Image and Vision Computing 101 (2020) 103968

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Explaining VQA predictions using visual grounding and a knowledge base

Felipe Riquelme [a,1,*], Alfredo De Goyeneche [a,1,*], Yundong Zhang [b], Juan Carlos Niebles [b], Alvaro Soto [a]

[a] *Pontificia Universidad Católica de Chile, Chile*
[b] *Stanford University, United States of America*

ARTICLE INFO

Article history:
Received 12 December 2019
Accepted 5 June 2020
Available online 28 June 2020

Keywords:
Deep Learning
Attention
Supervision
Knowledge Base
Interpretability
Explainability

ABSTRACT

In this work, we focus on the Visual Question Answering (VQA) task, where a model must answer a question based on an image, and the VQA-Explanations task, where an explanation is produced to support the answer. We introduce an interpretable model capable of pointing out and consuming information from a novel Knowledge Base (*KB*) composed of real-world relationships between objects, along with labels mined from available region descriptions and object annotations. Furthermore, this model provides a visual and textual explanations to complement the *KB* visualization. The use of a *KB* brings two important consequences: enhance predictions and improve interpretability. We achieve this by introducing a mechanism that can extract relevant information from this *KB*, and can point out the relations better suited for predicting the answer. A supervised attention map is generated over the *KB* to select the relevant relationships from it for each question-image pair. Moreover, we add image attention supervision on the explanations module to generate better visual and textual explanations. We quantitatively show that the predicted answers improve when using the *KB*; similarly, explanations improve with this and when adding image attention supervision. Also, we qualitatively show that the *KB* attention helps to improve interpretability and enhance explanations. Overall, the results support the benefits of having multiple tasks to enhance the interpretability and performance of the model.

1. Introduction

Visual Question Answering (VQA) is a well-explored task in the computer vision community [10,7,28]. The task consists of answering a question formulated in natural language based on the contents of an image. VQA is a challenging task, it requires to parse the question, identifying key syntactic and semantic components (e.g., verbs, nouns), as well as to process the input image, identifying relevant image regions, objects, and their relations. Furthermore, it is also highly desirable to provide an explanation to support the selected answer [22].

Besides all these complexities, humans are incredibly robust to solve the VQA problem and to provide a sound explanation to support their answers. Among the abilities that distinguish how humans may solve the VQA problem, we highlight the following: (i) Their ability to focus attention on image regions that are relevant to answer each question, (ii) Their capability to use appropriate background knowledge, such as common sense knowledge, to construct suitable answers, and (iii) Their ability to support answers with a coherent explanation. In this work, we take inspiration from these three abilities, and we propose a VQA model that incorporates processing modules to (i) Mimic human visual attention, (ii) Exploit external sources with previous visual knowledge, and (iii) Provide an explanation in natural language to support each answer. We elaborate on these three ideas in the following paragraphs.

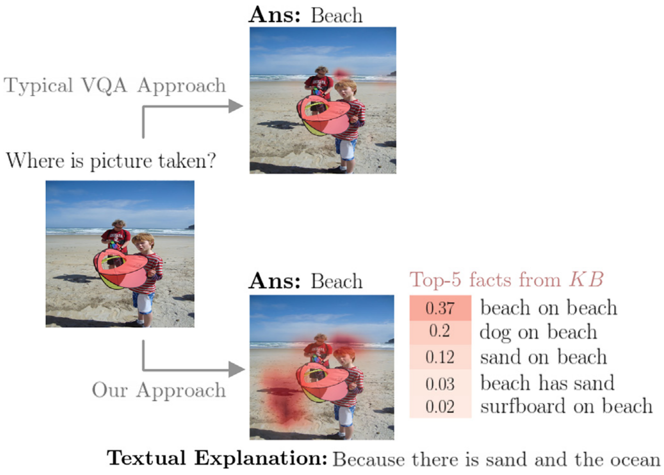


Fig. 1. Current VQA models lack of proper mechanisms to attend suitable image regions, to retrieve relevant facts from an external *KB*, and to support each answer with a suitable textual explanation. Our proposed approach aims to contribute with a model that incorporates these three types of mechanisms.

[*] Corresponding authors.
E-mail addresses: fariquelme@uc.cl (F. Riquelme), asdegoyeneche@uc.cl (A. De Goyeneche).
[1] Equal contribution.

https://doi.org/10.1016/j.imavis.2020.103968
0262-8856/© 2020 Elsevier B.V. All rights reserved.

First, in terms of visual attention, current VQA models usually incorporate a soft attention mechanism that operates over the input image. Specifically, during training, this mechanism uses an unsupervised training scheme that learns how to weight features from different image regions to answer each question [3]. However, recent works have demonstrated the limitations of such unsupervised attention scheme, showing that it leads to models encoding discriminative cues that do not ground image attention to the underlying semantics behind each question-answer pair [8,24].

In this work, we introduce a methodology that provides our VQA model with the ability to generate human interpretable attention maps that effectively ground each answer to relevant image regions. To achieve this, we frame visual attention as a supervised auxiliary task that is jointly trained along with the main VQA task. As a relevant novelty, we avoid costly human annotations by obtaining the attention labels automatically. Specifically, we propose a method that generates these labels by taking advantage of region descriptions and object annotations available in the Visual Genome (VG) dataset [14]. As a result of using this type of supervision, our model generates visual groundings that closely matches human attention annotations.

Second, in terms of background knowledge, VQA problems are particularly challenging in cases where the information that leads to a correct answer is not explicitly present in the question-image pair. To handle these cases, recent methods augment the question-image pair using an external knowledge base (*KB*) that provides complementary information, usually in the form of knowledge triplets (subject, relation, target) [18,28,15]. Unfortunately, current efforts along this line have not been able to bridge the semantic gap between models and human performance, especially in terms of the interpretability of the resulting models. As a relevant limitation, most current approaches use mainly the input question to retrieve relevant facts from the *KB*, ignoring visual cues from the input image.

In this work, we introduce a methodology that provides our VQA model with the ability to use both question and image information to select triplets from a *KB*. These triplets are then used as part of the process to predict an answer and build a supporting explanation. The *KB* is constructed automatically using visual relations extracted from the scene graphs in the VG dataset [14], as well as from question-answer pairs from both the VG and VQA [2] datasets. We employ this data to generate supervision labels that are then used to train an attention mechanism that selects relevant triplets from the *KB*.

Third, in terms of explaining answers, model interpretability is a highly desirable property because it provides a window to the internal representation behind the prediction process of the model. Furthermore, as AI-based systems start to operate in real-world applications, there is an increasing need to provide them with the ability to explain their decisions [9]. In the context of VQA, current approaches usually provide visual attention maps as a proxy to model interpretability [6,37,22]. Recently, [22] introduces a VQA model that includes a module to generate a textual explanation for each answer using natural language.

As shown in Fig. 1, in this work, we provide our model with three mechanisms to access insightful views about its internal representation. First, similarly to [22], our model provides a textual explanation to support its answer. As a relevant novelty, our model builds this explanation by fusing information from three sources: question, image, and *KB*. Second, we provide an attention map that selects *KB* triplets to be used by the model to build both the answer and explanation. Third, we provide visual attention maps over the image regions that are relevant to the question.

In summary, the main contributions of this work are as follows: (i) A model that incorporates visual attention as an auxiliary task that is trained using a supervised learning scheme; (ii) A model that incorporates a *KB* encoding general visual knowledge in the form of triplets that are adaptively selected using question and image information; (iii) A methodology to mine existing datasets, mainly VG, to automatically build a *KB* as well as to extract attention labels to train a visual and a *KB* attention mechanisms; (iv) A model with increased interpretability, as it is able to generate a textual explanation supporting each answer, as well as, attention maps highlighting relevant image regions and *KB* triplets used to build the answer; (v) Extensive experimental evaluation showing that our model increases both VQA accuracy and interpretability with respect to previous approaches.[2]

2. Related work

The VQA problem has attracted considerable attention in the computer vision community [29]. The proliferation of suitable datasets, its multimodal nature, and a simple evaluation protocol help to explain this interest. Most state-of-the-art methods learn to project the textual and visual inputs to a joint feature space that is then used to build the answer [2,7,36]. Most methods pose VQA as a classification problem, where output classes are given by a predefined set of most popular answers. Furthermore, following [3], most models incorporate a soft-attention scheme that is trained to attend to relevant regions in the input image [29,7,35,36]. More elaborated soft-attention mechanisms have also been proposed, such as an iterative attention scheme [34] or a bidirectional co-attention mechanism [17].

Das et al. [6] analyze the visual grounding provided by models based on soft-attention mechanisms. In particular, they compare image areas selected by humans and state-of-the-art VQA techniques to answer the same visual question. Interestingly, they conclude that current machine-generated attention maps exhibit a poor correlation with respect to the human counterpart, suggesting that humans use different visual cues to answer the questions. Relevant to our work, we can find methods that also use a supervised attention mechanism [8,24]. Gan et al. [8] ask humans to select segmented regions in images from the COCO dataset that are relevant to answer visual questions. Afterwards, they use these areas as labels to train a deep learning model that is able to identify attention features. By augmenting a standard VQA technique with these attention features, they are able to achieve a small boost in performance in the VQA task. Qiao et al. [24] uses the attention labels in the VQA-HAT dataset [6] to train a network used to generate attention proposals for each image in the VQA dataset. These proposals are then used as labels to train a VQA model using a supervised attention scheme. This strategy results in a small boost in performance compared with a non-attentional strategy. In contrast, our work incorporates an automatic mechanism to obtain attention labels. Also our work includes a module to generate an explanation that benefits from image attention supervision. Furthermore, our work include information from a *KB*.

In terms of works that use an external *KB* to support or implement a VQA model, Wu et al. [32] augments the usual discriminative approach used by VQA models by introducing information extracted from an external *KB*. Specifically, following an image captioning approach, a set of visual attributes are selected to query the *KB*. Wang et al. [31] introduces the fact-VQA dataset (FVQA) that focuses on including question-image pairs that need a *KB* with previous knowledge to build correct answers. Using a scheme that jointly projects knowledge facts and question-image pairs to a common space, Narasimhan and Schwing [20] achieves state-of-the art results on the FVQA dataset. Su et al. [28] use a Memory Network architecture to jointly embed knowledge facts and visual attentive features. As a result, they report state-of-the-art accuracy on questions related to knowledge-reasoning. In contrast to our approach, these previous methods do not use the information extracted from the *KB* to generate an explanation to justify the corresponding answer.

[2] A preliminary version of our work appeared in [37]. That preliminary version only includes our contributions (i) and part of (iii), which relate to the use of visual attention as an auxiliary task to improve visual grounding.

In terms of explanaibility, recently, Park et al. [22] present a VQA model that includes a module to generate a textual explanation to justify each selected answer. This module follows the standard approach used to generate image captions, but it integrates information from the input question, attended image regions, and selected answer. Using a similar approach, Kim et al. [13] extend the method to the case of videos coming from a self-driving application. In contrast to our work, these previous works use an unsupervised scheme to select relevant visual regions. Furthermore, they do not consider information from an external source of previous knowledge (*KB*).

3. Proposed method

We first present our method to build the *KB* and to extract attention labels to train image and *KB* attention mechanisms. Afterwards, we present the details behind our VQA model.

3.1. Mining from visual genome dataset (VG)

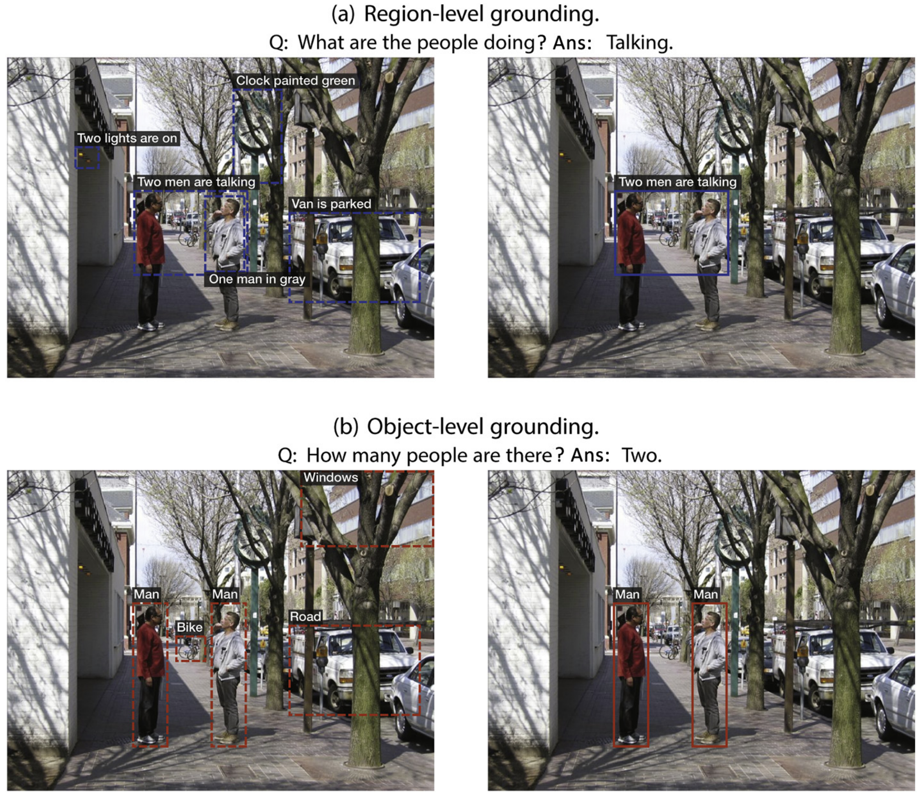
VG is a large dataset with more than 100 K images with dense annotations related to image regions, objects, attributes, and relationships between them [14]. Furthermore, it contains over 1.7 M question-answer pairs. In this work, we use VG to automatically infer three sources of information: i) visual attention labels to ground each question-answer pair, ii) semantic object relations to build the *KB*, and iii) *KB* attention labels indicating the *KB* rules that are relevant to each question-answer. To achieve this, we use two annotations related to images in the question-answer pairs from VG. First, we use the bounding boxes annotated to identify different regions and objects. Second, we use the annotations in the scene graphs which describe semantic relations between the objects annotated in each image, ex. {man, driving, car}. Next, we describe the details behind our approach to mine information from VG.

3.1.1. Generating image attention labels

To generate the image attention labels we use the region and object annotations provided in VG. As shown in Fig. 2, these annotations provide complementary information. Specifically, region annotations are highly relevant for questions related to the interaction between objects. In contrast, object annotations are more valuable for questions related to properties of specific objects. Consequently, we exploit both region and object annotations to generate our image attention labels.

In the case of image regions, for each training instance (I_i, Q_i, A_i) in VG, we find the region description D_i associated with image I_i that has the most number of words in common with the union of the text from the question Q_i and answer A_i. To achieve this, we only consider nouns and verbs as valid words, and we apply a minimum threshold of two common words to select a region. To maximize the chances of word overlapping, we augment the set of candidate words with their synonyms. Specifically, we match region's words if at least one of the following conditions is met: (1) their raw text as they appear in Q_i or A_i is the same; (2) their lemmatizations (using NLTK [5]) are the same; (3) their synsets in WordNet [19] are the same; (4) their aliases (provided from VG) are the same. We refer to the resulting labels as *region-level* groundings. Fig. 2a illustrates an example of a region-level grounding.

In the case of image objects, we select the bounding box of an object as a valid grounding label, if the object label matches at least one word in Q_i or A_i. To score each match, we use the same criteria as region-level groundings. Additionally, if an instance (I_i, Q_i, A_i) has a valid region grounding, each corresponding object-level grounding must be inside this region to be accepted as valid. This is important to avoid wrong grounding due to images containing multiple instances from the same object class. In our experiments, we use as a threshold a minimum


(a) Region-level grounding.
Q: What are the people doing? Ans: Talking.

(b) Object-level grounding.
Q: How many people are there? Ans: Two.

Fig. 2. (a) Example of region-level groundings from VG. Left: image with region description labels; Right: our mined results. Here "men" in the region description is lemmatized to be "man", whose aliases contain "people"; the word "talking" in the answer also contributes to the word matching. So the selected regions have two matchings words which is the most among all candidates. (b) Example of object-level grounding from VG. Left: image with object instance labels; Right: our mined results. Note that in this case region-level grounding outputs the same result as in (a), however, object-level grounding improves the spatial localization of the attended image region.

intersection over union of 60%. We refer to the resulting labels as *object-level* groundings. Fig. 2b illustrates an example of an object-level grounding.

As a result, by combining both region-level and object-level groundings, we obtain about 700 K attention map labels out of 1 M (I_i, Q_i, A_i) triplets in VG. We make these labels publicly available.

3.1.2. Constructing the KB and its attention labels

We design the *KB* to be organized as a collection of triplets that encode knowledge about typical interactions between objects in the real world. Our triplets have the format *{Object (Obj), Relationship (Rel), Subject (Subj)}*, for example: *{Man, Playing, Tennis}*. Instead of manually constructing the triplets in our *KB*, we automatically mine them from the scene graphs in VG. Furthermore, we also mine *KB* rules from the text in the question-answer pairs in both VG and VQA2.0 [10].

For the case of the scene graphs in VG, every node in the graph represents an object present in the image, while edges between them indicate their interactions or relationships. Our goal is to identify relationships in these scene graphs that are relevant to each question-answer pair, and store them in our *KB* as triplets (*{Obj, Rel, Subj}*). To achieve this, we mine from the scene graphs all the relations associated with the region-labels and object-labels selected by the method described in Section 3.1.1.

For the case of directly mining *KB* rules from the text in the question-answer pairs in VG and VQA2.0 [10], we do not use image region information but only text matching. Specifically, we consider all triplets in all the scene graphs from VG and search from question-answer pairs in VG and VQA-2 that match three or more words with the question-answer pairs. For example, if the question-answer pair is *"Q: What is the man eating? A: Hot-dog"*, the words match those from the triplet *{man, eat, hot-dog}* in VG.

Finally, by combining all the triplets from the two methods described above, we gathered over 480 K question-answer pairs with labels, and over 640 K triplets for our *KB*. However, due to memory constraints, we further compress it into smaller *KB*s using 3 different methods that we describe in Section 4.2.

3.2. VQA model

Fig. 3 illustrates the overall architecture of our VQA model. The first step is the generation of embeddings for the question, image, and *KB*. These are refered in Fig. 3 as f_Q, f_I and f_{KB}, respectively (yellow blocks). Then, the Image Attention Module (green block) combines the image and question embeddings to create f_{IQ} and an attended image feature vector f_I^α. Next, the *KB* Attention Module (blue) takes f_I^α, the question embedding f_Q, and the *KB* embedding f_{KB} to generate an attention map α_{KB} and an attended *KB* feature f_{KB}^α over the *KB*. Afterwards, the Answering Module (orange block) combines f_Q, the attended image feature f_I^α and the attended *KB* feature f_{KB}^α to predict an answer.

To generate explanations, we first create an answer embedding f_A. Afterwards, the Visual Explanations Module (purple block) combines the image feature f_I, image-question feature f_{IQ}, and answer embedding f_A to output the visual explanation and create an attended image feature f_{Iexp}^α. Finally, a Textual Explanations Module (light orange block) takes f_{Iexp}^α, f_Q, f_A, and the attended *KB* feature f_{KB}^α to generate the textual explanation. Similar to [22], we use our pre-trained answering model and freeze its weights when training the explanations modules.

Our answering module builds upon a modified version of the work of Multimodal Compact Bilinear Pooling (MCB) [7]. However, by replacing all MCB layers with fully connected layers followed by element-wise operators, we boost speed and decrease memory usage with a low impact on performance. The architecture of our baseline explanation module is based on the PJ-X model [22], but we replace their answering module with our best answering model with *KB* (KB 12 K). The *KB* module is based on the idea of Key-Value Memory Networks used for reading comprehension [18]. Next, we provide more details behind the main modules that compose our model. These can be seen in Fig. 4.

3.2.1. Input representations

We create embeddings for the image, question, *KB*, and answer. For the image, we extract f_I from layer *res5c* of the pre-trained ResNet-152 [11]. This feature vector has shape 14 × 14 × 2048:

$$f_I(I) = ResNet152_{res5c}(I) \quad (1)$$

For the question, we initially embed the words using Glove [23]. Afterwards, we process the resulting word vectors using two *LSTM* layers. This process generates a 2048-dimensional feature-vector f_Q by concatenating the output of the first and second *LSTM*:

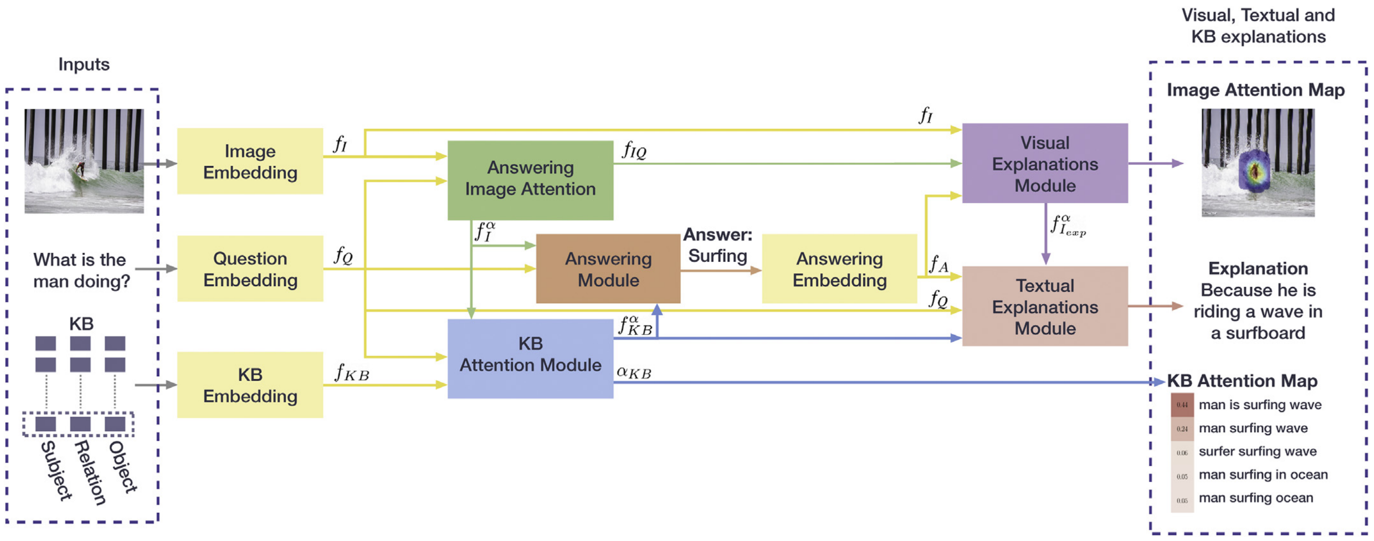


Fig. 3. Our model architecture (best viewed in color). Image, question, and *KB* are combined using several intermediate representations. During training, we train image and *KB* attention as auxiliary tasks to guide the learning of semantically meaningful intermediate representations. At test time, besides the answer, the model is able to provide a textual explanation as well as visual and *KB* attention maps.

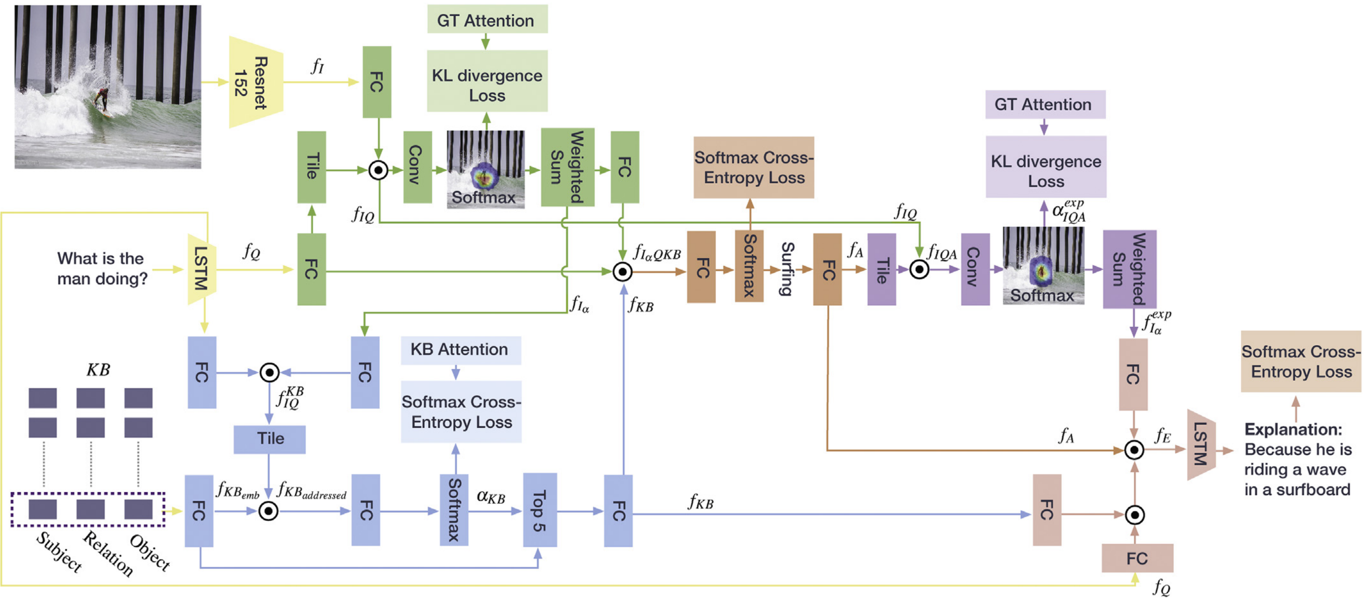


Fig. 4. Detailed model architecture. Image, Question, *KB* and, Answer embedding modules in yellow. Answering Image Attention module in green. *KB* Attention module in blue. Answering module in orange. Visual Explanations module in purple. Textual Explanations module in peach.

$$\begin{aligned} f_{Q1}(Q) &= LSTM_1([Glove(Q); FC_1(Q)]) \\ f_Q(Q, f_{Q1}) &= [LSTM_2(f_{Q1}); f_{Q1}] \end{aligned} \tag{2}$$

For the *KB*, the element in each triplet is embedded with Glove [23] using a 300-D vector. These vectors are then concatenated to create a 900-D vector for each triplet. Next, each triplet is passed through a shared fully connected (*FC*) layer with kb_s units. Given the size of the *KB*, we picked an embedding size $kb_s = 1024$ to reduce memory usage:

$$f_{KB_{emb}}(KB) = FC_2(Glove(KB)) \tag{3}$$

For the answer, we initially use a one-hot encoding corresponding to 3000-D (most frequent answers in the dataset). Afterwards, we embed this one-hot vector to a 2048-dimensional vector f_A. Specifically, this answer embedding consists of two fully connected layers with a tanh activation function after the first layer:

$$f_A\left(\hat{A}\right) = FC_4\left(FC_3\hat{A}\right) \tag{4}$$

3.2.2. Answering image attention module

Using the initial embeddings f_I and f_Q, we combine the question and image using fully connected layers followed by an element-wise multiplication (⊙) to generate an image-question feature f_{IQ}. To do so, we first tile the f_Q embedding 14 × 14 times to match the dimensions of f_I. This operation is done to multiply each of the 14 × 14 image region features by the question feature. After the element-wise multiplication, we apply signed square root ($\sqrt[\pm]{\ }$) and L2 normalization:

$$f_{IQ}(f_Q, f_I) = L2\left(\sqrt[\pm]{Tile(FC_5(f_Q), 14 \times 14) \odot (FC_6(f_I))}\right) \tag{5}$$

We generate the image attention by applying two convolutional layers (*Conv*) to f_{IQ}. We use a *ReLU* activation for the first convolution and a Softmax activation for the second. The final convolutional layer generates a feature vector of size 14 × 14 × 2 that contains two attention maps (region-level and object-level) [37].

$$\alpha_I(f_{IQ}) = Softmax(Conv_2(Conv_1(f_{IQ}))) \tag{6}$$

We supervise each attention map using the corresponding region or object visual groundings α_{GT} with a Kullback–Leibler (KL) Divergence Loss.

$$AttLoss_{ans}(\alpha_{GT}, \alpha_I) = KL(\alpha_{GT}, \alpha_I) \tag{7}$$

We use the attention maps to create Soft-Attentions [7] over f_I. We concatenate these features and use a fully connected layer to generate the final image feature f_I^α:

$$f_I^\alpha(f_I, \alpha_I) = FC_7([SoftAtt(f_I, \alpha_{I_1}); \; SoftAtt(f_I, \alpha_{I_2})]) \tag{8}$$

3.2.3. KB attention module

We incorporate the ability to consume and point out relevant information from the *KB*. The objective of this module is to generate a feature vector that contains only the relevant information from the *KB* for a given image-question pair. We achieve this by creating an attention map over the *KB* triplets. To reduce noise from unrelated triplets, the attention map is used to pick a subset of only the *k* most informative relations (where $k = 5$) for the given image-question pair and generate the final vector containing a reduced representation of the selected triplets. We combine information from the question and the image to generate the *KB* attention map, and supervise it with our labels to guide the model to extract relevant information.

We use the initial question embedding f_Q and the attended image feature f_I^α to address and identify relevant *KB* triplets using an attention mechanism. Using fully connected layers, we downsize the dimensions of these two vectors to kb_s, and then fuse them via element-wise multiplication followed by signed square root and L2 normalization to create f_{IQ}^{kb}:

$$f_{IQ}^{KB}(f_I^\alpha, f_Q) = L2\left(\sqrt[\pm]{FC_8(f_I^\alpha) \odot FC_9(f_Q)}\right) \tag{9}$$

We then combine f_{IQ}^{KB} with each triplet embedding of the *KB*. We tile the vector |*KB*| times, and apply an element-wise multiplication followed by Signed Square Root and L2 Normalization:

$$f_{KB}^{addressed}\left(f_{IQ}^{KB}, f_{KB}\right) = L2\left(\sqrt[\pm]{Tile\left(f_{IQ}^{KB}, |KB|\right) \odot f_{KB}}\right) \tag{10}$$

We create an attention map over the *KB* using two fully connected layers to create an embedding of size *KB* with a final softmax activation:

$$\alpha_{KB}\left(f_{KB}^{addressed}\right) = Softmax\left(FC_{11}\left(FC_{10}\left(f_{KB}^{addressed}\right)\right)\right) \tag{11}$$

The attention map represents the probability distribution over the *KB*, where triplets that are closely aligned with f_{IQ}^{KB} will have a high score. We supervise this attention with the collected *KB* labels using a softmax cross-entropy loss. Since each image-question pair can have more than one label, we randomly sample one (l) for each example on the batch:

$$Loss_{KB}(\alpha_{KB}, l) = -\sum_{i=1}^{|KB|} l_i \cdot \log\left(\alpha_{KB_i}\right) \tag{12}$$

Once we have the *KB* attention map, we use it to select the top $k = 5$ triplets with highest scores:

$$top\_k_{indices}(\alpha_{KB}, k) = argmax(\alpha_{KB}, k) \tag{13}$$

$$f_{KB_k}\left(f_{KB}, \mathrm{top}\_k_{indices}\right) = f_{KB}\big|_{top\_k_{indices}} \tag{14}$$

$$\alpha_{KB_k}\left(\alpha_{KB}, \mathrm{top}\_k_{indices}\right) = \alpha_{KB}\big|_{top\_k_{indices}} \tag{15}$$

We concatenate the top-k triplets with their corresponding attention coefficients, and generate a final embedding using two fully connected layers of size 2048 with a *tanh* activation:

$$f_{KB}^{\alpha}\left(f_{KB_k}, \alpha_{KB_k}\right) = FC_{13}\left(FC_{12}\left(\left[f_{KB_k}; \alpha_{KB_k}\right]\right)\right) \tag{16}$$

This vector contains all the information from the top k selected triplets.

3.2.4. Answering module

To predict the answer, we proceed to combine the question, image, and *KB* information source. Specifically, the Answering Module combines: the original question feature vector f_Q, the attended image f_I^{α}, and the attended *KB* f_{KB}^{α}. First, we pass f_Q through a fully connected layer, and fuse all the features with an element-wise product to create f_{QIKB}, which is the final feature that contains the information from our three sources:

$$f_{QIKB}\left(f_Q, f_I^{\alpha}, f_{KB}^{\alpha}\right) = L2\left(\sqrt[\pm]{FC_{14}\left(f_Q\right) \odot f_{KB}^{\alpha} \odot f_I^{\alpha}}\right) \tag{17}$$

We process f_{QIKB} using a fully connected layer with a Softmax activation. This outputs an L dimensional feature vector that represents the probability of each answer for the classification task. Similar to prior work, we set the answer space to the most frequent $L = 3000$ answers in the VQA dataset. The predicted answer is the one with the highest score in the L-dimensional embedding:

$$A\left(f_{IQKB}\right) = Argmax\left(Softmax\left(FC_{15}\left(f_{IQKB}\right)\right)\right) \tag{18}$$

3.2.5. Visual explanations module

Once we have the predicted answer, we generate the visual explanation. The Image-Question feature vector f_{IQ} from Section 3.2.2 is embedded using a fully connected layer, and then combined with f_A (Section 3.2.1) through an element-wise multiplication, followed by a signed square root and L2 normalization:

$$f_{IQA}\left(f_A, f_{IQ}\right) = L2\left(\sqrt[\pm]{Tile\left(f_A\right) \odot FC_{16}\left(f_{IQ}\right)}\right) \tag{19}$$

As in the answering module, we use f_{IQA} to create image attention maps used for explanations. Two attention maps (object-level and region-level [37]) are created using two convolutional layers, followed by a softmax activation:

$$\alpha_{I_{exp}}\left(f_{IQA}\right) = Softmax\left(Conv_4\left(Conv_3\left(f_{IQA}\right)\right)\right) \tag{20}$$

Following the process described in the answering module (Section 3.2.4), we generate the final image feature $f_{I_{\alpha}}^{exp}$ with Soft-Attention:

$$\begin{aligned} f_{I_{exp}}^{\alpha}\left(f_I, \alpha_{I_{exp}}\right) = \\ FC_{17}\left(\left[SoftAtt\left(f_I, \alpha_{I_{exp1}}\right);\ SoftAtt\left(f_I, \alpha_{I_{exp2}}\right)\right]\right)\right) \end{aligned} \tag{21}$$

During training, we supervise the image attention to build the explanation. We use the same method used to supervise the image attention to predict the answer (Section 3.2.2). Specifically, we use a KL-divergence loss to supervise each attention map with the corresponding labels (region or object attention labels).

$$AttLoss_{exp}\left(\alpha_{GT}, \alpha_{I_{exp}}\right) = KL\left(\alpha_{GT}, \alpha_{I_{exp}}\right) \tag{22}$$

3.2.6. Textual explanations module

Finally, we generate the textual explanation by combining the internal representation of different components of the model. We incorporate the *KB* information using the feature vector f_{KB}^{α}, which is fused with the attended image feature $f_{I_{\alpha}}^{exp}$, the original question feature f_Q, and the answer embedding f_A. To do this, we use fully connected layers and element-wise products to create f_E:

$$\begin{aligned} f_E\left(f_{I_{\alpha}}^{exp}, f_Q, f_{KB}^{\alpha}, f_A\right) = \\ L2\left(\sqrt[\pm]{FC_{18}\left(f_{I_{\alpha}}^{exp}\right) \odot FC_{19}\left(f_Q\right) \odot FC_{20}\left(f_{KB}^{\alpha}\right) \odot f_A}\right) \end{aligned} \tag{23}$$

f_E is the explanation feature that generates the textual explanation using an LSTM decoder to generate a sequence of words. Each generated word is conditioned on the previous one (w_{t-1}) and the hidden state h_t of the recurrent neural network:

$$h_t\left(f_E, w_{t-1}, h_{t-1}\right) = LSTM\left(f_E, w_{t-1}, h_{t-1}\right) \tag{24}$$

$$w_t(h_t) = Softmax(FC_{21}(h_t)) \tag{25}$$

The generated sentences are determined by the information that comes from the question, image, *KB*, and the predicted answer. During training, we supervise the generated explanations with a softmax cross-entropy loss. To do so, we use the dataset presented in [22].

4. Experiments & results

In this section, we present the experiments and results that validate our model. We evaluate the performance of our model in the following tasks: answering performance, visual explanations, and textual explanations. To evaluate the impact of our contributions, we analyze the effects of incorporating our external knowledge base *KB* to the model, and the impact of using visual attention supervision for answering and explaining. Our framework is able to generate a model that has a better semantic understanding of the question-image pair. We demonstrate this through a set of qualitative and quantitative experiments presented in this section.

4.1. Evaluation metrics

To evaluate our answering performance, we use the accuracy metric from [2]. Answers are considered 100% accurate if three or more annotators gave the same answer:

$$\text{Accuracy}(\boldsymbol{ans}) = \min\left\{\frac{\#\text{humans that said}\,\boldsymbol{ans}}{3}1\right\} \tag{26}$$

We evaluate the results of our answering model on the VQA dataset, using its test-dev and test-std splits, as well as the EvalAI evaluation server [33]. For the Explanation module, we evaluate visual explanations using the Rank-Correlation metric [27]. Finally, we evaluate the quality of textual explanations using the standard scores of BLEU-4 [21], METEOR [4], ROUGE [16], CIDEr [30], and SPICE [1].

4.2. KB construction

We evaluate three criteria for building our *KB*:

(a) **Most frequent:** From our collected triplets, we pick the $N = 12000$ (12 K) triplets that appear most frequently as labels. However, a *KB* of this size is both memory and computationally expensive. We develop two other approaches to mitigate this issue: (b) clustering and (c) pre-filtering.

(b) **Clustering:** Manually inspecting the *KB*, we discover that it contains redundant triplets. As an example, (man, fly, kite) is similar to (person, fly, kite) and to (man, flying, kite). In consequence, to avoid redundancies, we include an step to cluster similar rules in the *KB*, leaving as part of the *KB* only the center of each cluster. Specifically, we first obtain a 900-dimensional feature vector by concatenating the Glove embedding of the elements in each triplet. Then, we obtain N_c clusters using the K-Means algorithm [12]. To represent each cluster, we pick the average of its feature vector triplets. We generate our *KB* of size $N_c = 3$ K by clustering the 12 K *KB* from (a), and for $N_c = 12$ K by clustering a *KB* of size 36 K (most-frequent).

(c) **Question-image pre-filtering:** A potential drawback of using a clustering approach is that triplets encoding different semantic information might end up in the same cluster. To avoid this, we also try-out a pre-filtering approach. Specifically, for each question-image pair we apply a pre-filtering step to select a tailored subset of N relevant relations from the *KB*. First, we take all the collected triplets (over 120 K) and remove those that appeared as label less than 4 times, leaving 36 K relations. Since we do not have annotations for regions, objects or relationships in the VQA dataset, we do not know what are the relevant objects and regions in each image (only VG has that information). Hence, we use an object detector (YoloV3 [25]) to detect objects within each image. We combine the detected object names along with nouns and verbs from the question to select from the *KB* a reduced subset of triplets for each question-image pair. To select these triples, we fist sort all triplets in descending order according to the number of words in common with the nouns, verbs, and detected object names. We then select the top N relations from the sorted list. In our experiments, we test using $N = 1000$ and $N = 100$.

4.3. Question answering results

4.3.1. Effect of image attention supervision on answers

We demonstrated the effects of image attention supervision for the answering module in our preliminary work [37], where we obtained an attention map that had a better correlation with human attention without hurting performance. Using our visual attention labels, we improved the Ranked Correlation of three state-of-the-art models, MCB [7], MFB [35], MFH [36], by 0.30, 0.14, and 0.13 on the VQA-HAT dataset, and 0.14, 0.04, 0.03 on the VQA-X dataset, respectively. Furthermore, we also obtained a small boost of 0.14% accuracy with respect to the state-of-the-art model (MFH) in the VQA test-std split. These results can be seen in Table 1.

Table 1
Evaluation of image attention supervision on VQA models on rank correlation and answer prediction accuracy. The reported accuracies are evaluated using the VQA-2.0 test-standard set [37,33]. Results in bold are the best in their category.

Model	Rank correlation		Accuracy %
	VQA-HAT	VQA-X	VQA-2.0
Human [6]	0.623	–	80.62
PJ-X [22]	0.396	0.342	–
MCB [7]	0.276	0.261	62.27
Att-MCB, $\alpha = 1$ (ours)	**0.580**	**0.396**	60.51
Att-MCB (ours)	0.517	0.375	62.24
MFB [35]	0.276	0.299	65.22
Att-MFB (ours)	0.416	0.335	65.36
MFH [36]	0.354	0.350	66.17
Att-MFH (ours)	0.483	0.376	**66.31**

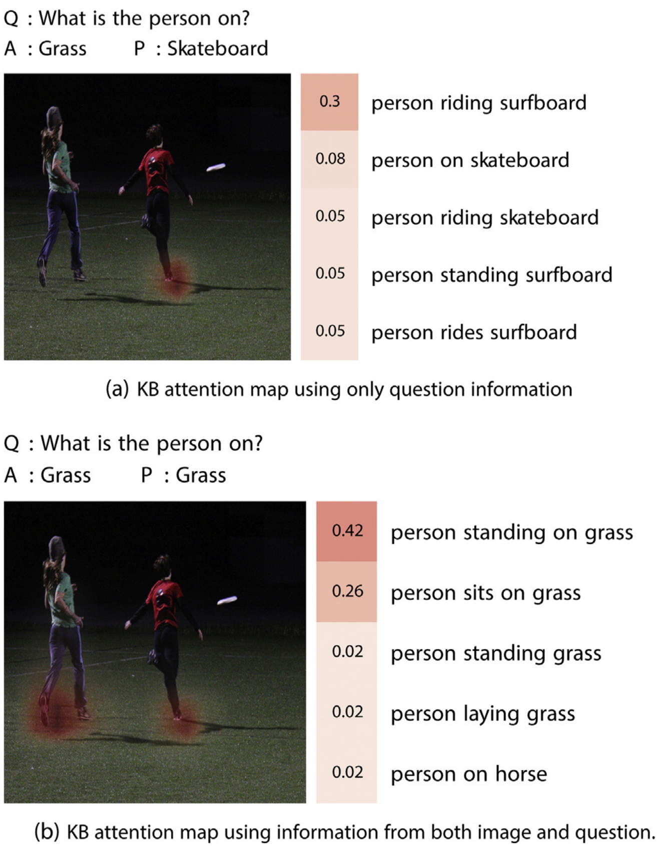
4.3.2. Effect of KB on answers

As we mentioned in Section 4.2, we build three types of *KB*: (a) The full size *KB* that is based on the 12 K most popular triplets, (b) Using clustering to merge triplets, here we test with sizes: $|KB| = 12$ K and $|KB| = 3$ K, and (c) Pre-filtering the *KB* using the content of each image-question pair, here we test with sizes $|KB| = 1$ K and $|KB| = 100$. In this section, we test our model using all these variations of the *KB*. Furthermore, we present results for the following cases: (i) A model that only uses the question to address the *KB*, (ii) A model that only uses the image to address the *KB*, and (iii) A model without *KB* supervision. These last three models are all trained using the *KB* without clustering and size $|KB| = 12$ K.

Table 2 shows the result of running our model on the VQA 2.0 dataset using each type of *KB*. In Table 2 the baseline corresponds to the modified MCB model [7], where MCB layers were replaced with fully connected layers followed by element-wise operators. With respect to this baseline, when incorporating the *KB*, our best *KB* model improves accuracy by more than 1% on *test-dev*, and almost 1% on *test-std* on the test-std set. The best results are obtained when using the largest *KB*; however, there is still some improvement when using smaller *KB* using our (b) and (c) approaches. Results also show that the best performance is achieved when addressing the *KB* using both the question and image information.

Table 2
Global accuracies for VQA models with *KB* and visual grounding. The reported accuracies are evaluated using the VQA-2.0 test-dev and test-standard sets. [33].

Model	Accuracy	
	Test-dev	Test-std
Baseline (not using *KB*)	61.88	62.02
KB 100 prefilter	61.91	–
KB 1 K prefilter	62.09	62.24
KB 3 K–12 K clustering	62.16	–
KB 5 K–12 K clustering	62.07	–
KB 12 K–31 K clustering	62.81	62.94
KB 12 K only I	62.63	–
KB 12 K only Q	62.54	–
KB 12 K (unsupervised Att)	62.73	–
KB 12 K + I + Q	**62.90**	**62.96**


Q : What is the person on?
A : Grass P : Skateboard
0.3 person riding surfboard
0.08 person on skateboard
0.05 person riding skateboard
0.05 person standing surfboard
0.05 person rides surfboard
Q : What is the person on?
A : Grass P : Grass
0.42 person standing on grass
0.26 person sits on grass
0.02 person standing grass
0.02 person laying grass
0.02 person on horse

(a) KB attention map using only question information

(b) KB attention map using information from both image and question.

Fig. 5. A: Ground truth answer, P: Predicted Answer. Left: input image including heatmap highlighting regions with highest attention. Right: Triplets from *KB* with highest attention weights. (a) *KB* attention map using only question information. In this case, the model predicts a wrong answer. (b) *KB* attention map using information from both image and question. Using key visual cues from the image, the model is able to attend relevant triplets from the *KB*. As a consequence, the model predicts the correct answer.

A distinguishing feature of our model is that we address the *KB* using information from the question and image inputs. Previous work has only used image textual attributes [15] to address the *KB*, or used image fact annotations directly as their *KB* [20,31]. However, in many cases information from the input image can play an important role to filtered out irrelevant hypothesis. As an example, consider the question "what is the man holding?". In this case, without information from the image, the system would attend from the *KB* every triplet related to "a man holding *anything*" instead of filtering out useless information. Fig. 5 shows an example where our model provides a wrong answer when addressing the *KB* using only information from the input question, but it is able to provide a correct answer when it also uses the image to address the *KB*.

Fig. 6 contains positive and negative qualitative examples of our model. The *KB* attention vector (to the right of each example), shows the top-5 highest scoring triplets along with their attention coefficients. Positive examples are cases where the model answers, explains and extracts information from the *KB* correctly; negative examples are cases where one of them fails.

In general, we notice that the answer provided by the model is closer to the correct answer when using the model with *KB*. We can see how the *KB* vector supports the answers with common sense facts and improves the model interpretability. The third example in Fig. 6a illustrate how the *KB* provides previous knowledge and contextual information to facilitate the answering process. In the second example, the question "what sport is this?" with its corresponding image will activate the weights related to tennis. In some cases, the answer is contained explicitly in one of the attended triplets of the *KB* (first and second positive example in Fig. 6a). An interesting idea could be to replace the prediction from the answering module with the information contained in the *KB* triplet following the approach in [20], where the answer is extracted from the most attended triplet.

Sometimes the *KB* attends irrelevant or incorrect triplets and might lead to incorrect predictions. Irrelevant triplets might be selected because the image attention module locates irrelevant regions of the input image, as shown in the third negative example in Fig. 6b. In the example, the *KB* attends triplets indicating that the wall has tiles, which

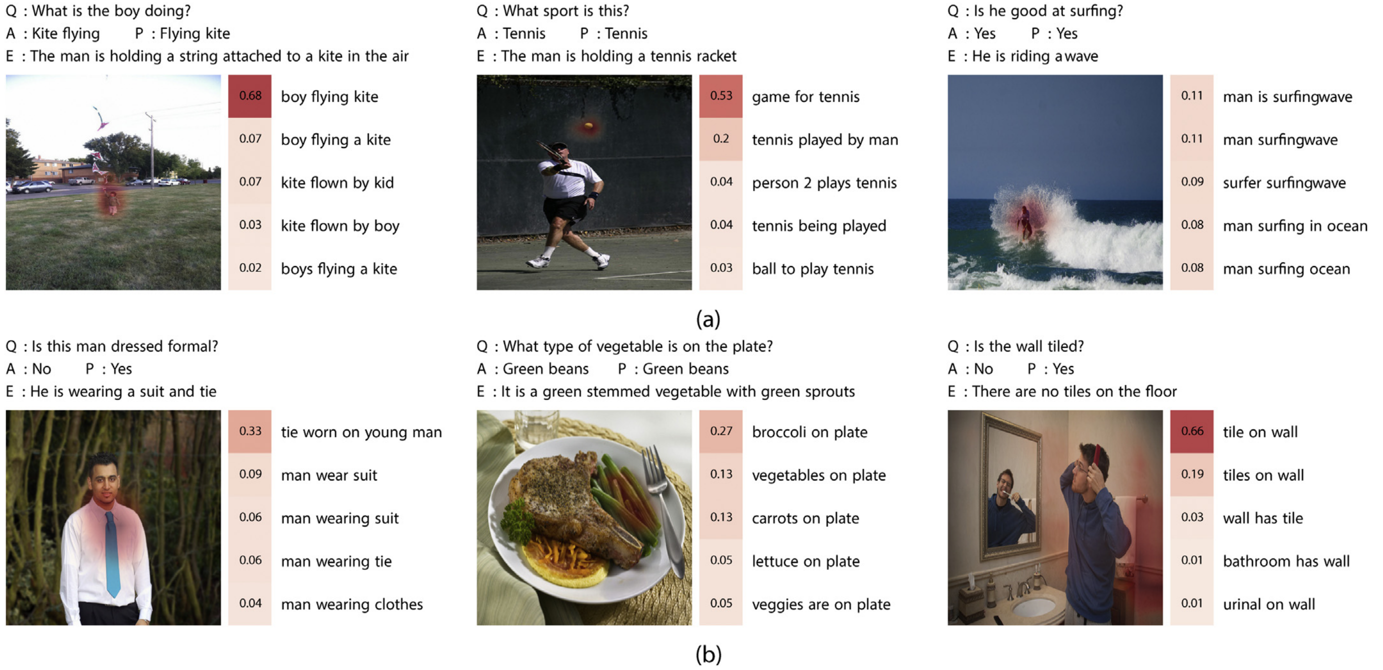


Fig. 6. Qualitative examples. a) Cases where our model has correct predictions, b) Cases where our model fails. We show our model predictions for the answer, attention over *KB*, visual explanations, and textual explanations (Q: Question, A: Ground truth answer, P: Predicted Answer, E Predicted Textual Explanation).

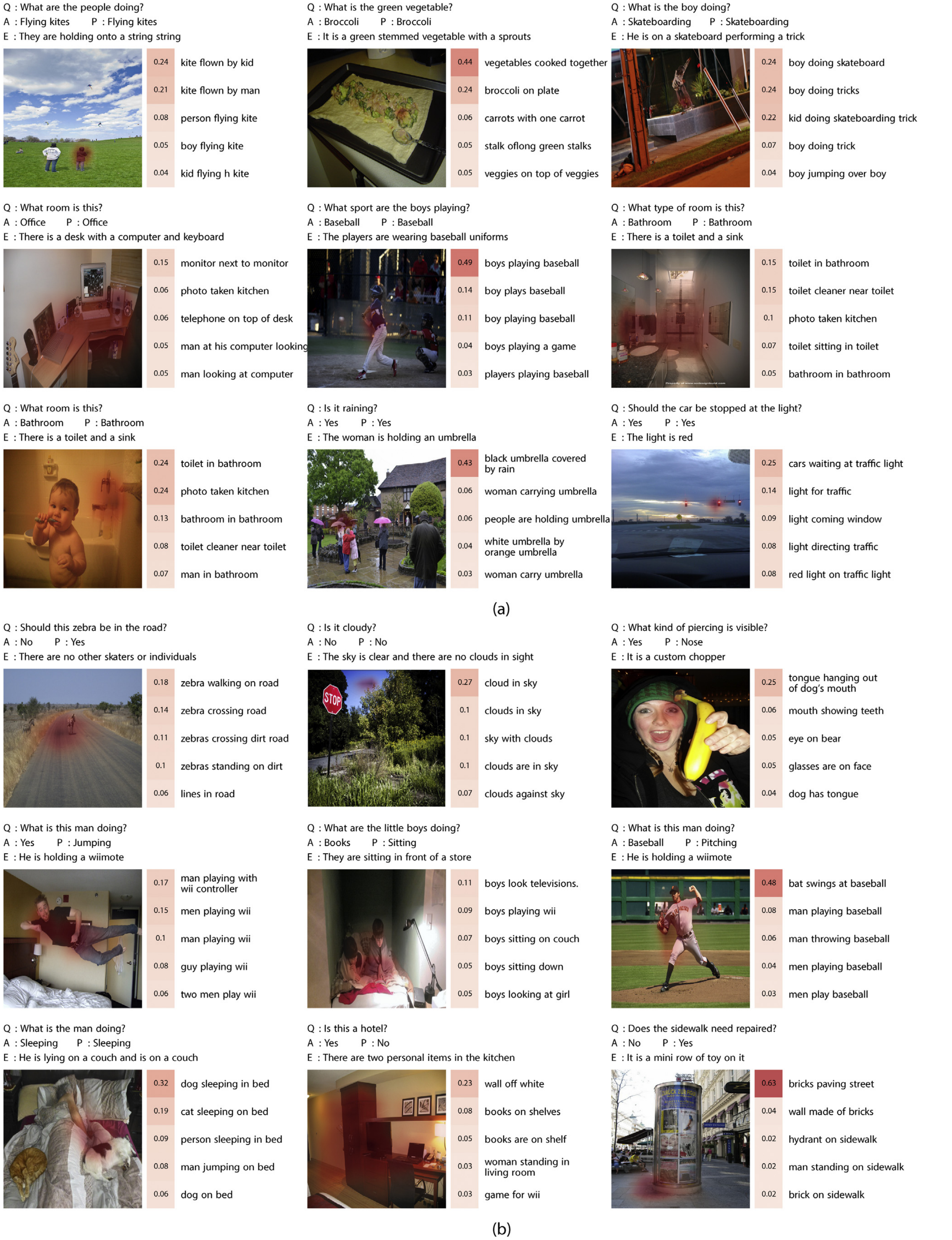

Q : What are the people doing?
A : Flying kites P : Flying kites
E : They are holding onto a string string
0.24 kite flown by kid
0.21 kite flown by man
0.08 person flying kite
0.05 boy flying kite
0.04 kid flying h kite
Q : What is the green vegetable?
A : Broccoli P : Broccoli
E : It is a green stemmed vegetable with a sprouts
0.44 vegetables cooked together
0.24 broccoli on plate
0.06 carrots with one carrot
0.05 stalk oflong green stalks
0.05 veggies on top of veggies
Q : What is the boy doing?
A : Skateboarding P : Skateboarding
E : He is on a skateboard performing a trick
0.24 boy doing skateboard
0.24 boy doing tricks
0.22 kid doing skateboarding trick
0.07 boy doing trick
0.04 boy jumping over boy
Q : What room is this?
A : Office P : Office
E : There is a desk with a computer and keyboard
0.15 monitor next to monitor
0.06 photo taken kitchen
0.06 telephone on top of desk
0.05 man at his computer looking
0.05 man looking at computer
Q : What sport are the boys playing?
A : Baseball P : Baseball
E : The players are wearing baseball uniforms
0.49 boys playing baseball
0.14 boy plays baseball
0.11 boy playing baseball
0.04 boys playing a game
0.03 players playing baseball
Q : What type of room is this?
A : Bathroom P : Bathroom
E : There is a toilet and a sink
0.15 toilet in bathroom
0.15 toilet cleaner near toilet
0.1 photo taken kitchen
0.07 toilet sitting in toilet
0.05 bathroom in bathroom
Q : What room is this?
A : Bathroom P : Bathroom
E : There is a toilet and a sink
0.24 toilet in bathroom
0.24 photo taken kitchen
0.13 bathroom in bathroom
0.08 toilet cleaner near toilet
0.07 man in bathroom
Q : Is it raining?
A : Yes P : Yes
E : The woman is holding an umbrella
0.43 black umbrella covered by rain
0.06 woman carrying umbrella
0.06 people are holding umbrella
0.04 white umbrella by orange umbrella
0.03 woman carry umbrella
Q : Should the car be stopped at the light?
A : Yes P : Yes
E : The light is red
0.25 cars waiting at traffic light
0.14 light for traffic
0.09 light coming window
0.08 light directing traffic
0.08 red light on traffic light
(a)
Q : Should this zebra be in the road?
A : No P : Yes
E : There are no other skaters or individuals
0.18 zebra walking on road
0.14 zebra crossing road
0.11 zebras crossing dirt road
0.1 zebras standing on dirt
0.06 lines in road
Q : Is it cloudy?
A : No P : No
E : The sky is clear and there are no clouds in sight
0.27 cloud in sky
0.1 clouds in sky
0.1 sky with clouds
0.1 clouds are in sky
0.07 clouds against sky
Q : What kind of piercing is visible?
A : Yes P : Nose
E : It is a custom chopper
0.25 tongue hanging out of dog's mouth
0.06 mouth showing teeth
0.05 eye on bear
0.05 glasses are on face
0.04 dog has tongue
Q : What is this man doing?
A : Yes P : Jumping
E : He is holding a wiimote
0.17 man playing with wii controller
0.15 men playing wii
0.1 man playing wii
0.08 guy playing wii
0.06 two men play wii
Q : What are the little boys doing?
A : Books P : Sitting
E : They are sitting in front of a store
0.11 boys look televisions.
0.09 boys playing wii
0.07 boys sitting on couch
0.05 boys sitting down
0.05 boys looking at girl
Q : What is this man doing?
A : Baseball P : Pitching
E : He is holding a wiimote
0.48 bat swings at baseball
0.08 man playing baseball
0.06 man throwing baseball
0.04 men playing baseball
0.03 men play baseball
Q : What is the man doing?
A : Sleeping P : Sleeping
E : He is lying on a couch and is on a couch
0.32 dog sleeping in bed
0.19 cat sleeping on bed
0.09 person sleeping in bed
0.08 man jumping on bed
0.06 dog on bed
Q : Is this a hotel?
A : Yes P : No
E : There are two personal items in the kitchen
0.23 wall off white
0.08 books on shelves
0.05 books are on shelf
0.03 woman standing in living room
0.03 game for wii
Q : Does the sidewalk need repaired?
A : No P : Yes
E : It is a mini row of toy on it
0.63 bricks paving street
0.04 wall made of bricks
0.02 hydrant on sidewalk
0.02 man standing on sidewalk
0.02 brick on sidewalk
(b)

Fig. 7. Additional qualitative examples. a) Cases where our model has correct predictions, b) Cases where our model fails. (Q: Question, A: Ground truth answer, P: Predicted Answer, E Predicted Explanation).

Table 3
Answer accuracy by category on validation set. We only present categories with a variation of over 1% with respect to our baseline.

Question type	Baseline	I + Q + KB 12 K	Gain
Why is the	17.67	21.89	4.22
What is the person	56.23	58.81	2.58
What is the man	53.86	56.43	2.57
Is it	84.92	87.49	2.57
Can you	71.88	74.07	2.19
What is this	58.17	60.33	2.16
Is he	77.03	79.13	2.10
What is in the	47.72	49.34	1.62
How	27.73	29.20	1.47
Is this an	76.51	77.94	1.43
What number is	8.17	9.54	1.37
What is	42.32	43.69	1.37
Is the	74.64	76.00	1.36
What type of	53.03	54.36	1.33
Is the man	73.56	74.85	1.29
What are	52.76	53.95	1.19
What kind of	53.56	54.72	1.16
What color	65.78	66.95	1.17
What is the	46.32	47.40	1.08
Is this a	77.29	78.34	1.05
How many people are in	44.66	45.71	1.05
Which	42.82	43.86	1.04
What color are the	69.57	70.58	1.01
What sport is	86.57	87.57	1.00
Could	78.66	79.66	1.00
What time	24.04	23.01	−1.03
Has	74.81	72.69	−2.12

leads to an incorrect answer. We also notice overfitting and bias problems related to the training dataset. Fig. 7 provides more qualitative examples illustrating the performance of our best model.

The *KB* is a useful tool that provides insights when the model fails to provide the correct answer or explanation (5). For instance, for the second image in the Negative examples, the explanation describes a vegetable that sounds like broccoli, but the correct answer is green beans. This is odd since the image attention is focused on the green beans. However, looking closely at the *KB*, we see that the model is actually pointing to broccoli. There are also cases where the ground truth answer is ambiguous or wrong, as seen in the first negative example.

Further analysis shows that the *KB* helps to increase the performance of almost every category of questions in the VQA dataset [2] (53 out of 65 question types). In Table 3, we list the categories that present either a positive or negative accuracy variation of more than 1%. It is interesting to note that most categories in this table are questions that begin with'What'. Questions beginning with'What' usually have an answer that is a verb followed by a noun, or just a noun (E.g., Q: what is the man doing?, A: playing tennis).'What' questions usually benefit from information within the *KB*, since most relations in the *KB* are formatted as $\{Subj(Noun), Rel(Verb), Obj(Noun)\}$.

Table 5
Evaluation of Visual Explanations generated with and without ground-truth (GT) answer conditioning, compared against the VQA-X test set attention labels. Evaluated using Rank Correlation - higher is better, and Earth Mover's distance - lower is better. Our models compare favorably to the baselines. Results in bold are the best in their category.

Approach	Earth Movers (Lower is better)		Rank Correlation (Higher is better)	
	VQA-X	VQA-X GT	VQA-X	VQA-X GT
ME [22]	2.64	–	+0.3423	–
Baseline Explanations (BE)	2.998	2.989	+0.3465	+0.3467
BE + KB_{Ans}	2.963	2.950	+0.3468	+0.3456
BE + ImageSup	**2.413**	**2.404**	+0.3897	+0.3902
BE + KB_{Ans} + ImageSup	2.433	2.422	+0.3938	+0.3945
BE + KB_{Ans} + ImageSup + KB_{Exp}	2.429	2.413	**+0.4007**	**+0.4015**

Since the test-dev and test-std split of VQA are not public, we present the accuracy by category on the validation split from VQA. On the validation split, were our best *KB* model achieves an accuracy of 60.5, and the basline 59.57 (both models were trained without using the validation split for this study). The 12 categories in which the *KB* has a negative impact are related to questions that require knowledge about the properties of objects instead of relationships between them, which makes sense since we collected most of the *KB* triplets from the visual genome relationships [14] that depicts relations between objects.

4.4. Explanations results

We run several experiments to evaluate the impact of incorporating visual attention supervision and external information. Visual attention supervision is evaluated using the Earth Mover's Distance [26] and Rank Correlation [27] metrics (Table 5). Textual explanations are evaluated using the scores of BLEU-4 [21], METEOR [4], ROUGE [16], CIDEr [30], and SPICE [1] (Table 4). For Visual and Textual Explanations, each model is evaluated twice: first using the answer predicted by the model, and second using the ground truth answer. Results show the impact of incorporating the *KB* to the model, and the relevance of guiding the model attention in terms of building interpretable explanations.

Following, we describe the trained models:

(i) **Baseline explanations model (BE):** This model corresponds to our base answering module (without *KB*) and the explaining module from [22]. This model obtains similar results to the ones presented in [22], but there are small score differences in

Table 4
Evaluation of textual justifications generated with and without ground-truth (GT) answer conditioning. Evaluated using automatic metrics: BLEU-4, METEOR, ROUGE, CIDEr and SPICE. All in %. Our models compare favorably to the baseline. Results in bold are the best in their category.

Approach	GT-ans conditioning	VQA-X test set score				
		BLEU-4	METEOR	ROUGE	CIDEr	SPICE
ME [22]	Yes	19.8	18.6	44.0	73.4	15.4
Baseline Explanations (BE)	Yes	19.8	18.7	43.4	72.6	15.5
BE + KB_{Ans}	Yes	21.9	**19.6**	45.5	80.7	**16.5**
BE + ImageSup	Yes	21.5	19.0	44.7	76.8	16.2
BE + KB_{Ans} + ImageSup	Yes	22.1	19.1	45.2	78.6	15.7
BE + KB_{Ans} + ImageSup + KB_{Exp}	Yes	**22.4**	19.4	**45.7**	**81.1**	16.1
ME [22]	No	19.5	18.2	43.4	71.3	15.1
Baseline Explanations (BE)	No	19.2	18.3	42.8	69.7	15.0
BE + KB_{Ans}	No	20.9	**19.2**	44.6	76.5	**15.9**
BE + ImageSup	No	20.8	18.6	44.4	74.32	15.7
BE + KB_{Ans} + ImageSup	No	21.4	18.8	44.7	75.6	15.2
BE + KB_{Ans} + ImageSup + KB_{Exp}	No	**21.8**	19.1	**45.3**	**78.3**	15.6

the evaluation metrics. We believe that these differences are related to different tuning of parameters, the number of iterations used to train the answering module or random weight initialization.

(ii) **BE + $\mathbf{KB_{Ans}}$:** This model replaces the answering module in (i) with our KB 12 K answering model outperforming the baseline model on all textual evaluation metrics.

(iii) **BE + Image Supervision:** Here, we add the explanations image attention supervision to (i). The resulting model outperforms the baseline under all evaluation metrics and presents a higher rank correlation than the model in [22].

(iv) **BE + $\mathbf{KB_{Ans}}$ + Image Supervision:** This model combines (ii) and (iii): it includes the *KB* answering module and image attention supervision. This improves the textual explanation quality in (iii) and visual rank correlation in (ii).

(v) **BE + $\mathbf{KB_{Ans}}$ + Image Supervision + $\mathbf{KB_{Exp}}$:** This model incorporates the *KB* feature vector f_{KB}^{α} from the answering module directly into the explanations attention vector f_E. This model achieves the highest scores overall for both textual and visual explanations.

4.4.1. Effect of visual attention supervision on explanations

From the results shown in Tables 4 and 5, it can be seen that the experiments that add image attention supervision usually help increase both textual and visual explanations performance. Specifically, we can see that when adding image supervision, the resulting model presents a higher rank correlation than our baseline and the model in [22]. Also, we obtain higher scores under all evaluation metrics for textual explanations.

4.4.2. Effect of KB on explanations

The use of the *KB* affects both textual and visual explanations. Table 4 shows the effect of *KB* on textual explanation. Furthermore, Table 5 shows that adding the *KB* not only helps to improve the accuracy of the textual explanation, but also increases the correlation between the model's image attention and ground truth from VQA-X.

When adding the *KB* only to the answering module (experiments: Base Model + KB_{Ans} and Base Model + KB_{Ans} + Image Supervision), we attribute the increase of textual and visual explanations accuracy to having better internal representations in the early stages. These better representations help later modules that use intermediate feature vectors from earlier stages. On the other hand, when we incorporate the *KB* directly to our explanations module (experiment: BE + KB_{Ans} + ImageSup + KB_{Exp}), the information extracted from the *KB* is used directly to generate better explanations.

Fig. 6 shows qualitative results of explanations with our model. It shows that image attention is usually highly correlated to the question asked. Sometimes the model fails to explain a correctly predicted answer (bottom-right example in Fig. 6), which probably shows overfitting and bias problems related to the training set.

5. Conclusions

In this work, we focus on improving the interpretability of VQA models, while also producing a competitive model in terms of the accuracy of the predicted answer. In particular, we measure interpretability accessing the quality of explanations and attentions generated by our VQA model. We achieved state-of-the-art explanation performance by incorporating image attention supervision and introducing an effective mechanism that leverages an external *KB* to produce better answers and explanations. We provide evidence that the generated visual groundings (while answering or explaining) achieve a high correlation with respect to human-provided attention annotations, outperforming previous works by a large margin. Our *KB* and its labels can be mined automatically from scene graphs in Visual Genome, and the VQA dataset. Furthermore, our algorithm can attend to a small number of relevant facts among a large number of entries in the *KB*. These facts show that incorporating external information is a crucial step towards generating models with a more meaningful semantic representation. Future work will explore knowledge acquisition from other sources as well as mechanisms for improving the accuracy of the answers themselves.

Acknowledgements

This work was partially funded by Fondecyt Grant 1181739, Chile and Millennium Institute for Foundational Research on Data, Chile.

References

[1] P. Anderson, B. Fernando, M. Johnson, S. Gould, Spice: semantic propositional image caption evaluation, Proceedings of the European Conference on Computer Vision (ECCV), 2016.

[2] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C.L. Zitnick, D. Parikh, VQA: visual question answering, International Conference on Computer Vision (ICCV), 2015.

[3] D. Bahdanau, K. Cho, Y. Bengio, Neural Machine Translation by Jointly Learning to Align and Translate, 2015.

[4] S. Banerjee, A. Lavie, METEOR: an automatic metric for MT evaluation with improved correlation with human judgments, Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation And/Or Summarization, Association for Computational Linguistics, Ann Arbor, Michigan June 2005, pp. 65–72.

[5] S. Bird, E. Loper, NLTK: the natural language toolkit, Proceedings of the ACL Interactive Poster and Demonstration Sessions, Association for Computational Linguistics, Barcelona, Spain July 2004, pp. 214–217.

[6] A. Das, H. Agrawal, L. Zitnick, D. Parikh, D. Batra, Human attention in visual question answering: do humans and deep networks look at the same regions? Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Pages 932–937, Austin, Texas, Association for Computational Linguistics, Nov. 2016.

[7] A. Fukui, D.H. Park, D. Yang, A. Rohrbach, T. Darrell, M. Rohrbach, Multimodal compact bilinear pooling for visual question answering and visual grounding, Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Pages 457–468, Austin, Texas, Association for Computational Linguistics, Nov. 2016.

[8] C. Gan, Y. Li, H. Li, C. Sun, B. Gong, Vqs: linking segmentations to questions and answers for supervised attention in vqa and question-focused semantic segmentation, International Conference on Computer Vision (ICCV), 2017.

[9] B. Goodman, S. Flaxman, European union regulations on algorithmic decision-making and a “right to explanation”, AI Mag. 38 (3) (Oct. 2017) 50–57.

[10] Y. Goyal, T. Khot, A. Agrawal, D. Summers-Stay, D. Batra, D. Parikh, Making the v in vqa matter: Elevating the role of image understanding in visual question answering, International Journal of Computer Vision 127 (4) (Apr. 2019) 398414.

[11] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016, pp. 770–778.

[12] X. Jin, J. Han, K-Means Clustering, Springer US, Boston, MA, 2010 563–564.

[13] J. Kim, A. Rohrbach, T. Darrell, J. Canny, Z. Akata, Textual explanations for self-driving vehicles, in: V. Ferrari, M. Hebert, C. Sminchisescu, Y. Weiss (Eds.), Proceedings of the European Conference on Computer Vision (ECCV), Springer International Publishing, Cham 2018, pp. 577–593.

[14] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D.A. Shamma, M.S. Bernstein, L. Fei-Fei, Visual genome: connecting language and vision using crowdsourced dense image annotations, Int. J. Comput. Vis. 123 (1) (May 2017) 32–73.

[15] A. Kumar, O. Irsoy, P. Ondruska, M. Iyyer, J. Bradbury, I. Gulrajani, V. Zhong, R. Paulus, R. Socher, Ask me anything: dynamic memory networks for natural language processing, in: M.F. Balcan, K.Q. Weinberger (Eds.), Proceedings of The 33rd International Conference on Machine Learning, Volume 48 of Proceedings of Machine Learning Research, PMLR, New York, New York, USA 20–22 Jun 2016, pp. 1378–1387.

[16] C.-Y. Lin, ROUGE: A package for automatic evaluation of summaries, Workshop on Text Summarization Branches Out, Post-Conference Workshop of ACL 2004, Barcelona, Spain, Association for Computational Linguistics, Barcelona, Spain July 2004, pp. 74–81.

[17] J. Lu, J. Yang, D. Batra, D. Parikh, Hierarchical question-image co-attention for visual question answering, Advances In Neural Information Processing Systems 2016, pp. 289–297.

[18] A. Miller, A. Fisch, J. Dodge, A.-H. Karimi, A. Bordes, J. Weston, Key-value memory networks for directly reading documents, Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Austin, Texas Nov. 2016, pp. 1400–1409.

[19] G.A. Miller, Wordnet: A lexical database for english, Commun. ACM 38 (11) (Nov. 1995) 3941.

[20] M. Narasimhan, A.G. Schwing, Straight to the facts: Learning knowledge base retrieval for factual visual question answering, Proceedings of the European Conference on Computer Vision (ECCV) 2018, pp. 460–477.

[21] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, Bleu: a method for automatic evaluation of machine translation, Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ACL 02, Association for Computational Linguistics, USA 2002, pp. 311–318.

[22] D.H. Park, L.A. Hendricks, Z. Akata, A. Rohrbach, B. Schiele, T. Darrell, M. Rohrbach, Multimodal explanations: justifying decisions and pointing to the evidence, 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition 2018, pp. 8779–8788.

[23] J. Pennington, R. Socher, C. Manning, GloVe: global vectors for word representation, Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Doha, Qatar Oct. 2014, pp. 1532–1543.

[24] T. Qiao, J. Dong, D. Xu, Exploring human-like attention supervision in visual question answering, Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[25] J. Redmon, A. Farhadi, Yolov3: An Incremental Improvement, CoRR, 2018 abs/1804.02767.

[26] Y. Rubner, C. Tomasi, L.J. Guibas, A metric for distributions with applications to image databases, Sixth International Conference on Computer Vision (ICCV) Jan 1998, pp. 59–66.

[27] C. Spearman, The proof and measurement of association between two things, Am. J. Psychol. 15 (1904) 72–101.

[28] Z. Su, C. Zhu, Y. Dong, D. Cai, Y. Chen, J. Li, Learning visual knowledge memory networks for visual question answering, 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition 2018, pp. 7736–7745.

[29] D. Teney, P. Anderson, X. He, A.V.D. Hengel, Tips and tricks for visual question answering: learnings from the 2017 challenge, 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition 2018, pp. 4223–4232.

[30] R. Vedantam, C. Zitnick, D. Parikh, Cider: consensus-based image description evaluation, 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2015, pp. 4566–4575.

[31] P. Wang, Q. Wu, C. Shen, A. Dick, A. van den Hengel, Fvqa: fact-based visual question answering, IEEE Trans. Pattern Anal. Mach. Intell. 40 (2018) 2413–2427.

[32] Q. Wu, P. Wang, C. Shen, A. Dick, A. Van Den Hengel, Ask me anything: Free-form visual question answering based on knowledge from external sources, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016, pp. 4622–4630.

[33] D. Yadav, R. Jain, H. Agrawal, P. Chattopadhyay, T. Singh, A. Jain, S. Singh, S. Lee, D. Batra, Evalai: Towards Better Evaluation Systems for AI agents, CoRR, 2019 abs/1902.03570.

[34] Z. Yang, X. He, J. Gao, L. Deng, A. Smola, Stacked attention networks for image question answering, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2016, pp. 21–29.

[35] Z. Yu, J. Yu, J. Fan, D. Tao, Multi-modal factorized bilinear pooling with co-attention learning for visual question answering, International Conference on Computer Vision (ICCV) 2017, pp. 1839–1848.

[36] Z. Yu, J. Yu, C. Xiang, J. Fan, D. Tao, Beyond bilinear: generalized multimodal factorized high-order pooling for visual question answering, IEEE Trans. Neural Networks Learn. Syst. 29 (12) (2018) 5947–5959.

[37] Y. Zhang, J.C. Niebles, A. Soto, Interpretable visual question answering by visual grounding from attention supervision mining, 2019 IEEE Winter Conference on Applications of Computer Vision (WACV) 2019, pp. 349–357.