Component Analysis for Visual Question Answering Architectures

Camila Kolling, Jônatas Wehrmann, and Rodrigo C. Barros
Machine Intelligence and Robotics Research Group
School of Technology, Pontifícia Universidade Católica do Rio Grande do Sul
Av. Ipiranga, 6681, 90619-900, Porto Alegre, RS, Brazil
Email: {camila.kolling,jonatas.wehrmann}@edu.pucrs.br, rodrigo.barros@pucrs.br

***Abstract*—Recent research advances in Computer Vision and Natural Language Processing have introduced novel tasks that are paving the way for solving AI-complete problems. One of those tasks is called Visual Question Answering (VQA). This system takes an image and a free-form, open-ended natural-language question about the image, and produce a natural language answer as the output. Such a task has drawn great attention from the scientific community, which generated a plethora of approaches that aim to improve the VQA predictive accuracy. Most of them comprise three major components: (i) independent representation learning of images and questions; (ii) feature fusion so the model can use information from both sources to answer visual questions; and (iii) the generation of the correct answer in natural language. With so many approaches being recently introduced, it became unclear the real contribution of each component for the ultimate performance of the model. The main goal of this paper is to provide a comprehensive analysis regarding the impact of each component in VQA models. Our extensive set of experiments cover both visual and textual elements, as well as the combination of these representations in form of fusion and attention mechanisms. Our major contribution is to identify core components for training VQA models so as to maximize their predictive performance.**

***Index Terms*—Visual Question Answering, Computer Vision, Natural Language Processing.**

I. Introduction

Recent research advances in Computer Vision (CV) and Natural Language Processing (NLP) introduced several tasks that are quite challenging to be solved, the so-called AI-complete problems. Most of those tasks require systems that understand information from multiple sources, i.e., semantics from visual and textual data, in order to provide some kind of *reasoning*. For instance, image captioning [1]–[3] presents itself as a hard task to solve, though it is actually challenging to quantitatively evaluate models on that task, and that recent studies [4] have raised questions on its AI-completeness.

The Visual Question Answering (VQA) [4] task was introduced as an attempt to solve that issue: to be an actual AI-complete problem whose performance is easy to evaluate. It requires a system that receives as input an image and a free-form, open-ended, natural-language question to produce a natural-language answer as the output [4]. It is a multi-disciplinary topic that is gaining popularity by encompassing CV and NLP into a single architecture, what is usually regarded as a multimodal model [5]–[7]. There are many real-world applications for models trained for Visual Question Answering, such as automatic surveillance video queries [8] and visually-impaired aiding [9], [10].

Models trained for VQA are required to understand the semantics from images while finding relationships with the asked question. Therefore, those models must present a deep understanding of the image to properly perform inference and produce a reasonable answer to the visual question [11]. In addition, it is much easier to evaluate this task since there is a finite set of possible answers for each image-question pair.

Traditionally, VQA approaches comprise three major steps: (i) representation learning of the image and the question; (ii) projection of a single multimodal representation through fusion and attention modules that are capable of leveraging both visual and textual information; and (iii) the generation of the natural language answer to the question at hand. This task often requires sophisticated models that are able to understand a question expressed in text, identify relevant elements of the image, and evaluate how these two inputs correlate.

Given the current interest of the scientific community in VQA, many recent advances try to improve individual components such as the image encoder, the question representation, or the fusion and attention strategies to better leverage both information sources. With so many approaches currently being introduced at the same time, it becomes unclear the real contribution and importance of each component within the proposed models. Thus, the main goal of this work is to understand the impact of each component on a proposed baseline architecture, which draws inspiration from the pioneer VQA model [4] (Fig. I). Each component within that architecture is then systematically tested, allowing us to understand its impact on the system's final performance through a thorough set of experiments and ablation analysis.

More specifically, we observe the impact of: (i) pre-trained word embeddings [12], [13], recurrent [14] and transformer-based sentence encoders [15] as question representation strategies; (ii) distinct convolutional neural networks used for visual feature extraction [16]–[18]; and (iii) standard fusion strategies, as well as the importance of two main attention mechanisms [19], [20]. We notice that even using a relatively simple baseline architecture, our best models are competitive to the (maybe overly-complex) state-of-the-art models [21],

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

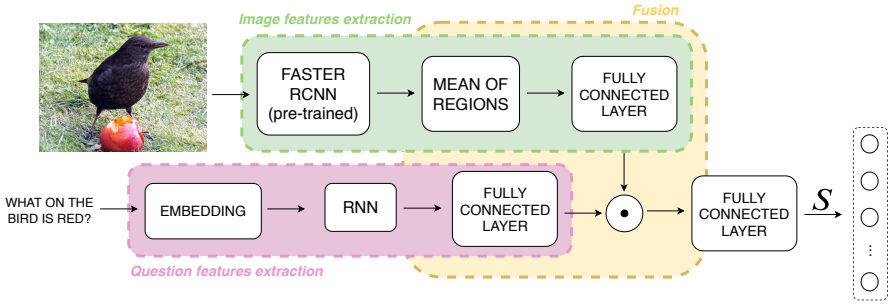

Fig. 1. Baseline architecture proposed for the experimental setup.

[22]. Given the experimental nature of this work, we have trained over 130 neural network models, accounting for more than 600 GPU processing hours. We expect our findings to be useful as guidelines for training novel VQA models, and that they serve as a basis for the development of future architectures that seek to maximize predictive performance.

II. Related Work

The task of VAQ has gained attention since Antol et al. [4] presented a large-scale dataset with open-ended questions. Many of the developed VQA models employ a very similar architecture [4], [23]–[28]: they represent images with features from pre-trained convolutional neural networks; they use word embeddings or recurrent neural networks to represent questions and/or answers; and they combine those features in a classification model over possible answers.

Despite their wide adoption, RNN-based models suffer from their limited representation power [29]–[32]. Some recent approaches have investigated the application of the Transformer model [33] to tasks that incorporate visual and textual knowledge, as image captioning [29].

Attention-based methods are also being continuously investigated since they enable reasoning by focusing on relevant objects or regions in original input features. They allow models to pay attention on important parts of visual or textual inputs at each step of a task. Visual attention models focus on small regions within an image to extract important features. A number of methods have adopted visual attention to benefit visual question answering [28], [34], [35].

Recently, dynamic memory networks [28] integrate an attention mechanism with a memory module, and multimodal bilinear pooling [21], [23], [36] is exploited to expressively combine multimodal features and predict attention over the image. These methods commonly employ visual attention to find critical regions, but textual attention has been rarely incorporated into VQA systems.

While all the aforementioned approaches have exploited those kind of mechanisms, in this paper we study the impact of such choices specifically for the task of VQA, and create a simple yet effective model. Burns et al. [37] conducted experiments comparing different word embeddings, language models, and embedding augmentation steps on five multimodal tasks: image-sentence retrieval, image captioning, visual question answering, phrase grounding, and text-to-clip retrieval. While their work focuses on textual experiments, our experiments cover both visual and textual elements, as well as the combination of these representations in form of fusion and attention mechanisms. To the best of our knowledge, this is the first paper that provides a comprehensive analysis on the impact of each major component within a VQA architecture.

III. Impact of VQA Components

In this section we first introduce the baseline approach, with default image and text encoders, alongside a pre-defined fusion strategy. That base approach is inspired by the pioneer of Antol et al. on VQA [4]. To understand the importance of each component, we update the base architecture according to each component we are investigating.

In our baseline model we replace the VGG network from [20] by a Faster RCNN pre-trained in the Visual Genome dataset [38]. The default text encoding is given by the last hidden-state of a Bidirectional LSTM network, instead of the concatenation of the last hidden-state and memory cell used in the original work. Fig. I illustrates the proposed baseline architecture, which is subdivided into three major segments: independent feature extraction from (1) images and (2) questions, as well as (3) the fusion mechanism responsible to learn cross-modal features.

The default text encoder (denoted by the pink rectangle in Fig. I) employed in this work comprises a randomly initialized word-embedding module that takes a tokenized question and returns a continuum vector for each token. Those vectors are used to feed an LSTM network. The last hidden-state is used as the question encoding, which is projected with a linear layer into a d-dimensional space so it can be fused along to the visual features. As the default option for the LSTM network, we use a single layer with 2048 hidden units. Given that this text encoding approach is fully trainable, we hereby name it Learnable Word Embedding (LWE).

For the question encoding, we explore pre-trained and randomly initialized word-embeddings in various settings, including Word2Vec (W2V) [13] and GloVe [12]. We also explore the use of hidden-states of Skip-Thoughts Vector [14] and BERT [15] as replacements for word-embeddings and sentence encoding approaches.

Regarding the visual feature extraction (depicted as the green rectangle in Fig. I), we decided to use the pre-computed features proposed in [20]. Such an architecture employs a ResNet-152 with a Faster-RCNN [16] fine-tuned on the Visual Genome dataset. We opted for this approach due to the fact that using pre-computed features is far more computationally efficient, allowing us to train several models with distinct configurations. Moreover, several recent approaches [21], [22], [39] employ that same strategy as well, making it easier to provide fair comparison to the state-of-the-art approaches. In this study we perform experiments with two additional networks widely used for the task at hand, namely VGG-16 [17] and ReSNet-101 [18].

Given the multimodal nature of the problem we are dealing with, it is quite challenging to train proper image and question encoders so as to capture relevant semantic information from both of them. Nevertheless, another essential aspect of the architecture is the component that merges them altogether, allowing for the model to generate answers based on both information sources [40]. The process of multimodal fusion consists itself in a research area with many approaches being recently proposed [21], [23], [41], [42]. The fusion module receives the extracted image and query features, and provides multimodal features that theoretically present information that allows the system to answer to the visual question. There are many fusion strategies that can either assume quite simple forms, such as vector multiplication or concatenation, or be really complex, involving multilayered neural networks, tensor decomposition, and bi-linear pooling, just to name a few.

Following [4], we adopt the element-wise vector multiplication (also referred as Hadamard product) as the default fusion strategy. This approach requires the feature representations to be fused to have the same dimensionality. Therefore, we project them using a fully-connected layer to reduce their dimension from 2048 to 1024. After being fused together, the multimodal features are finally passed through a fully-connected layer that provides scores (*logits*) further converted into probabilities via a softmax function (S). We want to maximize the probability $P(Y = y|X = x, Q = q)$ of the correct answer y given the image X and the provided question Q. Our models are trained to choose within a set comprised by the 3000 most frequent answers extracted from both training and validation sets of the VQA v*2.0* dataset [43].

IV. Experimental Setup

A. Dataset

For conducting this study we decided to use the VQA v*2.0* dataset [43]. It is one of the largest and most frequently used datasets for training and evaluation of models in this task, being the official dataset used in yearly challenges hosted by mainstream computer vision venues [1]. This dataset enhances the original one [4] by alleviating bias problems within the data and increasing the original number of instances.

[1]VQA Challenge: https://visualqa.org/challenge.html

VQA v*2.0* contains over $200,000$ images from MSCOCO [44], over 1 million questions and ≈ 11 million answers. In addition, it has at least two questions per image, which prevents the model from answering the question without considering the input image. We follow VQA v*2.0* standards and adopt the official provided splits allowing for fair comparison with other approaches. The splits we use are Validation, Test-Dev, Test-Standard.

In this work, results of the ablation experiments are reported on the Validation set, which is the default option used for this kind of experiment. In some experiments we also report the training set accuracy to verify evidence of overfitting due to excessive model complexity. Training data has a total of $443,757$ questions labeled with 4 million answers, while the Test-Dev has a total of $214,354$ questions. Note that the validation size is about 4-fold larger than ImageNet's, which contains about $50,000$ samples. Therefore, one must keep in mind that even small performance gaps might indicate quite significant results improvement. For instance, 1% accuracy gains depict $\approx 2,000$ additional instances being correctly classified. We submit the predictions of our best models to the online evaluation servers [45] so as to obtain results for the Test-Standard split, allowing for a fair comparison to state-of-the-art approaches.

B. Evaluation Metric

Free and open-ended questions result in a diverse set of possible answers [4]. For some questions, a simple *yes* or *no* answer may be sufficient. Other questions, however, may require more complex answers. In addition, it is worth noticing that multiple answers may be considered correct, such as *gray* and *light gray*. Therefore, VQA v*2.0* provides ten ground-truth answers for each question. These answers were collected from ten different randomly-chosen humans.

The evaluation metric used to measure model performance in the open-ended Visual Question Answering task is a particular kind of accuracy. For each question in the input dataset, the model's most likely response is compared to the ten possible answers provided by humans in the dataset associated with that question [4], and evaluated according to Equation 1. In this approach, the prediction is considered totally correct only if at least 3 out of 10 people provided that same answer.

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

C. Hyper-parameters

As in [21] we train our models in a classification-based manner, in which we minimize the cross-entropy loss calculated with an image-question-answer triplet sampled from the training set. We optimize the parameters of all VQA models using Adamax [46] optimizer with a base learning rate of 7×10^{-4}, with exception of BERT [15] in which we apply a 10-fold reduction as suggested in the original paper. We used a learning rate warm-up schedule in which we halve the base learning rate and linearly increase it until the fourth epoch where it reaches twice its base value. It remains the same

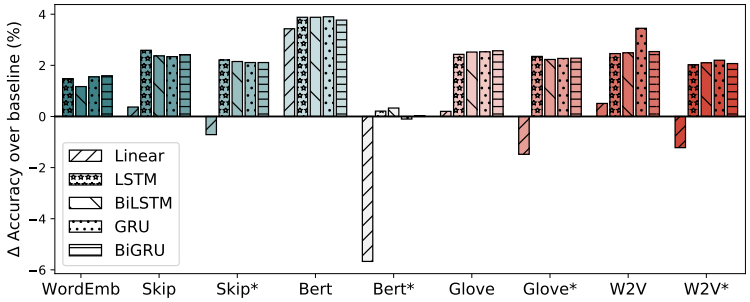

Fig. 2. Overall validation accuracy improvement (Δ) over the baseline architecture. Models denoted with * present fixed word-embedding representations, i.e., they are not updated via back-propagation.

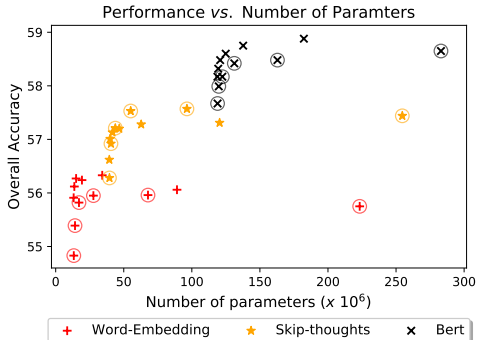

Fig. 3. Overall accuracy *vs.* number of parameters trade-off analysis. Circled markers denote two-layered RNNs. Number of parameters increases due to the number of hidden units H within the RNN. In this experiment we vary $H \in \{128, 256, 512, 1024, 2048\}$.

until the tenth epoch, where we start applying a 25% decay every two epochs. Gradients are calculated using batch sizes of 64 instances, and we train all models for 20 epochs.

V. Experimental Analysis

In this section we show the experimental analysis for each component in the baseline VQA model. We also provide a summary of our findings regarding the impact of each part. Finally, we train a model with all the components that provide top results and compare it against state-of-the-art approaches.

A. Text Encoder

In our first experiment, we analyze the impact of different embeddings for the textual representation of the questions. To this end, we evaluate: (i) the impact of word-embeddings (pre-trained, or trained from scratch); and (ii) the role of the temporal encoding function, i.e., distinct RNN types, as well as pre-trained sentence encoders (e.g., Skip-Thoughts, BERT).

The word-embedding strategies we evaluate are Learnable Word Embedding (randomly initialized and trained from scratch), Word2Vec [13], and GloVe [12]. We also use word-level representations from widely used sentence embeddings strategies, namely Skip-Thoughts [14] and BERT [15]. To do so, we use the hidden-states from the Skip-thoughts GRU network, while for BERT we use the activations of the last layer as word-level information. Those vectors feed an RNN that encodes the temporal sequence into a single global vector. Different types of RNNs are also investigated for encoding textual representation, including LSTM [47], Bidirectional LSTM [48], GRU [49], and Bidirectional GRU. For bidirectional architectures we concatenate both forward and backward hidden-states so as to aggregate information from both directions. Those approaches are also compared to a linear strategy, where we use a fully-connected layer followed by a global average pooling on the temporal dimension. The linear strategy discards any order information so we can demonstrate the role of the recurrent network as a temporal encoder to improve model performance.

Figure 2 shows the performance variation of different types of word-embeddings, recurrent networks, initialization strategies, and the effect of fine-tuning the textual encoder. Clearly, the linear layer is outperformed by any type of recurrent layer. When using Skip-Thoughts the difference reaches 2.22%, which accounts for almost $5,000$ instances that the linear model mistakenly labeled. The only case in which the linear approach performed well is when trained with BERT. That is expected since Transformer-based architectures employ several attention layers that present the advantage of achieving the total receptive field size in all layers. While doing so, BERT also encodes temporal information with special positional vectors that allow for learning temporal relations. Hence, it is easier for the model to encode order information within word-level vectors without using recurrent layers.

For the Skip-Thoughts vector model, considering that its original architecture is based on GRUs, we evaluate both the randomly initialized and the pre-trained GRU of the original model, described as [GRU] and [GRU (skip)], respectively. We noticed that both options present virtually the same performance. In fact, GRU trained from scratch performed 0.13% better than its pre-trained version.

Analyzing the results obtained with pre-trained word embeddings, it is clear that GloVe obtained consistently better results than the Word2Vec counterpart. We believe that GloVe vectors perform better given that they capture not only local

context statistics as in Word2Vec, but they also incorporate global statistics such as co-occurrence of words.

One can also observe that the use of different RNNs models inflicts minor effects on the results. It might be more advisable to use GRU networks since they halve the number of trainable parameters when compared to the LSTMs, albeit being faster and consistently presenting top results. Note also that the best results for Skip-Thoughts, Word2Vec, and GloVe were all quite similar, without any major variation regarding accuracy.

The best overall result is achieved when using BERT to extract the textual features. BERT versions using either the linear layer or the RNNs outperformed all other pre-trained embeddings and sentence encoders. In addition, the overall training accuracy for BERT models is not so high compared to all other approaches. That might be an indication that BERT models are less prone to overfit training data, and therefore present better generalization ability.

Results make it clear that when using BERT, one must fine-tune it for achieving top performance. Figure 2 shows that it is possible to achieve a 3% to 4% accuracy improvement when updating BERT weights with $1/10$ of the base learning rate. Moreover, Figure 3 shows that the use of a pre-training strategy is helpful, once Skip-thoughts and BERT outperform trainable word-embeddings in most of the evaluated settings. Is also make clear that using a single-layered RNNs provide best results, and are far more efficient in terms of parameters.

B. Image Encoder

Experiments in this section analyze the visual feature extraction layers. The baseline uses the Faster-RCNN [16] network, and we will also experiment with other pre-trained neural networks to encode image information so we can observe their impact on predictive performance. Additionally to Faster-RCNN, we experiment with two widely used networks for VQA, namely ResNet-101 [18] and VGG-16 [17].

Table I illustrates the result of this experiment. Intuitively, visual features provide a larger impact on model's performance. The accuracy difference between the best and the worst performing approaches is $\approx 5\%$. That difference accounts for roughly $10,000$ validation set instances. VGG-16 visual features presented the worst accuracy, but that was expected since it is the oldest network used in this study. In addition, it is only sixteen layers deep, and it has been shown that the depth of the network is quite important to hierarchically encode complex structures. Moreover, VGG-16 architecture encodes all the information in a 4096 dimensional vector that is extracted after the second fully-connected layer at the end. That vector encodes little to none spatial information, which makes it almost impossible for the network to answer questions on the spatial positioning of objects.

ResNet-101 obtained intermediate results. It is a much deeper network than VGG-16 and it achieves much better results on ImageNet, which shows the difference of the the learning capacity of both networks. ResNet-101 provides information encoded in 2048 dimensional vectors, extracted from the global average pooling layer, which also summarizes spatial information into a fixed-sized representation.

The best result as a visual feature extractor was achieved by the Faster-RCNN fine-tuned on the Visual Genome dataset. Such a network employs a ResNet-152 as backbone for training an RPN-based object detector. In addition, given that it was fine-tuned on the Visual Genome dataset, it allows for the training of robust models suited for general feature extraction. Hence, differently from the previous ResNet and VGG approaches, the Faster-RCNN approach is trained to detect objects, and therefore one can use it to extract features from the most relevant image regions. Each region is encoded as a 2048 dimensional vector. They contain rich information regarding regions and objects, since object detectors often operate over high-dimensional images, instead of resized ones (e.g., 256×256) as in typical classification networks. Hence, even after applying global pooling over regions, the network still has access to spatial information because of the pre-extracted regions of interest from each image.

C. Fusion strategy

In order to analyze the impact that the different fusion methods have on the network performance, three simple fusion mechanisms were analyzed: element-wise multiplication, concatenation, and summation of the textual and visual features.

The choice of the fusion component is essential in VQA architectures, since its output generates multi-modal features used for answering the given visual question. The resulting multi-modal vector is projected into a 3000-dimensional label space, which provides a probability distribution over each possible answer to the question at hand [40].

Table II presents the experimental results with the fusion strategies. The best result is obtained using the element-wise multiplication. Such an approach functions as a filtering strategy that is able to scale down the importance of irrelevant dimensions from the visual-question feature vectors. In other words, vector dimensions with high cross-modal affinity will have their magnitudes increased, differently from the uncorrelated ones that will have their values reduced. Summation does provide the worst results overall, closely followed by the concatenation operator. Moreover, among all the fusion

TABLE I
IMPACT OF THE NETWORK USED FOR VISUAL FEATURE EXTRACTION.

Embedding	RNN	Network	Training	Validation
BERT	GRU	Faster	79.34	**58.88**
		ResNet-101	76.14	56.09
		VGG-16	65.59	53.49

TABLE II
EXPERIMENT USING DIFFERENT FUSION STRATEGIES.

Embedding	RNN	Fusion	Training	Validation
BERT	GRU	Mult	78.28	**58.75**
		Concat	67.85	55.07
		Sum	68.21	54.93

strategies used in this study, multiplication seems to ease the training process as it presents a much higher training set accuracy ($\approx 11\%$ improvement) as well.

D. Attention Mechanism

Finally, we analyze the impact of different attention mechanisms, such as Top-Down Attention [20] and Co-Attention [19]. These mechanisms are used to provide distinct image representations according to the asked questions. Attention allows the model to focus on the most relevant visual information required to generate proper answers to the given questions. Hence, it is possible to generate several distinct representations of the same image, which also has a data augmentation effect.

1) Top-Down Attention: Top-down attention, as the name suggests, uses global features from questions to weight local visual information. The global textual features $\mathbf{q} \in \mathbb{R}^{2048}$ are selected from the last internal state of the RNN, and the image features $V \in \mathbb{R}^{k \times 2048}$ are extracted from the Faster-RCNN, where k represents the number of regions extracted from the image. In the present work we used $k = 36$. The question features are linearly projected so as to reduce its dimension to 512, which is the size used in the original paper [20]. Image features are concatenated with the textual features, generating a matrix C of dimensions $k \times 2560$. Features resulting from that concatenation are first non-linearly projected with a trainable weight matrix $W_1^{2560 \times 512}$ generating a novel multimodal representation for each image region:

$$\hat{C} = \phi(CW_1) \tag{2}$$

Therefore, such a layer learns image-question relations, generating $k \times 512$ features that are transformed by an activation function ϕ. Often, ϕ is ReLU [50], Tanh [51], or Gated Tanh [52]. The latter employs both the logistic Sigmoid and Tanh, in a gating scheme $\sigma(x) \times \text{TANH}(x)$. A second fully-connected layer is employed to summarize the 512-dimensional vectors into h values per region ($k \times h$). It is usual to use a small value for h such as $\{1, 2\}$. The role of h is to allow the model to produce distinct attention maps, which is useful for understanding complex sentences that require distinct viewpoints. Values produced by this layer are normalized with a SOFTMAX function applied on the columns of the matrix, as follows.

$$A = \text{SOFTMAX}(\hat{C}W_2) \tag{3}$$

TABLE III
EXPERIMENT USING DIFFERENT ATTENTION MECHANISMS.

Embedding	RNN	Attention	Training	Validation
BERT	GRU	-	78.20	58.75
		Co-Attention	71.10	58.54
		Co-Attention (L2 norm)	86.03	64.03
		Top Down	82.64	62.37
		Top Down (σ =ReLU)	87.02	**64.12**

It generates an attention mask $A^{k \times h}$ used to weight image regions, producing the image vector $\hat{\mathbf{v}}$, as shown in Equation 4.

$$\hat{\mathbf{v}_j} = \sum_i^k V_{i,\cdot} A_{ij} \tag{4}$$

Note that when $h > 1$, the dimensionality of the visual features increases h-fold. Hence, $\hat{\mathbf{v}}^{h \times 2048}$, which we reshape to be a $(2048 \times h) \times 1$ vector, constitutes the final question-aware image representation.

2) Co-Attention: Unlike the Top-Down attention mechanism, Co-Attention is based on the computation of local similarities between all questions words and image regions. It expects two inputs: an image feature matrix $V^{k \times 2048}$, such that each image feature vector encodes an image region out of k; and a set of word-level features $Q^{n \times 2048}$. Both V and Q are normalized to have unit L_2 norm, so their multiplication VQ^T results in the cosine similarity matrix used as guidance for generating the filtered image features. A context feature matrix $C^{k \times 2048}$ is given by:

$$C^T = Q^T(QV^T) \tag{5}$$

Finally, C is normalized with a SOFTMAX function, and the k regions are summed so as to generate a 1024-sized vector $\hat{\mathbf{v}}$ to represent relevant visual features V based on question Q:

$$\hat{\mathbf{v}} = \sum_i^k \text{SOFTMAX}(C)_i \tag{6}$$

Table III depicts the results obtained by adding the attention mechanisms to the baseline model. For these experiments we used only element-wise multiplication as fusion strategy, given that it presented the best performance in our previous experiments. We observe that attention is a crucial mechanism for VQA, leading to an $\approx 6\%$ accuracy improvement.

The best performing attention approach was Top-Down attention with ReLU activation, followed closely by Co-Attention. We noticed that when using Gated Tanh within Top-Down attention, results degraded 2%. In addition, experiments show that L_2 normalization is quite important in Co-Attention, providing an improvement of almost 6%.

VI. FINDINGS SUMMARY

The experiments presented in Section V-A have shown that the best text encoder approach is fine-tuning a pre-trained BERT model with a GRU network trained from scratch.

In Section V-B we performed experiments for analyzing the impact of pre-trained networks to extract visual features, among them Faster-RCNN, ResNet-101, and VGG-16. The best result was using a Faster-RCNN, reaching a 3% improvement in the overall accuracy.

We analyzed different ways to perform multimodal feature fusion in Section V-C. In this sense, the fusion mechanism that obtained the best result was the element-wise product. It provides $\approx 3\%$ higher overall accuracy when compared to the other fusion approaches.

TABLE IV
COMPARISON OF THE MODELS ON VQA2 TEST-STANDARD SET. THE MODELS WERE TRAINED ON THE UNION OF VQA 2.0 TRAINVAL SPLIT AND VISUALGENOME [38] TRAIN SPLIT. *All* IS THE OVERALL OPENENDED ACCURACY (HIGHER IS BETTER). *Yes/No*, *Numbers*, AND *Others* ARE SUBSETS THAT CORRESPOND TO ANSWERS TYPES. * SCORES REPORTED FROM [21].

Model	VQA2.0 Test-Dev				VQA2.0 Test-Std			
	All	Yes/No	Num.	Other	All	Yes/No	Num.	Other
MCB* [23]	-	-	-	-	62.27	78.82	38.28	53.36
ReasonNet* [53]	-	-	-	-	64.61	78.86	41.98	57.39
Tips&Tricks* [54]	65.32	81.82	44.21	56.05	65.67	82.20	43.90	56.26
BLOCK* [21]	67.58	83.60	47.33	58.51	67.92	83.98	46.77	58.79
BERT-GRU-Faster-TopDown	67.16	84.76	44.82	57.23	67.28	84.75	44.90	57.20
BERT-GRU-Faster-CoAttention	67.18	84.85	45.92	56.84	67.39	85.00	46.20	56.91

TABLE V
COMPARISON OF THE MODELS ON VQA2 TEST-DEV SET. *All* IS THE OVERALL OPENENDED ACCURACY (HIGHER IS BETTER). *Yes/No*, *Numbers*, AND *Others* ARE SUBSETS THAT CORRESPOND TO ANSWERS TYPES. * SCORES REPORTED FROM [21].

Model	VQA2.0 Test-Dev			
	All	Yes/No	Num.	Other
DEEPER-LSTM-Q [4]	51.95	70.42	32.28	40.64
MCB* [23]	61.23	79.73	39.13	50.45
BLOCK* [21]	66.41	82.86	44.76	57.30
BERT-GRU-Faster-CoAttention	65.84	83.66	44.36	55.50
BERT-GRU-Faster-TopDown	66.02	83.72	44.88	55.77

Finally, in Section V-D we have studied two main attention mechanisms and their variations. They aim to provide question-aware image representation by attending to the most important spatial features. The top performing mechanism is the Top-Down attention with the ReLU activation function, which provided an $\approx 6\%$ overall accuracy improvement when compared to the base architecture.

VII. COMPARISON TO STATE-OF-THE-ART METHODS

After evaluating individually each component in a typical VQA architecture, our goal in this section is to compare the approach that combines the best performing components into a single model with the current state-of-the-art in VQA. Our comparison involves the following VQA models: DEEPER-LSTM-Q [4], MCB [23], ReasonNet [53], Tips&Tricks [54], and the recent BLOCK [21].

Tables IV and V show that our best architecture outperforms all competitors but BLOCK, in both Test-Standard (Table IV) and Test-Dev sets (Table V). Despite BLOCK presenting a marginal advantage in accuracy, we have shown in this paper that by carefully analyzing each individual component we are capable of generating a method, without any bells and whistles, that is on par with much more complex methods. For instance, BLOCK and MCB require 18M and 32M parameters respectively for the fusion scheme alone, while our fusion approach is parameter-free. Moreover, our model performs far better than [23], [53], and [54], which are also arguably much more complex methods.

VIII. CONCLUSION

In this study we observed the actual impact of several components within VQA models. We have shown that transformer-based encoders together with GRU models provide the best performance for question representation. Notably, we demonstrated that using pre-trained text representations provide consistent performance improvements across several hyper-parameter configurations. We have also shown that using an object detector fine-tuned with external data provides large improvements in accuracy. Our experiments have demonstrated that even simple fusion strategies can achieve performance on par with the state-of-the-art. Moreover, we have shown that attention mechanisms are paramount for learning top performing networks, once they allow producing question-aware image representations that are capable of encoding spatial relations. It became clear that Top-Down is the preferred attention method, given its results with ReLU activation. It is is now clear that some configurations used in some architectures (e.g., additional RNN layers) are actually irrelevant and can be removed altogether without harming accuracy. For future work, we expect to expand this study in two main ways: (i) cover additional datasets, such as Visual Genome [38]; and (ii) study in an exhaustive fashion how distinct components interact with each other, instead of observing their impact alone on the classification performance.

ACKNOWLEDGMENT

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nivel Superior – Brasil (CAPES) – Finance Code 001. We also would like to thank FAPERGS for funding this research. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the graphics cards used for this research.

REFERENCES

[1] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2625–2634.

[2] H. Fang, S. Gupta, F. Iandola, R. K. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J. C. Platt *et al.*, "From captions to visual concepts and back," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1473–1482.

[3] X. Chen and C. Lawrence Zitnick, "Mind's eye: A recurrent visual representation for image caption generation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2422–2431.

[4] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh, "Vqa: Visual question answering," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2425–2433.

[5] J. Singh, V. Ying, and A. Nutkiewicz, "Attention on attention: Architectures for visual question answering (vqa)," *arXiv preprint arXiv:1803.07724*, 2018.

[6] J. Wehrmann, C. Kolling, and R. C. Barros, "Adaptive cross-modal embeddings for image-text alignment," in *AAAI 2020*, 2018, pp. 7718–7726.

[7] J. Wehrmann, D. M. Souza, M. A. Lopes, and R. C. Barros, "Language-agnostic visual-semantic embeddings," in *ICCV 2019*, 2019.

[8] 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 MultiMedia*, vol. 21, no. 2, pp. 42–70, 2014.

[9] J. P. Bigham, C. Jayant, H. Ji, G. Little, A. Miller, R. Miller, R. Miller *et al.*, "Vizwiz: nearly real-time answers to visual questions," in *Proceedings of the 23nd ACM Symposium on User Interface Software and Technology*, 2010, pp. 333–342.

[10] W. S. Lasecki, Y. Zhong, and J. P. Bigham, "Increasing the bandwidth of crowdsourced visual question answering to better support blind users," in *Proceedings of the 16th international ACM SIGACCESS conference on Computers & accessibility*. ACM, 2014, pp. 263–264.

[11] H. Xu and K. Saenko, "Ask, attend and answer: Exploring question-guided spatial attention for visual question answering," in *European Conference on Computer Vision*. Springer, 2016, pp. 451–466.

[12] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[13] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[14] R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, and S. Fidler, "Skip-thought vectors," in *Advances in neural information processing systems*, 2015, pp. 3294–3302.

[15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[16] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

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

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[19] K.-H. Lee, X. Chen, G. Hua, H. Hu, and X. He, "Stacked cross attention for image-text matching," in *ECCV*, 2018, pp. 201–216.

[20] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, "Bottom-up and top-down attention for image captioning and visual question answering," in *CVPR*, 2018, pp. 6077–6086.

[21] H. Ben-Younes, R. Cadene, N. Thome, and M. Cord, "Block: Bilinear superdiagonal fusion for visual question answering and visual relationship detection," *arXiv preprint arXiv:1902.00038*, 2019.

[22] R. Cadene, H. Ben-Younes, M. Cord, and N. Thome, "Murel: Multimodal relational reasoning for visual question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1989–1998.

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

[24] Y. Gao, O. Beijbom, N. Zhang, and T. Darrell, "Compact bilinear pooling," in *CVPR*, 2016, pp. 317–326.

[25] J. Lu, J. Yang, D. Batra, and D. Parikh, "Hierarchical question-image co-attention for visual question answering," in *NIPS*, 2016, pp. 289–297.

[26] M. Malinowski, M. Rohrbach, and M. Fritz, "Ask your neurons: A neural-based approach to answering questions about images," in *ICCV*, 2015, pp. 1–9.

[27] A. Mallya and S. Lazebnik, "Learning models for actions and person-object interactions with transfer to question answering," in *ECCV*. Springer, 2016, pp. 414–428.

[28] C. Xiong, S. Merity, and R. Socher, "Dynamic memory networks for visual and textual question answering," in *ICML*, 2016, pp. 2397–2406.

[29] M. Cornia, M. Stefanini, L. Baraldi, and R. Cucchiara, "M2: Meshed-memory transformer for image captioning," *arXiv preprint arXiv:1912.08226*, 2019.

[30] J. Wehrmann and R. C. Barros, "Bidirectional retrieval made simple," in *CVPR 2018*, 2018, pp. 7718–7726.

[31] J. Wehrmann, C. Kolling, and R. C. Barros, "Fast and efficient text classification with class-based embeddings," in *IJCNN 2019*. IEEE, 2017, pp. 2384–2391.

[32] J. Wehrmann and R. C. Barros, "Convolutions through time for multi-label movie genre classification," in *SAC 2017*, 2017, pp. 114–119.

[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017, pp. 5998–6008.

[34] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola, "Stacked attention networks for image question answering," in *CVPR*, 2016, pp. 21–29.

[35] K. J. Shih, S. Singh, and D. Hoiem, "Where to look: Focus regions for visual question answering," in *CVPR*, 2016, pp. 4613–4621.

[36] Z. Yu, J. Yu, C. Xiang, J. Fan, and D. Tao, "Beyond bilinear: Generalized multimodal factorized high-order pooling for visual question answering," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 12, pp. 5947–5959, 2018.

[37] A. Burns, R. Tan, K. Saenko, S. Sclaroff, and B. A. Plummer, "Language features matter: Effective language representations for vision-language tasks," *arXiv preprint arXiv:1908.06327*, 2019.

[38] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma *et al.*, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *International Journal of Computer Vision*, vol. 123, no. 1, pp. 32–73, 2017.

[39] Y. Bai, J. Fu, T. Zhao, and T. Mei, "Deep attention neural tensor network for visual question answering," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 20–35.

[40] B. Duke and G. W. Taylor, "Generalized hadamard-product fusion operators for visual question answering," in *2018 15th Conference on Computer and Robot Vision (CRV)*. IEEE, 2018, pp. 39–46.

[41] H. Ben-Younes, R. Cadene, M. Cord, and N. Thome, "Mutan: Multimodal tucker fusion for visual question answering," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2612–2620.

[42] J.-H. Kim, K.-W. On, W. Lim, J. Kim, J.-W. Ha, and B.-T. Zhang, "Hadamard product for low-rank bilinear pooling," *arXiv preprint arXiv:1610.04325*, 2016.

[43] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh, "Making the v in vqa matter: Elevating the role of image understanding in visual question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6904–6913.

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

[45] D. Yadav, R. Jain, H. Agrawal, P. Chattopadhyay, T. Singh, A. Jain, S. B. Singh, S. Lee, and D. Batra, "Evalai: Towards better evaluation systems for ai agents," *arXiv preprint arXiv:1902.03570*, 2019.

[46] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

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

[48] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[49] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "*Learning phrase representations using RNN encoder-decoder for statistical machine translation*," *arXiv preprint arXiv:1406.1078*, 2014.

[50] A. F. Agarap, "Deep learning using rectified linear units (relu)," *arXiv preprint arXiv:1803.08375*, 2018.

[51] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural networks: Tricks of the trade*. Springer, 2012, pp. 9–48.

[52] W. Xue and T. Li, "Aspect based sentiment analysis with gated convolutional networks," *arXiv preprint arXiv:1805.07043*, 2018.

[53] I. Ilievski and J. Feng, "Multimodal learning and reasoning for visual question answering," in *NIPS 2017*, 2017, pp. 551–562.

[54] D. Teney, P. Anderson, X. He, and A. van den Hengel, "Tips and tricks for visual question answering: Learnings from the 2017 challenge," in *CVPR 2018*, 2018, pp. 4223–4232.