Knowledge-Based Video Question Answering with Unsupervised Scene Descriptions

Noa Garcia and Yuta Nakashima

Osaka University, Japan
{noagarcia, n-yuta}@ids.osaka-u.ac.jp

Abstract. To understand movies, humans constantly reason over the dialogues and actions shown in specific scenes and relate them to the overall storyline already seen. Inspired by this behaviour, we design ROLL, a model for knowledge-based video story question answering that leverages three crucial aspects of movie understanding: dialog comprehension, scene reasoning, and storyline recalling. In ROLL, each of these tasks is in charge of extracting rich and diverse information by 1) processing scene dialogues, 2) generating unsupervised video scene descriptions, and 3) obtaining external knowledge in a weakly supervised fashion. To answer a given question correctly, the information generated by each inspired-cognitive task is encoded via Transformers and fused through a modality weighting mechanism, which balances the information from the different sources. Exhaustive evaluation demonstrates the effectiveness of our approach, which yields a new state-of-the-art on two challenging video question answering datasets: KnowIT VQA and TVQA+.

Keywords: video question answering, video description, knowledge bases

1 Introduction

Robots may not dream of electric sheep yet,[1] but in the last few years, artificial intelligence has shown significant progress towards human-like reasoning. This has been made possible by emulating snippets of human intelligence in constrained tasks [1,11], where machine performance is easily evaluated. Among those tasks, video story question answering [35,17,5] emerged as a testbed to approximate real-world situations, in which not only the spatial relationships between objects are important, but also the temporal coherence between past, present, and future events.

Video story question answering leverages the structure of video stories, such as movies and TV shows, to formulate questions about specific scenes in a video. Models, then, need to find the correct answer by reasoning over the scene and its underlying plot. However, as the video story unfolds, the details of the plot are often revealed to the spectator over multiple scenes, sometimes far apart from each other. To understand the whole story, humans have the capacity to

[1] 'Do androids dream of electric sheep?' (Philip K. Dick, 1968).

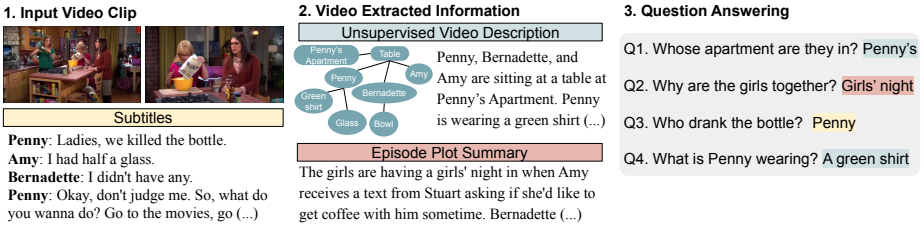

Fig. 1: ROLL performs video story question answering by generating unsupervised descriptions from video scene graphs and obtaining episode summaries.

constantly relate past events with what is currently being shown, acquiring contextual information that forms their story knowledge. We argue that for a full comprehension of video stories, not only what is happening in the current scene has to be considered, but also the knowledge acquired in previous scenes. Some examples are shown in Fig. 1; whereas the answer to Q1, Q3, and Q4 can be guessed from the video scene (and its subtitles), Q2 can only be inferred when the full context is known.

Previous work on video story question answering can be roughly divided into two categories. On one hand, there are models that extract information from the whole video story [35,23,20,14], and use attention mechanisms to find the parts that are relevant to each question. These models obtain contextual representations, which are used to answer general questions about the plot, but barely capture details at the scene level. On the other hand, other models extract detailed information from specific scenes [17,13], without looking at the whole video story. However, relying only on the content of short scenes is insufficient to answer insightful aspects about the story, such as the characters' motivations. To study multiple types of questions about video stories using both contextual and scene-specific information, a knowledge-based video question answering dataset has been recently introduced [5]. The proposed model combines contextual information from external resources with multi-modal representations from specific scenes. However, the contextual data in [5] is obtained from thousands of task-specific human-generated annotations, which are expensive to obtain and difficult to generalise to other domains.

In this paper, we introduce ROLL, Read, Observe, and Recall, a model that addresses knowledge-based video story question answering with both contextual and scene-specific information using unsupervised scene descriptions and weakly-supervised external knowledge. ROLL consists on a three-branch architecture inspired by three areas of human cognition playing an important role in video understanding: dialog comprehension (*read branch*), scene reasoning (*observe branch*), and storyline recalling (*recall branch*). Whereas the scene-specific details are summarised in the read and observe branches, the recall branch provides a contextual overview about the story using free online resources. To predict the correct answer, the three branches are lately fused through a modality weighting mechanism, which balances the signal from the three different sources.

Contributions: Our contribution is three-fold: 1) we propose a new unsupervised video representation based on video descriptions generated from video scene graphs; 2) we combine specific details from video scenes with weakly supervised external knowledge for a deep understanding of video stories; and 3) we incorporate a modality weighting mechanism to fuse data from different modalities without information loss. Our model is evaluated on two challenging video story question answering datasets: KnowIT VQA and TVQA+, outperforming previous work by more than 6.3% and 1.3%, respectively.

2 Related Work

We develop a model for video story question answering that 1) takes advantage of rich external knowledge sources, and 2) represents video content by generating unsupervised video captions from scene graphs. In the following, we first review work on video story question answering and visual reasoning with external knowledge before discussing scene graphs and methods for video description.

Video Story Question Answering Video story question answering is a modality in video question answering in which questions are not only related to the visual content of a video, but also to its plot. MovieQA [35] introduced a plot-oriented dataset with questions generated from movie summaries. Most proposed models [35,23,20,14] used frame-level features to represent the entire movie, applying attention mechanisms to find the relevant parts to each question. This provides a high-level overview of the story, but does not consider the details of each scene. Alternatively, PororoQA [15] and TVQA [17] formulated scene-level questions about specific events in the video. Models addressing these datasets described the details of each scene with features [40], captions [15] or visual concepts [17,13,52], but without attending to the ongoing plot in the video story. Recently, KnowIT VQA [5] introduced a combination of detailed questions about scenes and knowledge-based questions about the story. The proposed model relied on human-generated annotations to understand the insights of the plot. On the contrary, our model exploits both specific and general story information without task-specific annotations by using external knowledge bases.

Visual Reasoning with External Knowledge Using external knowledge in visual reasoning extends the visual question answering task (VQA) to address questions far beyond the visual content of images. Although the acquisition of knowledge depends on the task of interest, structured knowledge bases, such as DBpedia [2] or ConceptNet [34], are commonly used in most methods [45,42,41,25,24]. However, structured knowledge is usually represented as (subject, predicate, object) triplets, which is a hard constraint on the type of information being processed. Generic solutions [31,22] proposed to exploit unstructured resources in natural language, such as Wikipedia.[2] Following this direction, our model leverages unstructured online data to answer knowledge-based questions about video stories.

[2] https://www.wikipedia.org/

Scene Graphs Scene graphs [12] are structures that represent the objects depicted in an image and their relationships, providing a semantic description of the image. Most scene graph methods consist on an object detector, an attribute classifier and a relationship predictor [48,19,54,50,55,56]. Scene graphs have been used in multiple vision and language tasks, including image captioning [51,7,54] and VQA [36,32,46]. However, less attention has been paid to generating scene graphs from videos, in which relationships are both spatial and temporal. So far, video scene graphs have been mostly applied to cross-modal retrieval to find video fragments [47,38]. In this work, we rely on video scene graphs to generate unsupervised video scene descriptions.

Video Descriptions Video captioning aims to describe short video clips using natural language. Most approaches [53,26,3,21,39] use a sequential encoder-decoder framework, in which the input are visual features from multiple frames and the output is the generated sentence. For more detailed descriptions, dense video captioning [16,58] generates multiple sentences describing all the relevant events in the video. However, existing methods require to be trained on large-scale annotated datasets [29,27] with thousands of video-description pairs. We generate rich video scene descriptions in an unsupervised way using the semantic information from video scene graphs.

3 Model Overview

The goal of video story question answering is to understand movies or TV shows in a similar way as we humans do. We argue that there are at least three aspects of human intelligence involved in this task: 1) comprehension of what is being said, 2) comprehension of about what is being watched, and 3) recalling what happened in the story before. Our proposed model, ROLL, emulates each of those aspects in a three branch architecture, as shown in Fig. 2. Each branch in ROLL (read, observe, and recall) represents as text data the information from a different cognitive task, and encodes it through a Transformer with several self-attention layers. Then, the outputs from each Transformer are fused through a modality weighting mechanism to predict the correct answer.

Task definition We address video story question answering as a knowledge-based multiple-choice task. For each sample, the input is: 1) a question, 2) N_{ca} candidate answers, 3) a video scene, and 4) the subtitles associated with the scene. The output is the index of the predicted answer. As a knowledge-based task, models can access external resources to retrieve contextual information.

Introduction to Transformers Transformers [37] are sequence-to-sequence modelling architectures that entirely rely on self-attention mechanisms. They have rapidly become the state-of-the-art in many natural language processing tasks. ROLL incorporates three independent Transformers to model the language data extracted from each branch, which is represented by the input string:

$$s_m{}^c = [\text{CLS}] + \text{context}_m + [\text{SEP}] + \text{choice}_m{}^c + [\text{SEP}], \tag{1}$$

where m indicates the branch, context_m is an input sentence defined for each branch, $\text{choice}_m{}^c$ is a sentence for the c-th candidate answer with $c = 1, \cdots, N_{ca}$,

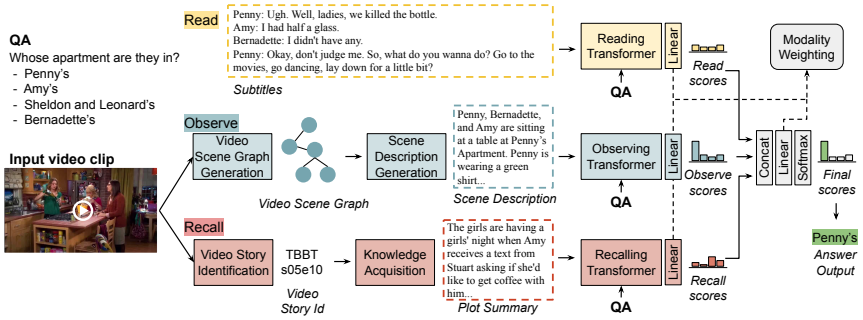

Fig. 2: ROLL overview. Each branch estimates a relevance score for each of the candidate answers based on different information. The read branch relies on subtitles, the observe branch generates unsupervised video descriptions, and the recall branch obtains external knowledge as plot summaries. To predict the correct answer, the three outputs are fused through a modality weighting mechanism.

[CLS] is the classification token used to obtain the output representation, [SEP] is the separator token for differentiating sentences, and + is string concatenation. For each sample, N_{ca} input strings are generated, one per candidate answer.

The input string ${s_m}^c$ is tokenised into a sequence of n tokens $\mathbf{x}^c = [x_1, \cdots, x_n]$, and fed into a Transformer network. For each token x_i in $\mathbf{x}^c$, the Transformer creates an input embedding, $\mathbf{h}_i^0 \in \mathbb{R}^{D_h}$ with D_h hidden size, by adding the word, segment, and position embeddings. For each self-attention layer $l = 1, \cdots, N_L$ in the Transformer, denoted by $\text{TBlock}^l(\cdot)$, the contextualised word representation for position i in the sequence is computed as:

$$\mathbf{h}_i^l = \text{TBlock}^l(\mathbf{h}_i^{l-1}) \tag{2}$$

The encoded representation of the input string ${s_m}^c$ is then obtained as the output of the position of the [CLS] token in the last layer:

$${\mathbf{y}_m}^c = \mathbf{h}_0^L \in \mathbb{R}^{D_h} \tag{3}$$

Our Transformers are the $\text{BERT}_{\text{BASE}}$ model [4] with $N_L = 12$ and $D_h = 768$.

4 Read Branch

In the read branch, ROLL extracts information from the dialogues of the video scene, which are obtained from the subtitles. The input string for this branch is:

$${s_r}^c = [\text{CLS}] + subs + q + [\text{SEP}] + a^c + [\text{SEP}], \tag{4}$$

where *subs* are the subtitles, q the question, and a^c with $c = 1, \cdots, N_{ca}$ each of the candidate answers. Each input string ${s_r}^c$ is fed into the Reading Transformer to obtain ${\mathbf{y}_r}^c$, which is forwarded into a single output linear layer with $\mathbf{w}_r$ weights and b_r bias, to compute a *read score* per candidate answer:

$${\alpha_r}^c = {\mathbf{w}_r}^\top \cdot {\mathbf{y}_r}^c + b_r \tag{5}$$

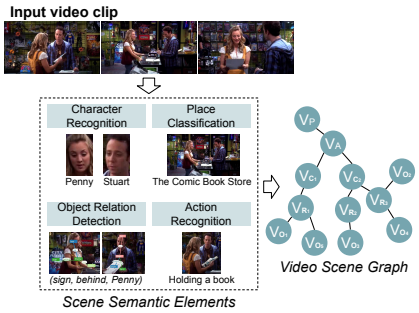

Fig. 3: Video scene graphs are generated from recognising the semantic elements of the scene (characters V_C, places V_P, objects V_O, relations V_R, and actions V_A) and connecting them.

Fig. 4: Examples of generated scene description. Although not natural, they accurately represents the semantics in the video scene.

5 Observe Branch

In the observe branch, ROLL summarises the semantics of the video scene into a video description. Generating descriptions from video is a challenging problem [49]. Standard video captioning models [3,21,39] require to be trained on large-scale datasets with annotated video and description pairs. As video story question answering datasets commonly do not provide such annotations, training a model for our task is impractical. Similarly, relying on pre-trained models may lead to poor results, as the generated descriptions will miss important information about the story (e.g., character names or frequent locations). Alternatively, we propose to generate unsupervised video descriptions by first creating a video scene graph. The descriptions are then fed into the Observing Transformer to predict a *observe score* for each candidate answer. Below, we first describe the video scene graph generation process, then we provide the details for the unsupervised video description, and finally we summarise the observing Transformer.

5.1 Video Scene Graph Generation

Fig. 3 shows the video scene graph generation process, which is built on top of state-of-the-art image and video recognition techniques. We use four modules to detect the most relevant details in the scene: character recognition, place classification, object relation detection, and action recognition. The video scene graph is then generated by building connections between the detected elements.
Character Recognition This module identifies the characters that appear in the scene using a face recognition classifier trained with images from the cast. We download about 10 images for each of the most common N_{C_T} characters based on IMDb.[3] We extract $f \in \mathbb{R}^{128}$ face representations with FaceNet [30]

[3] https://www.imdb.com/

and train a k-nearest neighbour (kNN) classifier, where $k = N_{C_T}$. At test time, the trained kNN classifier returns a score for the predicted character. If the score is below a threshold, we assigned it to the *unknown* class. Finally, we apply a spatio-temporal filter to remove mispredictions and duplicate characters. Details are provided in the suppl. material. As output, we obtain a set of N_C characters appearing in the scene $C = \{C_i | i = 1, \cdots, N_C\}$, and their bounding boxes.

Place Classification The place classification module detects where the scene is located. To learn the frequent locations in the video story, we fine-tune the pre-trained Places365 [57] network with ResNet50 [8] backbone in a weakly supervised way. To obtain place annotations, we use video transcripts from specialised websites.[4] We extract the locations that appear at least 10 times in the training set scripts, and include an *unknown* category for the rest. Training is performed at the frame level, i.e., each frame is considered as an independent image. For prediction, we accumulate the scores of the top 5 predicted classes for each frame in a video scene and output the most scored place, P.

Object Relation Detection This module detects the objects in the scene and their relations. We use the large-scale visual relationship understanding (VRU) [55] pre-trained on the VG200 dataset [48], with 150 object and 50 relation categories. For each frame, VRU returns a list of subject-relation-object triplets, their bounding boxes, and a prediction score for each triplet. We replace the objects and subjects assigned to a person class[5] with its corresponding character name by finding the overlap between the bounding boxes. We only keep triplets assigned to known characters and we filter out duplicates. After discarding the bounding boxes and scores, we obtain a list of N_T triplets, $T = \{T_i | i = 1, \cdots, N_T\}$ with $T_i = (S_i, R_i, O_i)$ and S_i, R_i, and O_i as subject, relation, and object.

Action Recognition The action recognition module detects the main action in the video scene. We use Long-Term Feature Banks (LFB) [44] pre-trained on the Charades dataset [33] with 157 action categories. LFB extracts information over the entire span of the video scene, improving performance with respect to using short 2-3 second clips. We input the entire scene into the network, and we obtain a predicted action as a result, A.

Graph Generation The video scene graph, $G = (V, E)$, semantically describes the visual contents of the scene by using a collection of nodes V, and edges E. We consider the following types of nodes:

- *Character nodes*, $V_C \subseteq V$, representing the characters in the scene. If C do not contain any *unknown* character, $V_C = C$. Otherwise, we remove the *unknown* characters $\{\text{UNK}_C\}$, as $V_c = C - \{\text{UNK}_C\}$.
- *Place nodes*, $V_P \subseteq V$, representing the location where the video scene occurs. $V_P = \{P\}$ if $P \neq$ *unknown*, otherwise $V_P = \emptyset$.
- *Object nodes*, $V_O \subseteq V$, representing the objects in the scene, which are obtained from the subjects and objects in the triplets that are not a character, as $V_O = Z - (Z \cap C)$ with $Z = S \cup O$.

[4] For example, https://bigbangtrans.wordpress.com/.

[5] Boy, girl, guy, lady, man, person, player, woman.

Table 1: Sentence generation from the video scene graph, $G = (V, E)$ with $V = \{V_C, V_P, V_O, V_R, V_A\}$. We define $e_{R_k,O} = \{e_{R_k,O_j}\}$ with $j \in [1, ..., |V_O|]$. $e_{R_k,C}$, e_{O,R_k}, and e_{C,R_k} are defined likewise.

Graph Condition	Generated Sentence	Example
$\|V_C\| = 0$ & $\|V_P\| = 0$	`Someone is` V_A.	Someone is lying on the floor.
$\|V_C\| = 1$ & $\|V_P\| = 0$	V_C `is` V_A.	Leonard is smiling.
$\|V_C\| > 1$ & $\|V_P\| = 0$	$V_{C_1}, ..., V_{C_{\|V_C\|-1}}$ `and` $V_{C_{\|V_C\|}}$ `are` V_A.	Penny and Amy are holding a bag.
$\|V_C\| = 0$ & $\|V_P\| = 1$	`Someone is` V_A `at` V_P.	Someone is walking at the street.
$\|V_C\| = 1$ & $\|V_P\| = 1$	V_C `is` V_A `at` V_P.	Sheldon is smiling at the bedroom.
$\|V_C\| > 1$ & $\|V_P\| = 1$	$V_{C_1}, ..., V_{C_{\|V_C\|-1}}$ `and` $V_{C_{\|V_C\|}}$ `are` V_A `at` V_P.	Amy and Raj are talking at the room.
$e_{C_i,R_k} \in E$ & $e_{R_k,O_j} \in E$ & $\|e_{R_k,O}\| = 1$	V_{C_i} V_{R_k} V_{O_j}.	Penny wearing shorts.
$e_{C_i,R_k} \in E$ & $e_{R_k,O_j} \in E$ & $\|e_{R_k,O}\| > 1$	V_{C_i} V_{R_k} $V_{O_1}, ..., V_{O_{\|V_O\|-1}}$ `and` $V_{O_{\|V_O\|}}$.	Raj holding bottle and book.
$e_{R_k,C_i} \in E$ & $e_{O_j,R_k} \in E$ & $\|e_{O,R_k}\| = 1$	V_{O_j} V_{R_k} V_{C_i}.	Board behind Sheldon.
$e_{R_k,C_i} \in E$ & $e_{O_j,R_k} \in E$ & $\|e_{O,R_k}\| > 1$	$V_{O_1}, ..., V_{O_{\|V_O\|-1}}$ `and` $V_{O_{\|V_O\|}}$ V_{R_k} V_{C_i}.	Chair, table and door behind Penny.

- *Relation nodes*, $V_R \subseteq V$, representing the relation between subjects and objects in the triplets, $V_R = R$.
- *Action nodes*, $V_A \subseteq V$, representing the action in the scene as $V_A = \{A\}$, with $|V_A| = 1$.

We use 6 types of directed edges:

- $e_{P,A} = (V_P, V_A) \in E$ between the place node V_P and the action node V_A.
- $e_{A,C_j} = (V_A, V_{C_j}) \in E$ between the action node V_A and each character V_{C_j}.
- $e_{C_i,R_j} = (V_{C_i}, V_{R_j}) \in E$ between a character node V_{C_i} and a relation node V_{R_j} when $V_{C_i} = S_k$ and $V_{R_j} = R_k$ in the triplet $T_k = (S_k, R_k, O_k)$.
- $e_{R_i,C_j} = (V_{R_i}, V_{C_j}) \in E$ between a relation node V_{R_i} and a character node V_{C_j} when $V_{R_i} = R_k$ and $V_{C_j} = O_k$ in the triplet $T_k = (S_k, R_k, O_k)$.
- $e_{O_i,R_j} = (V_{O_i}, V_{R_j}) \in E$ between an object node V_{O_i} and a relation node V_{R_j} when $V_{O_i} = S_k$ and $V_{R_j} = R_k$ in the triplet $T_k = (S_k, R_k, O_k)$.
- $e_{R_i,O_j} = (V_{R_i}, V_{O_j}) \in E$ between a relation node V_{R_i} and an object node V_{C_j} when $V_{R_i} = R_k$ and $V_{C_j} = O_k$ in the triplet $T_k = (S_k, R_k, O_k)$.

with i, j, and k being the index for a certain object in a set.

5.2 Scene Description Generation

Scene descriptions are generated from the video scene graph according to the set of rules in Table 1 in an unsupervised manner. For each true condition in Table 1, a single sentence is generated. The final scene description is the concatenation of all the generated sentences, which serves as a representation of the semantic content in the video scene. Examples are shown in Fig. 4.

5.3 Observing Transformer

The generated description, d, is used in the input string for the observe branch:

$$s_o{}^c = [\text{CLS}] + d + q + [\text{SEP}] + a^c + [\text{SEP}], \tag{6}$$

Each $s_o{}^c$ is fed into the Observing Transformer to obtain $\mathbf{y}_o{}^c$, which is forwarded into a single output linear layer to compute the *observe score*:

$$\alpha_o{}^c = \mathbf{w}_o{}^\top \cdot \mathbf{y}_o{}^c + b_o \tag{7}$$

6 Recall Branch

In the recall branch, ROLL emulates the human experience of watching a TV show by recalling the events that occurred previously. This is inspired by the human evaluation on [5], which provides some insights on human behaviour. In [5], evaluators were asked to answer questions about a popular sitcom under different conditions. Interestingly, the reported performance dropped dramatically when humans were not exposed to the videos. We speculate that this is because humans indirectly used the scene to remember the whole episode and answer questions about the plot. The recall branch imitates this behaviour by first identifying the video and then acquiring knowledge about the story plot.

6.1 Knowledge Acquisition

Differently from previous work [5], in which the external knowledge to answer each question is specifically annotated by humans, we rely on publicly available resources[6] and build a knowledge base (KB) using plot summaries from the Internet.[7] Given a video scene, we first identify the video story it belongs to as in video retrieval [6]. Frames are represented by the output of the second-to-last layer of a pre-trained ResNet50 [8]. We compute the cosine similarity between each frame representation in the scene and all frames in the dataset, keeping the video of the most similar frame. As a result, we obtain an identifier of the most voted video, which is used to query the KB and we obtain a document p with the plot. In this way, ROLL acquires external knowledge about the video story in an weakly supervised way as 1) the questions and the external knowledge base have not been paired in any way during their generation, 2) the model does not know if there is corresponding text in the external knowledge base that can be useful for a given question, 3) the model is not directly trained with ground-truth episode labels, and 4) the model is not trained with ground-truth text location.

6.2 Recalling Transformer

The document p is fed into the Recalling Transformer to predict a *recall score* for each candidate answer. As many documents exceed the maximum number of words the Transformer can take as input,[8] we adopt a sliding window approach [9,10] to slice p into multiple overlapping segments. To produce the segments k_j with $j = 1, \cdots, N_{s_{\text{MAX}}}$, we slide a window of length W_l with a stride r over the document p, obtaining $N_s = \lceil \frac{L_d - W_l}{r} \rceil + 1$ segments, where L_d is the number of words in the document. For training multiple samples in a minibatch, we set all the documents to have the same number of segments $N_{s_{\text{MAX}}}$, discarding segments

[6] For example, https://the-big-bang-theory.com/

[7] Generating video plot summaries automatically from the whole video story is a challenging task by itself and out of the scope of this work. However, it is an interesting problem that we aim to study as a our future work.

[8] In The Big Bang Theory, the longest summary contains 1,605 words.

if $N_s > N_{s_{\text{MAX}}}$, and zero-padding if $N_s < N_{s_{\text{MAX}}}$. We encode the plot segments along with the question and candidate answers into multiple input strings:

$$s_{ll}{}_j^c = [\text{CLS}] + q + [\text{SEP}] + a^c + k_j + [\text{SEP}] \tag{8}$$

Each $s_{ll}{}_j^c$ is fed into the Recalling Transformer to obtain $\mathbf{y}_{ll}{}_j^c$, which is forwarded into a single output linear layer to compute a score for an answer-segment pair:

$$\alpha_{ll}{}_j^c = \mathbf{w}_{ll}{}^\top \cdot \mathbf{y}_{ll}{}_j^c + b_{ll} \tag{9}$$

Then, the final recall score for each of the candidate answers $\alpha_{ll}{}^c$ is:

$$\alpha_{ll}{}^c = \max(\alpha_{ll}{}_j^c) \quad \text{with } j = 1, \cdots, N_{s_{\text{MAX}}} \tag{10}$$

7 Final Prediction

To output the final prediction score, the model concatenates the output of the three branches into a score vector $\boldsymbol{\alpha}^c = [\alpha_r{}^c, \alpha_o{}^c, \alpha_{ll}{}^c]$, which is input into a single layer classifier. The predicted answer $\hat{a}$ is then:

$$\omega^c = \mathbf{w}_c{}^\top \cdot \boldsymbol{\alpha}^c + b_c \tag{11}$$

$$\hat{a} = a^{\arg\max_c \boldsymbol{\omega}} \quad \text{with } \boldsymbol{\omega} = [\omega^1, \cdots, \omega^{N_{ca}}]^\top \tag{12}$$

Modality Weighting Mechanism Wang et al. [43] have shown that multi-modality training often suffers from information loss, degradating performance with respect to single modality models. To avoid losing information when merging the three branches in ROLL, we use a modality weighting (MW) mechanism. First, we ensure that each Transformer learns independent representations by training them independently. The multi-class cross-entropy loss is computed as:

$$\mathcal{L}(\boldsymbol{\delta}, c^*) = -\log \frac{\exp(\delta^{c^*})}{\sum_c \exp(\delta^c)} \tag{13}$$

where c^* is the correct answer, and $\boldsymbol{\delta} = [\delta^1, \cdots, \delta^{N_{ca}}]$ the vector with the scores of the candidate answers. Next, the Transformers are frozen and the three branches are fine-tuned together. To ensure the multi-modal information is not lost, the model is trained as a multi-task problem with $\beta_r + \beta_o + \beta_{ll} + \beta_\omega = 1$:

$$\mathcal{L}_{\text{MW}} = \beta_r \mathcal{L}(\boldsymbol{\alpha}_r, c^*) + \beta_o \mathcal{L}(\boldsymbol{\alpha}_o, c^*) + \beta_{ll} \mathcal{L}(\boldsymbol{\alpha}_{ll}, c^*) + \beta_\omega \mathcal{L}(\boldsymbol{\omega}, c^*)$$

8 Evaluation

Datasets We evaluate ROLL on the KnowIT VQA [5] and the TVQA+ [18] datasets. KnowIT VQA is the only dataset for knowledge-based video story question answering, containing 24,282 questions about 207 episodes of The Big

Table 2: Evaluation on KnowIT VQA test set.

		Data			Accuracy				
Method	**Encoder**	**Dialog**	**Vision**	**Know.**	**Vis.**	**Text.**	**Temp.**	**Know.**	**All**
Rookies [5]	-	-	-	No	0.936	0.932	0.624	0.655	0.748
Masters [5]	-	-	-	Yes	0.961	0.936	0.857	0.867	0.896
TVQA [17]	LSTM	Subs.	Concepts	-	0.612	0.645	0.547	0.466	0.522
ROCK Img [5]	BERT	Subs.	ResNet	Human	0.654	0.681	0.628	0.647	0.652
ROCK Cpts [5]	BERT	Subs.	Concepts	Human	0.654	0.685	0.628	0.646	0.652
ROCK Faces [5]	BERT	Subs.	Characters	Human	0.654	0.688	0.628	0.646	0.652
ROCK Caps [5]	BERT	Subs.	Captions	Human	0.647	0.678	0.593	0.643	0.646
ROLL-human	BERT	Subs.	Descriptions	Human	0.708	**0.754**	0.570	0.567	0.620
ROLL	BERT	Subs.	Descriptions	Summaries	**0.718**	0.739	**0.640**	**0.713**	**0.715**

Bang Theory TV show. Questions in the test set are divided into four categories: visual-based, textual-based, temporal-based, and knowledge-based, and each question is provided with $N_{ca} = 4$ candidate answers. Accuracy is computed as the number of correct predicted answers over the total number of questions. Even though our model is specifically designed for leveraging external knowledge, we also evaluate its generalisation performance on non knowledge-based video story question answering. For this purpose, we use the TVQA+ dataset, in which questions are compositional and none of them requires external knowledge. TVQA+ contains 29,383 questions, each with $N_{ca} = 5$ candidate answers.

Implementation Details We use the BERT uncased base model with pre-trained initialisation for our three Transformers. The maximum number of tokens is set to 512. For the single branch training, transformers are fine-tuned following the details in [4]. For the joint model training, we use stochastic gradient descent with momentum 0.9 and learning rate 0.001. In the observe branch, we extract the frames for the Character Recognition, Place Classification and Object Relation Detection modules at 1 fps, and for the Action Recognition module at 24 fps. In total, we use 17 characters, 32 places, 150 objects, 50 relations, and 157 action categories. In the recall branch, we use a window length $W_l = 200$, stride $r = 100$, and maximum number of segments $N_{s_{\text{MAX}}} = 5$. In the modality weighting mechanism, we set $\beta_r = 0.06$, $\beta_o = 0.06$, $\beta_{ll} = 0.08$, and $\beta_\omega = 0.80$ unless otherwise stated.

Evaluation on KnowIT VQA We compare ROLL against the latest reported results on the KnowIT VQA dataset: TVQA and four different models in ROCK. TVQA [17] is based on a two-stream LSTM encoder for subtitles and visual concepts, whereas ROCK [5] uses task-specific human annotations to inform a BERT based model with external knowledge. ROCK reports results using four different visual representations: ResNet features, visual concepts, list of characters, and generated captions. For a more complete comparison, we also report results of ROLL using the human annotations from [5] as external knowledge (ROLL-human). Results are found in Table 2. Main findings are summarised as:

1. Overall, ROLL outperforms previous methods in all the question categories by a large margin, with 6.3% improvement on the overall accuracy with respect to the best performing ROCK.

Method	Vision	Lang.	Acc
TVQA [17]	Concepts	LSTM	62.28
TVQA [17]	Regional	LSTM	62.25
STAGE [18]	Regional	GloVe	67.29
STAGE [18]	Regional	BERT	68.31
ROLL	Description	BERT	**69.61**

Table 3: Evaluation on the TVQA+ val set. No external knowledge is used.

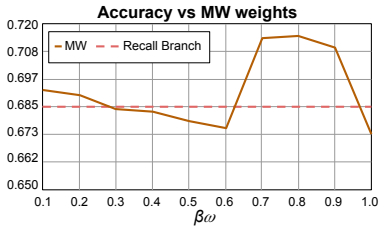

Fig. 5: ROLL accuracy according to β_ω on KnowIT VQA test set.

2. When comparing the visual representations, our proposed video descriptions contain more semantic information than previous methods, improving visual-based questions by at least 6.4%. Specially, the boost in performance in visual-based questions with respect to standard captioning (ROCK Captions) or visual concepts (ROCK Concepts, TVQA) validates our unsupervised video descriptions as the best representation for this task.
3. Additional evidence of the superior performance of our proposed unsupervised descriptions is shown when ROLL uses human annotations as external knowledge. Although the overall performance is lower because ROLL-human is not optimised to exploit this kind of information, on the visual-based questions our method improves best previous work by 5.4%. As the same source of knowledge is used, the superior performance can only be due to the contribution of our proposed visual representations.
4. On the knowledge-based samples, our method based on plot summaries outperforms task-specific human annotations by 6.7%, even when less annotations are required. This implies that the proposed slicing mechanism in the recall branch successfully extracts the relevant information from the long documents provided as external knowledge.
5. When compared against human performance, ROLL is 18% behind masters accuracy (humans that have watched the show) and it is closer to rookies non-knowledge accuracy (humans that have never watched the show). This shows how challenging this task is, with still plenty room for improvement.

Evaluation on TVQA+ To show ROLL generalisation performance even when external knowledge is not necessary, we additionally evaluate it on the TVQA+ dataset. For a fair comparison against previous work, 1) we remove the recall branch in ROLL and only use the read and observe branches, i.e., no external knowledge is used, and 2) we compare ROLL against models that use the answer labels as the only supervision for training, i.e., no extra annotations such as timestamps or spatial bounding boxes are used. Results are found in Table 3. Consistent with the results on the KnowIT VQA dataset, ROLL based on scene descriptions outperforms models based on other visual representations, such as visual concepts or Faster R-CNN [28] regional features, by at least 1.3%.

Ablation study We perform an ablation study to measure the contribution of each branch. Results when using one, two, or the three branches on the KnowIT

Table 4: ROLL ablation study.

Branch	Vis.	Text.	Temp.	Know.	All
Read	0.656	**0.772**	0.570	0.525	0.584
Observe	0.629	0.424	0.558	0.514	0.530
Recall	0.624	0.620	0.570	**0.725**	0.685
Read-Observe	0.695	0.732	0.570	0.527	0.590
Observe-Recall	0.712	0.601	0.628	0.704	0.691
Read-Recall	**0.722**	0.732	0.628	0.708	0.711
Full Model	0.718	0.739	**0.640**	0.713	**0.715**

Table 5: Fusion Methods Comparison.

Method	Vis.	Text.	Temp.	Know.	All
Average	0.726	0.710	0.628	0.648	0.672
Maximum	0.685	0.757	0.593	0.678	0.686
Self-att	**0.737**	**0.761**	**0.651**	0.641	0.677
QA-att	0.736	0.743	0.605	0.637	0.670
FC w/o MW	0.728	0.743	0.616	0.637	0.669
FC w/ MW	0.718	0.739	0.640	**0.713**	**0.715**

VQA dataset are reported in Table 4. When a single branch is used, the observe branch gets the worst overall accuracy and the recall branch performs the best. This is consistent with the types of questions in the dataset, with 22% being visual-based and 63% being knowledge-based. The read branch gets the best performance in the text-based questions (i.e., about the subtitles), and the recall branch gets the best accuracy in the knwoledge-based questions (i.e., about the storyline). When the observe branch is combined with other branches it consistently contributes to improve the results. Again, this result strongly suggests that the generated scene descriptions do contain meaningful information for the task. The full model combining the three branches performs the best.

Fusion Methods Comparison We also study the performance of our proposed MW mechanism and compare it against several fusion methods. Results are reported in Table 5. Given the three prediction scores from each of the branches, Average and Maximum compute the average and maximum score, respectively. The Self-att method implements a self-attention mechanism based on the Transformer outputs, and the QA-att mechanism attends each of the modality predictions based on the BERT representation of the question and candidate answers. The FC w/o MW predicts the answer scores by concatenating the scores of the three branches and feeding them into a linear layer, and FC w/ MW builds our proposed MW mechanism on top. The results show that most of the methods fail at properly fusing the information from the three branches, i.e., the overall performance is lower than the best single branch (recall, as reported in Table 4). This is probably because the fusion of the different modalities incurs in information loss. Our MW mechanism, in contrast, successfully balances the contribution from the three branches. Fig. 5 compares different values of β_ω in the MW against the best performing single modality, with β_r, β_o, and β_{ll} uniformly distributed. When the MW is not used ($\beta_\omega = 1$) the model obtains the worst performance. Likewise, when the loss contribution from the final prediction is too weak ($\beta_\omega < 0.6$), the model is not able to fuse the information correctly.

Qualitative Results We visually inspect ROLL results to understand the strengths and weaknesses of our model. An example of scene graph can be seen in Fig. 7, whereas results on visual-based questions are provided in Fig. 6. ROLL performs well on questions related to the general content of the scene, such as places, people, or objects, but fails in detecting fine-grained details. Performance of the individual video modules is reported in the supplementary material.

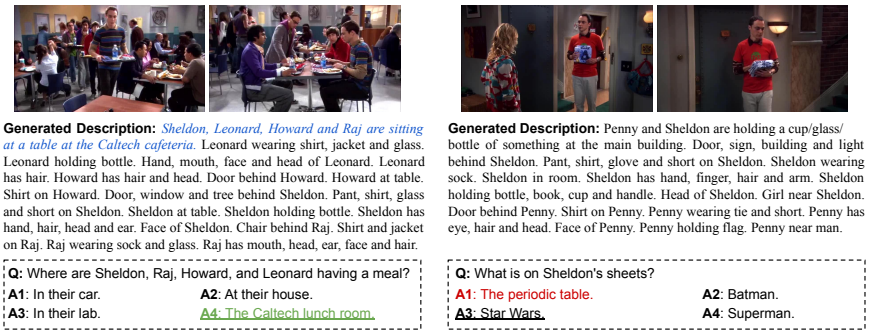

Fig. 6: ROLL visual results. Underline/colour for correct/predicted answers. The relevant part for the question in the generated description is highlighted in blue.

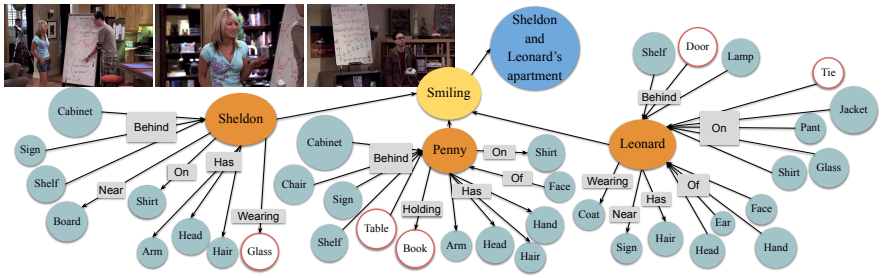

Fig. 7: Generated scene graph. Solid for correct and bordered for incorrect nodes.

9 Conclusion

We introduced ROLL, a model for knowledge-based video story question answering. To extract the visual information from videos, ROLL generates video descriptions in an unsupervised way by relying on video scene graphs. This new video representation led the model to an important increase of accuracy on visual-based questions on two datasets. Moreover, unlike previous work, ROLL leverages information from external knowledge without specific annotations on the task, easing the requirements of human labelling. This came without a drop in performance. On the contrary, as ROLL successfully fuses specific details from the scene with general information about the plot, the accuracy in KnowIT VQA and TVQA+ datasets was improved by more than 6.3% and 1.3%, respectively. Finally, by incorporating a modality weighting mechanism, ROLL avoided the information loss that comes from fusing different sources.

Acknowledgement This work was supported by a project commissioned by the New Energy and Industrial Technology Development Organization (NEDO), and JSPS KAKENHI Nos. 18H03264 and 20K19822. We also would like to thank the anonymous reviewers for they insightful comments to improve the paper.

References

1. Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, L.C., Parikh, D.: VQA: Visual question answering. In: Proc. ICCV. pp. 2425–2433 (2015)
2. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: A nucleus for a web of open data. In: The Semantic Web, pp. 722–735. Springer (2007)
3. Chen, Y., Wang, S., Zhang, W., Huang, Q.: Less is more: Picking informative frames for video captioning. In: Proc. ECCV. pp. 358–373 (2018)
4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proc. NAACL. pp. 4171–4186 (2019)
5. Garcia, N., Otani, M., Chu, C., Nakashima, Y.: KnowIT VQA: Answering knowledge-based questions about videos. In: Proc. AAAI (2020)
6. Garcia, N., Vogiatzis, G.: Asymmetric spatio-temporal embeddings for large-scale image-to-video retrieval. In: BMVC (2018)
7. Gu, J., Joty, S., Cai, J., Zhao, H., Yang, X., Wang, G.: Unpaired image captioning via scene graph alignments. In: Proc. ICCV (2019)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proc. CVPR. pp. 770–778 (2016)
9. Hewlett, D., Jones, L., Lacoste, A., Gur, I.: Accurate supervised and semi-supervised machine reading for long documents. In: Proc. EMNLP. pp. 2011–2020 (2017)
10. Hu, M., Peng, Y., Huang, Z., Li, D.: Retrieve, Read, Rerank: Towards end-to-end multi-document reading comprehension. In: Proc. ACL. pp. 2285–2295 (2019)
11. Johnson, J., Hariharan, B., van der Maaten, L., Fei-Fei, L., Zitnick, L.C., Girshick, R.: CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In: Proc. CVPR. pp. 2901–2910 (2017)
12. Johnson, J., Krishna, R., Stark, M., Li, L.J., Shamma, D., Bernstein, M., Fei-Fei, L.: Image retrieval using scene graphs. In: Proc. CVPR. pp. 3668–3678 (2015)
13. Kim, J., Ma, M., Kim, K., Kim, S., Yoo, C.D.: Progressive attention memory network for movie story question answering. In: Proc. CVPR. pp. 8337–8346 (2019)
14. Kim, K.M., Choi, S.H., Kim, J.H., Zhang, B.T.: Multimodal dual attention memory for video story question answering. In: Proc. ECCV. pp. 673–688 (2018)
15. Kim, K.M., Heo, M.O., Choi, S.H., Zhang, B.T.: DeepStory: Video story QA by deep embedded memory networks. In: Proc. IJCAI. pp. 2016–2022 (2017)
16. Krishna, R., Hata, K., Ren, F., Fei-Fei, L., Carlos Niebles, J.: Dense-captioning events in videos. In: Proc. ICCV. pp. 706–715 (2017)
17. Lei, J., Yu, L., Bansal, M., Berg, T.L.: TVQA: Localized, compositional video question answering. In: Proc. EMNLP. pp. 1369–1379 (2018)
18. Lei, J., Yu, L., Berg, T.L., Bansal, M.: TVQA+: Spatio-temporal grounding for video question answering. arXiv preprint arXiv:1904.11574 (2019)
19. Li, Y., Ouyang, W., Zhou, B., Wang, K., Wang, X.: Scene graph generation from objects, phrases and region captions. In: Proc. ICCV. pp. 1261–1270 (2017)
20. Liang, J., Jiang, L., Cao, L., Li, L.J., Hauptmann, A.G.: Focal visual-text attention for visual question answering. In: Proc. CVPR. pp. 6135–6143 (2018)
21. Liu, S., Ren, Z., Yuan, J.: SibNet: Sibling convolutional encoder for video captioning. In: Proc. ACM Multimedia. pp. 1425–1434 (2018)
22. Marino, K., Rastegari, M., Farhadi, A., Mottaghi, R.: OK-VQA: A visual question answering benchmark requiring external knowledge. In: Proc. CVPR. pp. 3195–3204 (2019)

23. Na, S., Lee, S., Kim, J., Kim, G.: A read-write memory network for movie story understanding. In: Proc. ICCV. pp. 677–685 (2017)
24. Narasimhan, M., Lazebnik, S., Schwing, A.: Out of the box: Reasoning with graph convolution nets for factual visual question answering. In: Proc. NIPS. pp. 2659–2670 (2018)
25. Narasimhan, M., Schwing, A.G.: Straight to the facts: Learning knowledge base retrieval for factual visual question answering. In: Proc. ECCV. pp. 451–468 (2018)
26. Pan, Y., Mei, T., Yao, T., Li, H., Rui, Y.: Jointly modeling embedding and translation to bridge video and language. In: Proc. CVPR. pp. 4594–4602 (2016)
27. Pini, S., Cornia, M., Bolelli, F., Baraldi, L., Cucchiara, R.: M-VAD Names: A dataset for video captioning with naming. Multimedia Tools and Applications **78**(10), 14007–14027 (2019)
28. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: Proc. NIPS. pp. 91–99 (2015)
29. Rohrbach, A., Rohrbach, M., Tandon, N., Schiele, B.: A dataset for movie description. In: Proc. CVPR. pp. 3202–3212 (2015)
30. Schroff, F., Kalenichenko, D., Philbin, J.: FaceNet: A unified embedding for face recognition and clustering. In: Proc. CVPR. pp. 815–823 (2015)
31. Shah, S., Mishra, A., Yadati, N., Talukdar, P.P.: KVQA: Knowledge-aware visual question answering. In: Proc. AAAI (2019)
32. Shi, J., Zhang, H., Li, J.: Explainable and explicit visual reasoning over scene graphs. In: Proc. CVPR. pp. 8376–8384 (2019)
33. Sigurdsson, G.A., Varol, G., Wang, X., Farhadi, A., Laptev, I., Gupta, A.: Hollywood in homes: Crowdsourcing data collection for activity understanding. In: Proc. ECCV. pp. 510–526 (2016)
34. Speer, R., Chin, J., Havasi, C.: ConceptnNet 5.5: An open multilingual graph of general knowledge. In: Porc. AAAI (2017)
35. Tapaswi, M., Zhu, Y., Stiefelhagen, R., Torralba, A., Urtasun, R., Fidler, S.: MovieQA: Understanding stories in movies through question-answering. In: Proc. CVPR. pp. 4631–4640 (2016)
36. Teney, D., Liu, L., van den Hengel, A.: Graph-structured representations for visual question answering. In: Proc. CVPR. pp. 1–9 (2017)
37. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Proc. NIPS. pp. 5998–6008 (2017)
38. Vicol, P., Tapaswi, M., Castrejon, L., Fidler, S.: MovieGraphs: Towards understanding human-centric situations from videos. In: Proc. CVPR. pp. 8581–8590 (2018)
39. Wang, B., Ma, L., Zhang, W., Jiang, W., Wang, J., Liu, W.: Controllable video captioning with POS sequence guidance based on gated fusion network. In: Proc. ICCV. pp. 2641–2650 (2019)
40. Wang, B., Xu, Y., Han, Y., Hong, R.: Movie question answering: Remembering the textual cues for layered visual contents. In: Proc. AAAI (2018)
41. Wang, P., Wu, Q., Shen, C., Dick, A., van den Hengel, A.: FVQA: Fact-based visual question answering. IEEE Trans. PAMI **40**(10), 2413–2427 (2018)
42. Wang, P., Wu, Q., Shen, C., Dick, A., Van Den Henge, A.: Explicit knowledge-based reasoning for visual question answering. In: Proc. IJCAI. pp. 1290–1296 (2017)
43. Wang, W., Tran, D., Feiszli, M.: What makes training multi-modal networks hard? In: Proc. CVPR. pp. 12695–12705 (2020)

44. Wu, C.Y., Feichtenhofer, C., Fan, H., He, K., Krahenbuhl, P., Girshick, R.: Long-term feature banks for detailed video understanding. In: Proc. CVPR. pp. 284–293 (2019)
45. Wu, Q., Wang, P., Shen, C., Dick, A., van den Hengel, A.: Ask me anything: Free-form visual question answering based on knowledge from external sources. In: Proc. CVPR. pp. 4622–4630 (2016)
46. Xiong, P., Zhan, H., Wang, X., Sinha, B., Wu, Y.: Visual query answering by entity-attribute graph matching and reasoning. In: Proc. CVPR. pp. 8357–8366 (2019)
47. Xiong, Y., Huang, Q., Guo, L., Zhou, H., Zhou, B., Lin, D.: A graph-based framework to bridge movies and synopses. In: Proc. ICCV. pp. 4592–4601 (2019)
48. Xu, D., Zhu, Y., Choy, C.B., Fei-Fei, L.: Scene graph generation by iterative message passing. In: Proc. CVPR. pp. 5410–5419 (2017)
49. Xu, J., Mei, T., Yao, T., Rui, Y.: MSR-VTT: A large video description dataset for bridging video and language. In: Proc. CVPR. pp. 5288–5296 (2016)
50. Yang, J., Lu, J., Lee, S., Batra, D., Parikh, D.: Graph R-CNN for scene graph generation. In: Proc. ECCV. pp. 670–685 (2018)
51. Yang, X., Tang, K., Zhang, H., Cai, J.: Auto-encoding scene graphs for image captioning. In: Proc. CVPR. pp. 10685–10694 (2019)
52. Yang, Z., Garcia, N., Chu, C., Otani, M., Nakashima, Y., Takemura, H.: BERT representations for video question answering. In: Proc. WACV (2020)
53. Yao, L., Torabi, A., Cho, K., Ballas, N., Pal, C., Larochelle, H., Courville, A.: Describing videos by exploiting temporal structure. In: Proc. ICCV. pp. 4507–4515 (2015)
54. Zellers, R., Yatskar, M., Thomson, S., Choi, Y.: Neural motifs: Scene graph parsing with global context. In: Proc. CVPR. pp. 5831–5840 (2018)
55. Zhang, J., Kalantidis, Y., Rohrbach, M., Paluri, M., Elgammal, A., Elhoseiny, M.: Large-scale visual relationship understanding. In: Proc. AAAI. vol. 33, pp. 9185–9194 (2019)
56. Zhang, J., Shih, K.J., Elgammal, A., Tao, A., Catanzaro, B.: Graphical contrastive losses for scene graph parsing. In: Proc. CVPR. pp. 11535–11543 (2019)
57. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: A 10 million image database for scene recognition. IEEE Trans. PAMI (2017)
58. Zhou, L., Zhou, Y., Corso, J.J., Socher, R., Xiong, C.: End-to-end dense video captioning with masked transformer. In: Proc. CVPR. pp. 8739–8748 (2018)