VISUAL QUESTION ANSWERING UNDER DIFFERENT SETTINGS

Jiateng Liu
School of Computer Science and Technology
Zhejiang University
3190101988@zju.edu.com

Enze Wu
School of Computer Science and Technology
Zhejiang University
3190102297@zju.edu.com

Li Sun
School of Computer Science and Technology
Zhejiang University
3190104757@zju.edu.com

Congshan Yu
School of Computer Science and Technology
Zhejiang University
3190103165@zju.edu.com

June 28, 2022

ABSTRACT

Visual Quesition answering is a classic and challenging problem in the multimodal field. As students trying to give a solution over this task with limited data and computing resources, admittedly, it is relatively hard for us to obtain state of the art results. However,it doesn't mean that we can't make any more contribution. In this paper, we propose different settings include a basic classification task setting, a sequence to sequence answer generation setting and further extend to a contrastive learning answer retrieval setting. Motivated by existing models and training objectives, we give algorithm and models to solve the VQA problem under all these circumstances and obtain relatively good result. We also give comprehensive step illustration and code analysis to our solutions.

1 Introduction

The research on VQA is developed on the basis of two studies on Text Question Answering and Image Captioning. The hallmark of the VQA question raised is the release of the dataset of the same name in 2015. Generally, VQA describes tasks that given an image and a natural language question related to the image, the computer provides a correct answer.

VQA is a classic and challenging problem in the multimodal field, which is a academic hotspot in recent years. Considering its high challenge, VQA is now generally considered as an AI-complete task which can be used as an alternative to visual Turing test. For example, Google reCAPCHA, the human-machine verification system of Google web pages, asks people choose matching images based on natural language questions, which can also be treated as a derivative problem of VQA.

Running environments and Resource limited situation In our experiment settings, we have mindspore as the main deep learning framework to train. We used both the platform and environment provided by HuaWei Cloud and our own

GPU configurations. On the HuaWei Cloud platform, we have mindspore-1.3.0 with Ascend as the main processor. On our own Server, we used one RTX 3090 GPU with Cuda10.1 and Mindspore-gpu 1.7.0. We did not use any other main-stream deep-learning framework like pytorch, tensorflow,etc. Due to the limited computing resource and limited training data, it's unlikely for us to reach SOTA results. As a result, we turn to finding a way which can help us achieve relatively better result under a resource limited situation. Except from using a bert pre-trained model to extract text features, we train the series of models from scratch.

Our general motivation Despite the sparsity of computing resources and limited training data. We still want to learn from Image-text Pre-training models and achieve better performance on the task on VQA. Under this circumstance, we want to fully employ the advantage of pre-training models by using similar model architectures like a Dual-stream model or a Single-stream model and by employing pre-trained well-defined encoders.For example, we can directly apply popular model architectures like Bert, ResNet, LSTM .etc. We also want to Design different training objectives under different problem settings. Like using Imag-text matching/aligning to design and using contrastive learning concepts to design. We also want to employing Seq2seq model loss to design under a situation without candidate answers provided. Finally, We aims at solving VQA problems within limited data and limited computing resources.We do experiments under different settings to test whether our ideas and designs work.

2 Related work

2.1 Non-pretrain approaches

In the early days after VQA was proposed, there were mainly the following four categories of practices:

Joint embedding approaches Joint embedding is a classic idea for dealing with multimodal problems. Its specific approach is to jointly encode the multimodal processing results, that is, the image and the question in this problem. The baselines of modern image processing and text processing are convolutional neural networks and recurrent neural networks, respectively. Thus, the whole process of this approach can be described as that the image and the question are encoded by CNN and RNN firstly to obtain their respective features, and then jointly input to another encoder to obtain joint embedding, and finally output the answer through the decoder.

Attention mechanisms The attention mechanism originated from the problem of machine translation. The purpose is to allow the model to dynamically adjust the weight to each part of the input item, thereby improving the focus performance of the model apparently. Using the Attention mechanism in VQA aims at making the model determine the preference of local search at inference time based on images and questions, rather than directly doing global search. Therefore, the model seems to be able to capture key image parts more efficiently. In our work, we also resort to the attention mechanism to obtain fine-grained features of Images and text.

Compositional Models The idea of Compositional Models stems from building different models for different types of problems. The core step of Compositional Models is to design a modularized model. Its most important feature is to dynamically assemble modules according to the type of question to generate answers. A sample approach is to divide keywords according to the question, and then assign different attention to each part to get the answer phase by phase.

Models using external knowledge base The last category of methods is used to solve some problems that require prior knowledge. These question is hard to answer even for people who lack common sense. The main difference between this approach and other models is that images and texts need to be classified with multiple labels in advance. After searching in the knowledge base with these labels, the labels and search information are uniformly encoded and then input to the RNN.

2.2 Vision language pre-training models

With the overwhelming results brought by pretrained vision-language models, the development of VQA has come to a new era.

Encoder choices Encoders are widely used to extract features from Images or texts, they are widely used in the vision language pre-training models and achieve effective results. Some encoders which are widely used is introduced here and we select some of them to directly apply to our model in Section3. Firstly, on the Image side, we have convolutional neural network to extract Image grid features, object-detector based neural network to extract Image region features and Vision Transformers to extract Image patch features. We do not give detailed intorduction to each of the encoders here.

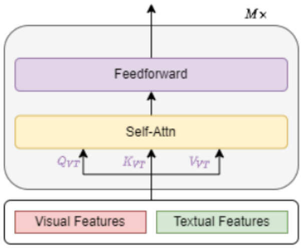


Figure 1: Example of a Single-stream model

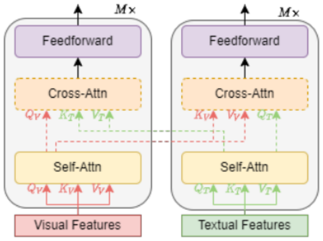


Figure 2: Example of a Dual-stream model

Since object detector based network is usually larger and harder to train, we did not resort to this method despite its powerful locating abilities. But both CNN and ViT models are worth a try.On the text side, we have Byte Pair Encoding, the popular Bert model and a few sequence based models like Long Short Term Memory. Generally,we can use either embedding or transformer to do feature extraction for texts.We mainly want to adopt models which are more effective on the text side.

Model architecture design Both single-stream models and Dual-stream models are popular in the designation of pre-train model architectures. Both model architectures can be found in Figure 1 and Figure 2.

Training objective design Pre-training objectives are the keys to learning a well universal representation. Different objectives represents a specific angel that the researchers are looking into the problem. In this part, we summarize the most popular training objectives and their purposes. Sometimes these objectives can be slightly modified and combined to gain a better performance on a specific task. Since our work is mainly not the pre-training task, we only introduce the useful and inspiring training objectives in this section

2.2.1 Vision Language Matching

Vision Language Matching can be one of the most frequently used training objectives.In this direct Image-text matching context, we concatenate the representations from both sides and feed them to a FC layer. We use a sigmoid function to project this result to 0 and 1 to judge whether the image and text match each other. In the training process, the negative samples need to be constructed from other training samples by randomly choosing mismatched pairs.

2.2.2 Vision Language Contrastive Learning

Vision Language Contrastive Learning is a substitution method for simple vision language matching. In this case, we no longer need to use random sampling to construct mismatched pairs. Instead, we use the images in a batch to construct positive and negative samples. We use the similarity between image and text vectors as training objectives, trying to maximize the difference between negative pairs and minimize the difference between positive pairs.This is firstly introduced in CLIP and the following figure (see Figure 3) shows a typical usage of this pre-training objective.

Pre-training models conclusion We list some of the current SOTA works and their generic designings in the following figure (See Figure 4 for details) , which is extracted from a survey published in February 2022. This means the approaches we learn from are really up-to-date, and it's more likely that our model will reach corresponding good results.

3 Models

3.1 Problem settings

Previous works often provide machine models with answer candidates and regard VQA as a classification task.For example, some piece of work select the most frequent 350 types from the answer set, then requires the model to give a prediction over these types. However, some right answers may not even exist in this given answer candidate set, thus leading the model to make a wrong prediction. In fact, VQA can either be regarded as a multi-answer classification problem or as an answer generation problem. Sometimes can even be extended to a retrieval task. We propose the problem settings in a mathematical form as below:

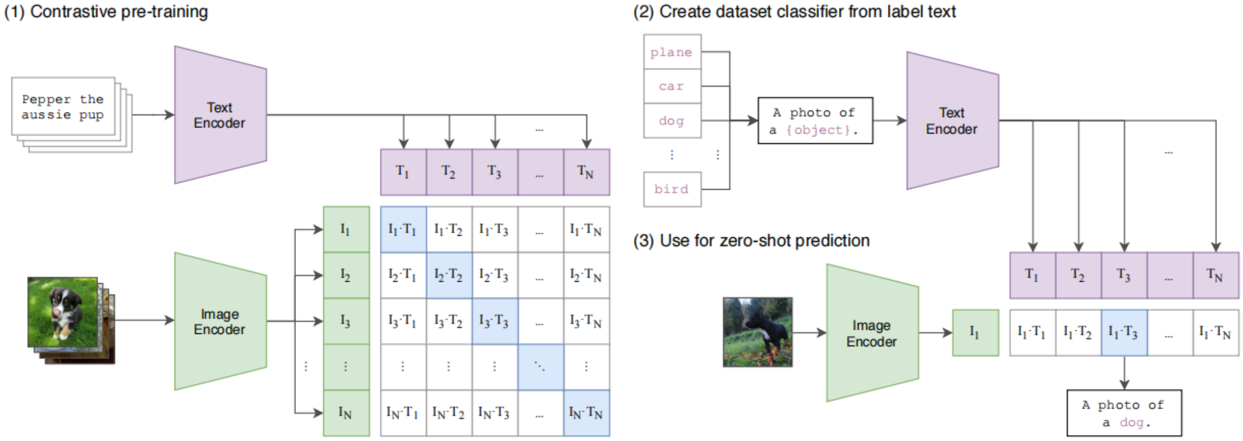
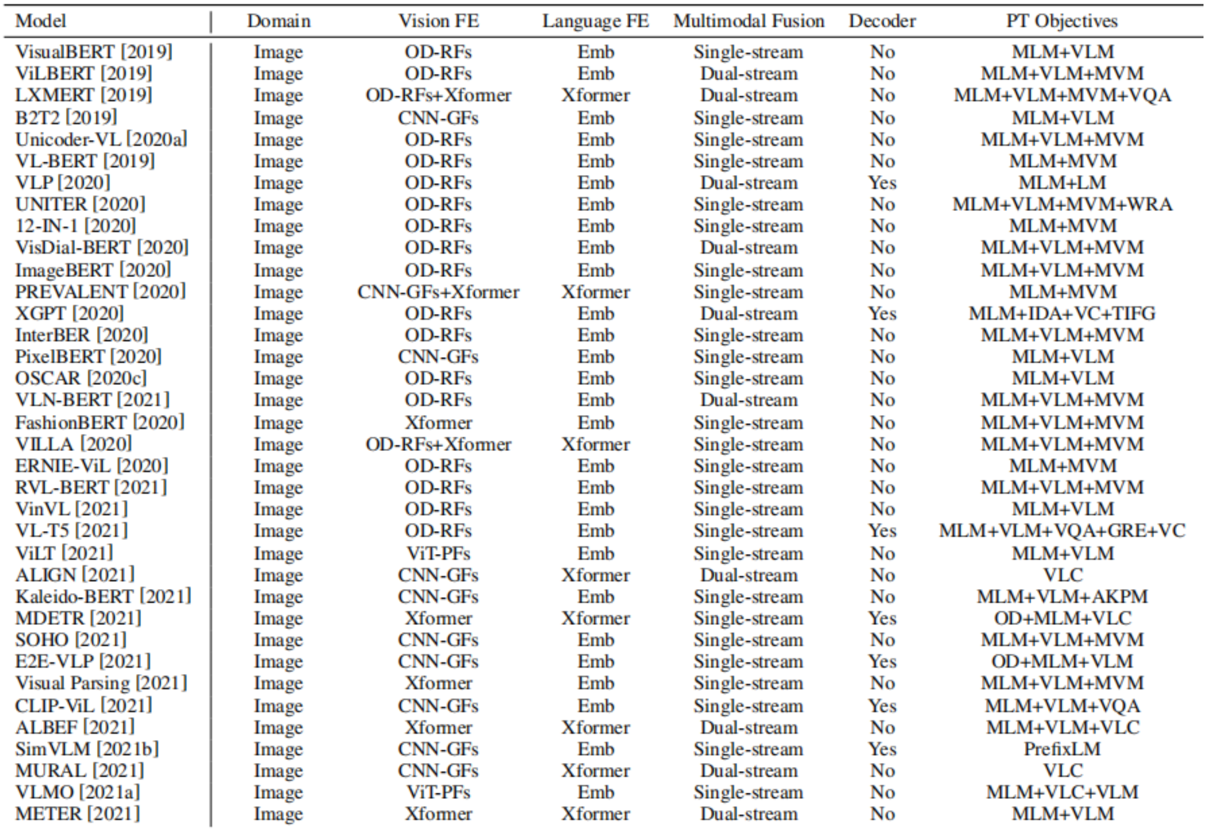

(1) Contrastive pre-training
Pepper the aussie pup
Text Encoder
Image Encoder
T1 T2 T3 ... TN
I1 I2 I3 ⋮ IN
I1·T1 I1·T2 I1·T3 ... I1·TN
I2·T1 I2·T2 I2·T3 ... I2·TN
I3·T1 I3·T2 I3·T3 ... I3·TN
IN·T1 IN·T2 IN·T3 ... IN·TN
(2) Create dataset classifier from label text
plane
car
dog
bird
A photo of a {object}.
Text Encoder
(3) Use for zero-shot prediction
Image Encoder
I1
T1 T2 T3 ... TN
I1·T1 I1·T2 I1·T3 ... I1·TN
A photo of a dog.

Figure 3: The Contrastive Learning method raised by CLIP



Model	Domain	Vision FE	Language FE	Multimodal Fusion	Decoder	PT Objectives
VisualBERT [2019]	Image	OD-RFs	Emb	Single-stream	No	MLM+VLM
ViLBERT [2019]	Image	OD-RFs	Emb	Dual-stream	No	MLM+VLM+MVM
LXMERT [2019]	Image	OD-RFs+Xformer	Xformer	Dual-stream	No	MLM+VLM+MVM+VQA
B2T2 [2019]	Image	CNN-GFs	Emb	Single-stream	No	MLM+VLM
Unicoder-VL [2020a]	Image	OD-RFs	Emb	Single-stream	No	MLM+VLM+MVM
VL-BERT [2019]	Image	OD-RFs	Emb	Single-stream	No	MLM+MVM
VLP [2020]	Image	OD-RFs	Emb	Dual-stream	Yes	MLM+LM
UNITER [2020]	Image	OD-RFs	Emb	Single-stream	No	MLM+VLM+MVM+WRA
12-IN-1 [2020]	Image	OD-RFs	Emb	Single-stream	No	MLM+MVM
VisDial-BERT [2020]	Image	OD-RFs	Emb	Dual-stream	No	MLM+VLM+MVM
ImageBERT [2020]	Image	OD-RFs	Emb	Single-stream	No	MLM+VLM+MVM
PREVALENT [2020]	Image	CNN-GFs+Xformer	Xformer	Single-stream	No	MLM+MVM
XGPT [2020]	Image	OD-RFs	Emb	Dual-stream	Yes	MLM+IDA+VC+TIFG
InterBER [2020]	Image	OD-RFs	Emb	Single-stream	No	MLM+VLM+MVM
PixelBERT [2020]	Image	CNN-GFs	Emb	Single-stream	No	MLM+VLM
OSCAR [2020c]	Image	OD-RFs	Emb	Single-stream	No	MLM+VLM
VLN-BERT [2021]	Image	OD-RFs	Emb	Dual-stream	No	MLM+VLM+MVM
FashionBERT [2020]	Image	Xformer	Emb	Single-stream	No	MLM+VLM+MVM
VILLA [2020]	Image	OD-RFs+Xformer	Xformer	Single-stream	No	MLM+VLM+MVM
ERNIE-ViL [2020]	Image	OD-RFs	Emb	Single-stream	No	MLM+MVM
RVL-BERT [2021]	Image	OD-RFs	Emb	Single-stream	No	MLM+VLM+MVM
VinVL [2021]	Image	OD-RFs	Emb	Single-stream	No	MLM+VLM
VL-T5 [2021]	Image	OD-RFs	Emb	Single-stream	Yes	MLM+VLM+VQA+GRE+VC
ViLT [2021]	Image	ViT-PFs	Emb	Single-stream	No	MLM+VLM
ALIGN [2021]	Image	CNN-GFs	Xformer	Dual-stream	No	VLC
Kaleido-BERT [2021]	Image	CNN-GFs	Emb	Single-stream	No	MLM+VLM+AKPM
MDETR [2021]	Image	Xformer	Xformer	Single-stream	Yes	OD+MLM+VLC
SOHO [2021]	Image	CNN-GFs	Emb	Single-stream	No	MLM+VLM+MVM
E2E-VLP [2021]	Image	CNN-GFs	Emb	Single-stream	Yes	OD+MLM+VLM
Visual Parsing [2021]	Image	Xformer	Emb	Single-stream	No	MLM+VLM+MVM
CLIP-ViL [2021]	Image	CNN-GFs	Emb	Single-stream	Yes	MLM+VLM+VQA
ALBEF [2021]	Image	Xformer	Xformer	Dual-stream	No	MLM+VLM+VLC
SimVLM [2021b]	Image	CNN-GFs	Emb	Single-stream	Yes	PrefixLM
MURAL [2021]	Image	CNN-GFs	Xformer	Dual-stream	No	VLC
VLMO [2021a]	Image	ViT-PFs	Emb	Single-stream	No	MLM+VLC+VLM
METER [2021]	Image	Xformer	Xformer	Dual-stream	No	MLM+VLM

Figure 4: The Loss function designed in the baseline model

Classification task setting The classification task setting turns the problem of Visual question answering into a classification task by giving all the possible answer candidates. The model need to predict the probability of each answer being the right one and pick the most probable answer. Given: an image sequence of length $K : I = [I_0, I1, ..., I_k]$, K question sequences random length $Q_i = [Q_0, Q1, ..., Q_{random}]$, $i \subset 0, 1, 2...k$ and candidate answer sequence $Answer = [Answer_0, Answer_1, ...Answer_M]$. For each pair of the corresponding image and questions.We need to give a probability distribution over the answer set.

Text Generation task setting Classification task setting is very basic in the VQA setting. This approach may sometimes give right answers, but it erase the model's generation ability. In reality, answers to a certain question may actually varies from each other.Toward a certain question, even real humans give different answers under different circumstances. To imitate this behaviour, researchers use sequence to sequence models to give various of model output.In the VQA setting, the task can be illustrated as below: Given: an image sequence of length $K : I = [I_0, I1, ..., I_k]$, K question sequences random length $Q_i = [Q_0, Q1, ..., Q_{random}]$, $i \subset 0, 1, 2...k$.For each corresponding pair of question and image, we want to predict a possible answer sequence $Answer_i = [token_0, token_1, ..., token_M]$. In this sequence, each of the token is a word exists in the pre-defined vocabulary. Usually the length of the predicted sequence is not limited, the sentence only ends with a specified symbol,which is often referred as $< EOS >$.

Answer Retrieval task setting In an answer Retrieval task setting, we choose positive samples and negative samples within a batch to construct training loss. Given: an image sequence of length K where K is the batch size :$I = [I_0, I1, ..., I_k]$, K question sequences random length $Q_i = [Q_0, Q1, ..., Q_k]$ and candidate answer sequence $Answer = [Answer_0, Answer_1, ...Answer_K]$.We construct samples as $[I_j, Q_j, Answer_t]$. Altogether we will have K^2 different samples. For those samples where $j = t$, we call it a positive sample, otherwise, it is a negative sample. During the evaluation process, given an image together with a question, we compute the similarities of all Image-question and answer pairs and choose the answer from candidate answers if the answer has largest similarity with the image-question pair.

3.2 Baseline model design

3.2.1 model architecture

Our baseline model is designed under a classification task setting, which is the most basic settings among the candidate settings. In this model, we adopted a dual-stream model architecture with Bert and LSTM as text encoders and applied Resnet50 as an Image encoder. We used the attention mechanism to align the Image and Text features to make them have the same dimension. After that, the fine-grained features of Image and Text is fused by concatenation, then use a feed-forward network to get an output. This output will be send into a softmax layer before make predictions over the candidate answers.We applied cross-entropy loss over the predictions and labels to obtain the overall training loss. Given Image I_i and Question Q_i, We obtain image fine-grained features $Feature_{image-i} = [i_0, i_1, ..., ik]$ and text features $Feature_{text-i} = [t_0, t_1, ..., t_k]$ before concatenate them into a global modality fused feature vector $Feature_{fusion-i} = [f_0, f_1, ...fk]$ We then used a MLP to project the k dims into N dimensions, which represent the number of candidate answer classes numbers N. Finally, we used a softmax layer to give the predictions over k different candidate answers.For the loss function,we used the cross-entropy loss as the only training objective. The overeall equations is listed as below:

$$Feature_{image-i} = Imageencoder(I_i)$$

$$Feature_{text-i} = Textencoder(Q_i)$$

$$Feature_{fusion-i} = Concatenate(Feature_{image-i}, Feature_{text-i})$$

$$Predictions = Softmax(Feed - Forward(Feature_{fusion-i}))$$

$$Loss = Cross - Entropy(Predictions, Labels)$$

3.2.2 code analysis

The code for the forwarding process and loss computation is shown in the following figures, annotations are given in detail, so we do not make any more comments here to illustrate the meanings of each line. (See Figure 7 and 8 to see the detailed code) We also give the loss curve of this model here (See Figure 6) :

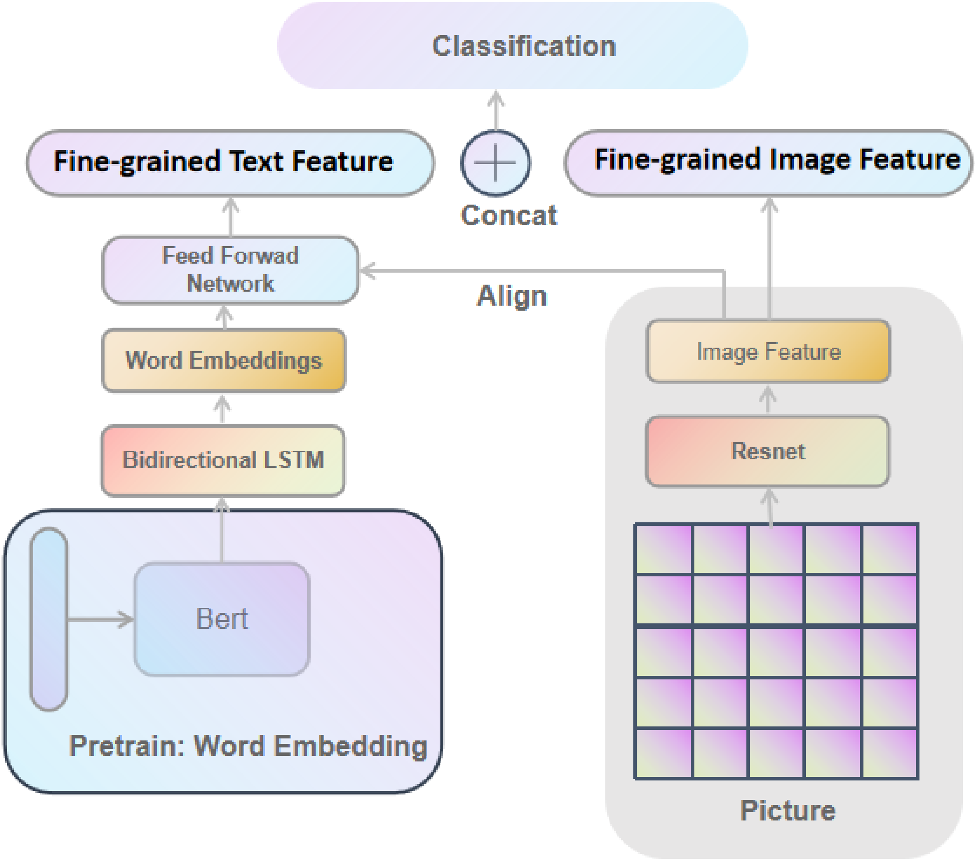


Figure 5: Our baseline model design. In this model, we adopted a dual-stream model architecture with Bert and LSTM as text encoders and applied Resnet50 as an Image encoder. We used the attention mechanism to align the Image and Text features to make them have the same dimension. After that, the fine-grained features of Image and Text is fused by concatenation, then use a feed-forward network to get an output. This output will be send into a softmax layer before make predictions over the candidate answers.

3.3 Contrastive learning based model design

The popularity of employing contrastive learning method in vision language pre-training was raised by the surprising good zero-shot performance given by CLIP. By constructing positive and negative samples, CLIP build a new constrastive learning format which is worth referring. See Figure for a closer look at the original model.

3.3.1 model architecture

In order to perform contrastive learning, we firstly change the problem setting to a retrieval task setting, and then do modifications based on our baseline model. For every processed batch in the training stage, we construct positive examples and negative examples for train. The overall model architecture can be found in Figure 9. For a batch of image-question and answer pairs,we have an image sequence of length K where K is the batch size :$I = [I_0, I1, ..., I_k]$, K question sequences random length $Q_i = [Q_0, Q1, ..., Q_k]$ and candidate answer sequence $Answer = [Answer_0, Answer_1, ...Answer_K]$. We firstly resemble the approach in the baseline model to obtain their features and project these features into the same dimension. Then we compute the contrastive learning loss. The following numpy like pseudocode gives the computing process:

$$Feature_{image-i} = Imageencoder(I_i)$$

$$Feature_{text-i} = Textencoder(Concatenate(Q_i, Answer_i))$$

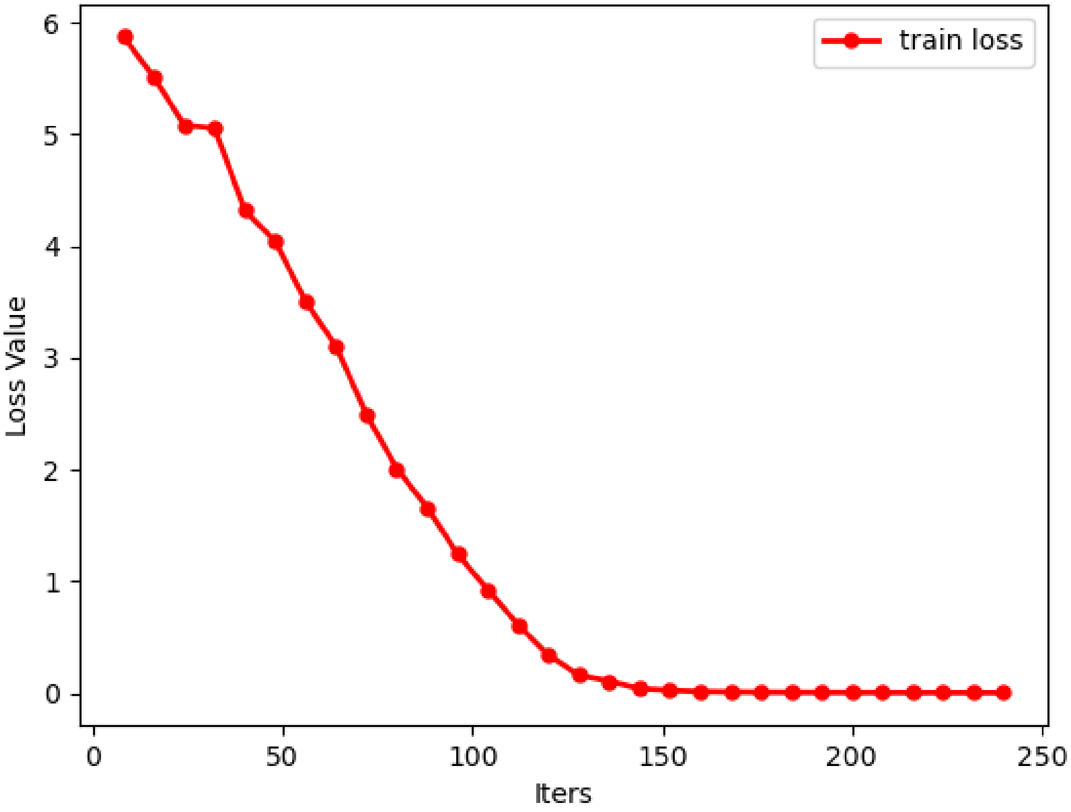


Figure 6: The Loss function designed in the baseline learning model

$$Logits = np.dot(Feature_{image}, Feature_{text-i}.transpose())$$

$$Labels = np.arange(n)$$

$$loss_i = Cross - entropy - loss(Logits, Labels, axis = 0)$$

$$loss_t = Cross - entropy - loss(Logits, Labels, axis = 1)$$

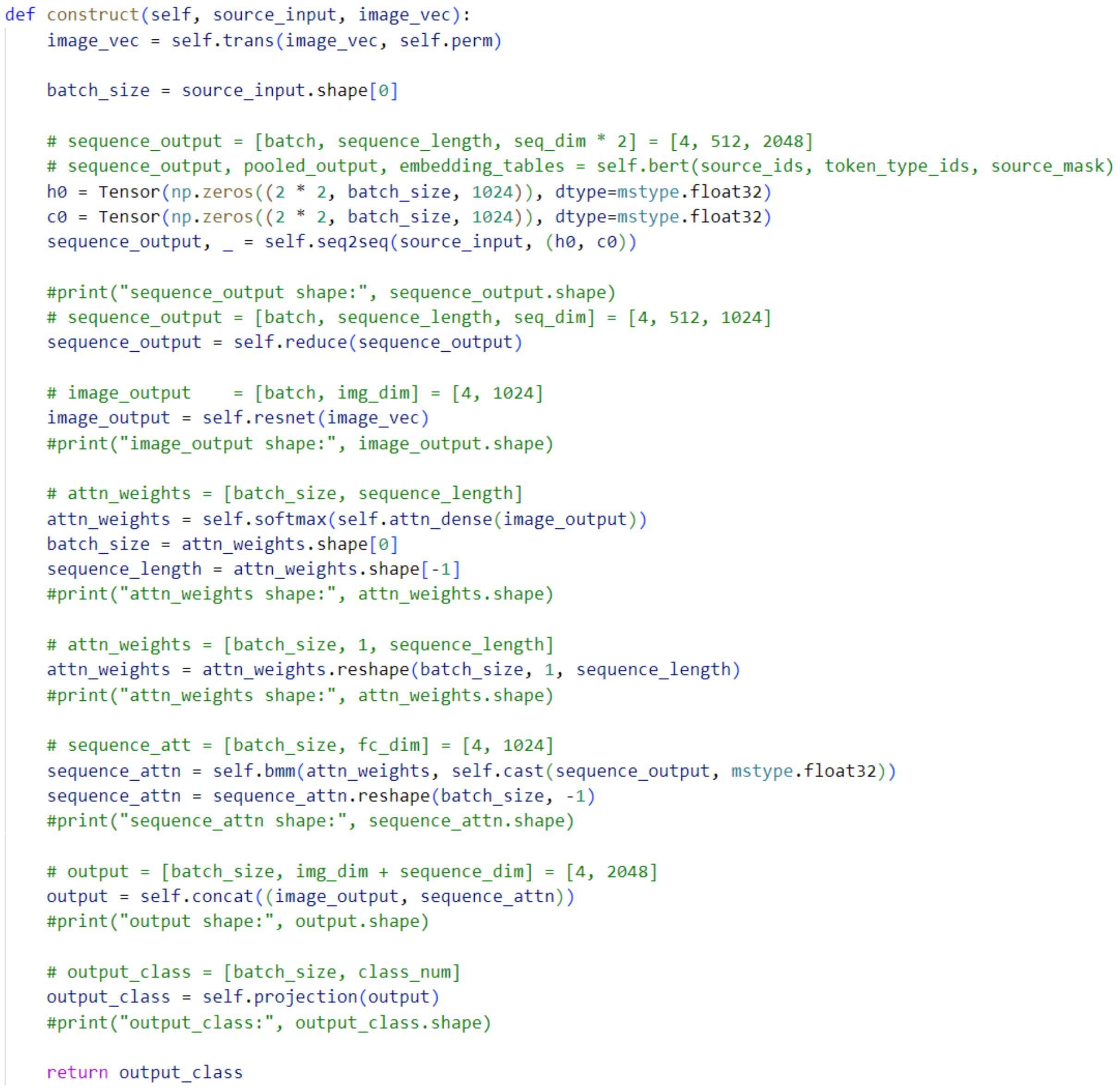
$$loss = (loss_i + loss_t)/2$$

3.3.2 code analysis

We omit the detailed code analysis part here since most of the code is similar to the baseline model.The contrastive learning accuracy is shown in the Figure above (See Figure 10). And we also give the loss curve of this training process here (See Figure 11).

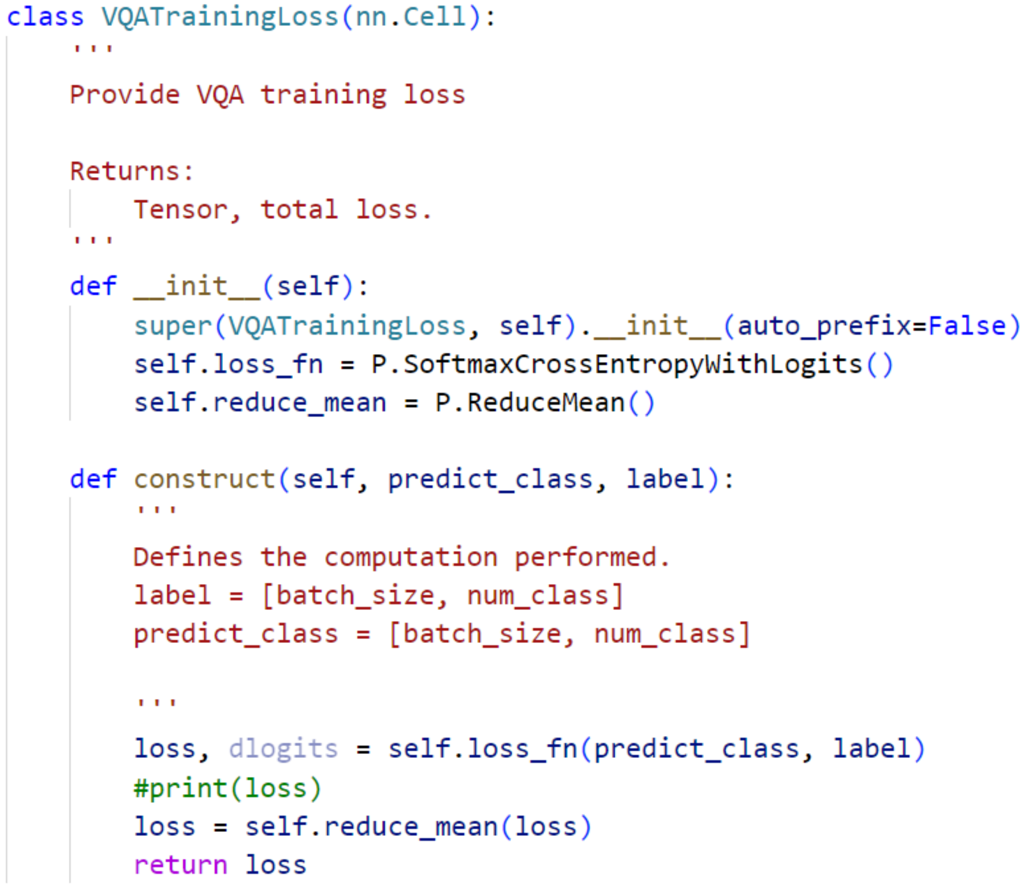
3.4 Seq2seq based model design

As is mentioned above, the problem of VQA can be also regarded as a text generation problem.Inspired by the single-stream model design and the original transformer architecture. We come up with a model to directly deal with the VQA problem. In this design, we concatenate the coarse features of input images and the features of text before sending them into a complete transformer architecture. Thus, the output of transformer encoder will contain the information of both images and questions, which makes it possible to generate ideal answers.



```
def construct(self, source_input, image_vec):
    image_vec = self.trans(image_vec, self.perm)

    batch_size = source_input.shape[0]

    # sequence_output = [batch, sequence_length, seq_dim * 2] = [4, 512, 2048]
    # sequence_output, pooled_output, embedding_tables = self.bert(source_ids, token_type_ids, source_mask)
    h0 = Tensor(np.zeros((2 * 2, batch_size, 1024)), dtype=mstype.float32)
    c0 = Tensor(np.zeros((2 * 2, batch_size, 1024)), dtype=mstype.float32)
    sequence_output, _ = self.seq2seq(source_input, (h0, c0))

    #print("sequence_output shape:", sequence_output.shape)
    # sequence_output = [batch, sequence_length, seq_dim] = [4, 512, 1024]
    sequence_output = self.reduce(sequence_output)

    # image_output    = [batch, img_dim] = [4, 1024]
    image_output = self.resnet(image_vec)
    #print("image_output shape:", image_output.shape)

    # attn_weights = [batch_size, sequence_length]
    attn_weights = self.softmax(self.attn_dense(image_output))
    batch_size = attn_weights.shape[0]
    sequence_length = attn_weights.shape[-1]
    #print("attn_weights shape:", attn_weights.shape)

    # attn_weights = [batch_size, 1, sequence_length]
    attn_weights = attn_weights.reshape(batch_size, 1, sequence_length)
    #print("attn_weights shape:", attn_weights.shape)

    # sequence_att = [batch_size, fc_dim] = [4, 1024]
    sequence_attn = self.bmm(attn_weights, self.cast(sequence_output, mstype.float32))
    sequence_attn = sequence_attn.reshape(batch_size, -1)
    #print("sequence_attn shape:", sequence_attn.shape)

    # output = [batch_size, img_dim + sequence_dim] = [4, 2048]
    output = self.concat((image_output, sequence_attn))
    #print("output shape:", output.shape)

    # output_class = [batch_size, class_num]
    output_class = self.projection(output)
    #print("output_class:", output_class.shape)

    return output_class
```

Figure 7: The construction function in the baseline model, which reflects our overall model design



```
class VQATrainingLoss(nn.Cell):
    '''
    Provide VQA training loss

    Returns:
        Tensor, total loss.
    '''
    def __init__(self):
        super(VQATrainingLoss, self).__init__(auto_prefix=False)
        self.loss_fn = P.SoftmaxCrossEntropyWithLogits()
        self.reduce_mean = P.ReduceMean()

    def construct(self, predict_class, label):
        '''
        Defines the computation performed.
        label = [batch_size, num_class]
        predict_class = [batch_size, num_class]

        '''
        loss, dlogits = self.loss_fn(predict_class, label)
        #print(loss)
        loss = self.reduce_mean(loss)
        return loss
```

Figure 8: The Loss function designed in the baseline model

3.4.1 model architecture

The general model architecture is shown in figure . The only difference between this model architecture and the original transformer architecture which was used to do translation lies in that it added a image patch sequence. Following the operations created by vision transformers, we cut the images into small patches and regard each of the patch a 'token' that transformer need to deal with. In practice, we used a 2D Convlution kernel to realize this. Given a certain image, it firstly divides a image by reshaping it from $I_i \in \mathbb{R}^{H\times W\times C}$ to a sequence of image patches $I_p \in \mathbb{R}^{N\times(P^2\times C)}$ Where N is the number of patches obtained by the dividing process. Given text feature of dimension Dim,Image I_i and the size of the patch $patch_size$. We obtain the Image feature I_p by the following equation :

$$I_p = Conv2D(in_channels = 3, out_channels = Dim, kernel_size = patch-size*patch-size, stride = patch-size)$$

See Figure 12 for the model details. Due to time reasons, we only realize the model part of this idea and did not run the training and evaluation code.

4 Experiments and results

We did comprehensive experiments over part of the dataset VQA V2.0 and get relatively good results.To illustrate the training and evaluating process of our model, we give detailed descriptions about the events happened during runtime. We also give images for the loss curve, and the overall accuracy on the overall dataset. To show our results in a clearer way, we give test results by classifying the test set by distinguishing question type. We give both the top-1 accuracy and top-10 accuracy of our model. See Figure 10 for the overall accuracy and see Figure 14 , Figure 15 for an image that dipicts the overall accuracy and visualization result.

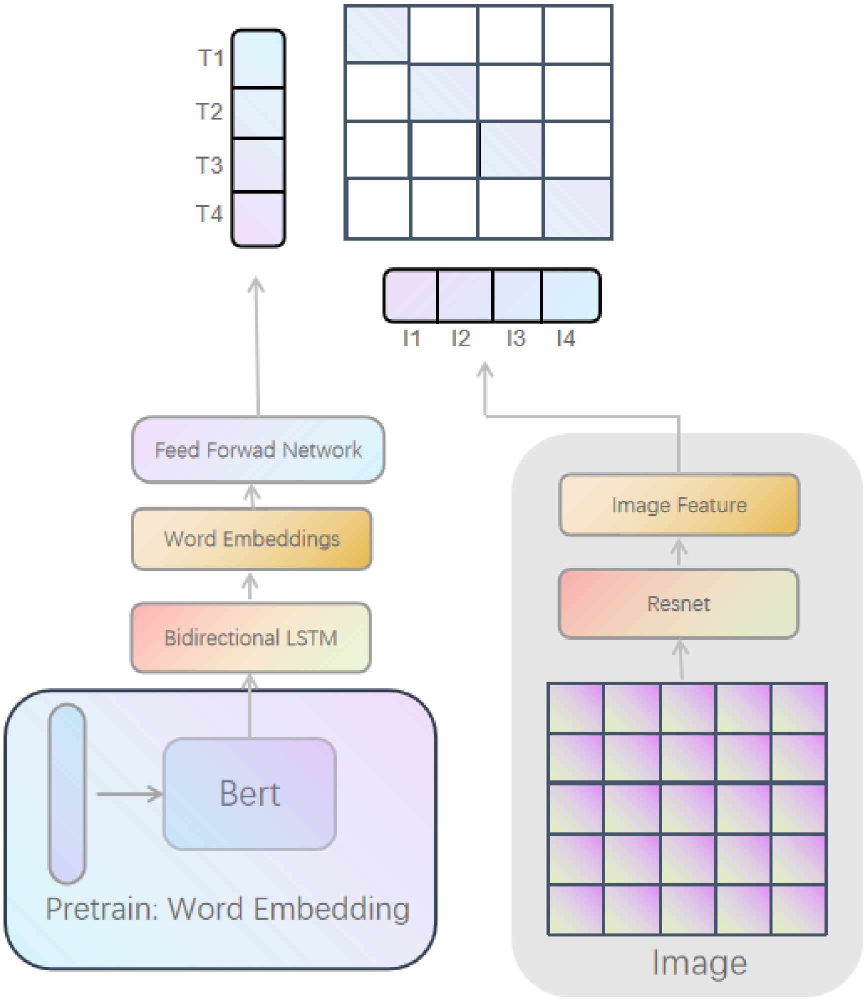


Figure 9: Model design under contrastive learning settings.For every processed batch in the training stage, we construct positive examples and negative examples for train.In the final step, the training loss is by maximize the similarity between positive image-question and answer pairs, and minimize the similarity between negative pairs,

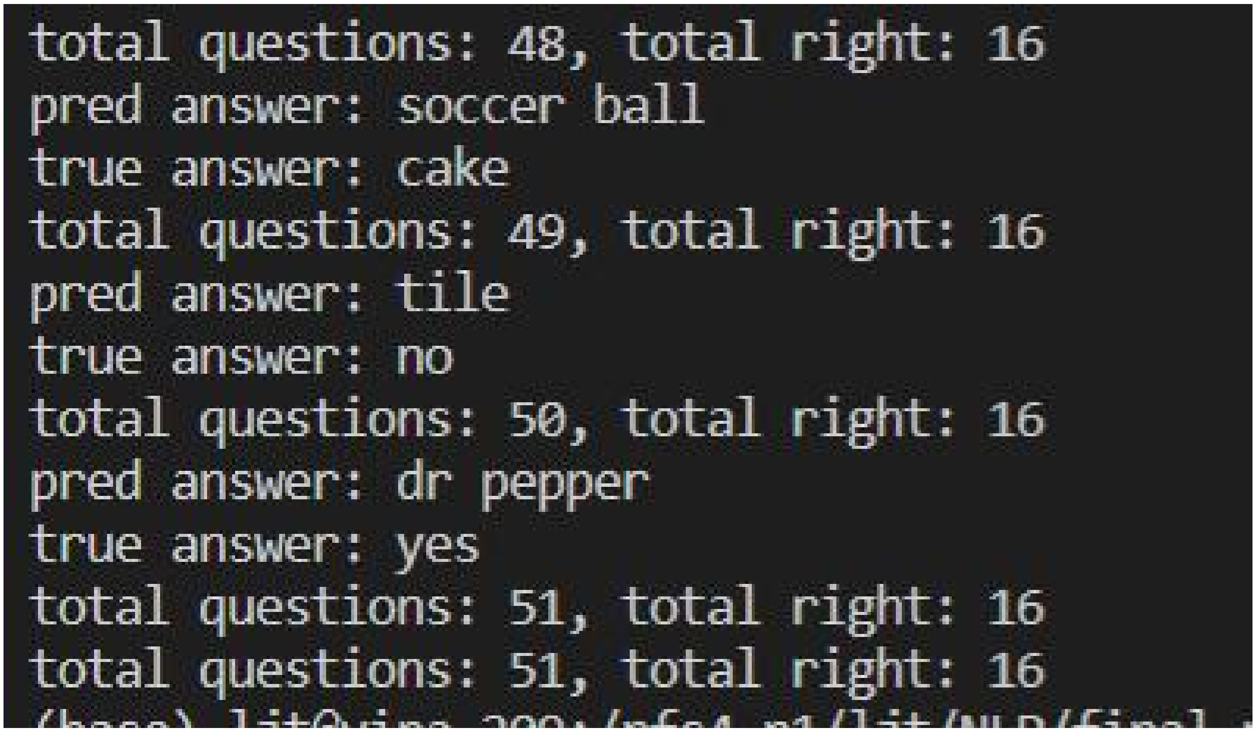
5 Utility analysis

Since in the baseline model, it seems strange for us to add an extra LSTM model after Bert. We designed a utility analysis module to prove the LSTM module actually works and helps imporve overall training accuracy. We show the result of erasing the LSTM module by comparing it to the model which uses LSTM modules. The comparison image is shown in Figure 16-19.

6 Conclusion and Frontiers

6.1 Conclusion

To conclude, after doing comprehensive survey into the area of pure visual question answering and Vision Language pre-training. We are inspired by some of the model architecture and training objective design. After analyze all the possible settings in the VQA context, we create specific models to solve the problem under three different situations. We found out that our model and idea works but still falls a little bit behind high expectations. In all of our training



```
total questions: 48, total right: 16
pred answer: soccer ball
true answer: cake
total questions: 49, total right: 16
pred answer: tile
true answer: no
total questions: 50, total right: 16
pred answer: dr pepper
true answer: yes
total questions: 51, total right: 16
total questions: 51, total right: 16
```

Figure 10: The Loss function designed in the baseline model

process, we only use one training objective at a time due to limited time to run experiments. Maybe these training objectives can be combined and obtain a better result.

6.2 Frontiers

Issues of data set biases Current VQA techniques are trained on datasets that rely heavily on bias. An obvious result of this is that the gap between the blinded VQA and the nonblinded is actually far less than we expected. But in fact our expectations for the two results are 0 and 1, respectively. In addition, in view of the huge degree of freedom of the topic of VQA, the current dataset still has many deficiencies in terms of diverse answers and practicality, resulting in poor generalization performance of the generated model.

Generative VQA and metrics The expectations for VQA are far more than doing multiple-choice questions. In recent years, there have been some attempts at generative VQA. At present, the biggest problem of generative VQA is that loss function and metrics are difficult to define. There are many studies in this direction, and some even try BERTScore, which defines another model for the evaluation of VQA. But so far, there is no metric that convinces most researchers.

Issue of count The region feature of attention-based VQA is represented by the weighting of adjacent regions, which makes it difficult to learn the quantitative features of the target. Therefore, counting is still a difficult problem in the current VQA. It is still extremely difficult to build a model that can generalize to all problems.

References

[1] Jia, C., Yang, Y., Xia, Y., Chen, Y. T., Parekh, Z., Pham, H., ... Duerig, T. (2021, July) Scaling up visual and vision-language representation learning with noisy text supervision. *In International Conference on Machine Learning (pp. 4904-4916). PMLR.*

[2] Li, J., Selvaraju, R., Gotmare, A., Joty, S., Xiong, C., Hoi, S. C. H. (2021). Align before fuse: Vision and language representation learning with momentum distillation. In International Conference on Machine Learning (pp. 4904-4916). PMLR.Advances in neural information processing systems, 34, 9694-9705.

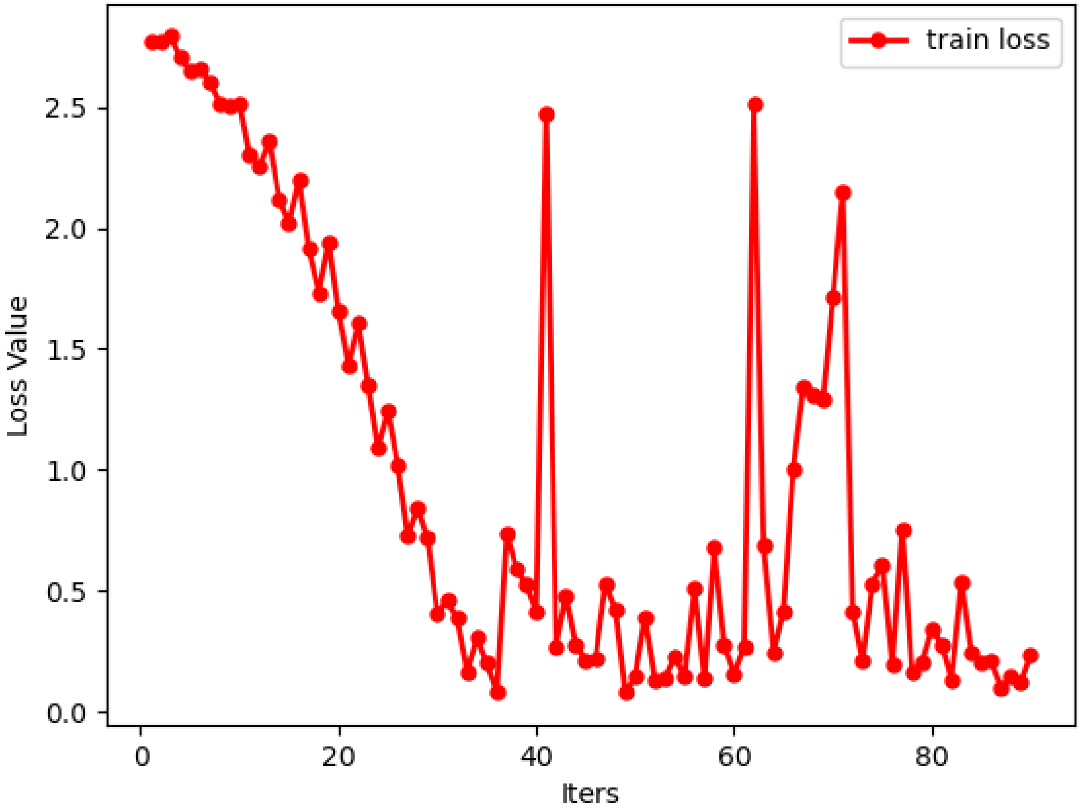


Figure 11: The Loss function designed in the contrastive learning model

[3] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., ... Sutskever, I. (2021, July). Learning transferable visual models from natural language supervision. *In International Conference on Machine Learning (pp. 8748-8763). PMLR.*

[4] Gan, Z., Chen, Y. C., Li, L., Zhu, C., Cheng, Y., Liu, J. (2020). Large-scale adversarial training for vision-and-language representation learning. Advances in Neural Information Processing Systems, 33, 6616-6628.

[5] Chen, Y. C., Li, L., Yu, L., El Kholy, A., Ahmed, F., Gan, Z., ... Liu, J. (2019). Uniter: Learning universal image-text representations.

[6] Li, X., Yin, X., Li, C., Zhang, P., Hu, X., Zhang, L., ... Gao, J. (2020, August). Oscar: Object-semantics aligned pre-training for vision-language tasks. *In European Conference on Computer Vision (pp. 121-137). Springer, Cham.*

[7] Yao, L., Huang, R., Hou, L., Lu, G., Niu, M., Xu, H., ... Xu, C. (2021). Filip: Fine-grained interactive language-image pre-training. preprint arXiv:2111.07783.

[8] Li, W., Gao, C., Niu, G., Xiao, X., Liu, H., Liu, J., ... Wang, H. (2020). Unimo: Towards unified-modal understanding and generation via cross-modal contrastive learning. arXiv preprint arXiv:2012.15409.

[9] Dou, Z. Y., Xu, Y., Gan, Z., Wang, J., Wang, S., Wang, L., ... Zeng, M. (2022). An empirical study of training end-to-end vision-and-language transformers. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 18166-18176).*

[10] Kim, W., Son, B., Kim, I. (2021, July). Vilt: Vision-and-language transformer without convolution or region supervision. In International Conference on Machine Learning (pp. 5583-5594). PMLR.

[11] Duan, J., Chen, L., Tran, S., Yang, J., Xu, Y., Zeng, B., Chilimbi, T. (2022). Multi-modal alignment using representation codebook. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 15651-15660).

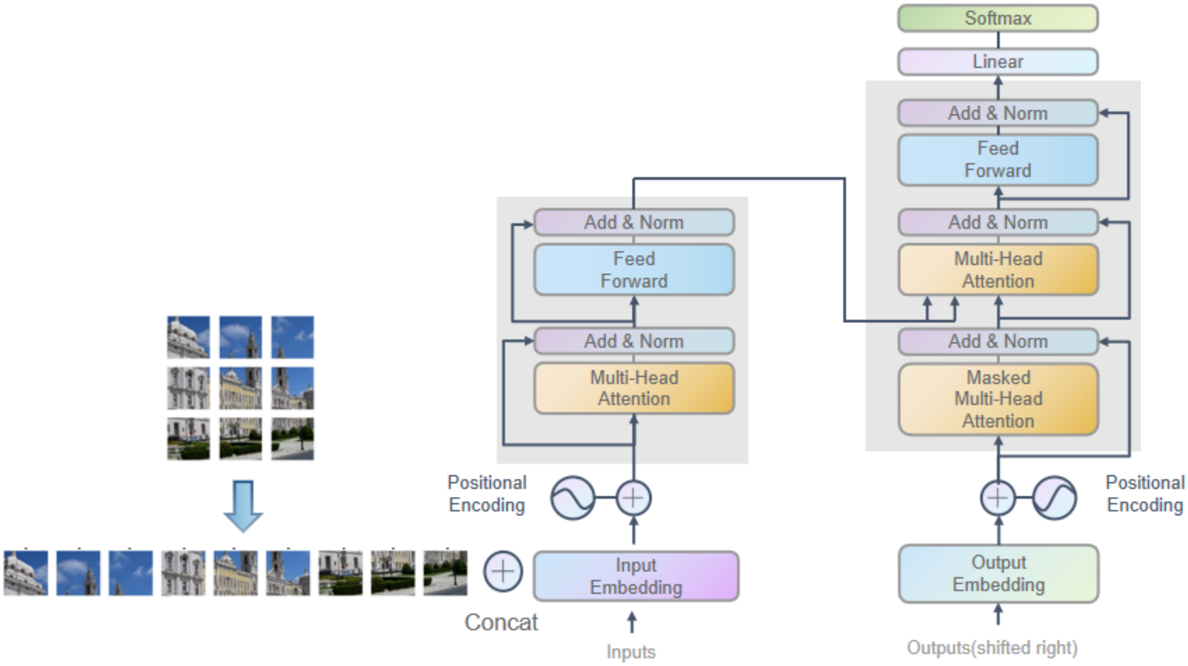
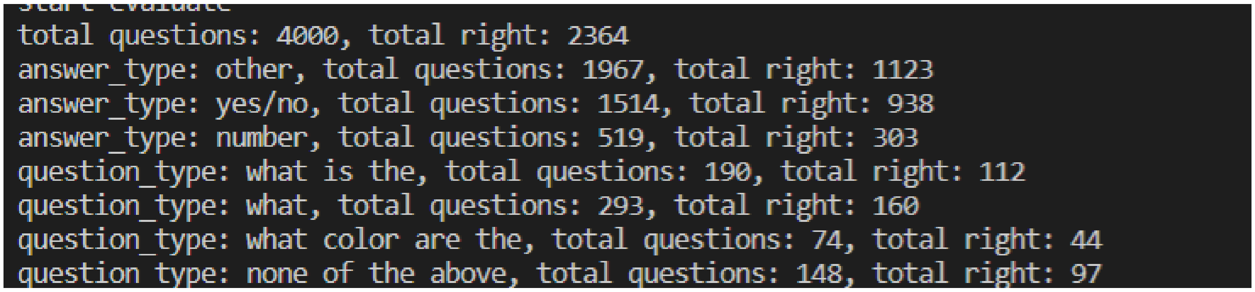


Figure 12: Model design under under seq2seq learning seettings. Notice that the image concatenate with the input text just after the embedding process, but before the position encoding process.



```
start evaluate
total questions: 4000, total right: 2364
answer_type: other, total questions: 1967, total right: 1123
answer_type: yes/no, total questions: 1514, total right: 938
answer_type: number, total questions: 519, total right: 303
question_type: what is the, total questions: 190, total right: 112
question_type: what, total questions: 293, total right: 160
question_type: what color are the, total questions: 74, total right: 44
question_type: none of the above, total questions: 148, total right: 97
```

Figure 13: Model design under under seq2seq learning seettings. Notice that the image concatenate with the input text just after the embedding process, but before the position encoding process.

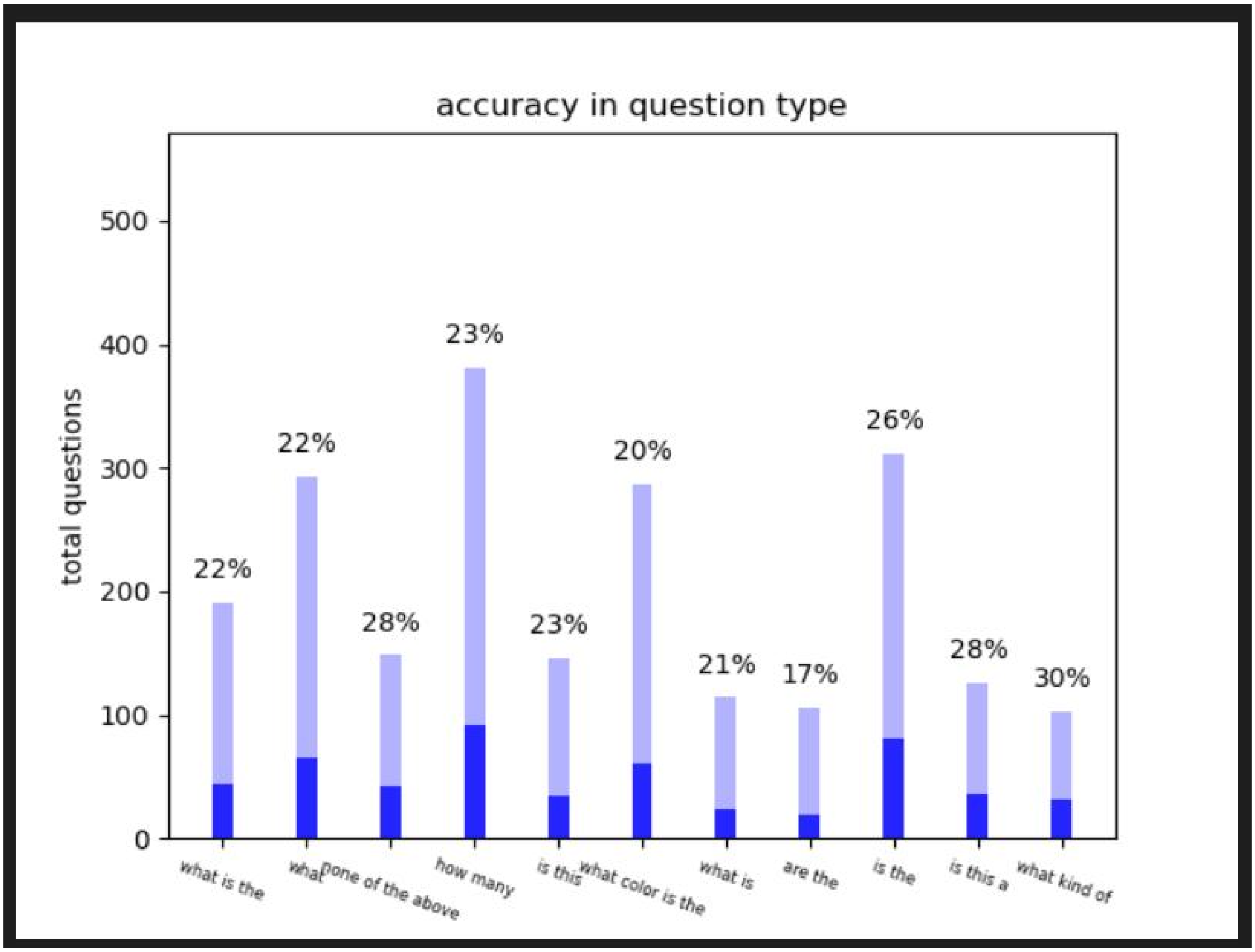


Figure 14: Baseline model Top-1 testing results on VQA dataset: we split the dataset by estimating on different types of quesitions.

[12] Huang, Z., Zeng, Z., Huang, Y., Liu, B., Fu, D., Fu, J. (2021). Seeing out of the box: End-to-end pre-training for vision-language representation learning. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 12976-12985).*

[13] Li, W., Gao, C., Niu, G., Xiao, X., Liu, H., Liu, J., ... Wang, H. (2022). UNIMO-2: End-to-End Unified Vision-Language Grounded Learning. arXiv preprint arXiv:2203.09067.

[14] Wang, J., Hu, X., Gan, Z., Yang, Z., Dai, X., Liu, Z., ... Wang, L. (2021). UFO: A unified transformer for vision-language representation learning. *arXiv preprint arXiv:2111.10023.*

[15] Lei, J., Chen, X., Zhang, N., Wang, M., Bansal, M., Berg, T. L., Yu, L. (2022). LoopITR: Combining Dual and Cross Encoder Architectures for Image-Text Retrieval. arXiv preprint arXiv:2203.05465.

[16] Cui, Q., Zhou, B., Guo, Y., Yin, W., Wu, H., Yoshie, O. (2021). ZeroVL: A Strong Baseline for Aligning Vision-Language Representations with Limited Resources. *arXiv preprint arXiv:2112.09331.*

[17] Huang, Z., Zeng, Z., Liu, B., Fu, D., Fu, J. (2020). Pixel-bert: Aligning image pixels with text by deep multi-modal transformers. preprint arXiv:2004.00849.

[18] Chen, F., Zhang, D., Han, M., Chen, X., Shi, J., Xu, S., Xu, B. (2022). Vlp: A survey on vision-language pre-training. arXiv preprint arXiv:2202.09061.

[19] Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., Parikh, D. (2017). Making the v in vqa matter: Elevating the role of image understanding in visual question answering. *Proceedings of the IEEE international conference on computer vision* (pp. 2425-2433).

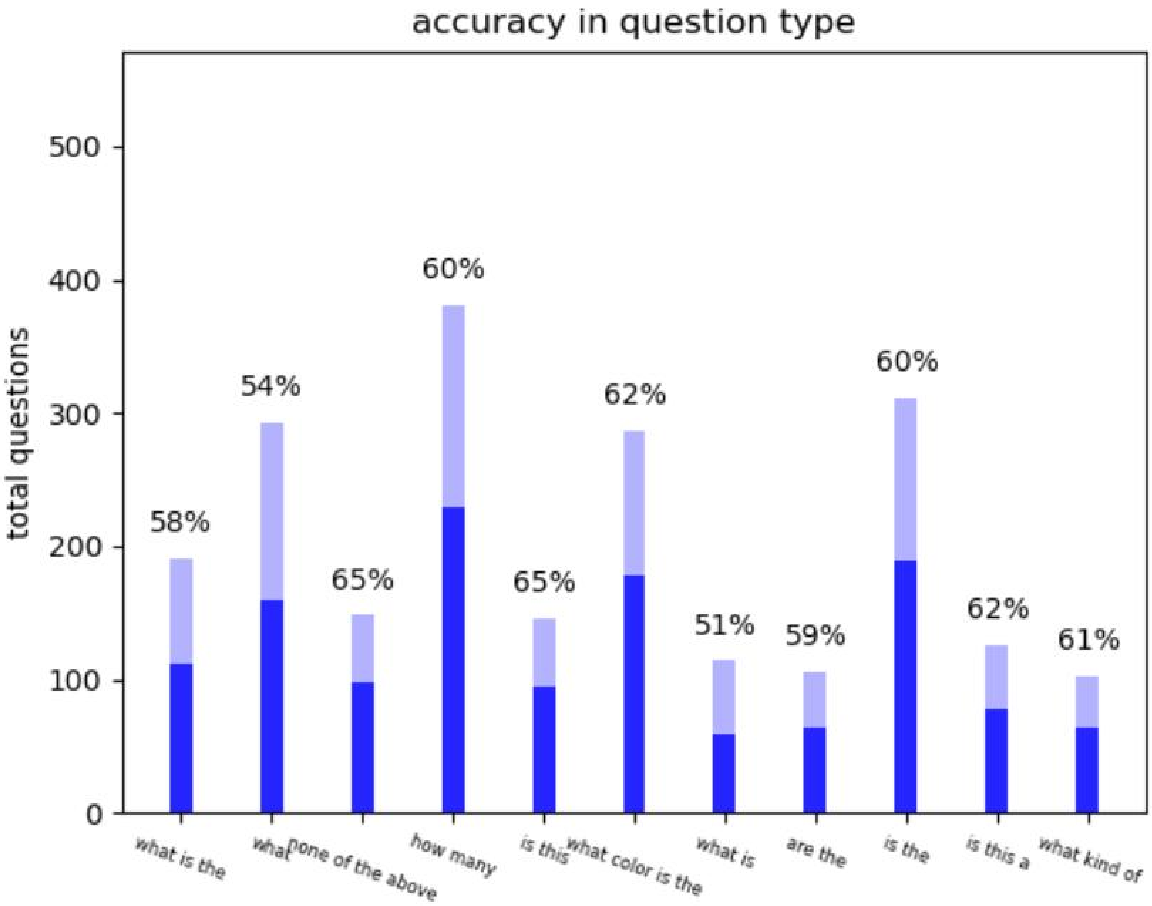


Figure 15: Baseline model Top-10 testing results on VQA dataset: we split the dataset by estimating on different types of quesitions.

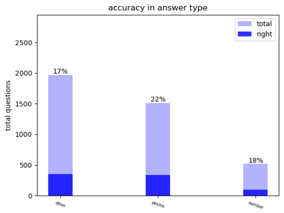


Figure 16: Top-1 Results without LSTM models

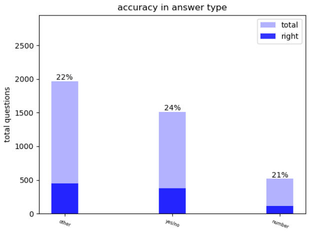


Figure 17: Top-1 Results witht LSTM models

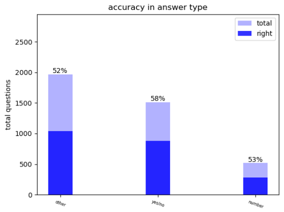


Figure 18: Top-10 Results without LSTM models

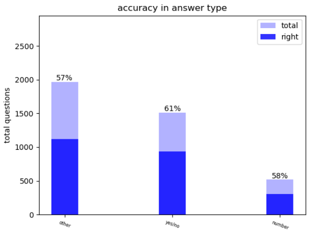


Figure 19: Top-10 Results witht LSTM models

*[20] Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C. L., Parikh, D. (2015). Vqa: Visual question answering. *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6904-6913).*