11-711 Final Report: Adapting Multilingual VQA for Code-Mixed VQA

Sayali Kandarkar
Carnegie Mellon University
skandark@cs.cmu.edu

Vedant Bhasin
Carnegie Mellon University
vedantbhasin@cmu.edu

Mihir Bansal
Carnegie Mellon University
mihirban@cs.cmu.edu

Abstract

Code-mixing, the linguistic phenomenon of seamlessly blending two or more languages within a single utterance, poses a unique challenge in the domain of Visual Question Answering (VQA). Despite its prevalence in multilingual societies, there remains a conspicuous gap in VQA models tailored to address code-mixed language inputs. This paper addresses this gap by exploring the fine-tuning methodologies of the mBLIP (Multilingual BLoOMZ for Image Processing) model for code-mixed VQA tasks. Our study focuses on datasets, such as MaXM, xGQA, and the MCVQA dataset, aiming to evaluate and enhance the mBLIP model's performance across diverse linguistic variations. The absence of dedicated models for code-mixed VQA prompts our investigation into two fine-tuning approaches: one centered on code-mixed language generation and the other specifically tailored for code-mixed question-answering. In our experiments, we scrutinize the effectiveness of Low-Rank Adaptation (LoRA) in the context of fine-tuning Large Language Models (LLMs) for code-mixed language understanding. Our results showcase significant improvements of 59.17% in accuracy over the code-mixed VQA dataset, underscoring the importance of dedicated models for code-mixed VQA and offering insights for future advancements in multilingual natural language processing.

1 Introduction

Visual Question Answering (VQA) stands as a pivotal challenge in the realm of artificial intelligence, requiring models that seamlessly integrate visual understanding with natural language comprehension. The complexity of VQA is further amplified in multilingual and multicultural settings where code-mixing, the amalgamation of two or more languages within a single sentence or conversation, becomes prevalent. This phenomenon is particularly significant in regions with linguistic diversity, such as India, where questions are often posed in a fusion of languages.

Multilingual models, equipped with Large Language Models (LLMs), have showcased remarkable capabilities in handling diverse linguistic contexts. The mBLIP (Multilingual BLoOMZ for Image Processing) model is one such exemplar, renowned for its prowess in various language tasks. However, the intricacies of code-mixing pose unique challenges to VQA models, necessitating specialized adaptation strategies.

In this context, our study focuses on the fine-tuning of the mBLIP model for code-mixed VQA tasks. Leveraging datasets that reflect the multilingual and code-mixed nature of questions, we aim to enhance the model's performance across languages and linguistic variations. The exploration encompasses datasets such as MaXM and xGQA, allowing us to evaluate the model's adaptability in English, Hindi, and the amalgamated Hinglish language.

Our investigation includes two distinct fine-tuning approaches. The first centers on code-mixed language generation, hypothesizing that bolstering the model's understanding of code-mixed linguistic structures would enhance its performance. The second approach is tailored specifically for code-mixed question-answering, employing the MCVQA dataset. This targeted fine-tuning aims to hone the model's ability to decipher nuanced queries posed in a fusion of English and Hindi, a common occurrence in multilingual societies. Our exploration uses the concept of Low-Rank Adaptation (LoRA), a technique designed to fine-tune LLMs effectively. LoRA matrices play a pivotal role in capturing and adapting to the intricacies of code-mixed languages during the fine-tuning process. This approach enables the model to understand and generate responses in code-mixed languages, offering a nuanced perspective to bridge language gaps within VQA systems.

As we navigate through these fine-tuning strategies, our ultimate goal is to unravel the potential of mBLIP in addressing the challenges posed by code-mixing in VQA. The findings of this study contribute not only to the advancement of multilingual VQA systems but also offer valuable insights into the broader domain of natural language processing in linguistically diverse contexts.

2 Related Work

2.1 Visual Question Answering

Visual Question Answering (VQA), a field pioneered in the seminal work of Agrawal et al. (Agrawal et al., 2016), involves generating accurate natural language answers to questions about given images. This groundbreaking work utilized LSTM-based techniques for processing questions and the deep convolutional network VGG (Simonyan and Zisserman, 2015) for analyzing images. Over time, VQA has seen the development of diverse datasets that test reasoning and comprehension across various domains, employing formats like multiple choice, span selection, and generative answering.

The landscape of VQA has evolved significantly, with current state-of-the-art models predominantly relying on large, multimodal transformer architectures. These models excel in learning joint image-language embeddings, thanks to specialized attention mechanisms that focus on query-specific tokens. A notable example is the BLIP2 model (Li et al., 2023), which employs a Querying Transformer, or *Q-Former*, to synthesize image and textual embeddings pertinent to the posed questions. This approach exemplifies the strides made in integrating visual and linguistic information, a testament to the field's advancement from its LSTM and CNN origins.

2.2 Multilingual VQA

In the literature, various Visual Question Answering (VQA) datasets have played pivotal roles in advancing research across multiple dimensions. Among these, EVJVQA(Nguyen et al., 2023) stands out, presenting over 33,000 question-answer pairs in Vietnamese, English, and Japanese, setting the stage for the evaluation of multilingual VQA systems. The crossmodal dataset, XM3600 (Thapliyal et al., 2022) , contributes to the landscape with 3,600 images annotated in 36 languages, fostering crossmodal understanding. Augmenting this, a substantial cultural heritage dataset encompasses 500,000 Italian assets and 6.5 million question-answer pairs. To further enrich our exploration, we draw insights from xGQA (Pfeiffer et al., 2022) and MaXM (Geigle et al., 2023), encompassing 8 and 7 languages, respectively. This literature review underscores the diversity and significance of these datasets in shaping the trajectory of multilingual VQA research.

A contemporary trend is emerging in the development of methodologies and resources for various Natural Language Processing (NLP) applications, particularly those involving multilingual languages. Notable contributions in this sphere encompass innovative approaches aimed at addressing the intricacies of language diversity and facilitating cross-cultural communication. Among these efforts, a significant focus lies on the pursuit of constructing a robust Multilingual and Code-Mixed Visual Question Answering System through Knowledge Distillation, realized via a transformer-based framework (Khan et al., 2021).

Additionally, strides have been made in the creation of a translation-based framework(Changpinyo et al., 2023a) for Multilingual Visual Question Answering (mVQA) data generation. This approach markedly reduces the dependence on extensive human annotation efforts, presenting a notable departure from conventional methodologies.

2.3 Code mixed VQA

In recent years, the intersection of Visual Question Answering (VQA) and multilingualism has given rise to Code-Mixed VQA systems, addressing the nuanced challenges posed by linguistic diversity within a language, such as script variations and code-switching. This literature review surveys key contributions in this emerging field, highlighting methodologies, datasets, and advancements in Code-Mixed VQA systems. While most existing VQA models focus on English, there is a growing need for models that can handle multilingual and code-mixed VQA.

Code-mixed VQA is particularly challenging due to the mixing of different languages within a single question. In this paper, (Gupta et al., 2020) propose a unified framework for multilingual and code-mixed VQA. Their framework consists of three main components: a multilingual encoder, a code-mixed encoder, and a fusion layer. The multilingual encoder is used to encode questions in different languages, while the code-mixed encoder

is used to handle code-mixed questions. The fusion layer is used to combine the features from the multilingual and code-mixed encoders, and to produce an answer to the question. Multilingual VQA research has set the foundation for Code-Mixed VQA systems. Datasets like EVJVQA (Nguyen et al., 2023) and XM3600 (Thapliyal et al., 2022) have paved the way by providing question-answer pairs in multiple languages, fostering crossmodal understanding. Such datasets serve as valuable resources for training and evaluating Code-Mixed VQA models.

(Khan et al., 2021) proposed a Knowledge Distillation approach to develop a robust Multilingual and Code-Mixed VQA system. Their work emphasizes the importance of leveraging transformer-based frameworks, utilizing pre-trained models for effective knowledge transfer. This approach lays the groundwork for addressing the intricacies of language diversity within Code-Mixed VQA. A notable departure from traditional methodologies involves the introduction of translation-based frameworks for Multilingual VQA data generation (Changpinyo et al., 2023a). By reducing dependence on extensive human annotation efforts, these approaches contribute to the scalability and efficiency of Code-Mixed VQA systems. Code-mixing, the practice of seamlessly combining two or more languages within a single sentence, has gained attention in enhancing the linguistic diversity of VQA systems.

Code-mixed language is a common form of communication in many parts of the world, including India. Hinglish is a particularly prevalent code-mixed language, combining Hindi and English. While there has been growing interest in natural language processing (NLP) for code-mixed languages, there is a lack of large-scale datasets for evaluating code-mixed language generation and evaluation. To address this gap, (Agarwal et al., 2023) propose the HinGE dataset, a collection of human-generated and rule-based code-mixed Hinglish sentences, along with parallel Hindi and English sentences. The HinGE dataset is a valuable resource for researchers who are working on NLP for code-mixed languages. It is the first large-scale dataset of its kind, and it provides a variety of challenges for researchers to explore. The work of Ye et al. (Ye et al., 2023) involves employing GPT-3.5 for code-mixing in the context of generating code-mixed captions. This approach not only diversifies linguistic content but also provides opportunities for creating paired image and caption datasets for Code-Mixed VQA.

The adaptation of Code-Mixed VQA systems to languages with distinct scripts introduces challenges in transliteration. The mBLIP model, evaluated on the MaXM dataset (Changpinyo et al., 2023b), reveals challenges in accurately handling script variations, especially in the case of Devanagari Hindi (Geigle et al., 2023). This highlights the need for advancements in transliteration mechanisms to improve Code-Mixed VQA system performance. Recent works, such as the error analysis conducted on the mBLIP model (Geigle et al., 2023), underscore the importance of addressing transliteration issues. Proposed extensions involve leveraging transliteration libraries and introducing code-mixing to enhance model adaptability (Changpinyo et al., 2023b).

In conclusion, the literature on Code-Mixed VQA systems reflects a growing awareness of the challenges and opportunities presented by linguistic diversity within a language. From knowledge distillation to translation-based approaches and code-mixed language generation, researchers are exploring diverse strategies to improve the robustness and adaptability of Code-Mixed VQA models. As the field continues to evolve, addressing challenges in transliteration and code-mixing will be pivotal for the success of Code-Mixed VQA systems in diverse linguistic contexts.

3 Experiments

The initial experimentation focused on evaluating the accuracy of the mBLIP (Geigle et al., 2023) model on the MaXM dataset (Changpinyo et al., 2023b) and the xGQA dataset (Pfeiffer et al., 2022) in the context of Visual Question Answering (VQA). The assessment involved classification evaluations for English (in standard Roman script), Hindi language representations in Romanized scripts and code-mixed Hinglish language in Roman script. Transliteration from Devanagari to Roman scripture was done by using the indic-transliteration library[1].

Evaluation was performed zero shot on the datasets and the accuracy was calculated according to the same methodology used by (Geigle et al., 2023). MaXM contains a candidate set of possible answers, the model output was considered correct

[1] https://github.com/indic-transliteration/indic_transliteration_py

if it was contained in the candidate set. xGQA contains a correct label we calculate two metrics accuracy and soft accuracy. accuracy is the percent of direct matches between the model output and the ground truth. Soft accuracy is the percent of examples where the model output starts or ends with the ground truth label, note that the ground truth labels in xGQA are short one-word or few-word responses while the model output consists of complete sentences which makes this evaluation paradigm feasible.

mBLIP was trained and evaluated on 96 candidate languages, and three different tasks: image captioning, visual question answering, and matching. For the purposes of our research we choose to evaluate the model on VQA for English and Hindi. We choose these two languages since our proposed extension involves code mixed English-Hindi data the performance on these two languages is most pertinent.

3.1 English VQA Performance

To evaluate VQA performance in English, we choose the BLOOMZ-7B variant since this attained a higher score on both datasets. The accuracy of the mBLIP model on MaXM and xGQA as obtained by (Geigle et al., 2023) were 55.70% and 43.35%, respectively. We attempt to reproduce the same results over the MaXM and the xGQA datasets. We summarise our results in Table 1. However, we achieve an accuracy of 55.25% and 43.11% on the MaXM and the xGQA datasets, respectively.

Important Clarification: The original research paper achieved results using int8 precision exclusively for LLM weights, whereas our approach necessitates the use of int8 for all weights. As a consequence, we observe a slight drop of around 0.5% in performance because HuggingFace currently lacks int8 support for certain model components. This warning is also mentioned in the official mBLIP github repository. [2]

Model	Dataset	Achieved Acc (%)
Gregor/mblip-bloomz-7b	MaXM	55.25
Gregor/mblip-bloomz-7b	xGQA	43.11
Gregor/mblip-bloomz-7b	MaXM (Hindi)	50.00
Gregor/mblip-bloomz-7b	MCVQA	10.67

Table 1: Experimentation Results for the mBLIP model on different visual question-answering datasets

[2] https://github.com/gregor-ge/mBLIP

3.2 Hindi VQA Performance

The mBLIP model achieved an accuracy of 50% in classifying visual questions in the Romanized script of the MaXM dataset. The evaluation criterion involved testing whether the predicted label fell within the labels assigned for the corresponding caption in the VQA system. These accuracy scores provide insights into the system's performance in accurately answering visual questions posed in Hindi, considering variations in script representations. It can be observed that the model achieves accuracy similar to that of its English counterpart, depicting the adaptability of the model to multilingual visual question-answering tasks.

3.3 Code-Mixed VQA Performance

We used the code-mixed MCVQA (Gupta et al., 2020) dataset for evaluating the visual question-answering models on code-mixed Hinglish language. The dataset consists of (question, answer, image) tuples, where these tuples are taken from the VQA dataset. (Gupta et al., 2020) convert the English questions to English-Hindi code-mixed questions. They use the Part-of-Speech and Named Entity tags of each question and replacing the nouns and adjectives with their respective lexical translations. The remaining words in Hindi are then replaced with their roman transliterations. For example, the question "Is this a salad?" in the VQA dataset is converted to "Kya yah ek salad hai?" in the MCVQA dataset. We evaluated the mBLIP model on the MCVQA dataset and the results obtained are displayed in Table 1. The mBLIP model is unable to perform visual question-answering on the code-mixed dataset and thereby achieves a 10.67% accuracy. It is observed that the model generally replicates the question as the answer on the code-mixed dataset. A complete analysis of the behavior of the model is analyzed in Section 5.3.

4 Methodology

In this section we explore two variants of our fine-tuning approach.

4.1 Code Mixed Language Generation

This approach is based on the hypothesis that the model has strong image processing and combined vision language reasoning abilities but achieves poor zero-shot performance on code mixed simply due to the fact that it has not been trained on sufficient amounts of code mixed data. Consequently,

this approach assumes that the performance bottleneck is the language model component of the overall system.

In this approach we deal exclusively with the language model, therefore the model is a BLOOMZ-7b language model (Muennighoff et al., 2022). All the LLM weights are frozen but the query, key, and value sub-modules of the attention mechanism are outfitted with LoRA adapters (Hu et al., 2021).

To train we use the HinGE dataset (Agarwal et al., 2023) and a standard Causal Language Modelling (CLM) loss which can be defined as:

$$L = -\frac{1}{||x||}\sum_{i}^{||x||}\log P(x_i|x_1, x_2 \cdots x_{i-1}) \quad (1)$$

4.2 Code Mixed Question Answering

The first approach has a few drawbacks which limit the performance boosts that can be gained:

1. Small Dataset
2. Language distribution differs from code mixed VQA distribution

To address these two issues our second approach involves fine tuning the language model on question-answer pairs using the MCVQA dataset (Gupta et al., 2020). In this approach we finetune on a Causal Language Modelling loss as in Equation 1 with the input prompted as: *question: {question from dataset} answer: {answer from dataset}*. The model set up for this approach is identical to the first one, where we isolate the language model, freeze the weights and attach LoRA adapters. The only difference between the two is the format of data used for fine-tuning.

Note that, the associated images are not provided during fine-tuning as we are only using the language model, since the goal of our fine-tuning is to improve code mixed understanding and generation rather than overall vision-language reasoning over the image, we feel that this is a reasonable approach since the language model learns the representations of the code-mixed language through questions and learns to generate the answer to it.

4.3 Comparison & Discussion

Both these approaches are summarized below in Table 2. Based on preliminary experiments it seems that the HinGE dataset is too small to finetune the LLM on, even with a more conservative LoRA configuration. The second approach remedies this as MCVQA has a much larger corpus of code mixed text in question answer format, but the biggest drawback of this approach is that it is potentially encouraging Disconnected Reasoning - DiRe (Trivedi et al., 2020), essentially answering questions based purely on the language modality without incorporating information from the image.

	Language Generation	**Question Answering**
Model	BLOOMZ-7B	BLOOMZ-7B
Dataset	HinGE	MCVQA
Prompting	None	Question - Answer
Fine Tuning	LoRA	LoRA

Table 2: Comparison of proposed methods

4.4 Model Details and Hyperparameters

Since we're conducting parameter-efficient fine-tuning, we chose a low learning rate of 1e-5 and only trained for 30 epochs. We choose the Adam optimizer with weight decay regularization set to 1e-2.

Training Configuration	
Learning Rate	1e-5
Epochs	30
Optimizer	AdamW
Weight Decay	1e-2
Criterion	Causal LM

Table 3: Training configuration

For the LoRA configuration *r* controls the rank of the low-rank matrix, it represents a tradeoff between flexibility and parameter efficiency. A smaller rank results in fewer parameters, making the model more efficient but potentially less flexible. The *alpha* parameter is a scaling factor applied to the LoRA matrices. This scaling allows for controlled adjustment of the impact these matrices have on the model's pre-trained weights, influencing the extent to which the original model is modified during adaptation. Our LoRA configurations are specified in Table 4. Note that due to the smaller size of the HinGE dataset, we use a more conservative LoRA specification for the language generation task. Moreover, since the HinGE dataset is a more generalized dataset than the MCVQA dataset for a question-answering task, thus we aim to adapt the model more towards the question-answering task. As a result, a higher value of r and Alpha encourage the model to be fine-tuned on a task-specific dataset like MCVQA.

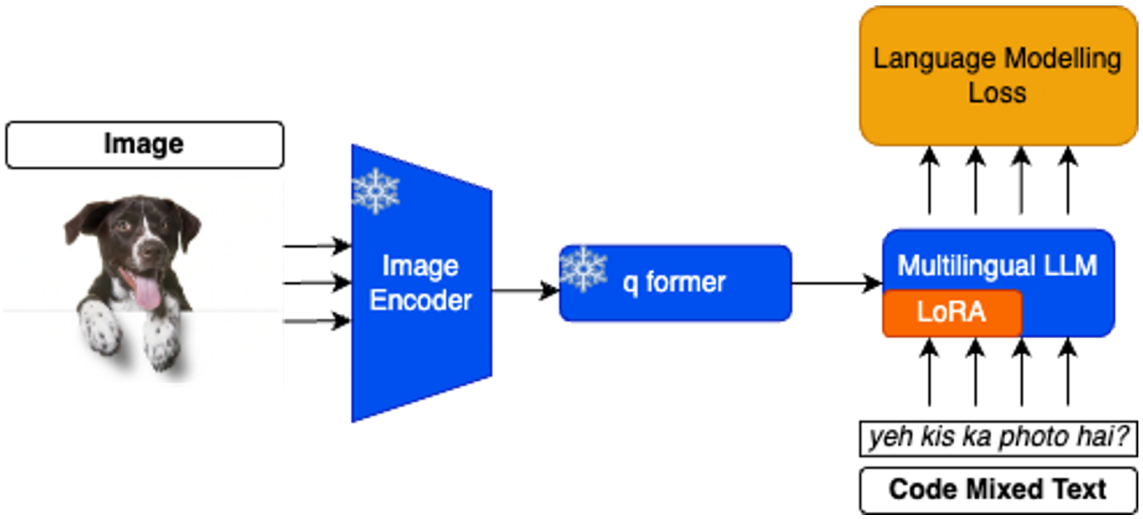

Figure 1: VQA Pipeline: This figure shows the pipeline for VQA using the mBLIP model. For our finetuning procedure, we take the language model in isolation and fine tune it on a causal language modelling task

LoRA Configuration		
	Language Generation	**Question Answering**
r	8	16
Alpha	4	8
‘ Dropout	0.05	0.05
Bias	None	None

Table 4: LoRA configuration parameters

5 Results

5.1 Evaluation Strategy

We used the accuracy metric for comparing the performance of the visual-question answering models over the MCVQA dataset. The ground truth of the questions in the MCVQA dataset consists of a list of 10 answers annotated by 10 different annotators. We compare the generated answer to the ground truth labels by performing an exact match between the generated answer and the ground truth. If the answer matches the answers of atleast 3 annotators, then we consider the answer as a hit. We then compute the accuracy of the model by using the average number of hits to the total number of samples in our dataset. We perform an exact match with the answer labels instead of a soft matching technique like comparing the Word2Vec (Mikolov et al., 2013) embeddings of the predicted output with the ground truth label because there is a possibility of existence of two words such as ’left’ and ’right’, which might give out a similar embedding, but in reality are two very different concepts. Thus, in order to avoid such errors, we perform an exact match with the ground truth labels.

5.2 Finetuning Results

Model	**Accuracy (%)**
Gregor/mblip-bloomz-7b	10.67
Gregor/mblip-bloomz-7b-HinGE-ft	**34.57**
Gregor/mblip-bloomz-7b-MCVQA-ft	**69.84**

Table 5: Results for the mBLIP model on the MCVQA dataset

In our finetuning experiments, we observed significant performance improvements in the mBLIP model for the code-mixed visual question-answering task, showcasing its adaptability and versatility. Specifically, we focused on two fine-tuning scenarios: HinGE and MCVQA-specific fine-tuning. The baseline model, denoted as Gregor/mblip-bloomz-7b, achieved an accuracy of 10.67%. This serves as our reference point for evaluating the effectiveness of finetuning strategies. The results have been summarized in Table 5.

When using the HinGE dataset during finetuning (Gregor/mblip-bloomz-7b-HinGE-ft), we observed a noteworthy boost in performance, with the accuracy increasing to 34.57%. This suggests that the VQA model benefits from the training focused on general code-mixed knowledge injection through the HinGE dataset, showcasing an improved ability to understand the nuances of the data. However, the most striking results were obtained when fine-tuning the mBLIP model specifically on the MCVQA dataset (Gregor/mblip-bloomz-7b-MCVQA-ft). In this scenario, the accuracy skyrocketed to 69.84%, surpassing both the baseline and hinge finetuned models. This emphasizes the

importance of task-specific finetuning, demonstrating that tailoring the model to the intricacies of the MCVQA dataset with causal language modeling significantly enhances its performance. It also enables the model learn to answer to questions and adjusting its parameters such that the language model understands code-mixed questions by learning to represent them in the same vector space as the English questions from the VQA dataset.

The success of the mBLIP model on the MCVQA dataset after fine-tuning reveals the effectiveness of training the model on data that reflects the linguistic diversity present in the MCVQA questions. Remarkably, we found that leveraging large language models (LLMs) for finetuning with code-mixed language data played a pivotal role in enhancing the model's ability to comprehend and answer questions presented in a code-mixed format. Fine-tuning with code-mixed language data also allows the model to adapt and understand the intricacies of mixed-language inputs, enabling it to handle the code-switching phenomena present in the MCVQA dataset more effectively. This targeted finetuning approach proves to be more beneficial than fine-tuning the complete model, as it hones the model's language understanding capabilities without compromising its existing knowledge.

In conclusion, our results underscore the significance of dataset-specific finetuning and the advantageous impact of leveraging code-mixed language data for improving the performance of multimodal VQA models like mBLIP on code-mixed visual question-answering tasks.

5.3 Error Analysis

We selected a few images and performed evaluation on the baseline model, the baseline finetuned on Hinge dataset as well as the baseline model finetuned on MCVQA dataset

We see the park figure - 2, where the question used for evaluation was *kya yeh park main he* which translates to *Is this in the park*, the ground truth for this question is *yes*. As we can see from the table 6, the baseline model just repeated the question again in the answer, the model finetuned on hinge just returned a bunch of dots but the model finetuned on MCVQA returned the correct answer. Althought the response isn't completely clean, it gave the response with 100% accuracy.

In the windows figure - 3, the question used for evaluation was *aap kitni windows dekh sakate hai* which translates to *How many windows can you see*, the ground truth for this question is *1*. As we can see from the table 7, the baseline model didn't return any response at all. It gave blank response specifically for this question, maybe because it was a numerical response and it couldn't interpret it at well. The model finetuned on hinge just returned a bunch of gibberish and the model finetuned on MCVQA returned the answer as *2 windows*. Although, it couldn't get the accuracy of the output correct this time, it still could interpret the fact that it had to 'count' the windows present. So, the factual accuracy is perhaps not due to code-mixing interpretation issues but maybe the image encoder couldn't get the image representations correct, from what we can understand.

Finally, in the street figure - 4, the question used for evaluation was *upper right side main kya he* which translates to *What is present on the upper right side*, the ground truth for this question is *tree*. As we can see from the table 8, the baseline model performed pretty well this time. It returned the result *?Upper right side main image is a man walking down a street.* So it was able to guess the context very well this time, maybe since the major part of the question *upper right side* is in English itself. Although, it understood the context, it returned the response saying "a man" (*who is present on the lower right side*) rather than a "tree", so maybe it couldn't get the "upper right" representation right which emphasizes on our previous point of the text-image representation of the baseline model being a bit weak in general. The model finetuned on hinge captured the context of the image correctly about a man on the road, however it couldn't capture the spacial information of the image regarding the upper right corner of the image. The model finetuned on MCVQA returned the answer as *building*. So, althought it could get the context, it couldn't accurately distinguish between upper left (*building is present on the upper left side*) and upper right correctly. So, there is definitely some scope of improvement here.

The evaluative scrutiny of the baseline model and its iteratively fine-tuned counterparts across distinct image-question pairs delineated nuanced performance nuances. The "Park" scenario witnessed the baseline's propensity for repetitive verbosity, the Hinge-adapted model's perplexing symbolic response, and the MCVQA-refined model's exceptional accuracy, underscoring the domain-specific efficacy of targeted training. Conversely, the "Windows" inquiry exposed the baseline's numerical

aversion, the Hinge model's nonsensical output, and the MCVQA variant's cognizance of counting imperatives yet faltering in absolute precision, illuminating potential inadequacies in numerical cognition and image representation. Lastly, the "Street" tableau elucidated the baseline's adept contextual grasp but flawed spatial delineation, the Hinge model's contextual adeptness yet spatial inadequacy, and the MCVQA-fortified model's contextual acuity yet discernment challenges between upper left and upper right spatial orientations. In synthesis, while fine-tuned models evinced discernible enhancements in designated contexts, lingering impediments beckon further semantic refinement and sophisticated contextual discernment for optimal text-image interplay.

Figure 2: A park image used for inference with the Hinglish question : kya yeh park main he

Figure 3: An image used for inference with the Hinglish question : aap kitni windows dekh sakate hai

Model	Output
Baseline model - Gregor/mblip-bloomz-7b	?park main he
Finetuned using Hinge	.....
Finetuned using MCVQA	yes
Ground Truth	yes

Table 6: Results for Figure 2

Figure 4: An image used for inference with the Hinglish question : upper right side main kya he

Model	Output
Baseline model - Gregor/mblip-bloomz-7b	*no response*
Finetuned using Hinge	the is the of of of of of of of
Finetuned using MCVQA	2 windows
Ground Truth	1

Table 7: Results for Figure 3

6 Conclusion

In conclusion, our investigation into code-mixed Visual Question Answering (VQA) not only provides valuable insights but also presents tangible advancements in addressing the challenges of linguistic complexity. Utilizing the mBLIP model, we fine-tuned our approaches on datasets reflecting the nuances of multilingual and code-mixed languages.

The mBLIP model's adaptability and learning prowess were evident, with substantial performance enhancements observed in both code-mixed language generation and question-answering fine-tuning approaches. The use of LoRA (Low-Rank Adaptation) matrices in the fine-tuning process played a pivotal role, contributing to the model's effectiveness in capturing and learning code-mixed language nuances.

Specifically, fine-tuning on the MCVQA dataset resulted in an impressive accuracy of 69.84%, surpassing baseline and hinge fine-tuned models. These results underscore the importance of targeted fine-tuning and showcase the effectiveness of incorporating LoRA matrices to enhance the model's understanding of code-mixed language structures.

Looking forward, our future work involves extending this approach to diverse language pairs, considering the rich linguistic diversity across regions. The efficacy of LoRA in fine-tuning provides a strong foundation for further exploration and enhancement of other multilingual Visual

Model	Output
Baseline model - Gregor/mblip-bloomz-7b	?Upper right side main image is a man walking down a street.
Finetuned using Hinge	man on road
Finetuned using MCVQA	building
Ground Truth	tree

Table 8: Results for Figure 4

Question Answering (VQA) models. This opens up possibilities for creating versatile systems capable of addressing nuanced language dynamics in various global contexts, underlining the pivotal role of low-rank adaptation in the success of fine-tuning strategies.

References

Rohit Agarwal, Ashutosh Kumar, and Sumit Sharma. 2023. Hinge: A dataset for generation and evaluation of code-mixed hinglish text.

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

Beer Changpinyo, Linting Xue, Michal Yarom, Ashish Thapliyal, Idan Szpektor, Julien Amelot, Xi Chen, and Radu Soricut. 2023a. Maxm: Towards multilingual visual question answering. In *Findings of ACL: EMNLP*.

Soravit Changpinyo, Linting Xue, Michal Yarom, Ashish V. Thapliyal, Idan Szpektor, Julien Amelot, Xi Chen, and Radu Soricut. 2023b. Maxm: Towards multilingual visual question answering.

Gregor Geigle, Abhay Jain, Radu Timofte, and Goran Glavaš. 2023. mblip: Efficient bootstrapping of multilingual vision-llms.

Deepak Gupta, Pabitra Lenka, Asif Ekbal, and Pushpak Bhattacharyya. 2020. A unified framework for multilingual and code-mixed visual question answering. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Humair Raj Khan, Deepak Gupta, and Asif Ekbal. 2021. Towards developing a multilingual and code-mixed visual question answering system by knowledge distillation.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, et al. 2022. Crosslingual generalization through multitask finetuning. *arXiv preprint arXiv:2211.01786*.

Ngan Luu-Thuy Nguyen, Nghia Hieu Nguyen, Duong T.D. Vo, Khanh Quoc Tran, and Kiet Van Nguyen. 2023. EVJVQA CHALLENGE: MULTILINGUAL VISUAL QUESTION ANSWERING. *Journal of Computer Science and Cybernetics*, pages 237–258.

Jonas Pfeiffer, Gregor Geigle, Aishwarya Kamath, Jan-Martin O. Steitz, Stefan Roth, Ivan Vulić, and Iryna Gurevych. 2022. xgqa: Cross-lingual visual question answering.

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

Ashish V. Thapliyal, Jordi Pont-Tuset, Xi Chen, and Radu Soricut. 2022. Crossmodal-3600: A massively multilingual multimodal evaluation dataset.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2020. Is multihop QA in DiRe condition? measuring and reducing disconnected reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8846–8863, Online. Association for Computational Linguistics.

Junjie Ye, Xuanting Chen, Nuo Xu, Can Zu, Zekai Shao, Shichun Liu, Yuhan Cui, Zeyang Zhou, Chao Gong, Yang Shen, Jie Zhou, Siming Chen, Tao Gui, Qi Zhang, and Xuanjing Huang. 2023. A comprehensive capability analysis of gpt-3 and gpt-3.5 series models.