Testing Egunean Behin Visual Question Answering Dataset with BLIP

Julen Etxaniz
University of the Basque Country (UPV/EHU)
jetxaniz007@ikasle.ehu.eus

Abstract

Egunean Behin is a popular Basque quiz game. The game consists on answering 10 daily multiple choice questions. Questions were translated to English because VQA models like BLIP are mainly trained on English questions. Three types of questions from the game were selected: figures, cubes and maze. All the images and questions were generated automatically. There are multiple questions for each image. Questions require counting figures, colors, cubes and understanding the dimensions of the pictures. Each question has one correct and two wrong answers. These can be used for multiple choice question answering.

1 Introduction

Egunean Behin is a popular Basque quiz game. The game consists on answering 10 daily multiple choice questions. Questions were translated to English because state-of-the-art VQA models are mainly trained on English questions. Three types of questions from the game were selected: figures, cubes and maze. All the images and questions were generated automatically. There are multiple questions for each image. Questions require counting figures, colors, cubes and understanding the dimensions of the pictures. Each question has one correct and two wrong answers. These can be used for multiple choice question answering.

This dataset can be used to test VQA models in an out of domain setting. It could also be used to fine-tune a model to answer these types of questions if enough data is generated. The code can be used to generate as many images as necessary.

In this work we will test state-of-the-art BLIP (Li et al., 2022) model on this dataset in a zero-shot setting. This will show the difficulty of each question type and the ability of the model to generalize to different types of images.

2 Related Work

This section explains related work on vision language pre-training and dataset bootstrapping. These are important to understand the limitations of previous methods and the solutions proposed in BLIP (Li et al., 2022).

2.1 Vision-language Pre-training

Vision and language are two of the most fundamental methods for humans to perceive the world. An important goal of AI has been to build intelligent agents that can understand the world through vision and language inputs, and communicate with humans through language.

Vision-language pre-training has emerged as an effective approach to achieve this goal. Deep neural network models are pre-trained on large scale image-text datasets to improve performance on downstream vision-language tasks, such as image-text retrieval, image captioning, and visual question answering.

Models are commonly pre-trained before they are fine-tuned on each task. Fine-tuning involves additional training of the pre-trained model, using data from the downstream task. Without pre-training, the model needs to be trained from scratch on each downstream task, which leads to worse performance.

Despite the success of vision-language pre-training, existing methods have two major limitations related to models and training data.

From the model perspective, most existing pre-trained models are not flexible enough to adapt to a wide range of vision-language tasks. On the one hand, encoder-based models such as CLIP (Radford et al., 2021) and ALBEF (Li et al., 2021) are less straightforward to directly transfer to text generation tasks. On the other hand, encoder-decoder models like SimVLM (Wang et al., 2021) have not been successfully adopted for image-text retrieval tasks.

From the data perspective, most models are pre-trained on image and alt-text pairs that are automatically collected from the web. However, these web texts often do not accurately describe the images, making them a noisy source of supervision.

2.2 Bootstrapping Language-Image Pre-training

To address these limitations, BLIP: Bootstrapping Language-Image Pre-training (Li et al., 2022) introduces two contributions, one from each perspective.

On the one hand, Multimodal mixture of Encoder-Decoder (MED) is a new model architecture that enables a wider range of downstream tasks than existing methods. An MED can operate either as a unimodal image or text encoder, or an image-grounded text encoder, or an image-grounded text decoder.

On the other hand, Captioning and Filtering (CapFilt) is a new dataset bootstrapping method for learning from noisy web data. A captioner model produces synthetic captions given web images, and a filter model removes noisy captions from both the original web texts and the synthetic texts.

BLIP achieves state-of-the-art performance on five vision-language tasks: image-text retrieval, image captioning, visual question answering, visual reasoning, and visual dialog. It also achieves state-of-the-art zero-shot performance on two video-language tasks: text-to-video retrieval and video question answering.

3 Methods

This section first introduces the new model architecture MED and its pre-training objectives, then explains dataset bootstrapping and finally finetuning architecture.

3.1 Model Architecture

In order to pre-train a unified vision-language model with both understanding and generation capabilities, BLIP introduces a multimodal mixture of encoder-decoder (MED) model which can operate in three functionalities. The model architecture can be seen in Figure 1.

(1) Unimodal encoders are trained with an image-text contrastive (ITC) loss to align the image and text representations. The image encoder is a visual transformer (Dosovitskiy et al., 2020), which divides an input image into patches and encodes them as a sequence of embeddings. The text encoder is the same as BERT (Devlin et al., 2018), where a `[CLS]` token is appended to the beginning of the text input to summarize it.

(2) Image-grounded text encoder is trained with a image-text matching (ITM) loss to distinguish between positive and negative image-text pairs. It has a cross-attention layer between the self-attention layer and the feed forward layer for each transformer block. The output embedding of the `[Encode]` token is used as the multimodal representation of the image-text pair.

(3) Image-grounded text decoder is trained with a language modeling (LM) loss to generate captions for given images. It replaces the bi-directional self-attention layers in the text encoder with causal self-attention layers. A `[Decode]` token is used to as the beginning of a sequence.

In order to perform efficient pre-training and improve multi-task learning, the text encoder and text decoder share all parameters except for the self-attention layers. This is enough to capture differences between encoding and decoding tasks.

3.2 Dataset Bootstraping

As human-annotated image-text pairs are scarce, vision-language pre-training relies on large-scale image-text pairs automatically collected from the web. However, the texts often do not accurately describe the visual content of the image, making them a noisy supervision.

To address this, BLIP adds two modules, a captioner and a filter. The learning framework can be seen in Figure 2. Both the captioner and the filter are initialized from the same pre-trained MED model, and finetuned individually on the COCO (Lin et al., 2014) dataset.

The captioner is an image-grounded text decoder. Given the web images, it to generates synthetic captions as additional training samples. The filter is an image-grounded text encoder. It removes noisy captions which do not match their corresponding images. Filtered image-text pairs are combined with the human-annotated pairs to form a new dataset, which is used to pre-train a new model.

3.3 Finetuning

On each downstream task, different paths of the pre-trained model are finetuned to achieve different objectives. As the task of our dataset is visual question answering, we will mainly focus on that task.

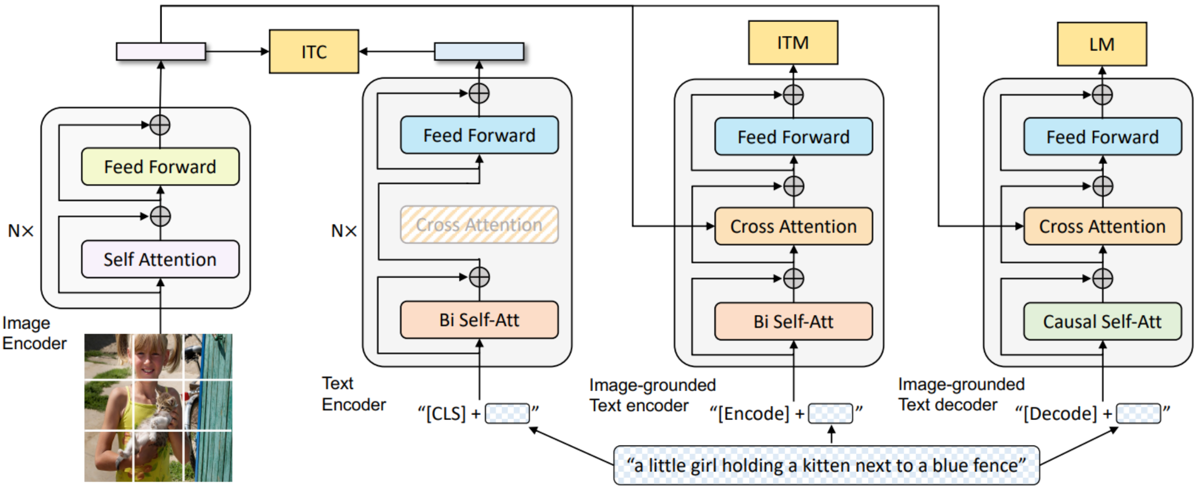
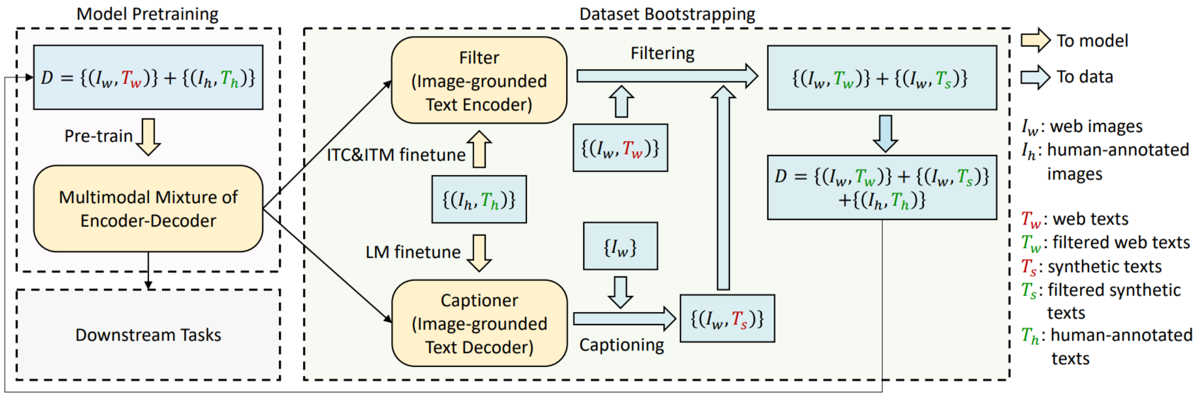


Figure 1: BLIP pre-training model architecture: multimodal mixture of encoder-decoder.



Model Pretraining
$D = \{(I_w, T_w)\} + \{(I_h, T_h)\}$
Pre-train
Multimodal Mixture of Encoder-Decoder
Downstream Tasks
Dataset Bootstrapping
Filter (Image-grounded Text Encoder)
ITC&ITM finetune
$\{(I_h, T_h)\}$
LM finetune
Captioner (Image-grounded Text Decoder)
Filtering
$\{(I_w, T_w)\}$
$\{I_w\}$
Captioning
$\{(I_w, T_s)\}$
$\{(I_w, T_w)\} + \{(I_w, T_s)\}$
$D = \{(I_w, T_w)\} + \{(I_w, T_s)\} + \{(I_h, T_h)\}$
To model
To data
I_w: web images
I_h: human-annotated images
T_w: web texts
T_w: filtered web texts
T_s: synthetic texts
T_s: filtered synthetic texts
T_h: human-annotated texts

Figure 2: BLIP learning framework: a captioner to produce synthetic captions and a filter to remove noisy captions.

VQA (Antol et al., 2015) is a popular vision and language task. Given an image and a question about the image, the task is to provide an accurate answer. VQA[1] dataset is commonly used as a benchmark to evaluate VQA systems. Questions are generally open-ended but multiple choices are provided for some questions. Visual Genome (Krishna et al., 2017) is another popular image-text dataset that was used to finetune the model.

The finetuning architecture for VQA can be seen in Figure 3. Image Encoder, Image-grounded Question Encoder and Answer Decoder are used for this task.

If we compare it to image captioning, VQA requires a more detailed understanding of the image and more complex reasoning (Antol et al., 2015). The finetuning architecture for image captioning can be seen in Figure 3. The architecture is simpler, only the Image Encoder and Image-grounded Text Decoder are needed.

4 Data

Three types of questions from Egunean Behin game were selected: figures, cubes and maze. There are multiple questions for each image. Questions require counting figures, colors, cubes and understanding the dimensions of the pictures.

All the images and questions were generated automatically in two steps. First, we generate as many images as we want. Then, we generate multiple questions for each image.

4.1 Figures

Images have geometric figures of different types and colors. Figures are selected randomly to create many different images. Images are saved to with a name that contains all the necessary data to create questions. The first two digits correspond to dimension of the image. The next digits correspond to the figures in each position of the image. For example, the name of the image in Figure 5 is figures_6_4_417148_466526_041585_724774.png.

18 questions of different types are created for each image. 3 questions about figure, column and row count. 3 questions about figure shape. 3 questions about figure color. 9 questions about figure shape and color combined. Table 1 shows example questions and answers for Figure 5.

Question	C	W1	W2
How many figures?	24	29	26
How many colums?	6	4	8
How many rows?	4	3	2
How many triangles?	6	5	8
How many squares?	9	7	11
How many circles?	9	11	8
How many red figures?	4	5	3
How many green figures?	13	15	17
How many blue figures?	7	9	10
How many red triangles?	1	3	0
How many green triangles?	3	4	1
How many blue triangles?	2	4	1
How many red squares?	0	1	2
How many green squares?	6	4	5
How many blue squares?	3	5	4
How many red circles?	3	5	4
How many green circles?	4	3	6
How many blue circles?	2	0	3

Table 1: Figures questions that correspond with the example image.

4.2 Cubes

See Figure 6.

Many questions of different types are created for each image. The number of questions depends on the dimenssions of the image. 3 questions about total, visible and non visible cubes. 4 questions about number of cubes in each x layer. 4 questions about number of cubes in each y layer. 3 questions about number of cubes in each z layer. Table 2 shows example questions and answers for Figure 6.

4.3 Maze

See Figure 7.

5 Results

6 Conclusions

References

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep

[1] https://visualqa.org

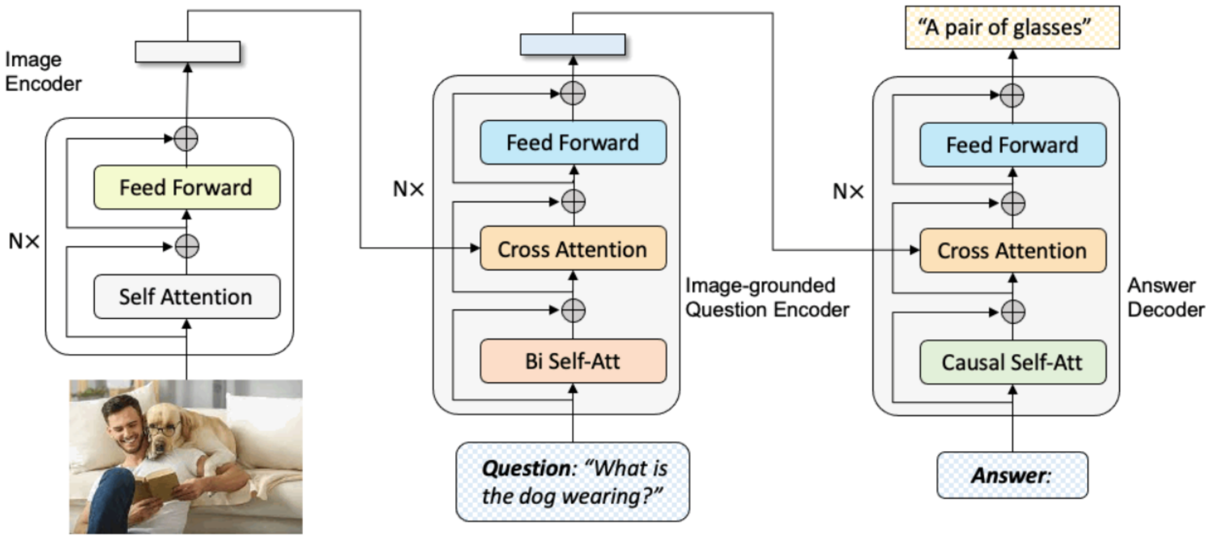


Figure 3: BLIP VQA finetuning architecture.

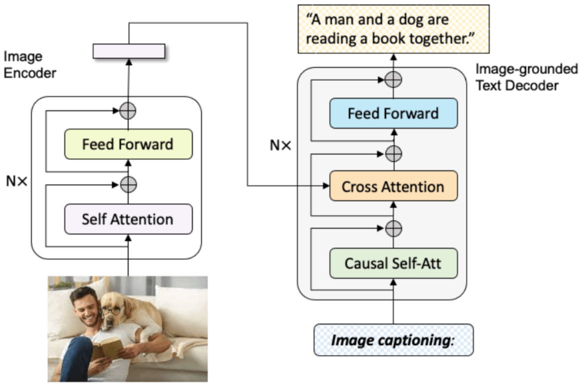


Figure 4: BLIP image captioning finetuning architecture.

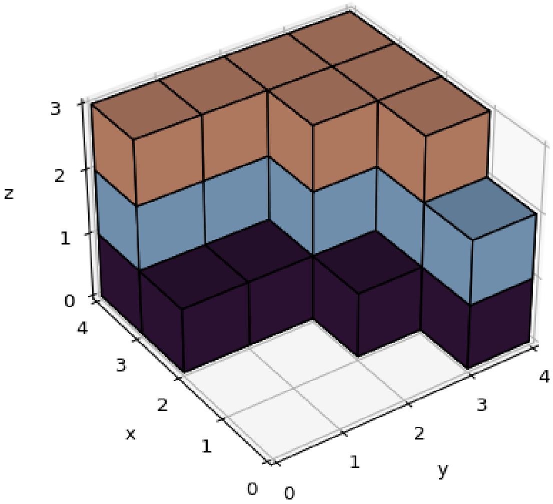


Figure 6: Cubes image example.

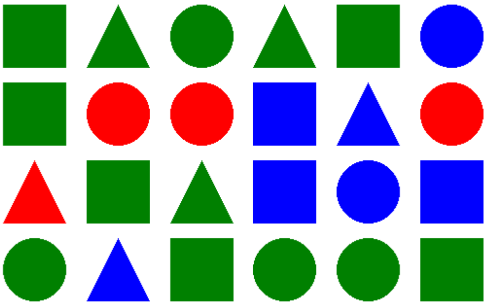


Figure 5: Figures image example.

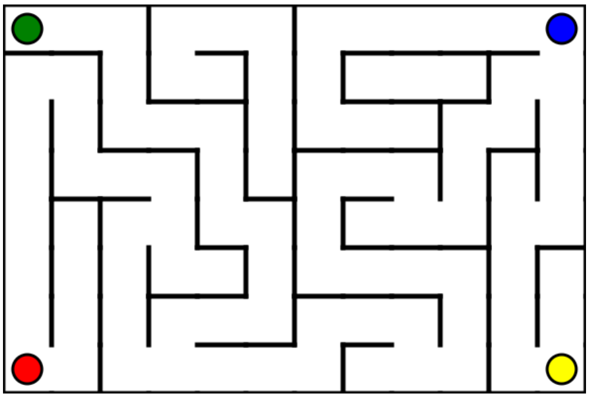


Figure 7: Maze image example.

Question	C	W1	W2
How many cubes in total?	26	22	21
How many visible cubes?	17	16	11
How many non visible cubes?	9	13	10
How many cubes in layer x 1?	2	0	1
How many cubes in layer x 2?	4	3	6
How many cubes in layer x 3?	8	11	10
How many cubes in layer x 4?	12	15	13
How many cubes in layer y 1?	4	6	7
How many cubes in layer y 2?	4	3	1
How many cubes in layer y 3?	7	9	8
How many cubes in layer y 4?	11	8	13
How many cubes in layer z 1?	11	10	12
How many cubes in layer z 2?	8	4	11
How many cubes in layer z 3?	7	11	10

Table 2: Cubes questions that correspond with the example image.

bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pretraining for unified vision-language understanding and generation. *arXiv preprint arXiv:2201.12086*.

Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. 2021. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in Neural Information Processing Systems*, 34.

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.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR.

Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. 2021. Simvlm: Simple visual language model pretraining with weak supervision. *arXiv preprint arXiv:2108.10904*.