Ask Me Anything: Free-form Visual Question Answering Based on Knowledge from External Sources

Qi Wu, Peng Wang, Chunhua Shen, Anthony Dick, Anton van den Hengel

School of Computer Science, The University of Adelaide, Australia

{qi.wu01,p.wang,chunhua.shen,anthony.dick,anton.vandenhengel}@adelaide.edu.au

Abstract

We propose a method for visual question answering which combines an internal representation of the content of an image with information extracted from a general knowledge base to answer a broad range of image-based questions. This allows more complex questions to be answered using the predominant neural network-based approach than has previously been possible. It particularly allows questions to be asked about the contents of an image, even when the image itself does not contain the whole answer. The method constructs a textual representation of the semantic content of an image, and merges it with textual information sourced from a knowledge base, to develop a deeper understanding of the scene viewed. Priming a recurrent neural network with this combined information, and the submitted question, leads to a very flexible visual question answering approach. We are specifically able to answer questions posed in natural language, that refer to information not contained in the image. We demonstrate the effectiveness of our model on two publicly available datasets, Toronto COCO-QA [23] and VQA [1] and show that it produces the best reported results in both cases.

Attributes:
umbrella
beach
sunny
day
people
sand
laying
blue
green
mountain

Internal Textual Representation:
A group of people enjoying a sunny day at the beach with umbrellas in the sand.

External Knowledge:
An umbrella is a canopy designed to protect against rain or sunlight. Larger umbrellas are often used as points of shade on a sunny beach. A beach is a landform along the coast of an ocean. It usually consists of loose particles, such as sand….

Question Answering:
Q: Why do they have umbrellas? **A :** Shade.

Figure 1. A real case of question answering based on an internal textual representation and external knowledge. All of the attributes, textual representation, knowledge and answer are produced by our VQA model. Underlined words indicate the information required to answer the question.

1. Introduction

Visual question answering (VQA) is distinct from many problems in Computer Vision because the question to be answered is not determined until run time [1]. In more traditional problems such as segmentation or detection, the single question to be answered by an algorithm is predetermined, and only the image changes. In visual question answering, in contrast, the form that the question will take is unknown, as is the set of operations required to answer it. In this sense it more closely reflects the challenge of general image interpretation.

VQA typically requires processing both visual information (the image) and textual information (the question and answer). One approach to Vision-to-Language problems,

such as VQA and image captioning, which interrelate visual and textual information is based on a direct method pioneered in machine language translation [7, 26]. This direct approach develops an encoding of the input text using a Recurrent Neural Network (RNN) and passes it to another RNN for decoding.

The Vision-to-Language version of this direct translation approach uses a Convolutional Neural Network (CNN) to encode the image, and an RNN to encode and generate text. This approach has been well studied. The image captioning methods developed in [9, 14, 21, 28, 32], for example, learn a mapping from images to text using CNNs trained on object recognition, and word embeddings trained on large scale text corpora. Visual question answer-

ing is a significantly more complex problem than image captioning, not least because it requires accessing information not present in the image. This may be common sense, or specific knowledge about the image subject. For example, given an image, such as Figure 1, showing 'a group of people enjoying a sunny day at the beach with umbrellas', if one asks a question 'why do they have umbrellas?', to answer this question, the machine must not only detect the scene 'beach', but must know that 'umbrellas are often used as points of shade on a sunny beach'. Recently, Antol *et al.* [1] also have suggested that VQA is a more "AI-complete" task since it requires multimodal knowledge beyond a single sub-domain. Our proposed system finally gives the right answer 'shade' for the above real example.

Large-scale Knowledge Bases (KBs), such as Freebase [4] and DBpedia [2], have been used successfully in several natural language Question Answering (QA) systems [3, 10]. However, VQA systems exploiting KBs are still relatively rare. Gao *et al.* [11] and Malinowski *et al.* [20] do not use a KB at all. Zhu *et al.* [33] do use a KB, but it is created specifically for the purpose, and consequently contains a small amount of very specific, largely image-focused, information. This in turn means that only a specific set of questions may be asked of the system, and the query generation approach taken mandates a very particular question form. The method that we propose here, in contrast, is applicable to general (even publicly created) KBs, and admits general questions.

In this work, we fuse an automatically generated description of an image with information extracted from an external KB to provide an answer to a general question about the image (See Figure 2). The image description takes the form of a set of captions, and the external knowledge is text-based information mined from a Knowledge Base. Given an image-question pair, a CNN is first employed to predict a set of attributes of the image. The attributes cover a wide range of high-level concepts, including objects, scenes, actions, modifiers and so on. A state-of-the-art image captioning model [30] is applied to generate a series of captions based on the attributes. We then use the detected attributes to extract relevant information from the KB. Specifically, for each of the top-5 attributes detected in the image we generate a query which may be applied to a Resource Description Framework (RDF) KB, such as DBpedia. RDF is the standard format for large KBs, of which there are many. The queries are specified using Semantic Protocol And RDF Query Language (SPARQL). We encode the paragraphs extracted from the KB using Doc2Vec [15], which maps paragraphs into a fixed-length feature representation. The encoded attributes, captions, and KB information are then input to an LSTM which is trained so as to maximise the likelihood of the ground truth answers in a training set.

The approach we propose here combines the generality of information that using a KB allows with the generality of question that the LSTM allows. In addition, it achieves an accuracy of 69.73% on the Toronto COCO-QA, while the latest state-of-the-art is 55.92%. We also produce the best results on the VQA evaluation server (which does not publish ground truth answers for its test set), which is 59.44%.

2. Related Work

Malinowski *et al.* [18] were among the first to study the VQA problem. They proposed a method that combines semantic parsing and image segmentation with a Bayesian approach to sample from nearest neighbors in the training set. This approach requires human defined predicates, which are inevitably dataset-specific. This approach is also very dependent on the accuracy of the image segmentation algorithm and on the estimated image depth information. Tu *et al.* [27] built a query answering system based on a joint parse graph from text and videos. Geman *et al.* [12] proposed an automatic 'query generator' that was trained on annotated images and produced a sequence of binary questions from any given test image. Each of these approaches places significant limitations on the form of question that can be answered.

Most recently, inspired by the significant progress achieved using deep neural network models in both computer vision and natural language processing, an architecture which combines a CNN and RNN to learn the mapping from images to sentences has become the dominant trend. Both Gao *et al.* [11] and Malinowski *et al.* [20] used RNNs to encode the question and output the answer. Whereas Gao *et al.* [11] used two networks, a separate encoder and decoder, Malinowski *et al.* [20] used a single network for both encoding and decoding. Ren *et al.* [23] focused on questions with a single-word answer and formulated the task as a classification problem using an LSTM. A single-word answer dataset COCO-QA was published with [23]. Ma *et al.* [17] used CNNs to both extract image features and sentence features, and fused the features together with another multimodal CNN. Antol *et al.* [1] proposed a large-scale open-ended VQA dataset based on COCO, which is called VQA. They also provided several baseline methods which combined both image features (CNN extracted) and question features (LSTM extracted) to obtain a single embedding and further built a MLP (Multi-Layer Perceptron) to obtain a distribution over answers. Our framework also exploits both CNNs and RNNs, but in contrast to preceding approaches which use only image features extracted from a CNN in answering a question, we employ multiple sources, including image content, generated image captions and mined external knowledge, to feed to an RNN to answer questions.

The quality of the information in the KB is one of the primary issues in this approach to VQA. The problem is that KBs constructed by analysing Wikipedia and similar
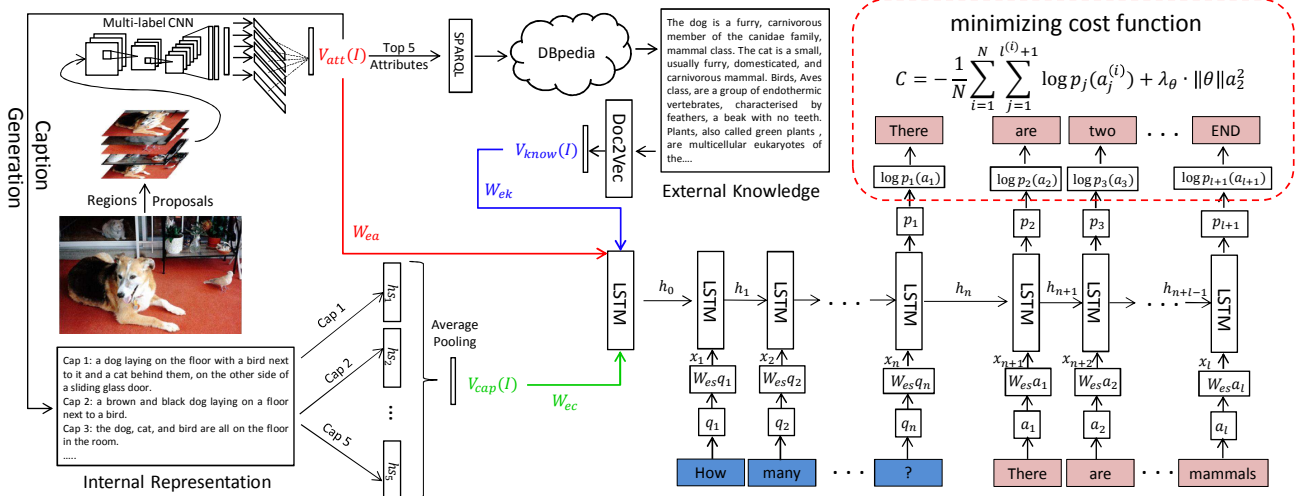
Figure 2. Our proposed framework: given an image, a CNN is first applied to produce the attribute-based representation $V_{att}(I)$. The internal textual representation is made up of image captions generated based on the image-attributes. The hidden state of the caption-LSTM after it has generated the last word in each caption is used as its vector representation. These vectors are then aggregated as $V_{cap}(I)$ with average-pooling. The external knowledge is mined from the KB (in this case DBpedia) and the responses encoded by Doc2Vec, which produces a vector $V_{know}(I)$. The 3 vectors $\mathbf{V}$ are combined into a single representation of scene content, which is input to the VQA LSTM model which interprets the question and generates an answer.

are patchy and inconsistent at best, and hand-curated KBs are inevitably very topic specific. Using visually-sourced information is a promising approach to solving this problem [16, 24], but has a way to go before it might be usefully applied within our approach. Thus, although our SPARQL and RDF driven approach can incorporate any information that might be extracted from a KB, the limitations of the existing available KBs mean that the text descriptions of the detected attributes is all that can be usefully extracted.

Zhu *et al.* [33], in contrast used a hand-crafted KB primarily containing image-related information such as category labels, attribute labels and affordance labels, but also some quantities relating to their specific question format such as GPS coordinates and similar. The questions in that system are phrased in the DBMS query language, and are thus tightly coupled to the nature of the hand-crafted KB. This represents a significant restriction on the form of question that might be asked, but has the significant advantage that the DBMS is able to respond decisively as to whether it has the information required to answer the question. Instead of building a problem-specific KB, we use a pre-built large-scale KB (DBpedia [2]) from which we extract information using a standard RDF query language. DBpedia has been created by extracting structured information from Wikipedia, and is thus significantly larger and more general than a hand-crafted KB. Rather than having a user to pose their question in a formal query language, our VQA system is able to encode questions written in natural language automatically. This is achieved without manually specified formalization, but rather depends on processing a suitable

training set. The result is a model which is very general in the forms of question that it will accept.

3. Extracting, Encoding, and Merging

The key differentiator of our approach is that it is able to usefully combine image information with that extracted from a KB, within the LSTM framework. The novelty lies in the fact that this is achieved by representing both of these disparate forms of information as text before combining them. Figure 2 summarises how this is achieved. We now describe each step in more details.

3.1. Attribute-based Image Representation

Our first task is to describe the image content in terms of a set of attributes. Each attribute in our vocabulary is extracted from captions from MS COCO [5], a large-scale image-captioning dataset. An attribute can be any part of speech, including object names (nouns), motions (verbs) or properties (adjectives). Because MS COCO captions are created by humans, attributes derived from captions are likely to represent image features that are of particular significance to humans, and are thus likely to be the subject of image-based questions.

We formalize attribute prediction as a multi-label classification problem. To address the issue that some attributes may only apply to image sub-regions, we follow Wei *et al.* [29] to design a region-based multi-label classification framework that takes an arbitrary number of sub-region proposals as input. A shared CNN is connected with each proposal, and the CNN outputs from different proposals are

Top 5 Attributes:
players, catch, bat, baseball, swing

Generated Captions:
A baseball player swing a bat at a ball.
A baseball player holding a bat on a field.
A baseball player swinging a bat on a field.
A baseball player is swinging a bat at a ball.
A batter catcher and umpire during a baseball game.

Top 5 Attributes:
field, two, tree, grass, giraffe

Generated Captions :
Two giraffes are standing in a grassy field.
A couple of giraffe standing next to each other.
Two giraffes standing next to each other in a field.
A couple of giraffe standing next to each other on a lush green field.

Top 5 Attributes:
pizza, bottle, sitting, table, beer

Generated Captions :
A large pizza sitting on top of a table.
A pizza sitting on top of a white plate.
A pizza sitting on top of a table next to a beer.
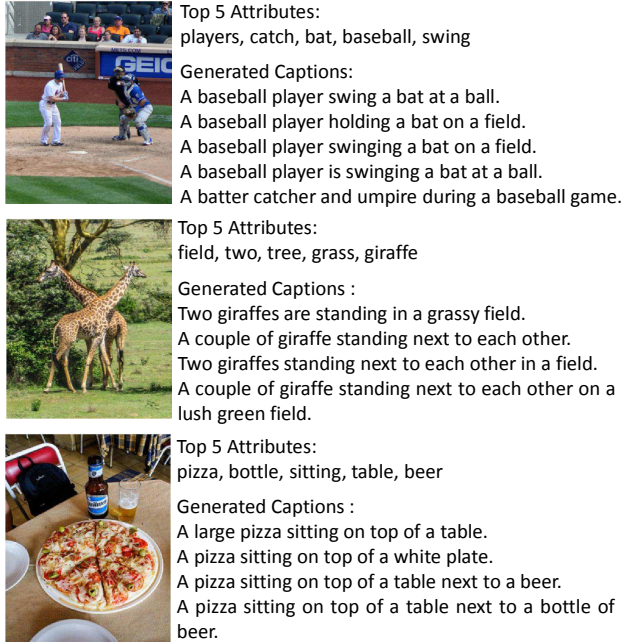A pizza sitting on top of a table next to a bottle of beer.

Figure 3. Examples of predicted attributes and generated captions.

aggregated with max pooling to produce the final prediction over the attribute vocabulary.

To initialize the attribute prediction model, we use the powerful VggNet-16 [25] pre-trained on the ImageNet [8]. The shared CNN is then fine-tuned on the multi-label dataset, the MS COCO image-attribute training data [30]. The output of the last fully-connected layer is fed into a c-way softmax which produces a probability distribution over the c class labels. The c represents the attribute vocabulary size, which here is 256. The fine-tuning learning rates of the last two fully connect layers are initialized to 0.001 and the prediction layer is initialized to 0.01. All the other layers are fixed. We execute 40 epochs in total and decrease the learning rate to one tenth of the current rate for each layer after 10 epochs. The momentum is set to 0.9. The dropout rate is set to 0.5. Then, we use Multiscale Combinatorial Grouping (MCG) [22] for the proposal generation. Finally, a cross hypothesis max-pooling is applied to integrate the outputs into a single prediction vector $V_{att}(I)$.

3.2. Caption-based Image Representation

Currently the most successful approach to image captioning [6, 9, 14, 21, 28] is to attach a CNN to an RNN to learn the mapping from images to sentences directly. Wu *et al.* [30] proposed to feed a high-level attribute-based representation to an LSTM to generate captions, instead of directly using CNN-extracted features. This method produces promising results on the major public captioning challenge [5] and accepts our attributes prediction vector $V_{att}(I)$ as the input. We thus use this approach to generate

5 different captions (using beam search) that constitute the internal textual representation for a given image.

The hidden state vector of the caption-LSTM after it has generated the last word in each caption is used to represent its content. Average-pooling is applied over the 5 hidden-state vectors, to obtain a 512-d vector $V_{cap}(I)$ for the image I. The caption-LSTM is trained on the human-generated captions from the MS COCO training set, which means that the resulting model is focused on the types of image content that humans are most interested in describing. Figure 3 shows some examples of the predicted attributes and generated captions.

3.3. Relating to the Knowledge Base

The external data source that we use here is DBpedia [2]. As a source of general background information, although any such KB could equally be applied, DBpedia is a structured database of information extracted from Wikipedia. The whole DBpedia dataset describes 4.58 million entities, of which 4.22 million are classified in a consistent ontology. The data can be accessed using an SQL-like query language for RDF called SPARQL. Given an image and its predicted attributes, we use the top-5 most strongly predicted attributes[1] to generate DBpedia queries. Inspecting the database shows that the 'comment' field is the most generally informative about an attribute, as it contains a general text description of it. We therefore retrieve the comment text for each query term. The KB+SPARQL combination is very general, however, and could be applied to problem specific KBs, or a database of common sense information, and can even perform basic inference over RDF. Figure 4 shows an example of the query language and returned text.
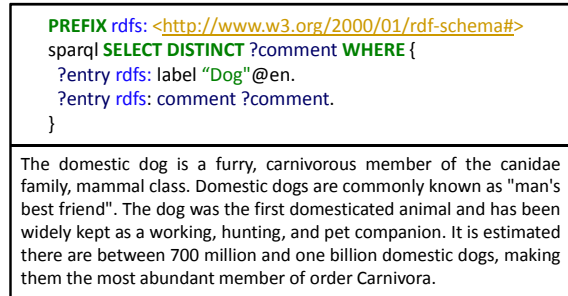
```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
sparql SELECT DISTINCT ?comment WHERE {
  ?entry rdfs: label "Dog"@en.
  ?entry rdfs: comment ?comment.
}
```

The domestic dog is a furry, carnivorous member of the canidae family, mammal class. Domestic dogs are commonly known as "man's best friend". The dog was the first domesticated animal and has been widely kept as a working, hunting, and pet companion. It is estimated there are between 700 million and one billion domestic dogs, making them the most abundant member of order Carnivora.

Figure 4. An example of SPARQL query language for the attribute 'dog'. The mined text-based knowledge are shown below.

Since the text returned by the SPARQL query is typically much longer than the captions generated in the section 3.2, we turn to Doc2Vec [15] to extract the semantic meanings. Doc2Vec, also known as Paragraph Vector, is an unsupervised algorithm that learns fixed-length feature representations from variable-length pieces of texts, such as

---

[1]We only use the top-5 attributes to query the KB because, based on the observation of training data, an image typically contains 5-8 attributes. We also tested with top-10, but no improvements were observed.

sentences, paragraphs, and documents. Le *et al.* [15] proved that it can capture the semantics of paragraphs. A Doc2Vec model is trained to predict words in the document given the context words. We collect 100,000 documents from DB-pedia to train a model with vector size 500. To obtain the knowledge vector $V_{know}(I)$ for image I, we combine the 5 returned paragraphs in to a single large paragraph, before extracting semantic features using our Doc2Vec model.

4. A VQA Model with Multiple Inputs

We propose to train a VQA model by maximizing the probability of the correct answer given the image and question. We want our VQA model to be able to generate multiple word answers, so we formulate the answering process as a word sequence generation procedure. Let $Q = \{q_1, ..., q_n\}$ represents the sequence of words in a question, and $A = \{a_1, ..., a_l\}$ the answer sequence, where n and l are the length of question and answer, respectively. The log-likelihood of the generated answer can be written as:

$$\log p(A|I, Q) = \sum_{t=1}^{l} \log p(a_t|a_{1:t-1}, I, Q) \qquad (1)$$

where $p(a_t|a_{1:t-1}, I, Q)$ is the probability of generating a_t given image information I, question Q and previous words $a_{1:t-1}$. We employ an encoder LSTM [13] to take the semantic information from image I and the question Q, while using a decoder LSTM to generate the answer. Weights are shared between the encoder and decoder LSTM.

In the training phase, the question Q and answer A are concatenated as $\{q_1, ..., q_n, a_1, ..., a_l, a_{l+1}\}$, where a_{l+1} is a special END token. Each word is represented as a one-hot vector of dimension equal to the size of the word dictionary. The training procedure is as follows: at time step $t = 0$, we set the LSTM input:

$$x_{initial} = [W_{ea}V_{att}(I), W_{ec}V_{cap}(I), W_{ek}V_{know}(I)] \quad (2)$$

where W_{ea}, W_{ec}, W_{ek} are learnable embedding weights for the vector representation of attributes, captions and external knowledge, respectively. In practice, all these embedding weights are learned jointly. Given the randomly initialized hidden state, the encoder LSTM feeds forward to produce hidden state h_0 which encodes all of the input information. From $t = 1$ to $t = n$, we set $x_t = W_{es}q_t$ and the hidden state h_{t-1} is given by the previous step, where W_{es} is the learnable word embedding weights. The decoder LSTM runs from time step $n + 1$ to $l + 1$. Specifically, at time step $t = n + 1$, the LSTM layer takes the input $x_{n+1} = W_{es}a_1$ and the hidden state h_n corresponding to the last word of the question, where a_1 is the start word of the answer. The hidden state h_n thus encodes all available information about the image and the question. The probability distribution p_{t+1} over all answer words in the vocabulary is then computed by the LSTM feed-forward process. Finally, for the

final step, when a_{l+1} represents the last word of the answer, the target label is set to the END token.

Our training objective is to learn parameters W_{ea}, W_{ec}, W_{ek}, W_{es} and all the parameters in the LSTM by minimizing the following cost function:

$$\mathcal{C} = -\frac{1}{N}\sum_{i=1}^{N} \log p(A^{(i)}|I, Q) + \lambda_{\boldsymbol{\theta}} \cdot ||\boldsymbol{\theta}||_2^2 \quad (3)$$

$$= -\frac{1}{N}\sum_{i=1}^{N} \sum_{j=1}^{l^{(i)}+1} \log p_j(a_j^{(i)}) + \lambda_{\boldsymbol{\theta}} \cdot ||\boldsymbol{\theta}||_2^2 \quad (4)$$

where N is the number of training examples, and $n^{(i)}$ and $l^{(i)}$ are the length of question and answer respectively for the i-th training example. Let $p_t(a_t^{(i)})$ correspond to the activation of the Softmax layer in the LSTM model for the i-th input and $\boldsymbol{\theta}$ represent the model parameters. Note that $\lambda_{\boldsymbol{\theta}} \cdot ||\boldsymbol{\theta}||_2^2$ is a regularization term.

5. Experiments

We evaluate our model on two recent publicly available visual question answering datasets, both based on MS COCO images. The Toronto COCO-QA Dataset [23] contains 78,736 training and 38,948 testing examples, which are generated from 117,684 images. There are four types of questions, relating to the object, number, color and location, all constructed so as to have a single-word answer. All of the question-answer pairs in this dataset are automatically converted from human-sourced image descriptions. Another benchmarked dataset is VQA [1], which is a much larger dataset and contains 614,163 questions and 6,141,630 answers based on 204,721 MS COCO images. This dataset provides a surprising variety of question types, including "What is...', "How Many" and even "Why...". The ground truth answers were generated by 10 human subjects and can be single word or sentences. The data train/val split follows the COCO official split, which contains 82,783 training images and 40,504 validation images, each has 3 questions and 10 answers. We randomly choose 5000 images from the validation set as our val set, with the remainder testing. The human ground truth answers for the actual VQA test split are not available publicly and only can be evaluated via the VQA evaluation server. Hence, we also apply our final model on a test split and report the overall accuracy. Table 1 displays some dataset statistics. We did not test on the DAQUAR dataset [19] as it is an order of magnitude smaller than the datasets mentioned above, and thus too small to train our system, and to test its generality.

5.1. Implementation Details

To train the VQA model with multiple inputs in the Section 4, we use Stochastic gradient Descent (SGD) with

	Toronto COCO-QA	VQA
# Images	117,684	204,721
# Questions	117,684	614,163
# Question Types	4	more than 20
# Ans per Que	1	10
# Words per Ans	1	1 or multiple

Table 1. Some statistics about the Toronto COCO-QA Dataset [23] and VQA dataset [1].

mini-batches of 100 image-QA pairs. The attributes, internal textual representation, external knowledge embedding size, word embedding size and hidden state size are all 256 in all experiments. The learning rate is set to 0.001 and clip gradients is 5. The dropout rate is set to 0.5.

5.2. Results on Toronto COCO-QA

Metrics Following [17, 23], the accuracy value (the proportion of correctly answered test questions), and the Wu-Palmer similarity (WUPS) [31] are used to measure performance. The WUPS calculates the similarity between two words based on the similarity between their common subsequence in the taxonomy tree. If the similarity between two words is greater than a threshold then the candidate answer is considered to be right. We report on thresholds 0.9 and 0.0, following [17, 23].

Evaluations To illustrate the effectiveness of our model, we provide a baseline and several state-of-the-art results on the Toronto COCO-QA dataset. The **Baseline** method is implemented simply by connecting a CNN to an LSTM. The CNN is a pre-trained (on ImageNet) VggNet model from which we extract the coefficients of the last fully connected layer. We also implement a baseline model **VggNet+ft-LSTM**, which applies a VggNet that has been fine-tuned on the COCO dataset, based on the task of image-attributes classification. We also present results from a series of cut down versions of our approach for comparison. **Att-LSTM** uses only the semantic level attribute representation V_{att} as the LSTM input. To evaluate the

Toronto COCO-QA	Acc(%)	WUPS@0.9	WUPS@0.0
GUESS[23]	6.65	17.42	73.44
VIS+BOW[23]	55.92	66.78	88.99
VIS+LSTM[23]	53.31	63.91	88.25
2-VIS+BLSTM[23]	55.09	65.34	88.64
Ma et al.[17]	54.94	65.36	88.58
Baseline			
VggNet-LSTM	50.73	60.37	87.48
VggNet+ft-LSTM	58.34	67.32	89.13
Our-Proposal			
Att-LSTM	61.38	71.15	91.58
Att+Cap-LSTM	69.02	76.20	92.38
Att+Know-LSTM	63.07	72.22	90.84
Cap+Know-LSTM	64.31	73.31	90.01
Att+Cap+Know-LSTM	**69.73**	**77.14**	**92.50**

Table 2. Accuracy, WUPS metrics compared to other state-of-the-art methods and our baseline on Toronto COCO-QA dataset.

COCO-QA	Object	Number	Color	Location
GUESS	2.11	35.84	13.87	8.93
VIS+BOW	58.66	44.10	51.96	49.39
VIS+LSTM	56.53	46.10	45.87	45.52
2-VIS+BLSTM	58.17	44.79	49.53	47.34
Baseline				
VggNet-LSTM	53.71	45.37	36.23	46.37
VggNet+ft-LSTM	61.67	50.04	52.16	54.40
Our-Proposal				
Att-LSTM	63.92	51.83	57.29	54.84
Att+Cap-LSTM	71.30	69.98	61.50	**60.98**
Att+Know-LSTM	64.57	54.37	62.79	56.98
Cap+Know-LSTM	65.61	55.13	62.02	57.28
Att+Cap+Know-LSTM	**71.45**	**75.33**	**64.09**	60.98

Table 3. Toronto COCO-QA accuracy (%) per category.

contribution of the internal textual representation and external knowledge for the question answering, we feed the image caption representation V_{cap} and knowledge representation V_{know} with the V_{att} separately, producing two models, **Att+Cap-LSTM** and **Att+Know-LSTM**. We also tested the **Cap+Know-LSTM**, for the experiment completeness. Our final model is the **Att+Cap+Know-LSTM**, which combines all the available information.

GUESS [23] simply selects the modal answer from the training set for each of 4 question types (the modal answers are 'cat', 'two', 'white', and 'room'). **VIS+BOW** [23] performs multinomial logistic regression based on image features and a BOW vector obtained by summing all the word vectors of the question. **VIS+LSTM** [23] has one LSTM to encode the image and question, while **2-VIS+BLSTM** [23] has two image feature inputs, at the start and the end. Ma *et al.* [17] encode both images and questions with a CNN.

Table 2 reports the results. All of our proposed models outperform the **Baseline** and all of the competing state-of-the-art methods. Our final model **Att+Cap+Know-LSTM** achieves the best results. It surpasses the baseline by nearly 20% and outperforms the previous state-of-the-art methods by around 15%. **Att+Cap-LSTM** clearly improves the results over the **Att-LSTM** model. This proves that internal textual representation plays a significant role in the VQA task. The **Att+Know-LSTM** model does not perform as well as **Att+Cap-LSTM** , which suggests that the information extracted from captions is more valuable than that extracted from the KB. **Cap+Know-LSTM** also performs better than **Att+Know-LSTM**. This is not surprising because the Toronto COCO-QA questions were generated automatically from the MS COCO captions, and thus the fact that they can be answered by training on the captions is to be expected. This generation process also leads to questions which require little external information to answer. The comparison on the Toronto COCO-QA thus provides an important benchmark against related methods, but does not really test the ability of our method to incorporate extra information. It is thus interesting that the additional external information provides any benefit at all.

Question Type	Our-Baseline	Our Proposal				
	VggNet + LSTM	Att + LSTM	Att+Cap + LSTM	Att+Know + LSTM	Cap+Know + LSTM	A+C+K + LSTM
what is	21.41	34.63	42.21	37.11	35.58	**42.52**
what colour	29.96	39.07	48.65	39.68	40.62	**48.86**
what kind	24.15	41.22	47.93	46.16	44.04	**48.05**
what are	23.05	38.87	47.13	41.13	39.73	**47.21**
what type	26.36	41.71	47.98	44.91	44.95	**48.11**
is the	71.49	73.22	74.63	74.40	73.78	**74.70**
is this	73.00	75.26	76.08	**76.56**	74.18	76.14
how many	34.42	39.14	46.61	39.78	44.20	**47.38**
are	73.51	75.14	76.01	75.75	75.78	**76.14**
does	76.51	76.71	78.07	76.55	77.17	**78.11**
where	10.54	21.42	25.92	24.13	16.09	**26.00**
is there	86.66	**87.10**	86.82	85.87	85.26	87.01
why	3.04	7.77	9.63	11.88	9.99	**13.53**
which	31.28	36.60	**39.55**	37.71	37.86	38.70
do	76.44	75.76	78.18	75.25	74.91	**78.42**
what does	15.45	19.33	21.80	19.50	19.04	**22.16**
what time	13.11	15.34	15.44	**15.47**	15.04	15.34
who	17.07	22.56	25.71	21.23	22.86	**25.74**
what sport	65.65	91.02	93.96	90.86	91.75	**94.20**
what animal	27.77	61.39	70.65	63.91	63.26	**71.70**
what brand	26.73	32.25	33.78	32.44	31.30	**34.60**
others	44.37	50.23	53.29	52.11	51.20	**53.45**
Overall	44.93	51.60	55.04	53.79	52.31	**55.96**

Table 4. Results on the open-answer task for various question types on VQA validation set. All results are in terms of the evaluation metric from the VQA evaluation tools. The overall accuracy for the model of **VggNet+ft+LSTM** is 50.01. Detailed results of different question types for this model are not shown in the table due to the limited space.

Table 3 shows the per-category accuracy for different models. Surprisingly, the counting ability (see question type 'Number') increases when both captions and external knowledge are included. This may be because some 'counting' questions are not framed in terms of the labels used in the MS COCO captions. Ren *et al*. also observed similar cases. In [23] they mentioned that "there was some observable counting ability in very clean images with a single object type but the ability was fairly weak when different object types are present". We also find there is a slight increase for the 'color' questions when the KB is used. Indeed, some questions like 'What is the color of the stop sign?' can be answered directly from the KB, without the visual cue.

5.3. Results on the VQA

Antol *et al*. in [1] provide the VQA dataset which is intended to support "free-form and open-ended Visual Question Answering". They also provide a metric for measuring performance: $\min\{\frac{\#\text{ humans that said answer}}{3}, 1\}$ thus 100% means that at least 3 of the 10 humans who answered the question gave the same answer. We have used the provided evaluation code to produce the results in Table 4.

Evaluation Inspecting Table 4, results on the VQA validation set, we see that the attribute-based **Att-LSTM** is a significant improvement over our **VggNet+LSTM** baseline. We also evaluate another baseline, the **VggNet+ft+LSTM**, which uses the penultimate layer of the attributes prediction CNN as the input to the LSTM. Its overall accuracy on the VQA is 50.01, which is still lower than our proposed mod-

els. Adding either image captions or external knowledge further improves the result. The model **Cap+Know** produces overall accuracy 52.31, slightly lower than Att+Know (53.79). This suggests that the attributes representation plays a more important role here. Our final model **A+C+K-LSTM** produces the best results, outperforming the baseline **VggNet-LSTM** by 11% overall.

Figure 5 relates the performance of the various models on five categories of questions. The 'object' category is the average of the accuracy of question types starting with 'what kind/type/sport/animal/brand...', while the 'number' and 'color' category corresponds to the question type 'how many' and 'what color'. The performance comparison across categories is of particular interest here because answering different classes of questions requires different amounts of external knowledge. The 'Where' questions, for instance, require knowledge of potential locations, and 'Why' questions typically require general knowledge about people's motivation. 'Number' and 'Color' questions, in contrast, can be answered directly. The results show that for 'Why' questions, adding the KB improves performance by more than 50% (Att-LSTM achieves 7.77% while Att+Know-LSTM achieves 11.88%), and that the combined A+C+K-LSTM achieves 13.53%.
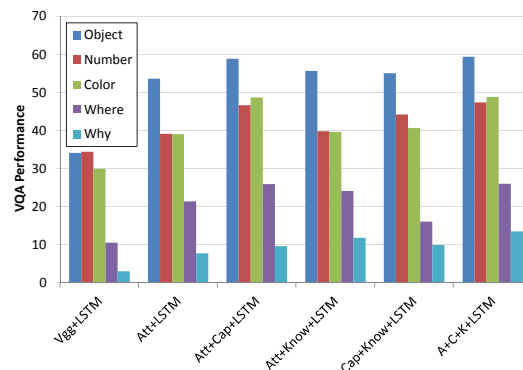

Figure 5. Performance on five question categories for different models. The 'Object' category is the average accuracy of question types starting with 'what kind/type/sport/animal/brand...'.

Table 5 provides some indicative results, more results can be found in the supplementary material, including failure cases.

We have also tested on the VQA test-dev and test-standard[2] consisting of 60,864 and 244,302 questions (for which ground truth answers are not published) using our final A+C+K-LSTM model, and evaluated them on the VQA evaluation server. Table 6 shows the server reported results.

Antol *et al*. [1] provide several results for this dataset. In each case they encode the image with the final hidden layer from VggNet, and questions are encoded using a BOW representation. A softmax neural network classifier with 2 hid-

---

[2]http://www.visualqa.org/challenge.html

	What color is the tablecloth?	How many people in the photo?	What is the red fruit?	What are these people doing?
Ours:	white	2	apple	eating
Vgg+LSTM:	red	1	banana	playing
Ground Truth:	white	2	apple	eating

	Why are his hands outstretched?	Why are the zebras in water?	Is the dog standing or laying down?	Which sport is this?
Ours:	balance	drinking	laying down	baseball
Vgg+LSTM:	play	water	sitting	tennis
Ground Truth:	balance	drinking	laying down	baseball

Table 5. Some example cases where our final model gives the correct answer while the base line model **VggNet-LSTM** generates the wrong answer. All results are from VQA. More results can be found in the supplementary material.

	Test-dev				Test-standard			
	All	Y/N	Num	Others	All	Y/N	Num	Others
Question [1]	40.09	75.66	36.70	27.14	-	-	-	-
Image [1]	28.13	64.01	0.42	3.77	-	-	-	-
Q+I [1]	52.64	75.55	33.67	37.37	-	-	-	-
LSTM Q [1]	48.76	78.20	35.68	26.59	48.89	78.12	34.94	26.99
LSTM Q+I [1]	53.74	78.94	35.24	36.42	54.06	79.01	35.55	36.80
Human [1]	-	-	-	-	83.30	95.77	83.39	72.67
Ours	**59.17**	**81.01**	**38.42**	**45.23**	**59.44**	**81.07**	**37.12**	**45.83**

Table 6. VQA Open-Ended evaluation server results for our method. Accuracies for different answer types and overall performances on test-dev and test-standard datasets are shown.

den layers and 1000 hidden units (dropout 0.5) in each layer with tanh non-linearity is then trained, the output space of which is the 1000 most frequent answers in the training set. They also provide an LSTM model followed by a softmax layer to generate the answer. Two versions of this approach are used, one which is given both the question and the image, and one which is given only the question (see [1] for details). Our final model outperforms the listed approaches according to the overall accuracy and all answer types.

6. Conclusion

Open-ended visual question answering is an interesting challenge for computer vision, not least because it represents a move away from purely image-based analysis and towards a broader form of artificial visual intelligence. In this paper we have shown that it is possible to extend the state-of-the-art RNN-based VQA approach so as to incorporate the large volumes of information required to answer general, open-ended, questions about images. The knowledge bases which are currently available do not contain much of the information which would be beneficial to this process, but none-the-less can still be used to significantly improve performance on questions requiring external knowledge (such as 'Why' questions). The approach that we propose is very general, however, and will be applicable to more informative knowledge bases should they become available. At the time of writing this paper, our system performs the best on two large-scale VQA datasets and produces promising results on the VQA evaluation server. Further work includes generating knowledge-base queries which reflect the content of the question and the image, in order to extract more specifically related information.

It seems inevitable that successful general visual question answering will demand access to a large external knowledge base. We have shown that this is possible using the LSTM-based approach, and that it improves performance. We hope this might lead the way to a visual question answering approach which is capable of deeper analysis of image content, and even common sense.

References

[1] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. VQA: Visual Question Answering. In *Proc. IEEE Int. Conf. Comp. Vis.*, 2015. 1, 2, 5, 6, 7, 8

[2] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. *Dbpedia: A nucleus for a web of open data.* Springer, 2007. 2, 3, 4

[3] J. Berant, A. Chou, R. Frostig, and P. Liang. Semantic Parsing on Freebase from Question-Answer Pairs. In *Proc. Conf. Empirical Methods in Natural Language Processing*, pages 1533–1544, 2013. 2

[4] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. ACM, 2008. 2

[5] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollar, and C. L. Zitnick. Microsoft COCO captions: Data collection and evaluation server. *arXiv:1504.00325*, 2015. 3, 4

[6] X. Chen and C. L. Zitnick. Learning a Recurrent Visual Representation for Image Caption Generation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2015. 4

[7] K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Proc. Conf. Empirical Methods in Natural Language Processing*, 2014. 1

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2009. 4

[9] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2015. 1, 4

[10] D. Ferrucci, E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A. A. Kalyanpur, A. Lally, J. W. Murdock, E. Nyberg, J. Prager, et al. Building Watson: An overview of the DeepQA project. *AI magazine*, 31(3):59–79, 2010. 2

[11] H. Gao, J. Mao, J. Zhou, Z. Huang, L. Wang, and W. Xu. Are You Talking to a Machine? Dataset and Methods for Multilingual Image Question Answering. In *Proc. Advances in Neural Inf. Process. Syst.*, 2015. 2

[12] D. Geman, S. Geman, N. Hallonquist, and L. Younes. Visual Turing test for computer vision systems. *Proceedings of the National Academy of Sciences*, 112(12):3618–3623, 2015. 2

[13] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. 5

[14] A. Karpathy, A. Joulin, and F. F. Li. Deep fragment embeddings for bidirectional image sentence mapping. In *Proc. Advances in Neural Inf. Process. Syst.*, 2014. 1, 4

[15] Q. V. Le and T. Mikolov. Distributed representations of sentences and documents. *arXiv preprint arXiv:1405.4053*, 2014. 2, 4, 5

[16] X. Lin and D. Parikh. Don't Just Listen, Use Your Imagination: Leveraging Visual Common Sense for Non-Visual Tasks. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, June 2015. 3

[17] L. Ma, Z. Lu, and H. Li. Learning to Answer Questions From Image using Convolutional Neural Network. *arXiv:1506.00333*, 2015. 2, 6

[18] M. Malinowski and M. Fritz. A multi-world approach to question answering about real-world scenes based on uncertain input. In *Proc. Advances in Neural Inf. Process. Syst.*, pages 1682–1690, 2014. 2

[19] M. Malinowski and M. Fritz. Towards a Visual Turing Challenge. *arXiv:1410.8027*, 2014. 5

[20] M. Malinowski, M. Rohrbach, and M. Fritz. Ask Your Neurons: A Neural-based Approach to Answering Questions about Images. In *Proc. IEEE Int. Conf. Comp. Vis.*, 2015. 2

[21] J. Mao, W. Xu, Y. Yang, J. Wang, and A. Yuille. Deep Captioning with Multimodal Recurrent Neural Networks (m-RNN). In *Proc. Int. Conf. Learn. Representations*, 2015. 1, 4

[22] J. Pont-Tuset, P. Arbeláez, J. Barron, F. Marques, and J. Malik. Multiscale Combinatorial Grouping for Image Segmentation and Object Proposal Generation. In *arXiv:1503.00848*, March 2015. 4

[23] M. Ren, R. Kiros, and R. Zemel. Image Question Answering: A Visual Semantic Embedding Model and a New Dataset. In *Proc. Advances in Neural Inf. Process. Syst.*, 2015. 1, 2, 5, 6, 7

[24] F. Sadeghi, S. K. Kumar Divvala, and A. Farhadi. VisKE: Visual Knowledge Extraction and Question Answering by Visual Verification of Relation Phrases. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, June 2015. 3

[25] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2014. 4

[26] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Proc. Advances in Neural Inf. Process. Syst.*, 2014. 1

[27] K. Tu, M. Meng, M. W. Lee, T. E. Choe, and S.-C. Zhu. Joint video and text parsing for understanding events and answering queries. *IEEE Trans. Multimedia*, 21(2):42–70, 2014. 2

[28] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2014. 1, 4

[29] Y. Wei, W. Xia, J. Huang, B. Ni, J. Dong, Y. Zhao, and S. Yan. CNN: Single-label to multi-label. *arXiv:1406.5726*, 2014. 3

[30] Q. Wu, C. Shen, A. v. d. Hengel, L. Liu, and A. Dick. What Value Do Explicit High Level Concepts Have in Vision to Language Problems? In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2016. 2, 4

[31] Z. Wu and M. Palmer. Verbs semantics and lexical selection. In *Proc. Conf. Association for Computational Linguistics*, 1994. 6

[32] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville. Describing videos by exploiting temporal structure. *arXiv:1502.08029*, 2015. 1

[33] Y. Zhu, C. Zhang, C. Ré, and L. Fei-Fei. Building a Large-scale Multimodal Knowledge Base for Visual Question Answering. *arXiv:1507.05670*, 2015. 2, 3