Computer Science

Capstone Report - Spring 2022

Visual Question Answering on VizWiz-VQA

Ze Qian,
Yang Liu,
Haochen Li

supervised by
Li Guo, Yik-Cheung (Wilson) Tam

Preface

In this project, we proposed a UNITER-based pre-training model combined with Optical Character Recognition for Visual Question Answering task to help blind people. This project aims to determine the answerability of visual questions, and then answer the answerable visual questions based on image features or texts in the image.

Acknowledgements

We'd like to express our deepest gratitude to Prof. Li Guo and Prof. Yik-Cheung (Wilson) Tam, our supervisors, for their professional guidance, enthusiastic encouragement and useful critiques of this work. We would also like to thank our parents and friends for their support and encouragement during our study, especially in this special quarantine time in Shanghai.

Abstract

Visual Question Answering (VQA) has showed great potential in helping disabled people in their daily lives, and various approaches has been proposed to achieve VQA through machine learning approaches. In this project, we mainly develop a UNITER-based pre-trained model to solve VQA problem on VizWiz VQA, a dataset containing visual questions asked by blind people. Our model aims to both determine whether a visual question is answerable and predict the possible answer for the answerable questions. In addition, we also integrate Optical Character Recognition (OCR) with our model to answer some questions with textual information directly extracted from the image. Our model has achieved a 65.8% average accuracy over the training set and a 49.7% average accuracy over the validation set. Besides, our model has shown high accuracy on the prediction of answerability, with 82.4% and 79.7% accuracy on unanswerable questions over training set and validation set, respectively.

Keywords

Visual Question Answering; Machine Learning; Pre-Trained Model; Optical Character Recognition;

Contents

1 Introduction

Visual Question Answering (VQA) is a task both involved in Computer Vision and Natural Language Processing. The definition of VQA is as follow. A VQA system takes an image and a free-form, open-ended, natural-language question about the image as its input, and then outputs a natural-language answer of the input question according to the input image [1]. Our project aims to propose a system to help blind people interact with visual scenes based on VQA. This system aims to take questions on visual scenes (images) as input, and then respond a corresponding answer to the blind people.

The dataset we will use is a dataset provided by the VizWiz Challenge, called VizWiz-VQA [2]. It originates from a natural visual question answering setting where blind people each took an image and recorded a spoken question about it, together with 10 crowd-sourced answers per visual question. Specially, all the questions in this dataset are asked by blind people. Thus, some of these questions may not be answerable in their corresponding images, since blind people cannot directly obtain information from these images. The answerability problem can be another challenge apart from obtaining results from visual images.

We observe that currently there are many powerful models using pre-training on a variety of tasks, e.g., Image Caption, Image-Text matching, Object Detection, etc. It is highly possible that these models can also be useful for VQA tasks with some revisions. In this project, we will mainly integrate some of the current pre-trained models like BERT, GPT and do some fine-tuning based on the VizWiz-VQA dataset.

Besides, we also noticed that the answer to some visual questions may directly originate from texts in the image rather than image features. This requires further recognition of textual information in images, which is a suitable scenario to implement Optical Character Recognition (OCR)[3]. For this challenge, we will integrate the results of OCR with our pre-training model to directly extract some textual information from the input image.

The rest of this paper is organized in the following order. Section 2 briefly reviews the related literature about various pre-training models. Section 3 illustrates the dataset, labels, and models used in our work. Section 4 displays our experimental results and findings. Section 5 evaluates and discusses the results of our experiment, and section 6 draws a conclusion to this work.

2 Related Work

VQA requires the understanding of visual contents, language semantics, and cross-modal alignments and relationships. Plenty of works have been proposed to develop backbone models to obtain better representations for images and texts. Most of these works provide us with powerful pre-trained models training on large datasets. Below we will briefly introduce several major models in these works, including BERT and CLIP.

2.1 BERT

The structure of the previous pre-training model is limited by the unidirectional language model (processing input left-to-right or right-to-left) [4, 5]. This can limit the representation ability of the language model, since it can only obtain unidirectional context information instead of bidirectional full context information. Bidirectional Encoder Representations from Transformers (BERT) [6] is proposed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers.

BERT uses Masked Language Model (MLM) for pre-training and uses a deep bidirectional Transformer component [7] to build the entire model. In this way, BERT can finally generate a deep bidirectional language representation that can fuse left and right context information without using RNN or LSTM. Besides, BERT models with the same pre-trained architecture can be fine-tuned onto different tasks to serve different downstream tasks, e.g., Question Answering, Natural Language Inference, etc.

2.2 CLIP

Predicting a fixed set of predetermined object categories may restrict the pretrained model's generality and usability when they meet unseen labels. Therefore, learning directly from raw text about images is a promising alternative. Alec Radford et al.[8] proposed Contrastive Language-Image Pretraining (CLIP) model training on 400 million (image, text) pairs collected from the internet and obtain the SOTA image representations. This pre-trained model's zero-shot performance can catch up with SOTA supervised model on some datasets.

There are also some works focus on developing Cross-Modality architectures to solve the VQA problems. Below we will simply show 3 of these architectures, which are LXMERT, VU-BERT, and UNITER.

2.3 LXMERT

Tan et al.[9] proposed the Learning Cross-Modality Encoder Representations from Transformers (LXMERT) framework to learn vision-and-language connections. LXMERT combines the feature of image and text by Cross-Modality Encoder, using cross-attention. They pre-train the model with large amounts of image-and-sentence pairs, via five diverse pre-training tasks: MLM, masked object prediction, cross-modality matching, and image question answering. LXMERT achieved the state-of-the-art results on two visual question answering datasets (i.e., VQA and GQA).

2.4 VU-BERT

Ye et al. [10] presented a simpler architecture based on BERT to realize visual dialog, called VU-BERT. Instead of extracting object-level features as vision features in [9], this architecture uses ViT [11] to extract patch embedding from the input images, which is faster than object detection methods. The image embedding and dialog embedding are concatenated together as the input of a BERT-like self-attention Transformer. VU-BERT also uses MLM as one of its task. MLM task can drive the VU-BERT network to model dependencies among dialog contents and image contents.

2.5 UNITER

UNiversal Image-TExt Representation Learning (UNITER)[12] is also an architecture based on BERT that deals with Vision-and-Language (V+L) tasks. Different from previous work that applies joint random masking, UNITER uses conditional masking on pre-training tasks. It uses 4 pre-train tasks including Image-Text Matching and Word-Region Alignment and has a state of the art performance for 6 V+L tasks including VQA over 9 datasets.

3 Solution

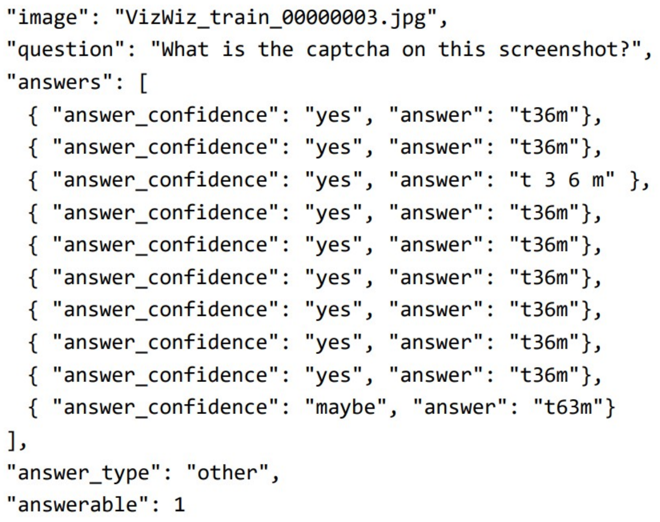
3.1 Dataset

As is mentioned before, our model aims to address the VQA problem on the VizWiz-VQA dataset[1], which consists of questions and their answers on given images. In this subsection, we will briefly introduce this dataset and some of its characteristics.

[1] https://vizwiz.org/tasks-and-datasets/vqa/

3.1.1 Data Format

The format of a visual question in VizWiz-VQA dataset is shown in Figure 1 below. We can observe that each visual question corresponds to a specific image ID, and 10 answers along with additional information about question / answers are also attached.



```
"image": "VizWiz_train_00000003.jpg",
"question": "What is the captcha on this screenshot?",
"answers": [
  { "answer_confidence": "yes", "answer": "t36m"},
  { "answer_confidence": "yes", "answer": "t36m"},
  { "answer_confidence": "yes", "answer": "t 3 6 m" },
  { "answer_confidence": "yes", "answer": "t36m"},
  { "answer_confidence": "yes", "answer": "t36m"},
  { "answer_confidence": "yes", "answer": "t36m"},
  { "answer_confidence": "yes", "answer": "t36m"},
  { "answer_confidence": "yes", "answer": "t36m"},
  { "answer_confidence": "yes", "answer": "t36m"},
  { "answer_confidence": "maybe", "answer": "t63m"}
],
"answer_type": "other",
"answerable": 1
```

Figure 1: A sample visual question in VizWiz-VQA dataset.

VizWiz-VQA dataset contains some question that cannot be answered due to inappropriate questions, unsuitable image, or low resolution image. For each visual question, an answerable field shows whether this question can be answered, and an answer_type field further indicates the type of this question. Table 1 below shows the values of fields answerable and answer_type.

answer_type	**answerable**	**meaning**
unanswerable	0	This question cannot be answered.
number	1	This question takes a number as its answer.
yes/no	1	This question takes “yes" or “no" as its answer.
other	1	This question’s answer is neither a number nor “yes"/“no".

Table 1: The values of fields answerable and answer_type in VizWiz-VQA dataset.

For an answerable question, its standard answer is the most frequent answer among the 10 given answers. In the sample visual question shown in Figure 1, the standard answer should be t36m, since it has the highest frequency (0.8) among all the answers. Standard answer can be viewed as the correct result to evaluate the prediction outputs of machine learning models. Besides, each answer also has a answer_confidence field to indicate the confidence of the answerer. This field is not used in our model, since we did not assume the impact of answer confidence on the final result.

3.1.2 Statistics on Labeled Questions

VizWiz-VQA dataset is divided into a training set and a validation set. The statistics of the 2 sets according to answer types are shown in Table 2 below.

	unanswerable	number	yes/no	other	Total
training	5,532	301	957	13,733	20,523
validation	1,385	48	195	2,691	4,319

Table 2: The statistics of training set and validation set in VizWiz-VQA dataset.

Consider that answer type other may contain more kinds of answers compared with number and yes/no, we also counted the number of unique answers to “other" type questions in both training set and validation set. There are 36,335 and 9,367 unique answers to “other" type questions in the training set and validation set, respectively. Additionally, there are 2,836 unique answers in training set that appeared in the validation set. In validation set, these answers have a total frequency of 14,622, which covers approximately 60% of the total answers in the validation set.

3.2 Image Label Selection

Since our model is based on classification on input image-question pairs, we need to generate target values (labels) corresponding to the possible answers to the questions. Consider that some answers may result from the texts on the input image, we also need to create several OCR labels indicating the corresponding OCR box on the image.

We generally divide label selection into two stages. In the first stage, we focused on selecting the answers already provided in the training set itself. In the second stage, we added OCR results for each picture. If the answer to a question is similar to the OCR result, we will replace the image label of the original answer with an OCR label. The details of the 2 stages will be given in the following subsections.

3.2.1 Labels Without OCR

In the first stage, we use different methods to select image labels from answers in the training set. For this stage, we have tried 4 different methods of label selection as follows. For the purpose of illustration, we will use the label selection results of VizWiz_train_00000003.jpg as examples for each method. The details of this picture are shown in Table 3 below.

The first method counted the word frequency of the 10 answers of each picture. For a given picture, its answer with the highest word frequency is used as its image label. For the example

Image	VizWiz_train_00000003.jpg
Question	What is the captcha on this screenshot?
Answer Type	other
Answer	**# of appearance**
t36m	8
t63m	1
t 3 6 m	1

Table 3: The details of VizWiz_train_00000003.jpg.

given in Table 3, t36m with the highest word frequency (0.8) is the label of this image. However, this method not only makes the image label difficult to understand, but it also makes the image labels of each picture almost unique. This could increase the model complexity due to excessive target values. As a result, we abandoned this method before it was implemented.

In the second method, we tried to minimize the number of labels based on the results of method 1. We counted the occurrence of each individual answer selected in method 1, and all the answers that appear only in 1 question are classified as Out-Of Vocabulary (OOV) words. Under this method, the label of the example given in Table 3 will be OOV rather than t36m, since the answer t36m only appear in this single picture. With this method, there are 1558 labels (including OOV) selected.

In the third method, we tried a different cutoff method on labels compared with method 2. We used the answers with the top 3000 frequencies as image labels, and the rest of the answers are all classified into the OOV category. With this method, the label of the example given in Table 3 will be t36m, since the frequency of t36m is among the top 3000 frequencies among all labels. With this method, there are 3001 labels (including OOV).

Previous 3 methods use hard labeling on inputs, which means each input only corresponds to a single label with probability 1.0. The fourth method added soft labeling on the basis of method 3. For each question, we use each of its answer as a soft label, with the frequency of this answer as its probability. The answers not within the top-3000 frequencies are still labeled as OOV. For the example given in Table 3, its soft label under method 4 will be {t36m: 0.8, OOV: 0.2}, since the answer t63m and t 3 6 m is not within the top-3000 frequency vocabulary and are both marked as OOV. This method also generate 3001 labels (including OOV) selected.

3.2.2 Labels With OCR

In VizWiz-VQA dataset, we believe that many questions belonging to various categories can be solved through Optical Character Recognition (OCR). Therefore, we conducted OCR on all pictures to get texts within the image, and at this stage we can judge which questions can be answered through the OCR results of its corresponding image.

In order to analyze whether an answer of a question can be obtained through OCR, we should compare the answer with the corresponding OCR results. We used edit distance to calculate the similarity between an answer and an OCR-obtained text. Consider that the length of strings may affect the largest possible value of edit distance, we divided edit distance with the length of the longer string to normalize it into [0,1]:

$$NormEditDistance = \frac{EditDistance}{max(len(ans), len(ocr))}$$

We used normalized edit distance at both character level and lexical level to evaluate the similarity between an answer ans and an OCR text ocr. For character level calculation, we directly calculate the normalized edit distance between ans and ocr. For lexical level calculation, we combine edit distance with word alignment to match individual words in ans and ocr.

For each word $word_{ans}$ in ans, word alignment requires to calculate its normalized edit distance with each remaining unmatched word in ocr, and find the word with the minimum edit distance $word_{ocr}$ in ocr. If this edit distance is less than a given threshold, then $word_{ans}$ can be considered to match $word_{ocr}$. We align words with consideration of their sequence. The matching of a word in ans must start from the next word of the last matched word in ocr. After aligning all the words, we also normalize the number of alignment pairs by dividing it with the larger word count of ans and ocr:

$$NormalizedAlign = \frac{Align}{max(words(ans), words(ocr))}$$

The alignment between 2 example strings are shown in Figure 2.

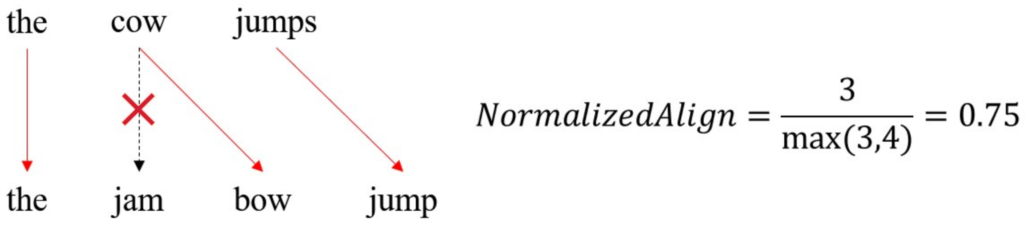


Figure 2: An example of word alignment.

By controlling the threshold of normalized edit distance and normalized alignment score, we could find out answers that are highly similar to its corresponding OCR result to add OCR labels. With the OCR processing mentioned above, we also take 2 different methods in this stage to assign OCR labels to the answers of each question based on the method 4 (soft labeling + top-3000 frequency cutoff) in stage 1.

In the first method, we directly calculated the edit distance and alignment score between all answers and OCR texts for each question. If the edit distance is less than 0.3 or the alignment score is greater than 0.6, the answer is considered to be represented by a corresponding OCR result. The OCR boxes are sorted by positions from left to right, top to bottom, represented with labels OCR1, OCR2, and OCR3. The answers that match a certain OCR box will be substituted with the label of the corresponding OCR box. For the example in Table 3, the OCR result of VizWiz_train_00000003.jpg is 136m, which has a normalized edit distance 0.25 and a normalized alignment score 1.0. Thus, the original label {t36m: 0.8, OOV: 0.2} will be changed into: {OCR1: 0.8, OOV: 0.2}.

In method 1, there are numerous problems with type unanswerable, and this can lead to a very uneven distribution of data. In addition, we also observed that there are many "unanswerable" answers in other types of problems, and these answers are prone to be affected by the uneven data distribution. Therefore, in our second method, we removed all the unanswerable problems in the original data and used method 1 onto the remaining problems.

3.3 VQA model

3.3.1 Baseline UNITER

Our baseline model is original UNITER, as shown in Figure 3. The input of UNITER model is the concatenate of image features and the whole sentence of the question. UNITER's structure is mainly based on BERT. The same as BERT, UNITER use the [CLS] position vector to do the classification.

For each image, we use pre-trained faster-RCNN to extract 36 image patches and for each patch, we save its 2048-dimensions feature vector (pooled ROI features) for UNITER fine-tuning.

3.3.2 UNITER with OCR

We try to improve our baseline UNITER model with OCR information. The same as image and question feature, we add OCR features by concatenating the features, as shown in figure 4.

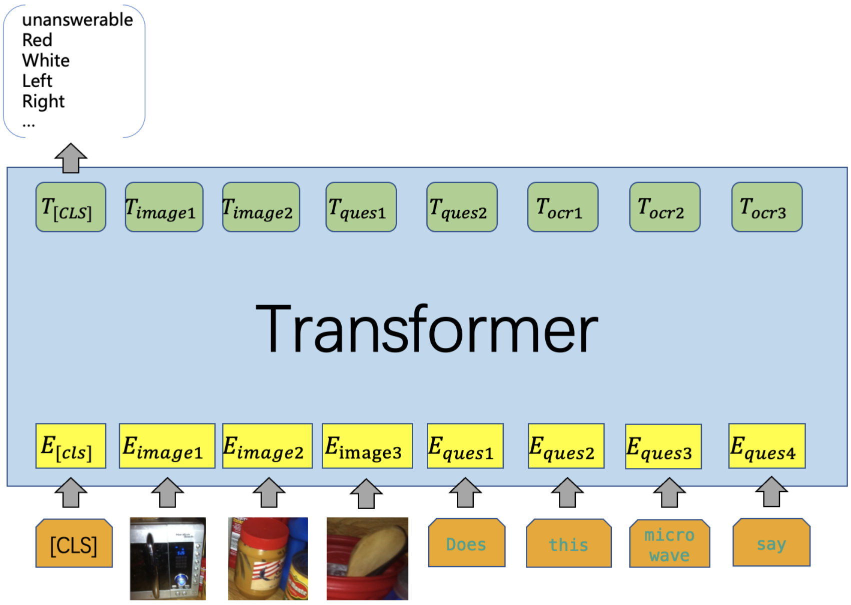


Figure 3: Baseline UNITER VQA model

Inspired by OCR-VQA[13], we use OCR text result's BERT [CLS] feature as part of OCR feature. Like UNITER, We also encode the position feature via a 8-dimensional vector[2].

Since the average number of OCR boxes for each image is about 2.6, we add 3 OCR blocks in our model to keep a balance between covering the true answer in and having less disturbance. If there are more than 3 detected OCR blocks, then we only take largest (area-wise) 3 blocks (for training dataset, we must include the correct OCR box in). If there are less than 3 detected OCR blocks, then each of the remaining blocks is represented by a zero vector.

4 Experimental Results

In this experiment, we mainly divide the experiment into two stages: using OCR and not using OCR. For these two stages, we also use different methods to improve the learning efficiency and accuracy of the model. Because we will divide the whole experimental process into two parts according to whether OCR is used or not, and discuss the results of several experiments respectively.

4.1 Experiment tools

We use the following tools to realize this experiment:

[2] $[i, x1, y1, x2, y2, w, h, w \times h]$ (sequence of box, normalized top/left/bottom/right coordinates, width, height, and area.)

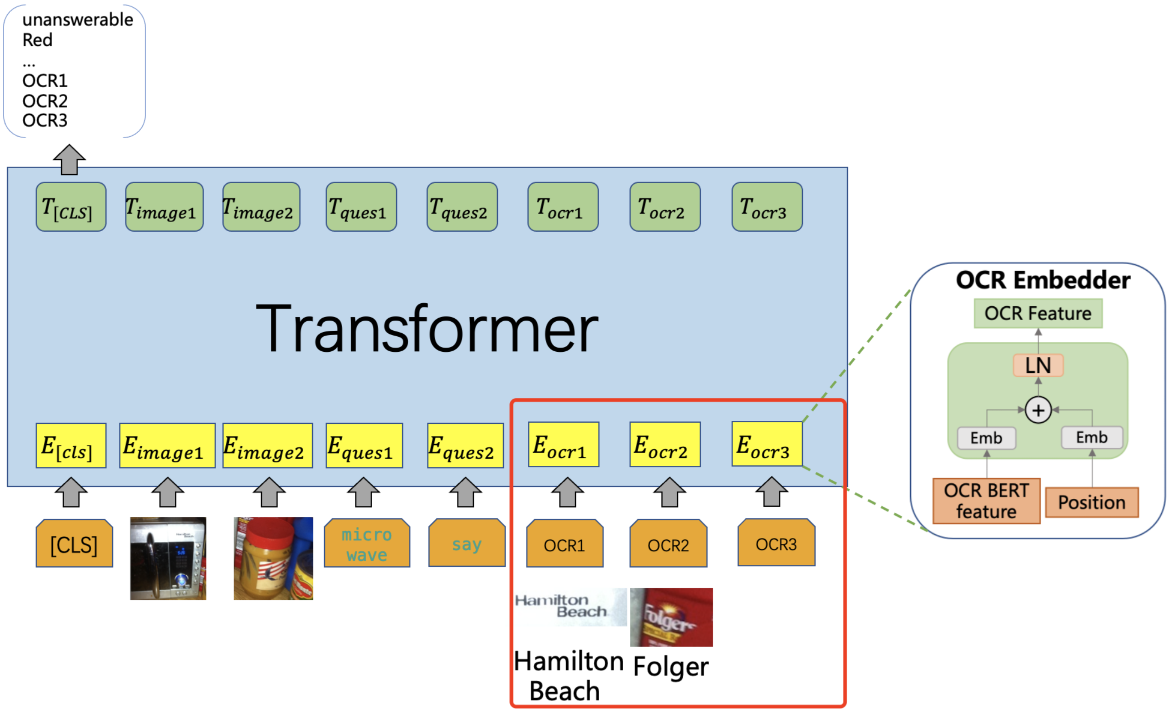


Figure 4: UNITER with OCR Embedder

- For image feature extraction, we use faster-RCNN based on detectron2.
- For UNITER model, We implement it with Python deep learning framework PyTorch. [3].
- For OCR part, we use EasyOCR to detect the OCR boxes and recognize the text.
- Since many images in VizWiz-VQA are in poor visibility, OCR system may results in misspell. So we use SymSpell to correct the OCR result with $editdistance = 2$.

4.2 Data tables

We divided the whole experimental process into two parts to discuss their results.

4.2.1 Result without OCR part

In this part, we mainly discuss the accuracy of the method without OCR. We implemented three methods to obtain image label, which we have mentioned in 3.2.1.

In the first method, we set all the answers appeared more than once as labels and other answers as OOV. We used the most frequent answer among the 10 answers to a question as its final label.

[3] Code is available at https://github.com/qz701731tby/VizWiz-VQA

In the second method, we set the top 3000 frequency answers as labels and others as OOV. We also used hard label to show the most frequency answer for a question.

In the third way, we set the top 3000 frequency answers as labels and others as OOV. Instead of hard labeling, we used soft label in this method to show the proportion of each unique answer for a given question.

Table 5 shows the experimental results on validation set of our model with each of the 3 label selection strategies.

Accuracy without OCR			
NO.	Label num	hard/soft label	Acc(%)
1	1558	soft label	49.15
2	3001	hard label	48.95
3	3001	soft label	47.97

Table 4: The accuracy of different ways on `Selecting Image Label without OCR`.

We noticed that when using the first and second methods, the model's predictions for most of the images were unanswerable. Since unanswerable pictures account for a high proportion of answers, predicting most pictures as unanswerable will lead to an increase in the correct rate, but this does not reflect the role of the model, so we finally decided to use the third method, and compare and analyze the accuracy of the third method in different answer type categories.

Accuracy without OCR					
Dataset	Yes/no(%)	other(%)	number(%)	unanswerable(%)	average(%)
Val	49.86	28.12	21.68	85.49	48.86

Table 5: The accuracy of the soft-labeling method on `Selecting Image Label without OCR`.

4.2.2 Result with OCR part

In this part, we mainly discuss the accuracy of the method without OCR. As is mentioned in 3.2.2, we adopted 2 strategies to generate OCR labels based on the last method adopted (top-3000 frequency cutoff + soft labeling) in stage 1.

The first method calculated the edit distance and word alignment between an answer and an OCR text. If these 2 values match a given threshold (edit distance ≤ 0.3 or word alignment ≥ 0.6), then this answer will be replaced by a corresponding OCR tag. Based on the first method, the second method further removes questions with type unanswerable when labeling OCR tags to avoid uneven data distribution.

Table 6 shows the experimental results on validation set of our model with OCR label selection strategies.

Accuracy with OCR						
Dataset	Yes/no(%)	other(%)	number(%)	unanswerable(%)	average(%)	OCR(%)
Train	73.9	58.0	64.3	82.4	65.8	46.8
Val	50.0	33.3	30.6	79.7	49.7	3.2

Table 6: The accuracy of the OCR labeling strategy over training and validation datasets.

5 Discussion

We can see that there is little difference in the overall accuracy of the data in the part that does not consider OCR. Subdivided into different answer types, we find that the accuracy of data prediction of unanswerable class is very high, reaching 90%, while the accuracy of data of other categories is relatively low. Therefore, we believe that this difference is due to the fact that the characteristics of some pictures have not been learned because the sample size is too small, As a result, there is no way to accurately analyze the different characteristics of different pictures. At the same time, because the overall sample distribution is uneven, the data of unanswerable class is much more than that of other categories, so the average accuracy can not show the overall performance very well.

In terms of adding OOV, we found that the accuracy of data containing OOV is higher than that of data without OOV. We think this is also because the image features are not well learned, so the accuracy of including OOV tags will be partially improved in the case of fuzzy features. In fact, it is also true that only a small part of all answer labels have appeared many times. For this kind of label, we can learn its characteristics, but for most labels that have only appeared once or several times, we will have great difficulties in matching them with the characteristics of the picture, which also leads to the result that the final accuracy is not very high. besides. Since we can't answer the actual VQA questions with OOV, we may need to improve the sample size in the next research, so as to improve the ability of image label and image feature matching.

Considering that many problems need to recognize characters or numbers on the image as their answers, we choose to add OCR data. It can be seen that in the training set that the accuracy of data in each category has been relatively high. In addition, since OCR can clearly reflect the picture information, we don't need to use the relatively vague OOV as the label of the picture. However, due to the different performance of OCR in identifying pictures with different clarity, some OCR results cannot be used as a reference. Therefore, for some pictures that should use OCR to obtain image label, we may not be able to connect their answers with OCR results, which also leads to the model's inability to accurately learn which problems should use OCR. In the future work, we hope to find a better OCR model pre-trained on product dataset to improve the accuracy of OCR, so as to accurately mark the pictures that need to use OCR in the training data set, help the model better learn the relevant characteristics of the pictures and problems that need to use OCR.

Here is some points that can explain the low accuracy in validation's OCR labels compared with the accuracy on training set. First, our OCR label selection are based on automatic process. We have to set a high threshold to make sure the correctness of our OCR label, which leads to low recall of all the correct OCR labels. According to a VQA data analysis paper, about 45% of questions in VizWiz-VQA ought to be solved by OCR[14]. In our experiment, only about 10% questions have OCR label through our process, which means there are still large proportion of OCR answerable image are not included. This will lead to the similar distribution of OCR data and non-OCR data. Second, we analysis the distribution of train and val answer for OCR answerable questions and find that there is a big difference between train and val. From this aspect, We can add some common features, like NER tag or whether OCR text is a branch name or not to strengthen the connection between answer and question.

6 Conclusion

This paper mainly focused on realizing VQA using UNITER-based pre-trained model over VizWiz-VQA dataset. For data pre-processing and label selection, we applied different strategies in 2 stages to select labels for prediction. For model design, we fine-tune UNITER based on the VQA task, and we also integrate OCR text results with our model to answer questions involving text extraction from images. Our experiment shows that our model achieves a 65.8% average accuracy over the training set and a 49.7% average accuracy over the validation set. Besides, our model has shown high accuracy on prediction of answerability, with 82.4% and 79.7% accuracy on unanswerable questions over training set and validation set, respectively.

However, this model still has some shortcomings. Although it can achieve 46.8% accuracy of OCR predictions over training set, but the accuracy of OCR predictions over validation set is 3.2% due to the poor image clarity and the lack of ways to determine questions requiring OCR.

Besides, the limited size of training data also affects the performance of our model. The features and labels of the image-question pairs in validation set are not well covered during our training process, resulting in the relatively low performance over the validation set.

As for the future work, we plan to manually do OCR labelling task on all the image instead of automatically. By this way, we can have a more accurate labelled dataset. Also, we will explore more features that we can use to enhance our OCR system, like the NER tag of OCR text etc.. Besides, we will also research on methods to determine whether a question should be solved using OCR. Hopefully, these future improvements could boost the performance of our model over both normal questions and OCR-involved questions.

References

[1] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, "Vqa: Visual question answering," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[2] D. Gurari, Q. Li, A. J. Stangl, A. Guo, C. Lin, K. Grauman, J. Luo, and J. P. Bigham, "Vizwiz grand challenge: Answering visual questions from blind people," *CoRR*, vol. abs/1802.08218, 2018. [Online]. Available: http://arxiv.org/abs/1802.08218

[3] J. Memon, M. Sami, R. A. Khan, and M. Uddin, "Handwritten optical character recognition (ocr): A comprehensive systematic literature review (slr)," *IEEE Access*, vol. 8, pp. 142 642–142 668, 2020.

[4] A. Mnih and G. E. Hinton, "A scalable hierarchical distributed language model," *Advances in neural information processing systems*, vol. 21, 2008.

[5] M. E. Peters, W. Ammar, C. Bhagavatula, and R. Power, "Semi-supervised sequence tagging with bidirectional language models," *arXiv preprint arXiv:1705.00108*, 2017.

[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. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[8] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," *CoRR*, vol. abs/2103.00020, 2021. [Online]. Available: https://arxiv.org/abs/2103.00020

[9] H. Tan and M. Bansal, "LXMERT: learning cross-modality encoder representations from transformers," *CoRR*, vol. abs/1908.07490, 2019. [Online]. Available: http://arxiv.org/abs/1908.07490

[10] T. Ye, S. Si, J. Wang, R. Wang, N. Cheng, and J. Xiao, "Vu-bert: A unified framework for visual dialog," *arXiv preprint arXiv:2202.10787*, 2022.

[11] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[12] Y. Chen, L. Li, L. Yu, A. E. Kholy, F. Ahmed, Z. Gan, Y. Cheng, and J. Liu, "UNITER: learning universal image-text representations," *CoRR*, vol. abs/1909.11740, 2019. [Online]. Available: http://arxiv.org/abs/1909.11740

[13] A. Mishra, S. Shekhar, A. K. Singh, and A. Chakraborty, "Ocr-vqa: Visual question answering by reading text in images," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 2019, pp. 947–952.

[14] X. Zeng, Y. Wang, T.-Y. Chiu, N. Bhattacharya, and D. Gurari, "Vision skills needed to answer visual questions," *Proceedings of the ACM on Human-Computer Interaction*, vol. 4, no. CSCW2, pp. 1–31, oct 2020. [Online]. Available: https://doi.org/10.1145%2F3415220