KnowIT VQA: Answering Knowledge-Based Questions about Videos

Noa Garcia,¹ Mayu Otani,² Chenhui Chu,¹ Yuta Nakashima¹
¹Osaka University, Japan
{noagarcia, chu, n-yuta}@ids.osaka-u.ac.jp
²CyberAgent, Inc., Japan
otani_mayu@cyberagent.co.jp

Abstract

We propose a novel video understanding task by fusing knowledge-based and video question answering. First, we introduce KnowIT VQA, a video dataset with 24,282 human-generated question-answer pairs about a popular sitcom. The dataset combines visual, textual and temporal coherence reasoning together with knowledge-based questions, which need of the experience obtained from the viewing of the series to be answered. Second, we propose a video understanding model by combining the visual and textual video content with specific knowledge about the show. Our main findings are: (i) the incorporation of knowledge produces outstanding improvements for VQA in video, and (ii) the performance on KnowIT VQA still lags well behind human accuracy, indicating its usefulness for studying current video modelling limitations.

Introduction

Visual question answering (VQA) was firstly introduced in (Malinowski and Fritz 2014) as a task for bringing together advancements in natural language processing and image understanding. Since then, VQA has experienced a huge development, in part due to the release of a large number of datasets, such as (Malinowski and Fritz 2014; Antol et al. 2015; Krishna et al. 2017; Johnson et al. 2017; Goyal et al. 2017). The current trend for addressing VQA (Anderson et al. 2018; Kim et al. 2017a; Ben-Younes et al. 2017; Bai et al. 2018) is based on predicting the correct answer from a multi-modal representation, obtained from encoding images with a pre-trained convolutional neural network (CNN) and attention mechanisms (Xu et al. 2015), and encoding questions with a recurrent neural network (RNN). These kinds of models infer answers by focusing on the content of the images (e.g. How many people are there wearing glasses? in Fig. 1).

Considering that the space spanned by the training question-image pairs is finite, the use of image content as the only source of information to predict answers presents two important limitations. On one hand, image features only capture the static information of the picture, leaving temporal coherence in video unattended (e.g. How do they finish the conversation? in Fig. 1), which is a strong constraint in real-world applications. On the other hand, visual content by itself does not provide enough insights for answering questions that require knowledge (e.g. Who owns the place were they are standing? in Fig. 1). To address these limitations, video question answering (VideoQA) (Tapaswi et al. 2016; Kim et al. 2017b; Lei et al. 2018) and knowledge-based visual question answering (KBVQA) (Wu et al. 2016; Wang et al. 2018) have emerged independently by proposing specific datasets and models. However, a common framework for addressing multi-question types in VQA is still missing.

The contribution of this work lies in this line, by introducing a general framework in which both video understanding and knowledge-based reasoning are required to answer questions. We first argue that a popular sitcom, such as The Big Bang Theory¹, is an ideal testbed for modelling knowledge-based questions about the world. With this idea, we created KnowIT VQA, a dataset for KBVQA in videos in which real-world natural language questions are designed

¹https://www.cbs.com/shows/big_bang_theory/
to be answerable only by people who is familiar with the show. We then cast the problem as a multi-choice challenge, and introduce a two-piece model that (i) acquires, processes, and maps specific knowledge into a continuous representation inferring the motivation behind each question, and (ii) fuses video and language content together with the acquired knowledge in a multi-modal fashion to predict the answer.

Related Work

**Video Question Answering** VideoQA addresses specific challenges with respect to the interpretation of temporal information in videos, including action recognition (Maharaj et al. 2017; Jang et al. 2017; Zellers et al. 2019; Mun et al. 2017), story understanding (Tapaswi et al. 2016; Kim et al. 2017b), or temporal coherence (Zhu et al. 2017). Depending on the video source, the visual content of videos may also be associated with textual data, such as subtitles or scripts, which provide an extra level of context for its interpretation. Most of the proposed datasets so far are mainly focused on either the textual or the visual aspect of the video, without exploiting the combination of both modalities. In MovieQA (Tapaswi et al. 2016), for example, questions are mainly plot-focused, whereas in other collections, questions are purely about the visual content, such as action recognition in MovieFIB (Maharaj et al. 2017), TGIF-QA (Jang et al. 2017), and MarioVQA (Mun et al. 2017), or temporal coherence in Video Context QA (Zhu et al. 2017). Only few datasets, such as PororoQA (Kim et al. 2017b) or TVQA (Lei et al. 2018), present benchmarks for exploiting multiple sources of information, requiring models to jointly interpret multi-modal video representations. Even so, reasoning beyond the video content in these kinds of approaches is complicated, as only the knowledge acquired in the training samples is used to generate the answer.

**Knowledge-Based Visual Question Answering** Answering questions about a visual query by only using its content constrains the output to be inferred within the space of knowledge contained in the training set. Considering that the amount of training data in any dataset is finite, the knowledge used to predict answers in standard visual question answering is rather limited. In order to answer questions beyond the image content, KBVQA proposes to inform VQA models with external knowledge. The way of acquiring and incorporating this knowledge, however, is still in early stages. For example, (Zhu et al. 2015) creates a specific knowledge base with image-focused data for answering questions under a certain template, whereas more generic approaches (Wu et al. 2016) extract information from external knowledge bases, such as DBpedia (Auer et al. 2007), for improving VQA accuracy. As VQA datasets do not envisage questions with general information about the world, specific KBVQA datasets have been recently introduced, including KB-VQA (Yang et al. 2017) with question-images pairs generated from templates, R-VQA (Lu et al. 2018) with relational facts supporting each question, FVQA (Wang et al. 2018) with supporting facts extracted from generic knowledge bases, KVQA (Shah et al. 2019) for entity identification, or OK-VQA (Marino et al. 2019) with free-form questions without knowledge annotations. Most of these datasets impose hard constraints on their questions, such as being generated by templates (KB-VQA) or directly obtained from existing knowledge bases (FVQA), being OK-VQA the only one that requires handling unstructured knowledge to answer natural questions about images. Following this direction, we present a framework for answering general questions that may or may not be associated with a knowledge base by introducing a new VideoQA dataset, in which questions are freely proposed by qualified workers to study knowledge and temporal coherence together. To the best of our knowledge, this is the first work that explores external knowledge questions in a collection of videos.

**KnowIT VQA Dataset**

Due to the natural structure of TV shows, in which characters, scenes, and general development of the story can be known in advance, TV data has been exploited for modelling real-world scenarios in video understanding tasks (Namgrani and Zisserman 2017; Freirman, Cohen, and Lapata 2018). We also rely on this idea and argue that popular sitcoms provide an ideal testbed to encourage progress in knowledge-based visual question answering, due to their additional facilities to model knowledge and temporal coherence over time. In particular, we introduce the KnowIT VQA dataset, (standing for knowledge informed temporal VQA), a collection of videos from The Big Bang Theory annotated with knowledge-based questions and answers about the show.

| Dataset         | VQA-Type | Domain          | # Imgs | # QAs | Answers | Vis. | Text. | Temp. | Know. |
|-----------------|----------|-----------------|--------|-------|---------|------|------|-------|-------|
| MovieQA         | Video    | Movie           | 6,771  | 14,944| MC₅     | ✓    | ✓    | ✓     | -     |
| KB-VQA          | KB       | COCO            | 700    | 2,402 | Word    | ✓    | -    | -     | ✓     |
| PororoQA        | Video    | Cartoon         | 16,066 | 8,913 | MC₅     | ✓    | ✓    | ✓     | -     |
| TVQA            | Video    | TV show         | 21,793 | 152,545| MC₅     | ✓    | ✓    | ✓     | -     |
| R-VQA           | KB       | Visual Genome   | 60,473 | 198,889| Word    | ✓    | -    | -     | ✓     |
| FVQA            | KB       | COCO, ImgNet    | 2,190  | 5,826 | Word    | ✓    | -    | -     | ✓     |
| KVQA            | KB       | Wikipedia       | 24,602 | 183,007| Word    | ✓    | -    | -     | ✓     |
| OK-VQA          | KB       | COCO            | 14,031 | 14,055| Word    | ✓    | -    | -     | ✓     |
| KnowIT VQA (Ours)| VideoKB | TV show         | 12,087 | 24,282| MC₄     | ✓    | ✓    | ✓     | -     |
Table 2: KnowIT VQA data splits and the average lengths.

|                  | Train | Val  | Test | Total |
|------------------|-------|------|------|-------|
| # Episodes       | 167   | 20   | 20   | 207   |
| # Scenes         | 2,007 | 225  | 240  | 2,472 |
| # Clips          | 9,731 | 1,178| 1,178| 12,087|
| # QAs            | 19,569| 2,352| 2,361| 24,282|
| Len. Subtitles   | 56.49 | 55.57| 57.45| 56.49 |
| Len. Questions   | 7.5   | 7.38 | 7.48 | 7.49  |
| Len. CA          | 4.55  | 4.51 | 4.46 | 4.54  |
| Len. WA          | 4.14  | 4.12 | 4.06 | 4.13  |
| Len. KNOWLEDGE  | 10.43 | 10.10| 10.30| 10.39 |

Video Collection

Our dataset contains both visual and textual video data. Videos are collected from the first nine seasons of The Big Bang Theory TV show, with 207 episodes of about 20 minutes long each. For the textual data, we obtained the subtitles directly from the DVDs. Additionally, we downloaded episode transcripts from a specialised website[^1]. Whereas subtitles are annotated with temporal information, transcripts associate dialog with characters. We align subtitles and transcripts with dynamic programming so that each subtitle is annotated to both its speaker and its timestamp. Transcripts also contain scene information, which is used to segment each episode into video scenes. Scenes are split uniformly into 20 seconds clips, obtaining 12,264 clips in total.

QA Generation

To generate real-world natural language questions and answers, we used Amazon Mechanical Turk (AMT[^2]). We required workers to have a high knowledge about The Big Bang Theory and instructed them to write knowledge-based questions about the show. Our aim was to generate questions answerable only by people familiar with the show, whereas difficult for new spectators. For each clip, we showed workers the video and subtitles, along with a link to the episode transcript and summaries of all the episodes for extra context. Workers were asked to annotate each clip with a question, its correct answer, and three wrong but relevant answers. The QA generation process was done in batches of one season at a time in two different rounds. During the second round, we showed the already collected data for each clip in order to 1) get feedback on the quality of the collected data and 2) obtain a diverse set of questions. The QA collection process took about 3 months.

Knowledge Annotations

We define knowledge as the information that is not contained in a given video clip. To approximate the knowledge the viewers acquire by watching the series, we annotated each QA pair with expert information:

- **KNOWLEDGE**: the information that is required to answer the question represented by a short sentence. For example, for the question *Why did Leonard invite Penny to lunch?*, the information *Penny has just moved in* is key to respond the correct answer, *He wanted Penny to feel welcomed into the building*, over the other three candidates[^1].
- **KNOWLEDGE TYPE**: whether the knowledge is from the same episode (episode-specific) or it occurs repeatedly during the show (recurrent). The distribution between the two classes is shown in Fig. 2[^3], with 6.08% of the samples being recurrent and the rest being almost uniformly distributed over the nine seasons.
- **QUESTION TYPE**: we establish four different types of questions: 1) visual-based (22%), in which the answer is found in the video frames, 2) textual-based (12%), in which the answer is found in the subtitles, 3) temporal-based (4%), in which the answer is predictable from the current video clip at a specific time, 4) knowledge-based (62%), in which the answer is not found in the current clip, but in another sequence of the show. To encourage the development of general purpose models, QUESTION TYPE is only provided for the samples in the test set.

More details and examples in the supplementary material.

Data Splits

In total, we collected 24,282 samples from 12,087 video clips. We randomly split the episodes into training, validation, and test sets, so that questions and clips from the same episode were assigned to the same set. The number of episodes, clips, and QA pairs in each split are detailed in Table 2[^4], as well as the average number of tokens in subtitles, questions, answers, and KNOWLEDGE. Correct answers (CA) are slightly longer than wrong answers (WA), which is a common bias in QA datasets [Tapaswi et al. 2016][Lei et al. 2018].

Dataset Comparison

In Table 1[^5] we compare our dataset against other VideoQA and KBVQA datasets. KBVQA datasets are usually smaller than standard VQA datasets, as QA generation is often more challenging. Nevertheless, KnowIT VQA with 24k questions is the largest KBVQA human-generated dataset, far

[^1]: https://bigbangtrans.wordpress.com/
[^2]: https://www.mturk.com
[^3]: Figure 2: Number of questions by KNOWLEDGE TYPE.
[^4]: 1) Because he didn’t have enough money to eat alone, 2) Because he wanted Sheldon to practice his social skills, and 3) Because he was in love with Penny.
[^5]: Table 1: Comparing our dataset with other VQA and KBVQA datasets.
from the 2.4k questions in KB-VQA, 5.8k in FVQA, and 14k in OK-VQA. Also, KnowIT VQA is the first collection addressing the four aforementioned types of questions.

Note that the visual domain in KnowIT VQA is not new, sharing a small portion of videos (about 34%) with TVQA. However, whereas TVQA uses 3.6k clips per show in average, the KNOWLEDGE annotations in our dataset required a larger set of clips, in order to approximate the knowledge that spectators acquire by watching the series.

**Human Evaluation**

We performed human evaluation on the KnowIT VQA test set with a four-fold aim: 1) to evaluate whether video clips are relevant to answer questions; 2) to evaluate whether the questions do require knowledge to be answered; 3) to evaluate whether the KNOWLEDGE annotations are useful for answering the questions in the dataset; and 4) to introduce a human performance baseline for model comparison.

**Evaluation Design** We used AMT with independent groups of workers for each task. We split workers according to their experience with the show, i.e., masters, who have watched at least the first nine seasons of the show, and rookies, who have never watched any episode. We conducted two main tasks: evaluation on the questions and evaluation on the KNOWLEDGE annotations.

**Evaluation on the questions** We further split masters and rookies into 3 different sub-groups according to the data provided to answer each question: Blind (only QAs), Subs (QAs and subtitles), and Video (QAs, subtitles, and clips). For each question in the test set, we asked workers to choose the correct answer from the four given candidates and to provide the reason for their response, from six possible options. In each group, each question was answered by 3 workers. Results are reported in Table 3. The accuracy gap between Subs and Video groups confirms the relevance of the video content in the dataset. With respect to knowledge, the difference between masters and rookies strongly supports the claim that KnowIT VQA is extremely challenging when not knowing the show. When looking into the reasons for choosing the answer (Fig. 3), we saw that masters mostly based their choices on the knowledge acquired when watching the show, whereas rookies admitted not knowing the correct answer in most of their responses.

**Evaluation on the knowledge** We studied the quality of the collected KNOWLEDGE and its relevance to the questions in the dataset. We asked a group of rookies to answer the questions in the test set. For each question and candidate answers, we provided the subtitles and video clip. After answering, we showed the associated KNOWLEDGE and asked them to answer again. As a result, we found that the accuracy increased from 0.683 (before KNOWLEDGE exposition) to 0.823 (after KNOWLEDGE exposition), verifying the relevance of the KNOWLEDGE annotations to the questions.

**ROCK Model**

We propose ROCK (Retrieval Over Collected Knowledge), a model for addressing knowledge-based visual question answering in videos, as depicted in Fig. 4. ROCK is based on the availability of language instances representing the show information in a knowledge base (KB), which retrieves and fuses with language and spatio-temporal video representations for reasoning over the question and predict the correct answer.

**Knowledge Base**

We first create a knowledge base (KB) to emulate the knowledge a viewer acquires when watching the series. Differently from previous work (Wu et al. 2016; Wang et al. 2018), which is based on generic knowledge graphs, such as DBpedia (Auer et al. 2007), our problem requires to access specific information about the show. Thus, we rely on the AMT workers annotations provided in the KNOWLEDGE field during the dataset collection.

The collected KNOWLEDGE is provided as natural language sentences. For example, to the question What was Raj doing at Penny’s?, the annotated KNOWLEDGE is:

Raj wanted to ask Missy on a date, because Howard and Leonard had already asked her but failed, however his medication wore off and he couldn’t do it.

As future work, we plan to study how to automatically learn to generate similar explanations from another module that directly ‘watches’ the series and extracts knowledge from videos.

### Table 3: Human evaluation on KnowIT VQA test set.

| Group           | Acc  | Group      | Acc  |
|-----------------|------|------------|------|
| Rookies, Blind  | 0.440| Masters, Blind | 0.651|
| Rookies, Subs   | 0.562| Masters, Subs  | 0.789|
| Rookies, Video  | 0.748| Masters, Video  | 0.896|

![Figure 3: Distribution of reasons for answering by groups.](image-url)
As it is unclear how to capture such complex processes in an structured fashion such as knowledge graphs, we build a KB, \( K = \{ w_j | j = 1, \ldots, N \} \), such that knowledge instances \( w_j \)'s are represented as natural language sentences. We additionally performed a cleaning process to remove near-duplicate instances, reducing \( N \) from 24,282 to 19,821.

Details are provided in the supplementary material.

**Knowledge Retrieval Module**

Inspired by the ranking system in (Nogueira and Cho 2019), the knowledge retrieval module uses a question \( q_i \) and its candidate answers \( a^c_i \) with \( c \in \{0,1,2,3\} \) to query the knowledge base \( K \) and rank knowledge instances \( w_j \in K \) according to a relevance score \( s_{ij} \).

We first obtain a sequence input representation \( x_{ij} \) as a concatenation of strings:

\[
x_{ij} = [\text{CLS}] + q_i + a^0_i + a^1_i + a^2_i + a^3_i + [\text{SEP}] + w_j + [\text{SEP}],
\]

where \([\text{CLS}]\) is the token to indicate the start of the sequence and \([\text{SEP}]\) is the token to separate semantic groups, in this case the input text used for querying and the knowledge to be queried. Although preliminary experiments showed that the order of the answers \( a^c_i \) does not have a high impact on the results, for an invariant model we automatically sort the answers according to a prior relevance score \( \alpha_c \).

\( \alpha_c \) is then the original position of the answer with \( c \)-th highest score. Details on the prior score computation are provided in the supplementary material.

We tokenise \( x_{ij} \) into a sequence of \( n \) words \( x_{ij} \) and input it into a BERT network (Devlin et al. 2019), namely BERT-scoring denoted by \( \text{BERT}_S(x_{ij}) \), whose output is the vector corresponding to the \([\text{CLS}]\) token. To compute \( s_{ij} \), we use a fully connected layer together with a sigmoid activation as:

\[
s_{ij} = \text{sigmoid}(w^T_S \cdot \text{BERT}_S(x_{ij}) + b_S),
\]

where \( w_S \) and \( b_S \) are the weight vector and the bias scalar of the fully connected layer, respectively. \( \text{BERT}_S^T, w_S, \) and \( b_S \)

are fine-tuned using matching (i.e. \( i = j \)) and non-matching (i.e. \( i \neq j \)) QA-knowledge pairs with the following loss:

\[
\mathcal{L} = - \sum_{i \neq j} \log(s_{ij}) - \sum_{i \neq j} \log(1 - s_{ij})
\]

For each \( q_i \), all \( w_j \)'s in \( K \) are ranked according to \( s_{ij} \). The top \( k \) ranked instances, i.e. the most relevant samples for the query question, are retrieved.

**Video Reasoning Module**

In this module, the retrieved knowledge instances are jointly processed with the multi-modal representations from the video content to predict the correct answer. This process contains three components: visual representation, language representation, and answer prediction.

**Visual Representation**

We sample \( n_f \) frames from each video clip and apply four different techniques to describe their visual content:

- **Image features**: Each frame is fed into Resnet50 (He et al. 2016) without the last fully-connected layer and is represented by a 2,048-dimensional vector. We concatenate all vectors from the \( n_f \) frames and condense it into a 512-dimensional vector using a fully-connected layer.

- **Concepts features**: For a given frame, we use the bottom-up object detector (Anderson et al. 2018) to obtain a list of objects and attributes. We encode all the objects and attributes in the \( n_f \) frames into a \( C \)-dimensional bag-of-concept representation, which is projected into a 512-dimensional space with a fully-connected layer. \( C \) is the total number of available objects and attributes.

- **Facial features**: We use between 3 to 18 photos of the main cast of the show to train the state-of-the-art face recognition network in (Parkhi et al. 2015). For each

9Characters trained in the face recognition network are: Amy, Barry, Bernadette, Dr. Beverly Hofstadter, Dr. VM Koothrappali, Emily, Howard, Leonard, Leslie, Lucy, Mary Cooper, Penny, Priya, Raj, Sheldon, Stuart, and Wil Wheaton.
clip, we encode the detected faces as a $F$-dimensional bag-of-faces representation, which is projected into a 512-dimensional space with a fully-connected layer. $F$ is the total number of people trained in the network.

- **Caption features:** For each frame, we generate a caption to describe its visual content using [Xu et al. 2015]. The $n_f$ captions extracted from each clip are passed to the language representation model.

### Language Representation

Text data is processed using a fine-tuned BERT model, namely BERT-conditioning. We compute the language input, $y^c$, as a concatenation of strings:

$$y^c = [CLS] + caps + subs + q + [SEP] + a^c + w + [SEP],$$

where $caps$ is the concatenated $n_f$ captions (ordered by timestamp), $subs$ the subtitles, and $w$ the concatenated $k$ retrieved knowledge instances. For each question $q$, four different $y^c$ are generated, one for each of the candidate answers $a_i^c$ with $c = \{0, 1, 2, 3\}$. We tokenize $y^c$ into a sequence of $m$ words, $y^c$, as in BERT-scoring. Let $BERT_R$ denote BERT-conditioning, whose output is the vector corresponding to the [CLS] token. For $a_i^c$, the language representation $u^c$ is obtained as $u^c = BERT_R(y^c)$.

### Answer Prediction

To predict the correct answer, we concatenate the visual representation $v$ (i.e. image, concepts, or facial features) with one of the language representations $u^c$:

$$z^c = [v, u^c]_t,$$

$z^c$ is projected into a single score, $o^c$, with a fully-connected layer:

$$o^c = w_R^T z^c + b_R,$$

The predicted answer $\hat{a}$ is obtained with the index of the maximum value in $o = (o^0, o^1, o^2, o^3)^T$, i.e., $\hat{a} = \underset{a}{\text{arg max}} o$. Being $c^*$ the correct class, $BERT_R$, $w_R$, and $b_R$ are fine-tuned with the multi-class cross-entropy loss as:

$$\mathcal{L}(o, o^*) = -\log \frac{\exp(o^*_{c^*})}{\sum_c \exp(o^c)}.$$ (5)

### Experimental Results

We evaluated and compared ROCK against several baselines on the KnowIt VQA dataset. Results per question type and overall accuracy are reported in Table 4. Models were trained with stochastic gradient descent with momentum of 0.9 and learning rate of 0.001. In BERT implementations, we used the uncased base model with pre-trained initialization.

### Answers

To detect potential biases in the dataset, we evaluated the accuracy of predicting the correct answer by only considering the candidate answers:

- **Longest/Shortest:** The predicted answer is the one with the largest/smallest number of words.
- **word2vec/BERT sim:** For word2vec, we use 300-dimensional pre-trained word2vec vectors [Mikolov et al. 2013]. For BERT, we encode words with the output of the third-to-last layer of pre-trained BERT. Answers are encoded as the mean of their word representations. The prediction is the answer with the highest cosine similarity to the other candidates in average.

In general, these baselines performed very poorly, with only Longest being better than random. Other than the tendency of correct answers to be longer, results do not show any strong biases in terms of answer similarities.

### QA

We also evaluated several baselines in which only questions and candidate answers are considered.

- **word2vec/BERT sim:** Questions and answers are represented by the mean word2vec or pre-trained BERT word representation. The predicted answer is the one with highest cosine similarity to the question.
- **TFIDF:** Questions and answers are represented as a weighted frequency word vector (tf-idf) and projected into a 512-dimensional space. The question and the four candidate answers are then concatenated and input into a four-class classifier to predict the correct answer.
- **LSTM Emb./BERT:** Each word in a question or in a candidate answer is encoded through an embedding layer or a pre-trained BERT network and input into an LSTM (Hochreiter and Schmidhuber 1997). The last hidden state of the LSTM is used as a 512-dimensional sentence representation. Question and answers are concatenated and input into a four-class classifier for prediction.

| Model | Vis. | Text. | Temp. | Know. | All |
|-------|------|-------|-------|-------|-----|
| Random | 0.250 | 0.250 | 0.250 | 0.250 | 0.250 |
| Longest | 0.324 | 0.308 | 0.395 | 0.342 | 0.336 |
| Shortest | 0.241 | 0.236 | 0.233 | 0.297 | 0.275 |
| word2vec sim | 0.166 | 0.196 | 0.233 | 0.189 | 0.186 |
| BERT sim | 0.199 | 0.239 | 0.198 | 0.226 | 0.220 |
| word2vec sim | 0.108 | 0.163 | 0.151 | 0.180 | 0.161 |
| BERT sim | 0.174 | 0.264 | 0.209 | 0.190 | 0.196 |
| TFIDF | 0.434 | 0.377 | 0.488 | 0.485 | 0.461 |
| LSTM Emb. | 0.444 | 0.428 | 0.512 | 0.515 | 0.489 |
| LSTM BERT | 0.446 | 0.464 | 0.500 | 0.532 | 0.504 |
| ROCKQA | 0.542 | 0.475 | 0.547 | 0.535 | 0.530 |
| Humans (Rookies, Blind) | 0.406 | 0.407 | 0.416 | 0.418 | 0.417 |
| ROCKQA | 0.432 | 0.362 | 0.512 | 0.496 | 0.467 |
| LSTM BERT | 0.452 | 0.446 | 0.547 | 0.530 | 0.504 |
| TVQA | 0.602 | 0.551 | 0.512 | 0.468 | 0.509 |
| ROCKQA | 0.651 | 0.754 | 0.593 | 0.534 | 0.587 |
| Humans (Rookies, Subs) | 0.618 | 0.837 | 0.453 | 0.498 | 0.562 |
| TVQA | 0.612 | 0.645 | 0.547 | 0.466 | 0.522 |
| ROCKQA Image | 0.643 | 0.739 | 0.581 | 0.539 | 0.587 |
| ROCKQA Concepts | 0.647 | 0.743 | 0.581 | 0.538 | 0.587 |
| ROCKQA Facial | 0.649 | 0.743 | 0.581 | 0.537 | 0.587 |
| ROCKQA Caption | 0.666 | 0.772 | 0.581 | 0.514 | 0.580 |
| Humans (Rookies, Video) | 0.936 | 0.932 | 0.624 | 0.655 | 0.748 |
| ROCKQA Image | 0.654 | 0.685 | 0.628 | 0.646 | 0.652 |
| ROCKQA Facial | 0.654 | 0.688 | 0.628 | 0.646 | 0.652 |
| ROCKQA Caption | 0.647 | 0.678 | 0.593 | 0.643 | 0.646 |
| ROCKQA | 0.747 | 0.819 | 0.756 | 0.708 | 0.731 |
| Humans (Masters, Video) | 0.961 | 0.936 | 0.857 | 0.867 | 0.896 |
What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?

Wednesday

Thursday

Friday

Saturday

Monday

What night is it?
References

[Anderson et al. 2018] Anderson, P.; He, X.; Buehler, C.; Teney, D.; Johnson, M.; Gould, S.; and Zhang, L. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In Proc. CVPR, 6077–6086.

[Antol et al. 2015] Antol, S.; Agrawal, A.; Lu, J.; Mitchell, M.; Batra, D.; Lawrence Zitnick, C.; and Parikh, D. 2015. VQA: Visual question answering. In Proc. ICCV, 2425–2433.

[Auer et al. 2007] Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; and Ives, Z. 2007. DBpedia: A nucleus for a web of open data. In The Semantic Web. Springer. 722–735.

[Bai et al. 2018] Bai, Y.; Fu, J.; Zhao, T.; and Mei, T. 2018. Deep attention neural tensor network for visual question answering. In Proc. ECCV, 20–35.

[Ben-Younes et al. 2017] Ben-Younes, H.; Cadene, R.; Cord, M.; and Thome, N. 2017. MUTAN: Multimodal tucker fusion for visual question answering. In Proc. ICCV, 2612–2620.

[Devlin et al. 2019] Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proc. NAACL, 4171–4186.

[Frermann, Cohen, and Lapata 2018] Frermann, L.; Cohen, S. B.; and Lapata, M. 2018. Whodunnit? Crime drama as a case for natural language understanding. Trans. ACL 6:1–15.

[Goyal et al. 2017] Goyal, Y.; Khot, T.; Summers-Stay, D.; Batra, D.; and Parikh, D. 2017. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In Proc. CVPR, 6904–6913.

[He et al. 2016] He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In Proc. CVPR, 770–778.

[Hochreiter and Schmidhuber 1997] Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. Neural Computation 9(8).

[Jang et al. 2017] Jang, Y.; Song, Y.; Yu, Y.; Kim, Y.; and Kim, G. 2017. TGIF-QA: Toward spatio-temporal reasoning in visual question answering. In Proc. CVPR, 2758–2766.

[Johnson et al. 2017] Johnson, J.; Hariharan, B.; van der Maaten, L.; Fei-Fei, L.; Zitnick, C. L.; and Girshick, R. 2017. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In Proc. CVPR, 1988–1997.

[Kim et al. 2017a] Kim, J.-H.; On, K.-W.; Lim, W.; Kim, J.; Ha, J.-W.; and Zhang, B.-T. 2017a. Hadamard product for low-rank bilinear pooling. Proc. ICLR.

[Kim et al. 2017b] Kim, K.-M.; Heo, M.-O.; Choi, S.-H.; and Zhang, B.-T. 2017b. DeepStory: Video story QA by deep embedded memory networks. In Proc. IJCAI, 2016–2022.

[Krishna et al. 2017] Krishna, R.; Zhu, Y.; Groth, O.; Johnson, J.; Hata, K.; Kravitz, J.; Chen, S.; Kalantidis, Y.; Li, L.-J.; Shamma, D. A.; et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. IJCV 123(1):32–73.

[Lei et al. 2018] Lei, J.; Yu, L.; Bansal, M.; and Berg, T. L. 2018. Tvqa: Localized, compositional video question answering. In Proc. EMNLP, 1369–1379.

[Lu et al. 2018] Lu, P.; Ji, L.; Zhang, W.; Duan, N.; Zhou, M.; and Wang, J. 2018. R-vqa: learning visual relation facts with semantic attention for visual question answering. In Proc. KDD, 1880–1889.

[Maharaj et al. 2017] Maharaj, T.; Ballas, N.; Rohrbach, A.; Courville, A.; and Pal, C. 2017. A dataset and exploration of models for understanding video data through fill-in-the-blank question-answering. In Proc. CVPR, 6884–6893.

[Malinowski and Fritz 2014] Malinowski, M., and Fritz, M. 2014. A multi-world approach to question answering about real-world scenes based on uncertain input. In Proc. NIPS, 1682–1690.

[Marino et al. 2019] Marino, K.; Rastegari, M.; Farhadi, A.; and Mottaghi, R. 2019. Ok-vqa: A visual question answering benchmark requiring external knowledge. In Proc. CVPR, 3195–3204.

[Mikolov et al. 2013] Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In Proc. NIPS.

[Mun et al. 2017] Mun, J.; Hongsuck Seo, P.; Jung, I.; and Han, B. 2017. MarioQA: Answering questions by watching gameplay videos. In Proc. ICCV, 2867–2875.

[Nagrani and Zisserman 2017] Nagrani, A., and Zisserman, A. 2017. From Benedict Cumberbatch to Sherlock Holmes: Character identification in TV series without a script. In Proc. BMVC.

[Nogueira and Cho 2019] Nogueira, R., and Cho, K. 2019. Passage re-ranking with bert. arXiv preprint arXiv:1901.04085.

[Parkhi et al. 2015] Parkhi, O. M.; Vedaldi, A.; Zisserman, A.; et al. 2015. Deep face recognition. In Proc. BMVC, 1–6.

[Shah et al. 2019] Shah, S.; Mishra, A.; Yadati, N.; and Talukdar, P. P. 2019. KVQA: Knowledge-aware visual question answering. In Proc. AAAI.

[Tapaswi et al. 2016] Tapaswi, M.; Zhu, Y.; Stiefelhagen, R.; Torralba, A.; Urtasun, R.; and Fidler, S. 2016. MovieQA: Understanding stories in movies through question-answering. In Proc. CVPR, 4631–4640.

[Wang et al. 2017] Wang, P.; Wu, Q.; Shen, C.; Dick, A.; and Van Den Hengel, A. 2017. Explicit knowledge-based reasoning for visual question answering. In Proc. IJCAI, 1290–1296.

[Wang et al. 2018] Wang, P.; Wu, Q.; Shen, C.; Dick, A.; and van den Hengel, A. 2018. FVQA: Fact-based visual question answering. IEEE Trans. PAMI 40(10):2413–2427.

[Wu et al. 2016] Wu, Q.; Wang, P.; Shen, C.; Dick, A.; and van den Hengel, A. 2016. Ask me anything: Free-form vi-
sual question answering based on knowledge from external sources. In Proc. CVPR, 4622–4630.

[Xu et al. 2015] Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R.; Zemel, R.; and Bengio, Y. 2015. Show, attend and tell: Neural image caption generation with visual attention. In Proc. ICML, 2048–2057.

[Zellers et al. 2019] Zellers, R.; Bisk, Y.; Farhadi, A.; and Choi, Y. 2019. From recognition to cognition: Visual commonsense reasoning. In Proc. CVPR.

[Zhu et al. 2015] Zhu, Y.; Zhang, C.; Ré, C.; and Fei-Fei, L. 2015. Building a large-scale multimodal knowledge base system for answering visual queries. arXiv preprint arXiv:1507.05670.

[Zhu et al. 2017] Zhu, L.; Xu, Z.; Yang, Y.; and Hauptmann, A. G. 2017. Uncovering the temporal context for video question answering. IJCV 124(3):409–421.
Supplementary Material

In this document, we provide more details about the KnowIT VQA dataset as well as additional ablation studies and insights of our proposed ROCK model. Specifically, this document contains:

A. KnowIT VQA Analysis
B. Data Collection Interface
C. Knowledge Base Cleaning
D. Knowledge Retrieval Details
E. Examples

A. KnowIT VQA Analysis

The fields provided for each sample in the dataset are described in Table 6 and a more detailed analysis of the contents of the dataset is provided below.

Question Types

The distribution of question types in the test set is as follows: 22% are visual-based, 12% are textual-based, 4% are temporal-based, and 62% are knowledge-based. We further study the differences between the four question types by plotting the distribution of question words in each class (Figure 6). The most frequent question word in textual-based, temporal-based, and knowledge-based classes is ‘What’, whereas in visual-based questions is ‘Who’, indicating that person identification is more prominent in video-related tasks. Interestingly, ‘Why’ question words only appear in a significant amount in knowledge-based and textual-based questions, being non-frequent in questions that are more related to the video content, such as visual-based and temporal-based.

Answers

For the total amount of 24,282 questions in the dataset, there are 59,523 unique candidate answers, with 15,293 unique correct answers. Table 7 shows the 20 most frequent words in the candidate answers for different question words. Note that the name of the main characters appear repeatedly in all the question words. Figure 7 shows the distribution of the 20 most frequent answers in the ‘Who’ questions. Both candidate answers and correct answers are almost uniformly distributed between the five main characters of the show.

KNOWLEDGE Annotations

The KNOWLEDGE field provides the information that is not contained in the video clip but it is necessary to answer the question. KNOWLEDGE can be recurrent (i.e. it appears in many episodes of the show) or episode-specific (i.e. it appears in a specific episode of the show). Some examples of recurrent and episode-specific KNOWLEDGE are shown in Figure 8. The joint distribution between the question type and the KNOWLEDGE type for the questions in the test set is shown in Figure 9.
Table 6: Data fields for each sample in the KnowIT VQA dataset. * indicates in the test set only.

| Field         | Description                          | Type | Values                      |
|---------------|--------------------------------------|------|-----------------------------|
| VIDEO CLIP    | Filename of the video clip           | str  |                             |
| SUBTITLES     | Subtitles annotated with speakers    | str  | Multiple sentences          |
| QUESTION      | Question                             | str  | Sentence                    |
| ANSWER1       | First candidate answer               | str  | Sentence                    |
| ANSWER2       | Second candidate answer              | str  | Sentence                    |
| ANSWER3       | Third candidate answer               | str  | Sentence                    |
| ANSWER4       | Fourth candidate answer              | str  | Sentence                    |
| QUESTION TYPE*| Type of question                     | str  | Visual, textual, temporal, or knowledge |
| KNOWLEDGE     | Information to answer the question   | str  | Sentence                    |
| KNOWLEDGE TYPE| Type of the annotated KNOWLEDGE     | str  | Episode-specific or recurrent |
| INDEX         | Index of the correct answer          | int  | 1, 2, 3, or 4               |

Table 7: Most frequent words in candidate answers.

| QWord | Words in answers                          |
|-------|-------------------------------------------|
| What  | sheldon, penny, leonard, howard, amy, raj, new, go, going, bernadette, wants, get, date, star, comic, book, got, work, apartment, movie |
| Who   | leonard, howard, raj, sheldon, penny, amy, bernadette, emily, stuart, leslie, barry, mother, mom, stephanie, sister, lucy, priya, new, kripke, zack |
| Why   | sheldon, leonard, penny, howard, wants, raj, amy, want, get, bernadette, going, work, go, trying, wanted, date, like, thinks, new, apartment |
| Where | apartment, penny, leonard, sheldon, howard, store, comic, raj, university, house, book, factory, cheesecake, amy, room, bernadette, lab, work, office, bedroom |
| Whose  | rajs, pennys, howards, leonard, sheldon, sheldons, leonards, penny, howard, raj, bernadette, amy, mom, bernadettes, amys, leslie, stuart, professor, wheaton, barry |
| How   | sheldon, penny, leonard, two, three, howard, dollars, one, told, raj, amy, four, months, five, years, hundred, bernadette, work, went, got |

B. Data Collection Interface

Figure 10 displays the interface that we show the AMT workers in order to create the QA pairs and the KNOWLEDGE annotations. We collected data per batches of one season at a time in two rounds. During the second round, we provided the previously generated question and correct answer for that scene.

C. Knowledge Base Cleaning

During the construction of the knowledge base (KB), \( K \) with \( N \) KNOWLEDGE instances, we found that a number of KNOWLEDGE instances were repeated in different scenes. This was especially true for the instances associated to the visual-based and textual-based questions, as KNOWLEDGE is often not required to answer these types of questions. In those cases, the KNOWLEDGE annotated by workers may have not been compliant with the provided knowledge definition (see an example in Table 8 row 2). To reduce the computation time, we implemented a cleaning process to remove near duplicate instances.

First we computed a high-dimensional projection of each KNOWLEDGE instance with the pre-trained uncased BERT network. Formally, for each \( w_j \in K \), we created the input sentence, \( w'_j \), as a concatenation of strings:

\[
w'_j = [\text{CLS}] + w_j + [\text{SEP}],
\]

which we tokenised into a sequence of 60 tokens, \( \tau_j \).

Let \( \text{BERT}_P(\cdot) \) be the pre-trained BERT network, which takes as input a sequence of tokens and outputs the vector corresponding to the [CLS] token. We obtained the high-dimensional projection, \( p_j \), of \( w'_j \) as:

\[
p_j = \text{BERT}_P(\tau_j),
\]

(6)

To measure similarity between a pair of instances, \( w_i, w_j \in K \), we computed a similarity score, \( \beta_{ij} \), as:

\[
\beta_{ij} = \text{sim}(p_i, p_j),
\]

(7)

where \( \text{sim}(\cdot, \cdot) \) is the cosine similarity.

Next, we built an undirected graph, \( G = (V, E) \), in which nodes, \( V = \{w_j | j = 1, \cdots, N\} \), corresponded to KNOWLEDGE instances, and edges, \( e = (w_i, w_j) \in E \), connected
instances when $\beta_{ij} > 0.998$\textsuperscript{10}

To find near duplicate instances, we created clusters of nodes, $C_l$ with $l = 1, \cdots, L$, by finding all the connected components in $G$, i.e. $C_l$ corresponded to the $l$-th subgraph in $G$, for which all nodes were connected to each other by a path of edges. We randomly chose one of the nodes in each cluster $w_r \in C_l$ and removed all the others. The final cleaned knowledge base, $K_c$, was then decreased to size $L$. Examples of KNOWLEDGE instances before and after the cleaning process are shown in Table 8.

\textbf{Table 8:} Instances in the KB before and after cleaning.

| After Cleaning | Before Cleaning |
|----------------|----------------|
| Bert is attracted to Amy. | Bert is romantically interested in Amy. Bert is attracted to Amy. |
| We can see them talking. | We can hear them speaking. We can hear them talking. We can see them together speaking. |
| It was a present for Penny. | It was a gift for Penny. It was a present for Penny. It was a gift for Penny. |
| Raj has selective mutism. | Raj has selective mutism. Raj has selective mutism. |
| Amy works at the university. | Amy works at the university. |

\textbf{D. Knowledge Retrieval Details}

In this section, we provide additional details and ablation studies for the ROCK knowledge retrieval module.

\textbf{Prior Score Computation Method}

In the knowledge retrieval module, we concatenate the four candidate answers into the input sentence string, $x_{ij}$. The model can produce different outputs even for the same set of candidate answers when the order of the candidate answers in the input sentence string is randomly altered. To prevent this behaviour, we create an answer order-invariant model by sorting answers, $a^c$ with $c = \{0, 1, 2, 3\}$, according to a prior score, $\xi^c$.

For a given question $q$, $\xi^c$ is obtained from predicting the score of $a^c$ being the correct answer. We first build an input sentence $e^c$ as the concatenation of the strings:

$$e^c = [\text{CLS}] + q + [\text{SEP}] + a^c + [\text{SEP}],$$

and we tokenised $e^c$ into a sequence of 120 tokens, $e$. If $\text{BERT}_E(\cdot)$ represents a BERT network whose output is the vector corresponding to the $[\text{CLS}]$ token, $\xi^c$ is obtained as:

$$\xi^c = w_E^T \text{BERT}_E(e^c) + b_E,$$  \hspace{1cm} (8)

Finally, all $\xi^c$ with $c = \{0, 1, 2, 3\}$ are sorted in descending order into $\xi$ and answers are ordered according to $\alpha_c = \delta$, where $\delta$ is the position of the $\delta$-th highest score in $\xi$. $\text{BERT}_E$, $w_E$, and $b_E$ are fine-tuned with the multi-class cross-entropy loss as in Eq. (5).

\textbf{Parameter Sharing Method}

In Table 9, we compared the prior score computation method against a parameter sharing method for the knowledge retrieval module. In the parameter sharing method, instead of concatenating the four answers in the input string, $x_{ij}$, only one answer, $a^c$, at a time is used as $x_{ij} = [\text{CLS}] + q_i + a_i^c + [\text{SEP}] + w_j + [\text{SEP}]$, which means that the same parameters are used for the four candidate answers. A score $s_{ij}^c$ is computed for each of the answers as in Eq. 1. The final score is obtained as $s_{ij} = \text{sigmoid}(\sum_c s_{ij}^c)$.

\textbf{Ablation Study}

Here we show how the parameters of the knowledge retrieval module affects the overall ROCK model.

- Size of the knowledge base, $N$. Table 9 shows the results of the knowledge retrieval module and the overall ROCK accuracy when using two different sizes of the KB. The All KB includes all 19,823 KNOWLEDGE instances in the dataset, whereas the Split KB only considers the
Table 9: Size of the KB impact on the overall accuracy. N indicates the number of KNOWLEDGE instances used at test time.

| Method         | KB       | N   | R@1 | R@5  | R@10 | R@100 | MR  | ROCK Acc |
|----------------|----------|-----|-----|------|------|-------|-----|---------|
| QA param sharing | All      | 19,823 | 0.083 | 0.197 | 0.268 | 0.557 | 67  | 0.596   |
|                | Split    | 2,078  | 0.155 | 0.386 | 0.496 | 0.819 | 11  | 0.671   |
| QA prior scores | All      | 19,823 | 0.114 | 0.259 | 0.318 | 0.576 | 53  | 0.652   |
|                | Split    | 2,078  | 0.180 | 0.409 | 0.508 | 0.814 | 10  | 0.713   |

Figure 11: Accuracy for different k retrieved KNOWLEDGE.

instances in the training, validation, or test set respectively. It can be seen that the smaller the KB, the easier to retrieve the associated KNOWLEDGE. The correlation between the retrieval module performance and the overall accuracy strongly suggests that the annotated KNOWLEDGE is very important for this task.

- Number of retrieved instances, k. Figure [11] shows the overall accuracy for different number of retrieved KNOWLEDGE samples. In general, the higher k, the higher the probability of retrieving the correct KNOWLEDGE and the better overall performance. Accuracy saturates from k = 5, since the input sequence becomes too large when more than 5 KNOWLEDGE instances are used.

E. Examples

In Figure[12] we compare visual results when using KNOWLEDGE and when not using KNOWLEDGE. In general, results are improved when the retrieved KNOWLEDGE is correct, whereas mispredictions are often produced because of a wrongly retrieved KNOWLEDGE.
...exploring a planet similar to Earth in the 1500s.

Raj: Oh, okay, how about this? You can go dressed as a Star Trek science officer...

Sheldon: Uh, just out of curiosity, why didn't you ask Leonard for advice

Howard: My family is the history of heart disease. There's a cave painting

Sheldon: I don't care. There are far too many historical anomalies for my comfort

Penny: I'm sorry. What was I thinking?

Leonard: Is there heart disease in your family?

Howard: Come on, Sheldon. There are so few places I can wear my jester costume.

Sheldon: I'm the wise man.

Sheldon: I'm sorry, I am not going back to the Renaissance fa

Why are they at the hospital?

Who is Sheldon claiming is a wise man?

What is Penny's father's name?

What is Penny's new job?

Who is Penny on the phone with?

Where are Sheldon, Raj, Howard, and Leonard having a meal?

Who is sitting in Sheldon's sport in this scene?

What is Penny's paper about?

Who is Penny's Dad?

Sheldon claims that spot has many perfections that other sitting spots do not have.

Figure 12: Comparison of results between using KNOWLEDGE ([W/ KG]) and not using KNOWLEDGE ([W/O KG]) in the ROCK (Image) model. In the first row, retrieving the correct KNOWLEDGE improves results with respect to when KNOWLEDGE is not used. In the second row, retrieving the incorrect KNOWLEDGE makes the prediction worse. Finally, in the third row, both [W/KG] and [W/O KG] either predict the correct answer or predict a wrong answer.