Talking To a Machine

Tejashwini Gundlapally
Information Technology
Vardhaman College of Engineering
Hyderabad, India
tejuaruna29@gmail.com

***Abstract*—Talking to a Machine is a Machine Learning Algorithm that can answer inquiries regarding an image's content. A statement, a phrase, or a single word might be used as the response. Our model has four parts: an LSTM for extracting the question representation, a CNN for extracting the visual representation, an LSTM for storing the language context in a response, and a fusing component for combining the input from the first three components and generating the answer [2] [3]. To train and assess our mQA model, we create a dataset called Freestyle Multilingual Image Question Answering (FM-IQA). It has approximately 150,000 photos as well as 310,000 freestyle question-answer pairs with English translations.Human judges use a Turing Test to assess the quality of our mQA model's produced responses on this dataset. [1]In particular, we combine human responses with our model. The human judges must be able to tell the difference between our model and the human. We provide techniques for keeping an eye on the quality of the evaluation process. Human judges cannot identify our model from humans in 64.7 percent of situations, according to the experiments. The overall average is 1.454. (1.918 for human). The FM-IQA dataset, as well as the specifics of this project.**

***Index Terms*—Recurrent Neural Network(RNN), Convolutional Neural Networks(CNN), Freestyle Multilingual Image Question Answering(FM-IQA), Long Short-Term Memory (LSTM),VGG16.**

I. Introduction

There has recently been a surge of interest in the topic of multi-modal learning, which includes both natural language and vision. Many research, in particular, have achieved rapid progress in the area of picture captioning [3]. The majority of them are based on deep neural networks (for example, deep Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Long Short-Term Memory (LSTM)) (LSTM). This advancement [4] is made possible by large-scale picture collections with sentence annotations. [5]Despite the effectiveness of these strategies, there are still a lot of questions that need to be answered. Picture captioning, in instance, simply requires general phrase descriptions of an image. However, in many circumstances, we are just interested in a certain section or item of a picture. The image captioning task lacks the interaction between the computer and the user.

The job of visual question responding is the emphasis of this study. The method must respond to a freestyle inquiry regarding the content of a picture in this challenge. To solve this problem, we offer the mQA paradigm. The model takes two inputs: a picture and a question. The model outputs in words and predicts the answer to the query.This project focuses on the task of visual question answering. In this task, the method needs to provide an answer to a freestyle question about the content of an image. We propose the mQA model to address this task [6]. The inputs of the model are an image and a question. The model predicts the answer for the question and outputs in words.

Vision-to- Language difficulties are a unique difficulty in Computer Vision because they necessitate the translation of two separate types of data [7]. In this way, the difficulty is comparable to that of language machine translation. There have been a number of breakthroughs in machine language translation that suggest that good performance may be obtained without establishing a higher-level model of the state of the world [8].

II. Objectives

It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow.

III. Literature Survey

A. Existing Systems

Deep neural network models have made substantial advances in the domains of computer vision and natural language processing in recent years. Convolutional Neural Network (CNN)-based approaches for computer vision deliver state-of-the-art performance in a variety of tasks, [8] [9]including object categorization, detection, and segmentation. Machine translation and voice recognition employ the Recurrent Neural Network (RNN) and the Long Short-Term Memory network (LSTM) for natural language.

The m-RNN model for image captioning and image-sentence retrieval tasks inspired the structure of our mQA model. For vision, it uses a deep CNN, and for language, it uses an RNN. [10]We expand the model to take question and picture pairings as input and create responses. In the studies, we discovered that by utilising the m-RNN model, we can learn how to pose a good question about an image, and that this question can then be answered by our mQA model. [11] A recent attempt has been made on the visual question answering problem. However, the majority of them employ a pre-determined and limited set of questions. A template is used to produce some of these questions. Furthermore, our FM-IQA dataset is far larger than theirs.
On this subject, there are various parallel and separate efforts. A large-scale dataset based on MS COCO is also proposed. On this dataset, [12]they also give some simple baseline approaches.

B. Limitations of Existing Systems

The existing project uses RNN and CNN to caption images, but it lacks the ability to analyse voice questions and extract solutions depending on picture configuration. They combine the query and the response before feeding it to the LSTM. Unlike them, we employ two independent LSTMs for questions and answers, [13] taking into account the varied features of questions and answers while allowing the word-embeddings to be shared [14].

IV. Proposed Methodology

LSTM and CNN are used to implement the model.They combine the query and the response before feeding it to the LSTM. Unlike them, we employ two independent LSTMs for questions and answers, taking into account the varied features of questions and answers while allowing the word-embeddings to be shared. Use the dataset offered, which is significantly smaller than our FM-IQA dataset. Using MS COCO's annotations, create a dataset with four pre-defined categories of questions (i.e. object, number, color, and location). It also summarises the response in a single word.

1) Extracting Image features with VGG16
   The VGG16 model from tensorflow keras is initially imported. The pre-process input module is imported to scale pixel values properly for the VGG16 model, and the image module is loaded to preprocess the picture object. The numpy module is used to handle arrays. The imagenet dataset's pre-trained weights are then fed into the VGG16 model. A convolutional layer is followed by one or more dense (or completely linked) layers in the VGG16 model. The feature extraction part of the model runs from the input layer to the last max pooling layer (labelled by 7 x 7 x 512), while the classification part of the model runs from the input layer to the [16] last max pooling layer (labelled by 7 x 7 x 512).

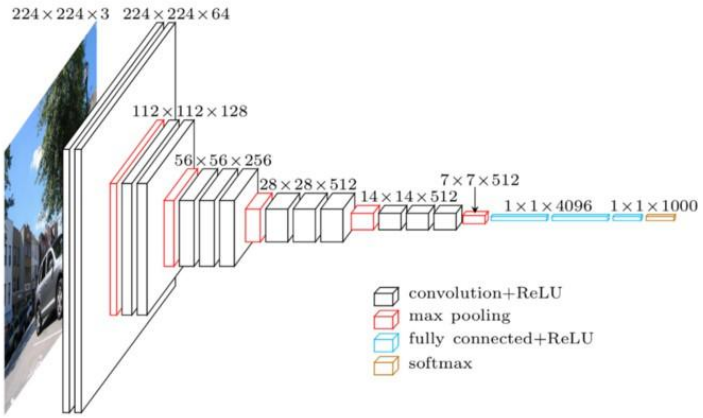


Fig. 1. Feature Extraction using VGG16

2) Training Data-points
   We employed a CNN that had already been trained on the ImageNet classification assignment. During the QA training, this component is corrected. We use a log-likelihood loss based on the answer's word sequence. Minimizing this loss function is the same as increasing the model's chances of generating groundtruth responses in the training set. Using the stochastic gradient descent approach, we jointly train the LSTM and CNN.

3) Visual Turing Test
   A human judge will be provided with a picture, a question, and the answer to the question created by the testing model or human annotators in the Visual Turing Test. He or she must decide whether the answer was supplied by a person (i.e. pass the exam) or a computer based on the response.
4) Score of the Generated Answer
   The Visual Turing Test only provides a preliminary assessment of the generated responses. We also do a fine-grained assessment using scores of "0," "1," and "2." [17] The numbers "0" and "2" indicate that the solution is completely incorrect and "2" indicates that it is absolutely accurate. "1" indicates that the answer is only partially accurate (for example, the main categories are correct but the sub-categories are incorrect) and makes sense to human judges.

A. Algorithm and Flow-Chart

ALGORITHM

1) Installation of necessary libraries
2) Training the VQA Model using the dataset using CNN and RNN
3) Step-wise implementation of python code
4) Input Image(Extracts features from image using VGG16)

5) Input Question
6) Extract Answer from Question
7) Output in Text

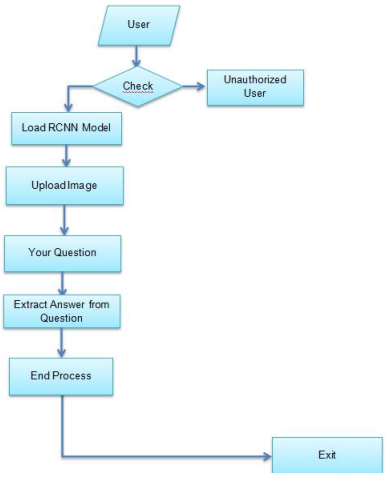


Fig. 2. Flow Chart

B. Design

The architecture of our mQA model, The model has four components:

- A Long Short-Term Memory (LSTM) for extracting semantic representation of a question
- A deep Convolutional Neural Network (CNN) for extracting the image representation
- An LSTM to extract representation of the current word in the answer and its linguistic context, and
- A fusing component that incorporates the information from the first three parts together and generates the next word in the answer. These four components can be jointly trained together

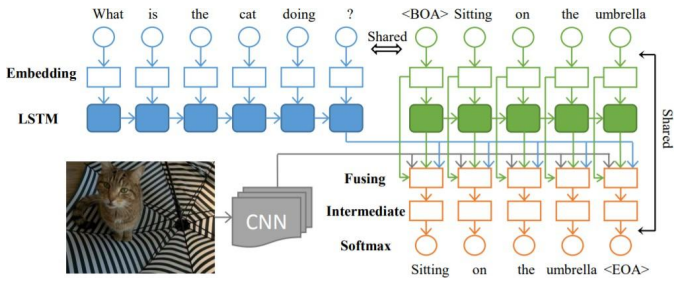


Fig. 3. Model Overview

V. Implementation

LSTM and CNN are used to implement the model. They combine the query and the response before feeding it to the LSTM. Unlike them, we employ two independent LSTMs for questions and answers, taking into account the varied features of questions and answers while allowing the word-embeddings to be shared. Adopt the suggested dataset, which is significantly smaller than our FM-IQA dataset. It also summarises the response in a single word.

1) The first component of the model extracts the semantic meaning of the query. It has a 512-dimensional word embedding layer and a 400-memory-cell LSTM layer. The word embedding layer's job is to map the word's one-hot vector into a dense semantic space. The LSTM layer receives this dense word representation.

2) A deep Convolutional Neural Network (CNN) creates the picture representation in the second component. The GoogleNet is used in this work. Other CNN models, like as AlexNet and VggNet, can be utilised as components in our model as well. The final SoftMax layer of the deep CNN is removed, and the remaining top layer is connected to our model.

3) A word embedding layer and an LSTM are also included in the third component. The structure is the same as the first component. The word embeddings and the activation of the memory cells for the words in the response will be input into the fusing component to produce the following words in the answer.

4) The fourth component, finally, combines the data from the preceding three levels. For the t th word in the response, the activation of the fusing layer f(t) may be computed as follows:
f(t) = g(VrQ rQ + VI I + VrA rA(t) + Vww(t));
where "+" denotes element-wise addition, rQ stands for the activation of the LSTM(Q) memory cells of the last word in the question, I denotes the image representation, rA(t) and w(t) denotes the activation of the LSTM(A) memory cells and the word embedding of the t th word in the answer respectively. VrQ , VI , VrA , and Vw are the weight matrices that need to be learned. g(.) is an element-wise non-linear function.

VI. Experimental Result

To gain further insights into these results, we computed accuracies by question type in Table 3. Interestingly, for question types that require more reasoning, such as "Is the" or "How many", the scene-level image features do not provide any additional information. However, for questions that can be answered using scene-level information, such as "What sport," we do see an improvement. Similarly, for questions whose answer may be contained in a generic caption we see improvement, such as "What animal". For all question types, the results are worse than human accuracies
We also analyzed the accuracies of our best model (deeper

LSTM Q + norm I) on a subset of questions with certain specific (ground truth) answers.

Question Type	Open-Ended K = 1000 Q	Q + I	Q + C	Human Q	Human Q + I	Human Age To Be Able To Answer	Commonsense To Be Able To Answer (%)
what is (13.84)	23.57	34.28	43.88	16.86	73.68	09.07	27.52
what color (08.98)	33.37	43.53	48.61	28.71	86.06	06.60	13.22
what kind (02.49)	27.78	42.72	43.88	19.10	70.11	10.55	40.34
what are (02.32)	25.47	39.10	47.27	17.72	69.49	09.03	28.72
what type (01.78)	27.68	42.62	44.32	19.53	70.65	11.04	38.92
is the (10.16)	70.76	69.87	70.50	65.24	95.67	08.51	30.30
is this (08.26)	70.34	70.79	71.54	63.35	95.43	10.13	45.32
how many (10.28)	43.78	40.33	47.52	30.45	86.32	07.67	15.93
are (07.57)	73.96	73.58	72.43	67.10	95.24	08.65	30.63
does (02.75)	76.81	75.81	75.88	69.96	95.70	09.29	38.97
where (02.90)	16.21	23.49	29.47	11.09	43.56	09.54	36.51
is there (03.60)	86.50	86.37	85.88	72.48	96.43	08.25	19.88
why (01.20)	16.24	13.94	14.54	11.80	21.50	11.18	73.56
which (01.21)	29.50	34.83	40.84	25.64	67.44	09.27	30.00
do (01.15)	77.73	79.31	74.63	71.33	95.44	09.23	37.68
what does (01.12)	19.58	20.00	23.19	11.12	75.88	10.02	33.27
what time (00.67)	8.35	14.00	18.28	07.64	58.98	09.81	31.83
who (00.77)	19.75	20.43	27.28	14.69	56.93	09.49	43.82
what sport (00.81)	37.96	81.12	93.87	17.86	95.59	08.07	31.87
what animal (00.53)	23.12	59.70	71.02	17.67	92.51	06.75	18.04
what brand (00.36)	40.13	36.84	32.19	25.34	80.95	12.50	41.33

Fig. 4. Open-ended test-dev results for different question types on real images (Q+C is reported on val). Machine performance is reported using the bag-of-words representation for questions. Questions types are determined by the one or two words that start the question. The percentage of questions for each type is shown in parentheses. Last and second last columns respectively show the average human age and average degree of commonsense required to answer the questions (as reported by AMT workers), respectively

After tuning hyper-parameters to optimize the algorithms, Stochastic Gradient Descent was found to be the best suited algorithm, taking both performance and time complexity into account. Following performance metrics were achieve:

- Accuracy: 92.81
- Precision: 96.97
- Recall: 91.94
- F1-Score: 94.39

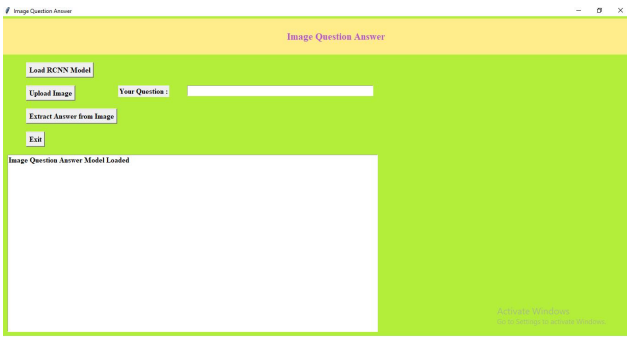
A. ScreenShots

The dataset you provided us can only answer limited questions like 'does sphere and cube colour match?' or 'is sphere present in image?' It will respond with [9] 'YES' or 'NO,' or the number of objects in the image,' but our project can describe anything in the image by asking a user question, and we're using RCNN to identify objects in images and LSTM to build vocabulary sentences in meaningful form to answer this.

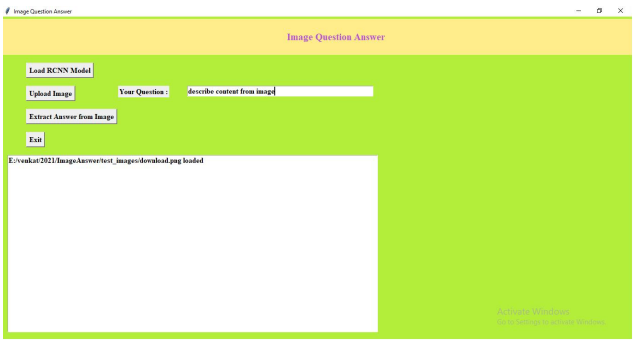
To run project double click on 'run.bat' file to get below screen

In above screen click on 'Load RCNN Model' button to load CNN model and to get below screen

In above screen in text area we can see model loaded and now click on 'Upload Image' button to upload



In above screen selecting and uploading 'download.png' file and then click on 'Open' button to get below screen



In above screen image loaded and I entered question as 'describe content from image' and now click on 'Extract Answer from Image' button to get below output

In above screen in yellow colour text we got answer about the image and now try another image

VII. Conclusion and Future work

It uses the query text to filter images based on categories and keywords, making image matching easier. The second QA step, information retrieval, looks for suitable responses in an internal library of resolved photo-based inquiries. The third, human-computation QA layer enlists the help of community specialists to address the most challenging instances. RCNN is used to recognise objects in pictures, while LSTM is used to construct coherent vocabulary phrases.

Future enhancements are being planned to further examine and improve the process so that the proper answer may be extracted without mistake. In order to extract more explicitly linked information, additional work includes creating knowledge-base queries that reflect the content of the query and the image. The Knowledge Base may be upgraded as well. [18]Open-IE, for example, gives more basic common-sense information such as 'cats eat fish.' Answering high-level questions will be easier with this information.

References

[1] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in Proc. Int. Conf. Learn. Representations, 2015.

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

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

[4] X. Chen and C. Lawrence Zitnick, "Mind's Eye: A Recurrent Visual Representation for Image Caption Generation," in Proc. IEEE Conf. Comp. Vis. Patt. Recogn., 2015.

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

[6] A. Karpathy, A. Joulin, and F. F. Li, "Deep fragment embeddings for bidirectional image sentence mapping," in Proc. Advances in Neural Inf. Process. Syst., 2014.

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

[8] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville, "Describing videos by exploiting temporal structure," in Proc. IEEE Int. Conf. Comp. Vis., 2015.

[9] J. Devlin, H. Cheng, H. Fang, S. Gupta, L. Deng, X. He, G. Zweig, and M. Mitchell, "Language models for image captioning: The quirks and what works," arXiv preprint arXiv:1505.01809, 2015.

[10] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in Proc. IEEE Conf. Comp. Vis. Patt. Recogn., 2014.

[11] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. "Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs". ICLR, 2015.

[12] M. Grubinger, P. Clough, H. Muller, and T. Deselaers. ''The iapr tc-12 benchmark: A new evaluation resource for visual information Systems". In International Workshop OntoImage, pages 13–23, 2006.

[13] A. Lavie and A. Agarwal. Meteor: ''An automatic metric for mt evaluation with high levels of correlation with human judgements". In Workshop on Statistical Machine Translation, pages 228–231. Association for Computational Linguistics, 2007. BibTeXb14 J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, and A. Yuille "Deep captioning with multimodal recurrent neural networks (m-rnn)". In ICLR, 2015.

[14] J. Zhu, J. Mao, and A. L. Yuille. ''Learning from weakly supervised data by the expectation loss svm (e-svm) algorithm". In NIPS, pages 1125–1133, 2014.

[15] A. Karpathy, A. Joulin, and F. F. Li, "Deep fragment embeddings for bidirectional image sentence mapping," in Proc. Advances in Neural Inf. Process. Syst., 2014.

[16] H. Agrawal, C. S. Mathialagan, Y. Goyal, N. Chavali, P. Banik, A. Mohapatra, A. Osman, and D. Batra. Cloudcv: Large-scale distributed computer vision as a cloud service. In Mobile Cloud Visual Media Computing, pages 265–290. Springer International Publishing, 2015

[17] S. Antol, C. L. Zitnick, and D. Parikh. Zero-Shot Learning via Visual Abstraction. In ECCV, 2014

[18] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. H. Jr., and T. M. Mitchell. Toward an Architecture for Never-Ending Language Learning. In AAAI, 2010.