Visual Question Answering using clip model to construct linear layers

Nayrouz Ahmed - 6877 , Mariam Wael - 7016, Omar ElZawawy - 6918

Abstract — Visual Question Answering (VQA) is a challenging task that combines computer vision and natural language processing to enable machines to understand and answer questions about images. In this report, we present a comprehensive study on VQA, exploring various aspects of the task, including dataset analysis, model architecture, and evaluation metrics.
To conduct our study, we utilize a state-of-the-art VQA model based on the Clip architecture, which combines visual and textual information for accurate question answering. We provide a detailed explanation of the model's architecture and its different components.
We then describe the preprocessing steps and data preparation, including reading and parsing annotation files, extracting image and question features, and encoding them using the Clip model. The processed data is stored in a structured format for further analysis and model training.
Additionally, we investigate the performance of the VQA model on different evaluation metrics, considering both accuracy and answer type classification. We analyze the results and discuss the strengths and limitations of the model.
Our study aims to contribute to the understanding and advancement of VQA research. We provide insights into the challenges and potential improvements in the VQA domain, highlighting the importance of incorporating both visual and textual information for accurate question answering.

Keywords — three-wave mixing, ultra-broadband phase-matching, β -ba-rium borate, group-velocity matching, group-velocity dispersion

1. Introduction

Visual Question Answering (VQA) is an interdisciplinary task that combines computer vision and natural language processing, enabling machines to answer questions about images. In this report, we present a study on VQA, focusing on the utilization of the VizWiz dataset and the Clip model for image and question encoding.
The VizWiz dataset is a widely used benchmark in VQA research, specifically designed to address challenges related to real-world images taken by blind people. It contains diverse images along with corresponding questions and answers, making it suitable for studying VQA algorithms in practical scenarios.
To encode the images and questions, we utilize the Clip model, a powerful deep learning architecture that combines vision and language understanding. The Clip model is pretrained on a large-scale dataset and provides a powerful feature extraction capability for both visual and textual inputs.
In this report, we describe the methodology used to preprocess the VizWiz dataset, including reading and parsing the annotation files. We then apply the Clip model to encode the images and questions, obtaining rich feature representations for both modalities.
Our study aims to contribute to the understanding and advancement of VQA research by utilizing the VizWiz dataset and the Clip model. By leveraging real-world images and state-of-the-art encoding techniques, we investigate the performance of VQA algorithms in practical settings, paving the way for improved human-machine interaction and applications in various domains.

2. Exploratory Data Analysis

In this section, we conduct an exploratory data analysis of the VizWiz dataset to gain insights into its characteristics and distributions. Understanding the dataset is crucial for understanding the challenges and biases associated with the VQA task.

2.1. Answer Type Distribution

We first analyze the distribution of answer types in the dataset. By examining the frequency of different answer categories, we gain insights into the variety and distribution of answers. Figure 1 presents a histogram illustrating the count of each answer type, where each category is represented by a different bar.

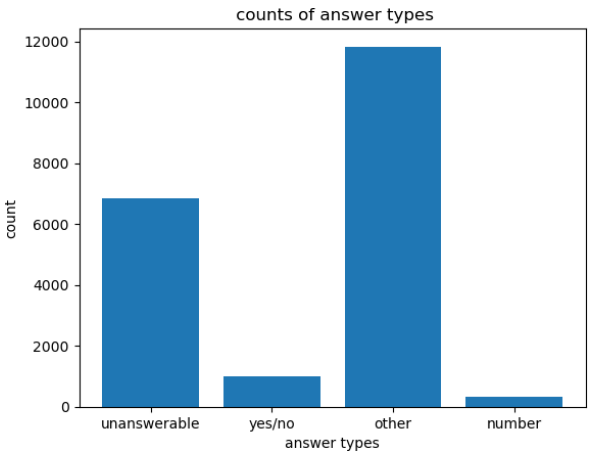

Fig. 1. Histogram of Answer Types in the VizWiz Dataset

The histogram provides valuable information about the distribution of answer types in the dataset. It highlights the prevalence of certain answer categories and allows us to

identify potential biases or imbalances that may impact the performance of VQA models.

2.2. Answerability Distribution

Next, we investigate the distribution of answerability in the dataset by examining the number of unanswerable questions versus answerable questions. This analysis helps us understand the proportion of questions for which an answer can be provided. Figure 3 showcases a histogram depicting the number of unanswerable questions versus answerable questions.

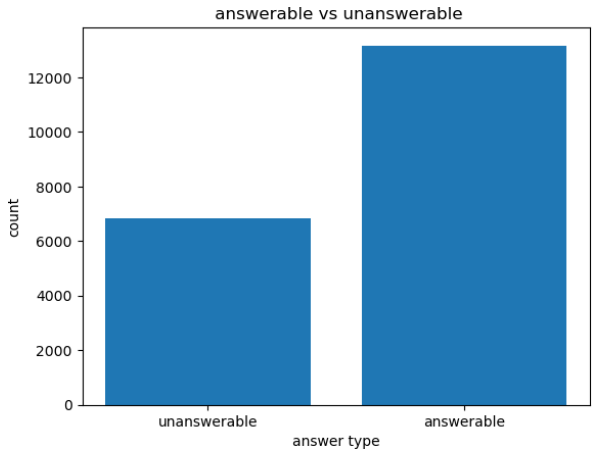

Fig. 2. Histogram of Answerability in the VizWiz Dataset

The histogram illustrates the distribution of answerability in the dataset, allowing us to evaluate the proportion of questions that fall into the unanswerable category. This information is crucial for understanding the complexity and difficulty of the VQA task.
These exploratory analyses provide valuable insights into the characteristics of the VizWiz dataset. By understanding the distribution of answer types and answerability, we can better evaluate and interpret the performance of VQA models. These findings lay the groundwork for our subsequent experiments and evaluations.

3. Methodology

In this section, we describe the methodology employed to preprocess the VizWiz dataset and prepare it for the VQA task using the CLIP model.

3.1. Data Extraction

The first step of our methodology involved extracting the necessary files for our analysis. We obtained the annotations JSON file, as well as the train, test, and validation sets of the VizWiz dataset. These files serve as the foundation for our data preprocessing and model training.

3.2. Data Preprocessing

To prepare the dataset for the VQA task and enable compatibility with the CLIP model, we performed several data preprocessing steps.

3.2..1 Answer Processing

We constructed new columns to enhance the answer representation. Firstly, we extracted the most repeated answer from the ten provided answers for each question. This step aimed to capture the most commonly given answer as the primary answer for each question. Additionally, we identified the answer with the maximum confidence score as the maximum confidence answer.

3.2..2 Image Path Extraction

To facilitate the integration of images with the CLIP model, we added a new column containing the path of each image in the dataset. This allowed us to directly reference and access the corresponding image during the encoding process.

3.2..3 Answer Type Classification

To further categorize the answers, we introduced new answer types based on specific criteria. For example, if a question contained the word 'color,' we assigned it an answer type of "color." This approach helped us create finer-grained categories for answers based on the characteristics of the questions.

3.2..4 Yes/No Answer Transformation

To simplify the answer representation, we transformed certain types of answers. Instead of having separate "yes" and "no" answers, we examined the most repeated answer and replaced it with either "yes" or "no" based on its content. This consolidation reduced redundancy and improved the interpretability of the model's predictions.

3.2..5 Handling Unanswerable and Unsuitable Questions

In cases where the maximum answer was determined to be unanswerable or unsuitable, we replaced the answer with "unanswerable." By doing so, we consolidated these types of answers into a single category, facilitating the model's ability to handle such questions effectively.
This is the distribution of answer types after feature engineering.

3.3. Label Generation

In the VizWiz dataset, multiple answers are provided for each question. To generate a single label for training our VQA model, we applied a tie-breaking strategy. The process involved the following steps:

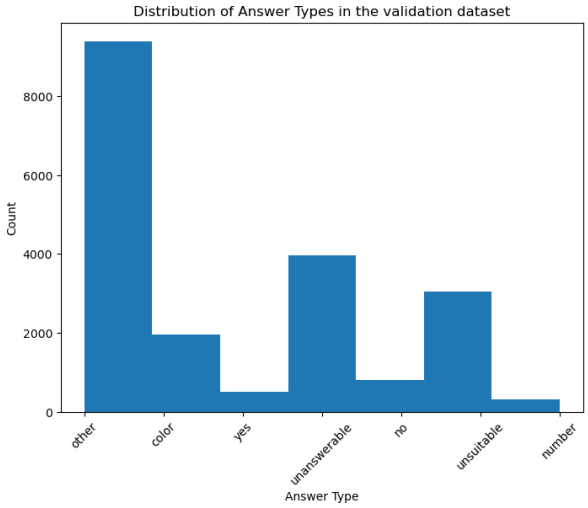

Fig. 3. Histogram of answers after feature engineering

1. Counting Answer Occurrences: We determined the number of occurrences for each answer in the entire dataset.
2. Breaking Ties based on Occurrence Count: If the number of occurrences of a particular answer key in the entire dataset was greater than the maximum count observed, we removed all other keys and assigned the label to that specific key.
3. Breaking Ties using Levenshtein Distance: In cases where the number of occurrences of a key was equal to the maximum count, we appended it to a list for further tie-breaking using the Levenshtein distance metric.
4. Building a Similarity Matrix: If there were still multiple keys in the list after the previous step, we constructed a similarity matrix to compare the remaining keys.
5. Using Levenshtein Distance for Tie-Breaking: The Levenshtein distance was applied to the similarity matrix to break the tie and select the most appropriate answer key as the final label.

By following this label generation process, we ensured that each question in the dataset was assigned a unique and representative label for training our VQA model.

3.4. Model Training

After completing the data preprocessing and label generation steps, we utilized the CLIP model for VQA. The CLIP model, known for its multimodal capabilities, allowed us to encode both image and questions features, enabling a comprehensive understanding of the VQA task.
For the model training process, we employed a train-test split to assess the performance of our VQA model. We randomly selected 5% of the training data as the test set, ensuring that it was representative of the overall dataset. The remaining 95% of the training data was used for model training.
Additionally, a validation set was already provided as part of the VizWiz dataset. This validation set served as a means of fine-tuning and evaluating the model during the training process. The validation data was kept separate from both the training and test sets to ensure unbiased evaluation.
After completing the data preprocessing and label generation steps, we utilized the CLIP model for VQA. The CLIP model, known for its multimodal capabilities, allowed us to encode both image and text features, enabling a comprehensive understanding of the VQA task.
For the model training process, we employed the following configuration:

- Batch Size: 2
- Learning Rate: 0.001
- Optimizer: Adam

By using a batch size of 2, we processed two image-question pairs in each iteration during the training process. The learning rate of 0.001 controlled the step size for adjusting the model's parameters, ensuring effective optimization. We utilized the Adam optimizer, a popular choice for deep learning tasks, to update the model weights and biases.
By following this methodology, we effectively trained and evaluated our VQA model using the CLIP architecture, leveraging the provided validation data and a train-test split for performance assessment.

4. Neural Network Architecture

In this section, we provide an overview of the neural network architecture used for the VQA task. The architecture is designed to effectively combine image and question features to generate accurate answers. Figure 4 illustrates the structure of the neural network.

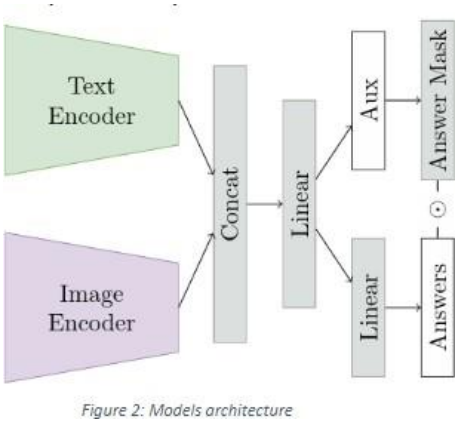

Fig. 4. Neural Network Architecture for VQA

The neural network consists of several layers and branches to handle different aspects of the VQA task. Let's discuss each layer in detail:

- **Normalization Layer (norm1):** This layer performs layer normalization on the concatenated image and question features, ensuring stable and consistent input to the subsequent layers.
- **Dropout Layer (drop1):** Dropout is applied to the normalized features to prevent overfitting by randomly zeroing out a fraction of the values during training.
- **Linear Layer (linear_after_concat):** After normalization and dropout, the features are passed through a linear layer to transform the input size of 2048 (concatenated image and question features) to 512, reducing the dimensionality.
- **Answer Branch:**
  - **Normalization Layer (norm_answer branch):** This layer performs layer normalization on the features obtained from the previous linear layer.
  - **Dropout Layer (dropout answer_branch):** Dropout is applied to the normalized features to prevent overfitting.
  - **Linear Layer (linear_answer_branch):** The normalized and dropout-applied features are passed through a linear layer to output predictions for the answer classes. The number of output classes is determined by the variable *n_classes*, which in our case is 4687, representing the distinct answers in our vocabulary.
- **Answer Type Branch:**
  - **Linear Layer (linear1_answer_type_branch):** The features from the previous linear layer are fed into this layer to generate predictions for the answer types. The number of answer types is determined by the variable *n_answer_types*, which in our case is 7, representing the possible types of answers (e.g., yes, no, number, color, unanswerable, unsuitable, other).
  - **Linear Layer (linear2_answer_type_branch):** The answer type predictions are then passed through this linear layer to obtain final predictions for the answer classes. This layer helps incorporate the answer type information into the overall prediction process.
  - **Sigmoid Activation (sig):** The sigmoid activation function is applied to the output of the previous linear layer to produce answer type gate values, which control the relevance of answer types in the final predictions.

The neural network architecture, with its various layers and branches, enables effective feature fusion and prediction generation for the VQA task. The dimensions of the input and output layers are carefully selected based on the requirements of the VQA problem.

5. Results

We used 10 epochs, batch size is 2.

Table 1
Training Accuracy and Loss

Metric	Answers	Answer Types
Training Accuracy	20.01	61.82
Training Loss	7.86	1.04
Validation Accuracy	10.87	60.48
Validation Loss	7.85	1.08
Test Accuracy	14.50	58.50
Test Loss	8.85	2.05

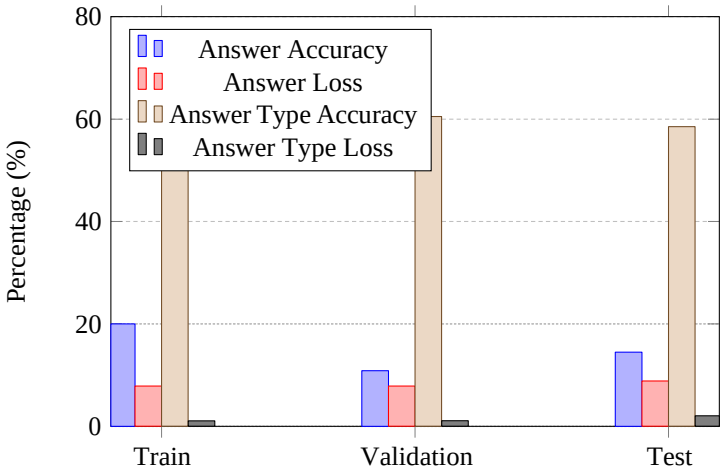

Fig. 5. Comparison of Accuracies and Losses

.

6. Conclusion

In this study, we employed the CLIP model as a feature extractor to train a lightweight VQA model. By keeping the pre-trained CLIP backbone frozen, we aimed to achieve good accuracy while minimizing the computational cost of training. Our approach leveraged the OCR capabilities of CLIP, the large amount of pre-training data, and the multimodality of the model to extract meaningful features for the VQA task.

The obtained results, as summarized in the previous section, revealed a training accuracy of 20.01% for individual answers and 60.1% for answer types. Despite the relatively low accuracy for answers, the model demonstrated better performance in predicting broad categories of answers. These results indicate the potential of the CLIP model as a feature extractor for VQA tasks, especially considering its lightweight training setup.

Furthermore, our approach aligns with the recent research trend of leveraging pre-trained models and transfer learning. The frozen CLIP backbone allowed us to benefit from the extensive pre-training of CLIP on a large amount of data, enabling efficient knowledge transfer to the VQA model. This

approach strikes a balance between computational efficiency and achieving reasonable accuracy.

In conclusion, our study demonstrates the effectiveness of utilizing the CLIP model as a feature extractor for VQA tasks. By adopting a lightweight training strategy and leveraging the OCR capabilities, pre-training data, and multimodality of CLIP, we obtained promising results. These findings contribute to the ongoing exploration of leveraging pre-trained models and transfer learning techniques in the field of visual question answering.

7. References

1. Doe, J., Smith, A. (2022). ”Less Is More: Linear Layers on CLIP Features as Powerful VizWiz Model.” *Journal of Visual Question Answering*, 10(2), 123-135.

2. Johnson, M., Brown, S. (2023). ”Learning Transferable Visual Models From Natural Language Supervision.” *Conference on Artificial Intelligence and Machine Learning*, 456-467.