Interactive Video Retrieval with Dialog

Sho Maeoki[1], Kohei Uehara[1], and Tatsuya Harada[1,2]
[1]The University of Tokyo, [2]RIKEN
{maeoki, uehara, harada}@mi.t.u-tokyo.ac.jp

Abstract

In the contemporary world, recording videos can be done quickly and easily. The quantity and availability of videos have continued to increase, therefore, an effective video retrieval method has also become important. To retrieve a target video from a large collection of videos, a video retrieval system needs to obtain appropriate queries from a user. Given a sentence query, there are many similar videos related to the query. The video retrieval system requires more information in addition to the sentence to distinguish the target video from others. If the system actively collects more information on the target video, we can perform video retrieval effectively. Thus, we propose a system to retrieve videos by asking questions about the content of the videos, and leveraging the user's responses to the questions and the dialog history. Additionally, we confirmed the usefulness of the proposed system through experiments using the dataset called AVSD which includes videos and dialogs about the videos.

1. Introduction

Nowadays, with the widespread use of smartphones anyone can easily record videos, leading to an ever-increasing amount of content. To retrieve target videos from large amount of videos, an effective video retrieval method is important. Personal videos, including home videos and life log videos, generally do not become highly popular and are not distinguished in most cases. Therefore, we cannot use elements such as the number of hits and tag data, which can be used when retrieving videos on the web; this makes retrieval difficulty relatively high. For example, some impressive and valuable videos are buried in a lot of other videos and cannot be found easily. The premise of this research is to retrieve target videos in this type of situation.

Personal videos reflect people's hobbies or preferences, hence individuals tend to record videos in similar situations or record similar videos semantically. When a retrieval system searches a target video given a sentence query, it is difficult to distinguish a target video from other similar videos for this reason. To facilitate the retrieval of target videos among similar items, let us see a video retrieval using a sentence as a query as shown in Fig. 1. For example, when a user tried to retrieve a video using a query "*A man reading a book*" as shown in Fig. 1, a retrieval system returned a lot of similar videos matching the query. The user should have added more discriminative keywords in the query sentence. However, in reality, it is difficult to assume that a user is aware of the sentence that is suitable enough to represent the videos they are searching for, or it is labor-intensive. Indeed, a user usually recorded a large number of similar videos, and we can consider that it is impossible to remember the details of all videos. Therefore, we propose introducing a dialog as shown in Fig. 1. The system asks a question to efficiently search for videos that the user wants, and then the user replies to the questions. We call the part of the system that controls the interaction with the user the

Figure 1. Overview of video retrieval with dialog. The picture shows some frames of videos of "*a man reading a book*". It is assumed that there are various situations for the same behavior. It is possible to narrow down the candidates making use of appropriate questions and their responses.

"*agent*". For example, suppose that there are six videos in Fig. 1 and we try to retrieve a video of "*a man reading a book*". If the search is for the upper left video (a), the agent should ask "*How many people are in the video?*". Furthermore, if asking "*Is the person reading a book while lying down?*", the lower left video (d) can be separated from the others in Fig. 1. If the agent asks the question "*Does the person read while standing*", the agent can search for the video in the upper middle (b) of Fig. 1. The agent can also distinguish videos by asking questions such as "*What was the person doing at the start of the video?*" and "*What was the person doing before/after reading a book?*". Moreover, since the user is assumed to have some knowledge of the target video, the user doesn't need to check all videos when answering these questions. That is to say, ideally the user does not have to look at the candidate videos and identify the optimal query and can instead simply search for the target video by answering the questions from the agent. It is clear that video retrieval can be performed effectively by introducing this type of dialog.

The purpose of this research is to retrieve target videos among similar items by introducing a dialog between the system and the user. The contributions of this research are threefold. (1) We proposed a new video retrieval method that utilizes interactive elements *i.e*., dialog. (2) We implemented a model to instantiate it and compared with several baselines. (3) We conducted the user study and confirmed that the model was effective for the task proposed in this study.

2. Related Work

2.1. Text-based Video Retrieval

In the video retrieval method using text, we typically first learn a mapping that transforms text and video features into a joint embedding space [23]. Then, in the learned joint embedding space, a video with a high degree of similarity to the sentence used as the input query is output as a search result. Previously, canonical correlation analysis (CCA) had been used as an approach to learn the mapping of the joint space. Training is performed to maximize the covariance of the distribution of the two different modalities in the embedding space. Presently, methods using deep neural networks (DNN) are popular [26, 17, 16, 24, 21, 25, 19, 6, 10] thanks to their impressive performance. There are two types of features embedded in the joint space in the case of video retrieval, a sentence feature and a video feature, the idea of which is based on the text-based image retrieval including [7]. The former is often obtained by inputting text such as captions to recurrent neural networks (RNN) that can handle time-series data and adopts its final hidden state as the representation. The latter is effectively obtained when considering features obtained by applying convolution neural networks (CNN) to each frame of a video in a multilateral manner [26, 17, 16, 24] .

In these related studies, basically, given one short sentence as input, the corresponding video is output. These methods are not sufficient for handling a history of dialog with multiple sentences. For this reason, these cannot be used in this research, which has to deal with dialog.

2.2. Vision and Dialog

The visual dialog proposed by Das *et al*. [3] is a task that takes an image and multiple questions as inputs and subsequently outputs a response to each question. Based on this, visual dialog is still actively researched, and research on video dialog that targets videos instead of images has begun [27, 12, 13, 20, 1]. However, these studies are aimed at returning better responses based on the contents of the videos, and there is no module for video retrieval.

In contrast, there are also studies that have proposed training methods for the goal-oriented visual dialog. Das *et al*. [4] enable interactive image retrieval with dialog. They generate an asymmetric scene in which an image can be viewed from an answerer who can see the image and a questioner who cannot see it. Under the circumstances, the questioner asks a question about the image to the answerer. The answerer in turn gives a response so that the questioner can gain a finer understanding of the corresponding image. At the time of training, the questioner tries to regress the image feature using the dialog history without knowing the image feature of the ground truth (GT) known to the answerer. They argue that it is possible to improve dialog performance by performing collaborative reinforcement learning on this task. Das *et al*. [4] proposed an image retrieval method using image features obtained by regression from the dialog history representation as a method for evaluating the questioner. This evaluation can be interpreted as an image retrieval with dialog. However, they only considered the point that the feature predicted by the questioner approaches the feature of the GT image known to the answerer in training. Therefore, it is impossible to distinguish between the target video and similar ones. Thus, we consider that this is not sufficient to achieve the purpose of this research.

3. Task and Model

3.1. Task Description

In this task, we assume a situation as follows. First, a user inputs a sentence query to a video retrieval system searching for a target video. Then, the system outputs candidate videos, which contain N items. This N can be varied. If the target video is included in the candidate videos, the retrieval is successfully finished. If not, the system asks a question to the user so that the system can distinguish the target video from other similar videos, consequently output

the target video in the N candidate videos. The user responds to this question based on the target video and the dialog history thus far as well as the candidate videos. After this round of dialog, candidate videos are updated, which are displayed to the user again. If the target video is not included in the candidate videos, the system asks another question and the next round of dialog follows. This type of Q&A is iteratively performed until the user reaches the target video in the candidate videos.

3.2. Modeling and Overview

The following function requirement (FR) is considered necessary to achieve the goal of this research, *i.e.*, FR: Making use of user's responses and dialog history. To utilize the dialog, the proposed system needs to be able to effectively use user responses and the history of dialog. For instance, when a response to a question is provided, the dialog history must be used adequately to improve retrieval performance. In addition, question generation in dialog requires the dialog history until a certain round. Moreover, to identify a target video, it is ideally necessary to generate a question on information that is not known yet.

In this section, we provide an overview of a model of the proposed method that satisfies the functional requirement mentioned above. Fig. 2 is an overview. The following describes the proposed system, assuming that one combination of a question and its response is called one round of dialog. As shown in Fig. 2, in the proposed system, there is an agent interacting with the user. The agent's main role is to generate the question and present candidate videos based on the dialog while interacting with the user. In this system, the user first inputs a query describing the video that he/she is searching for, and the agent outputs a question in return. The proposed system starts from the point where the user inputs a natural sentence describing the target video to the agent. When the first sentence D_0 is input, the agent uses that sentence as a query, and presents to the user several (N) videos that are close to the query in the feature space, namely $\mathrm{Cand}_0 = \{C_0^{(1)}, C_0^{(2)}, ..., C_0^{(N)}\}$. Note $C_0^{(1)}$ has the highest similarity to the query while $C_0^{(N)}$ has the lowest among Cand_0. In this study, we assume $N = 10$ as per [17]. It is defined as the 0th round of the dialog until this first sentence D_0 is input and the first candidate videos Cand_0 are output. After completing the 0th round dialog, the agent generates a question q_1, and the next round of the dialog begins based on this. Considering the t-th round dialog ($t = 1, ..., T$), the question q_t is generated after round $t-1$ of the dialog and the user responses a_t in return. That is to say, by denoting the t-th round question and answer pair as $H_t = [q_t, a_t]$ and the dialog history until the t-th round dialog as $\mathrm{DH}_t = \{D_0, H_1, ..., H_t\}$, the series of processes are expressed as follows.

$$\mathbf{s}_t = \mathrm{HistEnc}(\mathrm{DH}_t; \theta_{he}). \quad (1)$$

$$\mathrm{Cand}_t = \underset{v \in V}{\mathrm{top}\, N}\, S(f_{de}(\mathbf{s}_t; \theta_{de}), f_{ve}(V; \theta_{ve})). \quad (2)$$

$$q_{t+1} = \mathrm{QuesDec}(\mathrm{DH}_t, \mathrm{Cand}_t, V; \theta_{qd}). \quad (3)$$

$\mathrm{HistEnc}(\cdot)$ in Eq. 1 is a mapping to obtain the representation of the dialog history $\mathbf{s}_t$ until the t-th round. In the following, we call the representation of the dialog history $\mathbf{s}_t$ the "*history vector*". Eq. 2 represents the mapping that uses the history vector $\mathbf{s}_t$ obtained by Eq. 1 to obtain N candidate videos Cand_t from the video group V in the database. This mapping is composed of $f_{de}(\cdot)$ and $f_{ve}(\cdot)$ which embed the history vector $\mathbf{s}_t$ and the feature group V respectively into the joint space. In the joint embedding space, the similarity function $S(\cdot, \cdot)$ calculates the closeness between the embedded history vector and the embedded video features, and as a result N items considered to have a high similarity to the embedded history vector are selected. $\mathrm{QuesDec}(\cdot)$ in Eq. 3 is a mapping that outputs the next question q_{t+1} with the dialog history DH_t, candidate videos Cand_t and the feature group V as input. Eq. 1, Eq. 2, and Eq. 3 are the history encoding module (History Encoder), the feature embedding module (Feature Embedding) and the question generation module (Question Decoder) in Fig. 2. All modules are executed based on $\theta_{he}, \theta_{de}, \theta_{ve}, \theta_{qd}$. In this study, the first sentence D_0 is considered as a caption of a video, and the end is assumed to be the point where the round of dialog reaches a predetermined upper limit $T(= 10)$.

3.3. Model Architecture

History Encoder. This module is responsible for encoding dialog history for interpretation. Inspired by [4], we adopt hierarchical encoding for this module. The dialog in each round is decomposed into words and then represented as vectors by the word embedding matrix. These words are fed into an LSTM (Sentence Encoder) to obtain sentence level features as expressed in Eq. 4. Let $\mathbf{h}_0, ..., \mathbf{h}_t$ be the output from the Sentence Encoder. Note $\mathbf{h}_0$ is obtained by feeding D_0 as input, H_t is the concatenation of q_t and a_t.

$$\mathbf{h}_t = \mathrm{LSTM}(H_t). \quad (4)$$

These are input sequentially into an LSTM (State Encoder) in Eq. 5 which is different from the Sentence Encoder.

$$\mathbf{s}_t = \mathrm{LSTM}(\mathbf{h}_t | \mathbf{h}_0, \mathbf{h}_1, ..., \mathbf{h}_{t-1}). \quad (5)$$

The final hidden state obtained from the State Encoder is considered to be a feature that represents the entire dialog history effectively. Therefore, we adopted the final hidden state as the history vector $\mathbf{s}_t$.

The history vector $\mathbf{s}_t$ obtained as described above is a vector that semantically reflects the history of past conversations. Therefore, using this feature as an input to construct a later pipeline will satisfy the FR.

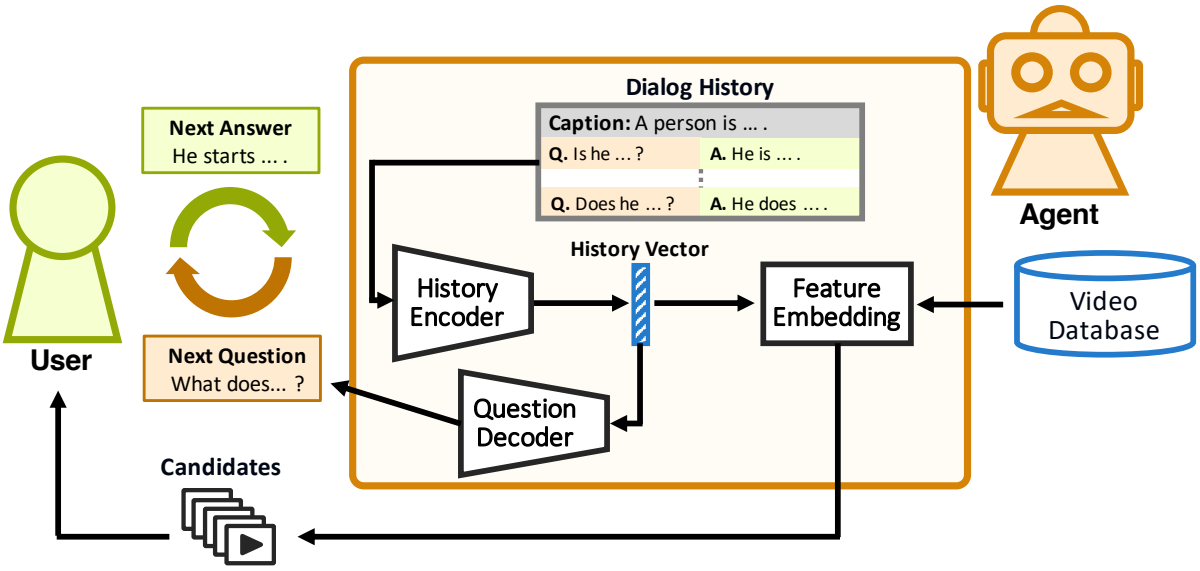


Figure 2. An overview of the proposed model and each module.

Feature Embedding. In this module, videos are searched using the dialog history as a query, and candidate videos are output. The history vector $\mathbf{s}_t \in \mathbb{R}^{|\mathbf{s}|}$ obtained by Eq. 1 and the video features V are used as input of this module, which is responsible for embedding into the joint space as in [17].

To reduce the computational cost in training, the video feature in this study uses pre-extracted features from a pre-trained model, by sequentially feeding segments, each of which includes several frames. Assuming that we have a pre-trained model $\mathrm{VideoEnc}(\cdot)$ and a video $v \in V$, then

$$\mathrm{VideoEnc}(v) = \{{}^{1}\mathbf{v}, ..., {}^{\tau_{max}}\mathbf{v}\}, \tag{6}$$

where $\{{}^{\tau}\mathbf{v}\}_{\tau=1}^{\tau_{max}} \in \mathbb{R}^{|\mathbf{v}|\times\tau_{max}}$ is a group of pre-extrated video features. The video v is embedded as per

$$\overline{\mathbf{v}} = f_{ve}(v). \tag{7}$$

Similarly, the history vector $\mathbf{s}_t$ is embedded according to

$$\overline{\mathbf{s}}_t = f_{de}(\mathbf{s}_t). \tag{8}$$

Specifically, the video v is processed to an embedded vector $\overline{\mathbf{v}}$ via $f_{ve}(\cdot)$ as formulated in Eq. 9 and Eq. 10.

$$f_{ve}(v) = \mathrm{FC}(\mathbf{v}), \tag{9}$$

where

$$\mathbf{v} = \mathrm{pooling}(\mathrm{VideoEnc}(v)). \tag{10}$$

Note FC is a fully connected layer and pooling is performed across the time direction. Also, we do not train $\mathrm{VideoEnc}(\cdot)$ in this study. As for $f_{de}(\cdot)$, a fully connected layer is adopted. More generally, $\mathbf{v}$ is obtained by

$$\mathbf{v} = \mathrm{VideoProc}(v). \tag{11}$$

In this paper, $\mathrm{VideoProc}(v)$ includes concatenation operation for pooled features from pre-trained models. Given the two embedded features $\overline{\mathbf{s}}_t$ and $\overline{\mathbf{v}}$, similarity between these two are calculated by $S(\cdot,\cdot)$ shown in Eq. 2. Then, we can obtain candidate videos Cand_t as per Eq. 2. For $S(\cdot,\cdot)$, we adopt consine similarity as in the study of Mithun *et al.* [17].

In this module, the history vector representing the dialog history and the feature of the videos are directly related to the joint space. Hence, video features treated here should reflect dialog content properly. Particularly by choosing $\mathrm{VideoProc}(\cdot)$ which is good at detecting motions in videos, we can expect better performance, for conversations held in this task could involve various kinds of topics including motions.

Question Decoder. This module is responsible for the question generation necessary for the video retrieval with dialog. This module consists of one LSTM, of which the hidden state output is used to calculate the probability of words in the generated question. Note this module takes as input the dialog history representation.

3.4. Model Learning

Feature Embedding. The trained joint embedding space determines video retrieval performance in the end. Video retrieval fails when the joint space is not properly trained, even if the agent succeeded in obtaining necessary information. Therefore, the loss function for embedding features is important. We train the joint embedding space by minimizing the following loss function Eq. 12 proposed by [7]. α in Eq. 12 is a margin and $S(\cdot,\cdot)$ calculates cosine similarity for input pairs. $\hat{\mathbf{v}}$ and $\hat{\mathbf{s}}_t$ are called hard negatives (*i.e.*, the negative video/dialog sample closest to a positive

matching $(\overline{\mathbf{v}}, \overline{\mathbf{s}}_t)$ pair), both of which are defined by $\hat{\mathbf{v}} = \arg\max_{\mathbf{v}^-} S(\mathbf{v}^-, \overline{\mathbf{s}}_t)$ and $\hat{\mathbf{s}}_t = \arg\max_{\mathbf{s}_t{}^-} S(\overline{\mathbf{v}}, \mathbf{s}_t{}^-)$. Note $\mathbf{v}^-$ and $\mathbf{s}_t{}^-$ are negative video/dialog sample for a positive pair $(\overline{\mathbf{v}}, \overline{\mathbf{s}}_t)$. By minimizing Eq. 12, the model learns to increase the similarity for positive matching pairs while decrease the similarity for hard negatives. That is to say, Eq. 12 attempts to bring positive samples closer than negative samples (hard negatives). In this way, we can train the Feature Embedding so that similar items are easier to distinguish in the joint space. In the actual implementation, training is performed within a mini-batch, so training will proceed by keeping away the hard negatives from matching samples in a mini-batch (*i.e.*, semi-hard negatives).

$$\begin{aligned}\mathcal{L}_{feat} = &\sum_{\overline{\mathbf{v}}} \max(0, \alpha - S(\overline{\mathbf{v}}, \overline{\mathbf{s}}_t) + S(\overline{\mathbf{v}}, \hat{\mathbf{s}}_t)) \\ &+ \sum_{\overline{\mathbf{s}}_t} \max(0, \alpha - S(\overline{\mathbf{v}}, \overline{\mathbf{s}}_t) + S(\hat{\mathbf{v}}, \overline{\mathbf{s}}_t)).\end{aligned} \tag{12}$$

Question Decoder. We would like to train the question decoder in a supervised way utilizing ideal questions that people actually uttered *i.e.*,

$$\hat{q}_t = human(\mathrm{DH}_{t-1}, \mathrm{Cand}_{t-1}, V), \tag{13}$$

for $t = 1, ..., T$. While training, we would like to use teacher forcing with cross entropy loss based on the dialog history including a group of questions for each video $\hat{Q} = \{\hat{q}_t\}_{t=1}^T$ and the responses $\hat{A} = \{\hat{a}_t\}_{t=1}^T$. In this manner, the question decoder in Eq. 3 can be reformulated as

$$q_{t+1} = \mathrm{QuesDec}(\mathbf{s}_t; \theta_{qd}), \tag{14}$$

in which θ_{qd} is expected to learn "*common sense*" about which question to generate for each topic of videos, as a result of supervised learning based on $\hat{Q}$. Note $\mathbf{s}_t$ reflects the dialog history DH_t including D_0, $\hat{Q}$ and $\hat{A}$.

In practice however, it is still challenging to construct a question decoder that satisfies Eq. 14 due to a lack of dataset which contains $\hat{Q}$. To mitigate this issue, we utilize a dataset which contains questions $\tilde{Q}$ similar to $\hat{Q}$. Details about the dataset are described in subsection 4.1.

Summary. When training the whole model, we minimize the linear sum in Eq. 15 by writing the loss function Eq. 12 in the Feature Embedding as $\mathcal{L}_{feat}$ and the Question Decoder's cross entropy loss as $\mathcal{L}_{ques}$. The coefficients $\lambda_{feat}, \lambda_{ques}$ in the Eq. 15 are hyperparameters that indicate the extent to which each module is emphasized.

$$\min_{\theta} \left(\lambda_{feat}\, \mathcal{L}_{feat} + \lambda_{ques}\, \mathcal{L}_{ques}\right). \tag{15}$$

4. Experiments

4.1. Settings

Dataset. In this study, we used the AVSD dataset [1]. This dataset was created by adding dialog data to the existing video dataset called Charades [22]. Charades is a video dataset of about 30 seconds, and the collection contains activities that people will be exposed to in their daily lives. Charades has many motions in one video (each video includes at least two actions as per [1]) and is characterized by the presence of many semantically similar videos in the dataset. The AVSD dataset contains ten rounds of questions and answers for each video. Questions include a lot of spatio-temporal information (e.g., actions, interactions, and development of events) as well as the audio information. In the annotation process of the AVSD dataset, as per [1], two workers on Amazon Mechanical Turk (AMT) were asked to help annotate a video, one of whom is a questioner and the other is an answerer. The questioner is presented with 3 frames from the video *i.e.*, beginning, middle and end of the video, then asks a question to obtain a good understanding of what is actually happening in the video. The answerer, who has already watched the video and read a script D_0 about the video, responds. The two workers are implicitly encouraged to hold conversations with rich information unique to videos in this protocol. Questions in the AVSD dataset ($\tilde{Q}$) are asked in order to guess the true video given partial information about the video, which is similar to Eq. 13. Thus, we can assume $\tilde{Q} \approx \hat{Q}$. For details, please see Sec. A in the supplementary material.

From the above, the AVSD dataset, which includes dialogs focusing on elements unique to the videos, targeting recordings by individuals including home videos and lifelog videos, is suitable for this research. AVSD has 7,985 samples for training. We used 863 samples for validation and 1,000 samples for testing. Part of the validation data was adopted as test data in this study.

Evaluation Metrics. Here we introduce the evaluation metrics used in this research. We measure the rank-based performance by Recall@k (R@k), Mean Rank (MeanR), and Mean Reciprocal Rank (MRR). R@k calculates the percentage that the GT video is found in the top-k retrieved points, which can be interpreted as the percentage that Cand_t with respect to $N = k$ includes the GT video. MeanR calculates the mean rank of all GT videos and MRR is the mean of multiplicative inverse of the rank for all GT videos. Note MRR is equivalent to mean average precision (mAP) in this case as there is only one correct target video for each dialog. Ideally, R@k, MeanR, and MRR indicate 100, 1 and 1 respectively. Higher is better for R@k and MRR, while lower is better for MeanR. Note that R@k ($k = 1, 5, 10$), MeanR, MRR in Table 1, Table 2 and Table 3 are the values obtained when 10 rounds of GT dialog data are input.

Baselines. As baselines, we prepare three types of models overall, namely L2 Loss, C Loss, and LSTM. In the L2 Loss model, L2 norm for positive pairs $(\overline{\mathbf{s}}_t, \overline{\mathbf{v}})$ expressed in

Eq. 16

$$\mathcal{L}_{feat} = \sum_{(\overline{\mathbf{s}}_t, \overline{\mathbf{v}})} ||\overline{\mathbf{s}}_t - \overline{\mathbf{v}}||^2 \tag{16}$$

is applied as L2 Loss in Feature Embedding instead of ranking loss in Eq. 12. L2 Loss centers on bringing the positive samples closer with no consideration for keeping the negative samples apart. This is equivalent to the architecture of Das *et al*. [4] except that the L2 Loss baseline takes different features from their paper. Note in this baseline based on their original implementation [18], $f_{ve}(v)$ in Eq. 9 is composed without FC layer. This is because if the FC layer can be trained as per Eq. 16 there is a high possibility both embedded features could be zero to minimize the loss, hence training fails. The C Loss model is trained with contrastive loss expressed as

$$\mathcal{L}_{feat} = \sum_{pairs} y_{sv} d + (1 - y_{sv})(\Delta - d), \tag{17}$$

where d is a distance between embedded vectors $\overline{\mathbf{s}}_t$ and $\overline{\mathbf{v}}$ for any $pairs$ in the batch, Δ is a margin, and y_{sv} is a label which gives 1 or 0 when the given pair is correct or incorrect respectively. Eq. 17 enforces margin for negative samples while bringing the positive samples closer. Thus, unlike the L2 Loss in Eq. 16, contrastive loss is able to embed features so that they are easier to distinguish in the joint space. Since distance is a relative concept, a positive sample should be closer than negative samples. However, contrastive loss solely takes into account given pairs. Consequently, some negative sample could be closer than a positive sample. For this reason, compared to the ranking loss in Eq. 12, contrastive loss has weak guarantee that the positive sample is closer than the negative samples. The LSTM model is a model which uses one LSTM in Eq. 18 as History Encoder. DH_t is decomposed into words and these are represented as word level features by the word embedding matrix. The difference from the proposed model is that the LSTM baseline does not encode the dialog history hierarchically. The LSTM model is based on the typical architecture in the text-based video retrieval represented by [17] in the sense that they encode sentences with an RNN, though commonly only D_0 is given as query.

$$\mathbf{s}_t = \mathrm{LSTM}(H_t | \mathrm{DH}_{t-1}). \tag{18}$$

Implementation Details. We prepare two types of features as a visual cue for the videos, the frame-wise feature extracted from ResNet152 [9] pre-trained with ImageNet [5] and the feature extracted from the I3D [2] model pre-trained with Kinetics [14]. Furthermore, the VGGish [11] feature pre-trained on the Audio Set [8] is prepared as an audio feature for supplementary data. These three types of features constitute $\mathrm{VideoProc}(v)$ in Eq. 10. Moreover, since the model proposed in this research is a complex configuration with multiple modules, it is difficult to train all parameters end-to-end at once. Therefore, training is divided into two steps; in the first step, the parameters of $f_{de}(\cdot)$ and $f_{ve}(\cdot)$ are fixed to the initial values, and in the second step, this constraint is released to train all parameters. The main hyperparameters are set as follows. The batch size is 32, the word embeddings are 300-d, the hidden state of the two LSTMs in the History Encoder is 512-d, the hidden state of an LSTM in the Question Decoder is 512-d, the dimension of the joint embedding space is 1,024, α in Eq. 12 is 0.2 and the coefficients in Eq. 15 are $\lambda_{feat} = 1,000, \lambda_{ques} = 2$. For some baselines, there are some differences in terms of the parameters because of the architecture difference. Δ in Eq. 17 is set to 1.0 for C Loss. As for the LSTM baseline, $\lambda_{feat} = 10,000$. Parameters are optimized using Adam [15] with an initial learning rate of 0.001. The number of dimensions of the video feature are ResNet152: 2,048, I3D: 1,024, VGGish: 128.

4.2. Results and Discussions

Table 1. Comparison of retrieval performance depending on the input video features. "M", "A", "S" in the table indicates I3D, ResNet, VGGish respectively, "+" represents concatenation.

	R@1	R@5	R@10	MeanR	MRR
M	3.90	12.4	20.5	117	0.0961
A	1.60	7.90	13.4	176	0.0591
S	0.300	1.90	3.00	365	0.0169
M + A	3.90	13.4	20.7	124	0.0970
M + S	4.20	**15.9**	24.2	**107**	0.112
A + S	2.70	9.20	15.1	174	0.0702
M + A + S	**4.40**	15.0	**24.4**	109	**0.113**

Feature Selection for Representing Videos. Here we select the appropriate representation for videos, *i.e*., $\mathrm{VideoProc}(\cdot)$ in Eq. 11. We compared the retrieval performance with the ResNet feature, I3D feature and VGGish feature, each of which is expected to represent appearance, motion and sound respectively. After that, experiments were also conducted with various types of features combined hoping that multiple features could be used effectively. This result is shown in Table 1. We adopt max pooling for pooling features. According to Table 1, the best performance is achieved by adopting a combination of three features basically aside for R@5 and MeanR. As M + A + S achieves highest R@10 and there is no big difference between M + S and M + A + S, in the following, we adopt M + A + S as the video representation. Comparing I3D and ResNet, I3D gives better results because the dataset used this time holds more dynamic behaviors, and I3D is more likely to reflect such features. The audio feature (VGGish) is poor in terms of stand-alone performance, but it

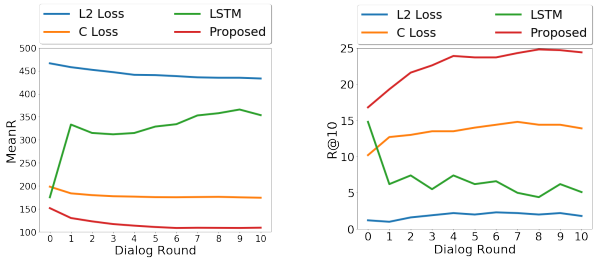
contributes to the improvement of the retrieval performance when considered simultaneously with I3D and/or ResNet. From the above, it appears crucial to choose an architecture sensitive to motions. Additionally, multimodal information contributes to the retrieval performance.

Table 2. Comparison of retrieval performance against the baselines regarding $\mathcal{L}_{feat}$. The proposed model achieves much better performance than the L2 Loss and the C Loss, while C Loss is still much better than the L2 Loss.

	R@1	R@5	R@10	MeanR	MRR
L2 Loss	0.300	0.700	1.80	433	0.0115
C Loss	2.50	9.4	13.9	174	0.0665
Proposed	**4.40**	**15.0**	**24.4**	**109**	**0.113**

Table 3. Comparison of retrieval performance against the baseline regarding the History Encoder. The proposed model achieves much better performance than the LSTM.

	R@1	R@5	R@10	MeanR	MRR
LSTM	0.500	2.70	5.10	354	0.0240
Proposed	**4.40**	**15.0**	**24.4**	**109**	**0.113**

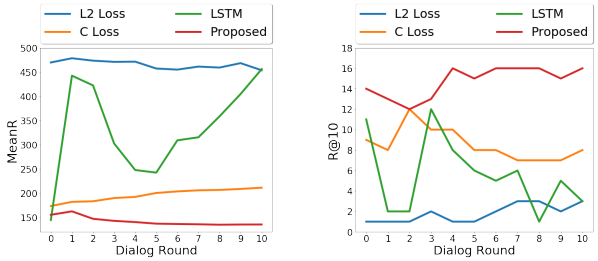


(a) (b)

Figure 3. Transition of MeanR (a) and R@10 (b) as the dialog proceeds. GT dialog data is used as input.

Comparisons against Baselines. Table 2 and Table 3 show the performance comparison against baselines. As can be seen in Table 2 and Table 3, we can confirm that the proposed method is superior for all metrics. Furthermore, Fig. 3 expresses the relationship between the number of dialog rounds and the retrieval performance. It is apparent that in the proposed method, MeanR tends to decrease as the dialog progresses. As for R@10, the progress of the dialog and the performance improvement are linked, and the significance of the dialog is apparent. First, we will compare the proposed model with two baselines in Table 2, namely L2 Loss and C Loss. According to Fig. 3, it appears that the L2 Loss tends to improve retrieval performance as the dialog progresses, as does the proposed model. However, the retrieval performance itself is poor. This result indicates that the loss function in Feature Embedding is insufficient with an L2 Loss. Meanwhile, C Loss achieves much better performance than the L2 Loss while showing a similar tendency to both the proposed model and the L2 Loss. Nevertheless, the proposed model which considers hard negatives indicates better performance. Thus, we can confirm that the retrieval performance itself can be improved by devising embedding loss into the joint space in the Feature Embedding. On the other hand, looking at Fig. 3, although the LSTM achieves almost the same performance as the proposed model in the 0th round, its performance deteriorates drastically in the first round and essentially deteriorates thereafter. We consider that this phenomenon occurs because the LSTM cannot effectively handle a long-term information sequence such as a dialog history. In other words, by finding a way to handle a long-term information sequence in the History Encoder, we can improve search performance along with the progress of the dialog.

4.3. User Study



(a) (b)

Figure 4. Transition of MeanR (a) and R@10 (b) as dialog proceeds when user study is performed with the proposed model and the other baselines.

User Study Settings. In the evaluation thus far, the dialog with humans is simulated using GT dialog data as input. However, an evaluation through a human-model interaction is also necessary. Therefore, we conducted a user study on AMT to check if the retrieval performance improves with the dialog when actually interacting with humans. We randomly selected 100 out of 1,000 videos of test data. Considering the purpose of this research, as a way of user study, ideally users should provide their own videos, and hold conversations searching for the videos. However, such an evaluation method is difficult in practice. For this reason, we used the videos in the dataset as the targets and had the workers hold conversation on the target videos. Specifically, we asked workers to respond to questions for ten rounds based on the content of the videos and the dialog history. The GT video ranks in the database were calculated using the dialog data obtained. Note the users were unaware that this task is related to video retrieval as users only need to respond in natural languages based on the target videos. For details, please see Sec. C in the supplementary material.

User Study Results. Fig. 4 illustrates the user study results. Looking at the result of the proposed model in Fig. 4, it ap-

caption		GT rank	Top video
A person is tidying some dishes . They **take a drink** from a coffee cup and lie down on the floor .		449	Video I
Q1:How many people are in the video ?	A1:There is one person in the video	467	
Q2:What is he doing at the beginning of the video ?	A2:He is washing dishes	141	
Q3:What room is he in ?	A3:In a kitchen	61	Video II
Q4:What is he doing in the beginning of the video ?	A4:Washing dishes	54	Video III

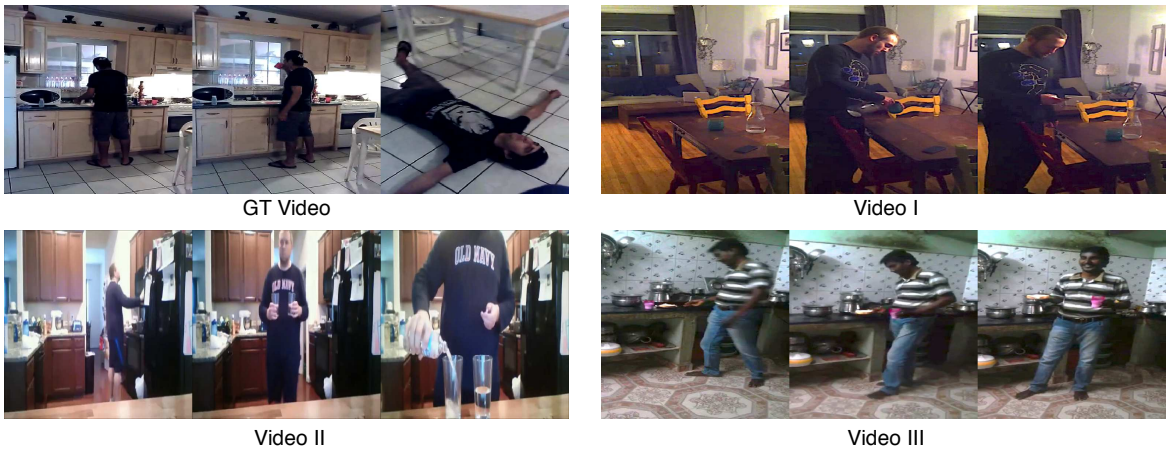


Figure 5. An example of qualitative results. The upper part shows the dialog targeting the GT video and its rank in other test (1000) videos as well as the top ranked videos in each round. The lower part shows the frames of the GT video and the top ranked videos mentioned in the upper part. All the top ranked videos (Video I, II, III) in the figure reflect "*take a drink*" in the caption. The background of the top videos changes from a living room to a kitchen as a result of Q3&A3; we can see both Video II and III are in kitchens. Moreover, after Q4&A4 a dish appears in the Video III in the man's right hand.

pears that the retrieval performance improves overall with the progress of the dialog for both MeanR and R@10, as in the case involving the GT dialog data shown in Fig. 3. We confirmed that even when a dialog with a human is actually performed, the retrieval performance can be enhanced using dialog for the proposed model. Other baselines excluding the C Loss indicate similar tendencies to Fig. 3. In the L2 Loss, the retrieval performance becomes better as dialog proceeds though the performance itself is poor. The LSTM achieves comparable performance with the proposed model first, however it deteriorates thereafter. Nevertheless, the C Loss performance tends to deteriorate as the dialog progresses, which is unlike the C Loss in Fig. 3. This tendency suggests that the C Loss failed to train the model properly.

Qualitative Results. Fig. 5 is the qualitative result of the proposed method when the user study is performed (Please see Sec. D in the supplementary material for more examples.) Targeting a video as a GT video, we can see the GT video rank becomes lower gradually, which indicates that the retrieval works well. Furthermore, the top ranked candidate videos reflect the dialog content. Clearly, the dialog content influences the retrieval result. However, as can be seen in Q2 and Q4, some questions generated by the model were similar or the same. This is because the question decoder is trained with teacher forcing using the dataset. The question decoder learned to generate frequently used questions in the dataset. Though the current model can adequately utilize the dialog history to improve video retrieval performance, questions with more varieties could improve the performance. Hence, improving the question decoder would be the next step in this research.

5. Conclusion

The purpose of this study was to introduce an interaction (dialog) in which the system generates questions based on the dialog history with the user, enabling the retrieval of the target video among similar ones. To achieve this, we proposed a model that can ask questions and utilize the dialog history. We showed its effectiveness in experiments using the AVSD dataset, with videos that are rich in scene developments. Furthermore, we conducted the user study and confirmed that the proposed model is effective when it actually interacts with humans.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number JP20H05556 and JST AIP Acceleration Research Grant Number JPMJCR20U3. We would like to thank Naoya Fushishita, Shunya Wakasugi, and Yusuke Mukuta for helpful discussions.

References

[1] Huda Alamri, Vincent Cartillier, Abhishek Das, Jue Wang, Stefan Lee, Peter Anderson, Irfan Essa, Devi Parikh, Dhruv Batra, Anoop Cherian, Tim K. Marks, and Chiori Hori. Audio-Visual Scene-Aware Dialog. In *CVPR*, 2019.

[2] Joao Carreira and Andrew Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *CVPR*, 2017.

[3] Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José M.F. Moura, Devi Parikh, and Dhruv Batra. Visual Dialog. In *CVPR*, 2017.

[4] Abhishek Das, Satwik Kottur, José M.F. Moura, Stefan Lee, and Dhruv Batra. Learning Cooperative Visual Dialog Agents with Deep Reinforcement Learning. In *ICCV*, 2017.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009.

[6] Jianfeng Dong, Xirong Li, and Cees GM Snoek. Word2visualvec: Image and Video to Sentence Matching by Visual Feature Prediction. *arXiv preprint arXiv:1604.06838*, 2016.

[7] Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. VSE++: Improving Visual-Semantic Embeddings with Hard Negatives. In *BMVC*, 2018.

[8] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. AUDIO SET: AN ONTOLOGY AND HUMAN-LABELED DATASET FOR AUDIO EVENTS. In *ICASSP*, 2017.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016.

[10] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan C. Russell. Localizing Moments in Video with Natural Language. In *ICCV*, 2017.

[11] Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, R. Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron J. Weiss, and Kevin W. Wilson. CNN ARCHITECTURES FOR LARGE-SCALE AUDIO CLASSIFICATION. In *ICASSP*, 2017.

[12] Chiori Hori, Huda Alamri, Jue Wang, Gordon Winchern, Takaaki Hori, Anoop Cherian, Tim Marks, Vincent Cartillier, Raphael Gontijo Lopes, Abhishek Das, Irfan Essa, Dhruv Batra, and Devi Parikh. End-to-End Audio Visual Scene-Aware Dialog using Multimodal Attention-Based Video Features. *arXiv preprint arXiv:1806.08409*, 2018.

[13] Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. TGIF-QA: Toward Spatio-Temporal Reasoning in Visual Question Answering. In *CVPR*, 2017.

[14] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The Kinetics Human Action Video Dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[15] Diederick P Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *ICLR*, 2015.

[16] Antoine Miech, Ivan Laptev, and Josef Sivic. Learning a Text-Video Embedding from Incomplete and Heterogeneous Data. *arXiv preprint arXiv:1804.02516*, 2018.

[17] Niluthpol Chowdhury Mithun, Juncheng Li, Florian Metze, and Amit K Roy-Chowdhury. Learning Joint Embedding with Multimodal Cues for Cross-Modal Video-Text Retrieval. In *ICMR*, 2018.

[18] Nirbhay Modhe, Viraj Prabhu, Michael Cogswell, Satwik Kottur, Abhishek Das, Stefan Lee, Devi Parikh, and Dhruv Batra. VisDial-RL-PyTorch. `https://github.com/batra-mlp-lab/visdial-rl.git`, 2018.

[19] Mayu Otani, Yuta Nakashima, Esa Rahtu, Janne Heikkilä, and Naokazu Yokoya. Learning Joint Representations of Videos and Sentences with Web Image Search. In *ECCV*, 2016.

[20] Ramakanth Pasunuru and Mohit Bansal. Game-Based Video-Context Dialogue. In *EMNLP*, 2018.

[21] Dian Shao, Yu Xiong, Yue Zhao, Qingqiu Huang, Yu Qiao, and Dahua Lin. Find and Focus: Retrieve and Localize Video Events with Natural Language Queries. In *ECCV*, 2018.

[22] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in Homes: Crowdsourcing Data Collection for Activity Understanding. In *ECCV*, 2016.

[23] Kaiye Wang, Qiyue Yin, Wei Wang, Shu Wu, and Liang Wang. A Comprehensive Survey on Cross-modal Retrieval. *arXiv preprint arXiv:1607.06215*, 2016.

[24] Masataka Yamaguchi, Kuniaki Saito, Yoshitaka Ushiku, and Tatsuya Harada. Spatio-temporal Person Retrieval via Natural Language Queries. In *ICCV*, 2017.

[25] Xiaoshan Yang, Tianzhu Zhang, and Changsheng Xu. Text2Video: An End-to-end Learning Framework for Expressing Text with Videos. *IEEE Transactions on Multimedia*, 20(9):2360–2370, 2018.

[26] Youngjae Yu, Jongseok Kim, and Gunhee Kim. A Joint Sequence Fusion Model for Video Question Answering and Retrieval. In *ECCV*, 2018.

[27] Zhou Zhao, Xinghua Jiang, Deng Cai, Jun Xiao, Xiaofei He, and Shiliang Pu. Multi-Turn Video Question Answering via Multi-Stream Hierarchical Attention Context Network. In *IJCAI*, 2018.