PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA

A KNOWLEDGE BASE APPROACH TO IMPROVE INTERPRETABILITY AND PERFORMANCE OF VISUAL QUESTION ANSWERING TASK USING DEEP LEARNING MODELS

FELIPE ANTONIO RIQUELME CALLEJAS

Thesis submitted to the Office of Research and Graduate Studies
in partial fulfillment of the requirements for the degree of
Master of Science in Engineering

Advisor:
ÁLVARO SOTO ARRIAZA

Santiago de Chile, Diciembre 2019

PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA

A KNOWLEDGE BASE APPROACH TO IMPROVE INTERPRETABILITY AND PERFORMANCE OF VISUAL QUESTION ANSWERING TASK USING DEEP LEARNING MODELS

FELIPE ANTONIO RIQUELME CALLEJAS

Members of the Committee:
ÁLVARO SOTO ARRIAZA
JUAN CARLOS NIEBLES
HANS LÖBEL DÍAZ
CRISTIÁN SANDOVAL MANDUJANO

Thesis submitted to the Office of Research and Graduate Studies
in partial fulfillment of the requirements for the degree of
Master of Science in Engineering

Santiago de Chile, Diciembre 2019

This work is dedicated to my beloved parents who encouraged and trusted me to accomplish my work. I'm specially grateful to my advisor, who taught me along the way.

ACKNOWLEDGEMENTS

We thank Ben Zahng, Juan Carlos Niebles, and Alfredo De Goyeneche for their good will and assistance with research.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

RESUMEN

Los modelos de Aprendizaje Profundo ó *Deep Learning* son vistos y tratados como cajas negras. Dada una entrada, estos generan una salida a modo de respuesta. Pero no se tiene mas que una noción vaga de lo que llevó al modelo a responder lo que respondió. Sin embargo, en muchas aplicaciones (aplicaciones bancarias, compañías de seguros, asistentes personales, etc) es deseable o incluso necesario saber que llevó al modelo a generar una determinada respuesta. En este trabajo nos enfocamos en el desafío llamado *Visual Question Answering* (VQA). Este consiste en lograr que un modelo responda preguntas basadas en imágenes que se le presentan. Logramos incorporar una nueva Base de Conocimiento o *knowledge base (KB)* que contiene relaciones entre objetos del mundo real, lo que ayuda a mejorar la interpretabilidad y el desempeño del modelo mediante la identificación y extracción de información relevante acorde a cada pregunta e imagen que se presenta. La extracción de información de la KB fue supervisada directamente para generar un mapa de atención usado por el modelo para identificar las relaciones relevantes a cada preguntae imágen. Se muestra cuantitativamente que las predicciones del modelo mejoran con la introducción de la KB. También mostramos cualitativamente la mejora en cuanto a interpretabilidad mediante la atención generada sobre las relaciones de la KB. Adicionalmente, mostramos cómo la KB ayuda a mejorar el desempeño en modelos de VQA que generan explicaciones. Los resultados obtenidos demuestran que el mecanismo de atención empleado en la KB ayuda mejorar la interpretabilidad del modelo. Y la información adicional extraida mejora la representación interna de eśte y por ende también el desempeño.

Keywords: atención, supervición, Base de Conocimiento, interpretabilidad, Aprendizaje Profundo.

ABSTRACT

Deep learning models are usually treated as black-boxes. Given an input, an output is generated without providing any insight about what led the model to reach its prediction. However, in many applications (Banking, Insurance, Personal Assistants, etc.), interpretability is highly desirable, if not required, feature. In this work, we focus on the Visual Question Answering (VQA) task, in which a model must answer a question based on an image. We introduce a Knowledge Base (KB) filled with real-world relationships between objects into our model. The information contained in this KB helps to generate better predictions, and improve interpretability by pointing out and retrieving information relevant to the question. We directly supervise this KB to generate an attention map and select the relevant relationships from the KB for each question-image pair. We quantitatively show how predictions improve with the KB introduction, we also qualitatively showcase the KB attention that helps improve interpretability. Additionally, we demonstrate how the KB can also help improve the quality of a VQA explanation model. The results obtained demonstrate the benefits of having a KB attention mechanism to improve the interpretability of the model, and how the external information allows the model to achieve better internal representations for the problem and therefore, better performance.

Keywords: Attention, Supervision, Knowledge Base, Interpretability, Deep Learning.

1. INTRODUCTION

Visual Question Answering (VQA) is a well-explored task in the computer vision community Fukui et al. [2014], Kumar et al. [2015], Su et al. [2018], Teney et al. [2017]. The task consists of answering a textual question formulated in natural language in regards to the contents of an image. In general, there is virtually an infinite number of questions that can be asked about a single image, and a single question can be asked about an uncountable number of images resulting in different answers. As a consequence, VQA is a challenging task that requires a semantic understanding of both natural language and visual elements. A suitable solution to the VQA problem needs to correctly parse the input question, being able to identify key structures and relations (verbs, nouns, etc.), as well as simultaneously understanding in-depth the contents of the input image. The model must be able to identify relevant regions, objects, and relations between them. Solving this task with a single model is very demanding, and even more without any previous knowledge of how the real world works.

Most current models, solve the VQA problem following a discriminative approach [Fukui et al., 2014, Lu et al., 2017, Teney et al., 2017]. These models pose VQA as a classification problem, where classes correspond to a set of the most common pre-defined candidate answers. The models also incorporate an attention mechanism that selects visual cues that will be used to answer each question. This attention mechanism is commonly applied using an unsupervised training process, where the so-called attention coefficients are considered as latent variables [Bahdanau et al., 2015]. Recent works have stressed the limitations of this attention scheme, showing that it leads to models encoding discriminative cues that cannot ground image attentions to the underlying semantics behind each question-answer pair [Das et al., 2016, Gan et al., 2017, Qiao et al., 2018, Zhang et al., 2018]. As a result, current VQA models lack of a suitable level of interpretability.

Interpretability is a highly desirable property because it provides a window to the internal representation of a model, and can be used to examine the predictions from the model. Furthermore, as AI-based systems start to operate in real-world applications, there is an increasing need to

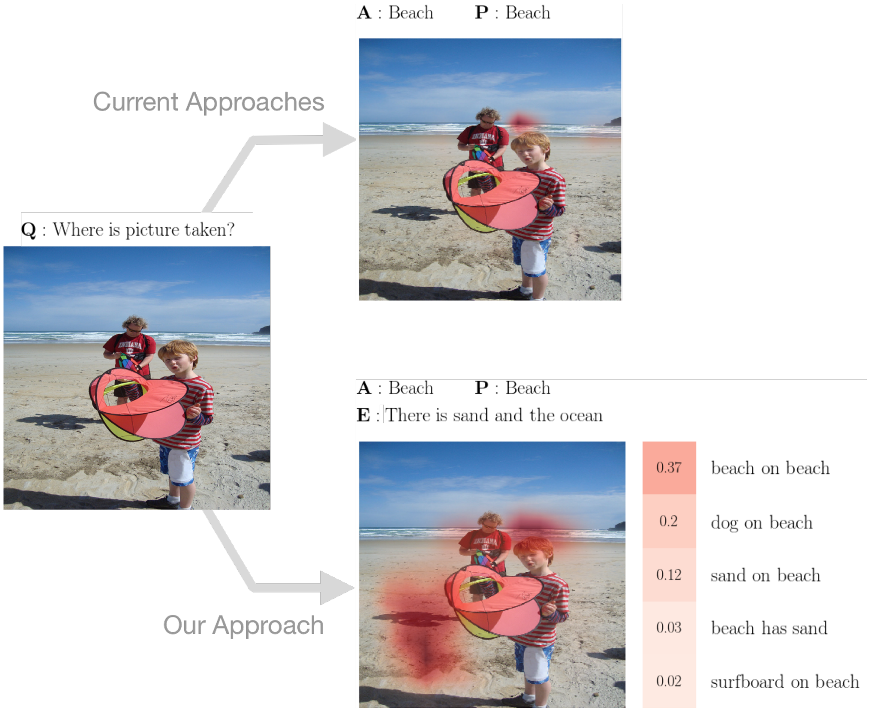


Figure 1.1. Current models that solve the VQA task lack a suitable mechanisms to attend relevant image regions or to retrieve relevant facts from an external knowledge base (bar with scores to the right of the image). Our proposed approach aims to solve those limitations. (**A**: Answer, **P**: Prediction, **E**: Explanation)

provide them with the ability to explain their decisions. This feature is starting to be required by legal regulations [Goodman & Flaxman, 2016] little by little. To quantify interpretability, recent VQA models have started to introduce new performance metrics oriented to assess the quality of the resulting visual groundings [Anderson et al., 2018, Fukui et al., 2014, Lu et al., 2017, Park et al., 2018, Zhang et al., 2018]. Similarly, others [Kim et al., 2018, Park et al., 2018] introduce VQA models that include a module to generate an explanation for each answer.

VQA problems are particularly difficult when the relevant information that leads to a correct answer is not contained in the question-image pair. This is usually the case when relevant background knowledge, such as common sense knowledge, is required to answer a question correctly. Efforts in this line have not been able to close the gap between algorithm and human performance, especially in terms of interpretability of the resulting models. To face this challenge, recent methods augment the regular question-image pair with external knowledge that provides complementary information, usually in the form of common-sense rules [Kumar et al., 2015, A. H. Miller et al., 2016, Su et al., 2018]. Most of these models only use the question information to retrieve relevant facts (or indirectly use the image), which leads to poor performance when the question solely depends on the image.

In contrast to current VQA models, humans have an outstanding talent to attend to suitable visual cues and to retrieve relevant information from previous knowledge to answer visual questions. Using these skills, humans can fill information gaps, filter out unlikely answers, and build suitable explanations to support their answers. Following these observations, in this work, we contribute to the VQA problem by proposing a model that points out and incorporates information from the knowledge base (KB) to improve its performance and interpretability. We then use the proposed model to enhance an explanations model to generate better explanations and support its answers. Our model can outperform current approaches in terms of interpretability. Specifically, our model provides an insightful view of the extracted data from the KB, along with attention coefficients that show the relevance of each piece of information given an image and a question.

This superior interpretability is achieved using direct supervision on the information extraction mechanism to guide the model's attention towards relevant information cues. To achieve this, we automatically generate a KB filled with real-world facts in the form of triplets (Subject, Relationship, Object) that embed relevant prior knowledge about the visual world. The triplets are extracted from the scene graphs provided by the Visual Genome (VG) dataset [Krishna et al., 2017], and the question-answer pairs from both VG and VQA [Antol et al., 2015] datasets. We then provide our model with the generated KB. Since we use a supervised approach on our attention mechanism for the KB, we also generate supervision labels to identify relevant facts for a subset of the questions present in [Antol et al., 2015] and [Krishna et al., 2017]. With the attention mechanism, our model can retrieve facts from the KB that are relevant to answer a given question-image pair.

In summary, the main contributions of this work are: (i) We propose a new method to build and use the common sense KB, with labels to supervise over 400K questions from Visual Genome and VQA. (ii) We implement a integrated model that makes use of (i). (ii) We improve the existing explanations model by incorporating information from the KB to our model.

Section 2 defines the problem and our main goals. Section 3 presents related work. Section 4 provides general background theory, Section 5 contains a small survey of the datasets used for our experiments. Section 6 fully describes our methodology. Section 7 states the frameworks, libraries and hardware used to implement the different models. Section 8 diplays the results obtained from the experiments. Finally, Section 9 presents overall conclusions and future work.

2. PROBLEM DESCRIPTION

Machine Learning models are better than humans on an uncountable number of logical tasks, clustering, finding the best curve to fit some data points, classifying data points in a high dimensional space, and many others. However, they are far from reaching human-level performance on tasks that require intuition, visual understanding, or using unstructured information to solve new problems. Navigating through a populated area, self-driving cars, and VQA are some examples where humans excel, and algorithms fall behind.

Humans are the best VQA solvers. One of the reasons is their ability to take previously acquired information and use it to solve new problems. Furthermore, they can back up their answers with facts and explain them. Those are abilities that most machine learning models lack, and we believe finding a way to replicate them, is a crucial step towards building better models that can solve the kind of challenges where those skills are necessary.

Our primary goals in this work are to provide Deep Learning models with a framework to imitate the capabilities of humans in terms of retrieving and using previous knowledge, and explaining why they answer as they do.

We achieve this by providing our model with a framework that lets it use previously known facts (or real-world information), and later point out which ones are useful to answer a given question-image pair. By doing so, we improve the interpretability of the model since we can observe which facts are being used to answer. If the model answers incorrectly, we can inspect the knowledge retrieved and have a better idea of where and why the model is failing.

In order to achieve this, we first need a knowledge base to provide the required information, so we set this as a relevant goal. Specifically, we create a suitable Knowledge Base (KB) that provides such information.

3. RELATED WORK

The VQA problem has attracted considerable attention in the computer vision community [Teney et al., 2017]. The proliferation of suitable datasets, its multimodal nature, and a simple evaluation protocol partially 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 [Antol et al., 2015, Fukui et al., 2014, Yu, Yu, Xiang, et al., 2017]. Similarly, most methods pose VQA as a classification problem, where a predefined set of most popular answers is set to be the output classes. Following [Bahdanau et al., 2015], most models incorporate a soft-attention scheme that is trained to attend to relevant regions in the input image [Fukui et al., 2014, Teney et al., 2017, Yu, Yu, Fan, & Tao, 2017, Yu, Yu, Xiang, et al., 2017]. More elaborated soft-attention mechanisms have also been proposed, such as an iterative attention scheme [Yang et al., 2016] or a bidirectional co-attention mechanism [Lu et al., 2016]. Das et al. [Das et al., 2016] 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 against their human counterpart, suggesting that humans use different visual cues to answer the questions.

In terms of works that use an external KB to support or implement a VQA model, [Wu et al., 2016] augments the usual discriminative approach used by VQA models by introducing information extracted from an external KB. They, use an image captioning approach where a set of visual attributes is selected to query the KB. [Wang et al., 2018] 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 shared space, [Narasimhan & Schwing, 2018a] achieves state-of-the-art results on the FVQA dataset. Su et al. [Su et al., 2018] use a Memory Network architecture [J. Weston, 2015] to jointly embed knowledge facts and attentive visual feature vectors. 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. A significant difference with our work and previous ones is that we always use question and image directly to both filter and attend the information from the KB.

Recently, Park et al. [Park et al., 2018] presents 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. [Kim et al., 2018] extend the method to the case of videos coming from a self-driving application. [Riquelme et al., 2019] adopts a supervised approach, and we extend this model by including information from the external KB to generate better explanations.

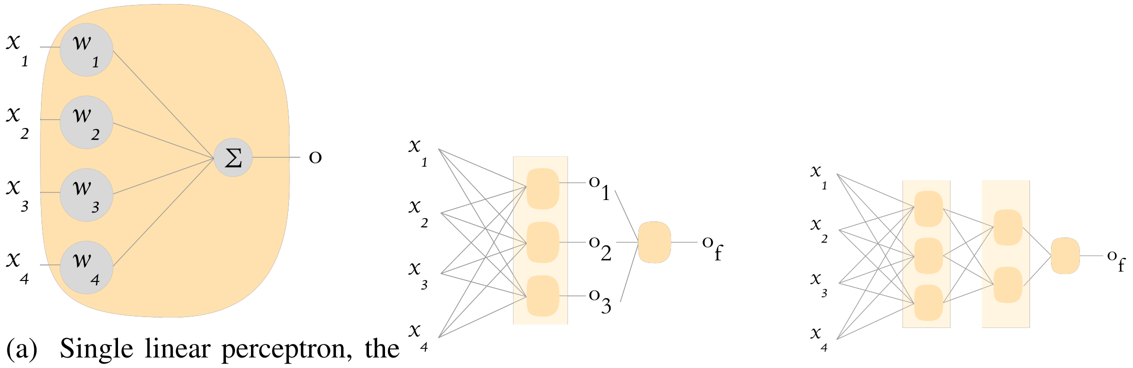
4. BACKGROUND THEORY

4.1. Deep Learning

Neural Networks (NN) are machine learning algorithms based on the idea of biological neurons that interact with each other creating many circuits or layers. Neurons receive an input signal and emit an output signal as a response to communicate with one another and generate one or more final outputs. The idea behind Neural Networks is the same; virtual neurons that connect to generate an output. A 'neuron' (Figure 4.1a) in this context is a linear unit that receives one or more input values $\vec{x}$, that are then multiplied by a set of weights $\vec{w}$ with the same dimension, and later summed generating what is known as the weighted sum ($w_s = \Sigma x_i \odot w_i$). A single weight w_b known as bias is added to the weighted sum, and an 'Activation function' f is applied to the result generating the final output $o = f(w_s + w_b)$. The bias is used to shift the activation function by a certain amount.

Neurons can be grouped to form what is called a layer (Figure 4.1b), n neurons form a layer with n outputs. Layers can be stacked (Figure 4.1c) to form a "deep neural network". Where each layer receives the outputs of the previous one as an input. The intermediate outputs of each layer, are often used as feature vectors representing the input vector $\vec{x}$.

All the weights and biases from a Neural Network, are learned from the input data during training. The output quality is quantified using a Loss Function that measures the error. This Function tells the Neural Network how off its predictions are. An algorithm called BackPropagation minimizes the error from the Loss function, updating the weights of each layer following the gradient decent approach, where the gradient of the loss function is computed and weights are updated on the opposite direction of the gradient (negative gradient) by a small step (the size of the step is a fixed or dynamic parameter called learning rate) to find weights that produce a smaller error. The previous stpes are performed in each stage of BackPropagation [Lecun, 1992].



(a) Single linear perceptron, the bias weight is ommited.

(b) Single Layer Network

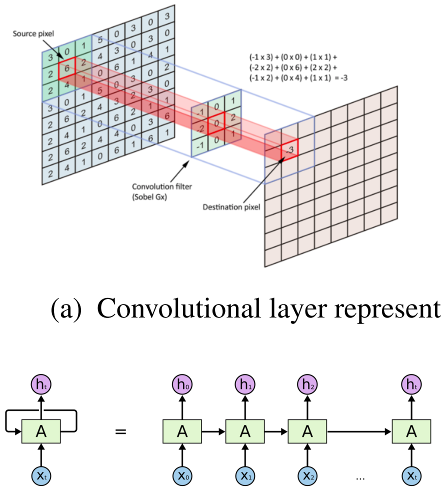
(c) 2 Layers deep neural network

Figure 4.1. (a) represents a single perceptron that can be grouped to form a layer (b), that can be stacked to create a deep neural network (c).

Deep learning has evolved fast in the last years, and multiple architectures emerged to solve different problems. On image related tasks, using Convolutional Neural Networks (CNN) [Lecun et al., 1998] has become the standard approach. They work by learning multiple filters (small tensors of weights) used to perform convolutions over the input image. For a filter $F \in R^{N \times M \times D}$ and an image I, the following equation gives the resulting convolution I_{conv} on each position (x, y).

$$\left(I_{conv}[x,y] = F * Image[x,y,c] = \sum_{d=0}^{D} \sum_{n=0}^{N} \sum_{m=0}^{M} F[n,m,d] \odot I[x-n, y-m, c-d] \right)$$

The convolution over the image can be thought of as a sliding window that performs a weighted sum on every position it slides to, as represented in Figure 4.2a for a single channel Image. For Natural Language Processing (NLP), the usual is to use Recurrent Neural Networks (RNN) [Rumelhart et al., 1986] Chung et al. [2014]. RNNs are networks with loops that allow them to maintain information through time. They have an internal state h that behaves like a memory. Here the input is not processed at once and is broken into multiple time steps (i.e., one word at a time for NLP problems). On each time step, part of the input is processed, and the memory is updated iteratively.


Source pixel
(-1 x 3) + (0 x 0) + (1 x 1) +
(-2 x 2) + (0 x 6) + (2 x 2) +
(-1 x 2) + (0 x 4) + (1 x 1) = -3
Convolution filter
(Sobel Gx)
Destination pixel
-3
h
A
X
=
A
A
A
A
...

(a) Convolutional layer representation

(b) Lstm Layer

Figure 4.2. Sub-figure (a) is a simplified representation of a CNN convolution over a single channel image. The filter slides to the first position, and later the weighted sum is performed, outputing -3. The weights from the filter of the CNN are learned with BackPropagation during training. Sub-figure (b) represents a RNN. On the right side of (b), we can see an equivalent unrolled version of the RNN, where each part of the input is processed, and the hidden state is updated within de RNN h_t for each time step. Figure 4.2a was taken from [*Simple Introduction to Convolutional Neural Networks*, 2019], and Figure 4.2b form [*Understanding LSTM Networks*, 2015].

4.2. Embeddings

An embedding in the context of Neural Networks is a low-dimensional numeric representation of a discrete or continuous variable. If we want to represent some variable with a vector $x \in \mathbb{R}^n$, one way is using one-hot encodings. This encoding is a zero vector that contains a single one. For example, if there are n animal races in the world, we can view them as different categories represented by one-hot vectors. The one-hot vector would contain a one on the dimension assigned to the category we want to represent. This is what is called a handcrafted embedding (designed by humans, not learned by machines).

Deep Neural Networks are much more powerful and can learn complex embeddings able to represent not only an animal's race, but also how long their legs are, how much it weighs, its color, etc. If we train a Deep NN to classify animal pictures, we can take the intermediate layer outputs, and use them as an embedding of the picture. i.e., an embedding of how the animal looks.

4.3. Attention Mechanisms

In some cases, it is useful to have a way of filtering out useless data or reducing the noise that it introduces. Attentions are mechanisms widely used across different neural networks. Given a vector $\vec{x}$ that represents something from the real world, the main idea is to learn a set S of coefficients $\{s_i \in [0,1] | \Sigma s_i = 1 \wedge s_i \in [0,1]\}$, that represent the level of importance of each dimension along $\vec{x}$, we then multiply each dimension of the vector by the corresponding attention coefficient s_i resulting on the attended vector $\vec{x_s}$. If s_j is ~ 1, then the dimension j of $\vec{x}$ is considered useful and $x_{s_j} \sim x_j$, the rest of the dimensions values will be much smaller in comparison, and $\vec{x_s}$ will be a better representation of the object that $\vec{x}$ is representing.

If we want to extend the attention mechanism to work with a set of vectors $\vec{S_x}$ and attend the most informative ones. The idea is the same, we learn a set of attention coefficients S, one for each vector $\vec{x_i} \in \vec{S_x}$, and depending on what we want to achieve, we could perform a weighted sum ($\Sigma s_i \cdot \vec{x_i}$) to reduce the set to just one vector that represents the whole set. Or alternatively, we could perform a $argmax$ selection of the k vectors with the highest attention coefficients and choose them as the representatives.

5. DATA

In order to train our models and generate the KB, we make use of three datasets listed and briefly described below.

5.1. VQA v2.0

VQA v2.0 [Goyal et al., 2016] is a dataset that contains multiple triplets (Image, Question Answer). VQA v2.0 builds upon the VQA dataset [Antol et al., 2015]. The dataset consists of three different splits, train, val, test. The test split has two splits own called test-dev and test-std, which are not publicly available to avoid overfitting and malpractices during the official VQA challenge competition. VQA v2.0 tries to fix the unbalanced nature of the VQA dataset, which has 614K free-form natural language questions (3 per image), and over 6 million different answers (10 answers per question).

VQA v2.0 addresses the balancing issues on VQA by collecting (Image, Question, Answer) triplets that use complementary images to make questions more dependable on the image. For a particular triplet (I, Q, A), they collect a complimentary one (I', Q, A'). Where the image I' changes the answer to A' for the same question Q. This prevents models from exploiting underlying dataset biases. They collected 443K (I, Q, A) triplets for the train split, 214K for val and 453K for the test set. Throughout this work, VQA v2.0 is referred to as VQA for notation convenience.

5.2. VQA-X

VQA-X [Park et al., 2018] is an extension of a subset of question-image pairs from VQA. They collected human Explanations for question-answer pairs that would require the intelligence of at least a 9-year-old person to avoid trivial cases. The dataset contains approximately 32.8K question-answer pairs from VQA v2. They divide the collected data into three splits, 24.8k question-answer pairs with explanations for train, 1.4k for val and 1.9K for test.

5.3. Visual Genome

Visual Genome [Krishna et al., 2017] is a well known multipurpose dataset for computer vision. It contains image descriptions, objects, attributes, relationships, and question-answer pairs. There are over 108K images taken from the intersection between MS-COCO and YFCC100M [Thomee et al., 2015]. Each one with an average of 21 objects, 16 attributes, and 18 pairwise relationships between objects. There are seven main annotation components for each image:

(i) Regions and Descriptions: Since a short caption often cannot describe in-depth an image, multiple regions are identified and annotated with bounding boxes and captions describing the region. There are, on average, 42 regions per image.

(ii) Objects and Bounding Boxes: Every object has a bounding box annotation. There are 21 objects per image on average.

(iii) Set of Attributes: Annotated objects can have zero or more attributes, such as color, states, length, and some others. Objects with visible attributes have the corresponding attribute annotated; on average, there are 16 attributes for objects per image.

(iv) Set of Relationships: Objects present in an image might be interacting with each other. To reflect this, they provide an average of 18 relationships between different annotated Objects per image.

(v) Set of region graphs: Using the objects, attributes, relationships, and regions, they build a localized graph for each region. Connecting each object within the region with their respective attributes and connecting pairs of objects with a third node representing the relationship between them within the region.

(vi) Scene Graph: It is the union of all-region the region graphs, thus containing all objects, attributes, and relationships.

(vii) Set of Question-Answer Pairs: Finally, they collect six types of questions for every image (what, when how, where who, why). Adding to a total of 1,773,258 QA pairs. There are 2 types of questions, *region based* and *free form* questions. For Region-Based questions, annotators use a particular region and formulate a question based on the contents of that region. If there is no region assigned to a question, then it is a *free form* question that has no region as visual grounding.

The dataset has three splits, train with 80% of the data, val 10%, and test 10%.

6. METHOD DESCRIPTION

Figure 6.1 illustrates the overall architecture of the proposed VQA model, including the explanations module. Using the image, question, and KB, the model predicts an answer and generates the corresponding visual and textual explanation, along with a visualization of th kB to support it. The model has three main components: the answering module (green), the KB module (blue), and the explanations module (orange). The answering module takes the image I, question Q, and the feature vector f_{KB} to predict the answer A. The KB module takes the question and image feature vectors f_Q and f_{I_α} extracted from the answering module to generate the feature vector f_{KB}. The latter contains the relevant information from the KB. Finally, the explanations module takes the generated answer A, alongside with the image embedding f_I, question embedding f_Q, f_{KB} embedding, and intermediate representations, to generate a textual and visual explanation.

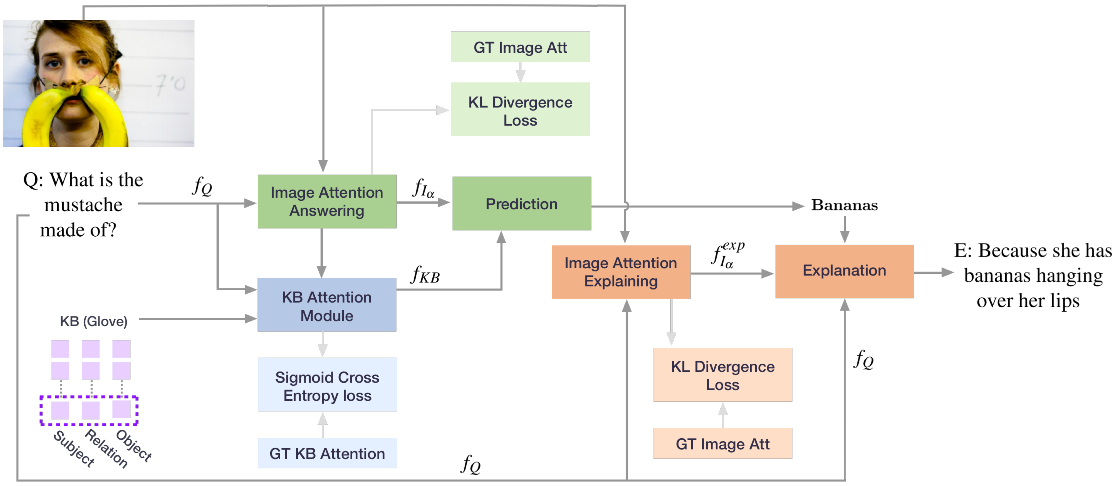


Figure 6.1. Model architecture. Answering module in green, explanations module in orange, KB module in blue. For a detailed version of this diagram, see Figure A.1 in the Appendix

6.0.1. Answering Module

The answering module builds upon a modified version of the Attn-MCB model from [Zhang et al., 2018], which in turn extends the work of Multimodal Compact Bilinear Pooling (MCB) [Fukui et al., 2014]. In this thesis, we replace all MCB layers with fully connected layers followed by element-wise operators; by doing so, we boost speed and decrease memory usage with low impact on performance. In this section, we provide a summarized description of the modified version of the Attn-MCB model.

First, to incorporate the image information, we extract the image feature vector f_I from a pre-trained CNN model. In particular we use layer $res5c$ from the ResNet-152 model [He et al., 2015]. The output of this layer is a tensor of size $14 \times 14 \times 2048$.

$$f_I(I) = ResNet152_{res5c}(I) \tag{6.1}$$

We then process the question Q to generate a 2048-dimensional vector f_Q using the concatenation of two $LSTM$ layers.

$$f_{lstm}(Q) = LSMT_1(Q) \tag{6.2}$$

$$f_Q(f_{lstm}) = [LSTM_2(f_{lstm}); f_{lstm}] \tag{6.3}$$

To combine the information from these two modalities (textual and visual), we use an FC layer to take f_I and f_Q to a joint embedding space of size 2048, to then combine them using an element-wise multiplication ($\odot$) to generate an image-question feature vector f_{IQ}. Since the shape of f_Q is 2048, we first tile f_Q a total of 14×14 times to match the dimensions of f_I. In this way, the question is going to be multiplied by each one of the 14×14 regions. After the element-wise multiplication, we apply signed square root (represented by $SgnSqrt()$) and L2 normalizations to keep the weights small. Eq (6.4) summarizes these operations.

$$f_{IQ}(f_Q, f_I) = L2(SgnSqrt(Tile(FC_1(f_Q), 14 \times 14) \odot (FC_2(f_I)))) \tag{6.4}$$

We use the f_{IQ} feature vector to generate a spatial image attention $\alpha_I(f_{IQ})$. We take f_{IQ} and apply two convolutional layers ($Conv$), followed by a Softmax activation layer. $\alpha_I(f_{IQ})$ is a tensor of size $14 \times 14 \times 2$ that contains two attention maps (region-level and object-level)Zhang et al. [2018]:

$$\alpha_I(f_{IQ}) = Softmax(Conv_2(Conv_1(f_{IQ}))) \tag{6.5}$$

As described in [Zhang et al., 2018], the attention map α_I is used to create the final image feature vector f_{I_α} applying Soft-Attention $(SoftAtt)$ [Fukui et al., 2014] over f_I. As in the following equation:

$$f_{I_\alpha}(f_I, \alpha_I) = SoftAtt(f_I, \alpha_I) \tag{6.6}$$

We supervise α_I using a Kullback-Leibler (KL) Divergence Loss. We use the labels extracted by [Zhang et al., 2018] from Visual Genome as follows:

$$AttLoss_{ans}(\alpha_{label}, \alpha_I) = KL(\alpha_{label}, \alpha_I) \tag{6.7}$$

We extend the answering module architecture with a KB module (blue boxes from Figure 6.1) capable of extracting and pointing out information from the KB. The KB module is described later on (subsection 6.1.1.6), for now we assume we have the feature vector f_{KB} with the information from the KB. To predict the answer, we need to combine all sources of information: the attended image feature vectors f_{I_α}, the original question f_Q, and the KB feature vector f_{KB}. To combine these feature vectors, each one is passed through a fully connected layer and then fused using an element-wise product. The resulting feature vector $f_{I_\alpha QKB}$ contains the information from all sources.

$$f_{I_\alpha QKB} = L2(SgnSqrt(FC_3(f_Q) \odot FC_4(f_{KB}) \odot FC_5(f_{I_\alpha}))) \tag{6.8}$$

Finally, we embed $f_{I_\alpha QKB}$ with a fully connected layer using a Softmax activation. This outputs a $L = 3000$ dimensional feature vector that represents the probability distribution over the possible answers. We select the answer with the highest probability as the output. $L = 3000$

is the answer space corresponding to the most frequent words in the VQA dataset answers, as used in Ben.

$$A = Argmax(Softmax(FC_6(f_{I_\alpha QKB}))) \tag{6.9}$$

6.1. Knowledge Base Module

As mentioned before, our primary goal is to improve the interpretability and performance of current VQA models. To do so, we provide models with an external KB with additional information that the model can point out and consume. We develop a method to access the KB to achieve it. When integrating previous knowledge from the real-world into the model, an insightful view of its internal representation becomes available by pointing out information that is considered relevant by the model. In our framework, the KB module (in blue in Figure 6.1) plays this role. The following subsections describe the structure of the KB and our automated mechanism to populate it, as well as the method to make it accessible by the model.

6.1.1. An automatically mined Knowledge Base

The KB is a collection of triplets that represent interactions between many objects from the real world. Our triplets are formated as {*Object (Obj), Relationship (Rel) , Subject (Subj)*}, for example: {*Man, Playing, Tennis*}. We automatically mine these triplets from VG and VQA, which makes its construction faster and cheaper as opposed to manual annotation.

Specifically, for every question-answer pair, we collect one or more relationships that are considered relevant and store them as triplets for the KB. To extract triplets from VG, we follow the next steps; Step 1) Identify relevant regions, Step 2) Identify relevant objects, Step 3) Store valuable relationships. Finally, for both VG and VQA, we follow Step 4) Store new relationships and labels, using just the question and answers text. Steps 1, 2, and 3 are only applicable to VG since VQA has less annotations. The strategy from Step 1) is an adaptation of the method used in Zhang et al. [2018] to extract image attention supervision labels.

6.1.1.1. Step 1) Identify relevant regions

To identify the important regions in an image, we take all the question-answer pairs, and region descriptions from that image, and use the words to match them against each other. If there is an overlap of two or more words between a particular question-answer pair and a region description, then the region with that description is considered relevant for the question-answer pair.

We process each word from the question-answer pairs and region descriptions using the Natural Language Processing Toolkit (NLTK) [Manning et al., 2014] from Stanford. Using NLTK, we tokenize and lemmatize the words; we also remove all non-informative words. For each question-answer pair and each region description, we separate the words into two sets: nouns and verbs. Additionally, we retrieve and store all words synset (synonyms set) from WordNet [G. A. Miller, 1995] to extend each set of words and increase the chances of a match between them. Since some questions from VG already have a region assigned (*region-based* questions described in Subsection5.3), we select that region as relevant and skip the matching process. We only detect relevant regions for *free form* question-answer pairs (described in Subsection 5.3).

6.1.1.2. Step 2) Identify relevant objects

Once we identify relevant regions for each question-answer pair on the image, we proceed to extract all the annotated objects that are present in a relevant region and match their names against the question-answer nouns set. Only the objects that have a match are considered as relevant objects and are stored to identify useful relationships later.

6.1.1.3. Step 3) Store useful relationships and labels

As described in Section 5, each image from VG has a set of relationships containing the interactions between the different objects in the image. We keep only those relationships that mention at least one of the relevant objects for that image. Using the same text processing from Step 1), we proceed to intersect the words present in each relationship with the words from each question-answer pair. We store all the relationships that have an overlap of two or more words as part of the KB and store them as additional labels for the question-answer pair that generated it.

6.1.1.4. Step 4) Store additional relationships and labels, using just the question and answers text

Finally, we collect more relationships and labels for each question-answer pair on every image for both VQA and VG. We first generate a massive set of triplets containing the relationships present in every single image from VG and store them all together in one set. The words of the relationships on this set are processed as in Step 1). For every question-answer pair from either VG or VQA, we intersect the word sets from each relationship with the words from the question-answer pair, and only keep the relationships that have an overlap of three or more words to avoid uninformative triplets (without using relevant regions or objects). The added relationships are stored as labels for the question-answer pairs that had a word match with it. For example, if the question-answer pair text is *'Q: What is the man going to eat? A: Hot-dog'*, we should get a match with the triplet $\{$*man, eat,hot-dog*$\}$, the relationship will be stored as part of the KB, and as a label for the given question.

6.1.1.5. KB construction.

We feed our model a KB built with one of the following methods:

A **Most frequent:** using the frequency of the relationships present in all labels, we select the top 12,000 with more presence. Since using a KB of 12,000 triplets is expensive in terms of computation and memory usage, we found it useful to create clusters.

B **Clustering:** we took the glove embedding for each component of the triplet. The glove embedding form the Subject, Relationship, and Object of the relationships are concatenated to obtain a 900-dimensional feature vector for each triplet. Using K-Means [MacQueen, 1967], we find N clusters with different groups of relationships representing different concepts. To represent all the triplets from one cluster, we took the average of the concatenated glove embedding of the triplets in the cluster. For $N = 12000$, we cluster the 36K most frequent triplets, and for $N = 3000$ we used the 12K KB from (a). We find that clustering is a good way to reduce the size of the KB, or include more information while maintaining the size since there are many triplets with redundant information (e.g., $\{$man, fly, kite$\}$, $\{$person, fly, kite$\}$, $\{$woman, fly, kite$\}$). However, there is a chance that triplets that have different information end up on the same cluster. That is why we propose the third and final method.

C **Question-Image Pre-filtering:** we do a question-image pair pre-filtering and build a KB of size $N = 1000$ specifically tailored for each question-answer pair. First, we take all the collected triplets (over 640K) and remove those that appeared 3 or fewer times as labels, which leaves approximately 36K relations. We then filter the 36K triplets according to the matching words present in the question and the names of the objects from the image. Since we don't have the answer during test and validation, and we can't know what objects are present in an image (only VG has that information, and it is not used to test the model), we use an object detector to get the name of the objects, and combine them with the nouns and verbs from the question. The same pre-filtering is performed for every question-image pair regardless of the split they come from. We use the object detector YoloV3 [Redmon & Farhadi, 2018] to find objects within the image. To generate our subset of N triplets, we first get the relationships that match three words from both the words on the question and detected object names, if there are less than N matching triplets, we proceed to store those that have two and one matching words respectively.

6.1.1.6. Module Structure

This module generates an attention map over the KB triplets and uses it to select only the most informative subset of $top-k$ triplets for the given question, to later generate the final vector containing a reduced representation of the selected triplets. Here, both information from the question and the image are combined to generate this KB attention map. We supervise the KB attention to lead the attention towards relevant triplets and improve the quality of the attention map. Once the model predicts an answer, we generate a visualization of the attention map over the $top-k$ triplets from the KB to improve interpretability and not only performance with the KB.

The KB module is based on the idea of Key-Value Memory Networks [A. H. Miller et al., 2016] used for reading comprehension placed under the context of VQA. Here, the image and question feature vectors f_{I_α} and f_Q (from the answering module - subsection 6.0.1) are used to address and identify relevant triplets from the KB using the supervised attention mechanism. Due to the size of the KB, all embedding sizes from this module are of $kb_s = 1024$ to reduce memory usage.

First, we embedd the KB using a FC layer.

$$f_{KB_{emb}}(KB) = FC_7(KB) \tag{6.10}$$

A new embedding of size kb_s is generated for f_Q and f_{I_α}. We then fuse these feature vectors via element-wise multiplication into f_{IQ}^{kb} into a joint embedding that represents both the attended image and the question:

$$f_{IQ}^{KB}(f_{I_\alpha}, f_Q) = L2\left(SgnSqrt(FC_8(f_{I_\alpha}) \odot FC_9(f_Q))\right) \tag{6.11}$$

Now, to address the triplets from the KB, f_{IQ}^{KB} is tiled and multiplied by each triplet embedding from the KB. The resulting vectors will have high weights if f_{IQ}^{KB} is aligned with the triplet on the joint space.

$$f_{KB_{addressed}}(f_{IQ}^{KB}, f_{KB_{emb}}) = L2\left(SgnSqrt(Tile(f_{IQ}^{KB}, |KB|) \odot f_{KB_{emb}})\right) \tag{6.12}$$

Finally, we create an attention map using two FC layers with a final softmax activation to reduce $f_{KB_{addressed}}$ to a feature vector of size $|KB|$. The resulting vector represents the probability distribution over all the KB triplets.

$$\alpha_{KB}(f_{KB_{addressed}}) = Softmax(FC_{11}(FC_{10}(f_{KB_{addressed}}))) \tag{6.13}$$

We supervise this attention with a softmax cross-entropy loss using the collected triplet labels. Since more than one label can be present for a single image-question pair, on each iteration, we randomly sample one label (l) and use it as the ground truth for this loss.

$$Loss_{KB}(\alpha_{KB}, l) = -\sum_{i=1}^{|KB|} l_i \cdot log(\alpha_{KB_i}) \tag{6.14}$$

Once we have the KB attention map, we use it to select the $top - k = 5$ triplets with the highest coefficients. By doing so, we reduce the noise from all the other triplets in the KB. Then we concatenate the $top - k$ triplets with their corresponding attention coefficients (6.18).

First we get the index of the highest scoring triplets:

$$top\_k_{indices}(\alpha_{KB}, k) = argmax(\alpha_{KB}, k) \tag{6.15}$$

We get the embedding for the $top - k$ triplets, and the corresponding attention coefficients:

$$f_{KB_k}(f_{KB_{emb}}, top\_k_{indices}) = f_{KB_{emb}}|_{top\_k_{indices}} \tag{6.16}$$

$$\alpha_{KB_k}(\alpha_{KB}, top\_k_{indices}) = \alpha_{KB}|_{top\_k_{indices}} \tag{6.17}$$

And we concatenate respective the attention coefficients to each one of the $top - k$ triplets embedding:

$$f_{KB_k\alpha}(f_{KB_k}, \alpha_{KB_k}) = [f_{KB_k}; \alpha_{KB_k}] \tag{6.18}$$

The final step is to generate an embedding that contains only the information from the $top-k$ triplets. We do this using two fully connected layers of size 2048 (6.19). This vector is used to predict the answer as described on subsection 6.0.1 (equations (6.17) through (6.19)).

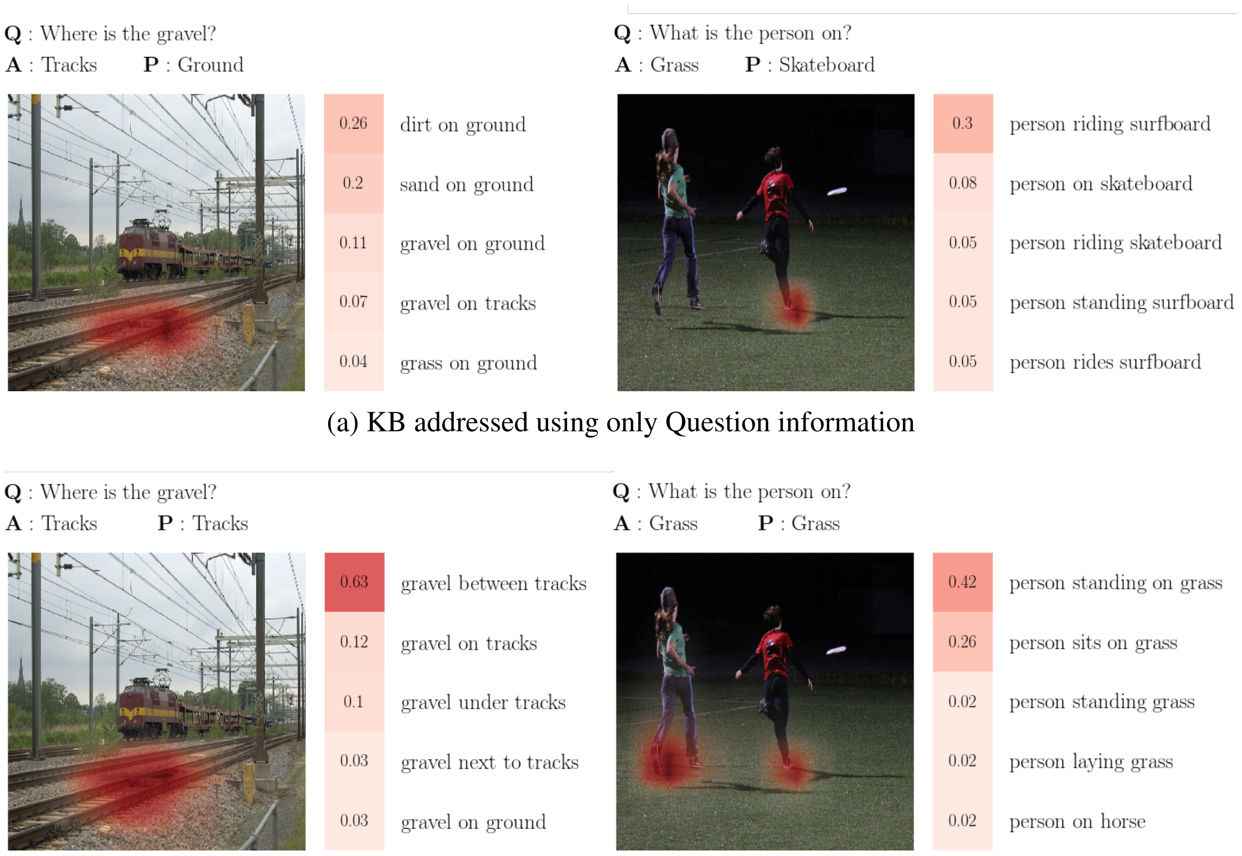
$$f_{KB}(f_{KB_k\alpha}) = FC_{13}(FC_{12}(f_{KB_k\alpha})) \tag{6.19}$$

Previous works have used only question information or image textual attributes [Kumar et al., 2015, Su et al., 2018] to address the KB instead of directly using a joint embedding. This might work on some cases but for questions that require visual information, such as "what is the man holding", global understanding of the image and the interaction between its contents is required. That is why we make use of both question and image to address the KB. Otherwise, the KB attends every triplet related to "a man holding *anything*" instead of filtering out useless triplets, Figure 6.2 shows two examples of similar cases 6.2.

6.1.2. Explanations Module

The final piece of the model is the explanations module we use to demonstrate how the KB enriches the internal representations of the model. The explanations architecture is based on the PJ-X model [Park et al., 2018], but our proposed model replaces the answering module with our answering module plus KB and uses the supervised attention from [Riquelme et al., 2019]. The explanations module is trained using our pre-trained answering model with KB, whose weights are frozen during the process. Here we use our best KB model as the answering module.

Information from the question Q, image I, answer A, KB, f_{KB}, and intermediate representations from the answering module are combined to generate explanations. Following [Park et



(a) KB addressed using only Question information

(b) KB addressed using information from both Image and Question

Figure 6.2. Here we present some cases where the question alone is not enough to identify relevant triplets. As seen on figure (b), when we include the image to address the KB, the selected triplets are more relevant. (**Q**: Question, **A**: Ground truth answer, **P**: Predicted Answer, **E** Predicted Explanation).

al., 2018], during training, the ground truth answer is used instead of the predicted answer A. We create a one-hot vector of size $L = 3000$ represents the answer, and a d-dimensional embedding f_A before feeding it to the explanations module. The answer embedding consists of two fully connected layers with a `tanh` activation function after the first layer:

$$f_A(\hat{A}) = FC_{15}(F_{14}(\hat{A})) \tag{6.20}$$

We merge f_A with the Image+Question feature vector f_{IQ} from the answering module. f_{IQ} is embedded using a fully connected layer, and then combined with f_A through an element-wise multiplication, followed by a signed square root layer and L2 normalization:

$$f_{IQA}(f_A, f_{IQ}) = L2\left(SgnSqrt(Tile(f_A) \odot FC_{16}(f_{IQ}))\right) \tag{6.21}$$

Similar to the answering module, this new feature vector is used to create new image attention maps. Two attention maps (object-level and region-level [Zhang et al., 2018]) are created using two layers of convolutions, followed by a softmax layer. These attentions are used to create a final image $f_{I_\alpha}^{exp}$ using the Soft Attention mechanism from MCB [Fukui et al., 2014].

$$\alpha_{IQA}^{exp}(f_{IQA}) = Softmax(Conv_4(Conv_3(f_{IQA}))) \tag{6.22}$$

$$f_{I_\alpha}^{exp}(f_I, \alpha_{IQA}^{exp}) = SoftAtt(f_I, \alpha_{IQA}^{exp}) \tag{6.23}$$

Following [Riquelme et al., 2019], textual and visual explanations are boosted by adding supervision for the Visual Attention of this module. Image supervision is done using a KL-divergence loss for each attention map, similar to Att-MCB[Zhang et al., 2018].

Finally, we combine the attended image feature vectors $f_{I_\alpha}^{exp}$, the original question f_Q, the answer embedding f_A, and also the KB feature vector f_{KB} to enrich the internal representation of the model. $f_{I_\alpha}^{exp}$, f_Q, and f_{KB} are embedded with a fully connected layer, and then fused with an element-wise product to create f_E.

$$f_E(f_{I_\alpha}^{exp}, f_Q, f_A) = L2\left(SgnSqrt(FC_{17}(f_{I_\alpha}^{exp}) \odot FC_{18}(f_Q) \odot FC_{19}(f_{KB}) \odot f_A)\right) \tag{6.24}$$

f_E is the feature vector that contains the information from all sources, including the KB. It will generate the answer through an LSTM decoder to generate a sequence of words for the textual explanations. As is standard, we condition each generated on the previous predicted word and the hidden state h_t of the RNN. We supervise the generated explanations with a softmax cross-entropy loss.

$$h_t(f_E, w_{t-1}, h_{t-1}) = LSTM(f_E, w_{t-1}, h_{t-1}) \tag{6.25}$$

$$w_t(h_t) = Softmax(FC_{20}(h_t)) \tag{6.26}$$

7. IMPLEMENTATION

All code from the model was implemented using a custom version of Caffe [Jia et al., 2014] that can be found here, To preprocess the text data and collect the KB, we use Standford's Natural Language processing Toolkit [Manning et al., 2014], and Tensorflow to generate the clusters. To train the VQA-KB model, we use three GeForce GTX1080Ti GPUs for around 49H when using the full-size KB (12K). We use one GPU and 72H when using prefiltering, and two GPU's for around 48 when using KB clustering. The code for our best model can be found here. The KB formats we use are provided here.

8. RESULTS AND EXPERIMENTS

To evaluate the answering module, we use the accuracy metric proposed in [Antol et al., 2015] described in Eq. (8.1). This means that we consider an answer 100% accurate if three or more annotators gave the same answer. For the Explanation module, we evaluate the impact of the KB by measuring the quality of the generated explanations using the following scores: BLEU-4 [Papineni et al., 2002], METEOR [Banerjee & Lavie, 2005], ROUGE [Lin, 2004], CIDEr [Vedantam et al., 2014], SPICE Anderson et al. [2016]. We also provide the Ranked Correlation to show that the internal representations of the model improve with the KB.

$$Accuracy(\mathcal{A}) = min\left(\frac{\text{\# humans that provided answer } \mathcal{A}}{3}, 1\right) \tag{8.1}$$

8.1. Answering with KB

We successfully create an architecture that extends the ability of current approaches to incorporate external information that goes beyond the image and question-answer pair. Thanks to the KB, we see an improvement of almost 1% on the test-std set of VQA for our best model. But most importantly, the interpretability of the model is enhanced by our attention mechanism as seen on Figures 8.1 and 8.2.

As mentioned in Subsection 6.1.1.5, we have three different KB types. i) Most frequent KB with the 12K most repeated triplets, ii) the Clustering KB, and iii) the Question-Image Prefiltering, Table 8.1, presents the results with the three versions of the KB. To validate our design, we display the results of a model that only uses the question to address the KB, one that only uses the image to address the KB, and a model without KB supervision. These last three models are all trained using $|KB| = 12,000$ (no clustering) and show how each part helps the model.

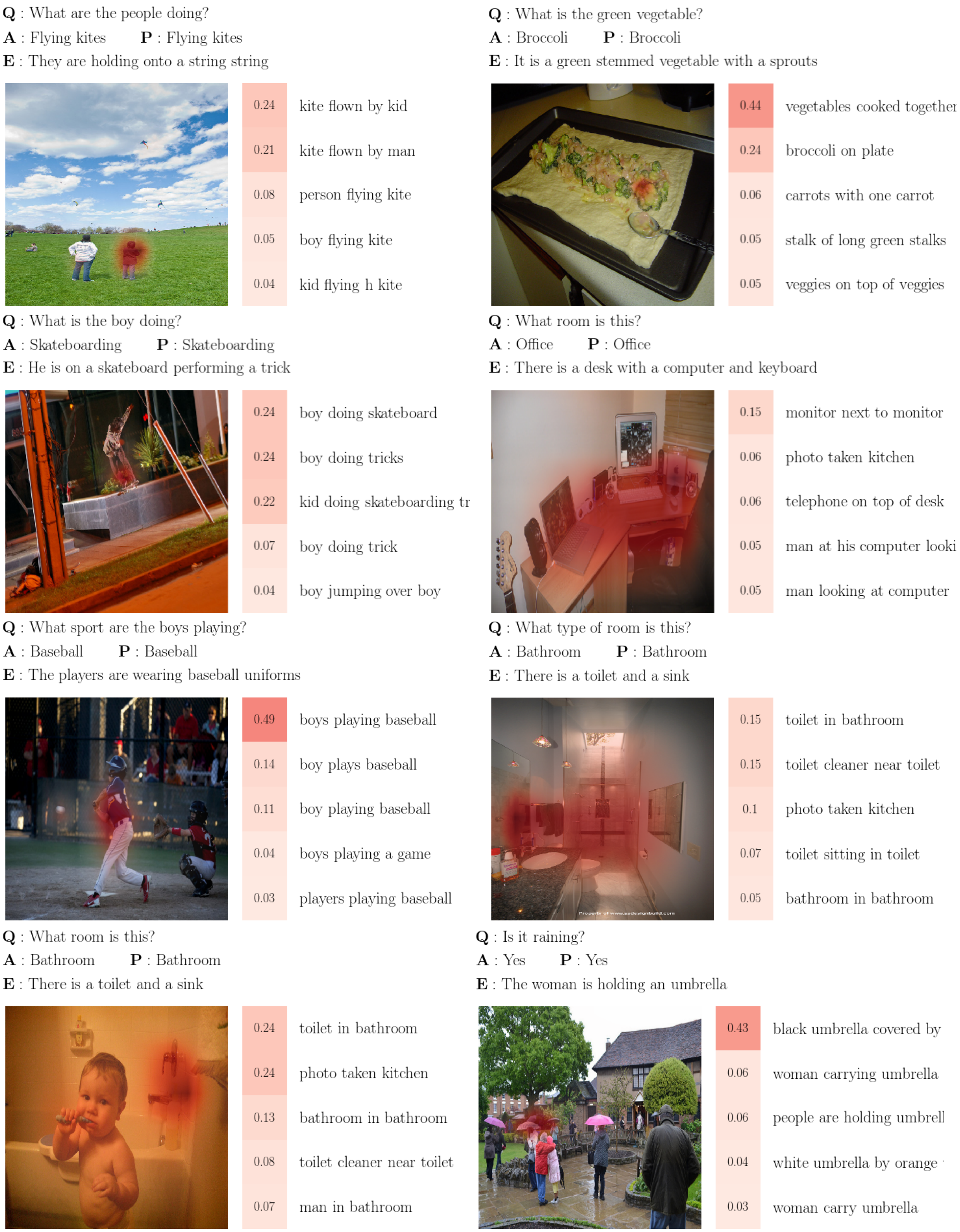
VQA Accuracy by Model		
Model	Accuracy	
	test-dev	test-std
Baseline	61.88	62.02
Baseline - KB 12K (No Supervised)	62.73	-
Baseline - KB 12K	62.9	62.96
Baseline - KB 12K Only I	62.63	-
Baseline - KB 12K Only Q	62.54	-
Baseline - KB 3K 12K Clustering	62.16	-
Baseline - KB 12K 31K Clustering	62.81	62.94
Baseline - KB 1K Prefilter	62.09	62.24

Table 8.1. Here we present the global accuracy for VQA using the VQA test-dev and test-std splits. Since the evaluation server only admits five submissions for the test-std split, we only evaluate the baselina and the best models for the test-std split. Results show how each piece of the model improves the baseline and the effect of the different KB types

Figures 8.1 and 8.2 show qualitative examples of question-answer pairs predicted by our model using the KB. For the KB attention vector, we shows the top-5 highest scoring triplets along with their attention coefficient (attention bar to the right of each image).

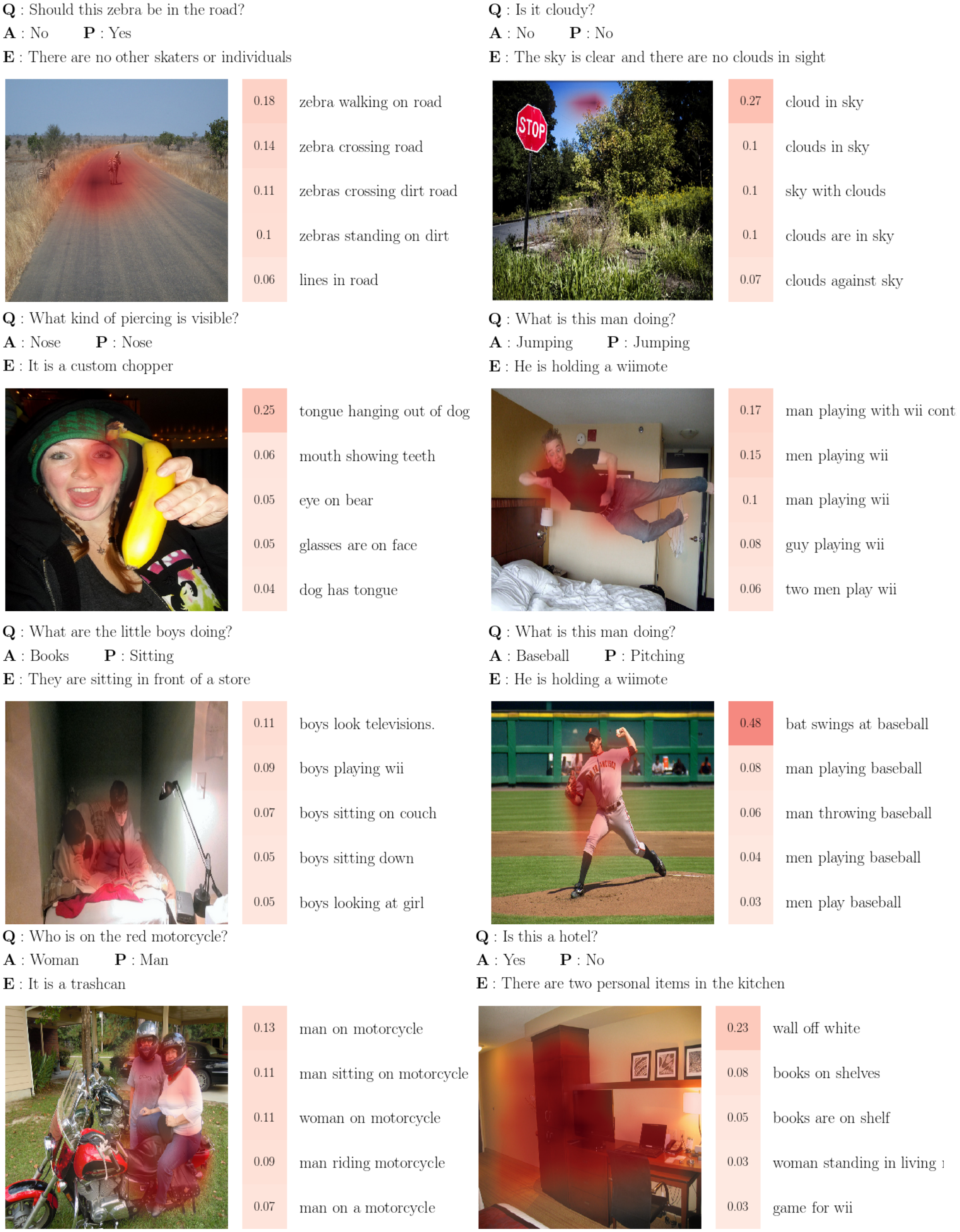
Positive examples in Figure 8.1 illustrate how the KB provides previous knowledge and contextual information to facilitate the answering process. Specifically, the KB provides factual information that is useful to build and support both the answer and the explanation. In many cases, the answer to the question is contained explicitly in one of the components of the KB triplets with the highest attention coefficient (first, third, fifth, sixth, and seventh positive examples from Figure 8.1). 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 Narasimhan & Schwing [2018b], where they predict a flag that indicates if the answer is present in the KB information and answer accordingly. In general, we notice that by using the KB the selected answer is semantically closer to the ground truth answer. This is a factor that helps to explain the great increase in the scores related to the quality of explanations generated by the model.

Negative cases are examples where the model fails in one of the three tasks: answer prediction, KB selection, and generation of explanations. Seeing what information comes from the KB



(a) Positive

Figure 8.1. Positive Qualitative Examples (**Q**: Question, **A**: Ground truth answer, **P**: Predicted Answer, **E** Predicted Explanation). The bar to the right of each image represents the attention vector for the *top* − 5 triplets used by the answering and explanations module.


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 : Nose P : Nose
E : It is a custom chopper
0.25 tongue hanging out of dog
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 : Jumping P : Jumping
E : He is holding a wiimote
0.17 man playing with wii cont
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 : Who is on the red motorcycle?
A : Woman P : Man
E : It is a trashcan
0.13 man on motorcycle
0.11 man sitting on motorcycle
0.11 woman on motorcycle
0.09 man riding motorcycle
0.07 man on a motorcycle
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
0.03 game for wii

(a) Negative

Figure 8.2. Negative Qualitative Examples (**Q**: Question, **A**: Ground truth answer, **P**: Predicted Answer, **E** Predicted Explanation). The bar to the right of each image represents the attention vector for the $top-5$ triplets used by the answering and explanations module.

helps us understand better why the model is failing. In Figure 8.2, useful insights are provided in cases where the model fails to provide the correct answer. We notice that when the KB attends irrelevant or wrong triplets, this prior might lead to incorrect predictions as seen on the fifth negative example from Figure 8.2. Overfitting and bias problems related to the training dataset are also present, for example, on the fourth negative example from Figure 8.2, the model incorrectly attends to multiple triplets that relate to the wii video game console for no apparent reason.

For further analysis of the results, Table B.1 of the appendix, contains the accuracy by each question category. The KB helps in almost all categories, but as expected fails to improve categories with questions like ”what time”, where it can not provide useful information.

8.2. Textual and Visual Explanations

The KB affects both visual and textual explanations in a positive way, we demonstrate this with quantitative and qualitative results that show how the KB leads to better internal representations and thus a better performance in regards to the explanations module.

Figure 8.1 shows qualitative results of explanations with our best model ($BE + KB_{ans} + ImageSup{+}KB_{Exp}$). In the positive cases, the visual explanation of the image (attention map) is usually focused on visual cues that are relevant to the question. The respective textual explanation is coherent and refers to visual elements pointed out in the visual explanation. For example, the visual explanation from the second positive example of Figure 8.1, is focused on a single broccoli, and accordingly, the textual explanation correctly refers to a *green stemmed vegetable with sprouts*, which also aligns with the top-5 triplets selected by the KB.

On negative cases, the model fails to explain a correctly predicted answer (third and fourth negative example from figure 8.1), which probably shows overfitting and bias problems related to the training dataset. But even though the textual explanation is wrong, in some cases it gives us a useful insight to understand why the model predicted an incorrect answer. In the fifth negative example from Figure 8.1, the model predicts *sitting*, and the explanation shows the model incorrectly believes the two children are sitting in front of a store.

Table 8.2 shows the quantitative results for the explanations module, each model is evaluated using the ground truth answer of the question and alternatively the predicted answer. We obtain a significant improvement in every score for all the metrics either indirectly incorporating the KB through the answering module, or directly with the KB vector as described in subsection 6.1.2.

Approach	GT-ans	VQA-X Test Set Score				
	Conditioning	BLEU-4	METEOR	ROUGE	CIDEr	SPICE
ME [Park et al., 2018]	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 + $\mathbf{KB}_{\mathbf{Ans}}$ + ImageSup + $\mathbf{KB}_{\mathbf{Exp}}$	Yes	**22.4**	19.4	**45.7**	**81.1**	16.1
ME [Park et al., 2018]	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 + $\mathbf{KB}_{\mathbf{Ans}}$ + ImageSup + $\mathbf{KB}_{\mathbf{Exp}}$	No	**21.8**	19.1	**45.3**	**78.3**	15.6

Table 8.2. Evaluation of textual explanations using the automatic metrics: BLEU-4, METEOR, ROUGE, CIDEr, and SPICE. Reference sentence for human and automatic evaluation is always an explanation — all in %. Our models compare favorably to baseline. *ImageSup* is the image supervision added in [Riquelme et al., 2019]

Table 8.3 shows the Ranked Correlation of the visual explanations attention maps. The KB usage helps improve the internal visual representations of the model, we can see that the attentions generated using the KB are more correlated with human attention.

Approach	Rank Correlation	
	VQA-X	VQA-X gt
ME [Park et al., 2018]	+ 0.3423	
Baseline Explanations (BE)	+ 0.3465	+ 0.3467
BE + KB_{Ans}	+ 0.3468	+ 0.3456
BE + ImageSup	+ 0.3897	+ 0.3902
BE + KB_{Ans} + ImageSup	+ 0.3938	+ 0.3945
BE + $\mathbf{KB}_{\mathbf{Ans}}$ + ImageSup + $\mathbf{KB}_{\mathbf{Exp}}$	**+ 0.4007**	**+ 0.4015**

Table 8.3. Evaluation results for the explanations Image Attention (without the use of ground truth answer). Compared against the VQA-X test set attention labels using the Rank Correlation Metric - higher is better. Our models compare favorably to the baselines thanks to the enriched internal representations that come from the KB. *ImageSup* is the image supervision added in [Riquelme et al., 2019]

9. CONCLUSIONS

In this work, we focus on improving the interpretability and performance of VQA models. In particular, we measure interpretability by measuring the quality of explanations generated by the VQA framework, and the inclusion of the KB visualization with the top-5 triplets. We achieve state-of-the-art explanation performance by introducing an effective mechanism that leverages an external Knowledge Base (KB) to produce better answers and explanations. We show that such KB can be mined automatically from scene graphs in Visual Genome. Furthermore, we show that our algorithm can attend to a small number of relevant facts among a large number of entries in the KB.

Interpretability is a feature that opens up the black-box nature of deep neural networks, as demonstrated with the qualitative results from Section 8, the attention visualization of the KB in conjunction with the visual and textual explanations, is a powerful tool to examine the answers from the model and evaluate whether it is failing due to miss interpretation, overfitting problems, or shortcomings.

However, the power of this tool comes at a price. On the one hand, the model needs to balance multiple losses (prediction, kb attention, visual explanations, and textual explanations) and use multiple datasets to train each one of the supervised auxiliary tasks (either one at a time or simultaneously) which is difficult. On the other hand, the size of the KB makes training slow and requires special care. We had to optimize the memory usage, and reduce the number of parameters of the model to be able to work with the available hardware. We reduced the number of parameters form the answering and KB modules, besides training the explanations module apart from the latter. Otherwise, it did not fit into the GPUs memory.

We propose two methods to reduce the size of the KB and tackle the problem of an extensive KB that does not fit into memory. The two approaches are as follows: i) Clustering and ii) Question-Image Prefiltering. With i), we reduced the KB from 12K to 3K and still surpassed the base model. We also manage to use a KB of size 31K clustered to a version of 12K obtaining a much better result. This demonstrates that regardless of the model, the amount of information has a significant impact on performance, and a bigger KB might yield even better results with the same model. Clustering is a great method to reduce the size, but it reduces the interpretability of the model since each element of the KB represents a group of triplets instead of just one. With ii), we were able to get a smaller KB while maintaining interpretability.

Here we present some of the most interesting ideas left for future work. The focus of future work should be on improving the quality of the clusters, text embeddings, and the extraction of triplets from different sources to improve the quality and amount of knowledge a model can use. So we propose the following: i) Clustering yielded good results, but not as good as we expected. We believe that using an algorithm that can learn the appropriate number of clusters should generate better clusters than K-Means and increase the performance for this approach. So we propose the usage of an unsupervised clustering algorithm like Gaussian Mixtures Model [Frigui & Krishnapuram, 1997]. ii) Using a combined approach of clustering and prefiltering to remove uninformative clusters according to each question-image pair might be a better way to reduce even more the size of the KB without losing the relevant information. iii) Increase the collected triplets by exploiting a structured knowledge base like ConceptNet [Liu & Singh, 2004] to feed the model a vast amount of information about the world. Finally, iv) we believe that using a better initial embedding for the question and triplets can help the KB module improve the triplets selection due to a semantically better embedding. To this end, we propose the use of a transformer network like BERT [Devlin et al., 2018].

This work demonstrates that incorporating mechanisms that grant models the ability to incorporate, use, and point out external information from a KB, is an essential step towards generating models with a richer internal representation of our world, and thus are more semantically understanding and interpretable.

REFERENCES

Anderson, P., Fernando, B., Johnson, M., & Gould, S. (2016, Jul). SPICE: Semantic Propositional Image Caption Evaluation. *arXiv e-prints*, arXiv:1607.08822.

Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., & Zhang, L. (2018). *Bottom-up and top-down attention for image captioning and visual question answering.* (arXiv:1707.07998)

Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C. L., & Parikh, D. (2015). VQA: visual question answering. *CoRR*, *abs/1505.00468*. Retrieved from http://arxiv.org/abs/1505.00468

Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate..

Banerjee, S., & Lavie, A. (2005). 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*, 65–72.

Chung, J., Gülçehre, Ç., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, *abs/1412.3555*. Retrieved from http://arxiv.org/abs/1412.3555

Das, A., Agrawal, H., Zitnick, C. L., Parikh, D., & Batra, D. (2016). Human attention in visual question answering: Do humans and deep networks look at the same regions?

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, *abs/1810.04805*. Retrieved from http://arxiv.org/abs/1810.04805

Frigui, H., & Krishnapuram, R. (1997, July). Clustering by competitive agglomeration. *Pattern Recogn.*, *30*(7), 1109–1119. Retrieved from http://dx.doi.org.pucdechile.idm.oclc.org/10.1016/S0031-3203(96)00140-9 doi: 10.1016/S0031-3203(96)00140-9

Fukui, A., Park, D. H., Yang, D., Rohrbach, A., Darrell, T., & Rohrbach, M. (2014).

Multimodal compact bilinear pooling for visual question answering and visual grounding. (arXiv:1606.01847)

Gan, C., Li, Y., Li, H., Sun, C., & Gong, B. (2017). Vqs: Linking segmentations to questions and answers for supervised attention in vqa and question-focused semantic segmentation. In *Iccv.*

Goodman, B., & Flaxman, S. (2016). *Eu regulations on algorithmic decision-making and a "right to explanation".* Retrieved from http://arxiv.org/abs/1606.08813 (cite arxiv:1606.08813Comment: presented at 2016 ICML Workshop on Human Interpretability in Machine Learning (WHI 2016), New York, NY)

Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., & Parikh, D. (2016). Making the V in VQA matter: Elevating the role of image understanding in visual question answering. *CoRR*, *abs/1612.00837*. Retrieved from http://arxiv.org/abs/1612.00837

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning for image recognition. *CoRR*, *abs/1512.03385*. Retrieved from http://arxiv.org/abs/1512.03385

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., … Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*.

J. Weston, S. C. . A. B. (2015). Memory networks. In *Iclr.*

Kim, J., Rohrbach, A., Darrell, T., Canny, J., & Akata, Z. (2018). *Textual explanations for self-driving vehicles.* (arXiv:1807.11546)

Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., … Fei-Fei, L. (2017, May). Visual genome: Connecting language and vision using crowdsourced dense image annotations. *Int. J. Comput. Vision*, *123*(1), 32–73. Retrieved from https://doi.org/10.1007/s11263-016-0981-7 doi: 10.1007/s11263-016-0981-7

Kumar, A., Irsoy, O., Su, J., Bradbury, J., English, R., Pierce, B., … Socher, R. (2015). Ask me anything: Dynamic memory networks for natural language processing. *CoRR*, *abs/1506.07285*. Retrieved from http://arxiv.org/abs/1506.07285

LeCun, Y. (1992). A theoretical framework for back-propagation. In P. Mehra & B. Wah (Eds.), *Artificial neural networks: concepts and theory.* Los Alamitos, CA: IEEE Computer Society Press.

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. In *Proceedings of the ieee* (pp. 2278–2324).

Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*.

Liu, H., & Singh, P. (2004, October). Conceptnet &mdash; a practical commonsense reasoning tool-kit. *BT Technology Journal*, *22*(4), 211–226. Retrieved from http://dx.doi.org/10.1023/B:BTTJ.0000047600.45421.6d doi: 10.1023/B:BTTJ.0000047600.45421.6d

Lu, J., Yang, J., Batra, D., & Parikh, D. (2016). Hierarchical question-image co-attention for visual question answering. In *Advances in neural information processing systems* (pp. 289–297).

Lu, J., Yang, J., Batra, D., & Parikh, D. (2017). *Hierarchical question-image co-attention for visual question answering.* (arXiv:1606.00061)

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations..

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., & McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In *Association for computational linguistics (acl) system demonstrations* (pp. 55–60). Retrieved from http://www.aclweb.org/anthology/P/P14/P14-5010

Miller, A. H., AdamFisch, Dodge, J., Karimi, A.-H., Bordes, A., & Weston, J. (2016). *Key-value memory networks for directly reading documents.* (arXiv:1606.03126)

Miller, G. A. (1995). Wordnet: A lexical database for english. *COMMUNICATIONS OF THE ACM*, *38*, 39–41.

Narasimhan, M., & Schwing, A. G. (2018a). Straight to the facts: Learning knowledge base retrieval for factual visual question answering. In *Eccv.*

Narasimhan, M., & Schwing, A. G. (2018b). Straight to the facts: Learning knowledge base retrieval for factual visual question answering. *CoRR*, *abs/1809.01124*. Retrieved from http://arxiv.org/abs/1809.01124

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. *Proceedings of the Annual Meeting of the As- sociation for Computational Linguistics (ACL)*, 311–318.

Park, D. H., Hendricks, L. A., Akata, Z., Rohrbach, A., Schiele, B., Darrell, T., & Rohrbach,

M. (2018). *Multimodal explanations: Justifying decisions and pointing to the evidence.* (arXiv:1802.08129)

Qiao, T., Dong, J., & Xu, D. (2018). Exploring human-like attention supervision in visual question answering..

Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. *CoRR*, *abs/1804.02767*. Retrieved from http://arxiv.org/abs/1804.02767

Riquelme, F., Soto, A., Goyeneche, A. D., & Niebles, J. C. (2019). Explaining vqa predictions using visual grounding and a knowledge base (en preparación). *Image and Vision Computing*, *XX*, XXX - XXX. Retrieved from http://www.sciencedirect.com/science/article/pii/xxxxxxxxxxx

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1. In D. E. Rumelhart, J. L. McClelland, & C. PDP Research Group (Eds.), (pp. 318–362). Cambridge, MA, USA: MIT Press. Retrieved from http://dl.acm.org/citation.cfm?id=104279.104293

Simple introduction to convolutional neural networks. (2019). http://web.archive.org/web/20080207010024/http://www.808multimedia.com/winnt/kernel.htm. (Accessed: 2019-08-26)

Su, Z., Zhu, C., Dong, Y., Cai, D., Chen, Y., Li, J., & Wechat, T. (2018). *Learning visual knowledge memory networks for visual question answering.* (arXiv:1806.04860)

Teney, D., Anderson, P., He, X., & van den Hengel, A. (2017). Tips and tricks for visual question answering: Learnings from the 2017 challenge. *CoRR*, *abs/1708.02711*. Retrieved from http://arxiv.org/abs/1708.02711

Thomee, B., Shamma, D. A., Friedland, G., Elizalde, B., Ni, K., Poland, D., ... Li, L. (2015). The new data and new challenges in multimedia research. *CoRR*, *abs/1503.01817*. Retrieved from http://arxiv.org/abs/1503.01817

Understanding lstm networks. (2015). https://colah.github.io/posts/2015-08-Understanding-LSTMs/. (Accessed: 2019-08-26)

Vedantam, R., Zitnick, C. L., & Parikh, D. (2014, Nov). CIDEr: Consensus-based Image Description Evaluation. *arXiv e-prints*, arXiv:1411.5726.

Wang, P., Wu, Q., Shen, C., van den Hengel, A., & Dick, A. R. (2018). FVQA: fact-based visual question answering. *CoRR*. Retrieved from `http://arxiv.org/abs/1606.05433`

Wu, Q., Wang, P., Shen, C., van den Hengel, A., & Dick, A. R. (2016). *Ask me anything: Free-form visual question answering based on knowledge from external sources.*

Yang, Z., He, X., Gao, J., Deng, L., & Smola, A. (2016). Stacked attention networks for image question answering. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 21–29).

Yu, Z., Yu, J., Fan, J., & Tao, D. (2017). Multi-modal factorized bilinear pooling with co-attention learning for visual question answering. In *Iccv.*

Yu, Z., Yu, J., Xiang, C., Fan, J., & Tao, D. (2017, 08). Beyond bilinear: Generalized multimodal factorized high-order pooling for visual question answering. *IEEE Transactions on Neural Networks and Learning Systems*, *PP*. doi: 10.1109/TNNLS.2018.2817340

Zhang, Y., Niebles, J. C., & Soto, A. (2018). *Interpretable visual question answering by visual grounding from attention supervision mining.* (`arXiv:1808.00265`)

APPENDIX

Here, we provide further analysis of the performance of our proposed model. In particular, we extend our discussion of the accuracy by category of the KB model.

In this section, we provide further details about the architecture of our model. We also describe the pre-filtering approach left for future work on section 6.1.1.5, and an analysis of the accuracy by questions categories of our model.

A. DETAILED MODEL ARCHITECTURE

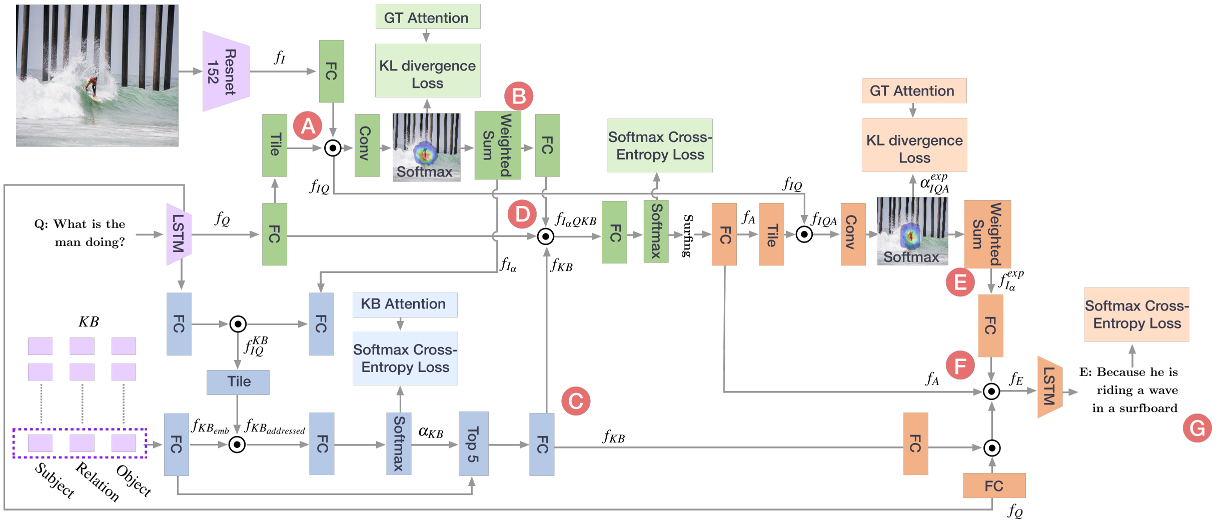


Figure A.1. Detailed model architecture. Answering module in green, explanations module in orange, KB module in blue. Dimensions are specified in gray for some intermediate feature vectors. Reference points for intermediate features are provided in red.

Figure A.1 shows the detailed architecture of our model. The answering module (green) takes the image f_I, question f_Q, and KB feature vector f_{KB} (from the KB module) to predict an answer. We start by combining the image and a question embedding to create f_{IQ} (point a in Figure A.1). We use this feature vector to generate an attention map over the image, which is supervised using Kullback-Leibler (KL) divergence loss. The attention map is used to create an attended image feature vector $f_{I\alpha}$ (point b in Figure A.1). Finally, this attention is combined with the question embedding and the KB feature vector to predict an answer (point d in Figure

A.1) as a classification task over the 3000 most repeated answers of VQA. This classification is supervised via Softmax Cross Entropy Loss.

The KB module (blue) takes the question f_Q, attended image $f_{I\alpha}$ and the Knowledge Base KB as inputs in order to generate a KB feature vector f_{KB} (point c in Figure A.1). We generate a new embedding for the question f_Q, image feature vector $f_{I\alpha}$ and KB, these embeddings are combined via element-wise multiplication to generate an attention map over the KB. We supervise this attention map with a softmax cross-entropy loss. The attention map is used to select the top $k = 5$ triplets from the KB. These k triplets are embedded using two fully connected layers that generate our final KB feature vector f_{KB}. This feature vector carries all the relevant KB information for the given input image and question.

The explanations module (orange), takes the question f_Q, image plus question f_{IQ}, predicted answer $\mathcal{A}$ and image f_I to generate both visual and textual explanations. We generate an embedding for the answer and f_{IQ}, and combine them via element-wise multiplication. This feature vector is used to generate a new attention map over the image, which corresponds to the visual pointing explanations (point e in Figure A.1). We supervise the image attention with a KL divergence loss. To generate the attended image embedding, we use the attention map $^{Exp}_{i_\alpha}$ to do an element-wise multiplication with f_{IQ}. The attended image, the question embedding, the KB embedding, and the predicted answer embedding are fused by another element-wise multiplication to generate an explanations feature vector f_E (point f in Figure A.1). This feature vectoris fed to an $LSTM$ decoder to produce the final textual explanations (point g in Figure A.1), which is supervised with Softmax Cross-Entropy Loss.

B. KB PERFORMANCE BY CATEGORY

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. Here our best KB model achieves an accuracy of 60.5, and the baseline 59.57 (both models were trained only using the train split for this study).

The KB helps to increase the performance of almost every category of questions in the VQA dataset [Antol et al., 2015] (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 $\{Subj(Noun), Rel(Verb), Obj(Noun)\}$.

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 [Krishna et al., 2017] that depicts relations between objects.

Type	**Baseline**	**Baseline-KB**	**Gain**
can you +++	74.071	71.881	2.19
could ++	79.66	78.657	1.003
how ++	29.195	27.733	1.462
how many people are in ++	45.713	44.663	1.05
is he +++	79.126	77.029	2.098
is it +++	87.487	84.924	2.563
is the ++	76.003	74.642	1.361
is the man ++	74.851	73.56	1.29
is this a ++	78.344	77.286	1.057
is this an ++	77.944	76.506	1.438
what are ++	53.952	52.763	1.189
what color ++	66.947	65.784	1.162
what color are the ++	70.581	69.567	1.013
what is ++	43.685	42.318	1.367
what is in the ++	49.342	47.715	1.627
what is the ++	47.399	46.318	1.081
what is the man +++	56.425	53.857	2.569
what is the person +++	58.811	56.233	2.578
what is this +++	60.33	58.166	2.164
what kind of ++	54.723	53.56	1.163
what number is ++	9.539	8.172	1.367
what sport is ++	87.569	86.565	1.004
what type of ++	54.356	53.025	1.332
which ++	43.86	42.819	1.042
why is the ++++	21.887	17.665	4.222
has —	72.685	74.81	-2.125
what time –	23.013	24.044	-1.031

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