Arabic Visual Question Answering

Mariam Safieldin
mas177@mail.aub.edu

Supervisor: Professor Shady Elbassuoni

This document is a report for the final project of the Advanced Machine Learning course CMPS 392.

***Abstract*—Visual Question Answering (VQA) is a system that generates an answer based on considering multiple dissimilar tasks related to both visual and textual interpretation of a question and an image all inside a single cohesive framework. VQA provides opportunities to better interpretation visual features in many domains. In this project, two visual question answering models that answer an Arabic question given an image was implemented, and compared against a baseline model. Experiments performed on the VQA dataset, which is a benchmark dataset for visual question answering, show that the models implemented result in a better accuracy in comparison to the baseline model and subset of previous works towards the same problem.**

I. Introduction

A Visual Question Answering (VQA) system answers a textual question asked about an input image to the system. This is a complex task as the VQA model must comprehend the image, understand its features, understand the given question, comprehend its textual features, and most importantly relate these visual and textual features together to produce the correct answer. VQA is not limited to one concept only. The model should be able to answer any question related to an image, and many question can be asked to the model given the same image. Accordingly, a VQA model should have diverse reasoning capabilities to capture relevanr data from input images and questions, and consquently be able to provide answers about object recognition, color recognition, locations, and counting [1].

Convolutional neural networks (CNNs) have shown a quite impressive promise in solving vision related tasks, also, recurrent neural networks (RNNs) have shown a considerable promise in solving language and textual related tasks. Both CNNs and RNNs have been recently used together to solve multi-modal tasks such as visual question answering, as VQA requires models to answer natural language questions about the visual content provided to the algorithms [2].

VQA systems have diverse beneficial applications. VQA systems can assist the interaction of visually impaired users with images. In addition, VQA systems can help in the analysis of visual data [3]. Despite the fact that VQA entails a variety of tasks that are defined by questions, current algorithms train a universal model generalized for all possible questions. This project introduces two different model architectures for VQA based on conclusions and analysis made by survey papers on previous work [3][4]. In addition, this work provides a visual question answering system in the Arabic language. Figure 1 below clarifies our target case showing the input and output of our proposed approach.

The remainder of the paper is organized as follows: In Section II a subset of the previous work is reported and compared to our work. Section III reports the proposed methodology, proposed experiment, and the dataset we wish to use in this paper. In Section IV, dataset used, experimental setup, and comparison to the baseline and previous work are discussed. Finally, the conclusion and future work are presented in Section V.

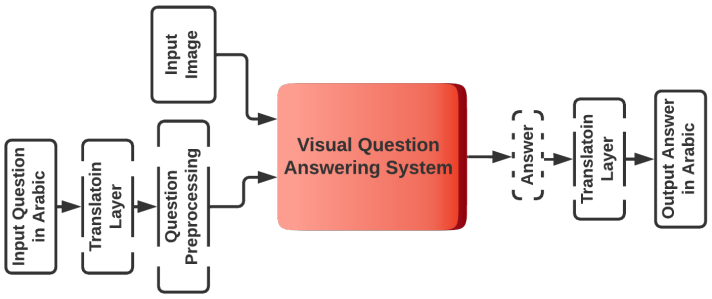

Fig. 1: Proposed Arabic Visual Question Answering System

II. Related Work

A dynamic memory network is introduced in [2] which consists of four parts. First, the input module which converts the data into facts. These facts are a set of vectors implemented based on the type of input data. Secondly, a vector representation of the question is computed using the question module which uses a gated recurrent unit (GRU). Thirdly, there is the episodic memory module which passes through the facts numerous times to achieve a transitive reasoning, and consequently retrieve the facts needed to respond to a specific question which is carried out through an attention strategy that chooses the most relevant facts, and then the update mechanism creates a new memory representation based on interactions between the current state and the facts retrieved. The question module provides an initialization for the first state. Finally, the answer module predicts the result based on the memory's final state and the question using either a multinomial classification for single words, or a GRU for datasets requiring a longer phrase.

A CNN-only system is presented in [5]. The system consists of 3 CNNs. The input image is encoded using a CNN model, and similarly, the input question is encoded using a CNN model, then both encodings are joined together using a third CNN model. The CNN model used for encoding the images has a similar architecture to the VGGnet [9] which derives a 4096-length vector from the network's second-last layer, which sent through another fully connected layer that output a 400-pixel vector representation of the image. The CNN model used to encode the question contains three convolutional layers followed by a maximum pooling layer. The kernal looks at a word along with its direct neighbors through setting the convolutional receptive field to 3. The final multi-modal CNN uses convolution with receptive field of size 2 across the question's representation. Every convolutional operation takes the image representation as an input. Finally, to predict the answer, a

softmax layer receives the final representation from the multi-modal CNN.

Ben-younes et al. [6] introduced a model named MUTAN which uses a ResNet-152 model to extract the image features, and a Gated Recurrent Units (GRU) recurrent network initialized with Skip-thought vectors to extract the question features. The extracted features are joined together through a Tucker decomposition of correlation tensor, which is a strategy they introduced to the aim of parametrizing the bi-linear interactions between the visual and textual encodings. The final feature vector is fed to a softmax layer for producing the output answer.

Yang et al. introduced the Stacked Attention Network (SAN) [7] which extracts a feature map using the final layer of VGG19 pretrained model. The question features are extracted using a CNN or an LSTM. The attention distribution across image locations is computed through using a softmax activation function with the extracted features represented as a single layer of weights. The proposed approach uses the top 1000 most frequent answer as possible outputs and this set of answers encompass 82.67% of all answers.

A. Fukui et al. introduced a strategy for combining image features and question features under the name of Multimodal Compact Bilinear (MCB) pooling [8]. The MCB pooling layer combines the feature vectors through approximating their product in a space with lower dimensions to the aim of obtaining a more intense interaction between the features vectors. The output of this process is used to extract the spatial features that are significant to the question. ResNet152 pretrained model was used to extract the image features. Finally, the feature vector is fed to a 1000-way classifier to generate an answer.

Shrestha et al. proposed a VQA system that extracts the image features using a Faster R-CNN. The question features are extracted using GRU initialized with Glove word embeddings [10]. The features extracted are joined using concatenating the textual features with regional image features, and afterwards they use bidirectional GRU to aggregate bimodal embedding. The aggregated embeddings are then fed to an answer classifier.

III. Methodologies

In this section, we introduce two deep learning models implemented for the visual question answering task. The models are composed of two main stages which are the image featurization and the question featurization models, followed by the joint phase, and finally outputting the final answer. The first model III-A uses a CNN of 7 layers for the image featurization, and for the question featurization it employs an embedding layer and 3 Long Short-Term Memory models (LSTMs). The second model III-B uses VGG16 pretrained model [9] for image featurization, and it uses bidirectional LSTMs with an attention mechanism for the question features extraction model. Concatenation of features is used as the joint procedure for the image and question features for both models. The input to the system is an image and a question in Arabic as shown in Figure 1, the question passes through a translation layer III-C then the output is processed as described in IV-B. Then, the preprocessed question and the image are passed to the VQA deep learning model to output an answer in English which is fed to the translation layer to output a final answer in Arabic.

A. First Model Architecture - CNNs + LSTMs

For the question featurization model, an embedding layer of dimension (300,) using Glove [10] word embeddings was used to obtain a good representation of the data since the dataset used is rich in vocabulary. Three LSTMs were used to extract the textual features in order to produce higher level features as the input values progress from layer to layer, since LSTMs work with sequence data, layering adds degrees of abstraction to the input observations over time. Essentially, chunking data throughout time or expressing the problem on several timeframes. The final question features extraction model is shown in Figure 2.

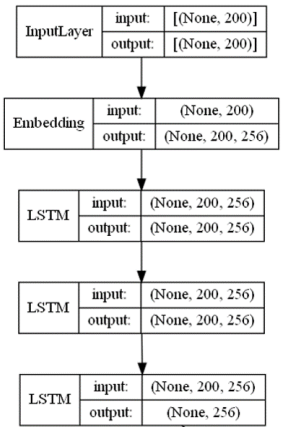

Fig. 2: III-A Question Feature Extraction Model Architecture

For the images features extraction model, a CNN that is composed of total of 7 convolution layers was implemented. The CNN consists of 2 convolutional layers using 64 3x3 filters, a 2x2 Max2DPooling layer, 2 convolutional layers using 128 3x3 filters, a 2x2 Max2DPooling layer, followed by 3 convolutional layers using 256 3x3 filters, and finally followed by Max2DPooling and a final flatten layer. The architecture of this model is shown in Figure 3.

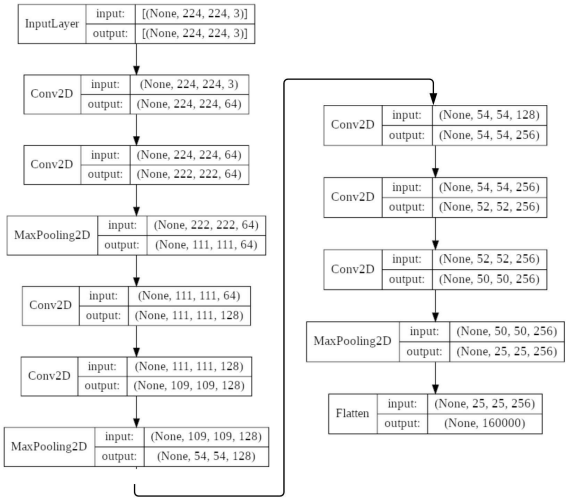

Fig. 3: III-A Image Feature Extraction Model Architecture

The image and question features are joined together through concatenation. The joined vector is then fed to a dense layer with Softmax as the activation function.

B. Second Model Architecture - VGG16 + Bi-LSTM + Attention Mechanism

Based on S. Manmadhan's and B. C. Kovoor's survey paper on VQA [4], VGG16 pretrained model [9] has proven its capability to extract image features from various datasets used in visual question answering systems. Accordingly, VGG16 pretrained model was used for the image featurization in this VQA system. L2 normalization was performed on the features extracted by the image encoder through dividing every feature by its l2 norm as it has an effect on improving the fusion performance as indicated by the authors of [11].

For the question feature extraction, Glove word embeddings of dimension (300,) were used as an input a bidirectional LSTM and the output vector was fed to an attention mechanism. The attention mechanism implemented is adapted from the attention strategies introduced in [12][13].

First, an MLP layer is applied on the hidden state of a word h_t of dimension K_w, which is the output hidden states from the BiLSTM and *t* being the position of the word in the input question and *b* being the bias term, and W being the embedding matrix of dimension K_wxK_w. The output is u_t the hidden representation of h_t.

$$u_t = tanh(Wh_t + b) \tag{1}$$

Second, then a dot product of the output u_t of the MLP and a vector v of the same dimension K_w is computed. The output of this dot product represents a word-level context vector $v^T u_t$, which is then passed to a softmax layer to produce the attention vector a_t.

$$a_t = softmax(v^T u_t) \tag{2}$$

Finally, the resulting attention vector a_t is combined with the initial the hidden state of the word h_t through element-wise multiplication to output the final vector s.

$$s = \sum_{t=1}^{M} a_t h_t \tag{3}$$

The extracted image feature vector and the feature vector extracted from the Bi-LSTM and the attention mechanism are then concatenated together and fed to a dense layer with Softmax as the activation function. The final model plot is shown in Figure III-B.

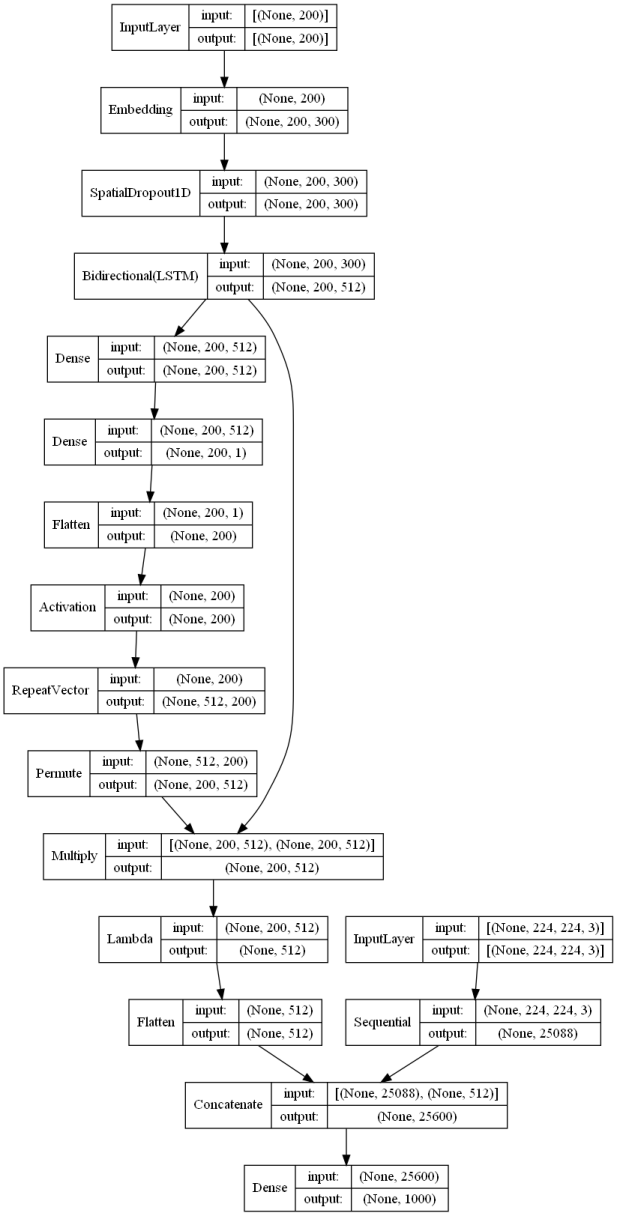

Fig. 4: III-B Model Architecture

C. Translation Layer

There exists no Arabic dataset to be used for training the models on Arabic data, and the translation attempts for the datasets were not successful given the time and the computational resources available. The followed attempts were made: First, a python library, deep-translator [14], was used to translate the questions and the answers from dataframes generated from the dataset, yet given the huge number of records available in the dataset, (214,354, 2,143,540) questions and answers for the validation dataset only, the translation consumed very long time and the python implemented translation function would disconnect after a certain number of runs due to memory issues. Second attempt was using Google Sheets, the dataframe generated from the dataset was uploaded to Google sheets, and Google's translation function was used to translate the questions and the answers, however, same memory related issues were encountered, in addition, the translation accuracy of Google Sheet's function was very low in terms of the interpretation of the meaning of the questions and the answers.

To provide a system in the Arabic language with the available dataset, a translation layer was implemented for the use on the application level. The translation layer is a python function implemented for use on the application level during the prediction of an answer. The function uses deep-translator python library [14] to translate an input Arabic sentence/question, an extension is added to translate English numbers into Arabic so that the final answers would have numbers displayed in Arabic. A similar procedure is applied to the answer generated from the VQA system, we feed to the translation layer, and the final answer is output in the Arabic language.

IV. Experiment and Result Evaluation

In this section, we will discuss the experiments conducted to train and test the implemented models, the dataset used, and the base model developed for comparison. For all the models implemented in this project, Adam optimizer was used with a learning rate of 0.001.

A. Dataset

The models implemented are evaluated using the VQA dataset, which is benchmark dataset for the visual question answering task which contains actual MSCOCO images [15]. The dataset has two versions. It has been demonstrated that models trained on VQA v1.0 have a learning bias due to the fact that questions and answers have a large correlation, accordingly the models tend to rely more on the textual features interpreted from the question than on both the visual and textual features when providing an answer to a question about a given image. To solve this issue, VQA 2.0 dataset was released. VQA v2.0 has the same target as VQA v1.0 in terms of modelling the questions and the answers, yet, version 2.0 has a strategy to eliminate this correlation between the questions and the answers through having for every question two images that have different answers based on the image given as an input, thus the model can not rely only to the question to provide the correct answer.

An average of three questions are created for each image. The questions fall under 5 categories: Y/N, Object, Number, Color, and Location. Human annotators provide ten answers for each image-question pair, and the most commonly given answer is chosen as the correct answer. There are (82,783, 443,757, 4,437,570), (40,504, 214,354, 2,143,540) MS-COCO images, questions, and answers, respectively, for the train and validation datasets. While (81,434, 447,793) images and questions are available for the testing dataset [16].

B. Data Preprocessing

All special characters, such as (@/"%$* :), along with punctuation are removed from the questions, as these characters add no value to text-understanding and induce noise into algorithms.

Contractions are expanded. Shortened versions of words or syllables are known as contractions. They are made by eliminating one or more particular letters from words. A contraction is often made up of more than one word. An apostrophe sign is used in writing to represent the absence of a letter. The conversion of the contractions to their original expanded format improves text standardization. For removing contractions, a dictionary of contractions was used to map every contraction to its expanded format. Only questions in the training and validation datasets were exposed to the contraction expansion phase, as the translation layer does not output any text with contractions.

Following [17][18][19], the visual question answering task was treated as a classification task over a predefined broad set of possible answers since the questions in the dataset are limited to certain categories as elaborated in IV-A. The frequency of each answer was calculated and saved to a dictionary of all answer frequencies, mapping every answer to its frequency. Then, the top 1000 frequent answers are selected, and consequently the questions that their answers do not exist in top 1000 frequent answers are filtered. These most 1000 frequent answers encompass approximately 82.67% of the answers in the complete VQA dataset [7][20].

The questions are then tokenized and converted to sequences, through representing every word by an identification integer. Then these identification integers are padded, with a maximum length of 200, to have an equivalent size of representation for all the words.

C. Baseline Model

A basic baseline model was implemented to be a reference point for the comparison with the other architectures implemented throughout this project.

A pre-trained VGG16 network [9] was used in our baseline model to extract the image features resulting in a feature vector of dimension (4096,). The size of the images input to the model was 224x224, which is the default input shape for the VGG16 network. While for the questions, an LSTM model was used to extract the textual features, the input to the LSTM model is the resulting embeddings of the questions after using Glove word embeddings of dimension (50,). Both the image encodings and the textual features are concatenated together. A dense layer with Softmax as the activation function learns the feature vector to outputs an answer. The baseline's model plot is shown in Figure 5.

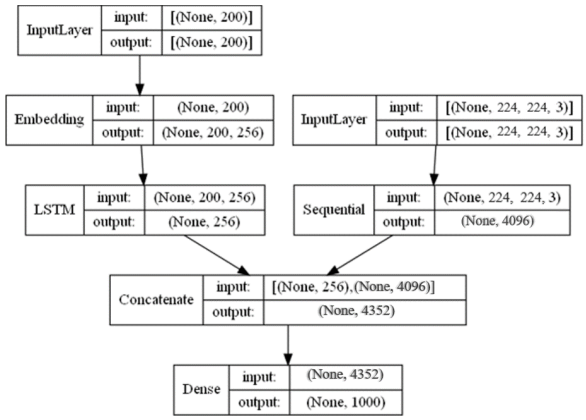

Fig. 5: Baseline Model Architecture

The baseline was trained on the complete VQA training dataset, and validated using the complete validation dataset as well using Jupyter platform. The accuracy of results obtained was 48.36% on the validation dataset. The training of this model consumed very long time and computational resources due to the huge size of the dataset used and the convolution computations performed on the images.

D. First Model Architecture - CNNs + LSTMs - Evaluation

The model introduced in III-A was evaluated using the same strategy as the baseline model using Jupyter platform. Training was performed using the entire VQA training dataset. Also validation was performed using the entire VQA validation dataset. The accuracy of the results on the validation dataset is 52.51% which is better than the accuracy obtained by the baseline model IV-C.

E. Second Model Architecture - VGG16 + Bi-LSTM + Attention Mechanism - Evaluation

This model was trained and validated on 45% of the training dataset, of which 80% used for training and 20% used for validation, in order to be able to upload the dataset on Google drive and accordingly train the model using Google Colab's GPU. The resulting accuracy for this model on the validation data was 59.68% which is quite higher than both the baseline and III-A models. A summary of the results is shown in Table I. It is worth mentioning that III-B model was trained and validated using much less data than IV-C and III-A models.

Model	Accuracy (%)
VGG16 + 1 LSTM (IV-C)	48.36
CNNs + 3 LSTMs (III-A)	52.51
VGG16 + BiLSTM + Attention Mechanism (III-B)	59.68

TABLE I: Results on Validation Data.

V. Conclusion

In this project, an Arabic visual question answering system was implemented. Two different models for the VQA system were implemented. III-A model employs CNNs and LSTMs for encoding the image and question, respectively. III-B employs VGG16 pretrained model and BiLSTM with an attention mechanism for encoding the image and question features, respectively. The resulting features are concatenated and fed to a softmax layer to output an answer that is passed to the translation layer to generate a final answer in Arabic. In our future work, we intend to train the III-B model on the entire dataset and compare the results. We also target extending the architecture of III-B model to add a decoder for the answer generation. Moreover, we target evaluating the model using different datasets. Finally, we target translating a VQA dataset to the Arabic language and train the model on Arabic data.

References

[1] Malinowski, M., Rohrbach, M., & Fritz, M. (2017). Ask Your Neurons: A Deep Learning Approach to Visual Question Answering. International Journal of Computer Vision, 125(1-3), 110–135.

[2] Kumar, A., Irsoy, O., Ondruska, P., Iyyer, M., Bradbury, J., Gulrajani, I., Zhong, V., Paulus, R., & Socher, R. (2015). Ask Me Anything: Dynamic Memory Networks for Natural Language Processing. ArXiv.org.

[3] Kafle, K., & Kanan, C. (2017). Visual question answering: Datasets, algorithms, and future challenges. Computer Vision and Image Understanding, 163, 3–20.

[4] Manmadhan, S., & Kovoor, B. C. (2020). Visual question answering: a state-of-the-art review. Artificial Intelligence Review.

[5] Ma, L., Lu, Z., & Li, H. (2015). Learning to Answer Questions From Image Using Convolutional Neural Network.

[6] Ben-Younes, H., Cadene, R., Cord, M., & Thome, N. (2017). MUTAN: Multimodal Tucker Fusion for Visual Question Answering.

[7] Yang, Z., He, X., Gao, J., Deng, L., & Smola, A. (2016). Stacked Attention Networks for Image Question Answering. ArXiv:1511.02274 [Cs].

[8] 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:1606.01847 [Cs].

[9] Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. ArXiv.org.

[10] Pennington, J., Socher, R., & Manning, C. (2014). GloVe: Global Vectors for Word Representation.

[11] Malinowski, M., Rohrbach, M., & Fritz, M. (2016). Ask Your Neurons: A Deep Learning Approach to Visual Question Answering. ArXiv:1605.02697 [Cs].

[12] Fucci, D., Romano, S., Baldassarre, M., Caivano, D., Scanniello, G., Thuran, B., & Juristo, N. (2017). A Longitudinal Cohort Study on the Retainment of Test-Driven Development.

[13] Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E. (2016). Hierarchical Attention Networks for Document Classification (pp. 1480–1489). Association for Computational Linguistics.

[14] deep-translator v1.5.5 https://pypi.org/project/deep-translator/

[15] Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Lawrence, Z. C., & Dollár, P. (2014). Microsoft COCO: Common Objects in Context. ArXiv.org.

[16] Gupta, A. K. (2017). Survey of Visual Question Answering: Datasets and Techniques. ArXiv:1705.03865 [Cs].

[17] Shrestha, R., Kafle, K., & Kanan, C. (2019). Answer Them All! Toward Universal Visual Question Answering Models. ArXiv:1903.00366 [Cs].

[18] Yu, Z., Yu, J., Cui, Y., Tao, D., & Tian, Q. (2019). Deep Modular Co-Attention Networks for Visual Question Answering. ArXiv:1906.10770 [Cs].

[19] Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., & Zhang, L. (2018, April 4). Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering.

[20] Agrawal, A., Lu, J., Antol, S., Mitchell, M., Zitnick, C. L., Parikh, D., & Batra, D. (2016). VQA: Visual Question Answering. International Journal of Computer Vision, 123(1), 4–31.