hendricks, L. A.; Rohrbach, M.; Venugopalan, S.; Guadarrama, S.; Saenko, K.; and Darrell, T. 2017. Long-term recurrent convolutional networks for visual recognition and description. *IEEE Trans. Pattern Anal. Mach. Intell.* 39(4):677–691.
- Dong, J.; Li, X.; Lan, W.; Huo, Y.; and Snoek, C. G. 2016. Early embedding and late reranking for video captioning. In *Proceedings of the 2016 ACM on Multimedia Conference*, 1082–1086. ACM.
- Fang, H.; Gupta, S.; Iandola, F. N.; Srivastava, R. K.; Deng, L.; Dollár, P.; Gao, J.; He, X.; Mitchell, M.; Platt, J. C.; Zitnick, C. L.; and Zweig, G. 2015. From captions to visual concepts and back. In *CVPR 2015*, 1473–1482.
- Farhadi, A.; Hejrati, M.; Sadeghi, M. A.; Young, P.; Rashtchian, C.; Hockenmaier, J.; and Forsyth, D. A. 2010. Every picture tells a story: generating sentences from images. 15–29.
- Jin, Q.; Chen, J.; Chen, S.; Xiong, Y.; and Hauptmann, A. 2016. Describing videos using multi-modal fusion. In *Proceedings of the 2016 ACM on Multimedia Conference*, 1087–1091. ACM.
- Karpathy, A., and Fei-Fei, L. 2017. Deep visual-semantic alignments for generating image descriptions. *IEEE Trans. Pattern Anal. Mach. Intell.* 39(4):664–676.
- Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *CoRR* abs/1412.6980.
- Kulkarni, G.; Premraj, V.; Dhar, S.; Li, S.; Choi, Y.; Berg, A. C.; and Berg, T. L. 2011. Baby talk: Understanding and generating simple image descriptions. 1601–1608.
- Liu, F.; Ren, X.; Liu, Y.; Wang, H.; and Sun, X. 2018. Step-wise image-topic merging network for generating detailed and comprehensive image captions. In *EMNLP 2018*.
- Ma, S.; Sun, X.; Li, W.; Li, S.; Li, W.; and Ren, X. 2018. Query and output: Generating words by querying distributed word representations for paraphrase generation. In *NAACL-HLT 2018*, 196–206.
- Mao, J.; Xu, W.; Yang, Y.; Wang, J.; and Yuille, A. L. 2014. Explain images with multimodal recurrent neural networks. *arXiv: Computer Vision and Pattern Recognition*.
- Pasunuru, R., and Bansal, M. 2017. Multi-task video captioning with video and entailment generation. *arXiv preprint arXiv:1704.07489*.
- Ramanishka, V.; Das, A.; Park, D. H.; Venugopalan, S.; Hendricks, L. A.; Rohrbach, M.; and Saenko, K. 2016. Multimodal video description. In *Proceedings of the 2016 ACM on Multimedia Conference*, 1092–1096. ACM.
- Rohrbach, A.; Rohrbach, M.; Qiu, W.; Friedrich, A.; Pinkal, M.; and Schiele, B. 2014. Coherent multi-sentence video description with variable level of detail. In *GCPR 2014*, 184–195.
- Rohrbach, A.; Rohrbach, M.; Tandon, N.; and Schiele, B. 2015. A dataset for movie description. In *CVPR 2015*, 3202–3212.
- Shen, Z.; Li, J.; Su, Z.; Li, M.; Chen, Y.; Jiang, Y.-G.; and Xue, X. 2017. Weakly supervised dense video captioning. In *CVPR 2017*, volume 2, 10.
- Shetty, R., and Laaksonen, J. 2016. Frame-and segment-level features and candidate pool evaluation for video caption generation. In *Proceedings of the 2016 ACM on Multimedia Conference*, 1073–1076. ACM.
- Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, 3104–3112.
- Tapaswi, M.; Zhu, Y.; Stiefel, R.; Torralba, A.; Urta-sun, R.; and Fidler, S. 2016. Movieqa: Understanding stories in movies through question-answering. In *CVPR 2016*, 4631–4640.
- Torabi, A.; Pal, C. J.; Larochelle, H.; and Courville, A. C. 2015. Using descriptive video services to create a large data source for video annotation research. *CoRR* abs/1503.01070.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. In *NIPS 2017*, 6000–6010.
- Venugopalan, S.; Rohrbach, M.; Donahue, J.; Mooney, R. J.; Darrell, T.; and Saenko, K. 2015a. Sequence to sequence - video to text. In *ICCV 2015*, 4534–4542.
- Venugopalan, S.; Xu, H.; Donahue, J.; Rohrbach, M.; Mooney, R. J.; and Saenko, K. 2015b. Translating videos to natural language using deep recurrent neural networks. In *NAACL HLT 2015*, 1494–1504.
- Vinyals, O.; Toshev, A.; Bengio, S.; and Erhan, D. 2015. Show and tell: A neural image caption generator. In *CVPR 2015*, 3156–3164.
- Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R.; Zemel, R.; and Bengio, Y. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *ICML 2015*, 2048–2057.
- Xu, J.; Mei, T.; Yao, T.; and Rui, Y. 2016. MSR-VTT: A large video description dataset for bridging video and language. In *CVPR 2016*, 5288–5296.
- Xu, J.; Sun, X.; Zeng, Q.; Ren, X.; Zhang, X.; Wang, H.; and Li, W. 2018. Unpaired sentiment-to-sentiment translation: A cycled reinforcement learning approach. In *ACL 2018*, 979–988.