Compositional Memory for Visual Question Answering

Aiwen Jiang[1,2]   Fang Wang[2]   Fatih Porikli[2]   Yi Li[* 2,3]

[1]Jiangxi Normal University   [2]NICTA and ANU   [3]Toyota Research Institute North America
[1]aiwen.jiang@nicta.com.au   [2]{fang.wang, fatih.porikli}@nicta.com.au   [3]yi.li@tema.toyota.com

Abstract

Visual Question Answering (VQA) emerges as one of the most fascinating topics in computer vision recently. Many state of the art methods naively use holistic visual features with language features into a Long Short-Term Memory (LSTM) module, neglecting the sophisticated interaction between them. This coarse modeling also blocks the possibilities of exploring finer-grained local features that contribute to the question answering dynamically over time.

This paper addresses this fundamental problem by directly modeling the temporal dynamics between language and all possible local image patches. When traversing the question words sequentially, our end-to-end approach explicitly fuses the features associated to the words and the ones available at multiple local patches in an attention mechanism, and further combines the fused information to generate dynamic messages, which we call episode. We then feed the episodes to a standard question answering module together with the contextual visual information and linguistic information. Motivated by recent practices in deep learning, we use auxiliary loss functions during training to improve the performance. Our experiments on two latest public datasets suggest that our method has a superior performance. Notably, on the DAQUAR dataset we advanced the state of the art by 6%, and we also evaluated our approach on the most recent MSCOCO-VQA dataset.

1. Introduction

Given an image and a question, the goal of Visual Question Answering (VQA) is to directly infer the answer(s) automatically from the image. This is undoubtedly one of the most interesting, and arguably one of the most challenging, topics in computer vision in recent times.

Almost all state-of-the-art methods use predominantly holistic visual features in their systems. For example, Malinowski *et al.* [1] used the concatenation of linguistic feature and visual feature extracted by a Convolutional Neural Network (CNN), and Ren *et al.* [2] considered the visual feature as the first word to initialize the sequential learning.

While the use of holistic approach is straightforward and convenient, it is, however, debatably problematic. For example, in the VQA problems many answers are directly related to the contents of some image regions. Therefore, it is dubious if the holistic features are rich enough to provide the information only available at regions. Also, it may hinder the exploration of finer-grained local features for VQA.

In this paper we propose a Compositional Memory for an end-to-end training framework. Our approach takes the advantage of the recent progresses in image captioning [3, 4], natural language processing [5], and computer vision to advance the study of the VQA. Our goal is to fuse local visual features and the linguistic information over time, in a Long Short-Term Memory (LSTM) based framework. The fused information, which we call "episodes", characterizes the interaction and dynamics between vision and language.

Explicitly addressing the interaction between question words and local visual features has a number of advantages. To begin with, regions provide rich info towards capturing the dynamics in question. Intuitively, parts of an image serve as "candidates" that may have varying importance at different time when parsing a question sentence. Recent study of image captioning [6], a closely related research topic, suggests that visual attention mechanism is very important in generating good descriptions. Obviously, this idea will also improve the accuracy in question answering.

Going deeper, this candidacy is closely related to the concept of semantic "facts" in reasoning. For example, one often begins to dynamically search useful local visual evidences at the same time when (s)he reads words. The use of facts has been explored in the natural language processing recently [5], but this useful concept cannot be explored without local visual information in computer vision.

While the definition of "visual facts" is still elusive, we can approach the problem through modeling interactions between vision and language. This "sensory interaction" plays a phenomenal role in the information processing and reasoning. It has a significant meaning in memory study as well. Eichenbaum and Cohen argued that part of the hu-

man memory needs to be modeled as a form of relationship between spatial, sensory and temporal information [7].

Specifically, our method traverses the words in a question sequentially, and explicitly fuses the linguistic and the ones available at local patches to episodes. An attention mechanism is used to re-weight the importance of the regions [6]. The fused information is fed to a dynamic network to generate episodes. We then feed the episodes to a standard question answering module together with the contextual visual information and linguistic information.

The use of local features inevitably leads to the quest about region selection. In principle, the regions can be 1) patches generated by object proposals, such as those obtained by edgebox [8] and faster-RCNN [9], and 2) overlapping patches that cover most important contents in image. In this paper, we choose the latter and use the features of the last convolutional layer in the CNNs.

Our experiments on two latest public datasets suggest that our method outperforms the other state of the art methods. We tested on the DAQUAR [10] and MSCOCO-VQA [11]. Notably, on the DAQUAR dataset we advanced the state of the art by 6%. We also compared a few variants of our method and demonstrated the usefulness of the Compositional Memory for VQA. We further verified our idea on the latest MSCOCO-VQA dataset.

The main contributions of the paper are:

- We present an end-to-end approach that explores the local fine grained visual information for VQA tasks,

- We develop a Compositional Memory that explicitly models the interactions of vision and language,

- Our method has a superior performance and it outperforms the state of the art methods.

2. Related Work

CNN, RNN, and LSTM   Recently, deep learning has achieved great success on many computer vision tasks. For example, CNN has set records on standard object recognition benchmarks [12]. With a deep structure, CNN can effectively learn complicated mappings from raw images to the target, which requires less domain knowledge compared to handcrafted features and shallow learning frameworks.

Recurrent Neural Networks (RNN) have been used for modeling temporal sequences and gained attention in speech recognition [13], machine translation [14], image captioning [3, 4]. The recurrent connections are feedback loops in the unfolded network, and because of these connections, RNNs are suitable for modeling time series with strong nonlinear dynamics and long time correlations. The traditional RNN is hard to train due to the vanishing gradient problem, *i.e.* the weight updates computed via error backpropagation through time may become very small.

Long Short Term Memory model [15] has been proposed as a solution to overcome these problems. The LSTM architecture uses memory cells with gated access to store and output information, which alleviates the vanishing gradient problem in backpropagation over multiple time steps. Specifically, in addition to the hidden state, the LSTM also includes an input gate, a forget gate, an output gate, and the memory cell. In this architecture, input gate and forget gate are sigmoidal gating functions, and these two terms learn to control the portions of the current input and the previous memory that the LSTM takes into consideration for overwriting the previous state. Meanwhile, the output gate controls how much of the memory should be transferred to the hidden state. These mechanisms allow LSTM networks to learn temporal dynamics.

Language and vision   The effort of combining language and vision attracts a lot of attention recently. Image captioning and VQA are two most intriguing problems.

Question answering (QA) is a classical problem in natural language processing [5]. When images are involved, the goal of VQA is to infer the answer of a question directly from the image [11]. Multiple questions and answers can be associated to the same image during training.

It has been shown that VQA can borrow the idea from image captioning. Being a related area, image captioning also uses RNN for sentence generation [3]. Attention mechanism is recently adopted in image captioning and proves to be a useful component [6].

LSTM for VQA   Because a VQA system needs to process language and visual information simultaneously, most recent work adopted the LSTM in their approaches. A typical LSTM-VQA uses holistic image features extracted by CNNs as "visual words", as shown in Figure 1. The visual word features are used either as the first or at the end of question sequence [2] or they are concatenated together with question word vectors into a LSTM [1, 16] .
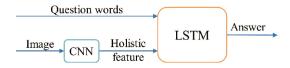

Figure 1. Classical LSTM-VQA model.

This LSTM-VQA framework is straightforward and useful. However, treating the feature interaction as feature vector concatenation lacks the capability that explicitly extracts finer-grained information. As we discussed before, details of facts may be neglected if global visual features are used. This leads to the quest for more effective information fusion model for language and image in VQA problems.

3. Our Approach

We present our approach in this section. First, we present our model overview. Then, we discuss the technical details and explain the training process.

3.1. Our End-to-End VQA model

Compared to the basic LSTM approach for VQA in Figure 1, We made two major improvements of the model. The diagram of the network is shown in Figure 2.
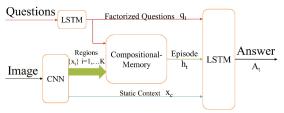

Figure 2. Proposed VQA network. Please see Figure 3 for the description of the Compositional Memory block

The first, and the most important addition, is a Compositional Memory. It reasons over those input features to produce an image-level representation for answer module. It reflects an experience over image contents.

The second addition is the LSTM module for parsing (factorizing) the question. It provides input for both the Compositional Memory and the question answering LSTM. In the experiment we will show the importance of this module for the VQA tasks.

In part, this implementation is aligned with the findings in cognitive neuroscience. It is well known that semantic (*e.g.*, visual features and classifiers) and episodic memory (*e.g.*, temporal questioning sentence) together make up the declarative memory of human beings, and the interactions among them become the key in representation and reasoning [17]. Our model captures this interaction naturally.

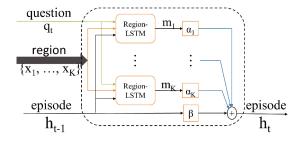### 3.1.1 Compositional Memory

Figure 3. Diagram of Compositional Memory.

We present the details of our Compositional Memory in this section. This unit consists of a number of Region-LSTM and α gates (Figure 3). All these Region-LSTM share the same set of parameters, and their results are combined to generate an episode at each time step.

The region-LSTMs are mainly in charge of processing input image region contents in parallel. It dynamically generates language-embedded visual information for each region, conditioned on previous episodes, local visual feature, and current question word. The state, gates and cells of of each Region-LSTM are updated as follows:

$$i_t^k = \sigma(W_{qi}q_t + W_{hi}h_{t-1} + W_{xi}X_k + b_i) \qquad (1)$$

$$f_t^k = \sigma(W_{qf}q_t + W_{hf}h_{t-1} + W_{xf}X_k + b_f) \qquad (2)$$

$$o_t^k = \sigma(W_{qo}q_t + W_{ho}h_{t-1} + W_{xo}X_k + b_o) \qquad (3)$$

$$g_t^k = \tanh(W_{qg}q_t + W_{hg}h_{t-1} + W_{xg}X_k + b_g) \qquad (4)$$

$$c_t^k = f_t^k \odot c_{t-1}^k + i_t^k \odot g_t^k \qquad (5)$$

$$m_t^k = o_t^k \odot \tanh(c_t^k) \qquad (6)$$

where the hidden state h_t denotes an episode at every time step, q_t is the language information (*e.g.,* features generated by *word2vec*), and X_k is the kth regional CNN features. m_t is the output of region-LSTM t. Please note that the superscripts are omitted in the above notations for simplicity.

In Region-LSTM[1], c_t denotes the memory cell, g_t denotes an input modulation gate. i_t and f_t are input and forget gates, which control the portions of the current input and the previous memory that LSTM takes into consideration. o_t is output gate that determines how much of the memory to transfer to the hidden state. $\odot$ is the element-wise multiplication operation, and $\sigma()$ and $tanh()$ denote the sigmoidal and tanh operations, respectively. These mechanisms allow LSTM to learn long-term temporal dynamics.

The implementation of Region-LSTM is similar to that of traditional LSTM. However, the important differences lie in the parallel strategy that all parameters $(W_{q*}, W_{h*}, W_{x*}, b_*)$ are shared across different regions. Please note that each Region-LSTM has its own gate and cell state, respectively.

The α gate is also conditioned on previous episode h_{t-1}, region feature X_k, and current input language feature q_t. It returns a single scalar for each region. This gate is mainly used for weighted combination of region messages for generating episodes, which are dynamically pooled into image-level information. Similar to $W_{zq}, W_{zh}, W_{zx}, b_z, W_\alpha, b_\alpha$, the parameters of α gate, are also shared across regions. At every time step t, α gate dynamically generates values α_k^t for k^{th} region.

$$z_k^t = \tanh\left(W_{zq}q_t + W_{zh}h_{t-1} + W_{zx}x_k + b_z\right)$$
$$\alpha_k^t = \sigma\left(W_\alpha z_k^t + b_\alpha\right) \qquad (7)$$

In order to summarize region-level information to an image-level feature, we employ a modified pooling mechanism of gated recurrent style [18]. The episode h_t acts as

---

[1]Please note that the diagram is not drawn for the purpose of simplicity.
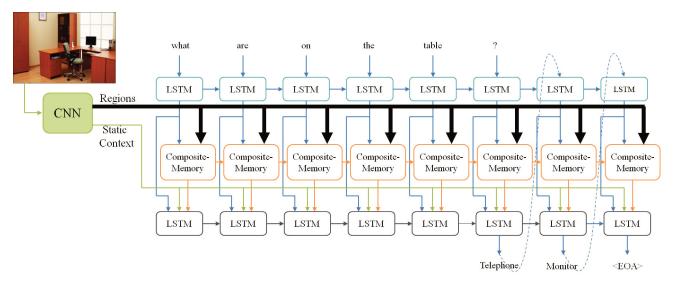
Figure 4. Unfolding our end-to-end VQA model. CM: Compositional Memory.

dynamic feature as well as hidden state of Compositional Memory unit. It is updated to renew the input information of Region-LSTMs together with language input at every time step.

$$\beta = 1 - \frac{1}{K} \sum_k \alpha_k$$
$$h_t = \beta h_{t-1} + \frac{1}{K} \sum_k \alpha_k^t m_t^k \qquad (8)$$

where K is the total number of regions.

One of the advantages of Compositional Memory is that it goes beyond traditional static image feature. The unit incorporates both the merits of LSTM and attention mechanism, and thus it is suitable to generate dynamic episodic messages for visual question answering applications.

3.1.2 Language LSTM

We use a Language LSTM to process linguistic inputs. We consider this representation as "factorization of questions", which captures the recurrent relations of question word sequence, and stores semantic memory information about the questions. This strategy for processing language has also been proven to be important in image captioning [3].

3.1.3 Other components

Answer generation The answer generation module takes dynamic episodes, together with language and static visual context generated by CNNs to generate the answers in a LSTM framework.

CNN Our approach is compatible with all the major CNN network, such as AlexNet [19] and GoogLeNet [20].

In each CNN, we use the last convolution layer as region selections. For example, GoogleNet, we use the $1024 \times 7 \times 7$ feature map from the inception_5b/output. This means that Compositional Memory operates on 49 regions, each of which is represented by 1024-dim feature vector. Following the current practice, we used the output of the last CNN layer as visual context.

3.2. Training

Generally, there are three types of the VQA tasks: 1) Single Word, 2) Multiple Words (or called "Open-ended"), and 3) Multiple Choice. The single word category restricts the answer to have only one single word. Multiple Words VQA is an open-ended task, where the length of answer is not limited and the VQA system needs to sequentially generate possible answer words until generating an answer-ending mark. The multiple choice task refers to select most probable answer from a set of answer candidates.

Among these three, the "Open-ended" VQA task is the most difficult one, thus we chose this category to demonstrate the performance of our approach. In this section, we briefly present the training procedure, the loss function, and other implementation details.

3.2.1 Protocol

During the training of our VQA system, CNN and LSTMs are jointly learned in an end-to-end way. The unfolded network of our proposed model is shown in Figure 4.

In this Open-ended category of VQA, questions may have multiple word answers. We consequently decompose the problem to predict a set of answer words $A = \{a_1, a_2, ..., a_M\}$, where a_i are words from a finite vocabulary Ω' and M is the number of answer words for a given

question and image. To deal with open-ended VQA task, we add an extra token $\langle EOA \rangle$ into the vocabulary $\Omega = \Omega' \cup \{\langle EOA \rangle\}$. The $\langle EOA \rangle$ indicates the end of the answer sequence. Therefore, we formulate the prediction procedure recursively as:

$$\widehat{a}_t = \arg \max p(a|X, q, \widehat{A}_{t-1}; \vartheta) \qquad (9)$$

where $\widehat{A}_{t-1}$ is the set of previously predicted answer words, with $\widehat{A}_0 = \{\}$ at start. The prediction procedure is terminated when $\widehat{a}_t = \langle EOA \rangle$.

As shown in Figure 2 and Figure 4, we feed the VQA system with a question as a sequence of words, *i.e.* $q = [q_1, q_2, ..., q_{n-1}, [\![?]\!]]$, where $[\![?]\!]$ encodes the end of question. In the training phase, we augment the question word sequence with the corresponding ground truth answer sequence a, *i.e.* $\widehat{q} := [q, a]$. During the test phase, at t^{th} time step we augment question q, with previously predicted answer words $\widehat{q}_t := [q, \widehat{a}_{1,...,t-1}]$.

3.2.2 Loss function

All the parameters are jointly learned with cross-entropy loss. The output predictions that occur before the ending question mark $[\![?]\!]$ are excluded from the loss computation, so that the model is solely penalized based on the predicted answer words.

Motivated by the recent success of GoogLeNet, We adopt a multi-task training strategy for learning the parameters of our network. Specifically, in additional to the question answering LSTM, we add a "Language Only" loss layer on the Language LSTM, and an "Episode Only" loss layer on the Compositional Memory. These two auxiliary loss functions are added during training to improve the performance, and they are removed during testing.

3.2.3 Implementation

We implemented our end-to-end VQA network using Caffe[2]. The CNN models are pre-trained, and then fine-tuned in our recurrent network training. The source code of our implementation will be available in public.

4. Experiments

We test our approach on two large data sets, namely, DAQUAR [10] and MSCOCO-VQA[3] [11]. In the experiments on these two data sets, our method outperforms the state of the arts in different well recognized metrics.

4.1. Datasets

DAQUAR contains 12,468 human question answer pairs on 1,449 images of indoor scene. The training set contains 795 images and 6,793 question answer pairs, and the testing set contains 654 images and 5,675 question answer pairs.

We run experiments for the full dataset with all classes, instead of their "reduced set" where the output space is restricted to only 37 object categories and 25 test images in total. This is because the full dataset is much more challenging and the results are more meaningful in statistics. The performance is reported using the "Multiple Answers" category but the answers are generated using open-ended approach.

MSCOCO-VQA is the latest VQA dataset that contains open-ended questions about images. This dataset contains 369,861 questions and 3,698,610 ground truth answers based on 123,287 MSCOCO images. These questions and answers are sentence-based and open-ended. The training and testing split follows MSCOCO-VQA official split. Specifically, we use 82,783 images for training and 40,504 validation images for testing.

4.2. Evaluation criteria

DAQUAR On the DARQUAR dataset, we use the Wu-Palmer Similarity (WUPS) [21] score at different thresholds for comparison. There are three metrics: Standard Metric, Average Consensus Metric and Min Consensus Metric. The Standard Metric is the basic score. The last two metrics are used to study the effects of consensus in question answering tasks. Please refer to [10] for the details.

MSCOCO-VQA On the MSCOCO-VQA dataset, we use the evaluation criteria provided by the organizers. For the open-ended tasks, the generated answers are evaluated using accuracy metric. It is computed as the percentage of answers that exactly agree with the ground truth provided by human. Please refer to [11] for details.

4.3. Experimental settings

We choose AlexNet and GoogLeNet in our experiments, respectively. For AlexNet, the region features are from Pool5 layer. For GoogLeNet we use the inception_poo5b/output layer. That means our Compositional Memory processes flattened 36 regions for AlexNet, and 49 for GoogLeNet.

In our current implementation, the parameters of region-LSTMs and α gates in Compositional Memory are shared across regions, therefore the computational burden is minimal. However, as regions have to store their respective memory states, the storage space is more than traditional LSTM. In our experiments, the dimension of region inputs d is 1024 for GoogLeNet and 256 for AlexNet. The dimen-
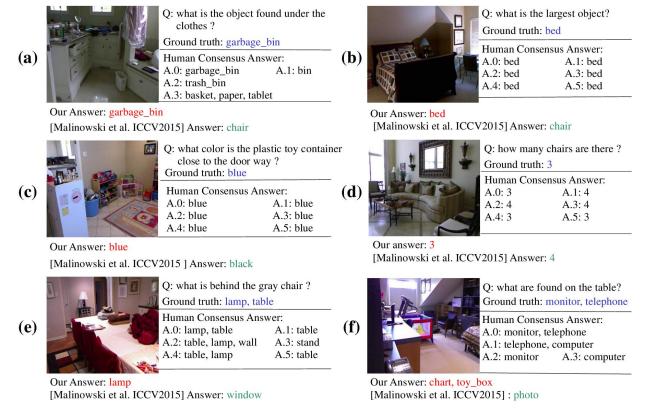
Figure 5. Examples of questions and answers on DAQUAR. Ground truth answers are colored in blue, our predicted answers in red and answers of state-of-the-art method in green. a)-d) are correct examples, and e)-f) are failure cases.

sion of the episodes is set to 200 for all LSTMs (including our Compositional Memory) on the DAQUAR dataset and the MSCOCO-VQA.

We used separate dictionaries for training and testing. To generate the dictionary, we first remove the punctuation marks, except those used in timing and measurements, separate them and convert them to lower cases.

We stop our training procedure after 100,000 iterations. The base learning rate is set to be 0.01. During training, it takes about 0.4 sec for one iteration on GTX Nvidia 780.

4.4. Results on DAQUAR

4.4.1 Examples

We first show some examples of the proposed method. Figure 5a-5d show some correct examples and Figure 5e-5f are failure cases. These examples demonstrate the effectiveness of the local features.

When questions are sophisticated and location dependent, local features help most. For example (Figure 5a and Figure 5c), the object name and the color cannot be easily obtained without the focus of the local region. Take Figure 5d) for another example, the number of chairs need to be counted. These cannot happen without the help of the local feature information encoded in the episodes.

We also show some failure cases in Figure 5e) and 5f). We observe that these are challenging cases, and even the human answers are not consistent. Yet, our model is still able to find partially correct answers. For example (Figure 5e), while our answer does not fully match the ground truth, we provide the information that is much closer to the comparison methods. It also should be noted that, although our answers may be incorrect (Figure 5f), we can still see the answers being related to object near the ground-truth objects. These answers, albeit wrong, still show that our regional feature episodes are useful and provide potentially meaningful answers.

4.4.2 Comparisons with state-of-the-art methods

We compare our proposed model with [1, 10]. Because they used different CNN methods, we tested both AlexNet and GoogLeNet. The performance on "Multiple Answers" ("Open-ended") category are shown in Table 1.

The statistical results shows that the performance of our model substantially is better than the state of the art. On the WUPS@0.9, our method is 6% higher (from 23.31% to 29.77%). When we lower the threshold in the WUPS, we are 5.25% superior than the state of the art.

In other two measurements, where "consensus" is cal-

Table 1. Comparisons on full DAQUAR dataset, "Multiple Answers" category. The numbers are shown in percentage.

	Accuracy	WUPS	
		@0.9	@0.0
Standard Metric			
Malinowski et al. [10]	7.86	11.86	38.79
Ask-Neurons [1]	17.57	23.31	57.49
Our Model (AlexNet)	21.92	27.67	**62.74**
Our Model (GoogleNet)	**24.37**	**29.77**	62.73
Human Answers	50.20	50.82	67.27
Average Consensus Metric			
Ask-Neurons [1]	11.31	18.62	53.21
Our Model (AlexNet)	14.72	22.58	**58.17**
Our Model (GoogleNet)	**16.29**	**23.95**	57.68
Min Consensus Metric			
Ask-Neurons [1]	22.74	30.54	68.17
Our Model (AlexNet)	29.48	37.60	**75.16**
Our Model (GoogleNet)	**31.52**	**39.30**	74.51

Table 2. WUPS@R: Comparisons of different variants of our model.

	WUPS			
	@0.9	@0.7	@0.5	@0.0
Baseline	15.64	36.12	52.43	54.41
Factorized Language Only	25.77	47.90	59.00	59.52
Episodes Only	27.43	49.92	60.95	61.35
Language + Episodes	28.73	51.16	61.28	61.70
Full Model	**29.77**	**52.64**	**62.35**	**62.73**

Table 2 shows the performance under different threshold in the WUPS metric. For example, it is over 4% better than the "Language Only" variant. Since language semantics are important for answering questions and logical reasoning, while regional contents are more critical for answering questions about the existence of objects in image, Their fusion can further improve the qualities of answers. As shown in "Language+Episodes", this fusion increases WUPS@0.9 from 25.77% to 28.73%. With all components, our full model is consistently better than other variants.

As a conclusion, these three types of information are complementary and their combinations improve the solution of the VQA problem.
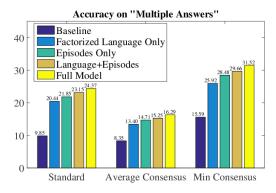
culated on the answers from multiple subjects, our method also outperforms the state of the art 2% to 7%. This further confirms our system is more accurate and robust.

We also find that our method has better performance when GoogLeNet is used in our framework, although the difference is marginally noticeable.

4.4.3   Effectiveness of Compositional Memory module

We further present our study on the effectiveness of Compositional Memory and the Language LSTM in our VQA system. Specifically, we show the comparison in performance when we toggle on and off these components.

We consider the configuration where all modules are used as the "full model", and also name the configuration of traditional approach (Fig. 1) as "baseline". We then introduce three variants, where only language is used ("Factorized Language Only"), only Compositional Memory is used ("Episodes Only"), and both are used together ("Language+Episodes").

The statistic results are shown in Figure 6 and Table 2. One can easily see that, the performance drops seriously when all the proposed memory are toggled off.The episodes provide more information than the language, because it contains visual information from local image patch. All together they achieve the score that each of them cannot reach. This demonstrates that our Compositional Memory module contains critical information for visual question answering and the importance of the Language LSTM.

Using regional information does not rule out the importance of the visual context. This holistic image features describes the abstraction of image. One can see that the performance decreases, for example, from 24.37% to 23.15% on standard accuracy without the context.

Figure 6. Comparisons of different variants of our model.

4.5. Results on MSCOCO-VQA

Compared to DAQUAR, MSCOCO-VQA is the latest VQA dataset. It is much larger and contains more scenes and question types that are not covered by DARQUAR.

Possibly because this is the latest outcome, there are different ways of evaluating the performances and reporting the results. For example, while the measurement of accuracy is well defined, the evaluation protocols are not standardized. Some practitioners use the organizer's previous release for training and validating, and further split the validation sets. Only until recently the organizers release their `test-dev` set online, however, there are still many ways of handling the input. For example, the official version [11] selects the most frequent 1000 answers, which covers only 82.67% of the answer set. Different selections of dictio-

Figure 7. Examples on MSCOCO-VQA. Ground truth answers are colored in blue, and our predicted answers in red.

nary can lead to fluctuations in the accuracy. Finally, the tokenizers used in different practitioners may lead to other uncertainties in accuracy.

Due to the above concerns, we conclude that it is in the early stage of the evaluation, and would like to clearly outline our practices when readers examine the numbers.

- We used a naive tokenizer as specified in Sec. 4.3.
- We used 13,880 words appeared in the training + validation answer set as our answer dictionary.
- We report results on both the `test-dev` and the full validation set.

We first show results of our method in Fig. 7. Compared to Fig. 5, one can see that MSCOCO-VQA is more diversified. More results are shown in Supplementary Materials.

4.5.1 Statistical results

MSCOCO-VQA is grouped to a number of categories based on the types of the questions, and the types of answers. We show the statistics on both categories in this section.

Answer type We report the overall accuracy and those of different answer types using both `text-dev` and the full validation set (Table 3). Please note that we used a larger answer dictionary, which means potentially it is more difficult to deliver correct answers, but still our method achieved similar performance of the state of the art.

One can notice that the accuracy of simple answer type (*e.g.* "yes/no") is very high, but the accuracies drop significantly when the answers become more sophisticated. This indicates the potential issues and directions of our method.

Question type We use validation set to report the accuracy of our method when question type varies (Figure 8). The colored bar chart and sorted according to the accuracy, with the numbers displayed next to the bars.

It is interesting to see a significant drop when the questions ascend from simple forms (*e.g.*, "is there?") to complicated ones (*e.g.*, "what","how"). This suggest that a practical VQA system needs to take this prior into consideration.
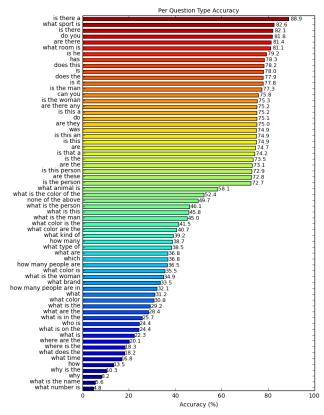

Figure 8. Accuracy on the Open-ended task of MSCOCO-VQA (validation) for different question types.

Table 3. Results on MSCOCO-VQA (Open-ended). Red, blue and green are used to denote top 3. The baseline is **LSTM Q+I** in [11].

	All	Yes/No	Number	Other
Baseline	53.74	78.94	35.24	36.42
Ours (test-dev)	52.62	78.33	35.93	34.46
Ours (val)	50.48	79.05	32.60	33.59

5. Conclusion

In this paper we propose to use the Compositional Memory as the core element in the VQA. Our end-to-end approach is capable of dynamically extracting local the features. The Long Short-Term Memory (LSTM) based approach fuses image regions and language, and generates

the episodes that is effective for high level reasoning. Our experiments on the latest public datasets suggest that our method has a superior performance.

References

[1] M. Malinowski, M. Rohrbach, and M. Fritz, "Ask your neurons: A neural-based approach to answering questions about images," in *ICCV*, 2015.

[2] M. Ren, R. Kiros, and R. Zemel, "Exploring models and data for image question answering," in *NIPS*, 2015.

[3] 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 *CVPR*, 2015.

[4] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *CVPR*, 2015.

[5] A. Kumar, O. Irsoy, J. Su, J. Bradbury, R. English, B. Pierce, P. Ondruska, I. Gulrajani, and R. Socher, "Ask me anything: Dynamic memory networks for natural language processing," *CoRR*, vol. abs/1506.07285, 2015.

[6] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *ICML*, 2015.

[7] N. Cohen and H. Eichenbaum, *Memory, Amnesia, and the Hippocampal System*. Bradford Books, MIT Press, 1995.

[8] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *ECCV*, 2014.

[9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *NIPS*, 2015.

[10] M. Malinowski and M. Fritz, "A multi-world approach to question answering about real-world scenes based on uncertain input," in *NIPS*, 2014.

[11] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, "VQA: visual question answering," *CoRR*, vol. abs/1505.00468, 2015.

[12] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.

[13] A. Graves, A. Mohamed, and G. E. Hinton, "Speech recognition with deep recurrent neural networks," *CoRR*, vol. abs/1303.5778, 2013.

[14] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *CoRR*, vol. abs/1409.0473, 2014.

[15] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[16] 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," *arXiv preprint arXiv:1505.05612*, 2015.

[17] E. Tulving, "Episodic and semantic memory," *Organization of Memory. London: Academic*, vol. 381, no. e402, p. 4, 1972.

[18] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *NIPS workshop on Deep Learning*, 2014.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.

[20] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *CVPR*, 2015.

[21] Z. Wu and M. Palmer, "Verbs semantics and lexical selection," in *ACL*, 1994.