Towards Developing a Multilingual and Code-Mixed Visual Question Answering System by Knowledge Distillation

Humair Raj Khan,* **Deepak Gupta,*** **Asif Ekbal**
Department of Computer Science and Engineering
Indian Institute of Technology Patna, India
`khumairraj@gmail.com, {deepak.pcs16,asif}@iitp.ac.in`

Abstract

Pre-trained language-vision models have shown remarkable performance on the visual question answering (VQA) task. However, most of the pre-trained models are trained by only considering monolingual learning, especially the resource-rich language like English. Training such models for multilingual setups demand high computing resources and multilingual language-vision dataset which hinders their application in practice. To alleviate these challenges, we propose a knowledge distillation approach to extend an English language-vision model (teacher) into an equally effective multilingual and code-mixed model (student). Different from the existing knowledge distillation methods, which only use the output from the last layer of the teacher network for distillation, our student model learns and imitates the teacher from multiple intermediate layers (language and vision encoders) with appropriately design distillation objectives for incremental knowledge extraction. We also create the large-scale multilingual and code-mixed VQA dataset in eleven different language setups considering the multiple Indian and European languages. Experimental results and in-depth analysis show the effectiveness of the proposed VQA model over the pre-trained language-vision models on eleven diverse language setups.

1 Introduction

Visual Question Answering (VQA) is a challenging problem in computer vision (CV) and natural language processing (NLP) that have gained popularity due to its many-fold benefits ranging from assisting visually impaired users to establishing effective communication with robots *via* intuitive interfaces.

The existing works (Tan and Bansal, 2019; Antol

et al., 2015) on VQA are mainly limited to the English questions, making it challenging to acknowledge progress in foreign languages. Moreover, the current language-vision models do not serve the purpose in the code-mixed setting, where the morphemes, words, phrases of one language are embedded into the other language. Since code-mixing has been a mean of communication in a multi-cultural and multi-lingual society, the next generation of artificial intelligence (AI) agents should be capable to understand the **M**ultilingual and **C**ode-**M**ixed (MCM) questions about the image.

In the recent past, the pre-trained language-and-vision models (Su et al., 2020; Tan and Bansal, 2019; Li et al., 2019; Lu et al., 2019) have become the state-of the-arts for solving a variety of CV and NLP problems. However, the majority of these models are predominantly built for resource-rich languages like English. Therefore, their abilities to process and answer the MCM questions are limited (*c.f.* Fig. 1).

To address this, we propose a highly effective and unified VQA method that allows us to extend the existing monolingual language-and-vision models to multilingual (in 6 **different languages**) and code-mixed (in 5 **different code-mixed languages**) scenarios. Specifically, we develop a novel knowledge distillation (Hinton et al., 2015) approach to distill the knowledge from the monolingual language-and-vision transformer network (teacher model) to multilingual and code-mixed language-and-vision transformer network (student model). This enables the student model to adapt to any language and code-mixed scenarios.

In order to effectively transfer the knowledge from the teacher network to the student network, we introduce multiple distillation objectives which ensure the incremental knowledge extraction from multiple intermediate layers of language-and-vision transformer model. These objectives are formulated to guide the student model to learn two

---
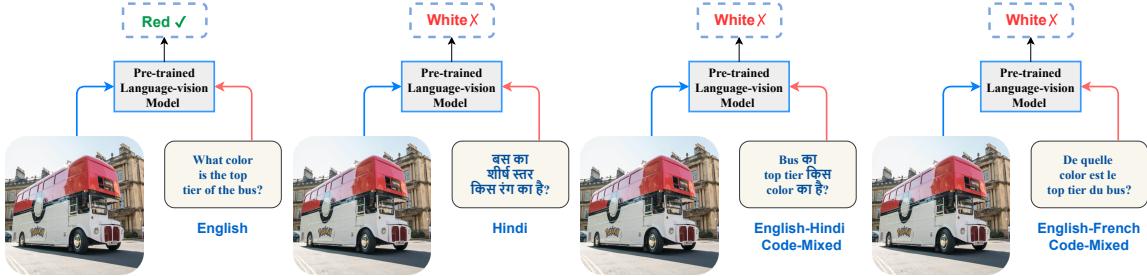*These authors contributed equally to this work.

Figure 1: Example multilingual and code-mixed questions (same question), where pre-trained language-and-vision models fails to correctly predict the answer except for English question.

key characteristics: **(1)** unified question representation across different languages, and **(2)** effective cross-modal (image-question) representation where the key objects in the image are attended irrespective of the language of the questions. With these characteristics, we are able to build a unified VQA model, which can correctly predict the answers to multilingual and code-mixed questions.

Furthermore, to combat the data scarcity, we also create a large-scale (3.7M image-question pairs) multilingual and code-mixed VQA dataset. Towards this, we utilize the English VQA1.0 (Antol et al., 2015) dataset and extend it to multiple languages. We also create their code-mixing counterparts by designing the linguistically informed strategy to formulate the code-mixed question by mixing the words/phrases from the English question and the foreign language question. We evaluate our proposed approach on the created dataset and achieve 11.74% average improvement across all the languages over the pre-trained language-vision model.

Contributions:

1. We devise a robust knowledge distillation framework for multilingual and code-mixed VQA by introducing multiple task-specific objective functions, which distill knowledge from the English pre-trained language-vision model to train and develop equally effective multilingual and code-mixed VQA system.

2. We create the large-scale (3.7M) multilingual and code-mixed VQA datasets in multiple languages: Hindi (*hi*), Bengali (*bn*), Spanish (*es*), German (*de*), French (*fr*) and code-mixed language pairs: *en-hi*, *en-bn*, *en-fr*, *en-de* and *en-es*. This dataset is publicly available here[1].

3. We demonstrate the effectiveness of our proposed single student model that can correctly predict the answers to the questions of the various language combinations (on eleven (11) different language-vision setups) including code-mixed setups over state-of-the-art pre-trained language-vision models.

2   Related Work

Multilingual and Code-Mixing: There is a recent trend in developing methods and resources for various NLP applications involving multilingual and code-mixed languages. Some of the works include question-answering (Raghavi et al., 2015; Gupta et al., 2018b), word embedding (Chen and Cardie, 2018; Lample et al., 2018; Pratapa et al., 2018b), code-mixed text generation (Pratapa et al., 2018a; Gonen and Goldberg, 2019; Gupta et al., 2020a), code-mixed language modelling (Winata et al., 2018; Gonen and Goldberg, 2019), and other NLP tasks (Gupta et al., 2018a, 2016a,b, 2017).

Visual Question Answering: In the literature, various VQA datasets (Silberman et al., 2012; Gao et al., 2015; Antol et al., 2015; Goyal et al., 2017) have been created to encourage multi-disciplinary research. The popular frameworks for VQA explore attention mechanisms to learn the joint representation of image and question (Fukui et al., 2016; Kim et al., 2017; Yu et al., 2017; Kim et al., 2018). Recently, with the success of Transformer (Vaswani et al., 2017), Tan and Bansal (2019) proposed cross-modality framework, LXMERT, for learning the connection between vision and language. There are other notable works (Su et al., 2020; Zhou et al., 2020; Li et al., 2020), where the Transformer-based models are pre-trained to learn the joint language-vision representation. Knowledge distillation has also been used in the literature for the VQA task for the optimal training strategy (Mun et al., 2018), knowledge transfer from tri-modal to bi-modal (Do et al., 2019), and the missing modalities (Cho et al., 2021). Unlike these,

---

[1]https://www.iitp.ac.in/~ai-nlp-ml/resources.html

our current work focuses on knowledge transfer from the monolingual pre-trained language-vision model to the multilingual and code-mixed VQA.

3 Multilingual and Code-Mixed VQA Dataset

Dataset Creation: We follow the large-scale VQA dataset (VQAv1.0) released by Antol et al. (2015). The VQAv1.0 dataset contains the triplet information in the form of the question, image, and answers. Gupta et al. (2020b) introduced a VQA dataset named MCVQA which comprises of questions in Hindi and Hinglish (i.e. code-mixed English and Hindi). However, the approach to create MCVQA dataset has two major shortcomings: **(1)** algorithm is not scalable to other languages, and **(2)** it requires the NLP components (part-of-speech tagger, named entity recognizer, transliteration, etc.) for the resource-scarce languages, which, themselves are an active research area for the resource-scare languages.

To address these shortcomings, in this work, we create the large-scale "**Mu**ltilingual and **Co**de-mixed **V**isual **Q**uestion **A**nswering" (MuCo-VQA) dataset which supports the five (5) languages (*hi*, *bn*, *es*, *de*, and *fr*) and five (5) different code-mixed settings (*en-hi*, *en-bn*, *en-es*, *en-de*, and *en-fr*). To generate the code-mixed questions, we follow the matrix language frame (MLF) theory (Myers-Scotton, 1997) of code-mixed text. According to MLF, a code-mixed sentence will have a dominant language (matrix-language) and inserted language (embedded-language). We utilize the Google machine translation to translate the English questions from VQAv1.0 dataset to the foreign language *xx* ∈ {*hi*, *bn*, *es*, *de*, *fr*}. From the parallel questions (*en-xx*), we learn the alignment of English words in the foreign language question. Given a pair of questions from the two languages, we identify the words following Gupta et al. (2020a) from the English question and substitute their aligned counterparts (in foreign language question) with the identified English words to synthesize the English embedded code-mixed questions. Please see **Appendix** for the implementation details and samples of the MuCo-VQA dataset.

Analysis: Similar to the VQAv1.0 dataset, our created MuCo-VQA dataset consists of $248,349$ training and $121,512$ test questions for each of the five different languages and five code-mixed settings. We perform a qualitative analysis of this dataset by randomly selecting $5,00$ questions, each from *en*, *hi* and corresponding *en-hi*. We seek annotation help from two bilingual (*en*, *hi*) experts to manually translate and create the code-mixed questions. Towards this, we compute the BLEU (Papineni et al., 2002), ROUGE (Lin, 2004) and Translation Error Rate (TER) (Snover et al., 2006) considering the manually created code-mixed questions as the gold standard; and the generated code-mixed questions from MuCo-VQA as the candidates. We compute the mean values of the individual scores obtained from both the experts. We found the BLEU: 78.34, ROUGE-L: 91.13, and TER: 8.23, which show the generated code-mixed questions are close to the human formulated code-mixed questions. The detailed analysis and statistics in terms of code-mixed complexity can be found in the **Appendix**.

4 Methodology

Our proposed knowledge distillation framework for the VQA model is tailored to predict the answer for multilingual and code-mixed questions. We utilize LXMERT (Tan and Bansal, 2019), a pre-trained English vision-language model, as the teacher network to train our student network. Our student network is inspired by the teacher network and has three components, *viz.* **(1) MCM Question Encoder** that processes and effectively encodes the multilingual and code-mixed questions, **(2) Image Encoder** which learns the representation of the objects detected in the image, **(3) Cross-Modality Encoder**, that learns the joint feature representation by applying the cross-attention on the language and image features, and **(4) Answer Prediction**, which predicts the answer for MCM questions.

4.1 Background

Transformer Block: For an input sequence $S^l = \{S_1^l, S_2^l, \ldots, S_{|S|}^l\}$ of length $|S|$ (which is the output of the lth transformer block) the $(l+1)^{th}$ transformer block computes the hidden states S^{l+1} as follows:

$$
\begin{aligned}
\hat{S}_i^{l+1} &= S_i^l + \text{MHA}(\text{LayerNorm}(S_i^l)) \\
S_i^{l+1} &= \hat{S}_i^{l+1} + \text{MLP}(\text{LayerNorm}(\hat{S}_i^{l+1}))
\end{aligned}
\tag{1}
$$

where, MHA(.) is Multi Head Attention (Vaswani et al., 2017), LayerNorm(.) is Layer Normalization (Ba et al., 2016) and MLP(.) is a feed-forward network. Based on Eq. 1, we define

Transformer-Block(.) as a function of input $S^l \in \mathcal{R}^{|S| \times d}$ as follows:

$$S^{l+1} = \text{Transformer-Block}(S^l) \quad (2)$$

4.2 Student Network

MCM Question Encoder: The input question $\mathcal{Q}$ is tokenized using the WordPiece tokenizer (Wu et al., 2016) to form the sequence of tokens $\{t_1, t_2, \ldots, t_n\}$ with length n. We compute the word embedding w_i for the ith token similar to the teacher network LXMERT. Since, in this work, we deal with multilingual and code-mixed questions; therefore, we utilize Multilingual-BERT (M-BERT) (Devlin et al., 2019) model as our language encoder. Multilingual-BERT is a single model pre-trained on the monolingual Wikipedia corpora from 104 languages. The word embedding sequence $\{w_{0=[CLS]}, w_1, \ldots, w_n\}$ (with the [CLS] token) is passed to the stack of M-BERT encoders. Each M-BERT encoder consists of the MHA(.) layer followed by point-wise feed-forward network with the residual connection. We obtain the hidden state representation $H^M = \{h_0^M, h_1^M, \ldots, h_n^M\}$ from M-BERT having M layers as follows:

$$
\begin{aligned}
h_0^1, \ldots, h_n^1 &= \text{M-BERT}^{l=1}(w_0, \ldots, w_n) \\
h_0^M, \ldots, h_n^M &= \text{M-BERT}^{l=M}(h_0^{M-1}, \ldots, h_n^{M-1})
\end{aligned}
\quad (3)
$$

For brevity, we will call $\{h_0^M, h_1^M, \ldots, h_n^M\}$ as $H = \{h_0, h_1, \ldots, h_n\}$ in rest of the paper.

Image Encoder: Given the input image $\mathcal{I}$, we extract k objects $\{o_1, o_2, \ldots, o_k\}$ from Anderson et al. (2018). For each object o_j, we obtain RoI features $r_j \in \mathcal{R}^{d_r}$ and bounding box co-ordinates $b_j \in \mathcal{R}^{d_b}$. We follow the object-relationship encoder from Tan and Bansal (2019) to obtain the image representation. We first project RoI and co-ordinates via a feed-forward network to obtain f_j and p_j, respectively. Then we obtain the object feature for the object o_j as $u_j = (f_j + p_j)/2 \in \mathcal{R}^d$. With k objects in the image, we obtain the object feature matrix $U^0 \in \mathcal{R}^{k \times d}$. We employ the stack of Transformer-Block (c.f. Eq. 2) to encode the image. For the first Transformer-Block, we fed the object feature matrix U^0 and obtain the hidden state representations $u_1^1, \ldots, u_k^1$. Subsequently, we obtain the final image representation $U = U^N \in \mathcal{R}^{k \times d}$ from the last layer (N) of Transformer-Block as follows:

$$u_1, \ldots, u_k = \text{Transformer-Block}(U^{N-1}) \quad (4)$$

Cross-Modality Encoder: Given the MCM question representation $H \in \mathcal{R}^{n \times d}$ and image representation $U \in \mathcal{R}^{k \times d}$, similar to Tan and Bansal (2019), we aim to compute the cross-modal representations using the layers of Transformer-Block. For a given layer l, the cross-modality encoder consists of two cross-attention layers (one from question to image another from image to question) and two Transformer-Block for each modality. Cross-attention layer X-Att(.) takes the query vector x^q of the representation x from one of the modals and compute the attention weight $\alpha_j = softmax(x^q.y_j^k)$ with the key vectors= y_j^k from the other modality. Thereafter, it computes the final cross-modal representation $\overline{x} = \sum \alpha_j y_j^v$ as the weighted average of the set of value vectors $\{y^v\}$. For the cross-modal representation $\overline{H}^l \in \mathcal{R}^{n \times d}$ from the lth layer of the question, we apply the X-Att followed by the Transformer-Block operation as follows:

$$
\begin{aligned}
\widetilde{h}_i^l &= \text{X-Att}(h_i^{l-1}, [u_1^{l-1}, u_2^{l-1}, \ldots, u_k^{l-1}]) \\
\widetilde{H}^l &= [\widetilde{h}_0^l, \widetilde{h}_1^l, \ldots, \widetilde{h}_n^l] \in \mathcal{R}^{n \times d} \\
\overline{H}^l &= \text{Transformer-Block}(\widetilde{H}^l)
\end{aligned}
\quad (5)
$$

Similarly, the cross-modal representation $\overline{U}^l$ for the image considering the question as another modal is computed. We use the L layers of cross-modal encoders to encode the cross-modal representation.

Answer Prediction: To predict the answer for the multilingual question, we take the output of question from the last (L^{th}) cross-modal encoders. We use the [CLS] token representation $\overline{h}_{[CLS]}^L \in \mathcal{R}^d$ and predict the answer as follows:

$$
\begin{aligned}
\overline{P} &= gelu(\mathbf{W_P}\overline{h}_{[CLS]}^L + c_P) \\
p(\mathcal{A}_i | \mathcal{X}; \theta^S) &= \sigma(\mathbf{W}_i \overline{P} + c_i)
\end{aligned}
\quad (6)
$$

where, $\mathbf{W_P} \in \mathcal{R}^{2d \times d}$ is the weight matrix and $c_P \in \mathcal{R}^{2d}$ is the bias vector. σ denotes the sigmoid function. W_i and c_i are the ith entry of weight matrix $W \in \mathcal{R}^{d \times |\mathcal{A}|}$ and bias vector $c \in \mathcal{R}^{|\mathcal{A}|}$. $\overline{h}_{[CLS]}^L \in \mathcal{R}^d$ is the hidden state representation of [CLS] token obtained from cross-modality encoder. $|\mathcal{A}|$ is the length of the answer vocabulary. $\mathcal{X}$ is the set of input $\{\mathcal{Q}, \mathcal{I}\}$. $p(\mathcal{A}_i | \mathcal{X}; \theta^S)$ is the probability of the ith answer from answer vocabulary $\mathcal{A}$.

4.3 Distillation Objectives

In our knowledge-distillation framework, we propose multiple objectives to transfer the knowledge

from the monolingual Teacher network (with θ^T parameters) to the MCM Student network (with θ^S parameters):

Objective 1 - `CLS` Token Distillation: The `[CLS]` token embedding learned at the cross-modality encoder represents the semantics of the monolingual question-image pair in teacher network and MCM question-image pair in student network. We argue that it should learn a similar representation to correctly predict the answer irrespective of the language. Towards this, we compute the `[CLS]` token loss by computing the Mean Squared Error (MSE) between the vector representation learned at the Cross-modality Encoder in the teacher network and student network.

$$\mathcal{L}_{CLS} = \sum_{i=1}^{i=|L|} \sum_{j=1}^{j=|MH|} \mathbf{MSE}(\overline{h}_{(i,j,[CLS])}^T, \overline{h}_{(i,j,[CLS])}^S) \tag{7}$$

where, $\overline{h}_{(i,j,[CLS])}^T \in \mathcal{R}^d$ and $\overline{h}_{(i,j,[CLS])}^S \in \mathcal{R}^d$ are the representation of the `[CLS]` token obtained from the ith cross-modal encoder layer under j^{th} attention head from teacher and student network, respectively. $|MH|$ is the number of attention head in the `Transformer-Block`.

Objective 2 - Object Attention Distillation: The answer to a given question is defined by the object detected in the image. It is to be noted that the answer to a question is independent of the language. We argue that in order to correctly predict the answer to MCM questions, the student network should attend the same object as the teacher network. This helps in aligning the question representation across different languages to the object representation and thus assists towards learning the effective language-agnostic cross-modal representation of the question-image pair. Towards this, we compute the object attention loss ($\mathcal{L}_{object}$), which measures the MSE between the raw score vectors $z \in \mathcal{R}^k$ (obtained using the dot product between `[CLS]` token's query vector and set of object's key vector) learned at the Cross-modality Encoder in the teacher network and student network.

$$\mathcal{L}_{object} = \sum_{i=1}^{i=|L|} \sum_{j=1}^{j=|MH|} \mathbf{MSE}(z_{(i,j)}^T, z_{(i,j)}^S) \tag{8}$$

where, $z_{(i,j)}^T \in \mathcal{R}^k$ and $z_{(i,j)}^S \in \mathcal{R}^k$ are the vector raw scores obtained from the ith layer under the jth attention head from Teacher and Student network, respectively.

Objective 3 - Prediction Distillation: In addition to imitating the behaviors of intermediate layers, we also use the knowledge distillation to mimic the predictions of teacher network. Specifically, we penalize the binary cross-entropy loss between the answer probabilities obtained from the teacher and student network.

$$\mathcal{L}_{pred} = - \sum_{i=1}^{i=|\mathcal{A}|} p(\mathcal{A}_i|\mathcal{X};\theta^T)log(p(\mathcal{A}_i|\mathcal{X};\theta^S)) + \\ (1 - p(\mathcal{A}_i|\mathcal{X};\theta^T))log(1 - p(\mathcal{A}_i|\mathcal{X};\theta^S)) \tag{9}$$

Objective 4 - Negative Log-likelihood Loss: We also penalize the binary cross-entropy loss between the gold answer probability y_i and model's predicted probability $p(\mathcal{A}_i|\mathcal{X};\theta^S)$ obtained from the student network.

$$\mathcal{L}_{nll} = - \sum_{i=1}^{i=|\mathcal{A}|} y_i log(p(\mathcal{A}_i|\mathcal{X};\theta^S)) + \\ (1 - y_i)log(1 - p(\mathcal{A}_i|\mathcal{X};\theta^S)) \tag{10}$$

4.4 Learning

To apply the knowledge distillation, first we need to train our Teacher network, having θ^T parameters with English question from the VQAv1.0 dataset. Thereafter, the Teacher network's parameters are frozen and the Student network is trained with the following objective function:

$$\mathcal{L} = \mathcal{L}_{\text{CLS}} + \mathcal{L}_{\text{object}} + \mathcal{L}_{\text{pred}} + \mathcal{L}_{\text{nll}} \tag{11}$$

During training, the Teacher network is fed with the English question and the corresponding image, and the Student network is fed with multilingual and code-mixed questions (one language at a time) and the corresponding image.

5 Dataset and Experiments

5.1 Datasets

We evaluate our proposed knowledge distillation framework on the `MuCo-VQA` dataset having eleven different language setups, and the MCVQA dataset (Gupta et al., 2020b) that consists of *en*, *hi*, *en-hi* language setups. We train the Student network with the training dataset from all these languages. We take out 5% of the training dataset as the validation dataset for evaluating and selecting the best Student model. The best Student model is used to evaluate the performance of the `MuCo-VQA` test dataset in all the language setups. For evaluation, we follow the accuracy metric as defined in Antol et al. (2015).

5.2 Implementation Details

We use the pre-trained Multilingual BERT[2] having 12 encoder layers, each having 12 attention heads and a hidden dimension of 768 for each token. In our proposed knowledge distillation framework, we train the model on `MuCo-VQA` training dataset for 16 epochs. Since the input image remains the same for the LXMERT model and our proposed model, we initialize our image encoder weights with the LXMERT object relationship encoders' weights.

We set the maximum question length to 20 words. The numbers of objects extracted from the image is $k = 36$ and the dimension of bounding box coordinates and RoI features are $d_b = 4$ and $d_r = 2048$, respectively. For the Teacher network, the language encoder has $M = 9$ layers, the image encoder has $N = 5$ layers and the cross-modality encoder has the $L = 5$ layers. Similarly in the Student network, the values of these layers are $M = 12, N = 5, L = 5$. During training, we fine-tune the top 4 M-BERT encoders and the top 2 image encoders. We learn the cross attention layer from scratch to align the multilingual and vision embeddings. For the `CLS` token distillation, we set the layers $i \in \{1, 4\}$ and attention head $j \in \{1, 4, 5\}$. Optimal values of the hyperparameters are chosen based on the model performance on the development set of `MuCo-VQA` dataset.

5.3 Baselines

We compare the performance of the proposed network with the following baseline models.
(1) LXMERT: We train the individual LXMERT model on the training dataset of each language from `MuCo-VQA` dataset and evaluate the performance on the respective test dataset.
(2) Joint LXMERT: We train the single LXMERT model on all the training datasets of each language from `MuCo-VQA` dataset and evaluate the performance on the respective test dataset.
(3) Joint LXMERT+ M-BERT: This baseline is similar to the Joint LXMERT but the monolingual language encoder is replaced with a multilingual M-BERT encoder.
(4) VL-BERT (Su et al., 2020): We also compare the performance of our proposed model with the VL-BERT base model (`vl-bert-base-e2e.model`). We train separate VL-BERT model on the training dataset

of each language from the `MuCo-VQA` dataset and evaluate the performance on the respective test dataset.
(5) VisualBERT (Li et al., 2019): Similar to the LXMERT, we also compare the performance of our proposed network on the `MuCo-VQA` dataset with the VisualBERT monolingual model.

5.4 Results

We report the performance of the baseline models and our proposed model on `MuCo-VQA` dataset in Table 1. We also reported the answer-type wise results on `MuCo-VQA` dataset in Table 3. Our proposed model achieves 70.76 overall accuracy and outperforms the best monolingual and multilingual baselines with significant improvements of 2.86 and 11.74, respectively. Our proposed approach also outperforms the state-of-the-art model on MCVQA dataset (*c.f.* Table 2) with considerable performance improvement of 5.07%. We could not observe a similar improvement on *en* language, because the LXMERT teacher model (*en*) is already pre-trained with the English VQA dataset.

It is to be noted that each monolingual model is trained separately with the respective language dataset and has a different model for each language setup. The results conclude two important claims: **(1)** effectiveness of knowledge distillation approach to handle MCM questions, and **(2)** scalability of our proposed single unified VQA model that can deal with questions from all the languages and their code-mixed setups.

We also perform the ablation study (*c.f.* Table 1) on different distillation objective functions. The results show that Object Attention Distillation ($\mathcal{L}_{object}$) is the most contributing objective function, removal of which leads to the 3.49% decrements in the overall average accuracy. We also observe the importance of the `CLS` Token Distillation ($\mathcal{L}_{CLS}$). This is the key loss function responsible to align the same multilingual and code-mixed questions in the vector space, and removing it leads to 1.68% decrements in overall average accuracy. Similarly, we observe 1.45% and 1.41% performance drops after the removal of $\mathcal{L}_{pred}$ and $\mathcal{L}_{nll}$ objective functions, respectively. The observed improvements over the multilingual baselines are statistically significant as $p < 0.05$ for the t-test using Dror et al. (2018). Please see the **Appendix** for additional results.

	Models	*bn*	*en-bn*	*de*	*en-de*	*es*	*en-es*	*fr*	*en-fr*	*hi*	*en-hi*	*en*	*Average*
Monolingual	LXMERT (Tan and Bansal, 2019)	60.74	64.95	67.95	70.52	68.66	**71.27**	68.43	71.12	59.83	69.95	**73.57**	67.90
	VL-BERT (Su et al., 2020)	58.53	61.30	64.29	65.33	64.79	66.09	64.72	65.84	59.40	65.28	67.30	63.89
	VisualBERT (Lu et al., 2019)	61.45	64.20	66.74	67.49	67.31	67.42	66.35	67.21	59.68	63.67	68.12	65.42
Multilingual	Joint LXMERT (Tan and Bansal, 2019)	48.44	60.07	58.02	62.79	58.41	63.07	58.34	62.96	50.28	61.52	65.41	59.02
	Joint LXMERT+ M-BERT	55.68	56.89	57.73	58.01	57.87	58.34	57.45	57.82	56.22	57.18	58.78	57.45
	Proposed Approach	**69.62**	**70.19**	**70.89**	**70.80**	**71.14**	71.11	**70.93**	**71.13**	**70.23**	**70.78**	71.66	**70.76**
	$-\mathcal{L}_{CLS}$	67.95	68.50	69.18	69.06	69.45	69.59	69.23	69.42	68.55	69.05	69.91	69.08
	$-\mathcal{L}_{object}$	66.02	66.56	67.30	67.32	67.60	67.74	67.32	67.56	66.66	67.13	68.80	67.27
	$-\mathcal{L}_{pred}$	68.17	68.77	69.40	69.32	69.65	69.88	69.42	69.62	68.77	69.30	70.11	69.31
	$-\mathcal{L}_{nll}$	68.17	68.83	69.50	69.41	69.70	69.88	69.48	69.47	68.80	69.35	70.28	69.35

Table 1: Performance comparison between the state-of-the-art baselines and our proposed model on the `MuCo-VQA` dataset. All the numbers are shown in % and denote the overall accuracy.

Models	*en*	*hi*	*en-hi*	*Average*
LXMERT (Tan and Bansal, 2019)	**73.02**	63.33	68.77	68.37
VL-BERT (Su et al., 2020)	67.28	59.32	63.28	63.29
VisualBERT (Li et al., 2019)	68.04	59.69	63.62	63.78
Gupta et al. (2020b)	65.37	64.51	64.69	64.85
Proposed Approach	71.37	**69.94**	**69.47**	**70.26**

Table 2: Performance comparison of different models on the MCVQA dataset.

Language	Number	Other	Yes/No	Overall
en	51.15	64.56	88.02	71.66
bn	50.62	62.16	85.97	69.62
en-bn	50.78	62.89	86.45	70.19
de	50.86	63.50	87.49	70.89
en-de	50.84	63.34	87.45	70.80
es	50.93	63.95	87.54	71.14
en-es	50.94	64.19	87.68	71.11
fr	50.95	63.47	87.61	70.93
en-fr	51.01	63.84	87.58	71.13
hi	50.72	62.57	87.01	70.23
en-hi	50.95	63.30	87.43	70.78

Table 3: Performance of our proposed model on different answer types across all the language setups in `MuCo-VQA` dataset

Figure 2: Performance comparison of different models for question understanding by varying the partial question as input to the model.

5.5 Discussion and Analysis

Behavior Analysis: We analyze the behavior of our proposed VQA model along the following dimensions:

(a) Question Understanding: Motivated from Agrawal et al. (2016), we analyze the performance of the model as a function of partial question length to establish the fact that the proposed model is more sensitive to MCM questions as compared to other pre-trained models. To examine this, we fed the LXMERT (monolingual), Joint-LXMERT (multilingual), and the proposed model (multilingual) with partial questions in the range of 20 to 100%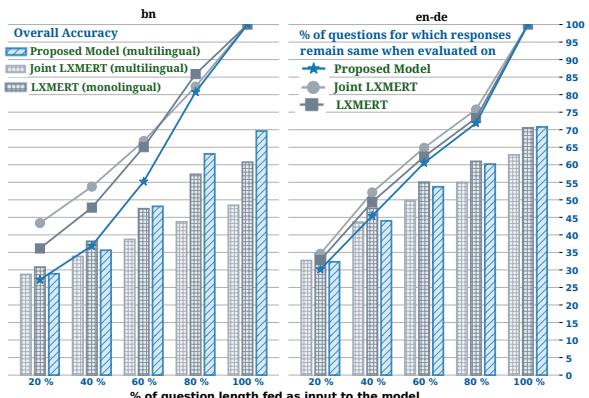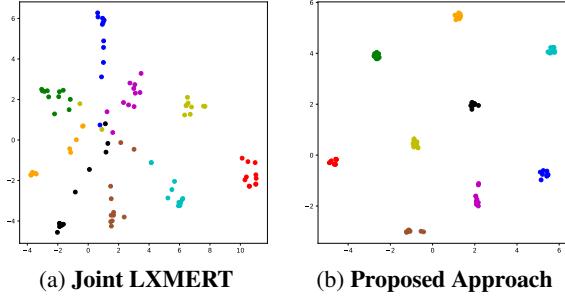 in an incremental manner. We observe (*c.f.* bar chart in Fig. 2) that our proposed model does not jump to quick conclusions by looking at partial questions as the overall accuracy is comparatively low for the proposed model for the incomplete questions. However, with full questions, the accuracies are high for the proposed model indicating that the model is sensitive to questions in different languages.

Furthermore, we also analyze what percentage of answers do not change when the partial questions are provided as input to the model. We can observe from the line chart of Fig. 2 that our proposed model is capable to change the answers when more question words are received as input to the model, unlike the LXMERT and Joint LXMERT model where the answers remain the same for around 50% of the questions. Additionally, to assess the role of syntax and semantics of the multilingual input questions, we analyze the performance of the system by feeding the randomly shuffled questions in Fig 6. The results show that our model is capable to understand the question semantics.

(b) Alignment: We also analyze the alignment of

(a) **Joint LXMERT** (b) **Proposed Approach**

Figure 4: t-SNE visualization for MCM questions in all eleven language setups. For proposed approach (b), we observe that the question representations of the same questions (shown in the same color) in different languages are very close in vector space unlike the Joint LXMERT model (a).
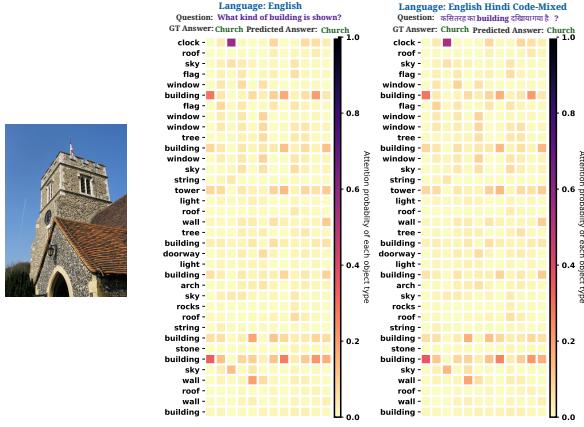

Figure 5: Heatmap of the learned attention weight for objects in the image from our proposed model. The proposed model is able to focus on the same object and correctly predict the answer irrespective of the language of the question. x-axis shows the heads of self-attention.
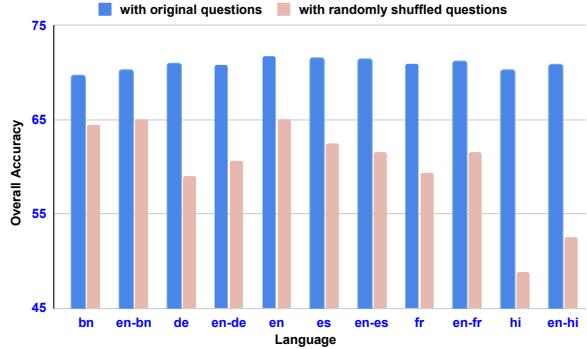

Figure 6: Performance comparison of the proposed model with shuffled questions and original questions

Figure 7: Zero-shot performance comparison of the proposed model on the different languages

the learned MCM question representation from our MCM Question Encoder. Towards this, we project the question representation (`[CLS]`) of the same question asked in different MCM settings using the t-SNE visualization (Van der Maaten and Hinton, 2008) in Fig 4. The plot shows that the question representations learned from the Joint LXMERT model are scattered in the vector space. In contrast, our proposed model learns the question representations which are very close in the vector space, indicating the capability of the model to learn the language-agnostic question representations, which help the model to correctly predict the answer of the MCM questions.

In addition, we also analyze the cross-modal alignment learned from our proposed model. Towards this, we plot the attention heatmap (*c.f.* Fig 5) from the cross-modal encoder (`X-Att`). We analyze that our proposed model is able to effectively learn the language-agnostic cross-modal represen-

tation, where the key objects from the images are attended to predict the correct answer for MCM questions. We also show (in **Appendix**) that the cross-modal representation learned from the proposed model is tightly coupled with the image and question as the attention to objects get change when the different questions are asked from the same image. Overall this analysis confirms that our model is not myopic to images and MCM questions to predict the answers.

(c) Zero-shot Capability: We also assess the zero-shot capability of our proposed model. Towards this, we perform the experiments on the six more languages, *viz.* Arabic (*ar*), Italian (*it*), Russian (*ru*), Urdu (*ur*), Polish (*pl*), and Portuguese (*pt*). We evaluate the performance of our proposed model in zero-shot manner on the 500 questions translated into the respective languages (using Google translation). We compare the performance (*c.f.* Fig 7) with the multilingual Joint LXMERT model. The proposed model achieves better overall accuracy compared to the Joint LXMERT model. This demonstrates the capability of our model on the unseen languages, which eventually confirms that the proposed distillation objectives have guided the student to learn the robust cross-modal

Figure 8: Examples from the various type of errors committed by our proposed model

representations. Please see the **Appendix** for the detailed qualitative analysis.

Error Analysis: We categorize the following major sources of errors by sampling 200 incorrectly predicted answers:

(a) Answer Specificity and Ambiguity (E1): This type of error occurs when the objects in the image can be interpreted in multiple ways based on their visual surroundings. In those cases, our model sometimes predicts the incorrect but semantically similar to the ground truth (GT) answer. For example, **Q1** in Fig. 8, the question is *"What is the man holding"*. Our model predicts the '*bird*' as the answer for all languages of the questions. However, the ground truth answer is '*Turkey*' which is more specific and semantically similar.

(b) Object Counting (E2): We observe that our proposed model sometime predicts the incorrect answer for the counting type questions. The example is shown as **Q2** of Fig 8.

(c) Character Recognition (E3): This type of error occurs when the answer to the MCM questions can only be predicted by recognizing the characters from the images. The example is shown in Fig. 8 (**Q3**), where the GT answer is '30' (speed limit) but the model predicts the incorrect answer '25' because it could not recognize the character written in the image.

(d) Spatial Interpretation (E4): Such errors occur when the model could not correctly interpret the spatial information in the image. The example is shown in Fig. 8 (**Q4**), where the model predicted the '*pillow*' as the answer instead '*broom*'.

(e) Answer Reasoning (E5): This type of error occurs for the question, which requires understanding the causal relationship or in-depth reasoning to correctly predict the answer. We show the example (Fig. 8 (**Q5**)), where to infer the age of the boy, the system has to establish the fact that *number of candles on the cake can determine the age*. There are

some other errors caused by parallel question alignment and translation of the questions. We found the error **E5** contributes to the maximum of 26.5%, **E1**: 23.5%, **E3**: 21%, **E2**: 16%, **E4**: 9% and other types of error contributes to 4% of the total errors.

6 Conclusion

In this paper, we have proposed a unified framework for multilingual and code-mixed VQA by distilling the knowledge from the monolingual language-vision pre-trained LXMERT model. To fully utilize the rich information from the question, image, and cross-modal encoders, we devise effective distillation objectives to encourages the student model to learn from the teacher through a multi-layer distillation process. To train and evaluate the proposed approach, we have created a large-scale `MuCo-VQA` dataset supporting eleven different MCM settings. Extensive experiments over the `MuCo-VQA` and MCVQA datasets demonstrate the effectiveness of our proposed approach.

Ethical Declaration

All the datasets used in this paper are publicly available. The dataset used in this paper is used only for the purpose of academic research. There are no ethical concerns associated with the research carried out here.

Acknowledgement

References

Aishwarya Agrawal, Dhruv Batra, and Devi Parikh. 2016. Analyzing the behavior of visual question answering models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1955–1960, Austin, Texas. Association for Computational Linguistics.

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. In *CVPR*, pages 6077–6086.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. VQA: Visual Question Answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2425–2433.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Xilun Chen and Claire Cardie. 2018. Unsupervised Multilingual Word Embeddings. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 261–270, Brussels, Belgium. Association for Computational Linguistics.

Jae Won Cho, Dong-Jin Kim, Jinsoo Choi, Yunjae Jung, and In So Kweon. 2021. Dealing with missing modalities in the visual question answer-difference prediction task through knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1592–1601.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Tuong Do, Thanh-Toan Do, Huy Tran, Erman Tjiputra, and Quang D Tran. 2019. Compact trilinear interaction for visual question answering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 392–401.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392. Association for Computational Linguistics.

Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648.

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. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 457–468. Association for Computational Linguistics.

Björn Gambäck and Amitava Das. 2014. On measuring the complexity of code-mixing. In *Proceedings of the 11th International Conference on Natural Language Processing, Goa, India*, pages 1–7. Citeseer.

Haoyuan Gao, Junhua Mao, Jie Zhou, Zhiheng Huang, Lei Wang, and Wei Xu. 2015. Are You Talking to a Machine? Dataset and Methods for Multilingual Image Question Answering. In *Advances in Neural Information Processing Systems*, pages 2296–2304.

Souvick Ghosh, Satanu Ghosh, and Dipankar Das. 2017. Complexity metric for code-mixed social media text. *Computación y Sistemas*, 21(4):693–701.

Hila Gonen and Yoav Goldberg. 2019. Language modeling for code-switching: Evaluation, integration of monolingual data, and discriminative training. 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 4175–4185, Hong Kong, China. Association for Computational Linguistics.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6904–6913.

Deepak Gupta, Asif Ekbal, and Pushpak Bhattacharyya. 2018a. A Deep Neural Network based Approach for Entity Extraction in Code-Mixed Indian Social Media Text. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Deepak Gupta, Asif Ekbal, and Pushpak Bhattacharyya. 2020a. A semi-supervised approach to generate the code-mixed text using pre-trained encoder and transfer learning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2267–2280, Online. Association for Computational Linguistics.

Deepak Gupta, Ankit Lamba, Asif Ekbal, and Pushpak Bhattacharyya. 2016a. Opinion mining in a code-mixed environment: A case study with government portals. In *Proceedings of the 13th International Conference on Natural Language Processing*, pages 249–258.

Deepak Gupta, Pabitra Lenka, Asif Ekbal, and Push-pak Bhattacharyya. 2018b. Uncovering code-mixed challenges: A framework for linguistically driven question generation and neural based question answering. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 119–130, Brussels, Belgium. Association for Computational Linguistics.

Deepak Gupta, Pabitra Lenka, Asif Ekbal, and Push-pak Bhattacharyya. 2020b. 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*, pages 900–913, Suzhou, China. Association for Computational Linguistics.

Deepak Gupta, Shubham Tripathi, Asif Ekbal, and Pushpak Bhattacharyya. 2016b. A Hybrid Approach for Entity Extraction in Code-Mixed Social Media Data. *MONEY*, 25:66.

Deepak Gupta, Shubham Tripathi, Asif Ekbal, and Pushpak Bhattacharyya. 2017. SMPOST: Parts of Speech Tagger for Code-Mixed Indic Social Media Text. *arXiv preprint arXiv:1702.00167*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Drew A. Hudson and Christopher D. Manning. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. 2018. Bilinear Attention Networks. In *Advances in Neural Information Processing Systems 31*, pages 1571–1581.

Jin-Hwa Kim, Kyoung Woon On, Woosang Lim, Jeonghee Kim, Jung-Woo Ha, and Byoung-Tak Zhang. 2017. Hadamard product for low-rank bilinear pooling. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.

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. *arXiv preprint arXiv:1908.03557*.

Xiujun Li, Xi Yin, Chunyuan Li, Xiaowei Hu, Pengchuan Zhang, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. 2020. Oscar: Object-semantics aligned pre-training for vision-language tasks. *arXiv preprint arXiv:2004.06165*.

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 workshop*, volume 8. Barcelona, Spain.

Ilya Loshchilov and Frank Hutter. 2017. SGDR: stochastic gradient descent with warm restarts. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems*, pages 13–23.

Jonghwan Mun, Kimin Lee, Jinwoo Shin, and Bohyung Han. 2018. Learning to specialize with knowledge distillation for visual question answering. In *NeurIPS*, pages 8092–8102.

Carol Myers-Scotton. 1997. *Duelling languages: Grammatical structure in codeswitching*. Oxford University Press.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318. Association for Computational Linguistics.

Adithya Pratapa, Gayatri Bhat, Monojit Choudhury, Sunayana Sitaram, Sandipan Dandapat, and Kalika Bali. 2018a. Language modeling for code-mixing: The role of linguistic theory based synthetic data. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1543–1553.

Adithya Pratapa, Monojit Choudhury, and Sunayana Sitaram. 2018b. Word Embeddings for Code-Mixed Language Processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3067–3072. Association for Computational Linguistics.

Khyathi Chandu Raghavi, Manoj Kumar Chinnakotla, and Manish Shrivastava. 2015. Answer ka type kya he? Learning to Classify Questions in Code-Mixed Language. In *Proceedings of the 24th International Conference on World Wide Web*, pages 853–858. ACM.

Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. 2012. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision*, pages 746–760. Springer.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of association for machine translation in the Americas*, volume 200.

Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. 2020. Vl-bert: Pre-training of generic visual-linguistic representations. In *International Conference on Learning Representations*.

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.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Genta Indra Winata, Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2018. Code-Switching Language Modeling using Syntax-Aware Multi-Task Learning. In *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*, pages 62–67.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Zhou Yu, Jun Yu, Jianping Fan, and Dacheng Tao. 2017. Multi-Modal Factorized Bilinear Pooling With Co-Attention Learning for Visual Question Answering. In *Proceedings of the IEEE International Conference on Computer Vision*.

Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason J Corso, and Jianfeng Gao. 2020. Unified vision-language pre-training for image captioning and vqa. In *AAAI*, pages 13041–13049.

Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. 2016. Visual7w: Grounded Question Answering in Images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4995–5004.

A   Multilingual and Code-Mixed VQA Dataset

A.1   `MuCo-VQA` Dataset Creation

We use the Indic-nlp-library[3] to tokenize the quesstions of the Indic languages and Moses based tokenizer[4] for remaining languages. Following, we learn the alignment matrix using the fast alignment technique proposed in Dyer et al. (2013). The alignment helps to select the words or phrases to be mixed in the code-mixed question. Thereafter, we construct the aligned phrases between the English and foreign language questions. We extract the PoS, named entity (NE), and noun phrase (NP) from the English questions, and mix them in the proper places of the corresponding Hindi questions. More specifically, we start with the NEs of types '*PER*', '*LOC*', and '*ORG*' in the English question, and replace the corresponding words in the foreign language questions with the detected NEs from the English question. Similarly, we replace the corresponding words in the foreign language questions with the detected NPs from the English question. Finally, we also follow the same for the PoS tags '*Adjective*'. We utilize the constructed phrase and alignment information to identify the appropriate places to insert English words in the foreign language questions.

A.2   Analysis

We compute the complexity of the generated code-mixed questions using the Code-Mixing Index (CMI) (Gambäck and Das, 2014), Switch Point Fraction (SPF) (Pratapa et al., 2018a; Gupta et al., 2020a) and Complexity Factor (CF) (Ghosh et al., 2017) for the entire code-mixed questions from `MuCo-VQA` dataset (Table 5) and aforementioned 500 questions. These are the standard metrics used in the literature to indicate the level of language

---

[3] https://github.com/anoopkunchukuttan/indic_nlp_library
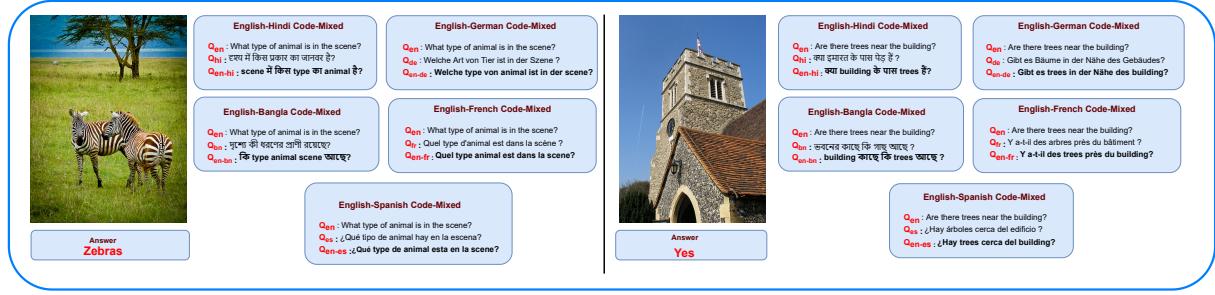[4] https://github.com/moses-smt/mosesdecoder

Figure 9: Sample questions (in multiple languages and code-mixed settings) with their corresponding images and answer from our `MuCo-VQA` dataset.

mixing in the code-mixed sentence. For the 500 questions, the mean values of the individual score obtained from each human expert are shown in Table 4. Our analysis shows that the code-mixed questions in `MuCo-VQA` dataset have similar CMI and SPF scores compared to the human formulated code-mixed questions. Similar observations are also made for the CF2 and CF3 metrics. The reported values in Table 4 also indicate that the automatically generated questions are slightly more complex (in terms of mixing the language) than the human-annotated code-mixed questions.

Metrics	BLEU	ROUGE-L	TER	CMI	SPF	CF2	CF3
`MuCo-VQA`	78.34	91.13	8.23	33.42	79.65	13.14	12.27
Human	NA	NA	NA	33.23	80.21	13.43	12.59

Table 4: Comparison of the generated code-mixed questions in terms of the level of code-mixing (CMI, SPF, CF2 and CF3) and quality of the generated code-mixed questions (BLEU, ROUGE-L and TER). Here, **NA**: Not applicable as the scores are computed against the human annotation itself.

B   Teacher Network

Learning Cross-Modality Encoder Representations from Transformers (LXMERT) (Tan and Bansal, 2019) is a pre-trained language model to learn the language-vision representation. It is built with the self-attention and cross-attention layers. The LXMERT model is pre-trained with a large amount of image-and-sentence pairs from VQA v2.0 (Goyal et al., 2017), GQA (Hudson and Manning, 2019), and VG-QA (Zhu et al., 2016) datasets. It is pre-trained on different tasks, such as *masked object prediction*, *masked language modeling*, *visual question answering*, and *cross-modality matching*.

Given a text and an image as inputs, LXMERT learns the language, image, and cross-modality (language-image) representations from the inputs. The language embedding is created using the word

and position embeddings followed by applying the layer normalization operation on the embeddings. The language encoder, which is composed of Transformer encoders takes the language embedding as input and generates the language representation. The image embedding is generated using the features of the detected objects from the image. Each detected object in the image is represented by its position and region-of-interest (RoI) features. The final image embedding is computed by averaging the revised position and RoI features using the layer normalization operation on the respective feature. The image embedding is passed into the image encoder, which is another transformer encoder. The cross-modality encoders are the stack of multiple encoder layers. Each encoder layer consists of two self-attention sub-layers, one bi-directional cross-attention sublayer, and two feed-forward sub-layers. The bi-directional cross-attention sub-layer contains one sub-layer from language to image and another from image to language.

C   Additional Implementation Details

To update the model parameters, we use the Adam (Kingma and Ba, 2015) optimization algorithm with the learning rate of $1e - 5$. We obtain the optimal hyper-parameter values based on the performance of the model on the validation set of `MuCo-VQA` dataset. We use a cosine annealing learning rate (Loshchilov and Hutter, 2017) decay schedule, where the learning rate decreases linearly from the initial rate set in the optimizer to 0. To avoid the gradient explosion issue, the gradient norm was clipped within 6. For doing the baseline experiments, we follow the official source code and train the model on the `MuCo-VQA` dataset. All the experiments are performed on a single GeForce GTX 1080 Ti GPU having GPU memory of 11GB. The average runtime (each epoch) for the proposed approach is 2.5 hrs.
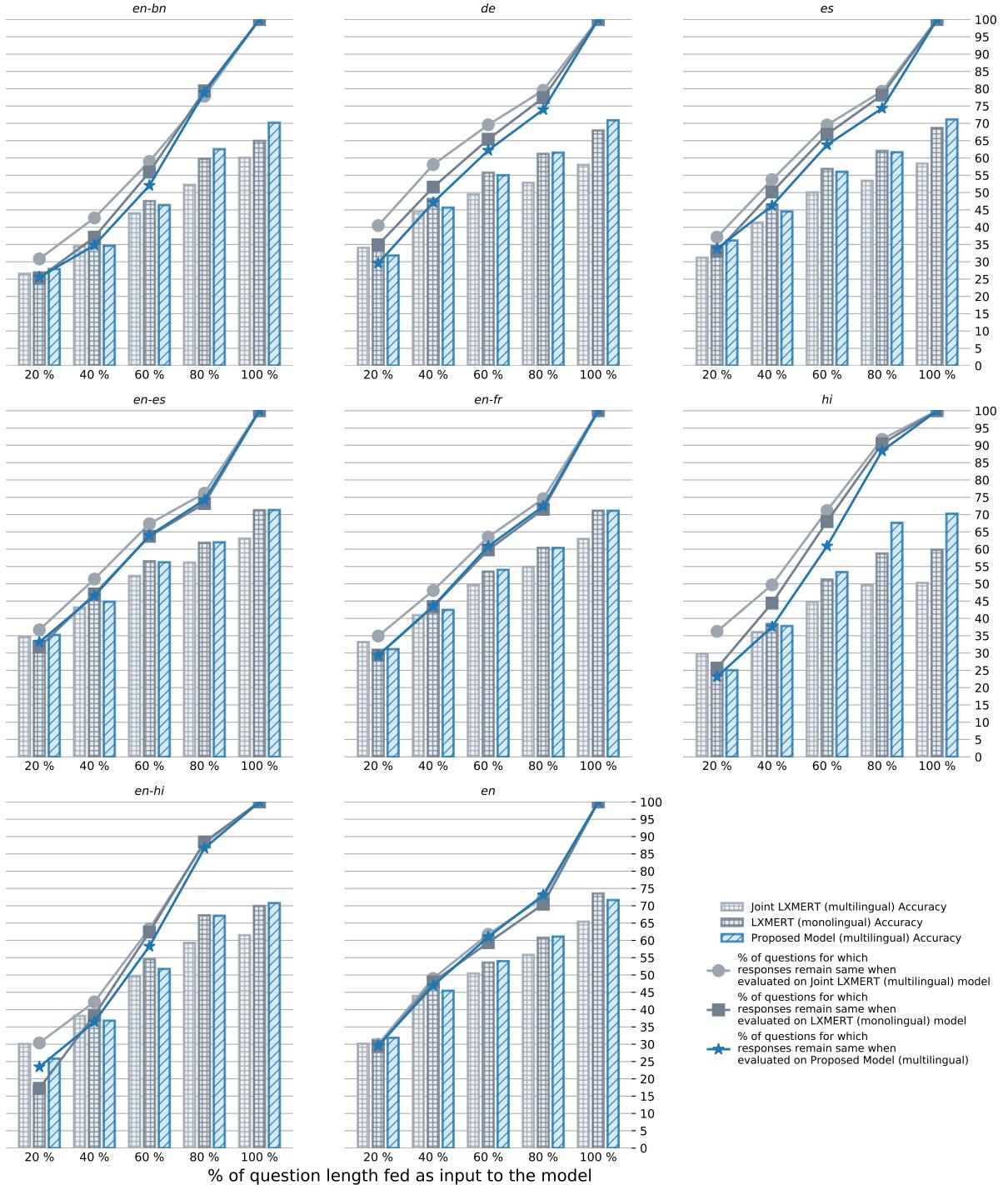
Figure 10: Performance comparison between the state-of-the-arts (LXMERT and Joint LXMERT) models and proposed model for question understanding by varying the partial question as input to the model.

Language Pairs	#Code-Mixed Question: Train	% of Code-Mixed	SPF	CMI	#Code-Mixed Question: Test	% of Code-Mixed	SPF	CMI
en-bn	243,203	97.93	92.47	35.65	118,989	97.92	92.21	36.14
en-de	242,854	97.79	81.22	33.96	118,895	97.85	81.46	34.05
en-es	234,570	94.45	74.80	31.69	114,747	94.43	74.80	31.70
en-fr	241,430	97.21	80.27	33.98	118,112	97.20	80.17	33.93
en-hi	242,963	97.83	78.35	32.82	118,935	97.88	78.54	32.80

Table 5: Statistics of generated code-mixed questions and along with the training and test set distributions. We also show the complexity of the generated code-mixed sentence in terms of SPF and CMI
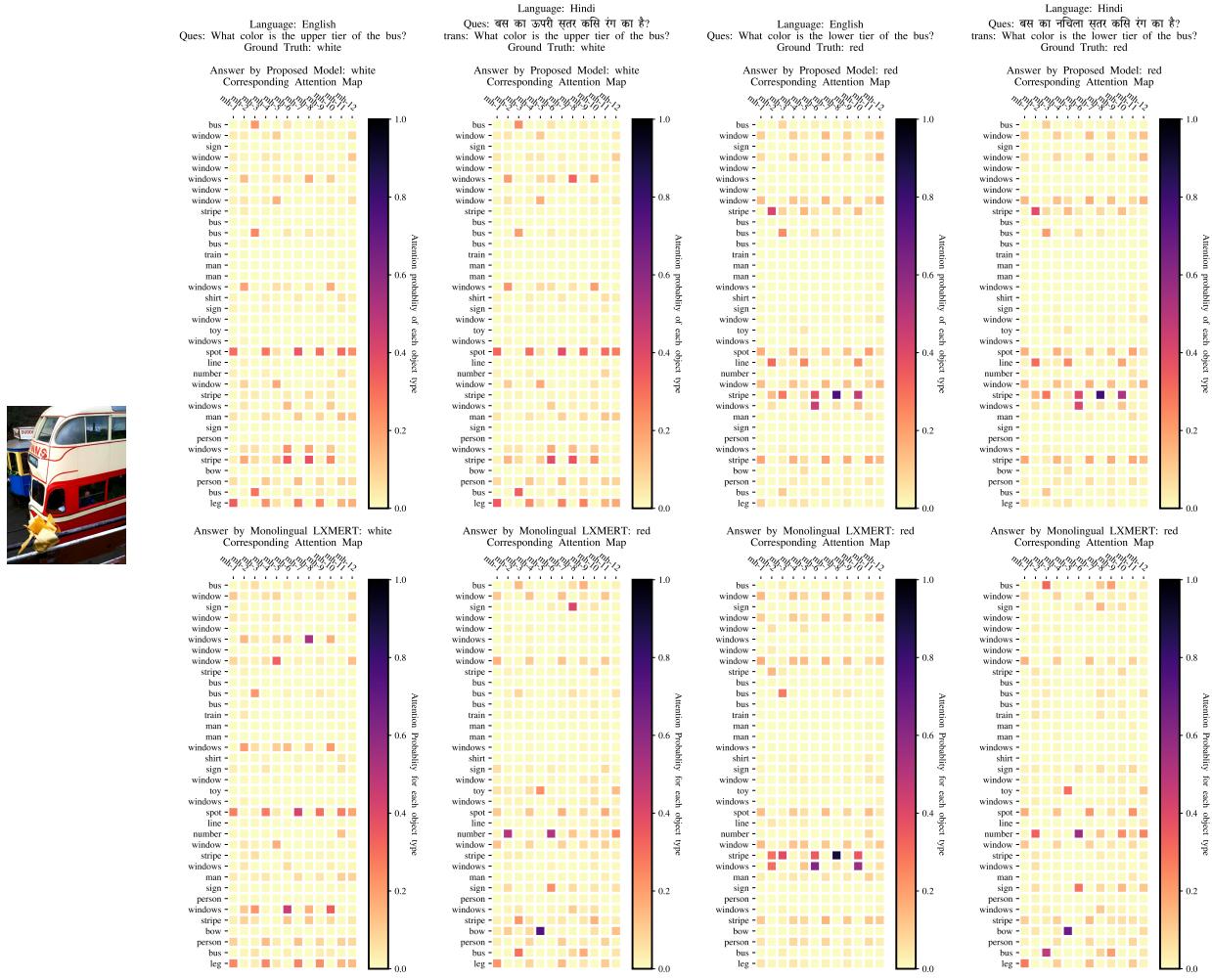
Figure 11: Heatmap of the learned attention weight for various objects in the image from our proposed model (Top) and LXMERT (Bottom). The proposed model is able to attend to the correct objects (the one attended by LXMERT when the English question is passed) in a language-agnostic way and hence predict the correct answer for MCM questions. However, the LXMERT monolingual model attends to the same objects and focuses only on the image giving same answers irrespective of the question. This shows the efficiency and robustness of the proposed model as it is sensitive to the question and maintains similar behavior across the languages.

Language	bn	en-bn	de	en-de	es	en-es	fr	en-fr	hi	en-hi	en	average
Validation Accuracy	72.98	73.51	74.08	73.88	74.29	74.52	74.25	74.43	73.42	74.13	74.86	74.03

Table 6: Performance of our proposed model on `MuCo-VQA` validation dataset of different languages.
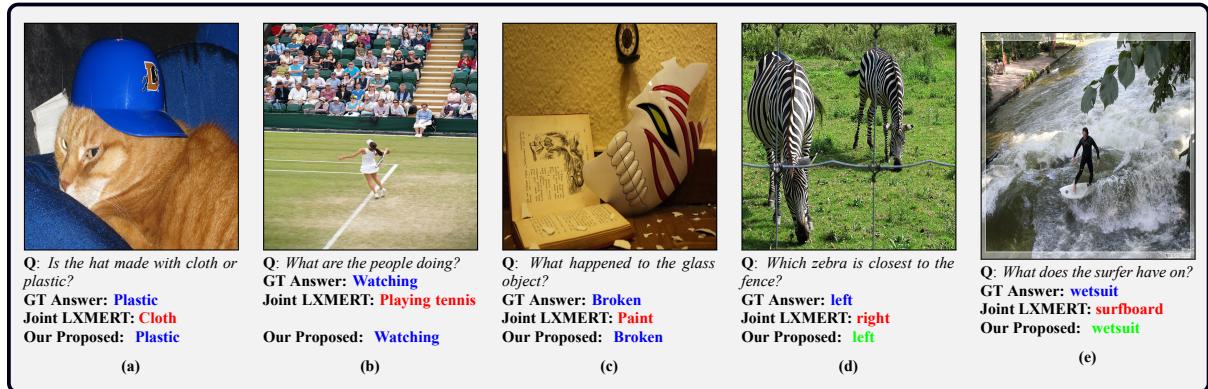

Figure 12: Sample questions where our proposed model perform better and correctly predict the answer compare to the multilingual Joint LXMERT model.