A Mini-Project Report on

Image-based Question Answering

carried out as a part of course Information Retrieval (IT362)

Submitted by

Vinay Mundada (14IT124)

Chetan Jaydeep (14IT235)

Vinay Bhat (14IT248)

VI Sem B.Tech (IT)

in partial fulfillment for the award of the degree

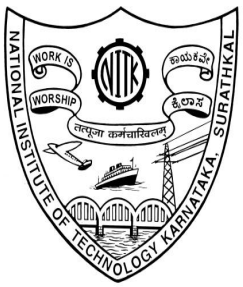
of

Bachelor of Technology

In

Information Technology

At



Department of Information Technology

National Institute of Technology Karnataka, Surathkal

April 2017

CERTIFICATE

This is to certify that the project entitled Image-based Question Answering is a bonafide work carried out as a part of the course **Information Retrieval (IT362)**, under my guidance by

1. Vinay Mundada (14IT124)
2. Chetan Jaydeep (14IT235)
3. Vinay Bhat (14IT248)

students of VI Sem B.Tech (IT) at the Department of Information Technology, National Institute of Technology Karnataka, Surathkal, during the academic year Jan- May 2017, in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Information Technology, at NITK Surathkal.

Place: NITK Surathkal

Signature of the Instructor

Date:

DECLARATION

We hereby declare that the project entitled "Image-based Question Answering" submitted as part of the partial course requirements for the course Information Retrieval (IT362) for the award of the degree of Bachelor of Technology in Information Technology at NITK Surathkal during the Jan-May 2017 semester has been carried out by us. We declare that the project has not formed the basis for the award of any degree, associateship, fellowship or any other similar titles elsewhere.

Further, we declare that we will not share, re-submit or publish the code, idea, framework and/or any publication that may arise out of this work for academic or profit purposes without obtaining the prior written consent of the Course Instructor.

Name and signature of the Students:

1. Vinay Mundada (14IT124)
2. Chetan Jaydeep (14IT235)
3. Vinay Bhat (14IT248)

Place: NITK Surathkal

Date:

Abstract

An image is a vital source of information. Interacting with an image in natural language has always been the dream task of artificial intelligence. This project addresses the problem of image-based question answering (also referred to as Image-QA). We make use of neural networks end-to-end, without intermediate stages such as object detection and image segmentation and predict best possible answer to simple questions.

Contents

List of Figures

1 Introduction

The combination of image understanding and natural language interaction is one of the grand dreams of artificial intelligence. Here, we aim at learning the image and text through a question-answering. Image question answering problem can be solved by learning a convolutional neural network (CNN) with a dynamic parameter layer whose weights are dynamically determined based on questions. Researchers studying image caption generation [1], recently have developed powerful methods of simultaneously learning from image and text inputs. The images along with the text together form higher level representations which can be coupled with models such as convolutional neural networks (CNNs). Such models can be then trained on object recognition and word embeddings trained on large text corpus. An extra layer of interaction between human and computers is involved in Image Question Answering. Here the model needs to focus on the details of the image instead of describing it randomly. Image labeling and object detection are the other areas of computer vision which also come under this problem. We hereby present our contribution or solution to the problem of generic end-to-end question-answering (QA) model using semantic embeddings to connect a convolutional neural network (CNN) and a LSTM recurrent neural net (RNN). Also importantly, we assume that the answers to be only single-word, which makes this a classification problem. It makes the evaluation of the model reliable and more robust.

2 Literature Survey

2.1 *Background*

Malinowski and Fritz [2] have released a data set containing images and the corresponding question-answer (QA) pairs, referred to as the]*"DAtaset for QUestion Answering on Real-world images (DAQUAR)"*. The data set consists of images with labellings such as Human segmentation, image depth values, and object labeling. There are only 3 types of questions in this dataset, which are - object type, object color, and number of objects. Most of the questions are easy but some questions are very hard to answer even for a human being. The work presented by Malinowski and Fritz combines semantic parsing and image segmentation. This is the first notable approach at image QA, although it has some limitations. Firstly, the type of questions very specific. Also, their model computes all possible spatial relations during the training time of the images. It becomes an expensive operation for a larger data set, even though the model limits this to the nearest neighbors of the test images. Very recently, a number of accomplishments have been made on both creating datasets and proposing new models for image QA. Both Antol et al.[4] and Gao et al.[5] used MS-COCO [3] images and created an open domain dataset with human generated questions and answers. In Anto et al.s work, cartoon pictures were also included by the authors besides real images. Logical reasoning is required for some questions in order to answer correctly. Malinowski et al. [2] and Gao et al.[5] use recurrent networks to encode the sentence and output the answer. Whereas Malinowski et al. use a single network to handle both encoding and decoding, Gao et al. used two networks, a separate encoder and decoder.

2.2 *Outcome of Literature Survey*

Based on the literature survey done, we discovered that we need to develop a model by applying various forms of neural networks such as a CNN (to train and extract features from images) and a RNN (to feed the image with the question and to achieve the best possible answer to that question. Also, we observed that, a more reliable and robust model can be generated if the type of questions to be queried on the image is limited. As a result, we make use of the dataset [3], containing only three types of questions: object type, object colour and number of objects.

2.3 *Problem Statement*

This project aims to address the problem of Image based Question-Answering using Neural networks and visual semantic embeddings without the intermediate stages of object recognition and image segmentation.

2.4 *Objectives*

1. A model that is able to answer simple questions based on images.
2. A trained VGG-19 model needs to be used to get image vector needed for the model.
3. Using image as the first word by reducing dimensions of VGG-19 vector to word2vec dimensions by using a linear transformation.
4. Models vocabulary of answers is limited to answer set it will be trained on. Hence, complexity of questions needs to be handled.

3 Methodology

3.1 *System Architecture*

The system consists of a model which is developed by applying various forms of neural networks and visual semantic embeddings. The input to this model is a 4096 dimensional image vector obtained from the *19-layer Oxford VGG Conv Net*[6] trained on ImageNet 2014 Challenge[7]. This image-vector along with word embeddings of the question, are then fed to a Long Short Term Memory Model (LSTM)[8] Recurrent Neural Network. Figure 3.1 shows the system architecture.

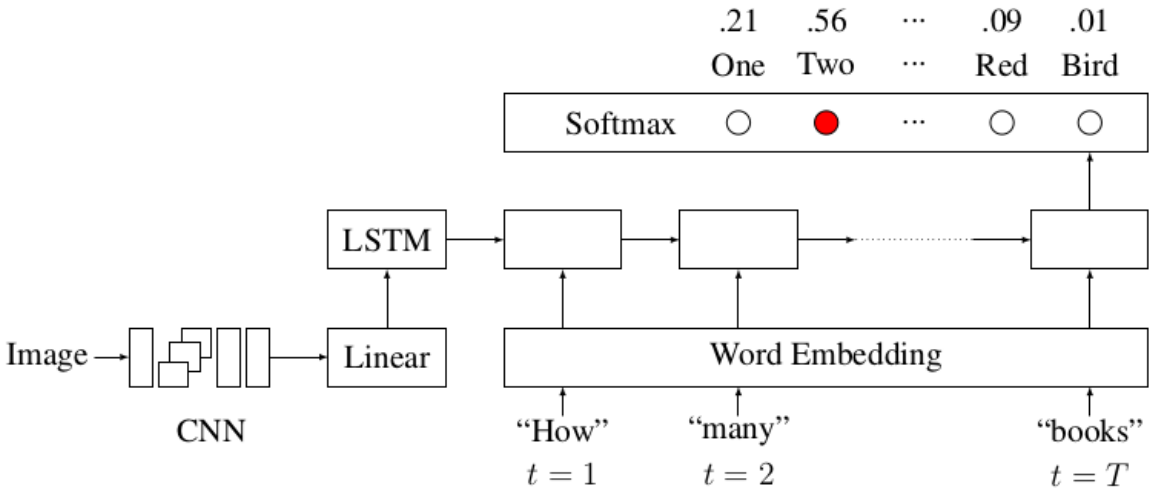


Figure 3.1: System Architecture

3.2 *Detailed Design Methodologies*

1. We use the VGG-19 ConvNets final hidden layer to get a 4096-dimensional feature vector which is going to be passed on to the LSTM.

2. The CNN part of our model is kept frozen during training, it is used during inference.

3. These high dimensional feature vectors act as visual embeddings.

4. We apply a linear transformation to the feature vector to convert it to our word vector dimensions, which is 300. The transformed feature vector is used as the first word, and passed to the Long Short Term Memory network.

5. The embedding layer converts the one-hot word representations to word vectors and word vectors are then passed one by one to the LSTM neural network.

6. The final output of the LSTM network, the output of the final timestep, is passed to a softmax layer to generate answers.

7. The softmax layer classifies the input over the vocabulary of the answer set to generate the answer word.

As mentioned earlier, a neural network approach is taken as opposed to object detection and image segmentation. Hence, the key neural networks used in methodology are explained below.

Convolutional Neural Networks

Convolutional Neural Networks as shown in Fig 3.2 are models used in Machine Learning for image classification, image captioning and other vision tasks. They are a type of feed forward artificial neural networks, where the connections between the various layers are inspired by the visual cortex. It uses the convolution operation, which along with using hierarchical deep layers, allows convolutional neural networks to extract high-dimensional features from images. These features can then be used for a variety of tasks including classification, captioning etc. In this case, the features are used for question answering.

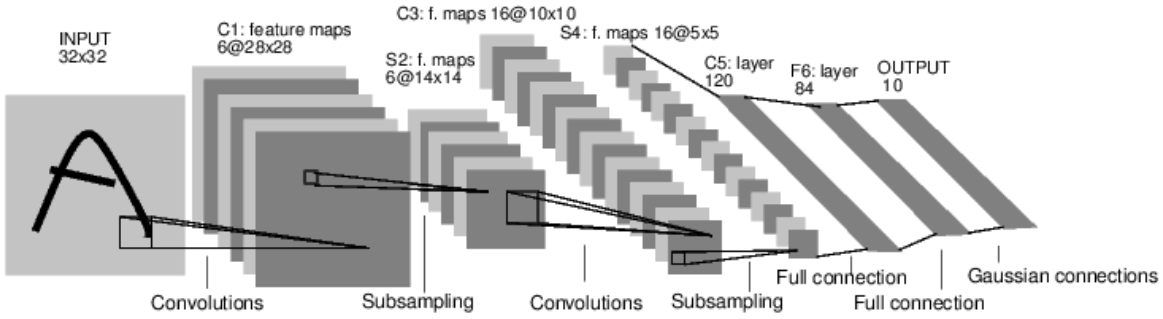


Figure 3.2: Convolutional Neural Network

VGG-19 ConvNet

University of Oxfords VGG-19 Convolutional Neural Network (refer Fig. 3.3) trained on ImageNet is used for extracting high level features from images. Their use of deep layers allows abstract features to be captured as vectors. We use the output of last hidden layer of 4096 neurons, to get a feature vector of 4096 dimensions.

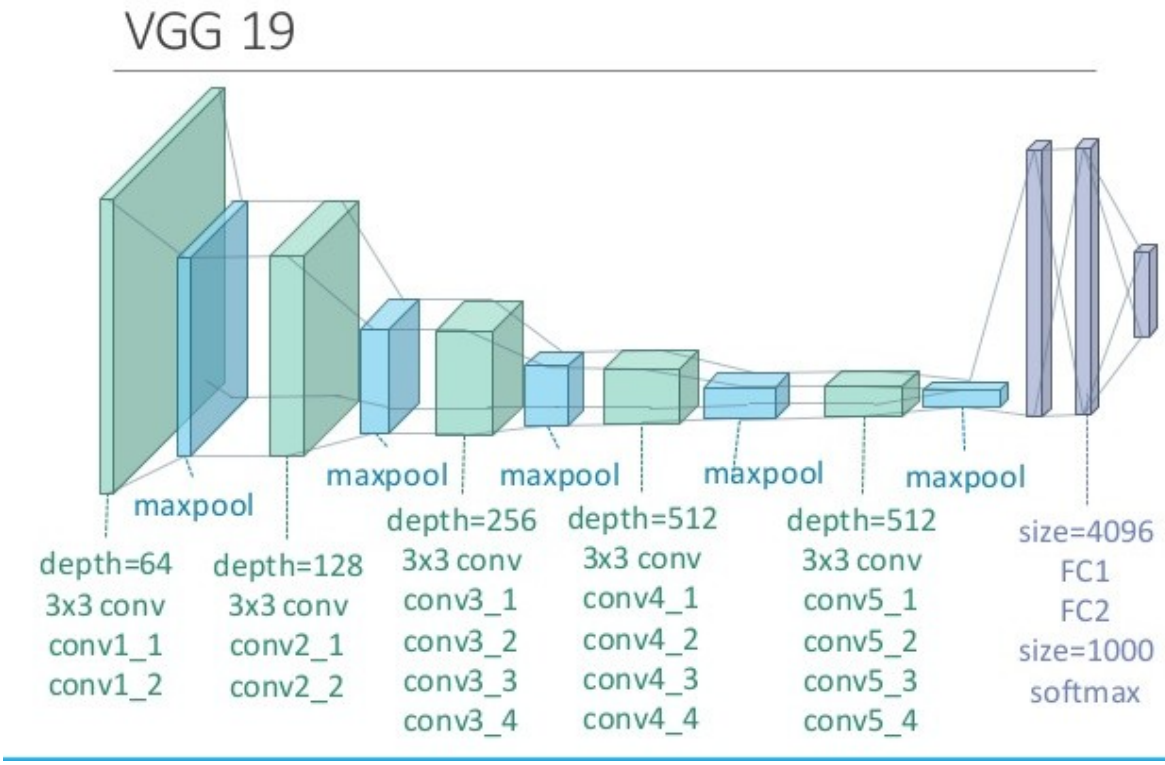


Figure 3.3: VGG-19 ConvNet

Recurrent Neural Networks

Recurrent Neural Networks as shown in Fig 3.4 are models used in Machine Learning for Natural Language Processing, language modelling and other language tasks. They are also a type of feedforward artificial neural networks. However, a core difference in RNNs is the ability of feedback, ie, RNNs can remember their previous outputs and use it along with the current input. RNNs operate over sequences of data, hence they have the notion of timesteps, where the previous timestep and the current input are fed to it. Recurrent Neural Networks are hence very well suited to language tasks as they are uniquely capable of modeling language.

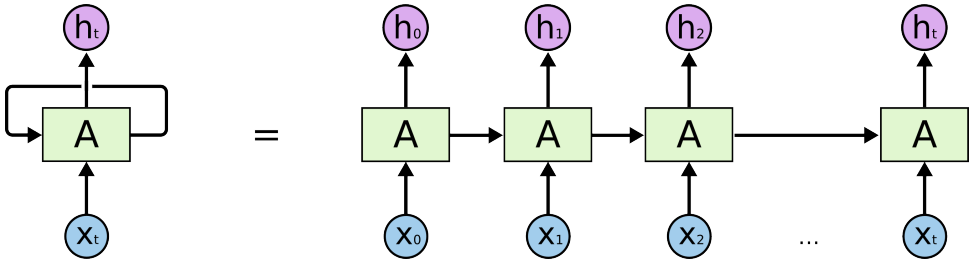


Figure 3.4: Recurrent Neural Network

Long Short Term Memory

Long Short Term Memory networks, or LSTMs are a type of RNNs which are capable of learning long-term dependencies between previous inputs. RNNs fail to propagate information over long sequences, because of exponential gradient increase or decrease. LSTMs combat this by using a gated structure containing an input gate, output gate, and a forget gate which allows the LSTM to selectively retain some information or forget it. Fig 3.5 shows a standard LSTM model.

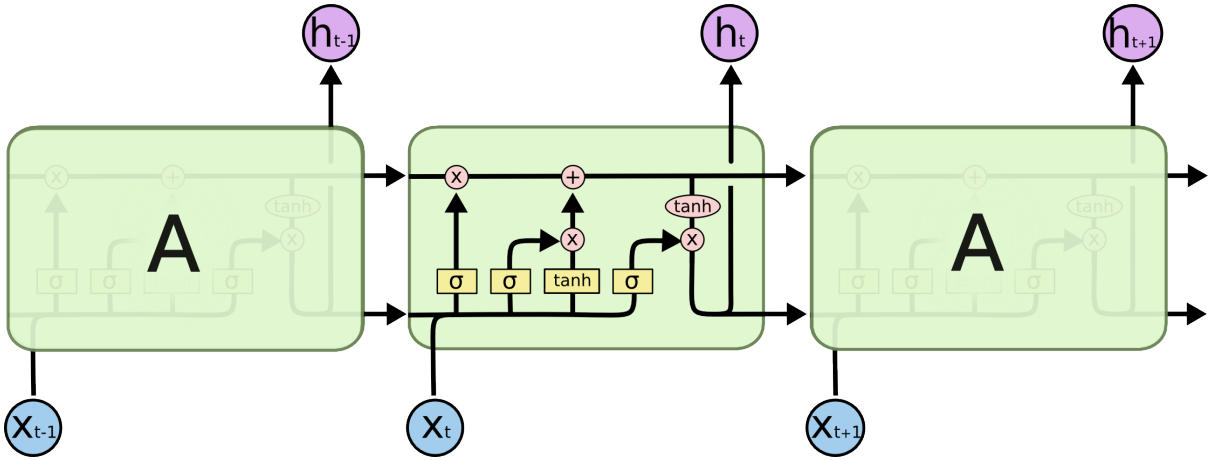


Figure 3.5: A standard LSTM model

Word Embeddings

LSTMs are great at modeling sequences, however they are even better if the individual inputs are dense vectors. The words in a sentence are usually modeled as one-hot vectors, where each word is simply a row of an identity matrix the size of the vocabulary. However, these vectors are sparse. Word Embeddings are the solution to this problem, they are dense vector representations of words, capable of storing relational information. We use CBOW word embeddings – Continuous Bag of Words. The vectors by generated by having a sliding window of words around a current word, taking the context words and using them to predict the current word. Hence, word vectors groups similar words – words that appear in similar contexts – onto a vector space. Words that are far apart are different.

4 Implementation

4.1 *Work Done*

Data Preprocessing:

1. Extracted the **Hidden Oxford MSCOCO dataset** file into a sparse matrix by the **h5py library**, deserializing the binary and loading it into a Python object including the indices, data, data pointers etc.

2. Converted the sparse matrix into a dense, normal matrix by using the **scipy library**. Scipy library provides the **sparse.csr_matrix API** which allows direct conversion of Python objects into sparse matrices and dense matrices.

3. Prepared the dataset of images, questions and answers of the training set by dereferencing the IDs from the given data, and by cross-matching it with the raw data. This allows us to have any train-test split and makes things convenient.

Model generation:

1. Implemented the **VGG-19 Convolutional Neural Network** for extracting features from the images. We used pre-trained weights for initializing the neural network and lightly trained it on new inputs. VGG-19 allows us to extract high-level features from the image.

2. Implemented the **Visual-LSTM-model**, involving multi-modal inputs and multi-modal outputs. The Visual-LSTM model involves two branches of input:

   (a) Taking the input image feature vector as one branch of input, then connecting that to a Dense fully-connected layer of neurons which convert the **4096-dimensional feature vector** to **300-dimensional** word vector by a linear transformation which is learnt during training.

   (b) Taking the question as one branch of input, first they are **tokenized** into individual words, these are then created into one-hot vectors. One-hot vectors are rows of an identity matrix of the size of the vocabulary, and each word is the row of its index. These one-hot vectors are then fed to the Embedding layer. The embedding layer converts the vectors into 300-dimensional word

vectors, which are capable of storing relational information and other high dimensional data.

(c) We combine the two branches by concatenating the two resultant Tensors on their first dimension axis so as to combine them in a way which treats the transformed image vector as the first word, and then followed by all the other word vectors.

(d) This final Tensor is then passed onto the **Long Short Term Memory** neural network cell which takes the input vectors one by one and produces an output for each time step.

(e) We take the LSTM output of the final timestep and feed it onto a **softmax classification layer** which produces a probability distribution over the vocabulary of words as an answer distribution.

(f) We take the word with the highest probability as the answer to the question.

3. The neural network implementation is done using the **Keras deep learning framework** in Python.

4. Implemented an interface, a GUI for user convenience and for interacting with the system. The interface allows for question answering on any user submitted image or on any image in the training set as well.

4.2 *Results and Analysis*

The following results and analysis have been performed on an established experimental setup.

Experimental Setup:

1. The experiments are carried out in a **Python's virtual environment** where all the required dependencies are installed seperately.

2. The model is built using:

   (a) Keras 1.2.2 Framework

   (b) TensorFlow 1.0.0 library

3. Linking the image QA model written in Pyton to the web interface is done using the **Flask server** which runs on "http://localhost:5000".

Analysis:

The LSTM model is trained using a **softmax loss function** (Cross Entropy Loss). In information theory, the cross entropy between two probability distributions p and q over the same underlying set of events measures the average number of bits needed to identify an event drawn from the set, if a coding scheme is used that is optimized for an "unnatural" probability distribution q, rather than the "true" distribution p.
The cross entropy for the distributions p and q over a given set is defined as follows:

$$H(p, q) = \mathrm{E}_p[-\log q] = H(p) + D_{\mathrm{KL}}(p \| q),$$

Figure 4.1: Cross Entropy for continuous Distribution

where H(p) is entropy of p, D(p or q) is the Kullback-Leibler divergence of q from p (also known as the relative entropy of p with respect to q)
For discrete p and q, we have-

$$H(p, q) = -\sum_{x} p(x)\ \log q(x).$$

Figure 4.2: Cross Entropy for discrete values

Since we use a limited set of words as vocabulary (431) for the one word answers, **accuracy** is the effective metric to evaluate the model. A plot of accuracy against the number of epochs is shown in Fig 4.3

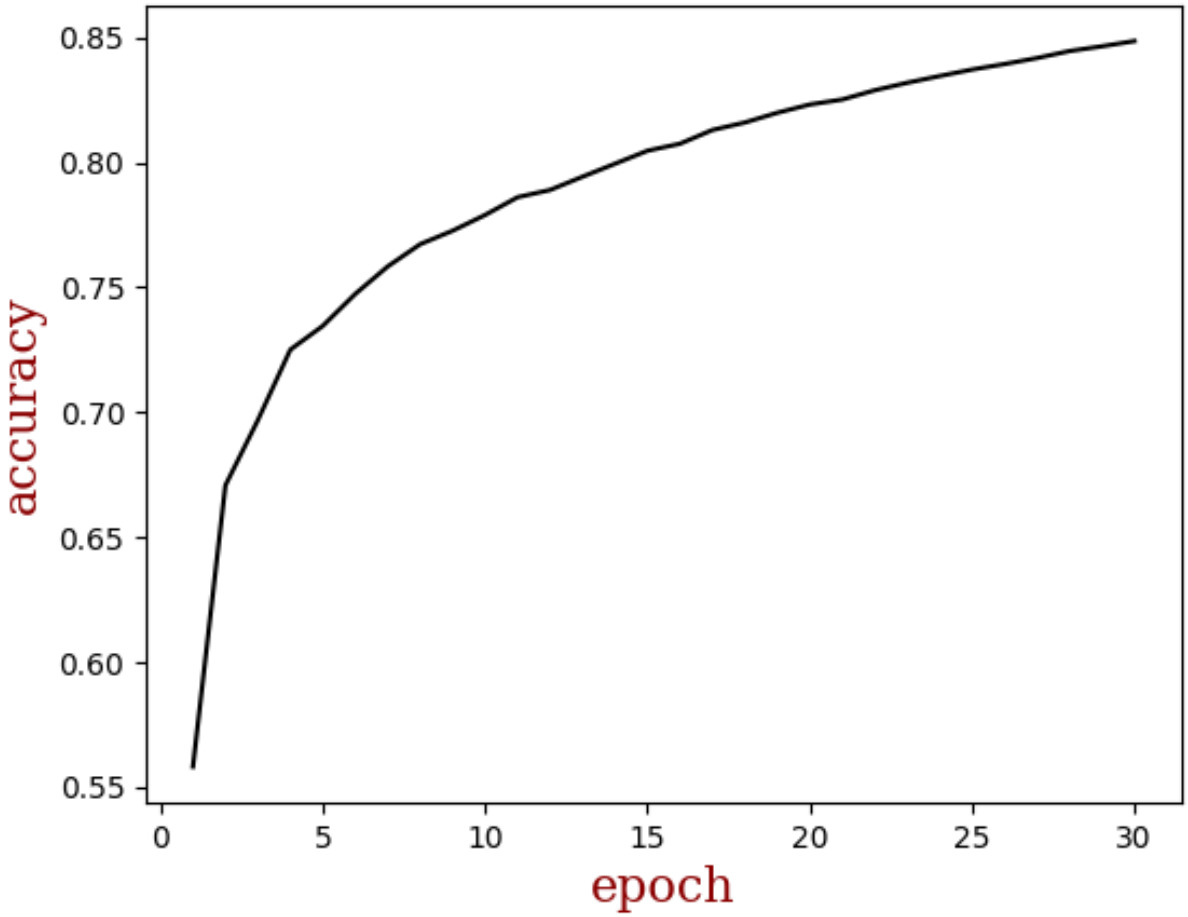


Figure 4.3: Accuracy of the model

The training loss calculated by the cross entropy loss decreases with the number of epochs. Fig 4.4 shows a plot of the training loss against the number of epochs.

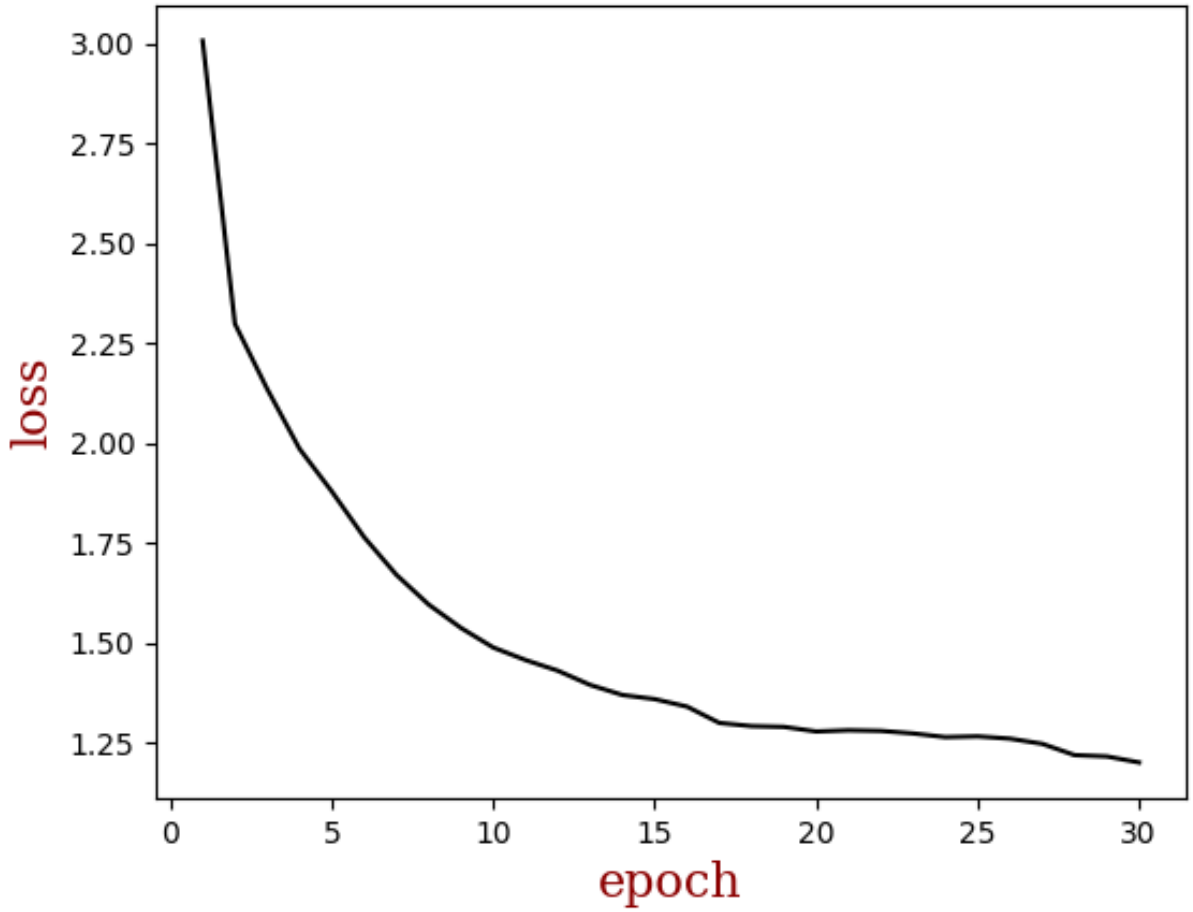


Figure 4.4: Training Loss of the model

4.3 *Innovative Work*

The additional component that has been implemented in this project apart from the base paper is as follows:

1. **Bi-directional LSTM model**

   (a) The image feature vector is treated as the first word of the LSTM and then subsequently all word vectors are passed. In unidirectional LSTMs, by definition the future input information cannot be accessed from the current state. Hence, they are slightly biased towards one direction.

   (b) Hence, we also experimented with Bidirectional LSTMs. Bidirectional LSTMs, also called BRNNs (Bidirectional Recurrent Neural Networks) are used to increase the amount of input information available to the network. Moreover,

their future input information is reachable from the current state.

(c) Fig 4.3 shows the basic idea of Bidirectional Recurrent Neural Networks, which is to connect two hidden layers of opposite directions to the same output.

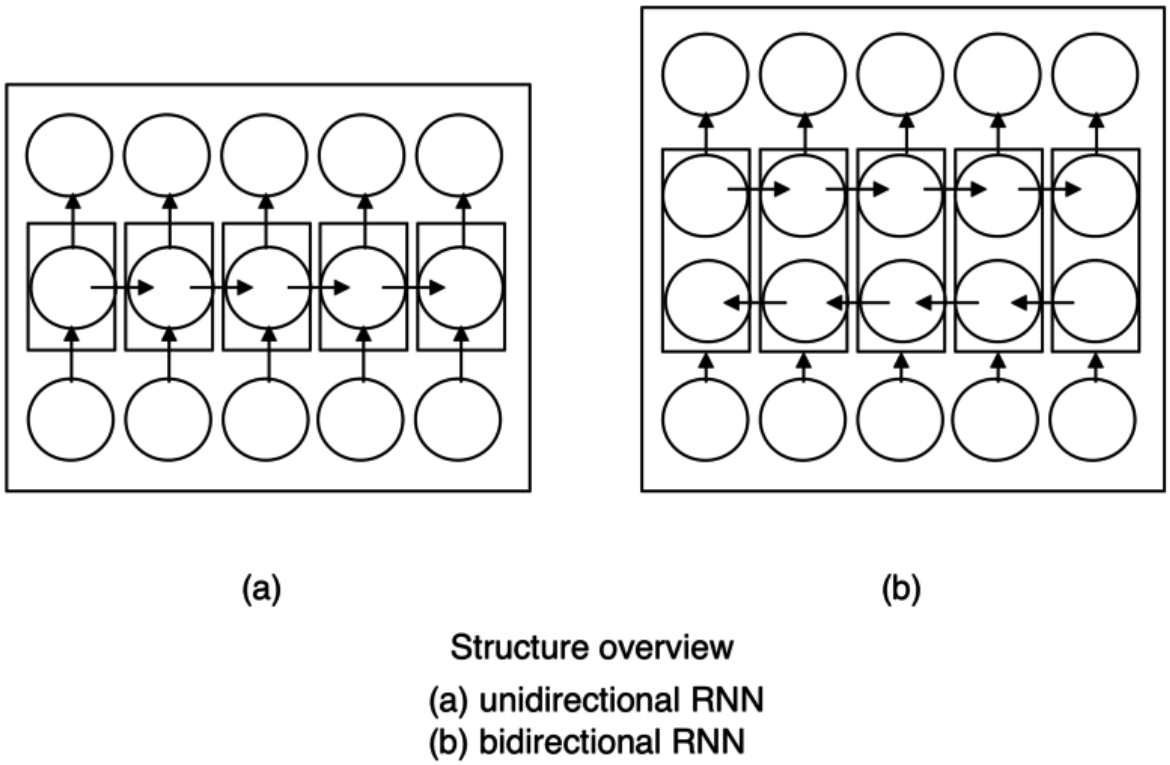


Figure 4.5: Using a Bidirectional LSTM

2. **Dropout**

(a) Dropout as shown in Fig 4.4 is a regularization technique for reducing overfitting in neural networks by preventing complex co-adaptations on training data.

(b) It is a very efficient way of performing model averaging with neural networks.

(c) The term Dropout refers to dropping out units (both hidden and visible) in a neural network.

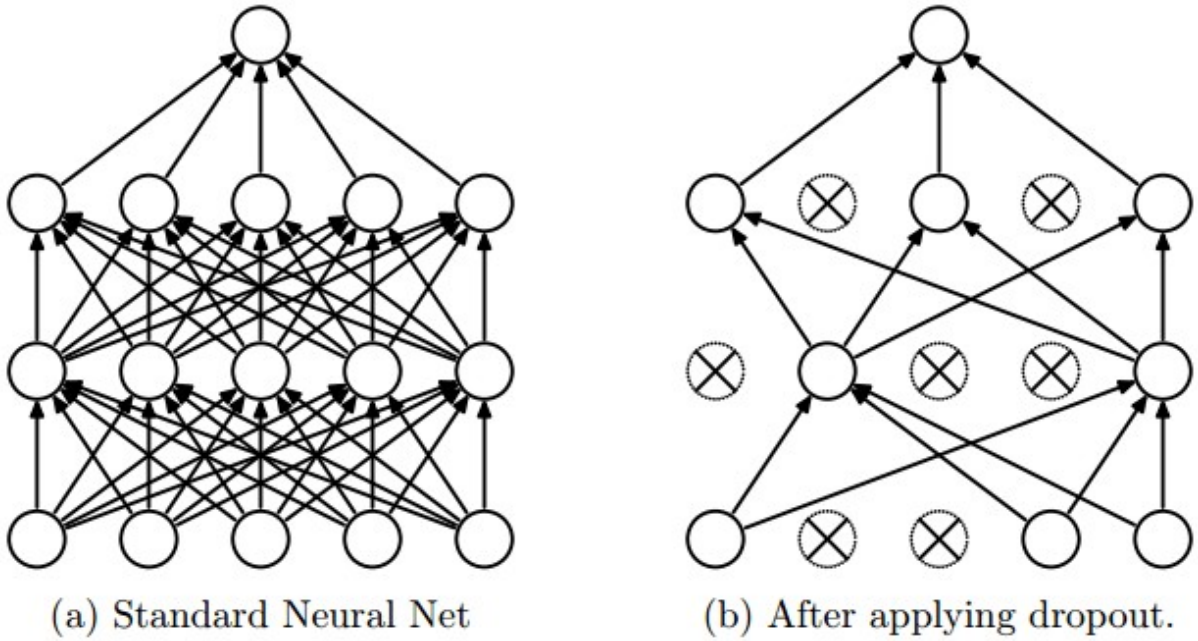


Figure 4.6: Using Dropout

4.4 *Individual Contributions of Team Members*

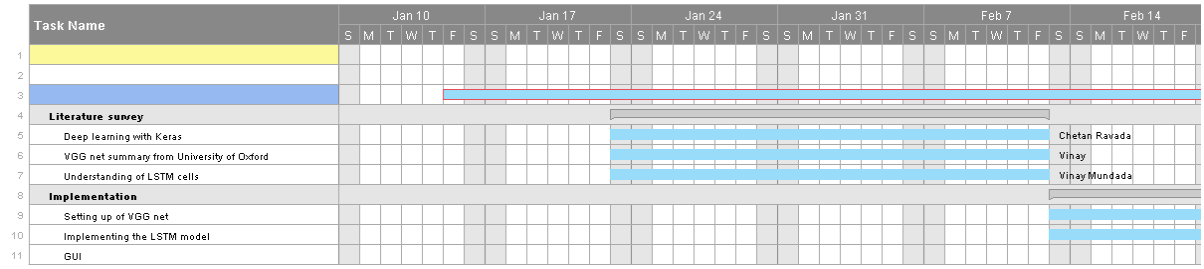


Figure 4.7: Individual Contributions-Phase 1

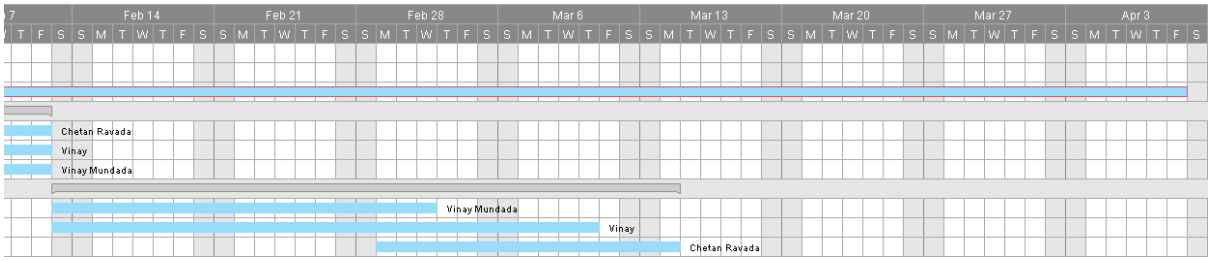


Figure 4.8: Individual Contributions-Phase 2

5 Conclusion & Future Work

We have implemented a model for end-to-end generation of answers to questions queried to a given image. Using the datasets as mentioned before, we have obtained promising results. The answers which the model is generating are quite satisfactory and being generated at very fast speeds, which signifies that the model being develped is efficient. Considering the image itself as the first word of the question helps reducing the number of computations and hence the computation time. The model can also be run on a completely new image (which is not present in the test dataset of MS-COCO).
As a part of the future work, we have decided to add more specifications to the model by:

1. Including more types of questions to be queried on the images, and not only three.

2. Generating answers to the questions with more than one word.

Implementations and rigorous testing of the above ideas have also been left as a part of future work.

References

[1] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, Show and tell: A neural image caption generator, in CVPR, 2015.

[2] M. Malinowski and M. Fritz, Towards a visual Turing challenge, in NIPS Workshop on Learning Se- mantics, 2014.

[3] MS-COCO Dataset ‘‘`http://mscoco.org/dataset/#download`’’

[4] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, VQA: Visual Question Answering, CoRR, vol. abs/1505.00468, 2015.

[5] H. Gao, J. Mao, J. Zhou, Z. Huang, L. Wang, and W. Xu, Are you talking to a machine? dataset and methods for multilingual image question answering, CoRR, vol. abs/1505.05612, 2015.

[6] K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition, in ICLR, 2015.

[7] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and L. Fei-Fei, Imagenet large scale visual recognition challenge, IJCV, 2015.

[8] S. Hochreiter and J. Schmidhuber, Long short-term memory, Neural Computation, vol. 9, no. 8, pp. 17351780, 1997.