공학석사학위논문

Multimodal Self-Attention Network for Visual Reasoning

시각적 추론을 위한 멀티모달 셀프어텐션 네트워크

2019 년 8 월

서울대학교 대학원

산업공학과

류 성 원

Multimodal Self-Attention Network for Visual Reasoning

시각적 추론을 위한 멀티모달 셀프어텐션 네트워크

지도교수 조성준

이 논문을 공학석사 학위논문으로 제출함

2019 년 6 월

서울대학교 대학원

산업공학과

류성원

류성원의 공학석사 학위논문을 인준함

2019 년 7 월

위원장 박종헌 (인)

부위원장 조성준 (인)

위원 장우진 (인)

Abstract

Multimodal Self-Attention Network for Visual Reasoning

Sungwon Lyu
Department of Industrial Engineering
The Graduate School
Seoul National University

Visual reasoning is more difficult than visual question answering since it requires sophisticated control of information from image and question. Extracted information from one source is used to extract information from the other and this process occurs alternately. This is natural since even human needs multiple glimpses of image and question to solve complicated natural language question with multi-step reasoning. One needs to handle information from earlier steps and use them in later steps to get the answer. Due to this difference, the results on these two tasks tend not to correlate closely.

In this paper, we propose Multimodal Self-attention Network (MUSAN) to solve visual reasoning task. Our model uses Transformer encoder by [22] to promote intimate interactions between images and the question in fine granular level. MUSAN achieved state-of-the-art performance in CLEVR dataset from raw pixels without prior knowledge or pretrained feature extractor. Also, MUSAN recorded 8th rank in the 2019 GQA challenge without functional or graphical information. Attention visualization of MUSAN shows that MUSAN performs stepwise reasoning with its own logic.

Keywords: Visual Question Answering, Visual Reasoning, Self-Attention

Student Number: 2017-25671

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Problems of industry are becoming more and more complex and available data to solve the problems are increasing in both volume and kinds. However utilizing different sources of data to solve a problem remain particularly difficult due to statistical difference in data sources.

1.1 Multimodality

In statistics, multimodal distribution refers to a continuous probability distribution with multiple modes showing distinct peaks. Analyzing multimodal distribution is usually more difficult because multimodal distribution usually has multiple factors affecting a distribution. In this perspective, solving multimodality problem meant disentangling different factors affecting a single distribution.

However, the meaning of multimodality problem has been widened due to increasing complexity of data. Compared to structured data, unstructured data such as images, texts and audios not only have multiple modes, but also have extremely complicated and distinct properties. Many machine learning techniques analyized these unstructured data by adapting datatype specific methods. However, these approaches were limited when a problem involves multiple sources of unstructured data.

1.2 Visual Question Answering

One of the most popular tasks tackling this multimodality is visual question answering. Visual question answering involves a image and a corresponding question which needs information of the image. This task requires to extract information from two very different unstructured data domains, images and texts.

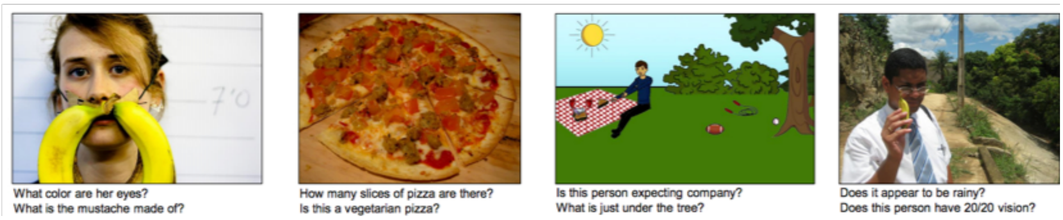

What color are her eyes?
What is the mustache made of?
How many slices of pizza are there?
Is this a vegetarian pizza?
Is this person expecting company?
What is just under the tree?
Does it appear to be rainy?
Does this person have 20/20 vision?

Figure 1.1: Examples of the VQA dataset [1]

Since a large amount of annotated image, quesion and answer pairs are difficult to collect, VQA researches are usually conducted with open dataset. [1] opened VQA challenge in 2016 with the release of VQA dataset and the challenge is held every year. Examples of the VQA problems are in Figure 1.1.

However, the original VQA dataset is known to have a drawback for machines to learn properly from dataset. Several cases reported that machines tend to solve VQA question without referring to the image. [8] pointed out that inherit structure of our world and bias in language are quiet strong enough for machines to learn the bias instead of actual problem solving. For example, according to [8], the most popular sports in VQA dataset is tennis which takes 34% of the sports related answers. Dataset inevitably reflects the bias of the real world. [8] enhanced the VQA dataset to VQA 2.0 to solve this problem by balancing the answer distribution of VQA dataset. However, this problem could not be eliminated.

1.3 Visual Reasoning

Visual reasoning task can be considered as a subcategory of visual question answering which requires reasoning with the objects in the images. While questions in visual question answering are quiet straightforward which ask existence or obvious property of objects, visual reasoning requires precise understanding of natural language question and complicated multi-step reasoning. Reasoning involves not only objects but also their relations. Visual reasoning questions can be richer and more diverse than normal visual questions.

Instead, visual reasoning dataset obviously need more effort than simple vqa dataset. Since possible question space of visual reasoning is exponentially larger, it is important to provide coherent terms about objects and relations for machines to learn visual reasoning with limited number of dataset. Also, structure of question can be much more diverse. For these reasons, visual reasoning dataset tends to be created programatically using image scene graph, cleaned up words and question templates. Automatic generation of questions can be strictly controlled so that the answer distribution of visual reasoning is less affected by language priors.

Visual reasoning is more difficult than visual question answering since it requires sophisticated control of information from image and question. Extracted information from one source is used to extract information from the other and this process occurs alternately. This is natural since even human needs multiple glimpses of image and question to solve complicated natural language question with multi-step reasoning. One needs to handle information from earlier steps and use them in later steps to get the answer. Due to this difference, the results on these two tasks tend not to correlate closely.

In this paper, we propose Multimodal Self-attention Network (MUSAN) to solve visual reasoning task. Our model uses Transformer encoder by [22] to promote intimate interactions between images and the question in fine granular level. MUSAN achieved state-of-the-art performance in CLEVR dataset from raw pixels without prior knowledge or pretrained feature extractor. Also, MUSAN recorded 8th rank in the 2019 GQA challenge without functional or graphical information. Attention visualization of MUSAN shows that MUSAN performs stepwise reasoning with its own logic.

This paper is composed of 5 chapters. In Chapter 2, we review the development of visual question answering, and visual reasoning models. In Chapter 3, we provide detailed description of MUSAN model and self-attention mechanism it primarily use. In Chapter 4, results and analysis of experiments on CLEVR and GQA datasets are presented. Finally, in Chapter 5, we give concluding remarks and possible future research directions of this paper.

Chapter 2

Related Works

2.1 Visual Question Answering Models

Visual reasoning models developed from visual question answering models. Usually VQA performs two steps: feature extraction and features fusion. Most of visual question answering models starts by extracting features from images and words separately since these two have different characteristics.

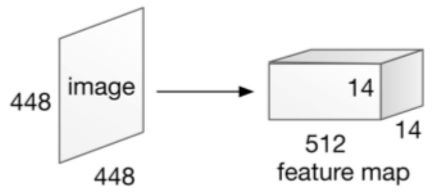


Figure 2.1: Spatial image encoder[24]

Convolution neural network(CNN) structure is proved to be effective in extracting features from data with spatial correlation like images. Rather than summarizing a image into a single vector, a group of object vectors is used to represent each objects in the image. Conventionally, the later part of CNN with the depth of channels is considered to represent a spatial features for each position as in Figure 2.1. Usually, pretrained classifier such as VGG-net[21] or Res-net[9] is used to extract spatial features from images. Later, object detector and even image segmenter is used to extract objects more precisely.

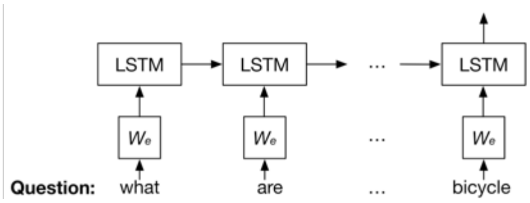


Figure 2.2: RNN question encoder[24]

The most common way to extract features from texts was recurrent neural network(RNN) as in Figure 2.2 since texts have clear sequential structure. The last hidden state of uni-directional RNN or concatenation of the last hidden states of bi-directional RNN is used to represent the text with a single vector. Recently there have been several attempts to use self-attention mechanism to extract features from texts in natural language tasks[22, 6] but few researches have been done in VQA task.

While the methods for feature extraction changed slowly, many VQA models focused on the second step, the fusion of extracted features, to get the answers. There are three main approaches in VQA according to [23]; Attention based models, relation based models and module based models.

2.1.1 Attention based models

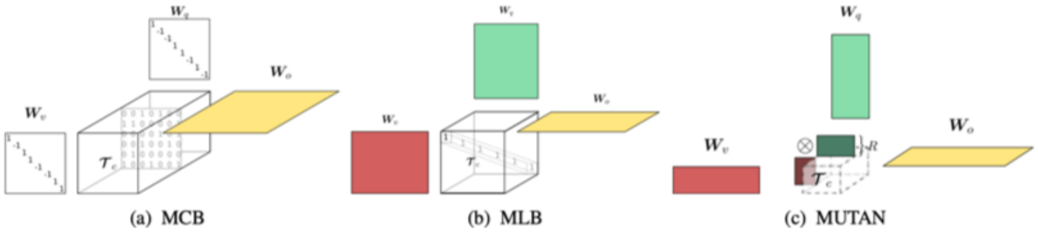
The most popular approach for visual question answering is using attention mechanism to manipulate information. Stacked Attention Network(SAN) established solid baseline for VQA. SAN tried to solved multi-step reasoning problems by stacking several levels of attention on image conditioned on question. After extraction of features, the relation between a question vector and image vectors are captured by the weights of attentionp_I which are determined by element-wise summationh_A.

$$\begin{gathered} h_A = \tanh(W_{I,A} v_I \oplus (W_{Q,A} v_Q + b_A)) \\ p_I = \text{softmax}(W_P h_A + b_P) \\ \tilde{v}_I = \sum_i p_i v_i \end{gathered} \tag{2.1}$$

Following papers kept the main idea of attention mechanism but tried to capture relation between image vectors and a question vector more accurately with expressive modeling. Many works tried to represent the relation with bilinear modeling[7, 16, 17, 5] since it is one of the most expressive methods to represent the relation of two vectors. However, full bilinear modeling between two vectors are inefficient since it needs quadratic number of parameters and computation.

Therefore, these models tried to minimize the number of parameters and reduce the computation while keeping the richness of bilinear modeling. Multimodal Compact Bilinear pooling model(MCB)[7] tried to reduce the computation by performing convolution operation in frequency domain. Multimodal Low-rank Bilinear model(MLB)[17] proved element-wise product after projection is equivalent to low-rank bilinear modeling generalizing Multimodal Residual Network(MRN)[16]. This means that bilinear modeling can be more efficient with low-rank Hadamard products.

$$\begin{gathered} f_i = \sum_{j=1}^{N} \sum_{k=1}^{M} w_{ijk} x_j y_k + b_i = \mathbf{x}^T \mathbf{W}_i \mathbf{y} + b_i \\ = \mathbf{x}^T \mathbf{U}_i \mathbf{V}_i^T \mathbf{y} + b_i = \mathbb{1}(\mathbf{U}_i^T \mathbf{x} \circ \mathbf{V}_i^T \mathbf{y}) + b_i \\ \mathbf{f} = \mathbf{P}^T (\mathbf{U}_i^T \mathbf{x} \circ \mathbf{V}_i^T \mathbf{y}) + \mathbf{b} \end{gathered} \tag{2.2}$$



(a) MCB (b) MLB (c) MUTAN

Figure 2.3: Generalization of former models by MUTAN[5]

Multimodal Tucker Fusion model(MUTAN)[5] represented the relation among three vectors; a image, question, and output vector. To restrain the cubic number of parameters, MUTAN used Tucker decomposition and imposed structured sparsity constraints on the slice matrices. MUTAN showed that former models can be generalized in perspective of Tucker decomposition. Also, the rank of the core tensor can be controlled by the summation of multiple low-rank tensors.

$$\begin{gathered} \tau \in \mathbb{R}^{d_q \times d_v \times |\mathcal{A}|} \\ \tau = ((\tau_{\mathbf{c}} \times_1 \mathbf{W}_q) \times_2 \mathbf{W}_v) \times_3 \mathbf{W}_O \end{gathered} \tag{2.3}$$

Several other works tried to improve the model by extending features. Bottom-up attention model[3] showed dramatically improved performance by using a neural object detector to extract more valid object features. BAN[15] tried to capture more sophisticated attention by modeling interaction of object vectors and word vectors instead of a single question vector. Despite of improvement in VQA dataset, attention based models were not competitive in visual reasoning which require complex natural language understanding and reasoning.

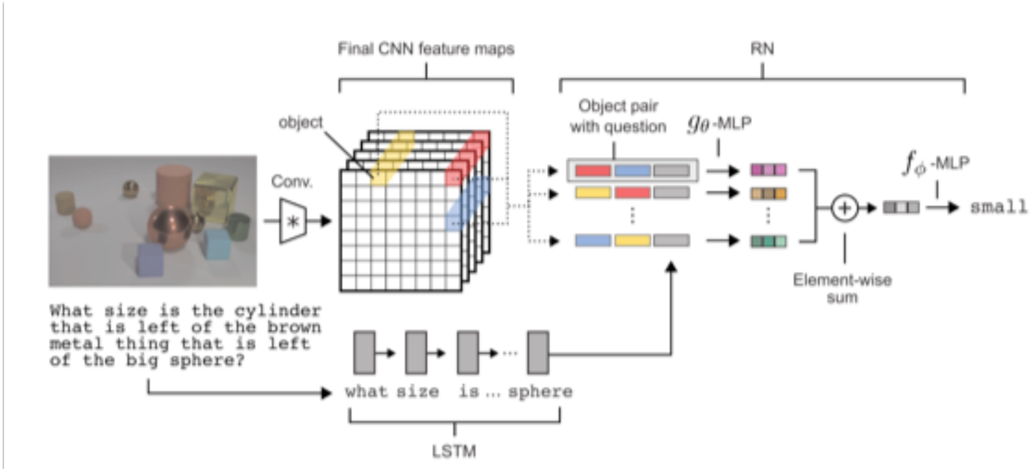

Final CNN feature maps
RN
object
Object pair with question
g_θ-MLP
f_ϕ-MLP
Conv.
small
Element-wise sum
What size is the cylinder that is left of the brown metal thing that is left of the big sphere?
what size is ... sphere
LSTM

Figure 2.4: Relational Network

2.1.2 Relation based models

Many of visual reasoning questions are based on the relationship among objects. Relational Network(RN)[20] provided inductive bias for relations by aligning all object in pairs. Despite of good performance of RN on CLEVR dataset, the use is limited due to quadratic increase of computation on the number of objects.

$$\mathrm{RN}(O) = f_\phi\Big(\sum_{i,j} g_\theta(o_i, o_j)\Big) \tag{2.4}$$

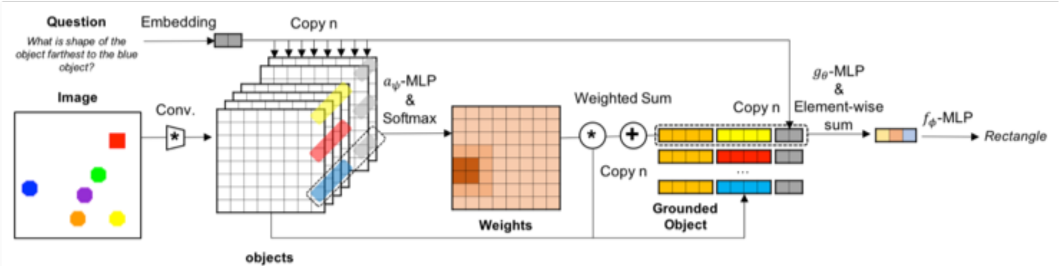


Figure 2.5: Sequential Attention Relational Network

Sequential Attention Relational Network(SARN)[2] tried to overcome the weakness of RN by sequential grounding objects. Instead of pairing all possible objects, SARN first finds a object that becomes the standard for the reasoning and pair it with the other objects. In this way, relations to be considered are remain linear to the number of objects.

$$\begin{aligned} \mathrm{SARN}(O) &= \sum g_\theta(o_R, o_i, q) \\ o_R &= \sum_i a_i * o_i \\ a_i &= a_\psi(o_i, q) \end{aligned} \tag{2.5}$$

Chain of Reasoning model(CoR)[23] expanded this idea to perform multi-step reasoning. CoR alternately updates the objects and their relations by sequentially choosing objects conditioned on the question and previously grounded objects. Although CoR kept the number of relation in linear level, the number of reasoning steps have to be explicitly determined. Furthermore, the interaction between an question vector and image vectors remained constrained since a single summarized vector of question is used.

2.1.3 Module based models

Another common approach to VQA is to formulate a executable program to perform on image features based on a question. Neural Module Network(NMN)[4] first tried to solve VQA by composing modules. NMN constructs a network architecture based on a given question. Primitive modules that can be composed into any configuration of questions are defined: attention, re-attention, combination, classification, and measurement. The key component of the modules is attention mechanism that allow

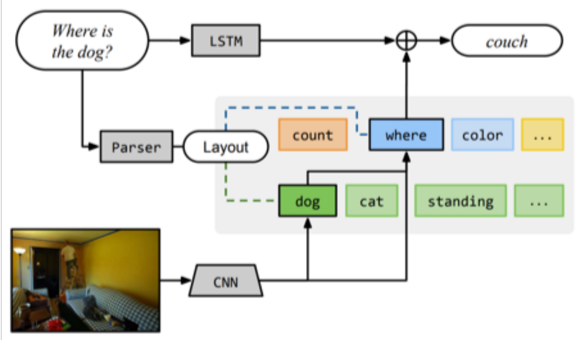


Figure 2.6: Neural Module Network[4]

model to focus on the parts of the given image.

Since there is no labels for the construction of modules, the network layout has to be learned from questions. In NMN, A question is parsed to form an universal dependency representation which can be map into a network layout. Embedded question is combined to the end of network to capture subtle differences. The composed model is trained end-to-end given network layouts.

Instead of using semantic parser to get the layout for the modules from question, End-to-end Neural Module Network(N2NMN)[10] uses encoder-decoder structure to get the layouts. RNN is used for question encoder and another RNN(with attention) is used to unroll layout policy. Since the layout policy is not fully differentiable, REINFORCE algorithm is used to approximate the gradient.

Since learning the layout from the scratch is challenging, additional knowledge(expert policy using parser) which is provided as initial supervision is shown to be effective. KL-divergence between layout policy and expert policy is added to loss function.

Inferring and Executing Program(IEP)[14] showed that this module network structure can be effective especially when ground truth layouts for the program

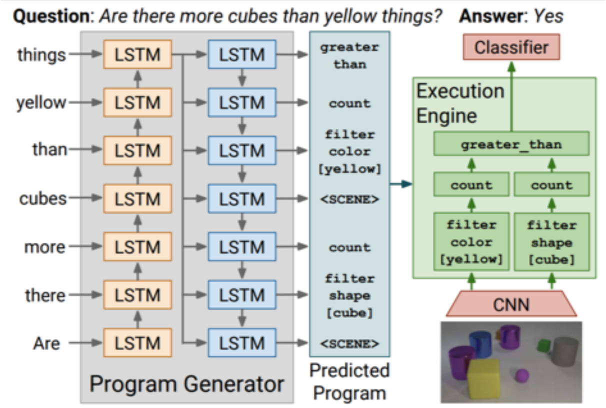


Figure 2.7: Inferring and Executing Program[14]

are provided. IEP achieved nearly perfect accuracy using the functional program information which are provided in CLEVR dataset.

These models are reputed for great interpretability, but their training process were noisy due to REINFORCE algorithm and their overall performance without the ground truth layout are yet behind the attention based models. Also, the needs for ground truth layouts are critical since there cannot be functional programs in real world except for synthetically created dataset such as CLEVR. Furthermore, modules that compose programs require a lot of human prior knowledge and efforts.

Several works tried to overcome the need of explicit annotation of program by creating implicit program. Feature-wise Linear Modulation(FiLM)[19] solve VQA by providing sophisticated condition on image encoder. FiLM generator takes questions as input and output betas and gammas for each ResBlocks. Each ResBlock contains FiLM after convolution and BN layer. By linearly transforming the output of convolution filter, FiLM conditionally choose certain filters. Several ResBlocks with FiLM stacks to form FiLM network. FiLM achieved state-of-the-art performance without

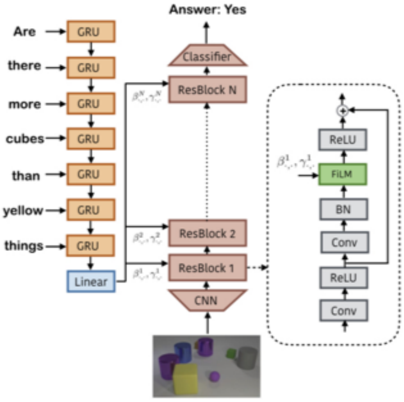


Figure 2.8: Feature-wise Linear Modulation[19]

the use of functional program information. However, huge question encoder with hidden size of 2048 is critical for FiLM which is responsible for huge number of parameters.

Recurrent Memory, Attention, and Composition network(MAC)[11] designed fully differentiable module that can store and retrieve information from memory for reasoning. MAC showed incredible performance on CLEVR dataset. However, MAC showed quite limited performance on GQA dataset which reflects real world.

Chapter 3

Multimodal Self-Attention Network

3.1 Model Architecture

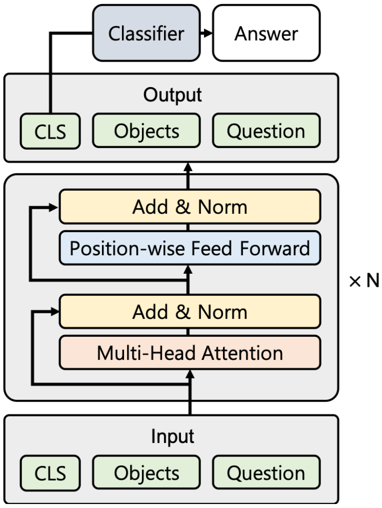


Figure 3.1: Overall Architecture of Multimodal Self-Attention Network

Most competitive visual reasoning models use a summarized question vector to impose attention on image, condition on relations or layout modules to execute. In our Multimodal Self-Attention Network(MUSAN), information from objects and question words interacts from scratch without encoding process of the question. MUSAN is visual reasoning model based on modified Transformer encoder by [22]. Since we only need to output the answer of the visual reasoning, we only used the encoder part of the transformer.

Self-attention mechansim is one of the most powerful approaches for encoding text since it first introduced by [22]. Instead of RNN which only encodes text linearly, self-attention encodes the information of words with all other words around it. This approach enables more active interaction between words including words in distance. Using self-attention mechanism, [22] showed incredible performance on translation. Recently, [6] broke the records of most of the natural language processing tasks with pretrained Transformer encoder.

3.2 Input Representation

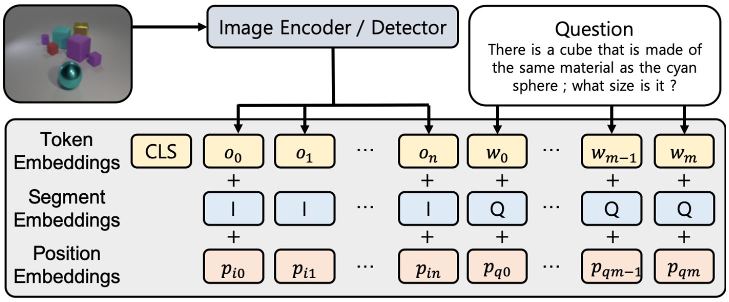


Figure 3.2: Input representation of MUSAN

In BERT[6] which also use Transformer encoder, the input representation consists of three parts: token embeddings, segment embeddings and position embeddings. These three kinds of embeddings are summed to be used as inputs. We've found that using summation of three embeddings as inputs is also effective for visual reasoning task.

Let a set of image features $\{\mathbf{i_1}, ..., \mathbf{i_n}\}$ and that of question indexes $\{q_1, ..., q_m\}$ with n represents the number of objects in image and m represents the length of question. In order to perform self-attention, the dimension of the two have to be the same. Extracted Image features are projected to the size of word embedding d_{model} to make object features $\{\mathbf{o_1}, ..., \mathbf{o_n}\}$ and question indexes are embedded into word vectors $\{\mathbf{w_1}, ..., \mathbf{w_m}\}$.

BERT used segment embedding to separate two sentences. In this model, segment embedding is used to separate object features and word vectors. Two learnable vectors $\mathbf{s_I}$ and $\mathbf{s_Q}$ are shared across the tokens.

In Transformer and BERT, sinusoidal position encoding is used to represent the position of tokens. However, the sinusoidal encoding would be redundant if positional information of images are provided. So the sinusoidal encoding $P_Q = \{\mathbf{p_{Q1}}, ..., \mathbf{p_{Qm}}\}$ is used only for words . The positional encodings of objects have to represent at least two dimensional position. 2 dimensional vector with x and y coordinate in spatial image features and 4 dimension vector with bounding box information for detector features are projected to d_{model} dimensional vector resulting $P_I = \{\mathbf{p_{I1}}, ..., \mathbf{p_{In}}\}$.

Three kinds of embeddings $C = \{\mathbf{cls}\}$, $O = \{\mathbf{o_1} + \mathbf{s_I} + \mathbf{p_{I1}}, ..., \mathbf{o_n} + \mathbf{s_I} + \mathbf{p_{In}}\}$, and $Q = \{\mathbf{w_1} + \mathbf{s_Q} + \mathbf{p_{Q1}}, ..., \mathbf{w_m} + \mathbf{s_Q} + \mathbf{p_{Qm}}\}$ are summed up to be used as input I.

3.3 Transformer Encoder

Transformer encoder[22] consists of several layers of encoder blocks. Each encoder blocks consists of a Multi-Head Attention part a position-wise fully connected feed-forward network. Each part have residual connection and a layer normalization at the end.

3.3.1 Multi-Head Attention layer

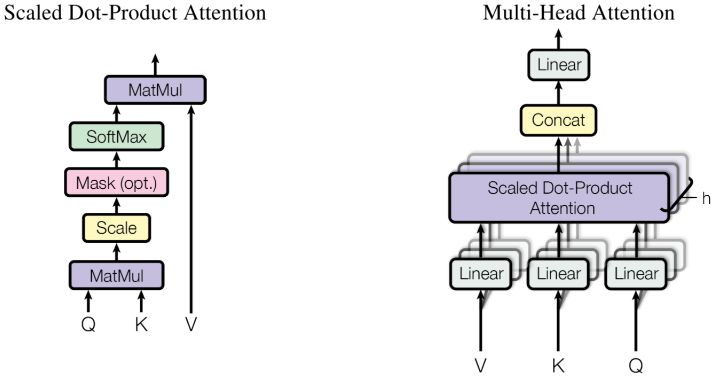


Figure 3.3: Multi-Head Attention[22]

In self-attention mechanism, every elements are represented by the mix of the other(contextual) elements. Every elements of the input are projected into three vectors: Query, key and value. Each elements are represented with the weighted sum of the value(V) of the other elements.

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{3.1}$$

The weights are decided by the inner product of the query vector(Q) of the element and the key vectors(K) of the other elements. The dimension of key and that of value is denoted as d_k and d_v respectively. This attention mechanism is called Scaled Dot-Product attention.

$$\begin{gathered}\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_h)W^O \\ \text{where head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)\end{gathered} \tag{3.2}$$

The projection matrices have following dimensions: $w_i^Q \in \mathbb{R}^{d_{model} \times d_k}$, $w_i^K \in$

$\mathbb{R}^{d_{model} \times d_k}$, $w_i^W \in \mathbb{R}^{d_{model} \times d_v}$, and $w_i^o \in \mathbb{R}^{hd_v \times d_{model}}$. Several scaled dot-product attention can be concatenated and projected into one vector. This grouping process is called Multi-head attention.

In MUSAN, we only used self-attention among input features stated above.

$$
\begin{aligned}
\text{MultiHead}(I) &= \text{Concat}(\text{head}_1, ..., \text{head}_h)W^O \\
\text{wherehead}_i &= \text{Attention}(IW_i^Q, IW_i^K, IW_i^V) \\
I &= [C, O, Q]
\end{aligned}
\tag{3.3}
$$

where [] indicate concatenation.

3.3.2 Position-wise Feed Forward layer

After multi-head attention part including residual path and layer norm, position-wise feed forword network performs 2 layers MLP on each of the elements. The parameter of this MLP is shared across the elements, but does not shared across the layers. The hidden size of position-wise feed forward networks is denoted by d_{ff}.

$$
\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \tag{3.4}
$$

3.3.3 Pooling layer

In MUSAN, only one vector is used to solve the classification problem instead of the whole sequence. Therefore, we need to pool the information to a vector from variable number of vectors. Several methods were tried to sum up the information after the last layer of the encoder: Max pooling, mean pooling, using the last layer of RNN encoder and using the vector of output vector at place of CLS token. The use of CLS token was first introduced in BERT[6]. BERT add additional CLS token to extract answer related information and used it for task specific classifier. After comparing 4 methods, we've found using the vector of CLS token the most effective in our architecture.

Chapter 4

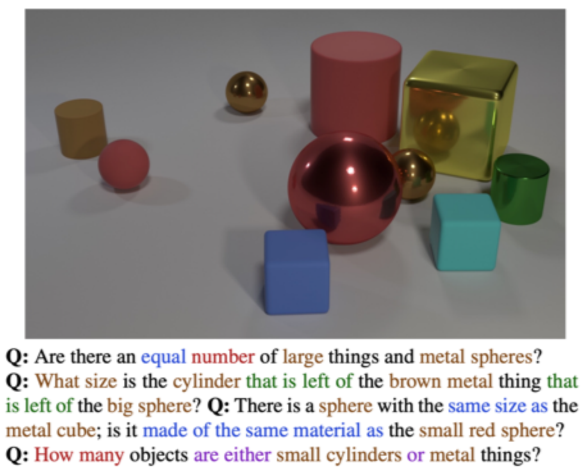
Experiments

As visual question answering task, visual reasoning task also needs manually created image, question and answer pairs in huge amount. Therefore, most of researches on visual reasoning are conducted with openly available dataset. Two major open visual reasoning datasets are CLEVR[13] and recently released GQA dataset[12].

Human annotated datasets are known to be expensive and very noisy especially when they include natural language annotation. As stated above, limited space of the language is critical in visual reasoning task. Therefore, there has been many attempts to programmatically create dataset. The two major datasets of visual reasoning also partially automatize the creation process of the dataset.

4.1 CLEVR

CLEVR dataset[13] is presented as a diagnostic dataset for visual reasoning task. Image, question, and answer pairs are created with program with the least human intervention. Figure 4.1 are example questions of CLEVR dataset.



Q: Are there an equal number of large things and metal spheres?
Q: What size is the cylinder that is left of the brown metal thing that is left of the big sphere? **Q:** There is a sphere with the same size as the metal cube; is it made of the same material as the small red sphere?
Q: How many objects are either small cylinders or metal things?

Figure 4.1: A example of CLEVR dataset[13]

4.1.1 Dataset

Images of CLEVR contains three object shapes (cube, sphere, and cylinder) that come in two sizes (small and large), two materials (shiny "metal" and matte "rubber"), and eight colors. Objects are spatially related via four relationships: "left", "right", "behind", and "in front". A scene graph consists of a group of these annotated objects as nodes and their relations as edges. Every scene contains between three and ten objects with random shapes, sizes, materials, colors, and positions. Randomly created scene graphs are rendered with computer program (Blender).

Every question in CLEVR dataset is related to a function program that can be executed on a scene graph. Functional program forms various question families with thier own natural language templates. This functional program can create diverse and clean questions. 70,000 / 15,000 / 15,000 images and 699,989 / 149,991 / 149,988 questions are provided as train, validation, test dataset. The answer distribution is carefully balanced with program so there is no problem of language prior.

4.1.2 Setting

Although visual reasoning task need complex reasoning of image and text, the text structure of the question is rather simple and structured compared to other natural language tasks. Therefore, we halved the most of the hyperparameters of Transformer except for the number of layers and dropout rate. d_{model}(Word embedding size) is 256 and d_{ff}(hidden dimension of feedforward network) is 1024. h(number of head of multihead attention) is set to 4 and d_k(dimension of query and key) and d_v(dimension of value) are set to 32. N(the number of encoder layers) is kept to 6 and also the dropout rate is kept to 0.1. Despite of these settings of the best model, MUSAN is found to be very robust to minor changes in hyperparameters.

A simple CNN structure is used for spatial image encoder. Channel size is uniformly set to 256 across all layers and height and width are halved by stride 2 in every layer. We stacked 4 layers regardless of input size. The size of image is set to 224 × 224 as convention, making the number of objects 14 × 14. For simplicity, we also tested on resized image size of 128× 128 with object size 8 × 8. The batch size is 256, learning rate is 2.5e-4 and early stopping is used due to computation constraints. Learning rate was scheduled to halved every 10 epochs since the last decrease in loss until 2.5e-6.

4.1.3 Result

Table 4.1 summarized the performance of MUSAN compared to other benchmarks. * indicates the use of functional program information, † indicates data augmentation, and ‡ indicates the use of pretrained image encoder. Most of attention based models introduced in Chapter 2 were introduced before CLEVR or did not report

Table 4.1: Validation results of CLEVR dataset.

Model	Count	Compare Numbers	Exist	Query Attribute	Compare Attribute	Overall
Human	86.7	96.6	86.5	95.0	96.0	92.6
Q-type Baseline	34.6	50.2	51.0	36.0	51.3	41.8
LSTM	41.7	61.1	69.8	36.8	51.8	46.8
CNN+LSTM	43.7	65.2	67.1	49.3	53.0	52.3
CNN+LSTM+SA+MLP	59.7	77.9	75.1	80.9	70.8	73.2
NMN*	52.5	72.7	79.3	79.0	78.0	72.1
N2NMN*	68.5	85.7	84.9	90.0	88.7	83.7
PG+EE*	92.7	97.1	98.7	98.1	98.9	96.9
CNN+LSTM+RN†	90.1	97.8	93.6	97.9	97.1	95.5
CNN+GRU+FiLM	94.3	99.3	93.4	99.3	99.3	97.6
CNN+GRU+FiLM‡	94.3	99.1	96.8	99.1	99.1	97.7
DDRprog*	96.5	98.4	98.8	99.1	99.0	98.3
TbD*‡	96.8	99.1	98.9	99.4	99.1	98.7
MAC‡	97.1	**99.5**	99.1	99.5	99.5	98.9
MUSAN(128)	97.2	98.3	99.2	99.5	99.3	98.7
MUSAN	**98.2**	99.0	**99.6**	**99.7**	**99.7**	**99.3**

the performance. According to our implementation, most of attention based models designed for VQA dataset did not perform well on CLEVR at least with the same models. Since Relational network first beated the performance of human with some data augmentation, many module based models were tested on CLEVR due to availability of functional program. Although recent module based models were competitive, they need functional program information which is unnatural in most of the settings. While FiLM and MAC are two most competitive models on CLEVR dataset, they used the pretrained image encoder trained from the other dataset.

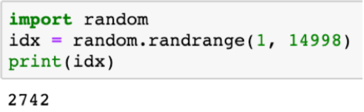
Our model, MUSAN achieved competitive result on CLEVR dataset compared to other model with additional information. Compared to the same settings, MUSAN achieved state-of-the-art result even with smaller image size(128).

4.1.4 Analysis

Interpreting the deep learning model is very important to understand how the model works. The most popular method to visualize the VQA model is to visualize the attentions. However, self-attention module is known to very difficult to interpret due to huge number of attentions. Given n inputs, $n \times n$ number of attentions with h number of attention head for every layer N are produce which makes visualization especially difficult. Fortunately, MUSAN uses CLS token to solve the task so that it is reasonable to assume that CLS token hold critical information. So we decided to visualize the attention of CLS token across layers.

Assuming each spatial feature have locality, attentions on spatial features can be visualized as a part of images. We shattered the value of features in Gaussian distribution with sigma 4 and set the transparency to 0.6 for clear visualization. Also, we used 128 × 128 model for clear visualization. In figures, L represent the index of layer and H represent the index of attention head.

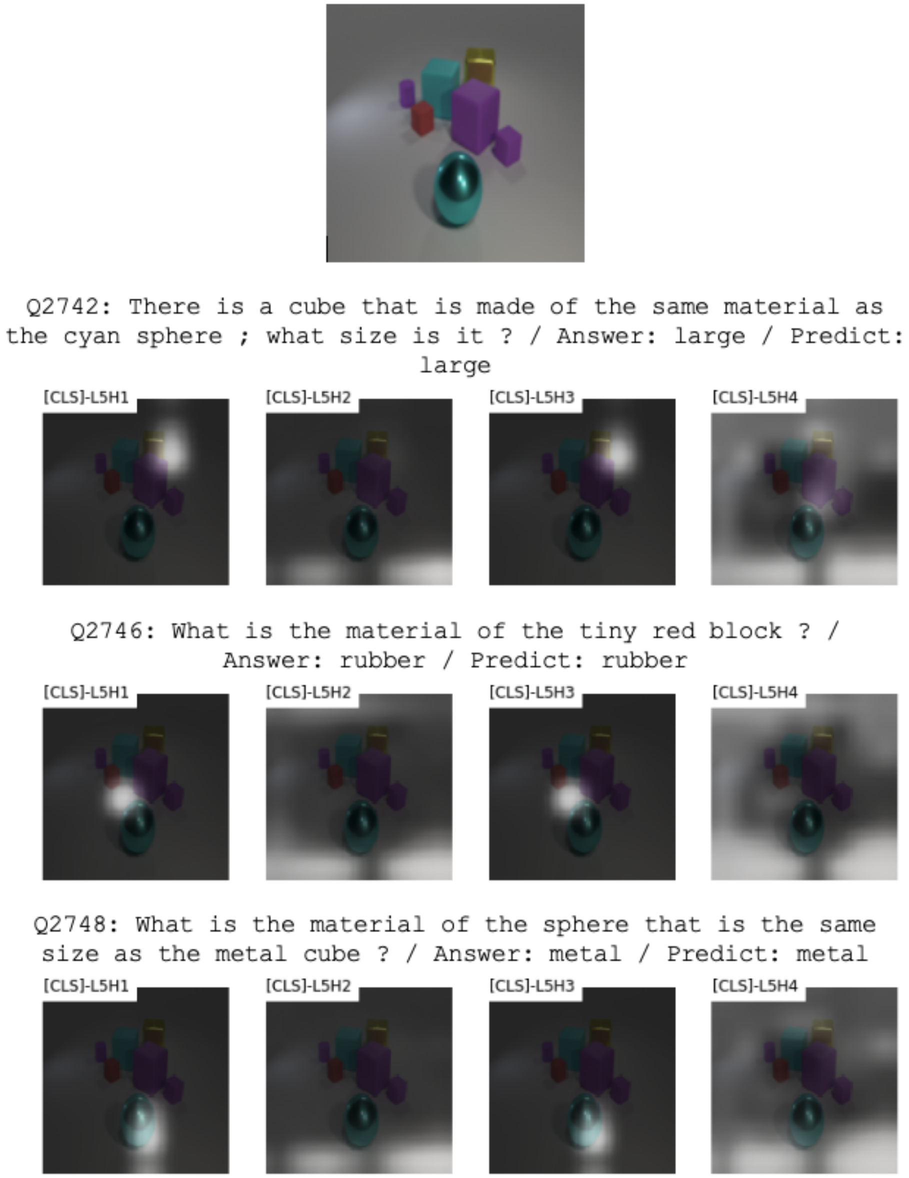
We note that the index of the question for this visualization is chosen at random in validation set as seen in Figure 4.2. These visualizations are not cherry-picked. We first choose index(2742) in random and compared with the other questions(2740-2749) with the same images. Since attentions are applied on both images and texts, it is accurate to visualize the both but visualizations are focused on one for simplicity. Full attentions of the questions can be found in appendix.



```
import random
idx = random.randrange(1, 14998)
print(idx)
```

```
2742
```

Figure 4.2: Random selection of question for visualization



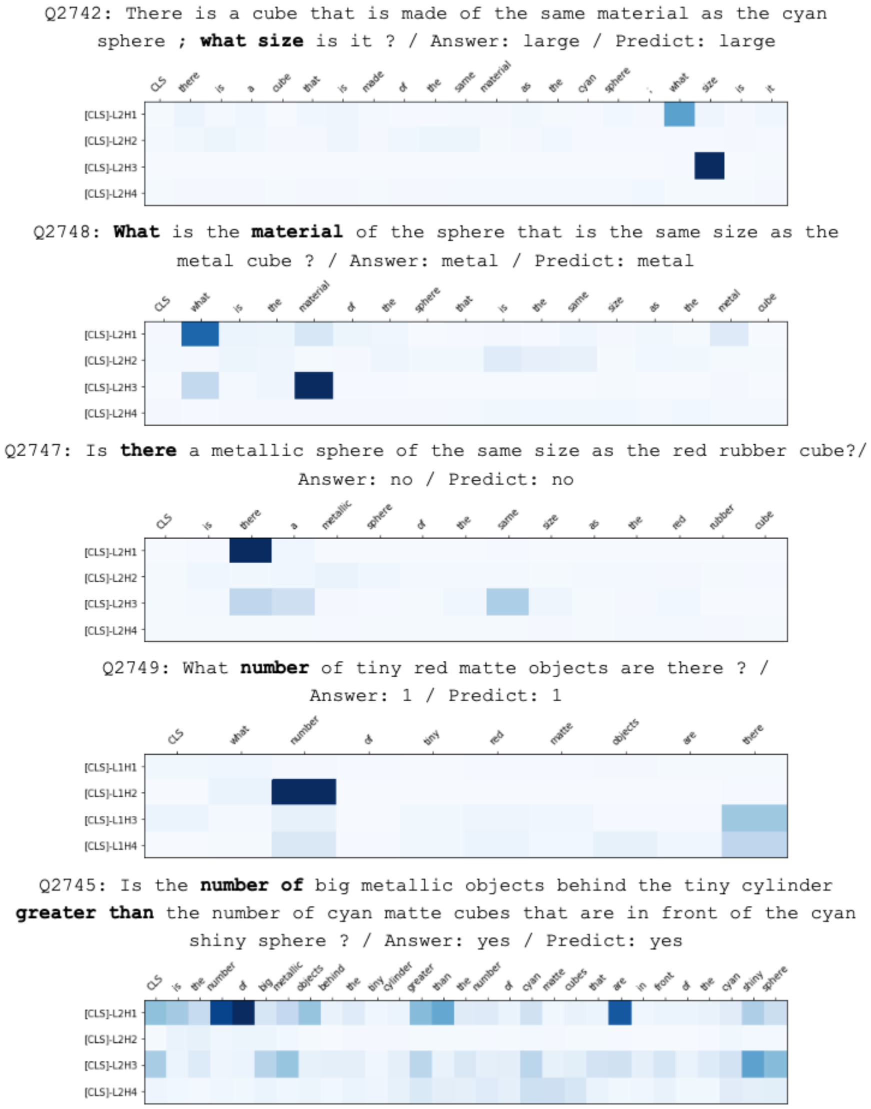
Q2742: There is a cube that is made of the same material as the cyan sphere ; what size is it ? / Answer: large / Predict: large

Q2746: What is the material of the tiny red block ? / Answer: rubber / Predict: rubber

Q2748: What is the material of the sphere that is the same size as the metal cube ? / Answer: metal / Predict: metal

Figure 4.3: Attention on images at layer 5

Figure 4.3 clearly shows the last part of the model focus on the object related to the answer of the reasoning question. Different questions for the same image resulted in attention to the other objects related to the answer of each question.


Q2742: There is a cube that is made of the same material as the cyan sphere ; **what size** is it ? / Answer: large / Predict: large

Q2748: **What** is the **material** of the sphere that is the same size as the metal cube ? / Answer: metal / Predict: metal

Q2747: Is **there** a metallic sphere of the same size as the red rubber cube?/ Answer: no / Predict: no

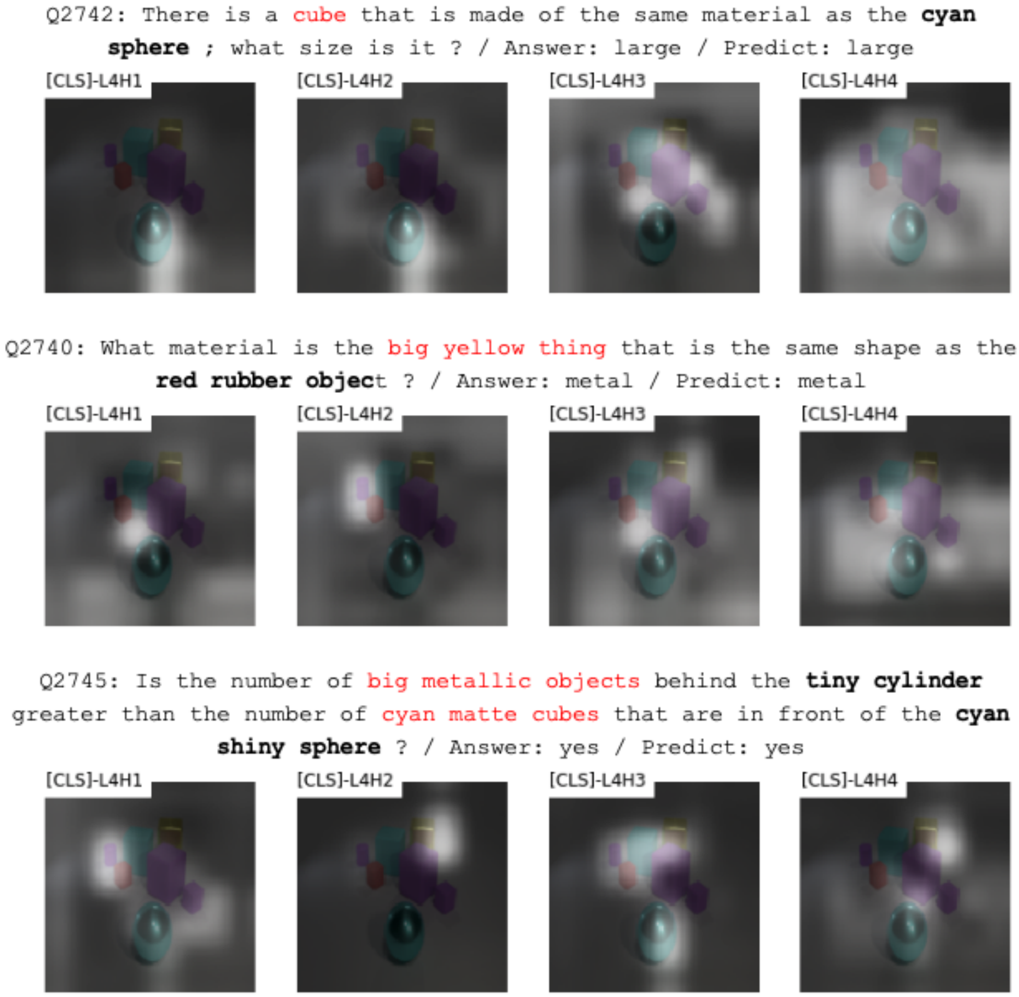
Q2749: What **number** of tiny red matte objects are there ? / Answer: 1 / Predict: 1

Q2745: Is the **number of** big metallic objects behind the tiny cylinder **greater than** the number of cyan matte cubes that are in front of the cyan shiny sphere ? / Answer: yes / Predict: yes

CLS there is a cube that is made of the same material as the cyan sphere ; what size is it
[CLS]-L2H1
[CLS]-L2H2
[CLS]-L2H3
[CLS]-L2H4
CLS what is the material of the sphere that is the same size as the metal cube
[CLS]-L2H1
[CLS]-L2H2
[CLS]-L2H3
[CLS]-L2H4
CLS is there a metallic sphere of the same size as the red rubber cube
[CLS]-L2H1
[CLS]-L2H2
[CLS]-L2H3
[CLS]-L2H4
CLS what number of tiny red matte objects are there
[CLS]-L1H1
[CLS]-L1H2
[CLS]-L1H3
[CLS]-L1H4
CLS is the number of big metallic objects behind the tiny cylinder greater than the number of cyan matte cubes that are in front of the cyan shiny sphere
[CLS]-L2H1
[CLS]-L2H2
[CLS]-L2H3
[CLS]-L2H4

Figure 4.4: Attention on questions at layer 2

Figure 4.4 shows that earlier part of the model seem to focus on the keywords of the questions. Instead of focusing on meaningless articles, model seems to focus on the keywords of the question which are informative to understand the structure of the question.



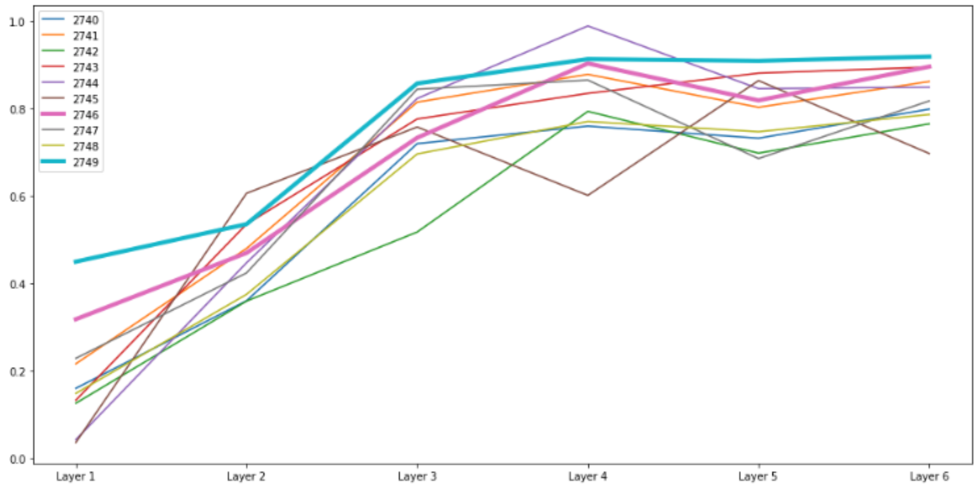
Q2742: There is a cube that is made of the same material as the **cyan sphere** ; what size is it ? / Answer: large / Predict: large

Q2740: What material is the big yellow thing that is the same shape as the **red rubber objec**t ? / Answer: metal / Predict: metal

Q2745: Is the number of big metallic objects behind the **tiny cylinder** greater than the number of cyan matte cubes that are in front of the **cyan shiny sphere** ? / Answer: yes / Predict: yes

Figure 4.5: Attention on images at layer 4

How the model reason on the image and the question is difficult to interpret since we did not provide any supervision for reasoning. Clues for its reasoning would be scattered across numerous attentions. Therefore, the attention of the middle part of the models are quite noisy. However, Figure 4.5 shows that middle layers of some questions focus on objects which are not the object of answer.



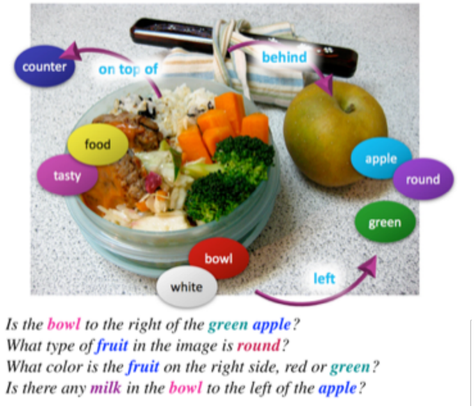
```
Q2740: What material is the big yellow thing that is the same shape as the red rubber object ?
Q2741: What number of cylinders are the same color as the big sphere ?
Q2742: There is a cube that is made of the same material as the cyan sphere ; what size is it ?
Q2743: What number of purple objects are either big cubes or big metallic objects ?
Q2744: There is a thing in front of the small purple block ; what number of matte cubes are left of it ?
Q2745: Is the number of big metallic objects behind the tiny cylinder greater than the number of cyan matte cubes tha
t are in front of the cyan shiny sphere ?
Q2746: What is the material of the tiny red block ?
Q2747: Is there a metallic sphere of the same size as the red rubber cube ?
Q2748: What is the material of the sphere that is the same size as the metal cube ?
Q2749: What number of tiny red matte objects are there ?
```

Figure 4.6: Portion of attention on image across layers

We visualized the portion of attention on images across layers by questions in Figure 4.6. Interestingly, MUSAN tends to focus on understanding the question text at earlier stage, and focus more on image in later stage. Two highlighted lines are two of the questions with relatively simple question structure. The two lines tends to focus more on image than text since the structure of the questions are rather simple.

4.2 GQA Dataset

4.2.1 Dataset



Is the bowl to the right of the green apple?
What type of fruit in the image is round?
What color is the fruit on the right side, red or green?
Is there any milk in the bowl to the left of the apple?

Figure 4.7: Examples of GQA dataset

Though CLEVR provided high-quality, and scalable data for visual reasoning task, it is criticized for its simplicity and unrealistic settings. GQA[12] is clean and scalable dataset which adopted the methodology of CLEVR to real world image dataset.

GQA made scene graphs of real world images by starting with Visual Genome[18] scene graphs dataset. GQA augmented the VG scene graphs with more detailed properties using object detector and additional human annotators. Then, the scene graphs are pruned and the language of the scene graphs are cleaned to limit the language space.

Next, question engine built questions with 274 structural patterns by traversing the scene graphs. Each question pattern is related with a functional representation as in CLEVR dataset. However, question building in GQA requires more supervision of human since it has to reflect real world. With semantic program, the questions are balanced in two granularity levels to minimize the language bias.

Through these process, GQA dataset provided 113,018 images with 22,669,678 questions in total with vocab size of 3097 for questions, and 1878 for answers. They not only provides raw images, they also provide spatial features and detection features inferred with pretrained classifier and object detector. For each of images, they provided detailed scene graph with objects, their property and location, and relations among them. Questions are provided with their functional program. Also, answers are provided in short, and long version.

GQA suggested new metrics other than accuracy such as consistency, validity, plausibility and distribution. Consistency measures responses consistency across different questions using entailment relation among questions. Validity metric checks whether a given answer is in the scope of the question. Plausibility score goes a step further, measuring whether the answer is reasonable, or makes sense, given the question. Distribution measures the overall match between the true answer distribution and the model predicted distribution.

4.2.2 Setting

We used object detector features for computational efficiency. However, we didn't use scene graph information of images nor the functional program information of questions since they would be only provided in special settings. We wanted the model to learn reasoning from data.

Basic model structure for GQA is almost the same as that for CLEVR. We tried various configuration of model for competition. The single best model for GQA was $d_{model} = 512$ and $N = 9$ while the rest of the structure remain the same. We used

BERT Adam optimizer to schedule the learning rate; warm up for 0.01% and linearly decrease for the rest. Exponentially moving average of weights with $\rho = 0.999$ is used. The last classifier of MUSAN choose one of the answers but simple auxiliary task to output the long version of answer with simple lstm seemed to have regularizing effect.

4.2.3 Result

Table 4.2: Dev results of GQA dataset

Model	Binary	Open	Consistency	Plausibility	Validity	Distribution	Accuracy
MUSAN(Ours)	78.79	41.88	96.95	86.17	91.92	2.02	59.15
MUSAN-Ensemble(Ours)	79.80	43.54	97.19	86.69	93.61	2.16	60.51

None of the other models reported the score yet for Dev.

Table 4.3: Test results of GQA dataset

Model	Binary	Open	Consistency	Plausibility	Validity	Distribution	Accuracy
Human Performance	91.20	87.40	98.40	97.20	98.90	0.00	89.30
LSTM-CNN	63.26	31.80	74.57	84.25	96.02	7.46	46.55
BottomUp	66.64	34.83	78.71	84.57	96.18	5.98	49.74
MAC	71.23	38.91	81.59	84.48	96.16	5.34	54.06
LCGN	73.77	42.33	84.68	84.81	**96.48**	**4.70**	57.07
BAN	76.00	40.41	91.70	85.58	96.16	10.52	57.10
MUSAN(Ours)	77.83	41.58	96.28	85.50	92.00	9.33	58.57
MUSAN-Ensemble(Ours)	**79.09**	**43.02**	**96.41**	**85.92**	93.72	10.01	**59.93**

MUSAN achieved 8th rank on 2019 CVPR GQA challenge[1]. Although many models outperformed MUSAN on the leaderboard, we only stated the performance of models with publications on Table 4.3. This is because it would be fair to compare the performance of models trained with the same use of information. Since GQA was released recently, there are not many publically reported performance.

[1] https://evalai.cloudcv.org/web/challenges/challenge-page/225/leaderboard/733

Chapter 5

Conclusion

In this work, we presented the MUSAN, multimodal self-attention network for visual reasoning. MUSAN uses the transformer encoder to impose self-attention on the objects of the image and the words of the question. Inspired by BERT, MUSAN used three kinds of embedding to be summed as inputs: token, segment and position embeddings. With its simple structure, the model effectively learns to reason from raw images and words and shows robustness to changes in hyperparameters. Visualizations show that the model learns to reason with its own logic. The model achieved state-of-the-art results on the CLEVR task for visual reasoning and reported 8th rank on 2019 GQA challenge without scene graph information or functional program information.

Future works can be incorporating these additional information to boost up the performance on visual reasoning task. Our model is not only a good visual reasoning model, but also a proof that self-attention can be effective in multimodal tasks other than image and text. We believe that MUSAN will provide good insights on dealing cross-domain or cross-type data interaction.

Bibliography

[1] A. Agrawal, J. Lu, S. Antol, M. Mitchell, C. L. Zitnick, D. Parikh, and D. Batra, *Vqa: Visual question answering*, International Journal of Computer Vision, 123 (2017), pp. 4–31.

[2] J. An, S. Lyu, and S. Cho, *Sarn: Relational reasoning through sequential attention*, arXiv preprint arXiv:1811.00246, (2018).

[3] 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.

[4] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein, *Neural module networks*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 39–48.

[5] 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.

[6] 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).

[7] 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).

[8] 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.

[9] 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.

[10] 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.

[11] D. A. Hudson and C. D. Manning, *Compositional attention networks for machine reasoning*, arXiv preprint arXiv:1803.03067, (2018).

[12] ——, *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.

[13] 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.

[14] 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.

[15] J.-H. Kim, J. Jun, and B.-T. Zhang, *Bilinear attention networks*, in Advances in Neural Information Processing Systems, 2018, pp. 1564–1574.

[16] J.-H. Kim, S.-W. Lee, D. Kwak, M.-O. Heo, J. Kim, J.-W. Ha, and B.-T. Zhang, *Multimodal residual learning for visual qa*, in Advances in neural information processing systems, 2016, pp. 361–369.

[17] 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).

[18] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, et al., *Visual genome: Connecting language and vision using crowdsourced dense image annotations*, International Journal of Computer Vision, 123 (2017), pp. 32–73.

[19] 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.

[20] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap, *A simple neural network module for relational reasoning*, in Advances in neural information processing systems, 2017, pp. 4967–4976.

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

[22] 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.

[23] C. Wu, J. Liu, X. Wang, and X. Dong, *Chain of reasoning for visual question answering*, in Advances in Neural Information Processing Systems, 2018, pp. 275–285.

[24] 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.

국문초록

시각적 추론은 이미지와 질문의 정교한 정보 제어가 필요하기 때문에 시각적 질문 응답보다 어렵다. 한 소스에서 추출 된 정보는 다른 소스에서 정보를 추출하는 데 사용되며 이 프로세스는 교대로 발생한다. 복잡한 자연어 문제를 다단계적 추리로 풀려면 인간조차도 이미지와 질문을 여러 번 흘끗 볼 필요가 있기 때문에 이것은 당연한 것이다. 초기 단계에서 얻은 정보를 처리하고 나중에 답을 얻기 위해 사용할 필요가 있다. 이러한 차이 때문에, 이 두 과제에 대한 결과는 밀접하게 연관되지 않는 경향이 있다.

본 논문에서는 시각적 추리 과제를 해결하기 위해 MUSAN(Multimodal Self-attention Network)을 제안한다. 본 모델은 [22]가 제안한 트렌스포머 인코더를 사용하여 세부적인 수준에서 이미지와 질문 간의 긴밀한 상호작용을 촉진한다. MUSAN은 사전 지식이나 사전 훈련된 피쳐 추출기 없이 원시 픽셀에서 CLEVR 데이터셋의 최고 성능을 달성했다. 또 2019년 GQA 챌린지에서 문제 생성 함수 정보나 그래프 정보 없이 8위를 기록했다. MUSAN의 어탠션 시각화는 MUSAN이 자신의 논리로 단계적 추론을 수행한다는 것을 보여준다.

주요어: 시각적 질의응답, 시각적 추론, 셀프 어텐션

학번: 2017-25671