Shahjalal University of Science and Technology

Department of Computer Science and Engineering

CSE 452

Visual Question Answering from Abstract and Real Images with BERT Embeddings

Raihan Kabir Fahim

Reg. No.: 2015331033

4th year, 2nd Semester

Naznin Haque

Reg. No.: 2015331047

4th year, 2nd Semester

Department of Computer Science and Engineering

Supervisor

Marium E Jannat

Assistant Professor

Department of Computer Science and Engineering

9th February, 2020

Visual Question Answering from Abstract and Real Images with BERT Embeddings

A Thesis submitted to the Department of Computer Science and Engineering, Shahjalal University of Science and Technology, in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering.

By

Raihan Kabir Fahim
Reg. No.: 2015331033
4th year, 2nd Semester

Naznin Haque
Reg. No.: 2015331047
4th year, 2nd Semester

Department of Computer Science and Engineering

Supervisor

Marium E Jannat
Assistant Professor
Department of Computer Science and Engineering

9th February, 2020

Recommendation Letter from Thesis Supervisor

The thesis entitled *Visual Question Answering from Abstract and Real Images with BERT Embeddings*
submitted by the students

1. Raihan Kabir Fahim
2. Naznin Haque

is under my supervision. I, hereby, agree that the thesis can be submitted for examination.

Signature of the Supervisor:
Name of the Supervisor: Marium E Jannat
Date: 9^{th} February, 2020

Certificate of Acceptance of the Thesis

The thesis entitled *Visual Question Answering from Abstract and Real Images with BERT Embeddings*

submitted by the students

1. Raihan Kabir Fahim
2. Naznin Haque

on 9th February, 2020

is, hereby, accepted as the partial fulfillment of the requirements for the award of their Bachelor Degrees.

Head of the Dept.
Mohammad Abdullah Al Mumin
Professor
Department of Computer Science and Engineering

Chairman, Exam. Committee
Dr Mohammad Reza Selim
Professor
Department of Computer Science and Engineering

Supervisor
Marium E Jannat
Assistant Professor
Department of Computer Science and Engineering

Abstract

VQA(Visual question answering) is a recent topic in the field of Deep Learning which many researchers has taken a lot of interest in. It is a multimodal problem which falls in the domain of both Computer Vision and Natural Language Processing. Visual question answering is the problem of giving a natural language answer to any natural language questions about any image. VQA is related to the vision turing task and solving it would go a long way in achieving human-like AI. In this article, we present a baseline vqa model which closely follows the architecture of the model from [1] but differs in the question representation part. Our work involves modifying the question feature extraction where we used the recently published pre-trained BERT (Bidirectional Encoder Representations from Transformers) model by Google [2] to extract sentence embeddings from questions. We present results of our model on VQA-abstract (61.3% accuracy) and VQA-real (57.35%). We discuss our findings and also compare our model with other baselines and VQA models.

Keywords: VQA, computer vision, natural language processing, BERT, multimodal learning

Acknowledgements

We would like to thank the Department of Computer Science and Engineering, Shahjalal University of Science and Technology, Sylhet 3114, Bangladesh, for supporting this research. We are very thankful to our honorable supervisor Marium E Jannat and our teacher Md Saiful Islam for their worthy supports and directions. Without their supports and directions this work couldn’t be done.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The field of computer vision has seen significant advancements in recent times. Especially since the development of neural networks, many computer vision problems such as: captioning, object recognition, object detection, scene classification has seen tremendous progress. Moreover, the advent of social networks especially image sharing websites has lead to an abundance of image data. Many efficient methods have been developed to extract and collect this data which has made large image datasets possible. Crowd-sourcing has also enabled to utilise human workers to quickly perform captioning, annotation etc to produce large training datasets needed for machine-learning algorithms used today. Same things can be said for NLP (Natural Language Processing) which is also the other domain VQA falls under.

The problem of VQA is that, given any image and any natural language question about that image, an appropriate natural language answer has to be given.

Despite the simple description VQA encompasses a lot of sub-problems that have to solved such as:

- Object detection: What is behind the tree?
- Object recognition: Are there any dogs in the image?
- Counting: How many houses are there?
- Scene classification: Is it raining?
- Attribute classification: Is the person happy?

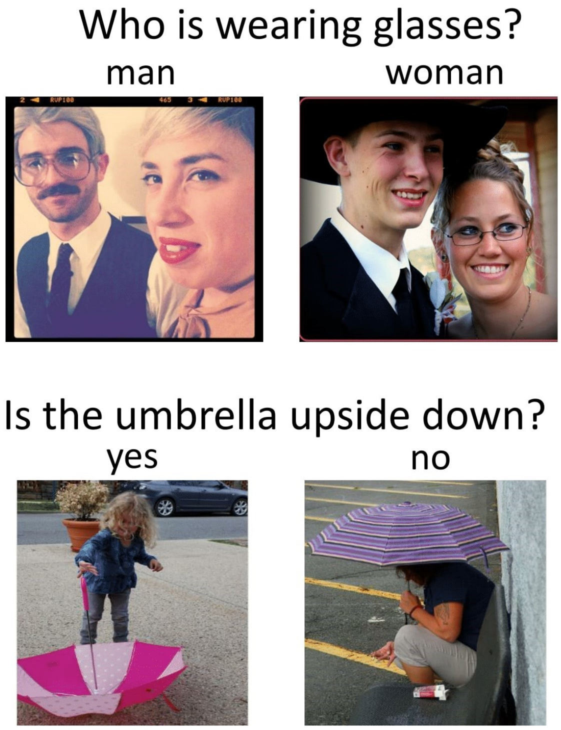


Figure 1.1: Example of VQA

Moreover, VQA not only poses vision related problems but NLP problems too. If the natural language question isn't understood properly any vision task won't produce the proper result. Answers have to be given in natural language too as to meet the expectation of the human user.

So what exactly is solving the VQA problem going to achieve? There are many applications VQA can be used for. The most obvious one is to help blind people understand their surroundings. Searching for images can be made easier as the user just has to ask a natural language question and the database is searched for images that can properly answer that question. But the ultimate goal of the VQA problem is to pass the vision Turing test which is an important goal of AI research. A significant AI must able to understand and utilise vision information properly and this understanding can be tested by VQA.

Chapter 2

Related Works

A lot of work has been done in VQA in a relatively short period of time since the first major paper [3] was published in 2015. Earlier models approached the problem in a relatively simple manner, first extracting image and question features and then combining them through some simple operation such as: concatenation or element-wise addition. Later models evolved to be more complex either employing more complex methods in feature extraction and preprocessing or using more complex operations in order to capture richer information from combining both modalities.

AYN[4], Foc-reg[5], CM[6] used grid-based CNN[7] to extract image features whereas BUTD[8], DFAF[9], FDA[10] used object-based Faster R-CNN[11]. In question feature extraction, earlier models used simple methods such as Bag-of-words(BOW) or one-hot vectors but most later models such as DCN[12], DRAU[13] switched to word2vec[14] or GloVe[15] which were used to extract word embeddings which were then used with LSTM[16] or GRU[17] to get the final question representation.

The image and questions features have to be combined to derive an answer. Most models at first used simple operations. DCN[12], MAN[18] used concatenation, SAN[19], CVA[20] used element-wise addition or BUTD[8], FDA[10] used element-wise multiplication. Later models either used attention where the question is used to produce an attention map over the image or multimodal fusion methods where complex matrix operations are used to combine the features.

Attention mechanism is the most widely used technique by VQA models to improve accuracy. Attention mechanism in VQA tries to mimic selective visual attention done by humans. When we are shown an image and asked a question about that image, we usually focus on the particular region in the image which helps to answer the question. VQA models try to replicate this. This improves accuracy because a lot of irrelevant noise is filtered out. Over the years many different attention mechanisms have been studied constantly leading to new insights in this area. Most models such as DCN[12], BUTD[8] usually use the question to produce an attention map over the image. This map assigns different weights to different regions/objects in the image according to their perceived relevance in answering the question. Many models use multiple attention layers to get multiple "glimpses" of the image to refine the attention process. SAN[19] was the first model to apply multi-step attention by stacking multiple layers. SMem[21] has two attention layers. The first layer uses each question word to guide over the image and in the second layer, the whole question is used. Some models like HieCoAtt[22], DAN[23], DRAU[13], HOA[24] also use attention on the question to focus on the keywords in the question that are more important for predicting the answer. These kinds of attention are called co-attention because either the question attends the image or the image attends the question. But some models use self-attention where the image/question attends itself. HOA[24], DFAF[9], MCAN[25] use self-attention on both image and question.

Another recent development in attention is dense co-attention. Dense co-attention means each word of the question attends to each region/object of the image and vice-versa. Before, the whole image attended to the whole question/individual words or the whole question attended to the whole image/individual region/objects. Dense co-attention seems to offer richer interaction between the two modalities, kind of similar to how fusion models try to simulate this by approximating the outer product. DCN[12] is an example of a model that uses dense co-attention

Another important aspect is channel attention. A recent model CVA [20] claims that attending over the channels of the output from the CNN(C from the output $W \times H \times C$) is important. Different channels of a CNN feature map is essentially activation response maps of the corresponding filter, and channel-wise attention can be viewed as the process of selecting semantic attributes according to the question. For example, when we want to predict cat, channel attention (e.g., in the conv5 3/conv5 4 feature map) will assign more weights on channel-wise feature maps generated by filters corresponding to semantics like furry texture, ear, and cat-like shapes. Essentially, each channel of a feature map in CNN is correlated to a convolutional filter which performs as a pattern detector. For instance, the lower-level filters detect visual clues such as edges and color, while the higher-level filters detect semantic patterns, such as attributes or object components.

Various fusion schemes are often used to combine image and question features in lieu of simpler operations. These fusion methods can be used with or without attention and aren't mutually exclusive. The outer product of two feature vectors gives a complete representation of all possible interactions between the elements of the vectors which offers enriched interaction between the two modalities. But computing this outer product is computationally infeasible. MCB[26] tries to approximate this outer product by computing the count-sketch of the outer product using efficient element-wise product in FFT space. MLB[27] argues MCB needs too many parameters and presents an alternative technique using matrix decomposition and the hadamard product. MFB[28] is a more advanced model where the authors show that MLB is a special case of MFB. Another model that uses matrix decomposition in it's fusion scheme is MUTAN[29] which uses tucker decomposition.

Most VQA models lack significant reasoning skills. They show good performance on general VQA datasets which lack questions that require complex reasoning ability to answer. They fail on datasets like CLEVR[30] designed specifically to test a model's high level reasoning skill. To correctly reason about complex questions, a model needs to be compositional. Here by composition we mean the ability to break the question down into individual reasoning steps which when done sequentially produces the correct answer.

NMN[31], D-NMN[32], N2NMN[33], IEP[34], DDRProg[35], TbD-Net[36], PTGRN[37], NS-VQA[38] treat each question as a dependency tree where each node represents a single reasoning step. When the tree is traversed and each node processed in a fixed way, we arrive at the answer. These models parse the question to produce such a dependency tree. They then assign to each node, a neural module from a fixed set of modules. The modules can be thought to perform a single specialized reasoning step such as a "find" module to locate an object/concept in the image. FiLM[39], CMM[40], MAC[41] process reasoning linearly rather than in tree-like fashion. Unlike the previously mentioned models they don't use multiple neural modules. They instead use a single reusable blackbox cell which is cascaded sequentially to form a long chain. Though multiple instances of the same cell is used in multiple layers, they only share their general structure. Through learning, different cells in different layers learn to specialize to perform different reasoning operations.

If we want a model to be able to really answer *any* question, it must have external knowledge gathering capability. Several VQA models do just that and are known as EKB models. These models transform the natural language question into a query that can be applied to a KB. The main limitation of this method is that there is only a few number of ways in which a KB can be queried, and this limits the types of questions that can be asked. Thus, although questions are asked in natural language, they must be reducible to one of the available query templates. Examples of such models include: AMA[42], Ahab[43], ITQ[44], KDMN[45], FVQA[46], STTF[47], OOTB[48].

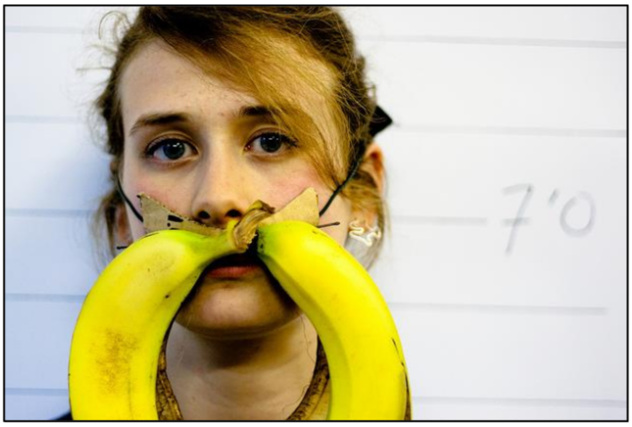
Chapter 3

Datasets

For our work, we used the VQA dataset (also known as VQA-v1) [3] published in 2016. Images from Microsoft COCO dataset [49] were used to make this dataset. Questions and answers were generated using AMT (Amazon Mechanical Turk) workers who were asked to try to ask questions that will stump *a smart robot*. 3 questions were asked for each image. For open-ended questions, answers were collected from 10 workers and for an open ended question's answer to be deemed 100% correct, at least 3 workers had to provide the exact same answer. For multiple choice answers, 18 candidate answers were chosen from four sets of answers: correct, plausible (answers collected without the image being provided) , popular and random. Workers were instructed to keep the answers matter-of-fact and as simple and short as possible. VQA has two parts: real and abstract. We evaluated our model on the "open-ended' task of both parts.

3.1 VQA-real

VQA-real consists of 82,783 training images, 40,504 validation images and 81,434 test images from the newly-released Microsoft Common Objects in Context (MS COCO) dataset. Given how general and high-level the task of VQA is the dataset needs to capture the diversity and complexity of visuals in real life. VQA-real images contain multiple objects from a wide variety of object classes and a rich variety of visual information.

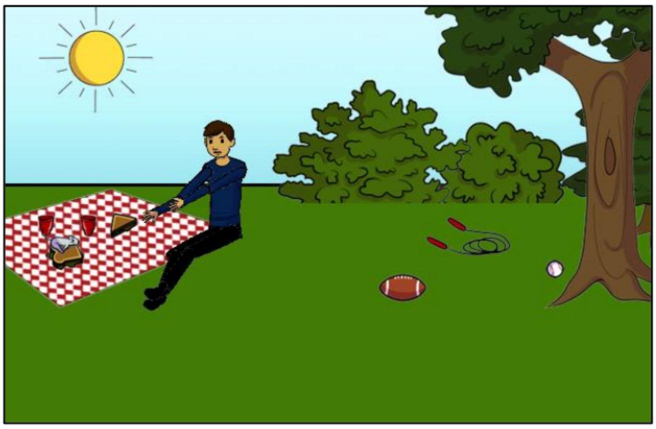


What color are her eyes?
What is the mustache made of?

Figure 3.1: Example image and question from VQA-real

3.2 VQA-abstract

VQA-abstract is a smaller dataset of 50K (20k training, 10k validation and 20k test) images. The task of VQA not only requires low-level visual recognition but also high-level reasoning. VQA-abstract was made keeping this in mind. Images were made using software from clipart objects. The dataset contains 20 “paperdoll” human models spanning genders, races, and ages with 8 different expressions. The set contains over 100 objects and 31 animals in various poses. Scenes are either ’indoors’ or ’outdoors’. The scenes were made so that they closely resemble actual scences in real life.

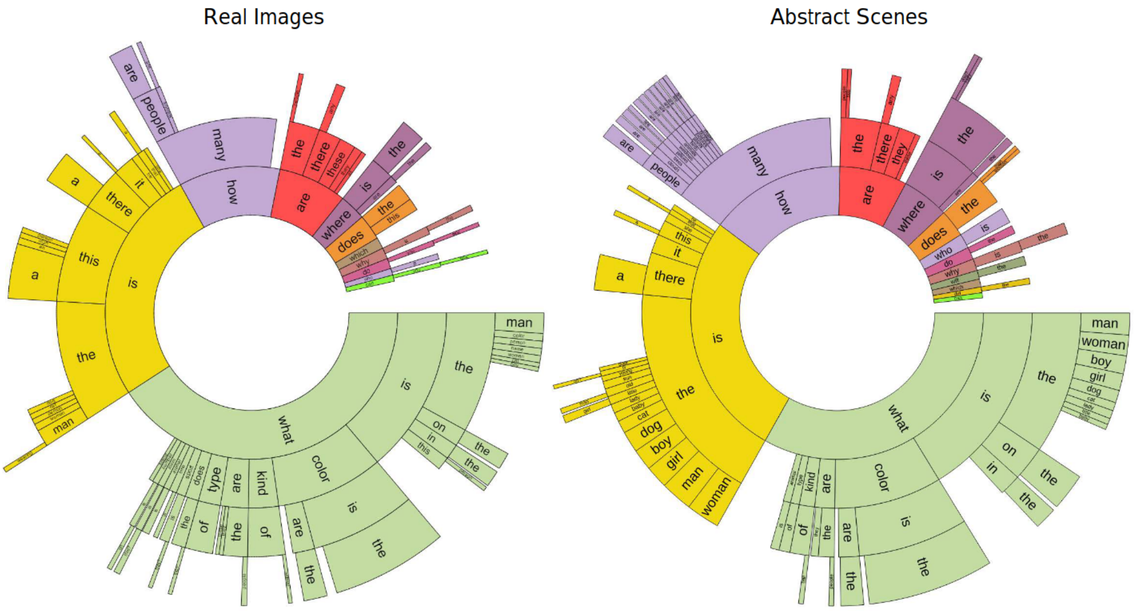


Is this person expecting company?
What is just under the tree?

Figure 3.2: Example image and question from VQA-abstract

3.3 Biases

Here we mention the biases present in the VQA dataset. Although VQA-v1 is the most used and popular dataset in VQA, it has some serious bias issues. Notably in the "yes/no" question category, a model can achieve 55.86% accuracy just by answering "yes" to every question. The answer distributions of other question types are also skewed. For example: the most common sport answer "tennis" is the correct answer for 41% of the questions starting with "What sport is" and "2" is the correct answer for 39% of the questions starting with "How many". There is also a visual priming bias that originates from the question generation process. For example: Most workers asked "is there a dog in the image?" only when there indeed was a dog in the image. Another example of the bias issue is blindly answering "yes" to all questions starting with "Do you see a ..?" which gets the correct answer 87% of the time. Many models tend to leverage these biases to boost their accuracies. This should be kept in mind when we present our results and before any comparison is done with other models.


Real Images

Abstract Scenes

Figure 3.3: Distribution of questions by their first four words for a random sample of 60K questions for real images (left) and all questions for abstract scenes (right). The ordering of the words starts towards the center and radiates outwards. The arc length is proportional to the number of questions containing the word. White areas are words with contributions too small to show. Figure from [3]

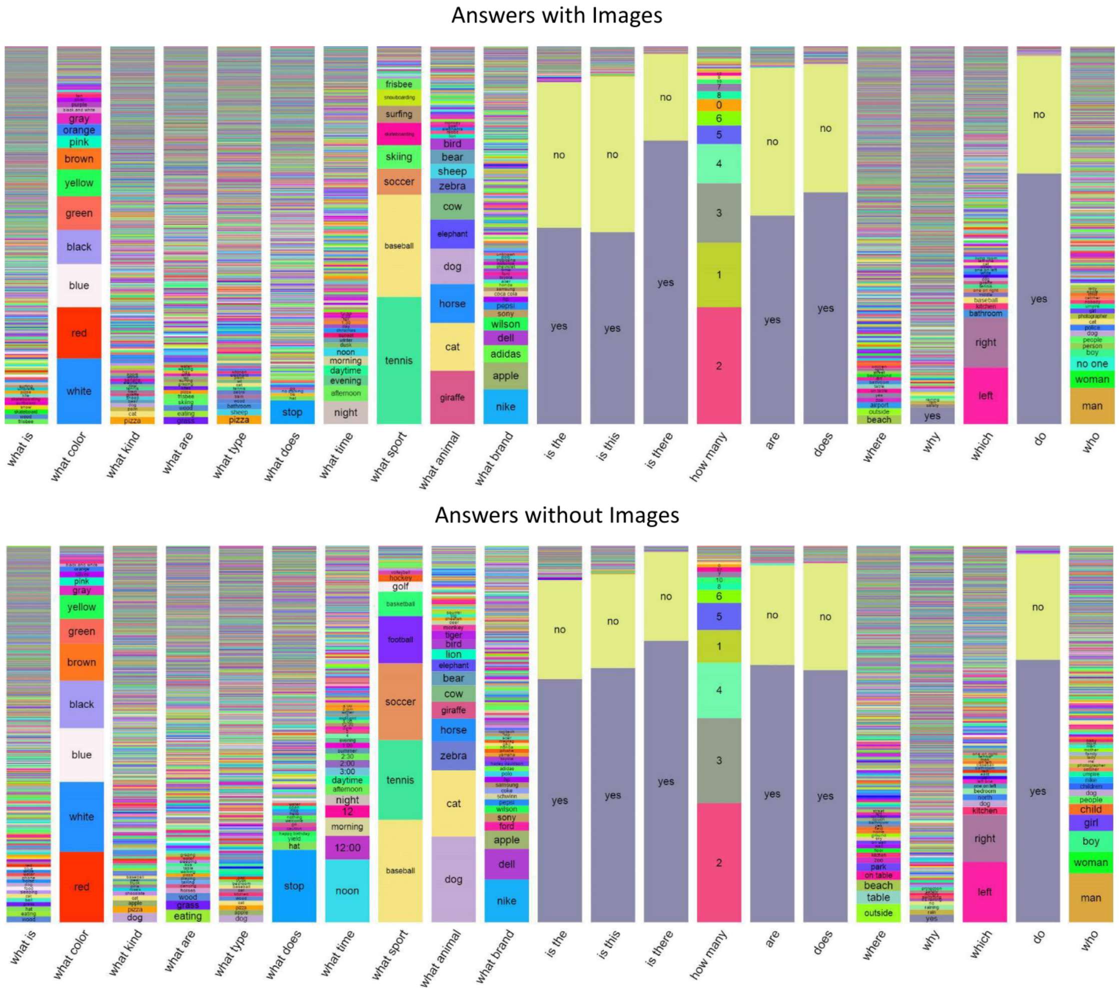


Figure 3.4: Distribution of answers per question type for a random sample of 60K questions for real images when subjects provide answers when given the image (top) and when not given the image (bottom). Figure from [3]

Chapter 4

Evaluation

Evaluating open-ended questions is difficult and there is still no widely agreed-upon evaluation metric for this task. Evaluation becomes difficult because there can be multiple valid answers for the same question. For example: “What is the color of the dress?” can be answered using “white”, “tan”, “off-white”, all of which are correct. So an exact matching scheme would be clearly unsuitable. For our task, we followed the “consensus” metric proposed in the VQA-v1 paper. The metric is:

$$Accuracy_{VQA} = min(\frac{n}{3}, 1) \tag{4.1}$$

where n is the number of annotators with the same answer as the predicted one. Basically this metric considers any prediction correct if it matches with at least 3 annotators. There are problems with this metric too. It turns out that not all questions have annotators agreeing what the answer should be. Especially 59% of “why” questions have no answer with more than two annotators which means the metric will consider any answer for these questions as incorrect. Moreover there are questions where directly conflicting answers (such as: yes and no) both have at least three annotators, meaning for some questions both “yes” and “no” are considered valid answers. Despite these faults, as there is no better evaluation method available, we use the metric mentioned above.

Chapter 5

Architecture

5.1 Basic Approach

Most VQA algorithms follow a basic structure:

- Image representation
- Question representation
- Fusion/Attention
- Answering

5.2 Baseline Model

In this work, we present a baseline model. This is an important distinction as a baseline model is very different from a fully-fledged model. Basically a baseline model is a smaller, simpler model without much complexity in terms of architecture. A baseline model works as a benchmark against which more sophisticated models can be compared. A baseline model has very simple architecture and approaches the problem in a very simple way. Often a baseline model is implemented with the first solution that comes to mind, a “no-brainer”. It is important to understand that the goal of a baseline model is not to achieve high accuracy or solve the problem task but rather to provide “a stick in the sand” which can be used to compare the results of other more serious and sophisticated

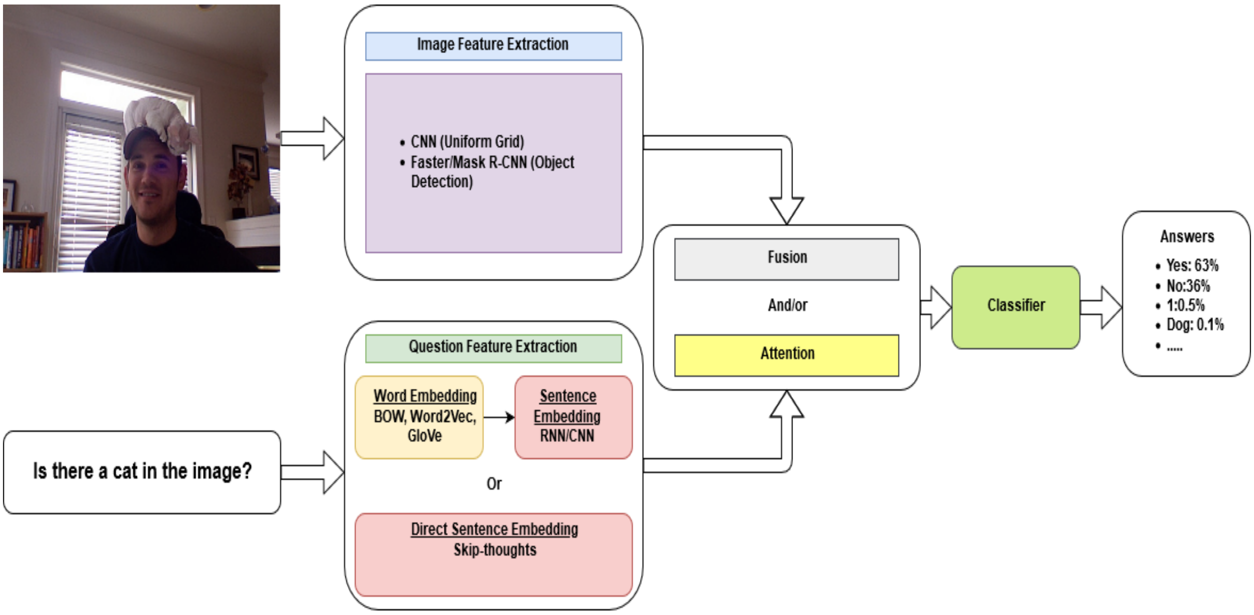


Figure 5.1: Basic structure of most VQA algorithms

models. If a model performs much better than the baseline it is a strong indication that the model works and the solution process is heading in the right direction. On the other hand if the model underperforms against the baseline or just does slightly better, it indicates that the model needs further work. If a substantial increase in model complexity doesn't result in a corresponding substantial increase in accuracy than the baseline results, it indicates that the solution might lie in another direction. Baseline models are an important step in solving any modern machine learning task.

We choose to implement a baseline model in order to evaluate BERT's performance for question embedding in VQA. BERT is a recent language model from Google which has achieved state-of-the-art results in many NLP tasks. BERT is said to produce superior word and sentence embeddings with better understanding of langauge context. We hypothesize that using BERT would result in a better baseline accuracy. To test this hypothesis we chose a baseline model instead of a fully fledged model for the following reasons:

- The simple structure of a baseline model means we have to be less concerned with unnecessary details and can concentrate fully on evaluating BERT's performance. As we are not concerned with achieving state-of-the-art result, a baseline model is more suitable for our aim.

- A baseline model is much simpler, easier and quicker to implement.

- Limited resources is a major limitation we faced and one of the main reasons behind our deciding to implement a baseline model. We do not have access to enough computation power necessary for implementing a fully-fledged vqa model in a reasonable amount of time. As the dataset is quite large, a baseline model is more feasible for us.

We compare our baseline to the original baselines proposed in the vqa-v1 dataset. We also compare our baseline to mainstream vqa models to give a fuller picture of the VQA landscape.

5.3 Our model

Our model is based on the architecture of the model in [1] with the key exception being in the question processing pipeline.

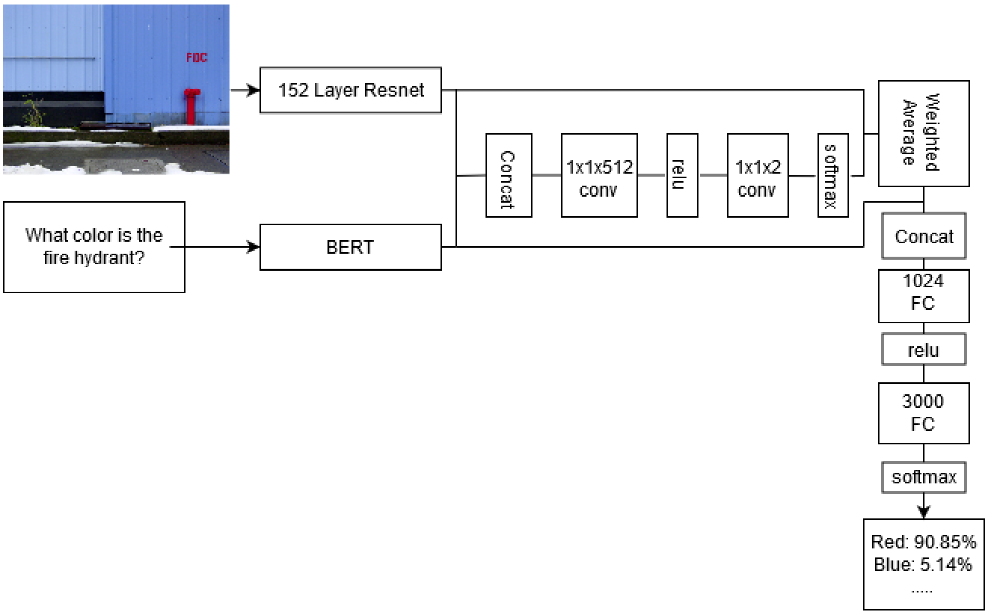


Figure 5.2: Architecture of our model

Chapter 6

BERT

The main goal of our work was to try and see how BERT performs in VQA. BERT[2] is a language model from Google released in late 2018. It has caused quite a stir by outperforming the state-of-the-art in many NLP tasks. In VQA, embeddings of question sentences have to be produced. Till now the most popular method of extracting these embeddings was to first produce word embeddings using either word2vec or GloVe and then feeding those word embeddings to an LSTM or GRU to produce the final sentence embedding. Since work on VQA began, many better language representation models have become available, one of which is BERT.

6.1 Background

Language modelling has always been a central research in natural language processing. Text can not be used directly to train machine learning models. They have to converted to some sort of numerical representation first. This is the key function of a language model. Given a word, sentence or document it produces, a numerical tensor which encodes the correct semantic meaning. Capturing the underlying semantic meaning is the key challenge in language modelling. With the advent of transfer-learning, pre-trained language models have become very useful. Anyone tackling a task which requires text as input and requires preserving the underlying semantics can extract embeddings from a pre-trained language model instead of having to train a new model from scratch. Moreover, these pre-trained models are usually trained on huge text databases such as: Wikipedia or large book corpora which means the embeddings they produce can capture language meaning to a significant level.

6.2 What is Bert

BERT is a language representation model. It can be used to produce embeddings of words and sentences which is basically turning word and sentences into numerical vectors which can then be used in a neural network pipeline. Other language models have been used in VQA such as word2vec and GloVe but what makes BERT better is that unlike word2vec and GloVe it is context sensitive. When word2vec or GloVe produces a word embedding, it doesn't take into account the context in which the word appears. So for example, for the word "cabinet" in the sentences "The cabinet has adjourned" and "Put the bottle in the cabinet", word2vec and GloVe would produce the same embeddings despite the meaning of "cabinet" in the two sentences being completely different. But BERT would produce two different embeddings taking into account the context of the surrounding words. word2vec and GloVe are said to produce static embeddings while BERT produces dynamic embeddings which is more useful for natural language processing.

Another big advantage of BERT is that it understands language flow better. This results from the unique way BERT was trained. Previous language models were mainly trained by reading text from left to right (or right to left) in a unidirectional manner or by combining both unidirectional trainings. But BERT was trained simultaneously for all sentence tokens which eliminated any direction bias resulting in better understanding of language flow and overall structure.

6.3 How Bert works

BERT makes use of Transformers [50], an attention mechanism that learns contextual relations between words (or sub-words) in a text. As opposed to conventional methods, transformers read the entire sequence of input text at once rather than from left to right. This makes BERT a bidirectional model(actually non-directional would be a better word).

The training scheme of a model is important to understand why the model can be useful. BERT was trained using two tasks: MLM (Masked Language Modelling) and NSP (Next Sentence Prediction). When BERT is trained, it is trained jointly for both MLM and NSP with the goal of minimizing the combined loss function of both tasks.

6.3.1 Masked Language Modelling

In masked language modelling, 15% of the input words are replaced with a “MASK” token and the model has to predict these masked tokens using the context provided by non-masked tokens. The input with it’s masked and non-masked tokens are passed through the model. The output from the model is multiplied by the BERT embedding matrix to transform them into vocabulary dimension which is then passed to a classification layer which predicts the masked tokens from the vocabulary. In truth, not all of the 15% tokens are replaced with the “MASK” token. 80% are replaced with the “MASK” token, 10% with a random token and 10% remain the same. The intuition behind this is as follows:

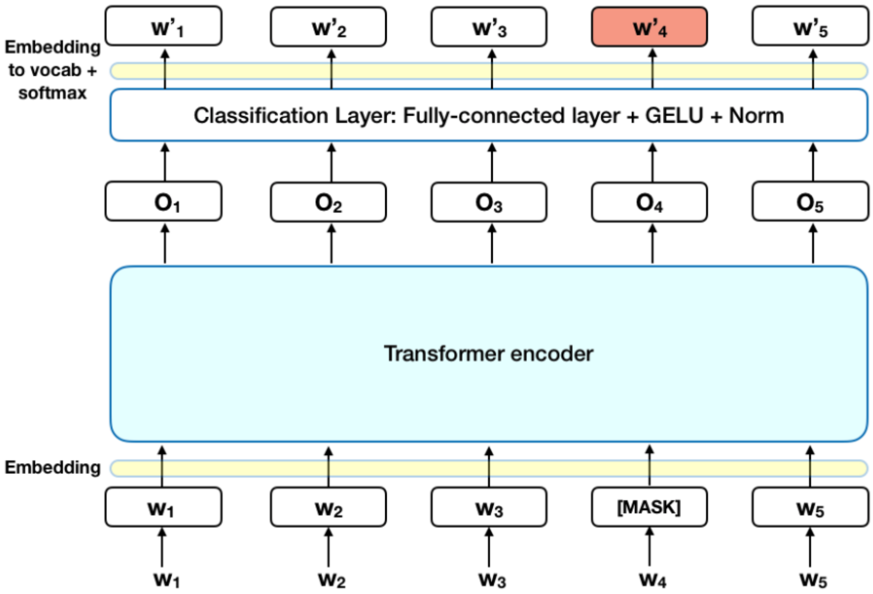

w'1
w'2
w'3
w'4
w'5
Embedding to vocab + softmax
Classification Layer: Fully-connected layer + GELU + Norm
O1
O2
O3
O4
O5
Transformer encoder
Embedding
w1
w2
w3
[MASK]
w5
w1
w2
w3
w4
w5

Figure 6.1: Model scheme for masked language modelling. Figure from [2]

- If 100% of the tokens were masked, the model would be optimized to predict only the masked tokens, not the non-masked ones.
- If 90% of the tokens were masked and 10% random tokens were used, this would teach the model that the observed word is never correct.
- If 90% of the tokens were masked and 10% were kept same, then the model could just trivially copy the non-contextual embedding.

6.3.2 Next Sentence Prediction

In next sentence prediction, two sentences are passed to the model and it has to predict whether the second sentence follows the first sentence. For 50% of the training pairs, the second sentence indeed follows the first sentence in the original document and for the other 50%, the second sentence is another random sentence from the document. NSP teaches the model to learn context on a sentence-wide level and understand the natural flow of language. A “CLS” token is inserted at the beginning of the first sentence and a “SEP” token is inserted at the end of both sentences. Sentence embedding indicates which sentence a particular token belongs to. It is set to 0 for the first sentence and 1 for the second. Positional embedding just embeds the position of the token.

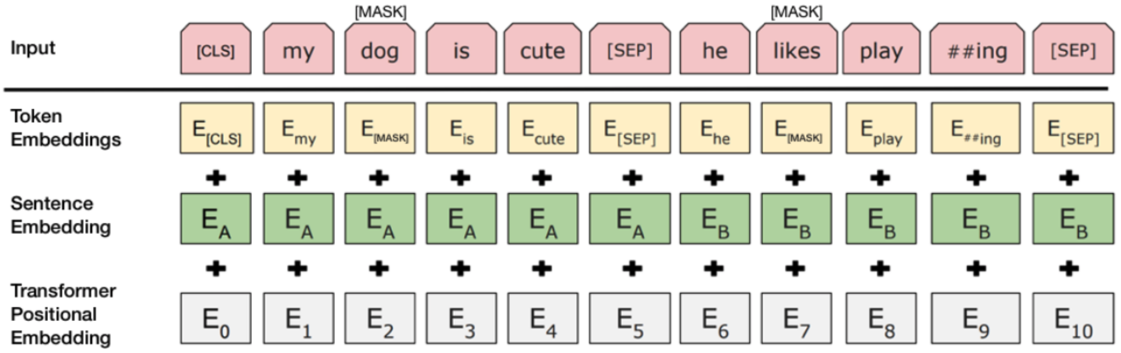


Figure 6.2: Model scheme for next sentence prediction. Figure from [2]

Chapter 7

Method

Figure 5.2 shows the overall structure of our model. The structure is mostly based on [1] but the novelty of our work comes from the question feature extraction part. The original model tokenized the question q into a set of n tokens $\mathbf{E} = \{e_1, e_2, ...e_n\}$ using a simple embedding layer. These tokens are then fed into an LSTM layer and the final state of the LSTM is used to represent the question. In our model, we use pre-trained BERT to extract sentence embeddings for every question.

7.1 Image Embedding

To extract image features we use a pre-trained CNN based on [51]. We divide each image into a 14×14 grid. The final output dimensions are $14 \times 14 \times 2048$ which is taken from the last layer before the final pooling layer. L2 normalization is also performed on the last dimension.

$$v = CNN(I), v \epsilon \mathbb{R}^{14\times14\times2048} \tag{7.1}$$

7.2 Question Embedding

For the question embedding we used BERT. Google offers various pre-trained versions of BERT. We used the “base” and “uncased” version. The “base” version is smaller in size than the “large” version which is more suitable for us due to limited resources. The “uncased” version ignores casing which isn’t important for our task. When given a question, BERT first tokenizes the question using its own tokenizer to produce t tokens. These tokens are then fed to the pre-trained model which outputs a $12 \times t \times 768$ tensor where 12 is the number of BERT layers. We take the second-to-last i.e. 11th layer embeddings and average them to get a final 768-dimensional vector which is the final represention of each question.

$$s = BERT(q), s \epsilon \mathbb{R}^{768} \tag{7.2}$$

7.3 Attention

Attention is a useful mechanism that is widely used in VQA. The question is used to produce different weights for different spatial locations of the image. These weights correspond to how relevant those image areas are for answering the question. Attention allows the model to focus on important regions while ignoring regions which will probably not contribute to predicting an answer. This prevents noise information from adversely affecting answer prediction and makes the model more efficient. Attention can be applied more than once in multiple “glimpses”.

We use two layers to glimpse each image two times. We compute attention distributions over the spatial dimensions of the image features.

$$\alpha_{c,l} \propto \exp F_c(s, v_l) \quad \sum_{l=1}^{L} \alpha_{c,l} = 1 \tag{7.3}$$

$$x_c = \sum_{l} \alpha_{c,l} v_l \tag{7.4}$$

Each image feature glimpse x_c is the weighted average of image features v over all the spatial locations $l = 1, 2, ..., L$. The attention weights $\alpha_{c,l}$ are normalized separately for each glimpse $c = 1, 2, ..., C$. In practice $F = [F_1, F_2, ..., F_c]$ is modeled with two layers of convolution. Consequently F_i's share parameters in the first layer. We solely rely on different initializations to produce diverse attention distributions.

7.4 Classifier

Ideally an open-ended question requires an open-ended answer. But there are currently no evaluation methods available to correctly assess open-ended answers. Though some models have used RNNs to generate answers dynamically, most models have opted for predicting answers rather than generate them. VQA models generally uses the top-K answers from the whole set of answers and uses a classifier to output a probability distribution over them. The answer with the most probability is deemed as the correct answer.

Finally we concatenate the attended features with the sentence embeddings and pass to a FC-relu layer followed by a FC-softmax layer which outputs a probability distribution over the top-3000 answers. The answer with the highest value is the predicted answer.

7.5 Implementation Details

Input images are scaled while preserving aspect ratio and center cropped to 299 × 299 dimensions. We implement attention by using two consequent convolution layers. The first convolution layer is 1×1 dimensional with depth 512 and is followed by a relu layer. The second layer is 1×1 with depth 2 followed by softmax. We only consider the top 3000 most frequent answers in our classifier. Other answers are ignored and do not contribute to the loss during training. This covers 92% of the answers in the validation set in VQA dataset. We apply a dropout rate of 0.5 to all inputs of all layers i.e. all convolution and fully-connected layers. This is necessary to prevent overfitting. We use the adam optimizer. We use exponential decay to gradually decrease the learning rate according to the following equation,

$$l_{step} = 0.5^{\frac{step}{learningratehalf-life}} l_0 \tag{7.5}$$

Chapter 8

Results and Discussion

8.1 VQA-abstract

8.1.1 Training Details

- Task: Open-ended
- Epochs: 50
- Batch size: 128
- Initial learning rate: 0.001
- Learning rate half-life: 4690 iterations (Learning rate is essentially halved every 10 epochs)

8.1.2 Accuracy and Loss

- Accuracy: 61.3
- Loss: 1.583

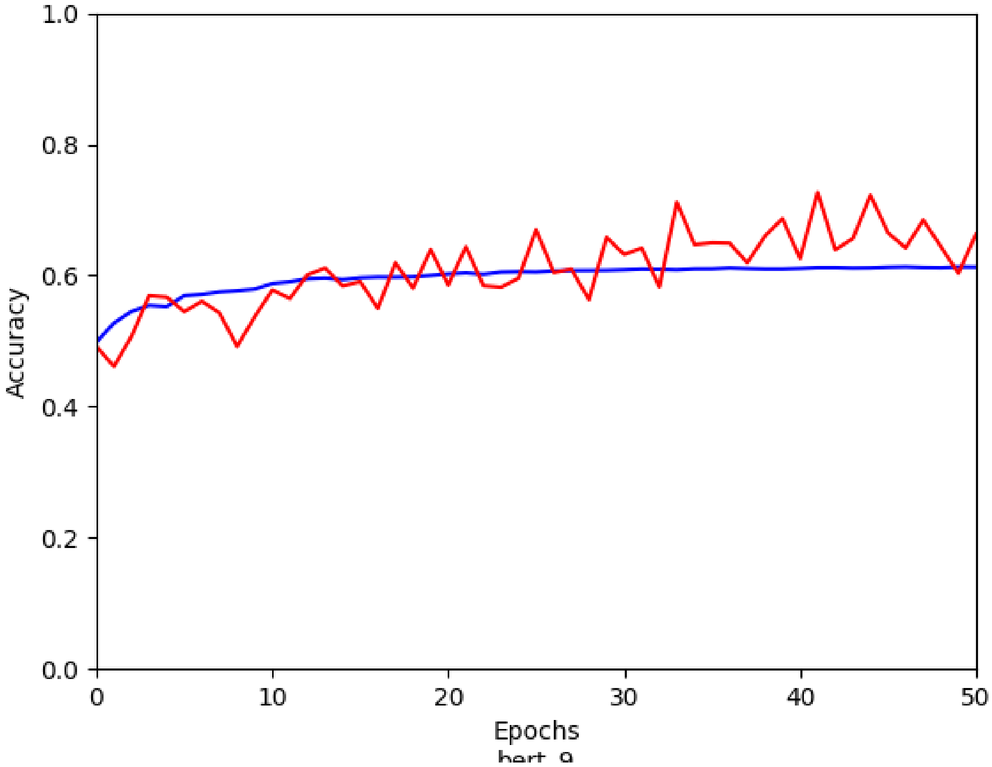


Figure 8.1: Graph of training(red) and validation(blue) accuracy for vqa-abstract

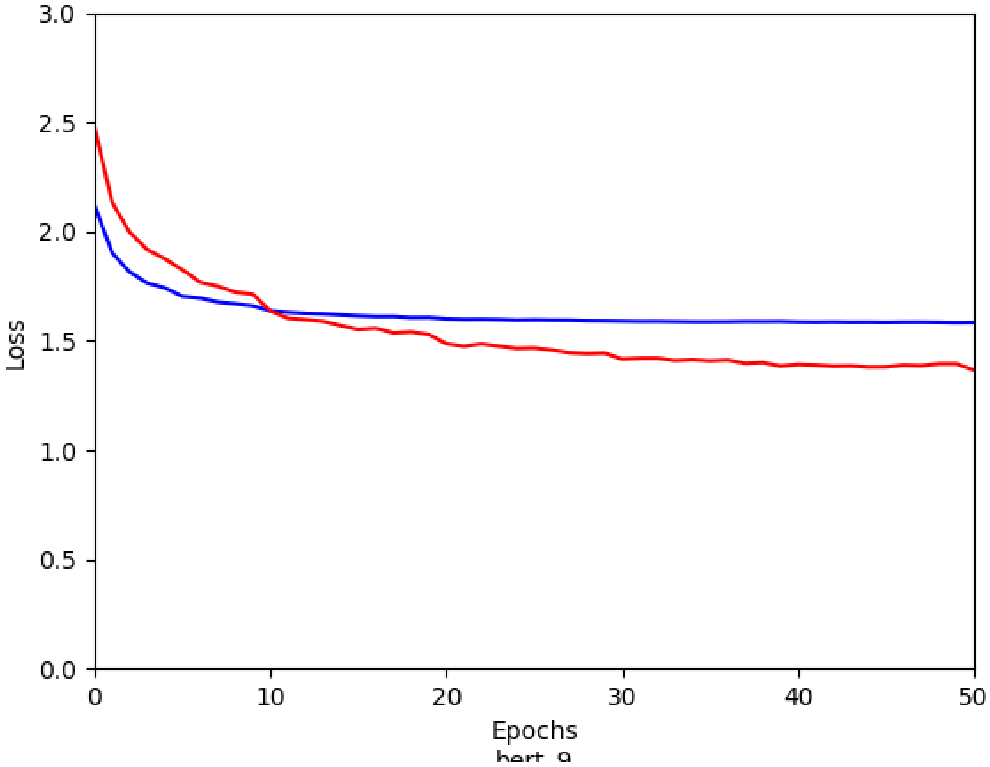


Figure 8.2: Graph of training(red) and validation(blue) loss for vqa-abstract

8.2 VQA-real

8.2.1 Training Details

- Task: Open-ended
- Epochs: 100
- Batch size: 128
- Initial learning rate: 1e-3
- Learning rate half-life: 50000 iterations

8.2.2 Accuracy and Loss

- Accuracy: 57.35
- Loss: 1.5661

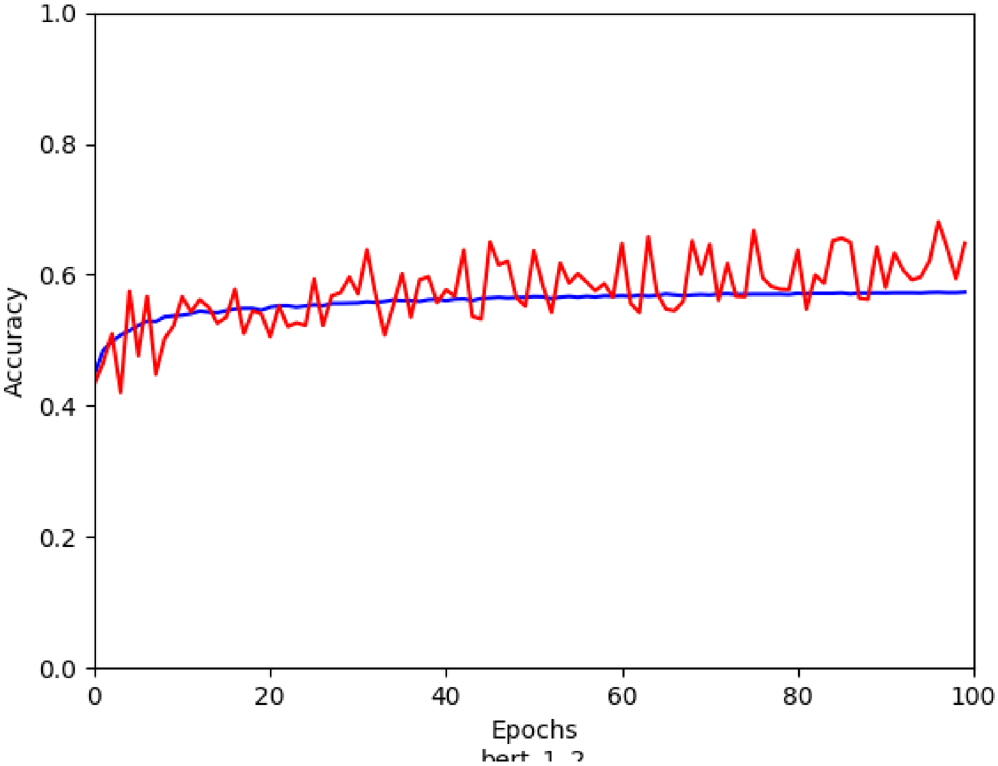


Figure 8.3: Graph of training(red) and validation(blue) accuracy for vqa-real

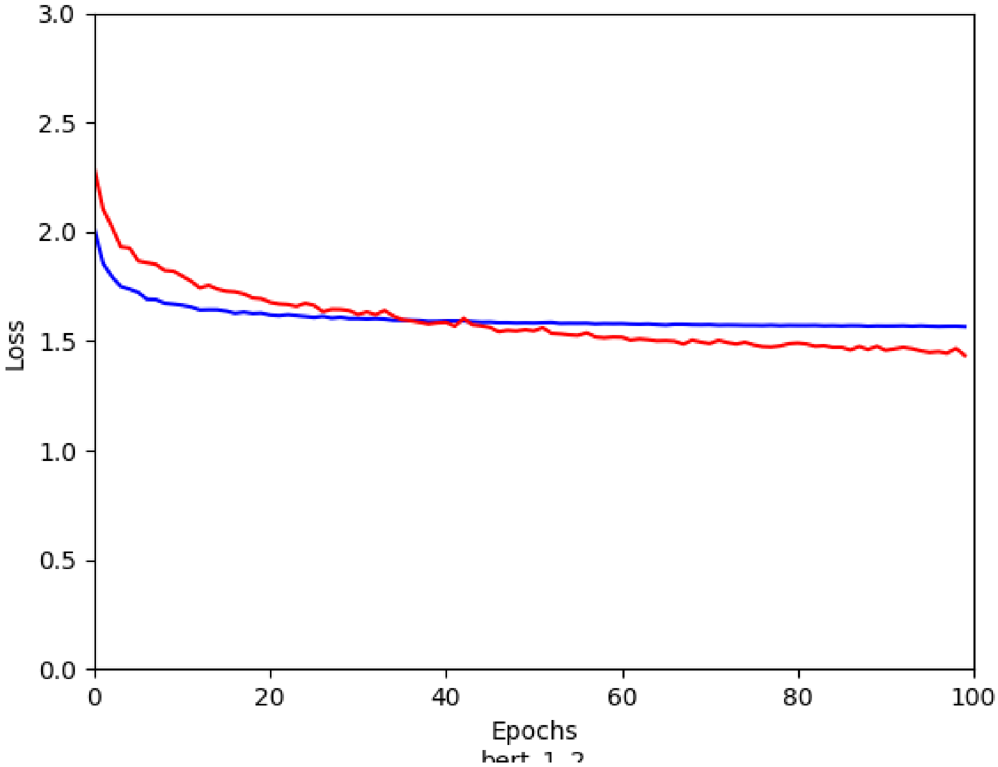


Figure 8.4: Graph of training(red) and validation(blue) loss for vqa-real

8.3 Discussion

We compare our performance on VQA-real with the baselines proposed in the original paper. The original paper proposed the following baselines:

- **random:** A random answer is chosen from top 1K answers from train+val split
- **always yes:** Always answer “yes” which is the most frequent answer in the entire dataset
- **per Q-type prior:** The most popular answer is given according to question type
- **nearest neighbour:** The most frequent answer from the top K most similar questions and their associated images

- **I:** Image embedding from a pre-trained CNN
- **norm I:** L2-normalized image embedding from a pre-trained CNN
- **BoW Q:** Question embedding using Bag-of-words
- **LSTM Q:** Question embedding using one hidden layer LSTM
- **deeper LSTM Q:** Question embedding using two hidden layer LSTM

random	10.00
I	28.13
always yes	29.66
per Q-type prior	37.54
nearest neighbour	42.70
BoW Q	48.09
LSTM Q	48.76
deeper LSTM Q	50.39
BoW Q + I	52.64
LSTM Q + I	53.74
Ours	**57.35**
deeper LSTM Q + norm I	57.75

Table 8.1: Comparison with other baseline models from [3]

Our model beats all baselines except one. But it should be noted here that most of these other baseline are in most likelihood using question condition biases to improve their accuracy. This can be seen from the fact that adding image to “BoW Q”, “LSTM Q” and “deeper LSTM Q” only results in an increase of 4.55%, 4.98% and 7.36% respectively. There is more chance of learning bias in using BoW, LSTM or deeper LSTM as the question embedding weights are not

iBOWIMG(Baseline) [52]	55.89
Ours	**57.35**
DPPNet [53]	57.36
SMem [21]	58.24
NMN [31]	58.7
SAN [19]	58.9
FDA [10]	59.54
HieCoAtt [22]	62.1
MCB + Att [26]	64.2
MLB [27]	65.07
HUMAN	83.30

Table 8.2: Comparison with other baselines and full VQA models

fixed beforehand but are learned completely from the biased dataset. On the other hand, BERT weights have been learned beforehand using balanced datasets and we only extract pre-trained embeddings for VQA questions. As there are now VQA datasets available with less bias issues, testing of BERT performance on them are likely to yield better results. We conclude from our work that BERT embeddings are indeed useful for embedding questions in the context of VQA.

For completeness, we also provide comparison with some other baselines and full VQA models.

Chapter 9

Future Work and Conclusion

9.1 Future Work

We chose VQA-v1 as our dataset despite it's numerous bias issues because it was easier in case of implementation and because we wanted to do a preliminary test of BERT's performance in VQA. There are now several other datasets available with less bias issues notably: VQA-v2[54], CLEVR[30], GQA[55], TDIUC[56]. VQA-v2 is a good choice for further testing of BERT. VQA-v2 mitigates the bias issues of VQA-v1 by having two complimentary images for each question but with two different answers. So any model is forced to look at the image to derive the correct answer and rely less on language based biases. This means evaluation and comparison of BERT with other baseline methods on this dataset would be much clearer and concrete and hopefully make BERT stand out.

9.2 Conclusion

In this work, we used BERT to extract sentence embeddings for VQA questions and tested a baseline model to evaluate BERT's performance in VQA. We presented our results on VQA-abstract (61.3% accuracy) and VQA-real (57.35% accuracy). We compared our baseline model to various other baselines and full VQA models. Finally we concluded that BERT sentence embeddings are indeed useful in the context of VQA and further research using BERT as the question embedding layer should be carried out.

References

[1] V. Kazemi and A. Elqursh, "Show, ask, attend, and answer: A strong baseline for visual question answering," *arXiv preprint arXiv:1704.03162*, 2017.

[2] 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.

[3] 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.

[4] M. Malinowski, M. Rohrbach, and M. Fritz, "Ask your neurons: A deep learning approach to visual question answering," *International Journal of Computer Vision*, vol. 125, no. 1-3, pp. 110–135, 2017.

[5] Y. Lin, Z. Pang, D. Wang, and Y. Zhuang, "Feature enhancement in attention for visual question answering." in *IJCAI*, 2018, pp. 4216–4222.

[6] A. Jiang, F. Wang, F. Porikli, and Y. Li, "Compositional memory for visual question answering," *arXiv preprint arXiv:1511.05676*, 2015.

[7] 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.

[8] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6077–6086.

[9] P. Gao, Z. Jiang, H. You, P. Lu, S. C. Hoi, X. Wang, and H. Li, "Dynamic fusion with intra-and inter-modality attention flow for visual question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6639–6648.

[10] I. Ilievski, S. Yan, and J. Feng, "A focused dynamic attention model for visual question answering," *arXiv preprint arXiv:1604.01485*, 2016.

[11] 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.

[12] D.-K. Nguyen and T. Okatani, "Improved fusion of visual and language representations by dense symmetric co-attention for visual question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6087–6096.

[13] A. Osmana and W. Sameka, "Drau: Dual recurrent attention units for visual question answering."

[14] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[15] 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.

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

[17] 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.

[18] C. Ma, C. Shen, A. Dick, Q. Wu, P. Wang, A. van den Hengel, and I. Reid, "Visual question answering with memory-augmented networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6975–6984.

[19] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola, “Stacked attention networks for image question answering,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 21–29.

[20] J. Song, P. Zeng, L. Gao, and H. T. Shen, “From pixels to objects: Cubic visual attention for visual question answering.”

[21] 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.

[22] J. Lu, J. Yang, D. Batra, and D. Parikh, “Hierarchical question-image co-attention for visual question answering,” in *Advances In Neural Information Processing Systems*, 2016, pp. 289–297.

[23] H. Nam, J.-W. Ha, and J. Kim, “Dual attention networks for multimodal reasoning and matching,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 299–307.

[24] I. Schwartz, A. Schwing, and T. Hazan, “High-order attention models for visual question answering,” in *Advances in Neural Information Processing Systems*, 2017, pp. 3664–3674.

[25] Z. Yu, J. Yu, Y. Cui, D. Tao, and Q. Tian, “Deep modular co-attention networks for visual question answering,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6281–6290.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. Lawrence Zitnick, and R. Girshick, “Clevr: A diagnostic dataset for compositional language and elementary visual reasoning,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2901–2910.

[31] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein, “Neural module networks,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[32] ——, “Learning to compose neural networks for question answering,” *arXiv preprint arXiv:1601.01705*, 2016.

[33] R. Hu, J. Andreas, M. Rohrbach, T. Darrell, and K. Saenko, “Learning to reason: End-to-end module networks for visual question answering,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 804–813.

[34] J. Johnson, B. Hariharan, L. van der Maaten, J. Hoffman, L. Fei-Fei, C. Lawrence Zitnick, and R. Girshick, “Inferring and executing programs for visual reasoning,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2989–2998.

[35] J. Suarez, J. Johnson, and F.-F. Li, “Ddrprog: A clevr differentiable dynamic reasoning programmer,” *arXiv preprint arXiv:1803.11361*, 2018.

[36] D. Mascharka, P. Tran, R. Soklaski, and A. Majumdar, “Transparency by design: Closing the gap between performance and interpretability in visual reasoning,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4942–4950.

[37] Q. Cao, X. Liang, B. Li, and L. Lin, “Interpretable visual question answering by reasoning on dependency trees,” *arXiv preprint arXiv:1809.01810*, 2018.

[38] K. Yi, J. Wu, C. Gan, A. Torralba, P. Kohli, and J. Tenenbaum, “Neural-symbolic vqa: Disentangling reasoning from vision and language understanding,” in *Advances in Neural Information Processing Systems*, 2018, pp. 1031–1042.

[39] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, “Film: Visual reasoning with a general conditioning layer,” in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[40] Y. Yao, J. Xu, F. Wang, and B. Xu, “Cascaded mutual modulation for visual reasoning,” *arXiv preprint arXiv:1809.01943*, 2018.

[41] D. A. Hudson and C. D. Manning, “Compositional attention networks for machine reasoning,” *arXiv preprint arXiv:1803.03067*, 2018.

[42] Q. Wu, P. Wang, C. Shen, A. Dick, and A. van den Hengel, “Ask me anything: Free-form visual question answering based on knowledge from external sources,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4622–4630.

[43] P. Wang, Q. Wu, C. Shen, A. v. d. Hengel, and A. Dick, “Explicit knowledge-based reasoning for visual question answering,” *arXiv preprint arXiv:1511.02570*, 2015.

[44] Y. Zhu, J. J. Lim, and L. Fei-Fei, “Knowledge acquisition for visual question answering via iterative querying,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1154–1163.

[45] G. Li, H. Su, and W. Zhu, “Incorporating external knowledge to answer open-domain visual questions with dynamic memory networks,” *arXiv preprint arXiv:1712.00733*, 2017.

[46] P. Wang, Q. Wu, C. Shen, A. Dick, and A. van den Hengel, “Fvqa: Fact-based visual question answering,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 10, pp. 2413–2427, 2018.

[47] M. Narasimhan and A. G. Schwing, “Straight to the facts: Learning knowledge base retrieval for factual visual question answering,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 451–468.

[48] M. Narasimhan, S. Lazebnik, and A. Schwing, “Out of the box: Reasoning with graph convolution nets for factual visual question answering,” in *Advances in Neural Information Processing Systems*, 2018, pp. 2654–2665.

[49] 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.

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

[51] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[52] B. Zhou, Y. Tian, S. Sukhbaatar, A. Szlam, and R. Fergus, "Simple baseline for visual question answering," *arXiv preprint arXiv:1512.02167*, 2015.

[53] H. Noh, P. Hongsuck Seo, and B. Han, "Image question answering using convolutional neural network with dynamic parameter prediction," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 30–38.

[54] 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.

[55] D. A. Hudson and C. D. Manning, "Gqa: A new dataset for real-world visual reasoning and compositional question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6700–6709.

[56] K. Kafle and C. Kanan, "An analysis of visual question answering algorithms," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1965–1973.