On the hidden treasure of dialog in video question answering

Deniz Engin[1,2] François Schnitzler[2] Ngoc Q. K. Duong[2] Yannis Avrithis[1]
[1]Inria, Univ Rennes, CNRS, IRISA [2]InterDigital

Abstract

High-level understanding of stories in video such as movies and TV shows from raw data is extremely challenging. Modern video question answering (VideoQA) systems often use additional human-made sources like plot synopses, scripts, video descriptions or knowledge bases. In this work, we present a new approach to understand the whole story without such external sources. The secret lies in the dialog: unlike any prior work, we treat dialog as a noisy source to be converted into text description via dialog summarization*, much like recent methods treat video. The input of each modality is encoded by transformers independently, and a simple fusion method combines all modalities, using soft temporal attention for localization over long inputs. Our model outperforms the state of the art on the KnowIT VQA dataset by a large margin, without using question-specific human annotation or human-made plot summaries. It even outperforms human evaluators who have never watched any whole episode before. Code is available at* https://engindeniz.github.io/dialogsummary-videoqa

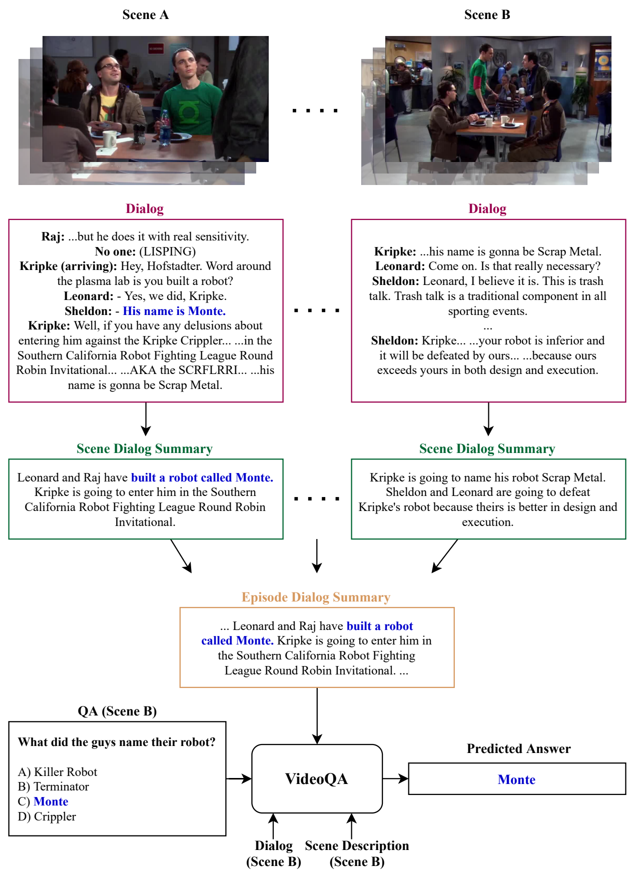


Figure 1: In VideoQA, a question is associated with Scene B, but it can only be answered by information from Scene A. We generate episode dialog summaries from subtitles and give them as input to our VideoQA system, dispensing with the need for external knowledge.

1. Introduction

Deep learning has accelerated progress in vision and language tasks. *Visual-semantic embeddings* [18, 9] have allowed zero-shot learning, cross-modal retrieval and generating new descriptions from embeddings. *Image captioning* [33] and *visual question answering* (VQA) [2] have demonstrated generation of realistic natural language description of images and a great extent of multimodal semantic understanding. The extension to *video captioning* [19, 32] and *video question answering* (VideoQA) [29, 20] has enabled further progress because video requires a higher level of reasoning to understand complex events [37].

VideoQA systems typically have similar architecture focusing on multimodal embeddings/description, temporal attention and localization, multimodal fusion and reasoning. While it is often hard to isolate progress in individual components, there are some clear trends. For instance, custom self-attention and memory mechanisms for fusion and reasoning [24, 17, 7] are gradually being streamlined by using *transformer* architectures [30, 16, 36]; while visual embeddings [29] are being replaced by semantic embeddings [20] and *text descriptions* by captioning [14, 3].

Datasets are essential for progress in the field, but often introduce bias. For instance, questions from text summaries are less relevant to visual information [29]; supervised temporal localization [20] biases system design towards two-stage localization→answering [21, 16]; fixed question structure focusing on temporal localization [20] often results in mere *alignment* of questions with subtitles and *matching* answers with the discovered context [14], providing little progress on the main objective, which is to study the level of understanding.

Bias can be removed by removing localization supervision and balancing questions over different aspects of comprehension, for instance visual, textual, or semantic [11]. However, the requirement of external knowledge, which can be in the form of hints or even ground truth, does not leave much progress in inferring such knowledge from raw data [11]. Even weakening this requirement to plain text *human-generated summaries* [10], still leaves a system unusable in the absence of such data.

In many cases, as illustrated in Figure 1, a question on some part of a story may require knowledge that can be recovered from dialog in other parts of the story. However, despite being textual, raw dialog is often informal and repetitive; searching over all available duration of such noisy source is error-prone and impractical. Inspired by the trend of video captioning, we go a step further and apply the same idea to *dialog*: We *summarize* raw dialog, converting it into *text description* for question answering.

Our finding is astounding: our dialog summary is not only a valid replacement for human-generated summary in handling questions that require knowledge on a whole story, but it outperforms them by a large margin.

Our contributions can be summarized as follows:

1. We apply *dialog summarization* to video question answering for the first time (Subsection 5.1).
2. Building on a modern VideoQA system, we convert all input sources into *plain text description*.
3. We introduce a weakly-supervised *soft temporal attention* mechanism for localization (Subsection 6.2).
4. We devise a very simple *multimodal fusion* mechanism that has no hyperparameters (Section 7).
5. We set a new state of the art on KnowIT VQA dataset [11] and we beat non-expert humans for the first time, working only with raw data (Section 8).

2. Related Work

Video Question Answering Progress on video question answering has been facilitated and driven by several datasets and benchmarks. VideoQA by Tapaswi *et al*. [29] addresses answering questions created from *plot synopses* using a variety of input sources, including video, subtitles, scene descriptions, scripts and the plot synopses themselves. Methods experimenting on MovieQA focus on *memory networks* capturing information from the *whole movie* by videos and subtitles [24, 15], scene-based memory attention networks to learn joint representations of frames and captions [17], and LSTM-based sequence encoders to learn visual-text embeddings [23].

TVQA [20] and TVQA+ [21] address *scene-based* questions containing *temporal localization* of the answer in TV shows, using video and subtitles. The questions are structured in two parts: one specifying a temporal location in the scene and the other requesting some information from that location. This encourages working with more than one modalities. Methods experimenting on these datasets focus on temporal localization and attention [21, 16], *captioning* [14, 3] and *transformer*-based pipelines capturing visual-semantic and language information [36, 30].

KnowIT VQA [11] is a *knowledge-based* dataset, including questions related to the scene, the episode or the entire story of a TV show, as well as *knowledge annotation* required to address certain questions, in the form of hints. *Transformer*-based methods are proposed to address this task by employing knowledge annotation [11] or external human-generated *plot summaries* [10]. Our method differs in substituting human-generated knowledge by summaries automatically generated from raw dialog.

Dialog Summarization Dial2Desc dataset [25] addresses generating *high-level short descriptions from dialog* using a transformer-based text generator. SAMSum corpus [12] is a human-annotated dialog summarization dataset providing speaker information. Methods experimenting on this dataset include existing *document summarization* methods [12], *graph neural networks* integrating cross-sentence information flow [39] and graph construction from utterance and commonsense knowledge [8]. Since dialog differs from structured text and requires extraction of the conversation structure, recent work focuses on representing the dialog from different *views* by sequence to sequence models [4]. We follow this approach.

3. Overview

We address knowledge-based video question answering on TV shows. Each episode is split in *scenes*. For each scene, we are given the *video* (frames) and *dialog* (speaker names followed by subtitle text) and a number of *multiple-choice questions*. Certain questions require high-level understanding of the whole episode or show. Garcia *et al*. [10] rely on human-generated *plot summaries* (or *plot* for short), which we use only for comparison. Our objective is to extract the required knowledge from raw data.

As shown in Figure 2, we first convert inputs into *plain text description*, including both video (by visual recognition) and dialog (by summarization) (Section 5). A number of separate *streams* then map text to embeddings, at the level of both *scene* (video and scene dialog summary) and *episode* (episode dialog summary and plot). The ques-

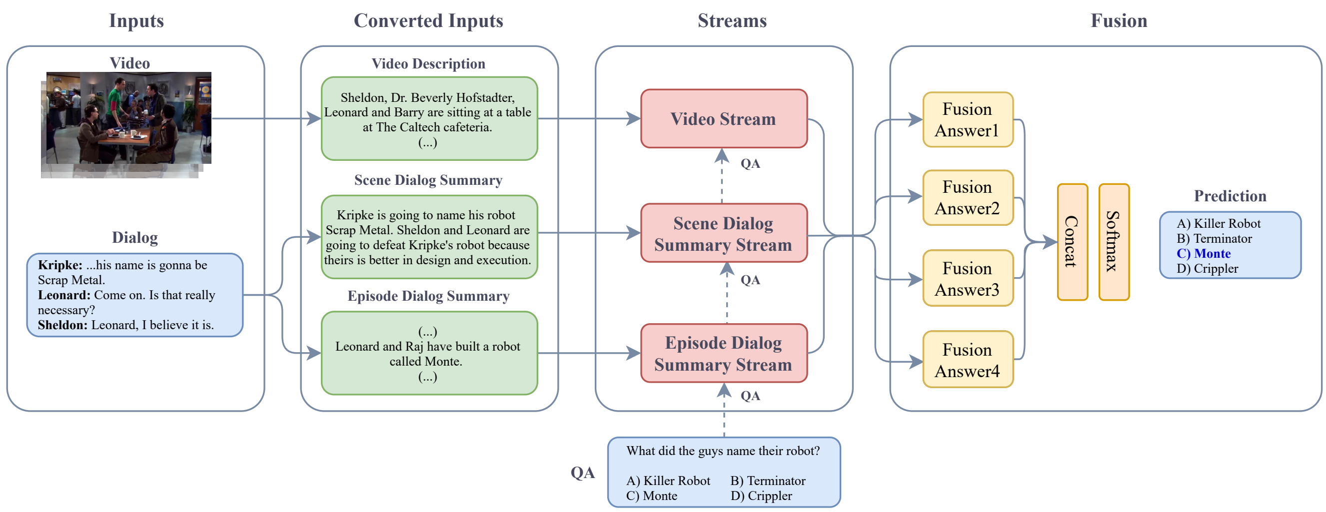


Figure 2: Our VideoQA system converts both video and dialog to text descriptions/summaries, the latter at both scene and episode level. Converted inputs are processed independently in streams, along with the question and each answer, producing a score per answer. Finally, stream embeddings are fused separately per answer and a prediction is made.

tion and answers are embedded together with the input text of each stream. A *temporal attention* mechanism localizes relevant intervals from episode inputs. Finally, question answering is addressed both in a *single-stream* (Section 6) and a *multi-stream* (Section 7) scenario. The latter amounts to *multi-modal fusion*. We begin our discussion with *transformer* networks (Section 4), which we use both for dialog summarization and text embeddings in general.

4. Transformers

The *transformer* [31] is a network architecture that allows for efficient pairwise interaction between input elements. Its main component is an *attention* function, which acts as a form of associative memory.

Multi-head attention is a fusion of several attention functions. The architecture is a stack of multi-head attention, element-wise fully-connected and normalization layers with residual connections. Originally developed for machine translation, it includes an *encoder* and a *decoder* stack. The decoder additionally attends over the output of the encoder stack and is *auto-regressive*, consuming previously generated symbols when generating the next.

BERT [6] is a transformer bidirectional *encoder* only, mapping a sequence of tokens to a sequence of d-dimensional vectors. It is pre-trained on unsupervised tasks including prediction of masked tokens and next sentence, and can be also fine-tuned on supervised downstream tasks. It can take a number of *sentences* as in input, where a sentence is an arbitrary span of contiguous text.

We use BERT as the backbone of our model architecture to represent text, using two sentences at a time. Given strings A and B, the input is given as

$$\operatorname{tok}_k([\texttt{CLS}] + A + [\texttt{SEP}] + B + [\texttt{SEP}]), \qquad (1)$$

where $+$ is string concatenation and tok_k is tokenization into k tokens, with zero padding if the input length is less than k and truncation if it is greater. Tokens are represented by WordPiece embeddings [28, 35], concatenated with *position embeddings* representing their position in the input sequence and *segment embeddings*, where segments correspond to sentences and are defined according to occurrences of the *separator* token [SEP]. The output vector in $\mathbb{R}^d$ corresponding to token [CLS] is an *aggregated representation* of the entire input sequence and we denote it as

$$f(A, B). \qquad (2)$$

Sentence-BERT [26] takes a single sentence as input and is trained by *metric learning* objectives, *e.g.* in a siamese or triplet structure, facilitating efficient sentence similarity search. It is learned by fine-tuning a pre-trained BERT model on supervised semantic textual similarity.

BART [22] combines a bidirectional *encoder* and an auto-regressive *decoder*. It is pre-trained as an unsupervised denoising autoencoder, *i.e.*, corrupting input text and learning to reconstruct the original, and fine-tuned on supervised classification, generation or translation tasks. It is particularly effective on *text generation*, including abstractive dialog, question answering and summarization tasks.

Following [4], we use sentence-BERT and BART to *segment* and *summarize dialog* respectively.

5. Input description

All input sources, *i.e.*, *video*, *dialog* and *plot*, are converted into *plain text description* before being used for question answering. Video is first converted into a *scene graph* by a visual recognition pipeline and then to text description by a set of rules. Importantly, although already in textual form, dialog is also converted into text description by *dialog summarization*. The plot, already in text description form, is used as is, but for comparison only: Our main contribution is to replace human-generated plots by automatically generated descriptions.

5.1. Dialog

As the main form of human communication, dialog is an essential input source for video understanding and question answering. We use dialog in three ways: *raw dialog* per scene, *dialog summary* per scene and the collection of dialog summary over a whole *episode*.

Raw scene dialog As in all prior work, we use the raw dialog associated to the scene of the question, *as is*. Although in textual form, it is *not* a text description. It may still contain more information than dialog summary, which is important to investigate.

Scene dialog summary Given the dialog associated to the scene of the question, we convert this input source into text description by *dialog summarization*. Despite being of textual form, dialog is very different from text *description*: conversations are often informal, verbose and repetitive, with few utterances being informative; while a description is a narrative in *third-person* point of view with clear information flow structured in paragraphs [4]. Identifying the speaking person is also substantial, especially with multiple people in a conversation. Rather than generic document summarization [12], we follow a dedicated dialog summarization method [4], which blends character names with events in the generated summaries.

A dialog is a sequence of *utterances*, each including a *speaker* (character) name and a *sentence* (sequence of tokens). Each utterance is mapped to a vector embedding by Sentence-BERT [26]. The sequence of embeddings over the entire dialog is segmented according to *topic*, *e.g. greetings*, *today's plan*, *etc*. by C99 [5], as well as *stage*, *e.g. opening*, *intention*, *discussion*, *conclusion* by a *hidden Markov model* (HMM) [1]. As a result, for each *view* (topic or stage), the dialog is represented by a sequence of *blocks*, each containing several utterances.

Given the above structure, the input is re-embedded and the summary is generated using an extension of BART [22]. In particular, there is one *encoder* per view, mapping each block to an embedding. An LSTM [13] follows, aggregating the entire view into one embedding, obtained as its last hidden state. The *decoder* attends over the output of each encoder using a *multi-view attention* layer to weight the contribution of each view. It is *auto-regressive*, using previous tokens from ground truth at training and previously predicted tokens by the encoder at inference.

We train the HMM on the dialog sources of our video QA training set; otherwise, we use Sentence-BERT and BART as used/trained by [4]. Once a scene dialog summary is generated, it is re-embedded by BERT [6] like all other input sources, as discussed in Section 6.

Episode dialog summary We collect the scene dialog summaries for all scenes of an episode and we concatenate them into an *episode dialog summary*. Assuming that the episode of the scene of the question is known, we make available the associated episode dialog summary for question answering. This is a long input source and requires *temporal attention*, as discussed in Subsection 6.2. Importantly, episode dialog summary is our most important contribution in substituting plot summary by an automatically generated description.

5.2. Plot summary

As part of our comparison to [10], we use publicly available plot summaries[1], already in text description form. Assuming that the episode of the scene of the question is known, we make available the associated plot *as is*, to help answering *knowledge-based questions*. A plot is shorter and higher-level than our episode dialog summary, but it is still long enough to require *temporal attention*. It is important to investigate whether we can dispense of such a human-generated input and how much more information it contains relative to what we can extract automatically.

5.3. Video

We use a visual recognition pipeline to convert raw input video into text description. Following [10], this pipeline comprises four components: *character recognition* [27], *place recognition* [40], *object relation detection* [38], and *action recognition* [34]. The outputs of these components are character, place, object, relation and action *nodes*. A directed *video scene graph* is generated by collecting all nodes along with edges and then a textual *scene description* is obtained according to a set of predefined rules.

6. Single-stream QA

As shown in Figure 2, there is one stream per input source, using a transformer to map inputs to embeddings. Following [10], we first attempt question answering on each stream alone. In doing so, we learn a linear classifier while fine-tuning the entire transformer representation per stream. Unlike most existing works, this allows adapting to the data at hand, for instance a particular TV show.

[1] https://the-big-bang-theory.com/

We differentiate *scene* from *episode* inputs, as discussed below. In both cases, the given question and candidate answer strings are denoted as q and a^c for $c = 1, \dots, n_c$ respectively, where n_c is the number of candidate answers.

6.1. Scene input sources

Scene input sources refer to the scene of the question, *i.e.*, *raw scene dialog*, *scene dialog summary* or *video*. The input string is denoted by x. For each $c = 1, \dots, n_c$, we embed x, q and a^c jointly to d-dimensional vector

$$y^c := f(x + q, a^c), \tag{3}$$

where $+$ is string concatenation and f is BERT (2). A linear classifier with parameters $\mathbf{w} \in \mathbb{R}^d$, $b \in \mathbb{R}$ yields a score per candidate answer

$$z^c := \mathbf{w}^\top \cdot y^c + b. \tag{4}$$

The *score vector* $z := (z^1, ..., z^{n_c})$ is followed by softmax and cross-entropy loss. At training, we use f as pre-trained and we fine-tune it while optimizing W, b on the correct answers of the QA training set. At inference, we predict $\arg\max_c z^c$.

6.2. Episode input sources

Episode input sources refer to the entire episode of the scene of the question, *i.e.*, *episode dialog summary* and *plot*. Because such input is typically longer than the transformer's maximum sequence length k (1), we split it into overlapping parts in a *sliding window* fashion. Each part contains the question and one answer, so the window length is $w = k - |q| - |a^c|$. Given an input of length ℓ tokens, the number of parts is $n := \left\lceil \frac{\ell - w}{s} \right\rceil + 1$, where s is the *stride*. Because all inputs in a mini-batch must have the same number of parts n_p to be stacked in a tensor, certain parts are zero-padded if $n < n_p$ and discarded if $n > n_p$.

Embedding The input strings of the parts are denoted by p_j for $j = 1, \dots, n_p$. Each part p_j is combined with each candidate answer a^c separately, yielding the d-dimensional vectors

$$y_j^c := f(p_j + q, a^c) \tag{5}$$

for $c = 1, \dots, n_c$ and $j = 1, \dots, n_p$. A classifier with parameters $\mathbf{w} \in \mathbb{R}^d$, $b \in \mathbb{R}$ yields a score per candidate answer c and part j:

$$z_j^c := \mathbf{w}^\top \cdot y_j^c + b. \tag{6}$$

Temporal attention At this point, unlike scene inputs (4), predictions from (6) are not meaningful unless a part j is known, which amounts to *temporal localization* of the part of the input sequence that contains the information needed to answer a question. In TVQA [20] and related work [21, 14, 16], localization ground truth is available, allowing a two-stage localize-then-answer approach. Without such information, the problem is *weakly supervised*.

Previous work [10] simply chooses the part j corresponding to the maximum score z_j^c over all answers c and all parts j in (6), which is called *hard temporal attention* in the following. Such hard decision may be harmful when the chosen j is incorrect, especially when the predicted answer happens to be correct, because then the model may receive arbitrary gradient signals at training. To alleviate this, we follow a *soft temporal attention* approach.

In particular, let S be the $n_p \times n_c$ matrix with elements z_j^c over all answers c and all parts j (6). For each part j, we take the maximum score over answers

$$s_j := \max_c z_j^c, \tag{7}$$

giving rise to a vector $s := (s_1, \dots, s_{n_p})$, containing a single best score per part. Then, by soft assignment over the rows of S—corresponding to parts—we obtain a score for each answer c, represented by *score vector* $z \in \mathbb{R}^c$:

$$z := \operatorname{softmax}(s/T)^\top \cdot S, \tag{8}$$

where T is a temperature parameter. With this definition of z, we have a single score vector and we proceed as in (4).

7. Multi-stream QA

Once a separate transformer has been fine-tuned separately for each stream, we combine all streams into a single question answering classifier, which amounts to multi-modal fusion. Here, we introduce two new simple solutions.

In both cases, we freeze all transformers and obtain d-dimensional embeddings y^c for each candidate answer c and for each stream. For scene inputs, y^c is obtained directly from (3). Episode input streams produce n_p embeddings per answer. Temporal localization is thus required for part selection, similar to single stream training. Again, *hard temporal attention* amounts to choosing the part with the highest score according to (6): $y^c := y_{j^*}^c$ where $j^* := \arg\max_j(z_j^c)$ and y_j^c is given by (5). Instead, similar to (8), we follow *soft temporal attention*:

$$y^c := \operatorname{softmax}(s/T)^\top \cdot Y_c^{emb}, \tag{9}$$

where Y_c^{emb} is a $n_p \times d$ matrix collecting the embeddings y_j^c (5) of all parts j. Finally, for each answer c, the embeddings y^c of all streams are stacked into a $n_s \times d$ *embedding matrix* Y_c, where n_s is the number of streams.

Multi-stream attention The columns of Y_c are embeddings of different streams. We weight them according to weights $w_c \in \mathbb{R}^{n_s}$ obtained from Y_c itself, using a *multi-stream attention* block, consisting of two fully connected layers followed by softmax:

$$Y_c^{\text{att}} = \operatorname{diag}(w_c) \cdot Y_c. \tag{10}$$

For each answer c, a fully connected layer maps the $d \times n_s$ matrix Y_c^{att} to a scalar score. All n_c scores are followed by softmax and cross-entropy loss, whereby the parameters of all layers are jointly optimized.

Self-attention Alternatively, Y_c is mapped to $Y_c^{\text{att}} \in \mathbb{R}^{d \times n_s}$ by a single *multi-head self-attention* block, as in transformers [31]:

$$Y_c^{\text{att}} = \text{MultiHeadAttention}(Y_c, Y_c, Y_c). \quad (11)$$

The remaining pipeline is the same as in the previous case.

8. Experiments

8.1. Experimental setup

Datasets The KnowIT VQA [11] dataset contains 24,282 human-generated questions associated to 12,087 scenes, each of duration 20 seconds, from 207 episodes of *The Big Bang Theory* TV show. Questions are of four types: *visual* (22%), *textual* (12%), *temporal* (4%) and *knowledge* (62%). Question types are only known for the test set. Knowledge questions require reasoning based on knowledge from the episode or the entire TV show, which differs from other video question answering datasets. Questions are multiple-choice with $n_c = 4$ answers per question and performance is measured by *accuracy*, per question type and overall.

Implementation details For scene dialog summary generation, we set the minimum sequence length to 30 tokens and the maximum to 100 in the BART [22] model. With this setting, episode dialog summaries are 2078 tokens long on average, while plot summaries are 659 tokens long.

We fine-tune the $\text{BERT}_{\text{BASE}}$ [6] uncased model with $N = 12$ transformer blocks, $h = 12$ self-attention heads and embedding dimension $d = 768$ for single-stream models. The maximum token length k is 512 for scene, 200 for plot and 300 for episode dialog summary inputs. The stride s is 100 for plot and 200 for episode dialog summary. The maximum number of parts n_p is 10 for both. The batch size is 8 for all single-stream models and 32 for multi-stream. We use SGD with momentum 0.9 scheduled with initial learning rate 10^{-4} for multi-stream fusions. We use $h = 1$ attention head, and $N = 2$ stacks for self-attention and multi-stream self-attention methods. The number of streams n_s varies per experiment.

8.2. Quantitative results

Table 1 compares of our method with the state of the art. Rookies and Masters are human evaluators: Masters have watched most of the show, whereas Rookies have never watched an episode before [11]. TVQA [20] encodes visual features and subtitles without considering knowledge information; its results are as reported in [11]. ROCK [11] uses four visual representations (image, concepts, facial, caption); $\text{ROCK}_{\text{facial}}$ is one of its best results. ROCK_{GT} [11] and $\text{ROLL}_{\text{human}}$ [10] use the human knowledge annotation provided by the dataset [11], while ROLL [10] uses human-written plot summaries instead. Our method uses scene video and scene dialog summary as well as the episode dialog summary that it automatically generates, without any human annotation. $\text{Ours}_{\text{plot}}$ additionally uses the same plot as [10]. TVQA uses LSTM; all other methods are based on BERT.

Our method outperforms the best state of the art method (ROLL [10]) by 6.6%, without any human annotation. By using additional human-generated plots, the gain decreases to 5.8%. This indicates that our episode dialog summary captures the required knowledge and removes the requirement of human-generated input; in fact, human-generated input is harmful. On temporal and knowledge questions in particular, we gain 13.9% and 7.6%, respectively, without any human annotation. This implies that our automatically generated episode dialog summary increases the understanding of the episode and helps answering all types of questions. Despite $\text{ROLL}_{\text{human}}$ [10] and ROCK_{GT} [11] using ground-truth knowledge, we outperform them by 16.1% and 5.0%, respectively, without any human annotation. We also outperform Rookies, presumably by having access to the dialog of the entire episode. Comparing to Masters, there is still room for improvement.

8.3. Qualitative analysis

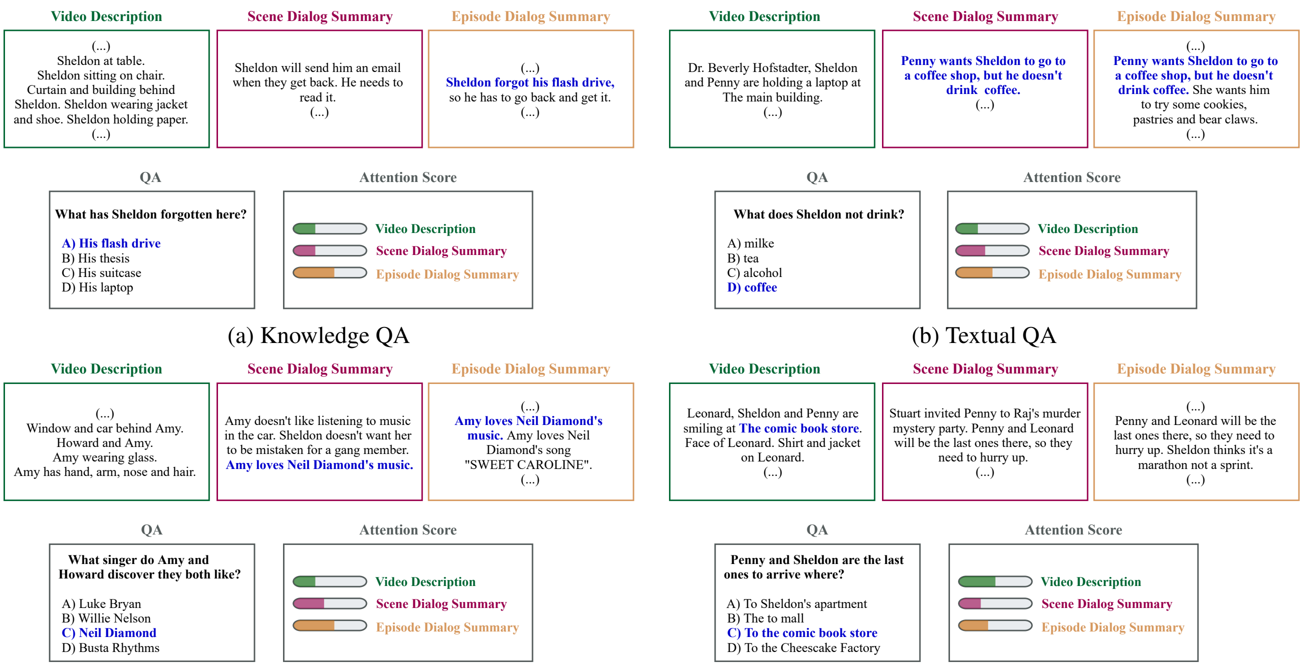
Figure 3 visualizes the correct predictions of our method with stream attention scores for different question types. In all examples, the model receives three input sources, question/answers and attention scores over inputs. Figure 3(a) shows a *knowledge* question, answered based on episode dialog summary, which has the highest attention score. As shown in Figure 3(b), a *textual* question can be answered by using scene dialog summary, but also by episode dialog summary, since the latter includes the former. *Temporal* questions can be answered from scene inputs such as scene dialog summary or video description. According to attention scores, the question in Figure 3(c) is answered by episode dialog summary, which includes the correct answer. Finally, Figure 3(d) shows a *visual* question answered by video description.

8.4. Ablation studies

Single-stream results Table 2 shows our single-stream QA results. We reproduce [10] for dialog, video, and plot inputs. We replace the plot stream by one using our new temporal attention (Subsection 6.2) and other improvements (Table 4) and we add two new sources automatically generated from dialog: scene dialog summary and episode dialog summary. Due to the dataset having a majority of knowl-

Method	Knowledge	Vis.	Text.	Temp.	Know.	All
Rookies [11]	–	0.936	0.932	0.624	0.655	0.748
Masters [11]	✓	0.961	0.936	0.857	0.867	0.896
ROCK_{GT} [11]	question GT	0.747	**0.819**	0.756	0.708	0.731
$\text{ROLL}_{\text{human}}$ [10]	question GT	0.708	0.754	0.570	0.567	0.620
TVQA [20]	–	0.612	0.645	0.547	0.466	0.522
$\text{ROCK}_{\text{facial}}$ [11]	dataset GT	0.654	0.688	0.628	0.646	0.652
ROLL [10]	plot	0.718	0.739	0.640	0.713	0.715
Ours	–	**0.755**	0.783	**0.779**	**0.789**	**0.781**
$\text{Ours}_{\text{plot}}$	plot	0.749	0.783	0.721	0.783	0.773

Table 1: *State-of-the-art accuracy* on KnowIT VQA. Ours uses the video and scene dialog summary as well as the episode dialog summary that we generate from the dialog of the entire episode. $\text{Ours}_{\text{plot}}$ also uses human-generated plot summaries, like [10]. TVQA uses an LSTM based encoder; all other methods use BERT. Rookies and Masters are humans.



(c) Temporal QA

(d) Visual QA

Figure 3: *Multi-stream attention visualization.* We highlight in blue the part of the source text that is relevant to answering the question. The most attended stream is episode dialog summary for (a), (b), (c) and video description for (d).

edge questions, episode dialog summary and plot inputs have higher accuracy than other input sources since they span an entire episode. Our episode dialog summary helps in answering questions better than the plot [10], bringing an accuracy improvement of 5.4%.

Multi-stream results We evaluate our two multi-stream QA methods introduced in Section 7, namely *multi-stream attention* and *self-attention*, comparing them with the following combinations/baselines/competitors:

1. *Multi-stream self-attention*: combination of multi-stream attention and self-attention: the output of the latter is weighted by the former. The remaining pipeline is the same as in multi-stream attention.
2. *Product*: Hadamard product on embeddings of all streams per answer, followed by a linear classifier per answer. The remaining pipeline is the same.
3. *Modality weighting* [10]: a linear classifier (4) and loss function is used as in single-stream QA but with transformers frozen for each stream separately. The ob-

Method	Input	Vis.	Text.	Temp.	Know.	All
ROLL [10]	D	0.656	0.772	0.570	0.525	0.584
	V	0.629	0.424	0.558	0.514	0.530
	P	0.624	0.620	0.570	0.725	0.685
ROLL [10]†	D	0.649	**0.801**	0.581	0.543	0.598
	V	0.625	0.431	0.512	0.541	0.546
	P	0.647	0.554	0.674	0.694	0.667
Ours	P	0.666	0.623	0.593	0.735	0.702
	S	0.631	0.746	0.605	0.537	0.585
	E	**0.676**	0.750	**0.779**	**0.785**	**0.756**

Table 2: *Single-stream QA accuracy* on KnowIT VQA. ROLL [10]: as reported; [10]†: our reproduction. Our model incorporates the scene dialog and video streams of the latter as well as the plot, scene dialog summary and episode dialog summary streams. Plot differs between [10]† and our model by our temporal attention and other improvements (Table 4). D: dialog; V: video; P: plot; S: scene dialog summary; E: episode dialog summary.

Method	Vis.	Text.	Temp.	Know.	All
Product	0.743	0.659	0.756	0.751	0.739
Modality weighting [10]	0.708	**0.786**	0.767	0.787	0.769
Self-attention	**0.759**	0.764	0.767	0.777	0.771
Multi-stream attention	0.755	0.783	**0.779**	**0.789**	**0.781**
Multi-stream self-attn.	0.755	0.768	0.756	0.777	0.770

Table 3: *Multi-stream QA accuracy* on KnowIT VQA, fusing video, scene dialog summary and episode dialog summary input sources. All fusion methods use soft temporal attention for localization of episode input sources. Top: baseline/competitors. Bottom: ours.

tained scores by single-stream classifiers are combined by a multi-stream classifier and another loss function applies. The overall loss all is a linear combination with weight β_ω on the multi-stream loss and $1 - \beta_\omega$ uniformly distributed over single-stream losses.

Table 3 shows results for fusion of video, scene dialog summary and episode dialog summary. For modality weighting, we set $\beta_\omega = 0.7$ according to the validation set. Our multi-stream attention outperforms other fusion methods. Besides, it does not require tuning of modality weight hyperparameter β_ω or selecting the number of heads and blocks for self-attention. Unless specified, we use multi-stream attention for fusion by default.

Improvements over [10] We reproduce ROLL [10] using official code by the authors and default parameters. This is our baseline, shown in the first row of Table 4. Then, we evaluate our improvements, adding them one at a time.

Method	Vis.	Text.	Temp.	Know.	All
ROLL [10]†	0.722	0.703	0.709	0.697	0.704
+ Multi-stream attention	0.724	0.721	0.721	0.691	0.703
+ More parts for plot	0.722	0.703	0.651	0.717	0.714
+ New order of plot inputs	0.730	0.710	0.686	0.712	0.715
+ Temporal attention	0.734	0.725	0.663	0.724	0.724
± Replacing P → E	0.753	**0.815**	**0.814**	**0.773**	0.775
± Replacing D → S	**0.755**	0.783	0.779	0.789	**0.781**

Table 4: *Accuracy improvements over* ROLL [10]. †: our reproduction. Each row adds a new improvement except the last two, where we replace streams. P: plot; E: episode dialog summary; D: dialog; S: scene dialog summary.

First, we replace modality weighting with *multi-stream attention*. Despite its simplicity, its performance is on par, losing only 0.1%, while requiring no hyperparameter tuning. Then, we increase the *number of parts* of plot summaries from 5 to 10, eliminating information loss by truncation and bringing an accuracy improvement of 1.1%. We change the *order of arguments* of BERT for episode input sources from $f(q, a^c+p_j)$ to $f(p_j+q, a^c)$ (5), which is consistent with (3) and improves only slightly by 0.1%. Our new *temporal attention* mechanism improves accuracy by 0.9%. Replacing plot with episode dialog summary, which is our main contribution, brings an improvement of 5.1%. Finally, the accuracy is improved by 0.6% by using *scene dialog summary* instead of raw dialog. The overall gain over [10] is 7.7%.

Note that the relative improvement of each new idea depends on the order chosen in Table 4. For instance, the order of BERT arguments brings improvements of up to 2.3% in experiments including the episode dialog summary.

9. Conclusion

KnowIT VQA is a challenging dataset where it was previously believed that some form of external knowledge was needed to handle knowledge questions, as if knowledge was yet another modality. Our results indicate that much of this required knowledge was hiding in *dialog*, waiting to be harnessed. It is also interesting that our *soft temporal attention* helps a lot more with our episode dialog summary than human plot summary, which may be due to the episode dialog summary being longer. This may also explain the astounding performance of episode dialog summary, despite its low overall quality: plot summaries are of much higher quality but may be missing a lot of information.

Acknowledgements This work was supported by the European Commission under European Horizon 2020 Programme, grant number 951911 - AI4Media. This work was granted access to the HPC resources of IDRIS under the allocation 2020-AD011012263 made by GENCI.

References

[1] Tim Althoff, Kevin Clark, and Jure Leskovec. Large-scale analysis of counseling conversations: An application of natural language processing to mental health. *Trans. ACL*, 4:463–476, 2016. 4

[2] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. VQA: Visual question answering. In *Proc. ICCV*, 2015. 1

[3] Aman Chadha, Gurneet Arora, and Navpreet Kaloty. iPerceive: Applying common-sense reasoning to multi-modal dense video captioning and video question answering. In *Proc. WACV*, 2021. 1, 2

[4] Jiaao Chen and Diyi Yang. Multi-view sequence-to-sequence models with conversational structure for abstractive dialogue summarization. In *Proc. EMNLP*, 2020. 2, 3, 4

[5] Freddy Y. Y. Choi. Advances in domain independent linear text segmentation. In *Proc. NAACL*, 2000. 4

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. NAACL*, 2019. 3, 4, 6

[7] Chenyou Fan, Xiaofan Zhang, Shu Zhang, Wensheng Wang, Chi Zhang, and Heng Huang. Heterogeneous memory enhanced multimodal attention model for video question answering. In *Proc. CVPR*, 2019. 1

[8] Xiachong Feng, Xiaocheng Feng, Bing Qin, and Ting Liu. Incorporating commonsense knowledge into abstractive dialogue summarization via heterogeneous graph networks. *arXiv preprint arXiv:2010.10044*, 2020. 2

[9] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. DeViSE: A deep visual-semantic embedding model. In *Proc. NIPS*, 2013. 1

[10] Noa Garcia and Yuta Nakashima. Knowledge-based video question answering with unsupervised scene descriptions. In *Proc. ECCV*, 2020. 2, 4, 5, 6, 7, 8, 12, 13

[11] Noa Garcia, Mayu Otani, Chenhui Chu, and Yuta Nakashima. KnowIT VQA: Answering knowledge-based questions about videos. In *Proc. AAAI*, 2020. 2, 6, 7

[12] Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*. ACL, 2019. 2, 4

[13] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. 4

[14] Hyounghun Kim, Zineng Tang, and Mohit Bansal. Dense-caption matching and frame-selection gating for temporal localization in VideoQA. In *Proc. ACL*, 2020. 1, 2, 5

[15] Junyeong Kim, Minuk Ma, Kyungsu Kim, Sungjin Kim, and Chang D Yoo. Progressive attention memory network for movie story question answering. In *Proc. CVPR*, 2019. 2

[16] Junyeong Kim, Minuk Ma, Trung Pham, Kyungsu Kim, and Chang D. Yoo. Modality shifting attention network for multi-modal video question answering. In *Proc. CVPR*, 2020. 1, 2, 5

[17] Kyung-Min Kim, Seong-Ho Choi, Jin-Hwa Kim, and Byoung-Tak Zhang. Multimodal dual attention memory for video story question answering. In *Proc. ECCV*, 2018. 1, 2

[18] Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*, 2014. 1

[19] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *Proc. ICCV*, 2017. 1

[20] Jie Lei, Licheng Yu, Mohit Bansal, and Tamara Berg. TVQA: Localized, compositional video question answering. In *Proc. EMNLP*, 2018. 1, 2, 5, 6, 7

[21] Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. TVQA+: Spatio-temporal grounding for video question answering. In *Proc. ACL*, 2019. 2, 5

[22] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proc. ACL*, 2020. 3, 4, 6

[23] Junwei Liang, Lu Jiang, Liangliang Cao, Li-Jia Li, and Alexander G Hauptmann. Focal visual-text attention for visual question answering. In *Proc. CVPR*, 2018. 2

[24] Seil Na, Sangho Lee, Jisung Kim, and Gunhee Kim. A read-write memory network for movie story understanding. In *Proc. ICCV*, 2017. 1, 2

[25] Haojie Pan, Junpei Zhou, Zhou Zhao, Yan Liu, Deng Cai, and Min Yang. Dial2desc: end-to-end dialogue description generation. *arXiv preprint arXiv:1811.00185*, 2018. 2

[26] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proc. EMNLP-IJCNLP*, 2019. 3, 4

[27] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. In *Proc. CVPR*, 2015. 4

[28] Mike Schuster and Kaisuke Nakajima. Japanese and korean voice search. In *Proc. ICASSP*, 2012. 3

[29] Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. MovieQA: Understanding stories in movies through question-answering. In *Proc. CVPR*, 2016. 1, 2

[30] Aisha Urooj, Amir Mazaheri, Niels Da Vitoria Lobo, and Mubarak Shah. MMFT-BERT: Multimodal fusion transformer with bert encodings for visual question answering. In *Proc. EMNLP*, 2020. 1, 2

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proc. NIPS*, 2017. 3, 6

[32] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence-video to text. In *Proc. ICCV*, 2015. 1

[33] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proc. CVPR*, 2015. 1

[34] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick. Long-term feature banks for detailed video understanding. In *Proc. CVPR*, 2019. 4

[35] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016. 3

[36] Zekun Yang, Noa Garcia, Chenhui Chu, Mayu Otani, Yuta Nakashima, and Haruo Takemura. BERT representations for video question answering. In *Proc. WACV*, 2020. 1, 2

[37] Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. From recognition to cognition: Visual commonsense reasoning. In *Proc. CVPR*, 2019. 1

[38] Ji Zhang, Yannis Kalantidis, Marcus Rohrbach, Manohar Paluri, Ahmed Elgammal, and Mohamed Elhoseiny. Large-scale visual relationship understanding. In *Proc. AAAI*, 2019. 4

[39] Lulu Zhao, Weiran Xu, and Jun Guo. Improving abstractive dialogue summarization with graph structures and topic words. In *Proc. COLING*, 2020. 2

[40] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Trans. PAMI*, 40(6):1452–1464, 2017. 4

Appendix

A. Additional qualitative analysis

Dialog summarization In the example of Figure 4, Howard says, "I invited *her*." in scene B. Our dialog summarization interprets this sentence by assigning the correct character name: "Howard invited *Bernadette* in." Hence, we can answer the question of scene A, "Who did Howard invite to join him and Raj in Raj's lab?" correctly. Thanks to the episode dialog summary spanning all scenes and the use of character names instead of pronouns, our method can answer character-related questions correctly.

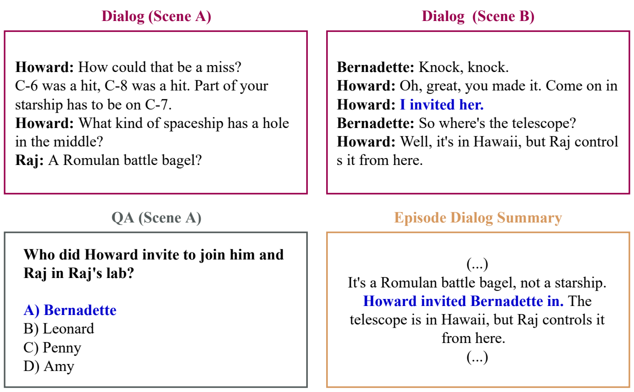


Figure 4: Dialog summarization converts pronouns in dialog to character names in episode dialog summary, supporting question answering. In particular, "I" is substituted by "Howard" and "her" by "Bernadette".

Plot *vs*. episode dialog summary A comparison of plot summary and episode dialog summary is given in Figure 5. There are three different topics in the story line, and each is highlighted with the same color in both summaries. The first topic, highlighted in purple, is "Sheldon's forgotten flash drive." The second, highlighted in yellow, is "Sheldon's grandmother." The third, highlighted in blue, is "Asking Summer out." The plot summary is topic-centered, while the episode dialog summary is following the narrative order. Hence, topics may be fragmented in the latter. The episode dialog summary has more detail than the plot. In particular, it contains enough information to answer the question *Why does Sheldon's grandmother call him Moon Pie?* That is, *because he's nummy-nummy*. This information is missing from the plot summary, which focuses on the main topics/events of an episode. Even though the episode dialog summary is noisy, it contains details that help in question answering.

Failure cases Figure 6 shows examples of failed predictions of our model along with stream attention scores for different question types. The model receives three input

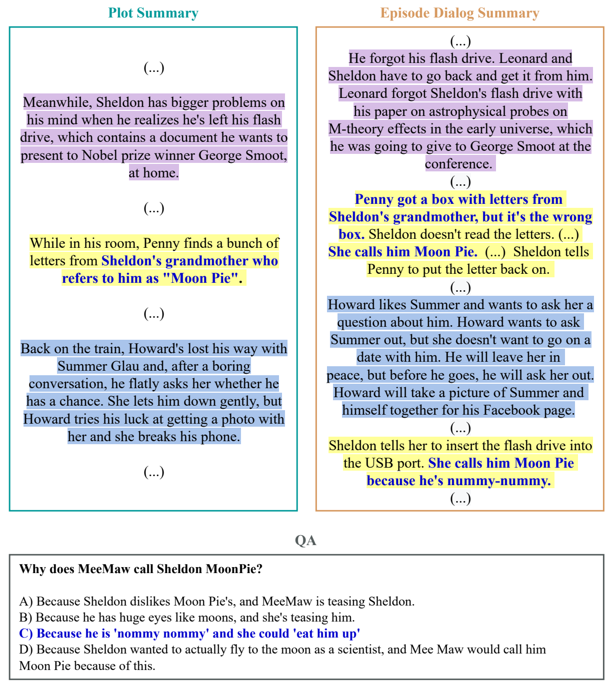


Figure 5: An example of plot summary and episode dialog summary, with each topic highlighted in the same color in both summaries. Phrases relevant to QA in blue. Only the episode dialog summary contains enough information to answer the question.

sources, question/answers and attention scores over inputs.

Figure 6(a) refers to a knowledge question, which requires recurrent knowledge of the whole TV show. In other words, the correct answer cannot be found in episode dialog summary. The question is answered as "a lasagna" found in episode dialog summary, even though it is wrong.

Figure 6(b) refers to a textual question, which should have been answered by scene dialog summary. However, scene dialog summary does not contain the correct answer. Our model gives most attention to episode dialog summary. The prediction is made according to the highlighted text, which is the same in both sources. However, this prediction refers to the wrong person.

Figure 6(c) refers to another knowledge question, which could be answered by the highlighted text in episode dialog summary. Even though episode dialog summary has the most attention, the prediction is incorrect.

Figure 6(d) refers to another textual question, which should have been answered by scene dialog summary. Although both scene dialog summary and episode dialog summary include the correct answer, and episode dialog summary has the most attention, the prediction indicates the wrong person.

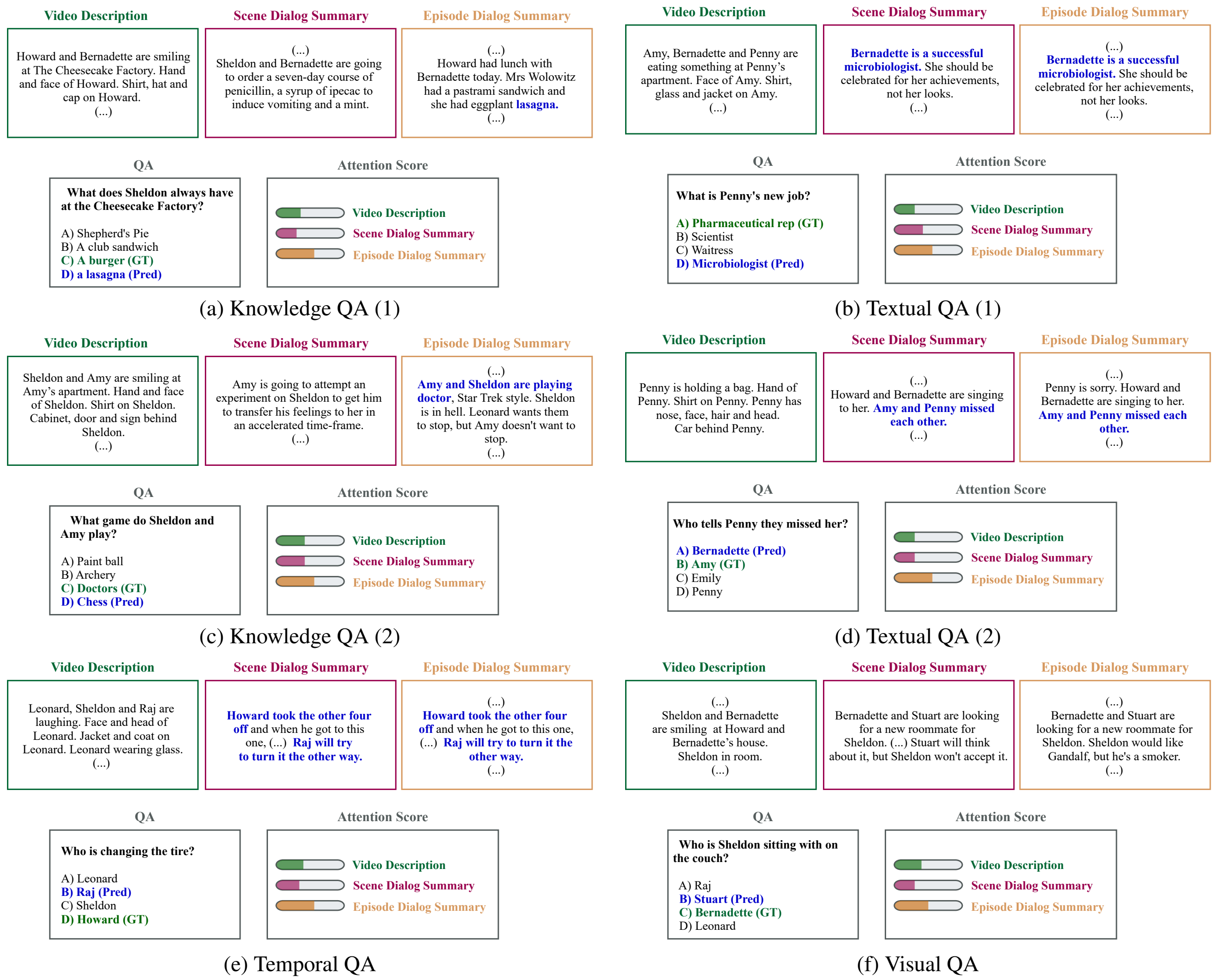


Figure 6: *Failed predictions of multi-stream attention*. We highlight in blue the part of the source text that might be relevant to answering the question. "Pred"/blue: model predictions. "GT"/green: ground truth.

Figure 6(e) refers to a temporal question. The scene dialog summary and episode dialog summary imply that *Raj* and *Howard* might be changing the tire. The video description is not helpful either. Hence, our model predicts *Raj*, while the correct answer is *Howard*.

Figure 6(f) is a visual question. However, the video description fails to convey relevant information to answer the question. The other inputs do not contain relevant information either. One of the character names appearing in episode dialog summary is predicted, which is incorrect.

B. Additional ablation studies

Hyperparameter validation *Modality weighting* [10] fusion method requires selection of hyperparameter β_ω. Figure 7 shows validation accuracy *vs.* β_ω for fusion of video, scene dialog summary and episode dialog summary. We choose $\beta_\omega = 0.7$ for both soft and hard temporal selection to report results in Table 3 and Table 6. The remaining weight of $1-\beta_\omega$ is evenly distributed over individual stream losses as 0.1 per stream.

Effect of temporal attention on single-stream QA We investigate the effect of our soft temporal attention (Subsection 6.2) on single-stream QA for episode input sources. We also evaluate the effect of single-stream training with soft or hard temporal attention on multi-stream attention, where we use soft temporal attention. According to Table 5, temporal attention improves the accuracy of plot and episode dialog summary by 1.9% and 3.3%, respectively. Accordingly, the accuracy of multi-stream QA on the same episode sources as well as video and scene dialog sum-

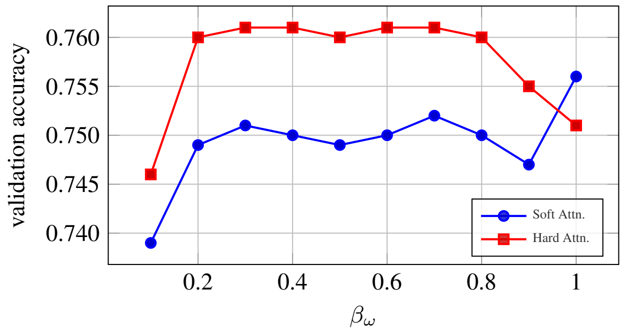


Figure 7: Accuracy *vs.* β_ω for fusion of video, scene dialog summary and episode dialog summary by modality weighting [10] on KnowIT VQA validation set.

Stream Inputs		Soft Attn.	Vis.	Text.	Temp.	Know.	All
Single	P	-	0.656	0.594	0.628	0.712	0.683
	P	✓	0.666	0.623	0.593	0.735	0.702
	E	-	0.604	0.721	0.733	0.765	0.723
	E	✓	**0.676**	**0.750**	**0.779**	**0.785**	**0.756**
Multi	V + S + P	-	0.732	0.688	0.674	0.720	0.717
	V + S + P	✓	0.739	0.699	0.628	0.728	0.723
	V + S + E	-	0.707	0.772	0.721	0.700	0.711
	V + S + E	✓	**0.755**	**0.783**	**0.779**	**0.789**	**0.781**

Table 5: *Effect of temporal attention on single-stream QA* on KnowIT VQA. Soft Attn.: soft temporal attention on single-stream training. We use soft temporal attention for multi-stream QA, but this is still affected by the temporal attention used in single-stream training. V: video; S: scene dialog summary; P: plot; E: episode dialog summary.

mary increases by 0.6% and 7.0%, respectively. The gain is higher when episode dialog summary is used, since the episode dialog summary is longer than plot.

Effect of temporal attention on multi-stream QA Table 6 shows the effect of *soft temporal attention* on multi-stream QA for fusion of video, scene dialog summary and episode dialog summary input sources. We use soft temporal attention for single-stream QA of episode dialog summary. In all fusion methods, the overall accuracy is improved by using soft temporal attention.

Different input combinations Table 7 shows the accuracy of multi-stream QA for different input combinations, where the number of input streams varies in $\{2, 3, 4, 5\}$. Scene dialog summaries improves the accuracy compared with single-stream QA results in Table 2. Moreover, using the episode dialog summary always improves the overall accuracy by a large margin. The best overall accuracy

Method	Soft Attn.	Vis.	Text.	Temp.	Know.	All
Product	-	0.728	0.645	0.744	0.756	0.736
	✓	0.743	0.659	0.756	0.751	0.739
Modality weighting [10]	-	0.716	**0.815**	0.791	0.776	0.768
	✓	0.708	0.786	0.767	0.787	0.769
Self-attention	-	0.753	0.804	**0.802**	0.766	0.769
	✓	**0.759**	0.764	0.767	0.777	0.771
Multi-steam attention	-	0.743	0.790	0.779	0.785	0.776
	✓	0.755	0.783	0.779	**0.789**	**0.781**
Multi-steam self attn.	-	0.749	0.797	0.791	0.768	0.768
	✓	0.755	0.768	0.756	0.777	0.770

Table 6: *Effect of temporal attention on multi-stream QA* on KnowIT VQA for fusion of video, scene dialog summary, and episode dialog summary input sources. Soft Attn.: soft temporal attention on multi-stream training. We use soft temporal attention for single-stream QA of episode dialog summary.

Analysed Inputs	Inputs	Vis.	Text.	Temp.	Know.	All
D V P	D+V	0.693	0.768	0.593	0.554	0.611
	D+P	0.732	0.721	0.674	0.723	0.723
	D+V+P	0.734	0.725	0.663	0.724	0.724
D V P S	D+S	0.664	0.786	0.628	0.548	0.604
	V+S	0.689	0.721	0.581	0.549	0.601
	P+S	0.716	0.710	0.628	0.727	0.719
	D+V+P+S	0.734	0.732	0.663	0.725	0.726
D V P S E	D+E	0.743	0.812	0.779	0.779	0.775
	V+E	0.732	0.761	0.767	0.788	0.772
	P+E	0.716	0.743	0.721	**0.791**	0.766
	D+S+E	0.743	**0.822**	**0.802**	0.771	0.772
	V+S+E	**0.755**	0.783	0.779	0.789	**0.781**
	P+S+E	0.739	0.779	0.733	0.783	0.771
	D+V+P+S+E	0.751	0.797	0.744	0.781	0.775

Table 7: *Multi-stream QA accuracy* on KnowIT VQA: comparison of different input combinations for multi-stream attention. D: dialog; V: video; P: plot; S: scene dialog summary; E: episode dialog summary.

of 0.781 is achieved by video, scene dialog summary, and episode dialog summary.

Question type $\leftrightarrow$ attention scores We perform significance testing for the dependence between the question type and attention scores. There are 2 independent variables in the scores of 3 streams, whose values we discretize into

10×10 bins. We form a $4 \times 10 \times 10$ joint histogram of question type (X) and scores (Y) and compute the mutual information $I(X;Y)$. We perform a G-test[2] with $G = 2N \cdot I(X;Y)$, where $N = 2361$ is the number of test questions. Finally, using a chi-square distribution of $3 \times 9 \times 9$ DoF, we find a p-value of 1.52×10^{-25} for the null hypothesis. This indicates that attention scores depend on question type.

Replacing attention scores with oracle scores determined by question type Assuming that we know the question type for the test set, we perform an *oracle* experiment where attention scores are based on question type rather than our fusion method. We only consider visual, textual, and knowledge types of question. In particular, we assign visual questions to video input, textual questions to scene dialog summary and knowledge questions to episode dialog summary. We exclude temporal questions since they can be answerable by scene dialog summary or video. Only 3.6% of questions are of temporal type in the test set. We find that our multi-stream attention method (0.781%) is 3.6% better than the oracle experiment (0.745%). This indicates that our fusion mechanism is more effective than a naïve oracle that assumes more knowledge.

[2]https://en.wikipedia.org/wiki/G-test