Visual Question Answering for Blind People

Senyu Tong* **Tianyi Lin** * **Yutian Zhao***
{senyut, tianyil2, yutianzh}@andrew.cmu.edu

Abstract

In recent years, many models have been proposed and achieved state-of-the-art performance on Vision-and-Language tasks. In this project, we chose LXMERT(Tan and Bansal, 2019) as our baseline model and conducted a comprehensive error analysis on its performance on the VizWiz dataset (Gurari et al., 2018). We also proposed a pipeline approach which incorporates LXMERT as the main model to predict more accurate answers to visual questions. Our model aims to improve LXMERT's confusion on different question types and the experiment results have shown that our approach achieves 0.6% accuracy over baseline model.

1 Introduction and Problem Definition

Deep neural networks have made significant progress on Vision-and-Language (V+L) tasks in the last few years. Models such as LXMERT (Tan and Bansal, 2019) and Pythia (Singh et al., 2019) have achieved state-of-the-art performance on many V+L tasks including visual question answering (VQA), visual commonsense reasoning (VCR), and image captioning. However, these models perform poorly on VizWiz dataset (Gurari et al., 2018) that collected by blind people.

In the project, we aim to build a more generalized algorithm that could address the interest of blind people using VizWiz dataset (Gurari et al., 2018). We focus on the VQA task/application since it allows blind people to naturally request what they want to know about the surrounding physical world, and helps them to overcome daily visual challenges.

The high-level problem definition of our project is to predict an accurate answer to a visual question given an image and question about it. We further categorize questions into five question types: unanswerable, binary (yes/no), number, color, and others. After conducting a comprehensive error analysis on baseline models performance, we found that models have trouble distinguishing between different question types. In order to identify the reasons behind such confusion between different question categories and improve upon it, we propose to use a pipeline approach that lets models first recognize the type of the question they are facing, and then incorporate this information into the following VQA task.

Our proposed model consists of two parts, a multi-class classifier and a main VQA model. We experimented on two training/fine-tune strategy: multitask learning and independent leaning, where we trained classifier and VQA model both jointly and independently. The experiment results have shown that our proposed approach achieved 0.6% accuracy improvement over baseline model.

2 Related Work and Background

2.1 Related Tasks

Over the years, there has been rapid progress in bridging vision and language in the research community. Various models have been developed and applied to a wide range of vision-and-language tasks. We'll briefly describe the problem and training objective of five well-established tasks and include relevant papers.

Visual Question Answering (VQA) has received a lot of attention in recent years. A natural language question regarding an image is presented and the model will output the correct answer to it. Various models are proposed for this specific task (Agrawal et al., 2016; Tan and Bansal, 2019; Ben-younes et al., 2017; Fukui et al., 2016).

Visual commonsense reasoning (VCR) is similar to VQA, but besides predicting the answer,

*Everyone Contributed Equally – Alphabetical order

the model is also expected to select the correct answer justification among multiple choices (Lu et al., 2019; Li et al., 2019; Tan and Bansal, 2019).

Visual grounding (VG) aims to localize an image region or object given a natural language question. Common approaches to this task including bounding box proposals and reranking a set of image region candidates (Lu et al., 2019; Li et al., 2019; Fukui et al., 2016; Yu et al., 2018).

Image captioning recognizes the context of an image and adds a descriptive sentence to it. Models targeting this task are mainly attention-based neural networks that could extract deep features (Anderson et al., 2018; Donahue et al., 2016; Xu et al., 2016).

Image-text matching aims to measure the visual-semantic similarity between a text and an image. It has been widely applied to other applications such as image search for a given query (Kim et al., 2018; Nam et al., 2017).

Besides these five tasks, there are also many other popular vision-and-language transfer tasks such as caption-based image retrieval (Lu et al., 2019) and emotion recognition and sentiment analysis (Delbrouck et al., 2020).

2.2 Related Techniques

In this section, we briefly review the techniques used in previous related work with a focus on the feature representation, attention mechanisms and fusion methods.

2.2.1 Feature representation

Generating feature representation has been a crucial step in transforming raw data to meaningful features. There has been substantial past works in developing single modality models separately for vision and language representations. For image representations, common backbone models include Faster R-CNN (Ren et al., 2016), R2+1D-(152) (Tran et al., 2015), VGG (Simonyan and Zisserman, 2015), ResNet (He et al., 2015) and fishNet (Sun et al., 2019). In terms of language representations, BERT (Devlin et al., 2019), a transformer language model pretrained on the BookCorpus (Zhu et al., 2015) and English Wikipedia is a popular choice nowadays.

Language embeddings and image embeddings are then combined through summation or multiplication (Lu et al., 2019; Delbrouck et al., 2020) to form a joint representation for fine tuning at a later stage. Another common approach to generate joint representation involves early fusion of image and text. Models are pretrained with multi-modal objectives such as masked language modeling with image and sentence-image prediction (Lu et al., 2019; Li et al., 2019; Tan and Bansal, 2019) to directly generate joint representations.

2.2.2 Attention Mechanism

Attention mechanism has been proven effective in many tasks including VQA. Previous attention approaches commonly used in VQA or related tasks can be classified into following four categories:

- Self attention that aggregate information inside each modality by query-key-value attention mechanism (Vaswani et al., 2017).
- Question-guided visual attention (Xu et al., 2016; Xu and Saenko, 2016) that compute attention on image region.
- Co-attention that jointly reason about both question and visual attention to interact across the two modalities. It uses image representation to guide the question attention and question representation to guide image attention (Lu et al., 2017, 2019; Yu et al., 2019). There also exist many variants of co-attention such as bilinear attention (Kim et al., 2018) and dense co-attention network (DCN) (Nguyen and Okatani, 2018) that considers interactions between every pair of question words and image regions.
- The intra- & inter-modal attention (DFAF) (Peng et al., 2019) that consider both inter-modality attention and intra-modality attention, where attention for intra-modality feature aggregation is dynamically modulated by the other modality using the pooled features.

2.2.3 Fusion Methods

The common multi-modal fusion approach is that visual and language features are extracted from the image and question independently as the first step, and then they are fused to compute the final results.

In previous studies, many works employed simple fusion methods that use element-wise product or summation of the visual and language features as final fused feature, and fed it to fully connected layers to predict results. (Lu et al., 2019; Delbrouck et al., 2020)

More complex fusion techniques have also been explored. For instance, Multimodal Compact Bilinear pooling (MCB) (Fukui et al., 2016) uses bilinear pooling method to compute the outer product between visual and language vectors as their fusion. In contrast to element-wise product approach, MCB allows a multiplicative interaction between all elements of both vectors. MUTAN fusion model (Ben-younes et al., 2017) is able to represent full bilinear interactions between visual and language modalities using Tucker decomposition of the correlation tensors, while maintaining the size of the model tractable. Multi-modality Latent Interaction Network (MLIN) (Gao et al., 2019) first encodes question and visual features into latent summarization vectors and then it go through the process of interaction, propagation and aggregation to achieve multi-modality reasoning.

3 Task Setup and Data

3.1 Task Formulation

The high level task of our project is to predict an accurate answer to a visual question given an image and question about it. As a sub task, we classified questions into five types (unanswerable, binary yes or no, number, color and others), and we aim to predict the correct type of a visual question. By explicitly or implicitly injecting question type information into model architecture, we believe models will have better performance in the visual question answering task. We will discuss this in more details in later sections.

3.2 Dataset Statistics

We split dataset into questions, answers and images, and calculate statistics for them independently.

Questions: We examine the questions in both sentence and word level. We first calculate the distribution of the questions and observe that the percentage of questions that appear more than once is 54.4%. "What is this?" is the most common question that occupies more than 13% of all questions while the second most common question "What color is this?" is only about 1.7%. The frequency distribution is shown in Figure 1. We also analyze question diversity by computing statistics on sentence lengths. The mean question length is 6.76 words and the 25th and 75th percentile lengths are four and eight words respectively. The longest question has 62 words and the shortest one has only two words.

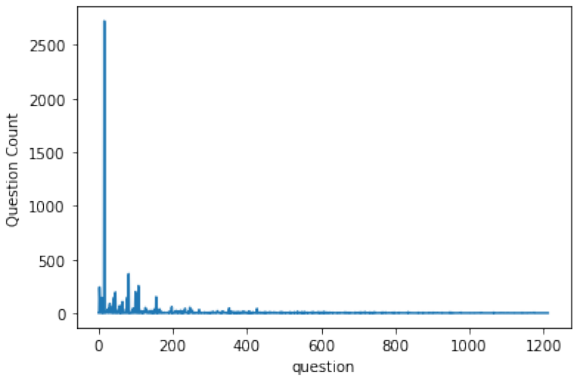

Figure 1: Question Frequency Distribution

Figure 2: Answer popularity word cloud excluding "unanswerable" and "unsuitable image" answers

Answers: Since VizWiz (Gurari et al., 2018) images are collected by blind people, a large portion of images have low quality and hence are not answerable. Based on our analysis, only 73.04% of visual questions are tagged as answerable.

We also analyze the diversity of answers by calculating statistics on answer length. The mean and median answer lengths are 1.66 and 1.00 words respectively, and the max answer length is 21 words. The result indicates that the visual questions in dataset tend to have short answers. We also generate a word cloud computed on all answers in the dataset excluding "unanswerable" and "unsuitable image", which is shown in Figure 2.

Images: We run pre-trained YOLO (You Only Look Once) v3 model (Redmon et al., 2016) as a naive, coarse-grained, first-step baseline to analyze the training images. The weights of the model have been obtained by training on COCO dataset (Lin et al., 2015), and we hereby adapt the setting, trying to detect objects from 80 classes.

We count the number of bounding boxes on each image after we run objectness score thresholding and Non-Maximum Suppression to avoid overlapping. We observe that, we get 24,480 bounding boxes in total, and most images have either 0 (9,280

out of 23,953) or 1 (8,609 images) bounding box. The average number of bounding boxes per image is 1.02, and the third-quartile count is 2.0.

3.3 Metrics

We use accuracy metric as our main evaluation metric following VizWiz challange official documentation:

$$accu = min(1, \frac{\#\ humans\ that\ provided\ that\ answer}{3}) \quad (1)$$

In order to evaluate model's performance on different question type categories, we also adopt precision, recall and F-score as our evaluation metrics:

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives} \quad (2)$$

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives} \quad (3)$$

$$F - score = \frac{2 * Precision * Recall}{Precision + Recall} \quad (4)$$

4 Baselines

4.1 Model Selection

We would run the following 2 models as baselines. Their results on VizWiz dataset are listed in table 2.

LSTM+CNN We ran a variant of model described in (Kazemi and Elqursh, 2017) referencing github [1]. Visual features are extracted using a pretrained ResNet-152 model on ImageNet. Questions are embedded and encoded with LSTM. Image features and question embedding are then combined to compute multiple attention distributions over image features. The attended image features and the questions are concatenated and fed into two fully connected layers. We trained model for 40 epochs with batch size 128 and learning rate 0.001.

LXMERT (Tan and Bansal, 2019) constructs separate representations for images and texts, use (Anderson et al., 2018) for images. Have two encoders each for one modality with self-attention, and then a cross-modality encoder with cross-attention. We ran LXMERT referencing github [2]. We used the image representation extracted using Faster-RCNN and sentence representation extracted using Bert. Following (Tan and Bansal, 2019), we finetuned the model for 20 epochs with batch size 32.

[1] https://github.com/DenisDsh/VizWiz-VQA-PyTorch
[2] https://github.com/airsplay/lxmert

4.2 Baseline Error Analysis

We defined an output answer to be incorrect if it doesn't equal to the most popular answer provided by the 10 crowd workers. Following the metric given on Vizwiz website, we also break down the questions into the following categories for more detailed error analysis:
1. Binary: if at least three reference answers are either "yes" or "no".
2. Number: if at least three reference answers are numbers. We define an answer as number if it is either an integer, float or double without any other characters.
3. Unanswerable: if at least three reference answers are unanswerable.
4. Color: if at least three reference answers are color.

The overall accuracy for LSTM+CNN and LXMERT is 48.97% and 49.63% respectively. Please see table 2 for detail test results.

For incorrect outputs on each of the above question categories, we conducted error analysis in terms of their question types. Please see table 1 for test statistics of each question category.

Binary questions constitute 6.34% of the total dataset, and LSTM+CNN and LXMERT's accuracy is about 66% and 77% respectively.

Number questions only constitute 1.63% of the total dataset, and the baseline models' accuracy is about 30-36%. We also analyzed the number of workers that agreed on the incorrect output answer and similarly to binary questions, more than 70% of the output answer doesn't exist in the workers reference answers. All three models tend to output "unanswerable" to number questions, failing to recognize the numbers in the image.

Unanswerable questions constitute 25.6% of the total dataset, and LSTM+CNN and LXMERT's accuracy is about 83% and 60.9% respectively. We noticed that LSTM+CNN's accuracy is higher than LXMERT in terms of unanswerable questions, indicating that LSTM+CNN model favors unanswerable as its output.

Color questions constitute 8.6% of the total dataset, and LSTM+CNN and LXMERT's accuracy is about 60% and 65% respectively.

We're interested in the models' ability to distinguish different kinds of questions and from the result in table 1, we can see that 10% of the incorrect answers for unanswerable questions are binary while only about 1% are number. LXMERT output

Method	Binary (yes/no)			
	yes/no swapped	unanswerable	ques begins with "can you .."	ques length stats
LSTM+CNN	31.0%	50.3%	19.3%	max: 157 min: 10 ave: 51
LXMERT	**52.9**%	**46.2**%	27.9%	max: 204 min: 10 ave: 54
Method	Number			
	% of non-numbers	unanswerable	ques begins with "can you .."	ques length stats
LSTM+CNN	76.7%	60.0%	10.0%	max: 129 min: 9 ave: 44
LXMERT	93.6%	50.0%	9.4%	max: 249 min: 8 ave: 43
Method	Nonanswerable			
	% of binary	% of numbers	ques begins with "can you .."	ques length stats
LSTM+CNN	9.6%	1.3%	7.6%	max: 249 min: 8 ave: 42
LXMERT	10.1%	0.8%	10.2%	max: 249 min: 8 ave: 42
Method	Color			
	% of non-color	% of unanswerable	ques begins with "can you .."	ques length stats
LSTM+CNN	55.0%	31.9%	0.6%	max: 118 min: 11 ave: 72
LXMERT	58.7%	24.0%	0.7%	max: 118 min: 11 ave: 77

Table 1: Error analysis of models' incorrect outputs on different question categories.

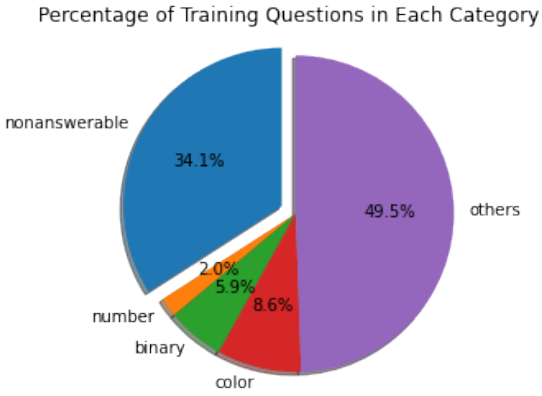

Figure 3: Question Type Distribution

"unanswerable" 63% of the time for binary questions and output binary answers for 10% of the unanswerable questions. We would like to identify the reasons behind such confusion between different question categories and improve upon it.

5 Proposed Approach

We categorize questions into five types as in Figure 3. We observe that models tend to have difficulties in distinguishing answer types as in Table 1. Based on our error analysis in previous section, we believe a pipeline approach would benefit by allowing models to first figure out which type of visual scenes and questions they are facing. Although pipelined model has been widely adopted in uni-modal tasks such as Question Answering (Kwiatkowski et al., 2019; Wang et al., 2020), Handwritten Text Recognition (Chung and Delteil, 2019) and Data Collection and Processing (Di Mitri et al., 2019), it was rarely used in multi-modal settings. To the best of our knowledge, we are the first to experiment with the hierarchical pipeline approach for multi-modal Visual Question Answering task.

Inspired by (Wang et al., 2020) that proposed a two-stage training procedure for Question Answering task, we propose to use a separate classifier that predicts the question types on top of the main VQA model. We would also experiment by training the two models jointly and also multitask learning. A more detailed discussion would be in Section 5.1. With this additional information on question type explicitly injected, we believe the model would be less likely to mix different kinds of questions.

5.1 Model Architecture

As shown in Figure 4, our model now has two parts. The left is a classifier, and the right is the main VQA model.

Classifier In our work, we are motivated to utilize a prior question type distribution as an augmentation to the main VQA model. In this project, we experimented LXMERT, LSTM+CNN and BERT as the classifier to generate this "prior" distribution.

Following (Tan and Bansal, 2019), we use the image features generated by Faster-RCNN and word embeddings generated by Bert as inputs and run three encoders: object relationship encoder, language encoder and a cross-modality encoder. The classifier has two intermediate outputs for language and vision respectively before the final MLP clas-

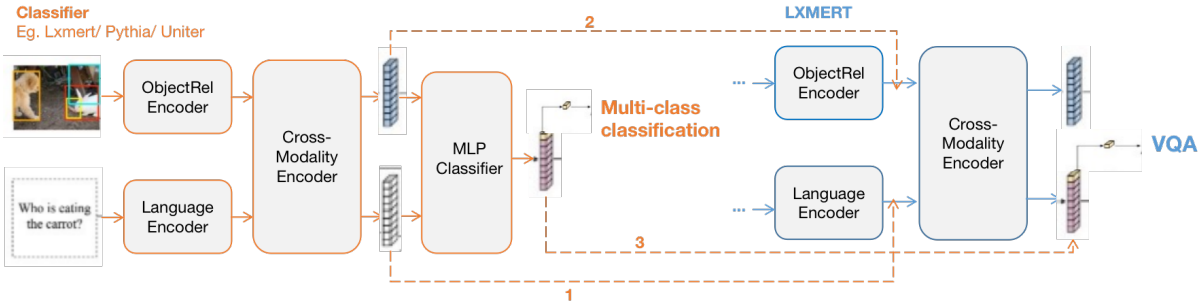

Figure 4: Proposed model architecture. Our model consists of a vision-language multi-class classifier and a main VQA model.

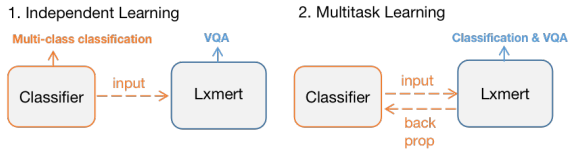

Figure 5: Finetune Strategy. The left figure represents two models learning independently and the right represents multi-task learning.

sification layer is applied. This final layer will predict the most probable question type.

VQA Model Besides the classifier, we propose to use LXMERT as our main VQA model as LXMERT outperformed LSTM+CNN in our baseline result analysis. Due to the fact that the classifier and the main VAQ models are inherently multimodal, both models will generate image, language and cross-modality representations as intermediate outputs, making further fusion desirable. This also makes our approach uniquely multimodal.

After the classifier outputs a prior probability distribution, the VQA model will adapt the image, language and cross-modality intermediate outputs from the classifier. As shown by the dashed lines 1 and 2 in Figure 4, we augment the language and image representations generated by the uni-modal encoders in the main LXMERT model with the two representations generated by the cross-modality encoders in the classifier model. Additionally, as indicated by dashed line 3, we fuse the hidden states of the final layer in the classifier into the output representation of the main LXMERT's cross-modality encoder.

Fusion Method We take the element-wise product of the image, language and the final representations described above as our fusion method.

5.2 Finetune Strategy

In terms of finetune strategy, we experimented training the classifier model and the main VQA model jointly and independently as shown in Figure 5.

Multitask Learning We also trained the classifier and the main VQA model simultaneously with loss combined for the two tasks. We expect it to generate more robust intermediate representations but result in a more complicated model structure and prone to overfitting. A more detailed discussion of Multi-Task Learning would be in section 5.3.

Independent Learning Alternatively, in order to prevent over-fitting, our classifier could be completely separate and independent of the main model, thus the classifier's parameters would not be updated during the main model's finetuning process.

5.3 Multi-Task Learning

As shown in Figure 5, in order to train the classifier and main VQA model jointly, we would define a new loss function to achieve this multi-task learning. First, following (Tan and Bansal, 2019), we would minimize the binary cross entropy for each task:

$$L = -y^* log(prob) - (1 - y^*) log(1 - prob) \quad (5)$$

Second, we will then train the entire model with the weighted loss:

$$L_{final} = \alpha L_{classifier} + \beta L_{VQA} \quad (6)$$

Here α and β are two hyperparameters that we will also finetune. Finally, this loss will be back-propagated through the entire model and the model parameters will be updated.

6 Experiments and Results

Our experiment results are shown in Table 2. Given question-type classification results, we conduct ex-

Method	Accuracy				Overall
	Binary (yes/no)	Number	Unanswerable	Color	accuracy
LSTM+CNN	66.42%	30%	83.02%	60%	48.97%
LXMERT	77.50%	35.68%	69.54%	64.82%	**49.63%**
LXMERT$_{HardMasking}$	73.97%	24.88%	55.43%	59.28%	42.43%
LXMERT$_{FusionLogits}$	71.29%	24.88%	56.55%	59.59%	42.55%
LXMERT$^{*}_{QuesTypeOracle}$	79.68%	27.70%	72.73%	70.77%	**51.29%**
BERT+LXMERT$_{FusionEarly}$	72.75%	36.15%	70.96%	61.13%	49.29%
BERT+LXMERT$_{FusionLate}$	71.78%	34.74%	72.31%	61.33%	49.59%
BERT+LXMERT$_{FusionE+L}$	77.01%	33.80%	70.26%	65.13%	49.53%
LSTMCNN+LXMERT$_{FusionEarly}$	75.43%	35.68%	71.23%	65.74%	49.85%
LSTMCNN+LXMERT$_{FusionLate}$	74.09%	33.33%	71.95%	66.05%	**50.08%**
LSTMCNN+LXMERT$_{FusionE+L}$	71.05%	37.09%	70.32%	63.9%	47.88%
LXMERT$_{Joint}$	73.48%	32.86%	73.59 %	61.03 %	**50.24%**

Table 2: VizWiz2020 QA development set experiment results. The top block rows are the baselines we run. The middle block contains results under Independent setting. The first row is for masking using types predicted by a seperate LXMERT classifier, and the last row is for masking using ground truth question types. The last block contains results for our pipeline models in a form of question-type Classifier + Main VQA model. We use multi-task learning to finetune classifier and main model jointly.

periments using hard-masking or soft-fusion. And for soft-fusion, we experimented fusing different classifiers and also different fusion methods.

6.1 Independent Learning

As in the middle block of Table 2, to test our hypothesis that knowing question types help predicting answers, we first use ground-truth types as oracles, and then we do hard-masking by discarding answer candidates with wrong types. We have a 2 percent improvement. However, when experimenting under our independent learning setting, which is to use the answer-type predicted by LXMERT to do hard-masking, we got negative results. We believe this is due to cascading failure as the LXMERT classifier only achieved around 73% accuracy on question type prediction.

6.2 Multi-task joint Learning

As shown in the last block of Table 2, we experimented multi-task learning to jointly update classifier and main model. Instead of masking out answer candidates, we now do soft-fusion by injecting the pooled output of the classifier into the main VQA model.

We use BERT and LSTM+CNN as separate classifiers finetuned jointly with main VQA model using multi-task loss described in previous sections. We also experiment using a single LXMERT for both answers and question types prediction, denoted as LXMERT$_{joint}$.

The subscript $FusionEarly$ stands for fusing the pooled output of the classifier, denoted as $cls\_poolout$, to the language representation before entering Cross-Modality Encoder in our main VQA model; $FusionLate$ stands for fusing $cls\_poolout$ after Cross-Modality Encoder outputs combined VL representation, before entering the final logit classification layer; $FusionE+L$ stands for fusing $cls\_poolout$ both before and after entering Cross-Modality Encoder.

6.2.1 BERT

We first finetune our BERT-base classifier alone to get 63% accuracy in predicting question types and store the parameters. Then we combine BERT classifier with the main VQA model, and load those weights for the classifier. We also load weights for LXMERT pretrained on VQA dataset. We then train the joint model for another 20 epochs.

Using late fusion outperforms other two fusion methods, though they are all relatively close. The performance of the joint model using late fusion is slightly below our baseline LXMERT by 0.04%. Also, contrary to what we envisioned, adding BERT classifier fails to improve accuracy under each question type.

6.2.2 LSTM+CNN

Similar to BERT, we first fine-tuned LSTM+CNN V+L classifier on VizWiz dataset with 70% accuracy on question type classification, and we use this to initialize weights for classifier part of the pipeline. We use LXMERT as VQA model part of the pipeline, and we also load pre-trained weights for it. Then we trained this joint model for another 20 epochs.

Among the three fusion strategies, $FusionEarly$ yielded the best result with overall 50.08% accuracy, slightly higher than baseline accuracy. $FusionLate$ is better than $FusionE + L$, and $FusionE + L$ failed to improve performance over baseline LXMERT. We believe $FusionE + L$ performs poorly due to repeated information being fused leads to model overfitting.

Although LSTM+CNN model also has image representation, we did not fuse image representation into LXMERT. Both LXMERT and LSTM+CNN's image features are already pre-generated, so we believe the fusion of image features is unnecessary in this case.

6.2.3 LXMERT

We reach highest overall accuracy using a single LXMERT model with two final classification layers: one for question types and the other for answers. The last answer-prediction layer takes in the element-wise product of the last hidden state from type-prediction layer, and the VL representation generated by Cross-Modality encoders. With this additional task, the overall accuracy increases from 49.63% to 50.24%.

7 Analysis

7.1 Ablation on Question-Type Classifier Structure

Comparing the performance of our updated methods to baseline model, our results indicate that explicitly adding the question-type classifier has no clear advantage over using the plain LXMERT VQA model, though we achieve a 0.6% in accuracy improvement using LXMERT joint.

However, from the last row of the middle block of Table 2, we see that using ground truth question types as oracles yields a 2% increase in overall accuracy. The performance of the model under three out of four question types also improves. We believe this suggests that by recognizing what type of the answer to look at in the first place, the model reduces its search space and is more likely to find the correct answer.

Figure 6 shows the heat map of the ratio of predictions with mixing types using baseline (on top) and LXMERT$_{Joint}$ (on bottom). We observe that, though our augmented model does not decrease the overall number of mixing type predictions, the errors now concentrate more on *unaswerable* to *others*. For *binary*, *number* and *color*, we manage to decrease the probability for these types to be mixed using LXMERT$_{Joint}$. Given we do not introduce significantly more parameters in LXMERT$_{Joint}$ (only an additional feed-forward layer for type-classification,) we conclude that our joint learning setting indeed helps models to know what type of questions they are looking at, even though there exists a bias towards *unanswerable*.

Figure 6: Ratio of predictions with mixing types (in percentage). Rows (y-axis) are predicted types; columns (x-axis) are actual types. The upper figure is the heatmap for Baseline model, lower figure is for LXMERT joint model

Given this, we suspect following reasons for why our improvements are not significant:

- Although intuitively, knowing question types would boost performance, the room for such improvement in this aspect may be very limited. The oracles only bring 2% increase in accuracy, suggesting that the model already learned this information relatively well **implicitly**.

- Our augmented models fail to decrease the overall number of instances where one answer type is mixed with others. From our experiments, regardless of which classifier we choose, we always have around 33% of total

Method	Unanswerability		
	Precision	Recall	F1 Score
LXMERT	**63.0%**	85.5%	**72.5%**
BERT$_{FusionEarly}$	59.6%	86.7%	70.6%
BERT$_{FusionLate}$	60.7%	**87.3%**	71.6%
BERT$_{FusionE+L}$	**62.3%**	85.8%	**72.2%**
LSTMCNN$_{FusionEarly}$	60.6%	**88.4%**	71.9%
LSTMCNN$_{FusionLate}$	61.0%	87.9%	72.0%
LSTMCNN$_{FusionE+L}$	60.1%	87.0%	71.1%
LXMERT$_{Joint}$	60.7%	**87.4%**	71.7 %

Table 3: Precision, recall, and F1 scores on predicting if the given visual question is unanswerable.

predictions not in their correct types. Some models even get more types wrong than our baseline (32%). This may due to how we fuse representations and initialize our model.

- As mentioned above, our joint model tend to have a bias on *unanswerable* questions. This is confirmed in Table 3 where we see the Recall score for unanswerable questions increase for all our updated models. To alleviate this, we might need **finer granularity** of types and additional information.

7.2 Comparison on Question-Type Classifier Structure

As shown in Table 4, LXMERT classifier achieves the highest accuracy and LSTM+CNN classifier outperforms BERT classifier by 7%. This result meets our expectation, since BERT only takes language inputs, while LXMERT and LSTM+CNN take both image and languages inputs to predict question type. We conclude that for a large number of samples, question text alone is insufficient and it's necessary to incorporate image information for question type classification.

However, though we see this performance discrepancy if we train the classifier alone, after we load the pretrained weights and finetune along with main VQA model, we see relatively the same performance for LSTM+CNN and BERT classifiers.

Classifier	Accuracy
LXMERT	0.73
LSTM+CNN	0.70
BERT+BiLSTM	0.63
BERT	0.62

Table 4: Question type prediction accuracy by training our classifier alone

7.3 Ablation on Fusion Method

For BERT classifier, late fusion yields the best performance and there is no clear improvement for injecting type information before the Cross-Modality Encoder. One intuitive explanation for this is we might suffer from more bias for incorrect type prediction and the model tend to overfit faster.

7.4 Ablation on Multi-task Loss

Without multi-task loss our classifier is completely separated from the main VQA model, and its parameters are not updated. Our experiment results have shown that this is undesirable. As in the middle block of Table 2, we get negative results both for hard-masking or hidden layers fusion. Therefore, we conclude that multi-task loss is necessary.

For our multi-task loss, we have α, β to scale the cross entropy loss for question type classifier component, and the binary cross entropy loss for answer prediction. We reach highest accuracy for $\alpha = 0.5, \beta = 1.0$ without normalizing the loss, this is probably because more focus needs to be on the main task.

8 Conclusion and Future Work

In this project, we conducted a comprehensive error analysis on the baseline model, LXMERT's performance on predicting answers to visual questions on Vizwiz dataset. Based on our analysis, we proposed a pipeline approach which tries to inject question type information into the main VQA model. We experimented with different classifier models and fusion methods. Our results have shown that training LXMERT with the question type classification and answer prediction task jointly achieved the highest accuracy. However, several results are unexpected and we also performed analysis on the possible reasons behind them. Based on our study, potential future directions include: understanding why our model having such a large bias on the type "*unanswerable*" and finding ways to prevent it; build upon current architecture, experiment with more fusion methods and locations, and tune the hyper-parameters more carefully; try to generate finer grained answer types by possibly leveraging Named Entity Recognition model; adapt our answer-type augmented setting on other VL models to see if there would be further improvement.

References

Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C. Lawrence Zitnick, Dhruv Batra, and Devi Parikh. 2016. Vqa: Visual question answering.

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering.

Hedi Ben-younes, Rémi Cadene, Matthieu Cord, and Nicolas Thome. 2017. Mutan: Multimodal tucker fusion for visual question answering.

Jonathan Chung and Thomas Delteil. 2019. A computationally efficient pipeline approach to full page offline handwritten text recognition. In *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, volume 5, pages 35–40.

Jean-Benoit Delbrouck, Noé Tits, Mathilde Brousmiche, and Stéphane Dupont. 2020. A transformer-based joint-encoding for emotion recognition and sentiment analysis. *Second Grand-Challenge and Workshop on Multimodal Language (Challenge-HML)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Daniele Di Mitri, Jan Schneider, MM Specht, and HJ Drachsler. 2019. Multimodal pipeline: a generic approach for handling multimodal data for supporting learning. In *First workshop on AI-based Multimodal Analytics for Understanding Human Learning in Real-world Educational Contexts*.

Jeff Donahue, Lisa Anne Hendricks, Marcus Rohrbach, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko, and Trevor Darrell. 2016. Long-term recurrent convolutional networks for visual recognition and description.

Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. 2016. Multimodal compact bilinear pooling for visual question answering and visual grounding.

Peng Gao, Haoxuan You, Zhanpeng Zhang, Xiaogang Wang, and Hongsheng Li. 2019. Multi-modality latent interaction network for visual question answering.

Danna Gurari, Qing Li, Abigale J. Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P. Bigham. 2018. Vizwiz grand challenge: Answering visual questions from blind people.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition.

Vahid Kazemi and Ali Elqursh. 2017. Show, ask, attend, and answer: A strong baseline for visual question answering.

Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. 2018. Bilinear attention networks.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. Visualbert: A simple and performant baseline for vision and language.

Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. 2015. Microsoft coco: Common objects in context.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks.

Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. 2017. Hierarchical question-image co-attention for visual question answering.

Hyeonseob Nam, Jung-Woo Ha, and Jeonghee Kim. 2017. Dual attention networks for multimodal reasoning and matching.

Duy-Kien Nguyen and Takayuki Okatani. 2018. Improved fusion of visual and language representations by dense symmetric co-attention for visual question answering.

Gao Peng, Zhengkai Jiang, Haoxuan You, Pan Lu, Steven Hoi, Xiaogang Wang, and Hongsheng Li. 2019. Dynamic fusion with intra- and inter- modality attention flow for visual question answering.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2016. Faster r-cnn: Towards real-time object detection with region proposal networks.

Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. 2019. Towards vqa models that can read.

Shuyang Sun, Jiangmiao Pang, Jianping Shi, Shuai Yi, and Wanli Ouyang. 2019. Fishnet: A versatile backbone for image, region, and pixel level prediction.

Hao Tan and Mohit Bansal. 2019. LXMERT: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5100–5111, Hong Kong, China. Association for Computational Linguistics.

Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2015. Learning spatiotemporal features with 3d convolutional networks.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Xuguang Wang, Linjun Shou, Ming Gong, Nan Duan, and Daxin Jiang. 2020. No answer is better than wrong answer: A reflection model for document level machine reading comprehension.

Huijuan Xu and Kate Saenko. 2016. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. 2016. Show, attend and tell: Neural image caption generation with visual attention.

Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. 2019. Deep modular co-attention networks for visual question answering.

Zhou Yu, Jun Yu, Chenchao Xiang, Zhou Zhao, Qi Tian, and Dacheng Tao. 2018. Rethinking diversified and discriminative proposal generation for visual grounding.

Yukun Zhu, Ryan Kiros, Richard Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books.