A Unified Framework for Multilingual and Code-Mixed Visual Question Answering

Deepak Gupta[‡], **Pabitra Lenka**[†*], **Asif Ekbal**[‡], **Pushpak Bhattacharyya**[‡]
[‡]Indian Institute of Technology Patna, India
[†]International Institute of Information Technology Bhubaneswar, India
[‡]{deepak.pcs16, asif, pb}@iitp.ac.in
[†]pabitra.lenka18@gmail.com

Abstract

In this paper, we propose an effective deep learning framework for multilingual and code-mixed visual question answering. The proposed model is capable of predicting answers from the questions in Hindi, English or Code-mixed (Hinglish: Hindi-English) languages. The majority of the existing techniques on Visual Question Answering (VQA) focus on English questions only. However, many applications such as medical imaging, tourism, visual assistants require a multilinguality-enabled module for their widespread usages. As there is no available dataset in English-Hindi VQA, we firstly create Hindi and Code-mixed VQA datasets by exploiting the linguistic properties of these languages. We propose a robust technique capable of handling the multilingual and code-mixed question to provide the answer against the visual information (image). To better encode the multilingual and code-mixed questions, we introduce a hierarchy of shared layers. We control the behaviour of these shared layers by an attention-based soft layer sharing mechanism, which learns how shared layers are applied in different ways for the different languages of the question. Further, our model uses bi-linear attention with a residual connection to fuse the language and image features. We perform extensive evaluation and ablation studies for English, Hindi and Code-mixed VQA. The evaluation shows that the proposed multilingual model achieves state-of-the-art performance in all these settings.

1 Introduction

Visual Question Answering (VQA) is a challenging problem that requires complex reasoning over visual elements to provide an accurate answer to a natural language question. An efficient VQA system can be used to build an Artificial Intelligence (AI) agent which takes a natural language question and predicts the decision by analyzing the complex scene(s). VQA requires language understanding, fine-grained visual processing and multiple steps of reasoning to produce the correct answer. As the existing research on VQA are mainly focused on natural language questions written in English (Antol et al., 2015; Hu et al., 2017; Fukui et al., 2016; Anderson et al., 2018; Li et al., 2018; Xu and Saenko, 2016; Shih et al., 2016), their applications are often limited.

Figure 1: Examples of questions (English, Hindi and Code-mixed) with their corresponding images and answers

Multilingual speakers often switch back and forth between their native and foreign (popular) languages to express themselves. This phenomenon of embedding the morphemes, words, phrases, etc., of one language into another is popularly known as code-mixing (Myers-Scotton, 1997, 2002). Code-mixing phenomena is common in chats, conversations, and messages posted over social media, especially in bilingual / multilingual countries like India, China, Singapore, and most of the other European countries. Sectors like tourism, food, education, marketing, etc. have recently started using code-mixed languages in their advertisements to attract their consumer base. In order to build an AI agent which can serve multilingual end users,

[*]Work carried out during the internship at IIT Patna

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
December 4 - 7, 2020.

a VQA system should be put in place that would be language agnostic and tailored to deal with the code-mixed and multilingual environment. It is worth studying the VQA system in these settings which would be immensely useful to a very large number of population who speak/write in more than one language. A recent study (Parshad et al., 2016) also shows the popularity of code-mixed English-Hindi language and the dynamics of language shift in India. Our current work focuses on developing a language agnostic VQA system for Hindi, English and code-mixed English-Hindi languages.

Let us consider the examples shown in Fig 1. The majority of the VQA models (Anderson et al., 2018; Li et al., 2018; Yu et al., 2018) are capable enough to provide correct answers for English questions $\mathbf{Q_E}$, but our evaluation shows that the same model could not predict correct answers for Hindi $\mathbf{Q_H}$ and Code-mixed question $\mathbf{Q_{CM}}$. The questions $\mathbf{Q_H}$ and $\mathbf{Q_{CM}}$ correspond to the same question $\mathbf{Q_E}$, but are formulated in two different languages. In this paper, we investigate the issue of multilingual and code-mixed VQA. We assume that there are several techniques available for monolingual (especially, English) VQA such that a strong VQA model can be built. However, we are interested in building a system that can answer the questions from different languages (multilingual) and the language formed by mixing up of multiple languages (code-mixed). We show that in a cross-lingual scenario due to language mismatch, applying directly a learned system from one language to another language results in poor performance. Thus, we propose a technique for multilingual and code-mixed VQA. Our proposed method mainly consists of three components. The first component is the *multilingual question encoding* which transforms a given question to its feature representation. This component handles the multilinguality and code-mixing in questions. We use multilingual embedding coupled with a hierarchy of shared layers to encode the questions. To do so, we employ an attention mechanism on the shared layers to learn language specific question representation. Furthermore, we utilize the self-attention to obtain an improved question representation by considering the other words in the question. The second component (*image features*) obtains the effective image representation from object level and pixel level features. The last component is *multimodal fusion* which is accountable to encode the question-image pair representation by ensuing that the learned representation is tightly coupled with both the question (language) and image (vision) feature.

It is to be noted that designing a VQA system for each language separately is computationally very expensive (both time and cost), especially when multiple languages are involved. Hence, an end-to-end model that integrates multilinguality and code-mixing in its components is extremely useful. We summarize our contribution as follows:

1. We create linguistically-driven Hindi and English-Hindi code-mixed VQA datasets. To the best of our knowledge, this is the very first attempt towards this direction.
2. We propose a unified neural model for multilingual and code-mixed VQA, which can predict answer of a multilingual or code-mixed question.
3. To effectively answer a question, we enhance the vision understanding by combining local image grid and object-level visual features. We propose a simple, yet powerful mechanism based on soft-sharing of shared layers to better encode the multilingual and code-mixed questions. This bridges the gap between VQA and multilinguality.
4. We perform extensive evaluation and ablation studies for English, Hindi and Code-mixed VQA. The evaluation shows that our proposed multilingual model achieves state-of-the-art performance in all these settings.

2 Related Work

Multilingual and Code-Mixing: Recently, researchers have started investigating methods for creating tools and resources for various Natural Language Processing (NLP) applications involving multilingual (Garcia and Gamallo, 2015; Gupta et al., 2019; Agerri et al., 2014) and code-mixed languages (Gupta et al., 2018a; Bali et al., 2014; Gupta et al., 2016; Rudra et al., 2016; Gupta et al., 2014). Developing a VQA system in a code-mixed scenario is, itself, very novel in the sense that there has not been any prior research towards this direction.

VQA Datasets: Quite a few VQA datasets (Gao et al., 2015; Antol et al., 2015; Goyal et al., 2017; Johnson et al., 2017; Shimizu et al., 2018; Hasan et al., 2018; Wang et al., 2018) have been created to encourage multi-disciplinary research involving Natural Language Processing (NLP) and

Computer Vision. In majority of these datasets, the images are taken from the large-scale image database MSCOCO (Lin et al., 2014) or artificially constructed (Antol et al., 2015; Andreas et al., 2016; Johnson et al., 2017). There are a few datasets (Gao et al., 2015; Shimizu et al., 2018) for multilingual VQA, but these are limited only to some chosen languages, and unlike our dataset they do not offer any code-mixed challenges.

VQA Models: The popular frameworks for VQA in the literature are built to learn the joint representation of image and question using the attention mechanism (Kim et al., 2018; Lu et al., 2016; Yu et al., 2017; Kafle and Kanan, 2017; Zhao et al., 2017). Hu et al. (2018) proposed a technique to separately learn the answer embedding with best parameters such that the correct answer has higher likelihood among all possible answers. There are some works (Chao et al., 2018; Liu et al., 2018; Wu et al., 2018) which exploit the adversarial learning strategy in VQA. VQA has also been explored in medical domains (Zhou et al., 2018; Gupta et al., 2021; Abacha et al., 2018; Ben Abacha et al., 2019). These learned representations are passed to a multi-label classifier whose labels are the most frequent answers in the dataset. Our analysis (c.f. Section 5.5) reveals that these models perform very poorly in a cross-lingual setting.

3 MCVQA Dataset

Dataset Creation: The popular VQA dataset released by Antol et al. (2015) contains images, with their corresponding questions (in English) and answers (in English). This is a challenging large scale dataset for the VQA task. To create a comparable version of this English VQA dataset in Hindi and code-mixed Hinglish, we introduce a new VQA dataset named "**M**ultilingual and **C**ode-mixed **V**isual **Q**uestion **A**nswering" (MCVQA) which comprises of questions in Hindi and Hinglish. Our dataset[1], in addition to the original English questions, also presents the questions in Hindi and Hinglish languages. This makes our MCVQA dataset suitable for multilingual and code-mixed VQA tasks. A sample of question-answer pairs and images from our dataset are shown in Fig 2.

We do not construct the answer in code-mixed language because a recent study (Gupta et al., 2018b) has shown that code-mixed sentences and their corresponding English sentences share the same nouns (common nouns, proper nouns, spatio-temporal nouns), adjectives, etc. For example, given an English and its corresponding code-mixed question:

$\mathbf{Q_E}$: *Where is the* ***tree*** *in this* ***picture****?*

$\mathbf{Q_{CM}}$: *Is* ***picture*** *me* ***tree*** *kahan hai?*

It can be observed that both $\mathbf{Q_E}$ and $\mathbf{Q_{CM}}$ share the same noun { ***picture***, ***tree***}. The majority of answers in the VQA v1.0 dataset are of type '*yes/no*', '*numbers*', '*nouns*', '*verbs*' and '*adjectives*'. Therefore, we keep the same answer in both English and Code-mixed VQA dataset.

We follow the techniques similar to Gupta et al. (2018b) for our code-mixed question generation, which takes a Hindi sentence as input and generates the corresponding Hinglish sentence as the output. We translate original English questions and answers using the Google Translate[2] that has shown remarkable performance in translating short sentences (Wu et al., 2016). We use this service as our original questions and answers in English are very short. For the code-mixed question generation, we first obtain the Part-of-Speech[3] (PoS) and Named Entity[4] (NE) tags of each question. Thereafter, we replace the Hindi words having the PoS tags (common noun, proper noun, spatio-temporal noun, adjective) with their best lexical translation. Same strategy is also followed for the words having the NE tags as *LOCATION* and *ORGANIZATION*. The remaining Hindi words are replaced with their Roman transliteration. In order to obtain the best lexical translation, we follow the iterative disambiguation algorithm (Monz and Dorr, 2005). We generate the lexical translation by training the Statistical Machine Translation (SMT) model on the publicly available English-Hindi (EN-HI) parallel corpus (Bojar et al., 2014). Please refer to the **Appendix** for the comparison with other VQA datasets.

Dataset Analysis: The MCVQA dataset consists of 248,349 training questions and 121,512 validation questions for real images in Hindi and Code-mixed. For each Hindi question, we also provide its 10 corresponding answers in Hindi. In order to analyze the complexity of the generated code-mixed questions, we compute the Code-mixing Index (CMI) (Gambäck and Das, 2014) and Complexity Factor

[1] The dataset can be found here: http://www.iitp.ac.in/~ai-nlp-ml/resources.html

[2] https://cloud.google.com/translate

[3] https://bit.ly/2rpNBJR

[4] https://bit.ly/2Qljan5

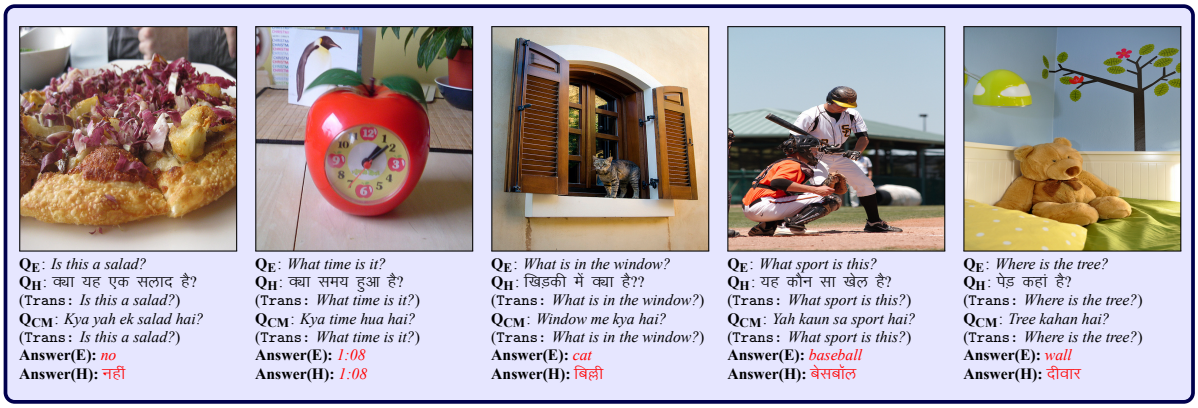


Figure 2: Sample questions (in English, Hindi and Code-Mixed) with their corresponding images and answers (in English, Hindi) from our MCVQA dataset

(CF) (Ghosh et al., 2017). These metrics indicate the level of language mixing in the questions. A detailed distribution of the generated code-mixed questions w.r.t to various metrics are in the **Appendix**.

We perform qualitative analysis by randomly selecting 5,200 questions from our MCVQA dataset. A bilingual (En, Hi) expert was asked to manually create the code-mixed questions and translate the English questions into Hindi. We compute the BLEU (Papineni et al., 2002), ROUGE (Lin, 2004) and Translation Error Rate (TER) (Snover et al., 2006) on the human translated questions and the translations obtained from the Google Translate. We achieve high BLEU and Rouge scores (BLEU 3: 80.22; ROUGE - L: 92.20) and lower TER (9.63).

4 Methodology for MVQA

Problem Statement: Given a natural language question $\mathcal{Q}$ in English, Hindi or code-mixed and a correlative image $\mathcal{I}$, the task is to perform a complex reasoning over the visual element of the image to provide an accurate natural language answer $\hat{\mathcal{A}}$ from all the possible answers $\mathcal{A}$. Mathematically:

$$\hat{\mathcal{A}} = \underset{\hat{\mathcal{A}} \in \mathcal{A}}{\arg\max}\, p(\hat{\mathcal{A}}|\mathcal{Q}, \mathcal{I}; \phi) \quad (1)$$

where ϕ is the network parameters. The architecture of our proposed methodology is depicted in Fig 3. Our proposed model has the following components:

4.1 Multilingual Question Encoding

Given a question[5] $Q = \{q_1, q_2, \ldots, q_T\}$ having T words, we obtain the multilingual embedding $q_t^e \in \mathbb{R}^d$ (c.f. Section 5.1) for each word $q_t \in Q$. The resulting representation is denoted by $\{q_t^e\}_{t=1}^T$. We use multilingual word-embedding to obtain the lower-level representation of the words from English, Hindi and English-Hindi code-mixed questions. However, only word-embedding is not capable enough to offer multilingual and code-mixing capability. For a better multilingual and code-mixing capability at a higher level, we introduce the shared encoding layers. In order to capture the notion of a phrase, first the embedded input $\{q_t^e\}_{t=1}^T$ is passed to a CNN layer. Mathematically, we compute inner product between the filter $F_l \in \mathbb{R}^{l \times d}$ and the windows of l word embedding. In order to maintain the length of the question after convolution, we perform appropriate zero-padding to the start and end of the embedded input $\{q_t^e\}_{t=1}^T$. The convoluted feature $q_t^{l,c}$ for l length filter is computed as follows:

$$q_t^{l,c} = tanh(F_l q_{t:t+l-1}^e) \quad (2)$$

A set of filters L of different window sizes is applied on the embedded input. The final output q_t^c at a time step t is computed by the max-pooling operation over different window size filters. Mathematically, $q_t^c = max(q_t^{l_1,c}, q_t^{l_2,c}, \ldots, q_t^{l_L,c})$. The final representation computed by CNN layer can be denoted as $\{q_t^c\}_{t=1}^T$. Inspired from the success in other NLP tasks (Luong et al., 2015; Yue-Hei Ng et al., 2015), we employ stacking of multiple Bi-LSTM (Hochreiter and Schmidhuber, 1997) layers to capture the semantic representation of an entire question. The input to the first layer of LSTM is the convoluted representation of the question $\{q_t^c\}_{t=1}^T$.

$$q_t^r = \text{Bi-LSTM}(q_{t-1}^r, q_t^c) \quad (3)$$

[5]It denotes the question in English, Hindi or Code-mixed

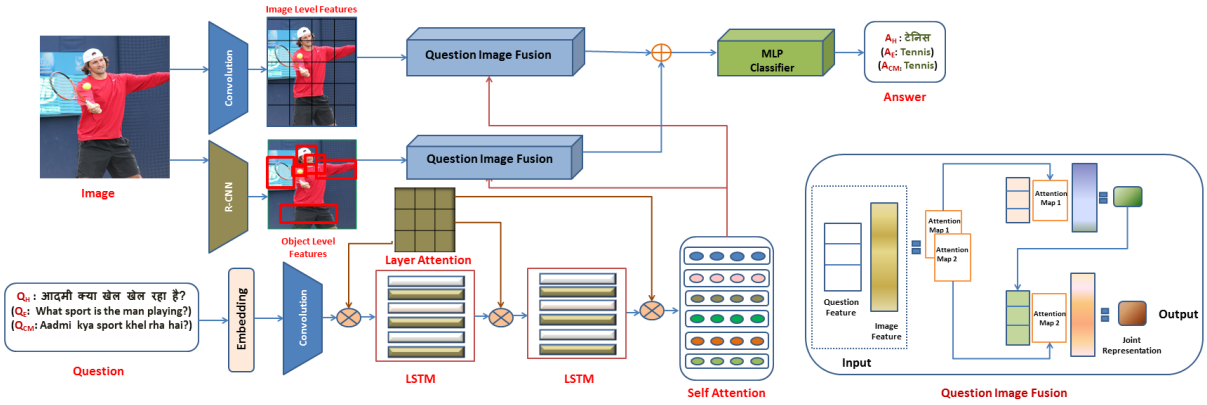


Figure 3: Architecture of the proposed multilingual VQA model. The input to the model is the multilingual question (one at a time). The bottom-right part of the image describes the *Question Image Fusion* component.

where, q_t^r and q_{t-1}^r are the hidden representations computed by the Bi-LSTM network at time t and $t-1$, respectively. Specially, we compute the forward $\overrightarrow{q_t^r}$ and backward hidden representation $\overleftarrow{q_t^r}$ at each time step t and concatenate them to obtain the final representation $q_t^r = \overrightarrow{q_t^r} \oplus \overleftarrow{q_t^r}$. The output from the previous layer of LSTM is passed as input to the next layer of LSTM.

4.1.1 Layer Attention

The encoding layers discussed in Section 4.1 are exploited by the questions from English, Hindi and English-Hindi code-mixed languages. It might not be the case that the representation of a question (in a given language) obtained from a particular encoding layer would also make a meaningful representation for the same question (in another language). In order to learn the language-specific control parameter for the encoding layer, we introduce an attention based mechanism over the encoding layer. Basically, our model learns an attentional vector over each encoder layer for each language. Our model learns a language importance weight matrix $W \in \mathbb{R}^{m \times n}$, where m and n correspond to the number of encoding layers and the number of different languages, respectively. The language importance weight matrix W is applied on a given language's (i) question representation in the j^{th} encoding layer. Let us assume that the j^{th} multilingual encoding layer generates the question representation: $Q^{i,j} = \{q_1^{i,j}, q_2^{i,j}, \ldots, q_T^{i,j}\}$. The language attentive representation for a language i and layer j is computed as follows:

$$\overline{q_t^{i,j}} = \overline{W_{i,j}} q_t^{i,j}, \quad t = \{1, 2, \ldots, T\}$$
$$\overline{W_{i,j}} = \frac{e^{-W_{i,j}}}{\sum_{k=1}^{n} e^{-W_{k,j}}} \quad (4)$$

The weighted question representation of i^{th} language obtained from the j^{th} layer can be denoted as $\overline{Q^{i,j}} = \{\overline{q_1^{i,j}}, \overline{q_2^{i,j}}, \ldots, \overline{q_T^{i,j}}\}$.

In our work, we use one layer of CNN and two layers of Bi-LSTM to encode multilingual questions. At each layer of encoding, we apply language specific weight to obtain the language specific encoding layer representation. We denote the question representation obtained from the final encoding layer after applying the language specific attention as $h = \{h_t\}_{t=1}^{T}$.

4.1.2 Self-Attention on Question

Inspired from the success of self-attention on various NLP tasks (Vaswani et al., 2017; Kitaev and Klein, 2018), we adopt self-attention to our model for better representation of a word by looking at the other words in the input question. The encoding obtained from multilingual encoding layer (c.f. Section 4.1.1) is passed to the self-attention layer. The multi-head self-attention mechanism (Vaswani et al., 2017) used in our model can be precisely described as follows:

$$Attention(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_h}})V \quad (5)$$

where, Q, K, V and d_h are the query, key, value matrices and dimension of the hidden representation obtained from the multilingual encoding layer, respectively. These matrices are obtained by multiplying different weight matrices to h. The value d_h is the dimension of the hidden representation obtained from the multilingual encoding layer. Firstly, multi-head attention linearly projects queries, keys and values to the given head (p) using different linear projections. These projections then perform

the scaled dot-product attention in parallel. Finally, these results of attention are concatenated and once again projected to obtain a new representation. Formally, attention head (z_p) at given head p can be expressed as follows:

$$\begin{aligned} z_p &= Attention(hW_p^Q, hW_p^K, hW_p^V) \\ &= softmax(\frac{(hW_p^Q)(hW_p^K)^T}{\sqrt{d_h}})(hW_p^V) \end{aligned} \quad (6)$$

where W_p^Q, W_p^K and W_p^V are the weight matrices. We exploit multiple heads to obtain the attentive representation. Finally, we concatenate all the attention heads to compute the final representation. The final question encoding obtained from the multilingual encoding layer can be represented by $U = \{q_1^h, q_2^h, \ldots, q_T^h\}$.

4.2 Image Features

Unlike the previous works (Fukui et al., 2016; Yu et al., 2017; Ben-Younes et al., 2017) on VQA, in this work we extract two different levels of features, *viz.* image level and object level. We employ ResNet101 (He et al., 2016) model pre-trained on ImageNet (Deng et al., 2009) to obtain the image level features $V_i \in \mathbb{R}^{d_i \times n_i}$, where n_i denotes the number of spatial location of dimension d_i. We take the output of pooling layer before the final softmax layer. To generate object level features, we use the technique as discussed in Anderson et al. (2018) by using Faster R-CNN framework (Ren et al., 2017). The resulting object level features $V_o \in \mathbb{R}^{d_o \times n_o}$ can be interpreted as ResNet features focused on the top-n_o objects in the image.

4.3 Multimodal Fusion

We fuse the multilingual question encoding (c.f. Section 4.1.2) and image features by adopting the attention mechanism described in Kim et al. (2018). Let us denote the question encoding feature by $U \in \mathbb{R}^{n_1 \times T}$ and the image feature by $V \in \mathbb{R}^{n_2 \times R}$. The k^{th} element representation using bi-linear attention network can be computed as follows:

$$f_k = (U^T X)_k^T \mathcal{M} (V^T Y)_k \quad (7)$$

where $X \in \mathbb{R}^{n_1 \times K}$, $Y \in \mathbb{R}^{n_2 \times K}$, $(U^T X)_k \in \mathbb{R}^T$, $(V^T Y)_k \in \mathbb{R}^R$ are the weight matrices and $\mathcal{M} \in \mathbb{R}^{T \times R}$ is the bi-linear weight matrix. The E.q. 7 computes the 1-rank bi-linear representation of two feature vectors. We can compute the K- rank bi-linear pooling for $f \in \mathbb{R}^K$. With K- rank bi-linear pooling, the bi-linear feature representation can be computed by multiplying a pooling vector $P \in \mathbb{R}^{K \times C}$ with f.

$$\overline{f} = P^T f \quad (8)$$

where C is the dimension of the bi-linear feature vector. The $\overline{f}$ is a function of U, V with the parameter (attention map) $\mathcal{M}$. Therefore, we can represent $\overline{f} = fun(U, V; \mathcal{M})$. Similar to Kim et al. (2018), we compute multiple bi-linear attention maps (called as visual heads) by introducing different pooling vectors. To integrate the representations learned from multiple bi-linear attention maps, we use the multi-modal residual network (MRN) (Kim et al., 2016). Using MRN, we can compute the joint feature representation in a recursive manner:

$$\overline{f_{j+1}} = fun_j(\overline{f_j}, V; \mathcal{M}_j).\mathbf{1}^T + \overline{f_j} \quad (9)$$

The base case $\overline{f_0}$ = U and $\mathbf{1} \in \mathbb{R}^T$ is the vector of ones. We extract the joint feature representation for image level $\overline{f}_i$ as well as object level feature $\overline{f}_o$.

4.4 Answer Prediction

Given the final joint representation of question with image level and object level features (c.f. Section 4.3), we augment both of these features to the counter feature (c_f) proposed in Zhang et al. (2018). The counter feature helps the model to count the objects. Finally, we employ a two-layer perceptron to predict the answer from a fixed set of candidate answers. It is predetermined from all of the correct answers in the training set that appear more than 8 times. To this end, the logits can be computed by the following equation:

$$A_{logits} = Relu\big(MLP(\overline{f}_i \oplus \overline{f}_o \oplus c_f)\big) \quad (10)$$

The A_{logits} is passed to a $softmax$ function to predict the answer.

5 Experimental Setup and Results

5.1 Datasets and Network Training

In our experiments, we use the VQA v1.0 dataset for English questions. There isn't a single setup for a multilingual VQA system which can handle both multilingual and code-mixed questions at the same time. Therefore, our primary motivation has been to set up a basic VQA system using the VQA v1.0 dataset. For Hindi and Code-mixed questions, we use our own multilingual VQA dataset (c.f. Section

3). Both the datasets have 248, 349 and 121, 512 questions in their training and test set, respectively. Each question has 10 answers. The test dataset of English VQA does not have publicly available ground truth answers. In order to make a fair comparison of the results in all the three setups, *viz.* English, Hindi and Code-mixed, we evaluate our proposed multilingual model on validation set of English and test set of Hindi and Code-Mixed dataset (MCVQA dataset).

The training is performed jointly with English, Hindi and Code-Mixed QA pairs by interleaving batches. We update the gradient after computing the loss of each mini-batch from a given language of sample (question, image, answer). The other baselines are trained and evaluated for each language separately. For evaluation, we adopt the accuracy metric as defined in Antol et al. (2015).

5.2 Hyperparameters

For English, we use the *fastText* (Bojanowski et al., 2016) word embedding of dimension 300. We use Hindi sentences from Bojar et al. (2014), and then train the word embedding of dimension 300 using the word embedding algorithm (Bojanowski et al., 2016). In order to obtain the embedding of Roman script, we transliterate[6] the Hindi sentence into the Roman script. These sentences are used to train the code-mixed embedding using the same embedding algorithm (Bojanowski et al., 2016), and we generate the embedding of dimension 300. These three word embeddings have the same dimensions but they are different in vector spaces. Finally, we align monolingual vectors of Hindi and Roman words into the vector space of English word embedding using the approach as discussed in Chen and Cardie (2018). While training, the model loss is computed using the categorical cross entropy function.

Optimal hyper-parameters are set to: maximum no. of words in a question=15, CNN filter size=$\{2, 3\}$, # of shared CNN layers=1, # of shared Bi-LSTM layers=2, hidden dimension =1000, # of attention heads=4, image level and object level feature dimension =2048, # of spatial location in image level feature =100, # of objects in object level feature=36, # of rank in bi-linear pooling=3, # of bilinear attention maps=8, # of epochs=100, initial learning rate=0.002. Optimal values of the hyperparameters are chosen based on the model performance on the development set of VQA v1.0 dataset. Adamax optimizer (Kingma and Ba, 2014) is used to optimize the weights during training.

[6] https://github.com/libindic/indic-trans

Dataset	Models	Overall	Other	Number	Yes/No
English	MFB	58.69	47.89	34.80	81.13
	MFH	59.07	48.04	35.42	81.73
	BUTD	63.50	54.66	38.81	83.60
	Bi-linear Attention	63.85	54.56	41.08	81.91
	Proposed Model	**65.37**	**56.41**	**43.84**	**84.67**
Hindi	MFB	57.06	46.00	33.63	79.70
	MFH	57.47	46.45	34.27	79.97
	BUTD	60.15	50.90	37.44	80.13
	Bi-linear Attention	62.50	52.99	40.31	82.66
	Proposed Model	**64.51**	**55.37**	**42.09**	**84.21**
Code-mixed	MFB	57.06	46.00	33.63	79.70
	MFH	57.10	46.09	33.56	79.71
	BUTD	60.51	51.68	36.47	80.37
	Bi-linear Attention	61.53	52.00	39.86	81.53
	Proposed Model	**64.69**	**55.58**	**42.57**	**84.28**

Table 1: Performance comparison between the state-of-the-art baselines and our proposed model on the VQA datasets. All the accuracy figures are shown in %. The improvements over the baselines are statistically significant as $p < 0.05$ for t-test. At the time of testing, only one language input is given to the model.

5.3 Results

In order to compare the performance of our proposed model, we define the following baselines: MFB (Yu et al., 2017), MFH (Yu et al., 2018), Bottom-up-Attention (Anderson et al., 2018) and Bi-linear Attention Network (Kim et al., 2018). These are the state-of-the-art models for VQA. We report the performance in Table 1.

The trained multilingual model is evaluated on the English VQA and MCVQA datasets as discussed in Section 5.1. Results of these experiments are reported in Table 1. Our proposed model outperforms the state-of-the-art English (with 65.37% overall accuracy), and achieves overall accuracy of 64.51% and 64.69% on Hindi and Code-mixed VQA, respectively. Due to the shared hierarchical question encoder, our proposed model learns complementary features across questions of different languages.

5.4 Comparison to the non-English VQA

Gao et al. (2015) created a VQA dataset for Chinese question-answer pairs and translated them to English. Their model takes the Chinese equivalent English question as input and generates an answer. A direct comparison in terms of performance is not feasible as they treat the problem as seq2seq learning (Sutskever et al., 2014) and their model was also trained on a monolingual (English) setup.

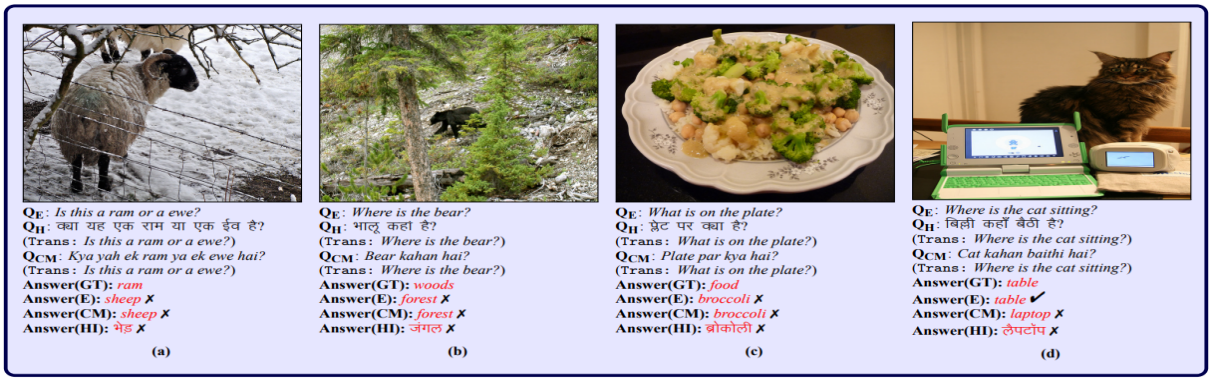


Figure 4: Some examples from MCVQA dataset where our model predicted incorrect answers. The notations are as follows:, **GT**: Ground Truth, **E**: English, **CM**: Code-mixed, **HI**: Hindi

Dataset	Training	Overall	Other	Number	Yes/No
English	English	63.85	54.56	41.08	83.91
Hindi		24.13	4.41	0.34	58.46
Code-mixed		28.04	11.95	0.49	58.79
English	Hindi	27.13	1.33	0.38	70.59
Hindi		62.50	52.99	40.31	82.66
Code-mixed		27.16	1.34	0.36	70.65
English	Code-mixed	30.92	9.45	0.39	69.85
Hindi		21.76	2.02	0.35	36.22
Code-mixed		61.53	52.00	39.86	81.53

Table 2: Results of cross-lingual experiments by training the Kim et al. (2018) model on the training dataset of one language and evaluating on the rest.

Models Component	English	Hindi	Code-mixed
Proposed	65.37	64.51	64.69
(-) CNN Layer	64.92	64.19	64.38
(-) Layer Attention	64.52	63.72	63.89
(-) Self Attention	64.31	63.59	63.63
(-) Image Level	64.88	63.97	64.10
(-) Object Level	64.29	63.39	63.52
(-) Counter	64.67	64.03	63.92
(-) Image (Language-only)	40.89	45.70	45.23
(-) Question (Vision-only)	24.13	26.44	21.68

Table 3: Effect of various components of the model in terms of overall accuracy on English, Hindi and Code-mixed VQA datasets. **(-) X** shows the VQA model architecture after removal of component '**X**'

We use their question encoding and language feature interaction component to train a model with English question and achieve overall accuracy of 57.89% on English validation dataset (our model achieves 65.37%). Recently, Shimizu et al. (2018) created a dataset for Japanese question-answer pairs and applied transfer learning to predict Japanese answers from the model trained on English questions. We adopt their approach, evaluate the model on English VQA and MCVQA dataset, and achieve 61.12%, 58.23%, 58.97% overall accuracy on English, Hindi and Code-mixed, respectively. In comparison to these, we rather solve a more challenging problem that involves both multilingualism and code-mixing.

5.5 Analysis and Discussion

We perform ablation study to analyze the contribution of various components of our proposed system. Table 3 shows the model performance by removing one component at a time. The self-attention on question and object-level features seem to have the maximum effect on the model's performance. The object-level features contribute more as compared to the image-level features because the object level-features focus on encoding the objects of an image, which assist in answering the questions more accurately. Image grid level features help the model to encode those parts of the image which could not be encoded by the object level features.

The proposed VQA model is built on two channels: vision (image) and language (question). We perform a study (Table 3) to know the impact of both the channels on the final prediction of the model. We turn off vision features and train the model with the textual features to assess the impact of vision (image) features. Similarly, we also measure the performance of the system with image features (object and image level) only. Our study provides answer to the following question: *"How much does a VQA model look at these channels to provide an answer?"*. The study reveals that the proposed VQA model is strongly coupled with both the vision and language channels. This confirms that the outperformance of the model is not because of the textual similarity between questions or pixel-wise similarity between the images.

We also perform experiments to evaluate the system in a cross-lingual setting. Towards this, we train the best baseline system (Kim et al., 2018) on the training dataset of one language and evaluate it on test datasets of the other two. The model performs pretty well when the languages for training and validation are the same. However, the performance of the model drops significantly when it is trained on one language and evaluated on a different language. We analyze the answers predicted by the model and make following observations: **(1)** Our model learns the question representation from different surface forms (English, Hindi and Hinglish) of the same word. It helps for much better representation of multilingual questions by encoding their linguistic properties. These rich information also interact with the image and extract language independent joint representation of question and image. However, the state-of-the-art models are language dependent. The question representation obtained from the state-of-the-art models could not learn language independent features. Therefore, they perform poorly in cross-lingual and multilingual setups (results are reported in Table 2). **(2)** We observe that the model performance on English VQA dataset is slightly better than Hindi and Code-mixed. One possible reason could be that the object-level features are extracted after training on the English Visual Genome dataset. Our VQA approach is language agnostic and can be extended to other languages as well.

Error Analysis: We perform a thorough analysis of the errors encountered by our proposed model on English VQA and `MCVQA` datasets. We categorize the following major sources of errors:
(i) Semantic similarity: This error occurs when an image can be interpreted in two ways based on its visual surroundings. In those scenarios, our model sometimes predicts the incorrect answer that is semantically closer to the ground truth answer. For example, in Figure 4(b), the question is *Where is the bear?*. Our model predicts the *forest* as the answer. However, the ground truth answer is *woods* which is semantically similar to *forest* and is a reasonable answer.
(ii) Ambiguity in object recognition: This error occurs when objects of an image have similar object and image-level features. For example, in Figure 4(a) the question is *Is this a ram or a ewe?*. Our model predicts *sheep* as the answer in all the three setups, but the ground truth answer is *ram*. As a *sheep*, a *ram* and an *ewe* have similar object and image-level features and all of them resemble the same, our model could not predict the correct answer in such cases.
(iii) Object detection at fine-grained level: This type of errors occur, when our model focuses on the fine-grained attributes of an image. In Figure 4(c), the question is *What is on the plate?*. The ground truth answer for this question is *food*. However, our model predicts *broccoli* as the answer. The food that is present on the plate is *broccoli*. This shows that our model is competent enough to capture the fine-grained characteristics of the image and thus predicts an incorrect answer.
(iv) Cross-lingual training of object-level features: Our proposed model has the capability to learn question features across multiple languages. However, the object-level features used in this work are trained on English language dataset (Visual Genome dataset). We observe (c.f. Figure 4(d)) that the model sometimes fails when the question is in Hindi or Hinglish.

6 Conclusion

In this work, we propose a unified end-to-end framework for multilingual and Code-mixed question answering and create a dataset for Hindi and Code-mixed VQA. We believe this dataset will enable the research in multilingual and code-mixed VQA. Our unified end-to-end model is capable of predicting answers for English, Hindi and Code-mixed questions. Experiments show that we achieve state-of-the-art performance on multilingual VQA. We believe our work will pave the way towards creation of multilingual and Code-mixed AI assistants. In the future, we plan to explore transformer-based architectures for VQA in multilingual and code-mixed setups considering various diverse languages.

Acknowledgment

Asif Ekbal gratefully acknowledges the Young Faculty Research Fellowship (YFRF) Award supported by the Visvesvaraya PhD scheme for Electronics and IT, Ministry of Electronics and Information Technology (MeitY), Government of India, and implemented by Digital India Corporation (formerly Media Lab Asia).

References

Asma Ben Abacha, Soumya Gayen, Jason J Lau, Sivaramakrishnan Rajaraman, and Dina Demner-Fushman. 2018. Nlm at imageclef 2018 visual question answering in the medical domain. In *CLEF (Working Notes)*.

Rodrigo Agerri, Josu Bermudez, and German Rigau. 2014. IXA Pipeline: Efficient and Ready to Use Multilingual NLP Tools. In *LREC*, pages 3823–3828.

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.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016. Neural Module Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 39–48.

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.

Kalika Bali, Jatin Sharma, Monojit Choudhury, and Yogarshi Vyas. 2014. "I am borrowing ya mixing?" An Analysis of English-Hindi Code Mixing in Facebook. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 116–126.

Asma Ben Abacha, Sadid A. Hasan, Vivek V. Datla, Joey Liu, Dina Demner-Fushman, and Henning Müller. 2019. VQA-Med: Overview of the medical visual question answering task at imageclef 2019. In *CLEF2019 Working Notes*, CEUR Workshop Proceedings, Lugano, Switzerland. CEUR-WS.org.

Hedi Ben-Younes, Rémi Cadene, Matthieu Cord, and Nicolas Thome. 2017. Mutan: Multimodal Tucker Fusion for Visual Question Answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2612–2620.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching Word Vectors with Subword Information. *arXiv preprint arXiv:1607.04606*.

Ondrej Bojar, Vojtech Diatka, Pavel Rychlỳ, Pavel Stranák, Vít Suchomel, Ales Tamchyna, and Daniel Zeman. 2014. HindEnCorp-Hindi-English and Hindi-only Corpus for Machine Translation. In *LREC*, pages 3550–3555.

Wei-Lun Chao, Hexiang Hu, and Fei Sha. 2018. Cross-Dataset Adaptation for Visual Question Answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5716–5725.

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.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A Large-Scale Hierarchical Image Database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255.

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.

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.

Marcos Garcia and Pablo Gamallo. 2015. Yet Another Suite of Multilingual NLP Tools. In *International Symposium on Languages, Applications and Technologies*, pages 65–75. Springer.

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

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 Languages Resources Association (ELRA).

Deepak Gupta, Asif Ekbal, and Pushpak Bhattacharyya. 2019. A Deep Neural Network Framework for English Hindi Question Answering. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 19(2):1–22.

Deepak Gupta, Pabitra Lenka, Asif Ekbal, and Pushpak 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. Association for Computational Linguistics.

Deepak Gupta, Swati Suman, and Asif Ekbal. 2021. Hierarchical deep multi-modal network for medical visual question answering. *Expert Systems with Applications*, 164:113993.

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

Deepak Kumar Gupta, Shubham Kumar, and Asif Ekbal. 2014. Machine Learning Approach for Language Identification & Transliteration. In *Proceedings of the Forum for Information Retrieval Evaluation*, pages 60–64. ACM.

Sadid A Hasan, Yuan Ling, Oladimeji Farri, Joey Liu, M Lungren, and H Müller. 2018. Overview of the ImageCLEF 2018 Medical Domain Visual Question Answering Task. In *CLEF2018 Working Notes. CEUR Workshop Proceedings, CEUR-WS., Avignon, France*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural computation*, 9(8):1735–1780.

Hexiang Hu, Wei-Lun Chao, and Fei Sha. 2018. Learning Answer Embeddings for Visual Question Answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5428–5436.

Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. 2017. Learning to Reason: End-to-end Module Networks for Visual Question Answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 804–813.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. 2017. CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1988–1997.

Kushal Kafle and Christopher Kanan. 2017. Visual Question Answering: Datasets, Algorithms, and Future Challenges. In *Computer Vision and Image Understanding*, volume 163, pages 3–20. Elsevier.

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, Sang-Woo Lee, Donghyun Kwak, Min-Oh Heo, Jeonghee Kim, Jung-Woo Ha, and Byoung-Tak Zhang. 2016. Multimodal Residual Learning for Visual QA. In *Advances In Neural Information Processing Systems 29*, pages 361–369.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*.

Nikita Kitaev and Dan Klein. 2018. Constituency Parsing with a Self-Attentive Encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2676–2686. Association for Computational Linguistics.

Qing Li, Qingyi Tao, Shafiq Joty, Jianfei Cai, and Jiebo Luo. 2018. VQA-E: Explaining, Elaborating, and Enhancing Your Answers for Visual Questions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 552–567.

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.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. In *European Conference on Computer Vision*, pages 740–755. Springer.

Yun Liu, Xiaoming Zhang, Feiran Huang, and Zhoujun Li. 2018. Adversarial Learning of Answer-Related Representation for Visual Question Answering. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1013–1022. ACM.

Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. 2016. Hierarchical Question-Image Co-Attention for Visual Question Answering. In *Advances In Neural Information Processing Systems*, pages 289–297.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421. Association for Computational Linguistics.

Mateusz Malinowski and Mario Fritz. 2014. A Multi-World Approach to Question Answering about Real-World Scenes based on Uncertain Input. In *Advances in Neural Information Processing Systems*, pages 1682–1690.

Christof Monz and Bonnie J Dorr. 2005. Iterative Translation Disambiguation for Cross-language Information Retrieval. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 520–527. ACM.

Carol Myers-Scotton. 1997. *Duelling Languages: Grammatical Structure in Codeswitching*. Oxford University Press.

Carol Myers-Scotton. 2002. *Contact Linguistics: Bilingual Encounters and Grammatical Outcomes*. Oxford University Press on Demand.

Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. 2012. Indoor segmentation and support inference from rgbd images. In *ECCV*.

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.

Rana D. Parshad, Suman Bhowmick, Vineeta Chand, Nitu Kumari, and Neha Sinha. 2016. What is India speaking? Exploring the "Hinglish" invasion. *Physica A: Statistical Mechanics and its Applications*, 449(C):375–389.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2017. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(6):1137–1149.

Koustav Rudra, Shruti Rijhwani, Rafiya Begum, Kalika Bali, Monojit Choudhury, and Niloy Ganguly. 2016. Understanding Language Preference for Expression of Opinion and Sentiment: What do Hindi-English Speakers do on Twitter? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1131–1141.

Kevin J Shih, Saurabh Singh, and Derek Hoiem. 2016. Where To Look: Focus Regions for Visual Question Answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4613–4621.

Nobuyuki Shimizu, Na Rong, and Takashi Miyazaki. 2018. Visual Question Answering Dataset for Bilingual Image Understanding: A Study of Cross-Lingual Transfer Using Attention Maps. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1918–1928.

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.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112.

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.

Peng Wang, Qi Wu, Chunhua Shen, Anthony Dick, and Anton van den Hengel. 2017. Explicit Knowledge-based Reasoning for Visual Question Answering. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 1290–1296.

Peng Wang, Qi Wu, Chunhua Shen, Anthony Dick, and Anton van den Hengel. 2018. FVQA: Fact-based Visual Question Answering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40:2413–2427.

Qi Wu, Peng Wang, Chunhua Shen, Ian Reid, and Anton van den Hengel. 2018. Are You Talking to Me? Reasoned Visual Dialog Generation through Adversarial Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6106–6115.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *CoRR*, abs/1609.08144.

Huijuan Xu and Kate Saenko. 2016. Ask, Attend and Answer: Exploring Question-Guided Spatial Attention for Visual Question Answering. In *European Conference on Computer Vision*, pages 451–466. Springer.

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*.

Zhou Yu, Jun Yu, Chenchao Xiang, Jianping Fan, and Dacheng Tao. 2018. Beyond Bilinear: Generalized Multi-modal Factorized High-order Pooling for Visual Question Answering. *IEEE Transactions on Neural Networks and Learning Systems*, (99):1–13.

Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and

George Toderici. 2015. Beyond Short Snippets: Deep Networks for Video Classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4694–4702.

Yan Zhang, Jonathon Hare, and Adam Prügel-Bennett. 2018. Learning to Count Objects in Natural Images for Visual Question Answering. In *International Conference on Learning Representations*.

Zhou Zhao, Qifan Yang, Deng Cai, Xiaofei He, and Yueting Zhuang. 2017. Video Question Answering via Hierarchical Spatio-Temporal Attention Networks. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3518–3524.

Yangyang Zhou, Xin Kang, and Fuji Ren. 2018. Employing inception-resnet-v2 and bi-lstm for medical domain visual question answering. In *CLEF (Working Notes)*.

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 Appendices

The detailed comparison of automatically created and manually code-mixed questions w.r.t the Code-mixing Index (CMI) score, Complexity Factor (CF2 and CF3) are shown in Table 4. We also show the comparison of our MCVQA dataset with other VQA datasets in Table 5. The analysis of MCVQA dataset are illustrated in Fig 5 and 6.

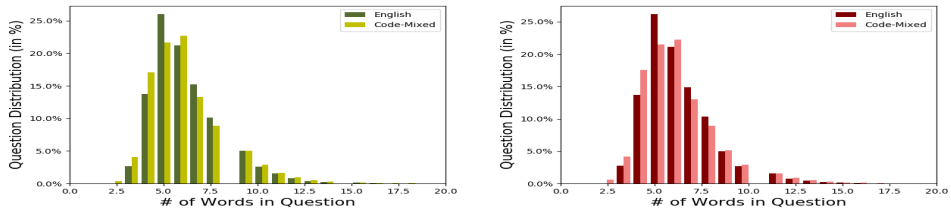
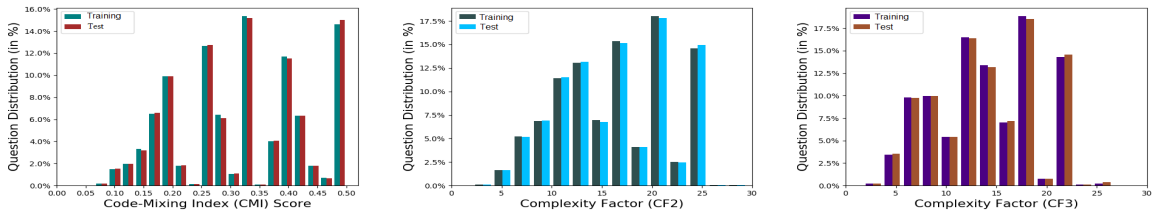


Figure 5: Analysis of question distribution w.r.t the question length between VQA v1.0 English and code-mixed, train and test dataset.



(a) (b) (c)

Figure 6: Analysis of code-mixed VQA dataset on various code-mixing metrics: **(a)**, **(b)** and **(c)** show the distribution of code-mixed questions from training and test set w.r.t the Code-mixing Index (CMI) score, Complexity Factor (CF2 and CF3), respectively.

Metrics	Manually Annotated	Automatically Generated	
		Training	Testing
CMI Score	0.2946	0.3223	0.3228
CF2	14.765	16.094	16.114
CF3	13.122	14.0708	14.096

Table 4: Comparison between manually annotated code-mixed questions and automatically generated code-mixed questions w.r.t the CMI score, CF2, and CF3.

Dataset	Images used	Created by	Multilingual	Code-Mixed
DAQUAR (Malinowski and Fritz, 2014)	NYU Depth V2	In-house participants, Automatically generated	✗	✗
FM-IQA (Gao et al., 2015)	MSCOCO	Crowd workers (Baidu)	✔	✗
VQA v1.0 (Antol et al., 2015)	MSCOCO	Crowd workers (AMT)	✗	✗
Visual7W (Zhu et al., 2016)	MSCOCO	Crowd workers (AMT)	✗	✗
CLEVR (Johnson et al., 2017)	Synthetic Shapes	Automatically generated	✗	✗
KB-VQA (Wang et al., 2017)	MSCOCO	In-house participants	✗	✗
FVQA (Wang et al., 2018)	MSCOCO	In-house participants	✗	✗
Japanese VQA (Shimizu et al., 2018)	MSCOCO	Crowd workers (Yahoo)	✔	✗
MCVQA (Ours)	**MSCOCO**	**Automatically generated**	✔	✔

Table 5: Comparison of VQA datasets with our MCVQA dataset. The images used are: MSCOCO (Lin et al., 2014) and NYU Depth v2 (Nathan Silberman and Fergus, 2012)

.