Revista Romana de Interactiune Om-Calculator 11 (1) 2018, 1-27

Combining Visual and Textual Attention in Neural Models for Enhanced Visual Question Answering

Cosmin Dragomir, Cristian Ojog, Traian Rebedea

University Politehnica of Bucharest
Splaiul Independentei nr. 313, sector 6, 060042, Bucuresti
E-mail: cosmin.gabriel.dragomir@gmail.com, crisojog@gmail.com, traian.rebedea@cs.pub.ro

Abstract. While visual information is essential for humans as it models our environment, language is our main method of communication and reasoning. Moreover, these two human capabilities interact in complex ways, therefore problems involving both visual and natural language data became widely explored in recent years. Thus, visual question answering aims at building systems able to process questions expressed in natural language about images or even videos. This would significantly ease the quality of life for visually impaired people by allowing them to get real-time answers about their surroundings. Unfortunately, the relations between images and questions are complex and the current solutions that exploit recent advanced in deep learning for text and image representation are not reliable enough. To improve these results, the visual and text representations must be fused into the same multimodal space. In this paper we present two different solutions for solving this problem. The first performs reasoning on the image by using soft attention mechanisms computed given the question. The second uses soft attention not just on the image, but the text as well. Although our models are more lightweight than state of the art solutions for this task, we achieve near top performance with the proposed combination of visual and textual representations.

Keywords: visual question answering, deep learning, natural language processing, computer vision, visual impaired users.

1. Introduction

With the advances in both Natural Language Processing (NLP) and Computer Vision (CV) attributed mainly to deep learning models, there has been a recent trend to tackle problems that require methods and techniques from both domains. One of the most promising such tasks has been question answering based on images and videos. The most successful image-based question answering task has been the Visual Question Answering (VQA) challenge introduced by Antol et al. (2015). This is indeed a multi-

disciplinary research problem that combines not only CV and NLP, but also Knowledge Representation and Reasoning and other Artificial Intelligence subdomains. The authors consider this challenge more complex than other CV-NLP joint problems, such as image captioning that only requires a scene-level understanding of an image paired with word n-gram statistics.

The challenge consists of four tracks, all of which require fine-grained recognition, activity recognition, knowledge base reasoning and commonsense reasoning. All four tracks consist of building a model that receives as input an image and a question based on that image and that has to output an answer or choose one out of a list of possible answers. The first two tracks contain open-ended questions on real and abstract images. The difference between those is that the abstract images were generated and require more precise attention over the spatial relations in the image. The other 2 tracks contain the same questions, but offer multiple-choice answers. In all 4 cases, there are 3 questions per image, with 10 possible answers, unknown at input time, out of which only one needs to be computed by the model, for the open-ended tracks, and 18 possible answers, known beforehand, out of which only one is correct and needs to be computed by the model, for the multiple-choice tracks. In this research, we have only considered the real open-ended track as it has been shown that the models designed for the open-ended track perform very well, with minor tweaks, for the multiple-choice track as well (Fukui et al., 2016; Lu et al., 2016).

VQA is also relevant for Human-Computer Interaction in several ways. First, such a solution can be used for various real-life situations, such as providing accessibility to visual information for visually impaired users or, further down the road, being integrated in personal assistants that can extract information from images based on the user's questions, such as asking an assistant where you were in a particular picture. Solving this task for static images can also be a first step in solving question answering based on video input, as each frame can be considered as a still image, while still needing to solve the problem of capturing the context of the previous frames.

Given the fact that the human baseline for the proposed dataset is surprisingly low (at 83.30%), there is also a potential at some point in time that computers might be able to provide answers to questions about visual context with an accuracy that surpasses humans. The human baseline for the VQA v1 dataset also proves that the questions themselves are particularly difficult and that some images are genuinely hard to interpret.

In this paper we propose two neural models with attention that improve the results of existing baselines for VQA. We show that the attention is an important mechanism for solving this problem efficiently and that combining visual and textual attention offers better results. Although there are several other more complex models combing visual and textual attention, the proposed methods are original, lightweight (with a smaller number of parameters) and competitive with existing state of the art.

The paper continues with a section describing the VQA dataset and challenge, together with existing methods for solving this task. Then we continue by introducing the two neural models with attention proposed for solving the problem. In Section 4 we present the results obtained by the proposed models together with a detailed discussion about the advantages of the attention mechanism. The paper ends with conclusions and future work.

2. Related work

In this section we describe in more detail the VQA v1 dataset and also introduce some of the techniques necessary for solving this task. Afterwards, we present some of the current state-of-the-art models that solve the VQA task including several models with attention.

2.1. Neural models for question representation

Models developed for VQA make use of recent advances in NLP and deep learning for word and sentence representation, mainly distributed word embeddings and Recurrent Neural Networks (RNNs).

Word embeddings

Vectorial representations of words are widely used in various NLP applications, such as question answering models, machine translation and information retrieval. Distributed word embeddings improve over basic bag of words models using tf-idf or other one-hot encodings for words. Two of the most used distributed word embeddings models are word2vec (Mikolov et al., 2013) and GloVe (Pennington, Socher and Manning, 2014).

Word2vec (Mikolov et al., 2013) proposed two different strategies for computing word embeddings: a Continuous Bag-of-Words model, which

takes multiple words from the past and future in order to predict the current word, and the Continuous Skip-gram model, which does the opposite, taking the current word and predicting the surrounding words. GloVe (Pennington, Socher and Manning, 2014) is another unsupervised learning algorithm, which can be used to get vector representations for words. As opposed to word2vec, though, GloVe is a log-bilinear model with a weighted least-squares objective, based on the intuition that ratios of word-word co-occurrence probabilities have the potential for encoding meaning. By log-bilinear model, we mean that the training objective is learning word vectors whose dot product equals the logarithm of the words' probability of co-occurrence.

As in our research we obtained better results with GloVe embeddings than with word2vec ones, we have only reported the results obtained with GloVe word embeddings. The embedding model was pre-trained on the Google News dataset, which consists of approximately 100 billion words. The model produces 300-dimensional vectors for the most common 1 million words and phrases in the dataset. An advantage of the skip-gram model over the other approaches that were considered is that this model allows the training of high-dimensional vector words on large amounts of data. This allows for querying much more subtle semantic relationships between words, such as finding the country in which a city is located, based on word analogies (Mikolov et al., 2013).

Modeling questions using Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are a type of neural networks specialized in processing sequences of data of varying lengths. Its main use-case is to provide persistence between several inputs, which is something that regular neural networks cannot do. The underlying idea in the case of text interpretation is to treat each new word with respect to all the previous words in the sequence.

RNNs suffer from a couple of issues, and cannot deliver on its promise of connecting previous information to the current task. The problem with learning long-term dependencies is that gradients propagated over many nodes in the past tend to either vanish or explode. While the latter can be solved with little to no elegance by simply clipping the gradients, the former does not have a simple solution. The intuition is that we do not always need to look to every single word shown in the past, as many of them might not

have any significance in interpreting the current word. Therefore, the main contribution of Long Short-Term Memory Networks (LSTMs) (Hochreiter and Schmidhuber, 1997) is the introducing of self-loops that produce paths where the gradient can flow for long durations, with a learn-able weight conditioned on the current context. Therefore the self-loop is gated and the network can decide when it no longer needs to keep the context in memory and can remove it, partially or totally.

Another way of using LSTMs for text representation is presented in Karpathy and Fei-Fei (2015). They use two LSTMs, one interpreting the text from the first to last word, and one interpreting the text the other way around, called BRNN (or BLSTM, BiLSTM), is said to provide context from both directions to a word embedding, also helping to create concepts from groupings of nearby words.

2.2. Neural models for image representation

The representation of the input images are constructed using deep convolutional neural networks (CNNs) pre-trained on the ImageNet classification task (Krizhevsky, Sutskever and Hinton, 2012). In order to reduce training time, we opted to pre-extract the image embeddings using the models described in this section and then to use these embeddings as input for the proposed VQA models.

The ResNet model (He et al., 2016) has been widely used by the solutions developed for the VQA task (Lu et al., 2016; Nam, Ha and Kim, 2016; Kazemi et al., 2017). This is a deeper convolutional neural network that is based on the principle of residual learning. A large problem of deep models is that the deeper they get the harder it is to backpropagate gradients that can significantly steer the weights of the first layers in the optimal direction. To tackle this issue, He et al. (2016) introduce the idea of residual blocks in which the input is also added at the end of several convolution layers forming a block that can ensure unaltered the propagation of the gradient backwards through the deep network. We have used two popular variants of the ResNet model, namely ResNet-50 and ResNet-152 (see Figure 1), pre-trained on the ImageNet dataset. There are several ways of getting the image representation from a ResNet model. In the case of a single embedding per image, we can use the output of the last average pooling layer, cutting just before the final classification Fully Connected (FC) layer, which gives a 2048-dimensional

embedding per image.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		[3×3, 64; 3×3, 64] ×2	[3×3, 64; 3×3, 64] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3
conv3_x	28×28	[3×3, 128; 3×3, 128] ×2	[3×3, 128; 3×3, 128] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×8
conv4_x	14×14	[3×3, 256; 3×3, 256] ×2	[3×3, 256; 3×3, 256] ×6	[1×1, 256; 3×3, 256; 1×1, 1024] ×6	[1×1, 256; 3×3, 256; 1×1, 1024] ×23	[1×1, 256; 3×3, 256; 1×1, 1024] ×36
conv5_x	7×7	[3×3, 512; 3×3, 512] ×2	[3×3, 512; 3×3, 512] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figure 1. ResNet neural architecture used for image classification (He et al., 2016)

2.3. VQA dataset

The VQA dataset (version 1) consists of the MS COCO dataset (Lin et al., 2014) with approximately 200k images, 600k questions, and 6 million ground truth answers. To this real-life images dataset, other abstract scenes were added for the abstract image tracks, and these contain 50k scenes, with 150k questions, and 1.5 million ground truth answers. The answers are generally one word long (over 89% for both real and abstract images).

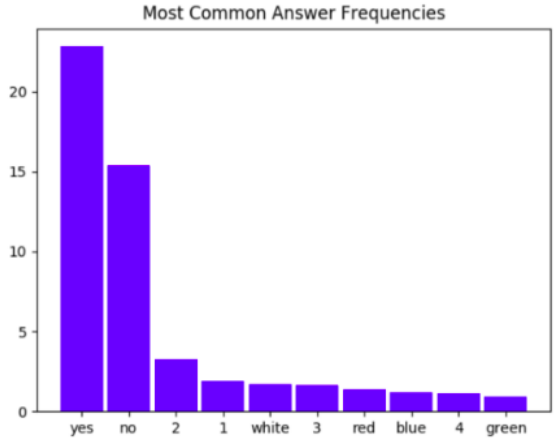

Figure 2. Histogram of the top 10 answers' frequencies

From Figure 2 we can infer that we can filter out some of the unpopular answers in order to avoid overfitting and simplify the learning for the models. Thus, most previous solutions transform the question-answering problem into a classification problem by training models to only output one of the most common 1000 or 3000 answers.

The real open-ended dataset consists of 123,287 training and validation images and 81,434 test images, which contain multiple objects and rich textual information. The questions are extremely varied, requiring either low-level knowledge, such as asking for colors of objects or agents in images, or commonsense knowledge such as asking how many people fit in a bus that is shown in the image, while also making sure that the questions cannot be answered correctly with commonsense knowledge alone. The open-ended answers can be clustered in multiple categories.

Many questions require a simple answer, but given that some answers are not obvious, the dataset provides 10 acceptable answers per question, which are not necessarily different from each other. These answers are gathered from ten separate workers on Amazon Mechanical Turk, and the correctness of an answer is verified using (1) by counting how many people offered a particular answer:

$$accuracy = min(\frac{\#\ humans\ that\ provided\ that\ answer}{3}, 1) \quad (1)$$

This means that an answer is considered 100% accurate if at least three out of ten people agree with it. Before any comparison, the answers are transformed to lowercase, numbers are converted to digits and punctuation is removed.

The questions can be binned into several types considering the first four words. The great variety of these question types also helps emphasize the complexity of this challenge. The majority of the questions are 5-6 words long, although some do reach lengths of up to 26 words.

Over 23k (unique) answers are one word long, comprising 89.32% of the total number of answers. 6.91% of the answers are two words long, while 2.74% of them are three words long. The short length of these answers can be attributed to the fact that the questions generally ask for specific facts or pieces of information from the image, and are not conversational in nature,

as opposed to the labels usually generated in image captioning tasks, where a generative model is almost mandatory.

Out of those answers, 38.37% of them are either "yes" or "no", with a strong bias towards "yes", with 58.85% out of the two. A second main type of answers is numerical, namely the answers two questions starting with "How many...", comprising 12.31% of the total. Out of those, the most popular numerical answer is "2", making up 26.04% of these answers.

2.4. Existing solutions for VQA

All competitive models proposed for visual question answering need to combine the textual (extracted from the question) and visual (extracted from the image) embeddings. The main difficulty is to align specific features from the image (or parts of it) with features from the question (or parts of it). The best performing VQA models use complex methods for achieving this alignment by employing attention mechanisms (Xu et al., 2015) both for the image and the text.

The winning solution for the VQA 1.0 challenge was developed by Fukui et al. (2016). Their main focus is to find a method for combining image and word embeddings in a manner in which all elements can interact, while maintaining a multiplicative interaction. This means that a general embedding of a sentence or an image would lose information about particular items in each one, and therefore would not be of much use to observe the interaction between those items. Therefore, by using a multiplicative interaction, they ensure that they can find similarities between image segments and words representing the same entity. While concatenation of the general embeddings solves the first problem, that of all elements being able to interact when classifying the answer, it does not allow a multiplicative interaction between features. The reverse happens with an element-wise multiplication of the embeddings.

The solution to provide interaction between all elements and of a multiplicative nature is an outer product / bilinear pooling. The problem in doing so with embeddings of size 2048 for both image and text is that you would produce too many activations and too many parameters to learn, and thus would not be feasible in practice. For this the authors propose to use a Compact Bilinear Pooling operation (Gao et al., 2016) that lowers the number of parameters significantly from about 4 million to 16k.

They further improve upon this model by adding an attention mechanism

(see Figure 3). By combining embeddings from sections of the input image with the tiled embedding of the input question using the same MCB Pooling operation mentioned before, as well as running the result through a series of Convolution, Relu and Softmax layers, as well as a weighted sum with the original image embedding, they are able to obtain a soft attention of the image in relation to the question, which is then combined again with a full embedding from the question using MCB Pooling and ran through Fully Connected and Softmax layers to obtain a classification of the 3000 most common answers. While this model was initially trained on the VQA 1.0 dataset, the final results reported in the competition were after training on the v1.0 dataset augmented with several others, in order to provide more examples for training. With the augmented dataset, this model achieves an accuracy of 66.9%, while the model trained on the v1.0 dataset alone has an accuracy of only 64.2%.

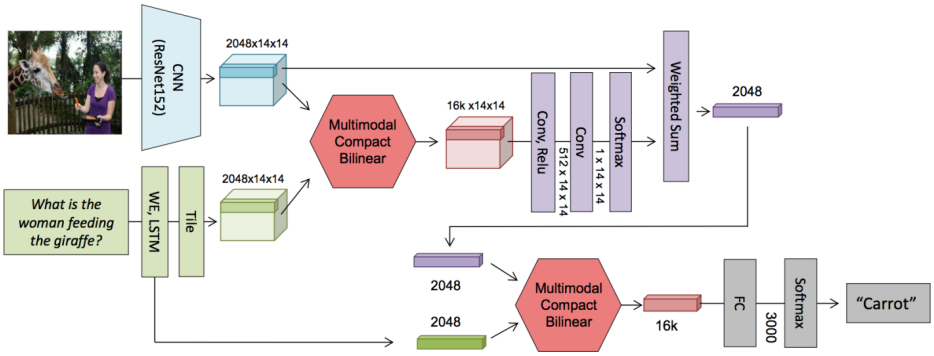

Figure 3. Multimodal Compact Bilinear Pooling model with attention for visual question answering (Gao et al., 2016)

In a different approach, Nam, Ha and Kim (2016) propose a Dual Attention Network, that uses both textual and image embeddings to produce attention maps on each other, which are then concatenated to produce a final answer. In addition to this, they show that there is a potential increase in accuracy if this attention block is thought of as a recurrent unit. This way, the first image attention can focus on, say, a general idea of the subject of a question, while the second image attention can focus on an action described in the same question.

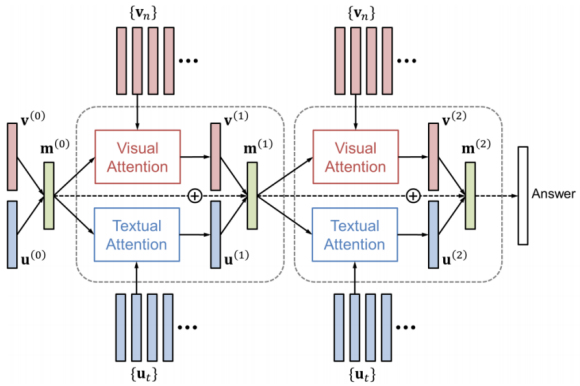

Figure 4. Dual Attention Network with 2 recurrent blocks (Nam, Ha and Kim, 2016)

This model uses a BLSTM to embed the textual input and ResNet152 to embed the image. The attention mechanism consists of a soft attention produced by the point-by-point multiplication of the outputs of two fully-connected layers, each ran through the visual and textual embeddings, respectively. This result is then ran through a softmax activation and then multiplied (via regular matrix multiplication) with the image input, or the textual input, to obtain the image attention, or the text attention, respectively. The model achieves an accuracy of 64.3% on the test-dev set and a 69.0% accuracy on the test-standard set.

3. Proposed models

In this section we will present a baseline model as well as two attention models, which are capable of identifying subjects of interest in the input image based on the given question.

3.1. Baseline model

The baseline model is similar to the "deeper LSTM Q + norm I" presented in (Antol et al., 2015). This was a good starting point to figure out the importance of each element of a network that can get a respectable result on the VQA challenge. At its simplest, the model tries to take textual and image embeddings and project them into a similar vectorial space, such that by multiplying the embeddings, the result will have high activations where similar features meet from both the input embeddings. The reasoning for

attempting such a model is presented in (Jabri et al., 2016), where it is stated that despite the popularity of attention networks or memory networks, there is still significant value in these baseline methods, which come very close in terms of accuracy and are less computationally intensive. We have explored several variations of this model, changing the image embedding model, the LSTM hidden size, using a CNN textual embedding model instead of the LSTM, varying the size of the output classification, and trying various methods of merging the visual and textual embeddings.

3.2. Attention models

For this type of model, instead of obtaining a single image embedding we use an earlier cut of the VGG or ResNet models, that produces a feature map that can be represented as a set of embeddings for a grid of segments over the image. There are several approaches for attention models, depending on the type of attention that we want to obtain. For instance, we could use the textual embeddings in combination with the feature map for the image in order to determine which image segments are important for answering the question, or we could use an overall image embedding to determine which words in the question are the most important to search for in the image. First we will present a Visual attention model that tries to learn an intermediary image heatmap focused on what in the image might be useful to answer the question. Then we will present a Dual attention model that uses both ideas in parallel, and merges the image combined with the attention maps and text embeddings to produce an answer.

Visual attention model

As we have observed in the related work that attention models perform better due to their reasoning capabilities, this subsection presents our own version of a stacked attention model. The high-level design of the model is presented in Figure 5. We will describe the model by splitting it into five components: the image embedding component, the question embedding component, the first attention layer, the second attention layer and lastly the two FC layers used for classification.

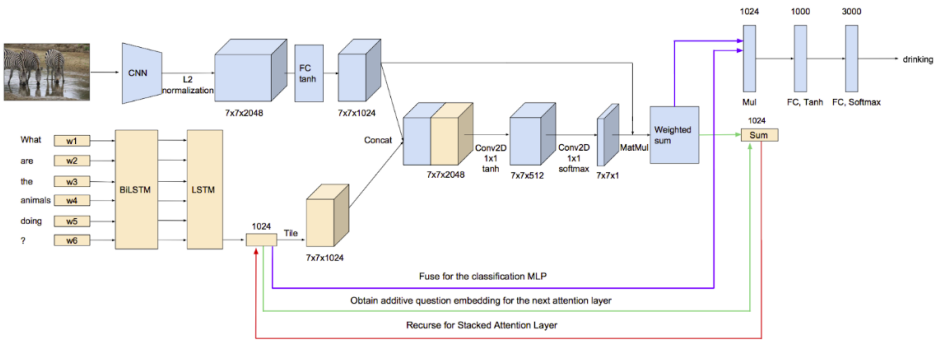

Figure 5. Attention model with stacked attention layers. The question representation is shown in yellow and the image and the fused components representation for the MLP are displayed in light blue. The red line shows that the question embedding is changed for the second attention layer, whereas the green lines highlight the components used to obtain it. Lastly, the blue lines distinguish the components that are used as input for the classification MLP.

The image embedding component uses a pre-trained ResNet152 model, φ, to obtain a tridimensional representation of the input, dividing the input image into a 7x7 grid. The features of each grid, $I_{i,j}$, are normalized and then each set of 2048 features is reduced to 1024 features through a FC layer with a tanh nonlinearity. By doing this, they are moved into a more similar space to the question embedding. After this step, the image is reduced to a cube of size 7x7x1024, img_c, and this is its final vectorial representation.

$$img_{emb} = \phi(image) \tag{2}$$

$$img_emb = \begin{bmatrix} I_{1,1} & I_{1,2} & \dots & I_{1,7} \\ I_{2,1} & I_{2,2} & \dots & I_{2,7} \\ I_{3,1} & I_{3,2} & \dots & I_{3,7} \\ \dots & & & \\ I_{7,1} & I_{7,2} & \dots & I_{7,7} \end{bmatrix} \tag{3}$$

$$I_{i,j-norm} = \frac{I_{i,j}}{\sqrt{\sum_{k=1}^{2048} I_{i,j,k}^2}}, 1 \leq i,j \leq 7 \tag{4}$$

$$img_c = tanh(W_{img} \cdot img_{emb-norm} + b_{img}) \tag{5}$$

Because the input from the VQA dataset requires reasoning on both image context and question context, the latter can sometimes become cluttered when using only a forward pass, as some tokens might have more meaning and influence over the entire context when they are enriched with the context from

both directions. For most of the longer questions, the subject is influenced by tokens surrounding it. In order to capture the question context, this component uses a Bi-LSTM to concatenate the contexts from the two directions. Having the enriched context for the words, those are passed as input to a forward LSTM, because most tokens influence the future tokens of the question. During testing we observed that this subcomponent for question embedding maximizes the importance of the question context for this attention model. The question context is a 1024 sized vector, $q_c^{(1)}$. Let Q_i be the i^{th} token and $Q = [Q_1 , Q_2 , ..., Q_d]$ the question containing d tokens. Furthermore, we will denote $Q \in R^{d \times n}$ as the word embedding matrix, where *d* is the number of words in the question and *n* is the word embedding size. The question embedding becomes:

$$q_c^{(1)} = LSTM(BRNN(Q)) \tag{6}$$

where BRNN concatenates the two passes for each token Q_i.

In order to start the first stacked attention layer, the two representations need to be fused. Therefore, the question embedding is tiled to have the tridimensional size of the image embedding, obtaining a 7x7 matrix $q^{(1)}_{c\text{-tiled}}$.

For the first attention layer, the two embedding matrices, img_c and $q^{(1)}_{c\text{-tiled}}$, are concatenated on the last axis and now each cell from the 7x7 grid has a 2048 feature vector (here [,] denotes the concatenation operation):

$$concat^{(1)} = \begin{bmatrix} [I_{c,1,1}, q_c^{(1)}] & [I_{c,1,2}, q_c^{(1)}] & \dots & [I_{c,1,7}, q_c^{(1)}] \\ [I_{c,2,1}, q_c^{(1)}] & [I_{c,2,2}, q_c^{(1)}] & \dots & [I_{c,2,7}, q_c^{(1)}] \\ [I_{c,3,1}, q_c^{(1)}] & [I_{c,3,2}, q_c^{(1)}] & \dots & [I_{c,3,7}, q_c^{(1)}] \\ \dots & & & \\ [I_{c,7,1}, q_c^{(1)}] & [I_{c,7,2}, q_c^{(1)}] & \dots & [I_{c,7,7}, q_c^{(1)}] \end{bmatrix} \tag{7}$$

We apply a 2D Convolution of size 1x1 with 512 filters, followed by a tanh activation layer. In order to obtain the attentions, we apply another 2D Convolution of size 1x1 with only one filter. We have tried to use two attention maps, similar to what Kazemi et al. (2017) have proposed, but the results were not improving.

$$conv_1^{(1)} = tanh(Conv2D(concat^{(1)}, ((1,1), 512))) \tag{8}$$

(9)

$$conv_2^{(1)} = Conv2d(conv_1^{(1)}, ((1,1),1))$$

The resulting matrix is flattened and transformed using a softmax activation, so the 49 attention values, $att_{i,j}, 1 \leq i,j \leq 7$, sum to the value 1. The highest values represent image regions that are important in relation with the question.

$$att^{(1)} = softmax(conv_2^{(1)}) \tag{10}$$

$$\sum_{1 \leq i,j \leq 7} att_{i,j}^{(1)} = 1 \tag{11}$$

Afterwards, a weighted sum is done between the image regions and the attention values, resulting a one dimensional vector of size 1024. This is the attended image after the first attention layer.

$$img_{att}^{(1)} = \sum_{1 \leq i,j \leq 7} att_{i,j}^{(1)} I_{i,j} \tag{12}$$

The second attention layer is similar to the first one, except that the initial question embeddings are enhanced by an additive element-wise operation with the attended image embedding $img^{(1)}_{att}$, thus obtaining $q_c^{(2)}$:

$$q_c^{(2)} = img_{att}^{(1)} + q_c^{(1)} \tag{13}$$

After the second attention layer obtains the weighted sum of the image, $img^{(2)}_{att}$, this representation and the initial question embedding are fused through an element-wise multiplication, obtaining a 1024-dimensional vector that is passed to the fifth component, the two layer MLP, which has a softmax activation at the end for classification:

$$f_1 = img_{att}^{(2)} \odot q_c^{(1)} \tag{14}$$

$$f_2 = tanh(W_{f_2} \cdot f_1 + b_{f_2}) \tag{15}$$

$$output = softmax(W_{output} \cdot f_2 + b_{output}) \tag{16}$$

Dual attention model

We have also created a model that follows closely the one presented by Parikh et al. (2016), but instead of comparing two sentences, we will compare the input question with the image segment embeddings from a pre-trained ResNet-152 network. The model that follows is the result of several iterations and minimizations in order to reduce overfitting.

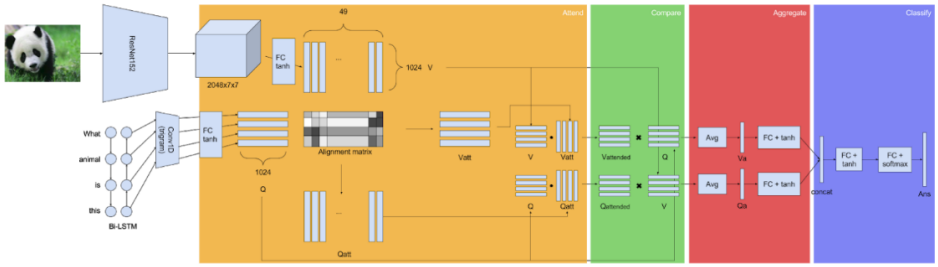

Figure 6. Dual-Attention model. The model can be split into 4 components: Attend (yellow), Compare (green), Aggregate (red) and Classify (blue).

The dual attention model makes use of 4 components which are highlighted in Figure 6. The Attend section extracts the image and text attention and combines each of those with the visual and textual embeddings. The Compare section is tasked with combining the "attended" image and question with the original question and image embeddings, respectively, in hopes of finding where there is correlation between a focused area of the image with a section of the question, or the other way around. The Aggregate section computes a weighted average of the previous layer and passes those through a FC layer. The Classify section concatenates the two embeddings and produces the final answer after 2 FC layers. Additional to the model from Figure 6, we represent the image embedding space as $V \in R^{t \times n}$, where t is the number of image segments and n is the embedding size, same as for words.

We need to place each word of the question in context with the rest of the sentence. This idea is also considered by Karpathy et al. (2015), where the argument is made that a Bidirectional RNN (or Bi-LSTM in our case) would learn to embed the concept of "dog running" from a question such as "Is the dog running?" when reaching the word "running". The bidirectional nature of the network makes sure that the word "dog" receives the same treatment regardless of word order. After observing intermediary outputs, we have noticed that the model would try to attend on the image for words in the

question that are non-descriptive, such as "where", or "the", as each word embedding would not gain enough context from nearby words. Therefore, it did not make much sense to attend on each word, so we chose instead to produce an image attention for each trigram in the question. This produced a significant improvement, which is why, after the Bi-LSTM we follow with a Conv1D layer, followed by a FC layer with "tanh" activation to bring the visual embeddings in the same vector space as the image embeddings.

The visual and textual embeddings which were brought to a similar vector space are then multiplied (by matrix multiplication) to produce an alignment matrix. Ideally, the matrix would get high activations for image segment embeddings which match with certain trigram embeddings. After obtaining the alignment matrix $C \in R^{d \times t}$, we want to know for each word, what is the image segment we should be focusing on, and vice-versa for image segments. To do this, we compute Q_{att} and V_{att}, which are essentially soft-attention maps over the words and image segments, respectively.

$$V_{att\ i} = \sum_{j=1}^{t} \left(\frac{exp(C_{ij})}{\sum_{k=1}^{t} (exp(C_{ik}))} \cdot V_j \right) \tag{17}$$

$$Q_{att\ j} = \sum_{i=1}^{d} \left(\frac{exp(C_{ij})}{\sum_{k=1}^{d} (exp(C_{kj}))} \cdot Q_i \right) \tag{18}$$

$$V_a = mean(V \cdot V_{att}^T) \tag{19}$$

$$Q_a = mean(Q \cdot Q_{att}) \tag{20}$$

The attended visual and textual embeddings are then concatenated (here [,] denotes again the concatenation operation).

$$x_{concat} = tanh(W^{(c)}[Q_a, V_a] + b^{(c)}) \tag{21}$$

Finally, the answer is obtained through a FC layer followed by softmax.

$$ans = softmax(W^{(o)} x_{concat} + b^{(o)}) \tag{22}$$

4. Results and discussion

This section starts by presenting the evaluation method used on the VQA 1.0 dataset for the VisualQA Challenge and continues by describing the experimental setup for our solutions. Afterwards we discuss the results for the proposed models and highlight the importance of the attention.

4.1. Evaluation metrics

Every question from the open-ended track of the VisualQA Challenge has ten answers and a response is considered to be 100% correct if at least three annotators gave the same response. Otherwise, the score for that input pair is proportional with the number of matched responses out of the ten from the dataset, as can be seen in (23). Unfortunately, sometimes even the annotators do not agree on a response and multiple questions do not have an answer that appears at least three times. In order to be consistent with the human accuracies, the VisualQA Challenge uses (24) as the evaluation metric, meaning that accuracies are averaged over all 10 choose 9 subsets of annotators' answers.

$$Accuracy(a) = min(\frac{\#annotators\ that\ provided\ the\ answer}{3}, 1) \tag{23}$$

$$Accuracy(a) = \frac{1}{10}\sum_{k=1}^{10} min(\frac{\sum_{1\leq j\leq 10, j\neq k}\mathbb{1}(a=a_j)}{3}, 1) \tag{24}$$

4.2. Experiments

All experiments were conducted on the real open-ended VQA 1.0 dataset. The models evaluated on the val subset were trained on the train subset and the models evaluated on the test-dev subset were trained on an aggregated subset of train and validation.

Model hyperparameters

The baseline model was trained with the RMSProp optimizer (Tieleman and Hinton, 2012), with a learning rate of 0.001, ρ of 0.9, ε of 1e-08. The model was trained for 80 epochs when on the train set and 120 epochs when on the

train + validation sets. After half the training epochs, the learning rate is halved. The word embeddings are obtained using SpaCy (https://spacy.io/), which determines embeddings starting from pre-trained GloVe vectors. The best results for this model were obtained with ResNet50 and ResNet152 embeddings for the images.

Because the dataset has a predisposition for overfitting due to the biases created by the unique questions, as mentioned in Section 2, regularization techniques are essential to improve the results. Dropout (Srivastava et al., 2014) is the most important method for the presented models and is used before all FC layers and also before convolutional layers for the attention model. For the final layers the dropout rate was 0.5 and on the initial layers it was 0.3 or 0.4. Another aspect which improved our results was to use a small dropout rate of 0.1 on the input embedding of the question and of 0.2 on the input embedding of the image. Also, we have used max-norm constraint of 0.3 on all FC and Convolutional layers. Lastly, another improvement was provided by the use of l2 regularization on the internal weights of LSTMs, with a penalty term of 0.05.

The visual attention model was trained with the Adam optimizer (Kingma et al., 2014), using a configuration of $\beta1 = 0.9$, $\beta2 = 0.999$, $\varepsilon = 1e{-}08$ and a variable learning rate, depending on the type of the model. For the simple models, we have used an initial learning rate of 0.00015, and for the attention model, we have used an initial learning rate of 0.001, both of them using exponential decay. For the simple models, α was varied between -0.0015 and -0.0025, and for the attention model its value was -0.0002. The minimum capping value for the simple models was 0.00002 and for the attention model it was 0.0002.

The dual attention model was also trained with Adam, with an initial learning rate of 0.001. The models are trained for 80 epochs when on the train set and 120 epochs when on the train + validation sets. After each epoch, the learning rate will be decreased via an exponential decay: $lr_{(i)} = max(lr_{(0)} \cdot exp(decay \cdot (i + 1), lr_{min}))$. The gradients are clipped to 0.1. Every FC layer uses a max-norm constraint of 3 as a form of regularization. The LSTM layers use an L2 regularization of 0.1. The best results for this type of model were obtained with ResNet152 embeddings for the images. Additional model hyperparameters are shown in Figure 7.

n - common embedding size	1024
d - number of words	< 27
t - number of image segments	49
dropout	0.2, 0.3 for input layers, 0.5 for before every other layer
batch size	128
word vector size	300
image vector size	$7 \times 7 \times 2048$

Figure 7. Additional model hyperparameters

4.2. Results and discussion

This section presents quantitative results for all proposed models and also makes a short analysis of these results.

Table 1. Accuracy of proposed models on the VQA validation set

Model	Overall	Other	Number	Yes/No
Baseline - 1k ans - ResNet50	56.46	42.80	34.55	82.06
Baseline - 3k ans - ResNet50	56.58	42.86	34.85	82.21
Baseline - 1k ans - ResNet152	57.04	43.40	34.73	82.00
Image-Stacked-Att	59.61	48.35	36.05	82.56
Dual-Att - 3k ans - ResNet152	59.64	48.82	36.52	81.84

To interpret the results presented in Table 1, we will go through each of the models, starting with “Baseline - 1k ans - ResNet50”. This is the implementation of the model presented in 3.1, using 1000 answers as output classes and ResNet50 embeddings. With proper regularization and using the RMSProp optimizer, this model reaches up to 56.58% accuracy, and is the backbone of all the baseline-related models. One massive improvement over "deeper LSTM Q + norm I" presented in Antol et al. (2015) is the application

of l2-regularization on the image embeddings. Surprisingly, there is no significant improvement in choosing the top 3k answers over the top 1k answers for the baseline models. A significant improvement is shown when moving from a ResNet50 model to a ResNet152 model, as we can see from Table 1. However, the per-answer responses show that these models still lack in the "number" and "other" categories.

As expected, a big difference of approximately 2.4% can be observed between the stacked attention model and the baseline model. It may be surprising that the difference is not bigger, but this shows that even though the model is capable of figuring out where to look in the picture, it still cannot quite use that information to full effect, which can be a consequence of the biased training set. The attention model outperforms the baseline in all categories, the most important difference being in the "Others" category showing that the model has a fine-grained object recognition due to the attention mechanisms.

Moving to the dual-attention model, we notice that it does a much better job at answering questions in the "other" and "number" categories than the baseline, while sacrificing a bit from the "yes/no" category, and only slightly better than the stacked attention model presented previously. We can also notice that there starts to be a significant difference between choosing the most common 3k answers over the most common 1k. This might be because deeper models capable of reasoning do a much better job at identifying different types of situations and adapting to them, unlike the baseline models which tend to overfit on specific answer types.

Table 2. Accuracy on the VQA test-dev set

Model	Overall	Other	Number	Yes/No
Baseline - 1k ans - ResNet50	60.07	45.63	37.46	82.96
Baseline - 3k ans - ResNet50	59.97	45.42	37.15	83.06
Baseline - 1k ans - ResNet152	57.04	46.26	38.19	82.76
Image-Stacked-Att	63.11	51.50	38.99	83.09

Dual-Att - 3k ans - ResNet152	63.12	51.96	38.76	82.63

Finally, we can observe the same behaviour as on the validation set apply to the test-dev set (Table 2). The best models are the ones who use visual and/or textual attention, reaching test-dev accuracies of ~63.12%, bringing them in the top 7 models for the VQA v1 competition (see results online at http://www.visualqa.org/roe.html). Also, we can observe that the baseline models greatly surpass the ones presented in Antol et al. (2015), validating the conclusions of Jabri et al. (2016) that baseline models can reach a high accuracy, close to the results of more complex models with attention. This proves either that attention models still have more work to do before we can accurately label them as capable of reasoning, or that the dataset is heavily biased such that simpler models able to overfit on particular answer types (effectively "learning the question") are almost as good as attention models.

4.4. Attention map analysis

This section presents a few examples of attention maps obtained using our model best performing models, Image_Stacked_Att, with 2 stacked attention layers, and Dual-Att, which has both image and text attention layers. The shown examples are from the validation set, after training the model on the train set. The visual attention maps are obtained using the values from the softmax layers representing the attention maps and by performing a bicubic interpolation on the 7 x 7 grid corresponding to the image regions.

Figure 8. Two examples of attention maps on each of the two attention layers. For each example the given answer with its corresponding probability are shown. The second layer improves the focus of the first layer.

Figure 8 shows examples of attention maps for the first and second attention layers, alongside the initial image. It can be observed that the second attention layer refines the focus of the first layer, which is a bit fuzzy. Interestingly, although sometimes the first attention layer focuses completely wrong, as shown in Figure 9, the second layer is able to correct its attention map. This phenomenon might explain the small increase in our accuracy if we use two stacked attention layers instead of a single one, because sometimes the first attention layer helps the overall process and sometimes it just complicates the process for the second layer, focusing on a wrong part of the image. Even if the focus of the second example from Figure 9 is correct, the response is wrong, and this might be because the first attention layer did not focus correctly, after which the second layer could have refined its attention to the snow.

Figure 9. Two examples of attention maps on each of the two attention layers where the first layer focuses completely wrong and the second layer corrects its focus.

Moving on to the Dual Attention model, in Figure 10 we can see what the attention layers for each word focus on separate trigram of words in the question. We can also see what the weighted sum of the attention of those layers looks like, which produces the final visual attention seen in the second photo of each figure. Furthermore we can see what the textual attention layers look like for each trigram of the question, which shows what level of importance the model attributes to each trigram when considering them for the final answer.

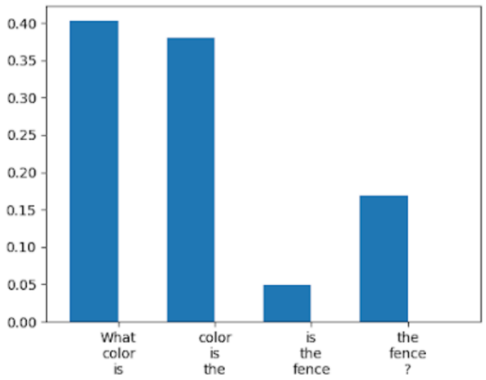

Figure 10. An example of visual and textual attention maps and vectors respectively. In the series of photos, the first one represents the original input image and question, the second one is the aggregated attention map for the image along with the answer and its certainty. The following four are attention maps for each trigram in the question. The last one represents textual attention vectors shown as bar charts

In Figure 11 we can observe the current limitations of our model. These are mainly caused by the input image which was resized to 224 × 224, effectively throwing away a lot of information and many details. Even if we were to resize to 448 × 448 like in Fukui et al. (2016) or Nam et al. (2016), the model would still have trouble with the first example, as every end-to-end model needs to resize the picture, thus making it incapable of reading small text. A solution to this type of question would be a modular network that would use a fully-convolutional network with no FC layers, specialized in extracting text from images that would permit the input image to be of variable size. The second example shows a limitation brought by the dataset itself, as the model could not learn the difference between "picture" and "icon" from that single example. The third example show a limitation brought both by resizing the image to 224 × 224 and by the limited vocabulary from the training examples. The model correctly identifies that there is a plant in the picture but fails to categorize it as a tree, also probably due to the obstruction in the image.

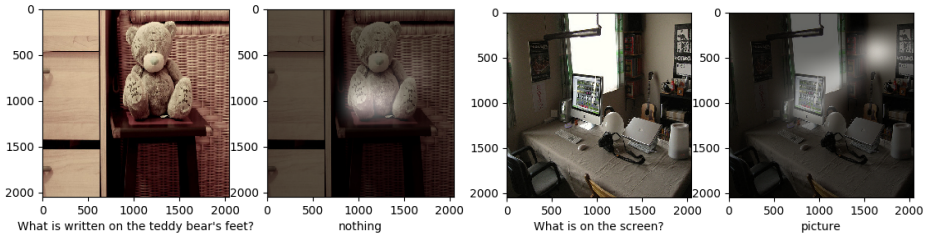

Figure 11. Examples showing the limitations of the Dual Attention model

5. Conclusions

In this paper we have presented two neural models for the Visual Question Answering task that make use of attention mechanisms in order to align the image and question embeddings. The two models, image stacked attention and image-text dual attention, achieve similar overall performance of about 63.10% on the test-dev dataset of the VQA v1 challenge, which places them in the top 10 solutions developed for this task.

From a computational point of view, the main findings are threefold. First, complex attention mechanisms are mandatory to align text and image embeddings and improve the performance of the solutions developed for VQA. Second, the developed models achieve a somehow similar performance, also in line with the other top performers for the challenge. Even the winner of the VQA challenge, using MCB Pooling, achieves only a slightly better performance of 64.20% without augmenting the training dataset (Fukui et al., 2016). This shows that current approaches are limited and there is a need for a fundamental breakthrough in the models. At last, the two proposed methods can be combined to achieve a slightly better accuracy as the dual attention model achieves constantly better results on the "Other" type of questions, while the stacked attention models performs better for the "Yes/No" questions.

At the end, solving the VQA challenge would offer substantial advances in the HCI community. Not only this technology would provide a substantially better method of interaction with the environment (including real-life and online) for vision impaired users, but it would also be a powerful means of querying large image and video collections in natural language for any user. However, the current performance of the VQA solutions does not enable the development of commercial tools and adoption for everyday users.

Acknowledgement

This research has been partially supported by the Text2NeuralQL (PN-III-P2-2.1-PTE-2016-0109) research project.

References

Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Lawrence Zitnick, C., & Parikh, D. (2015). Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 2425-2433).

Fukui, A., Park, D. H., Yang, D., Rohrbach, A., Darrell, T., & Rohrbach, M. (2016). Multimodal compact bilinear pooling for visual question answering and visual grounding. *arXiv preprint arXiv:1606.01847*.

Gao, Y., Beijbom, O., Zhang, N., & Darrell, T. (2016). Compact bilinear pooling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 317-326).

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.

Jabri, A., Joulin, A., & van der Maaten, L. (2016). Revisiting visual question answering baselines. In *European conference on computer vision* (pp. 727-739). Springer, Cham.

Karpathy, A., & Fei-Fei, L. (2015). Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3128-3137).

Kazemi, V., & Elqursh, A. (2017). Show, Ask, Attend, and Answer: A Strong Baseline For Visual Question Answering. *arXiv preprint arXiv:1704.03162*.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).

Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. and Zitnick, C.L. (2014). Microsoft coco: Common objects in context. In *European conference on computer vision* (pp. 740-755). Springer, Cham.

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)

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural*

information processing systems (pp. 3111-3119).

Nam, H., Ha, J. W., & Kim, J. (2016). Dual attention networks for multimodal reasoning and matching. arXiv preprint arXiv:1611.00471.

Parikh, A. P., Täckström, O., Das, D., & Uszkoreit, J. (2016). A decomposable attention model for natural language inference. *arXiv preprint arXiv:1606.01933*

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532-1543).

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, *15*(1), 1929-1958.

Tieleman, T., & Hinton, G. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, *4*(2), 26-31.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R. and Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning* , 2048-2057.

.