VEQA: Visual Question Answering from the Lens of Visual Entailment

Meghana Holla
Department of Computer Science
Virginia Tech
Blacksburg, VA 24060
mmeghana@vt.edu

Surendrabikram Thapa
Department of Computer Science
Virginia Tech
Blacksburg, VA 24060
sbt@vt.edu

Abstract

Visual Question Answering (VQA) is an important multimodal task that tests a vision-language model in terms of both its recognition as well as reasoning capabilities. Similarly, Visual Entailment (VE) is a vision-language task that takes inspiration from the textual entailment problem under the natural language inference area of research. We propose ***VEQA***, a question-answering framework that reformulates the VQA task into a VE task. ***VEQA*** comprises an answer scorer and n hypothesis generator module to score each question-answer pair as if it were predicting whether or not the input image entails this pair. We perform experiments to analyze whether there is some potential in this reformulation and pave way for possible future work in this direction.

1 Introduction

Multimodal learning encompassing visual and textual domains is a fast-emerging research area and the need for enhanced cross-modal understanding and reasoning has heightened in the past few years. An important task in this direction is that of Visual Question Answering (VQA)[1], which is the task of answering a textual question based on an input visual context. Most often, this visual context is in the form of an image. VQA has important applications in downstream tasks such as multimodal information retrieval systems as well as making technology more accessible to visually challenged users. This elevates the importance of advancements in this area and calls for methodologies that achieve improved generalization and high performance.

In the field of logic, entailment is defined as the unidirectional relationship between two given statements which evaluates to true if the former implies the latter. In other words, the first statement entails the second if, when the first statement is true, there is enough evidence to conclude that the second one is also true[2]. These statements are termed premise and hypothesis, respectively. A popular adaptation of this concept is seen in Natural Language Processing (NLP), which is termed textual entailment. Textual entailment is a classification problem defined over a pair of sentences - the first sentence being the premise and the second the hypothesis. The objective of the task is to determine whether the premise contains enough evidence to imply the hypothesis. Hence, the task is to classify their relationship as being "entailment" or "contradiction" (or sometimes even "neutral" when the premise neither entails nor contradicts the hypothesis)[3].

Recently, this concept was extended to the multimodal setting, when [4] proposed Visual Entailment (VE). VE comprises of an image-text pair and the task is to predict whether the image entails the text or contradicts it. In other words, in VE, the image is the premise and the text is the hypothesis. The image entails the text if there is enough evidence to conclude the text is true or contradicts it otherwise. Occasionally, as mentioned above, the task may also include a “neutral" label to represent the image text pairs where the image neither entails nor contradicts the text sample.

Visual entailment requires tremendous cross-modal understanding capabilities and demands abilities that go much beyond simple recognition and enter the realm of reasoning. In NLP, textual entailment has been commonly recognized as a powerful means to enhance question answering [5]. We find that such parallels are absent in the vision-language context. The benefits of the visual entailment task motivate us to explore the possibilities of leveraging the entailment mechanism for question answering in a multimodal context. In other words, our main research question is - "Can we leverage visual entailment for the process of visual question answering?". In this work, we attempt to understand and analyze the effects of reformulating VQA as a VE task. We propose a framework called ***VEQA***: **V**isual **E**ntailment for visual **Q**uestion **A**nswering in an attempt to answer this question. We go over our approach in more detail in Section 3. Our code and slides[1] can be found at `https://github.com/meghana-holla/VEQA`. The contributions of this work can be summarized as follows:

1. We propose ***VEQA***: **V**isual **E**ntailment for visual **Q**uestion **A**nswering, a visual question answering framework that reformulates the task into a visual entailment problem, and simplifies the question answering task into that of determining whether the source image entails a given answer.

2. We perform experiments on various configurations to obtain a compositional understanding of the framework and the vision-language understanding as large and pave way for future work in this direction.

2 Related Work

Entailment has been used in many applications within NLP due to its ability to quantitatively measure how sentences are connected with each other[6]. White predominant applications of textual entailment (TE) have been used in fields like text summarization[7], information extraction [8] and sentiment analysis [9], some works explore the possibility of using TE for question answering[10, 11]. [10] introduces question entailment to better approach the task of question answering in textual context. [11] go over various ways in which TE can be applied to sub-components of question answering.

In question answering, entailment can be used to determine whether a given document contains the answer to a given question. This involves comparing the question and the document to see if the information in the document logically entails the answer to the question. For example, if the question is "What is the capital of France?" and the document contains the sentence "The capital of France is Paris," the system could use entailment to conclude that the document does indeed contain the answer to the question [12]. To determine whether the document entails the answer to the question, the system would need to have some way of representing the meaning of the text in the question and the document, as well as some way of determining whether one piece of text logically entails another. This could be done using natural language processing techniques, such as parsing the text into a logical form and then using a model of logical inference to determine whether the premise (the document) logically entails the hypothesis (the question)[11].

Visual entailment has independently been used for evaluating various pre-trained vision-language models but has never been considered for visual question answering particularly. The only work that draws some relation between visual question answering and visual entailment is by Si et al. [13], which proposed a framework where the candidate answers relevant to the question or image are selected and the candidate answers are re-ranked by a visual entailment task. This approach helped to explore the authenticity of answers by not ignoring the semantics of candidate answers. However, this works uses visual entailment module as an auxiliary module for re-ranking the answers, and still largely depends on a visual question answering module. In contrast, our method attempts to completely eliminate existing visual question answering approaches and only rely on visual entailment. We believe this is useful since it allows the model to reason about the image and the question in order to generate a correct answer.

[1] `https://github.com/meghana-holla/VEQA/blob/main/assets/Slides.pdf`

3 Approach

3.1 Task Definition

We denote the input image as I, the question as Q, and the corresponding answer set as $A = \{a_1, a_2, \ldots, a_n\}$, where a_i is the i^{th} answer choice and only one of the n choices is the correct answer. The VQA task can be defined as $f_{VQA}(I, Q, A)$, which is expected to output the correct answer a^*. As for the VE task, we denote an image-text pair as $< I, T >$, where I is the image and $T = \{t_1, t_2, \ldots, t_m\}$ is the text input and t_i is the i^{th} token. The visual entailment task would be denoted by $f_{VE}(I, T)$, where f_{VE} outputs the probability score of I entailing T.

In ***VEQA***, we employ f_{VE} to perform VQA by using the original image I as the premise and transforming a question-answer pair $< Q, a_i >$ to a sentence using a pre-defined function $f_{hyp}(Q, a_i)$, which we use as the hypothesis, h_i. In other words our VQA model f_{VQA} is as follows:

$$f_{VQA}(I, Q, A) = f_{VE}(I, f_{hyp}(Q, A))$$

where $f_{hyp}(Q, A) = \{f_{hyp}(Q, a_i)\ \forall\ a_i \in A\}$. The output of $f_{VE}(I, f_{hyp}(Q, A))$ is a score vector $S = \{s_i\ \forall\ i \in [1, n]\}$, where $s_i \in [0, 1]$ and the correct answer prediction $a^* = argmax_A(S)$.

3.2 Approach Overview

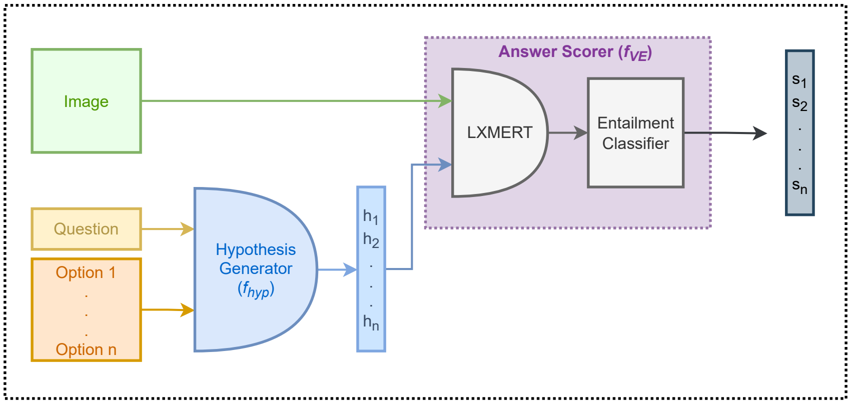

Figure 1: ***VEQA*** Framework for VQA. The question is passed with each answer option into the Hypothesis Generator, forming as many hypothesis statements. This is passed along with the input image, which becomes our premise. The answer scorer is an LXMERT model cascaded with a classfier head is used to predict the entailment probability, which is the score for that particular answer choice.

In an entailment setup, we assume that the premise image, I, would directly entail the right answer - i. e. , the premise image should have sufficient evidence to conclude that the hypothesis is true. In a similar vein, by definition, the premise image should not have enough evidence to entail the wrong answers, i. e. , the premise should either directly contradict the wrong answer hypotheses or should neither contradict nor entail them. For the scope of this project and for simplicity, we assume that the wrong answers would be directly contradicted by the premise image.

The ***VEQA*** framework mainly comprises two components - hypothesis generator (f_{hyp}) and answer scorer (f_{VE}). Figure 1 illustrates our framework's architecture. The hypothesis generator module is responsible for forming a natural language hypothesis from a question and a given answer choice. This module would hence output n hypotheses for every n answer choices of a given question. The answer scorer module is responsible for generating the answer scores. This module is designed to be a visual entailment (VE) mechanism. It should takes the input image I and the n hypothesis $H = \{h_i\ \forall\ i \in [1, n]\}$ from the hypothesis generator to output the answer scores, S for each question. Sections 3.2.1 and Section 3.2.2 explain the hypothesis generator and answer scorer module, respectively, in more detail.

3.2.1 Hypothesis Generator Module

For our hypothesis generator, we have used a rule-based hypothesis generator to generate sentences from the candidate answers. The multiple-choice questions have different candidate answers. The dataset also comes with question types. Each question thus has a question, candidate answers, and question type. There are a total of 65 question types. Each question only belongs to a single question type. For each question type, a set of rules is pre-defined so that the hypothesis (sentence formed with candidate answers) semantically matches the given question.

For example, if the question is "What color is the bird?" and the question type is "what color is the", the rule "The color is + <candidate answer>" would semantically make sense when candidate answers are semantically appropriate. The hypothesis thus generated can be "The color is red" with the option "red" and "The color is computer" if the candidate answer is "computer". The former would make much more sense. Earlier works do not take into account the semantics of candidate answers and hence the rule-based hypothesis generator can help to check the authenticity of answers.

3.2.2 Answer Scorer Module

We use our answer scorer to score each question-answer pair, $< Q, a_i >$ conditioned on the image I. In other words, this module would be responsible for directing the final decision of the correct answer choice. As previously mentioned, the underlying mechanism for this module is visual entailment, which predicts the answer score s_i for $< Q, a_i >$ as an entailment probability. The final answer would conform to the hypothesis h_i with the maximum answer score. Since this module needs to be capable of processing two distinct streams of data, ie., image and text, we employ a transformer-based vision-language model. Specifically, we adopt LXMERT proposed by [14] for our answer scorer. LXMERT is a popular vision-language model which is pre-trained using five versatile image-text objectives to ensure the enhancement of both intra-modal and cross-modal understanding [14]. Our answer scorer module is composed of an LXMERT instance followed by an entailment classifier head (as seen in Figure 1). A pre-trained LXMERT instance is employed, which is fine-tuned along with the classifier head for our answer-scoring purposes.

During training as well as inference, for every question, we pass the n image-hypothesis pairs generated by our framework as if each of these pairs were independent data examples. In other words, for a batch of size b, the model would receive $n * b$ data examples for the prediction of entailment probability. Following this, we extract these probabilistic outputs for every question to get the answer prediction for a given question (i. e. , group every n outputs together to get entailment probabilities/answer scores for a given question with n choices). Since our motivating rationale is that the image would only entail the correct answer choice and contradict the others, our ground truth entailment scores for the correct answer would be one (1.0) and the wrong answers would have a zero (0.0). We treat this problem as a binary classification problem, where an image-hypothesis pair would be assigned to the entailment/contradiction label.

4 Data

Our approach employs two major image-text datasets for this work. Firstly, we employ the Visual Question Answering dataset[1, 15]. This dataset is offered in two versions, but we turn to VQA v1 [15] since our work focuses on multiple choice answer questions. Additionally, we employ the SNLI-VE dataset for training some configurations of our answer scorer model for our experiments (Elaborated in Section 5.1). We briefly go over the statistics for VQA and SNLI-VE datasets:

Visual Question Answering (VQA v1): VQA v1 dataset comprises 204,721 images from the COCO[16] database spanning train, test and validation splits. Each of these images is paired with at least one question, summing up to 614,163 question annotations. Each question is mapped to a corresponding answer set annotations of size 18.

SNLI-VE: SNLI-VE comprises 31,783 images from the Flickr30k[17] database split between train, test and validation. The dataset contains a balanced distribution of textual annotations for each of the “entailment", “contradiction" and ‘neutral" tables with approximately $\sim$180,000 text-label samples under each class label.

Configuration	**Accuracy@1**	**Accuracy@2**	**Recall**	**Precision**
HG + $AS_{SNLI-VE}$	37.971%	55.128%	36.381%	36.597%
QA + $AS_{SNLI-VE}$	39.616%	**55.720%**	38.215%	38.167%
HG + AS_{VQA}	**50.927%**	51.612%	**51.065%**	**50.914%**
QA + AS_{VQA}	42.489%	45.758%	39.095%	39.221%

Table 1: Table showing ***VEQA*** performance on the four configurations of Hypothesis Generator and Answer Scorer. The best-performing model is displayed in bold and second-best is underlined. **HG** is the Hypothesis generation specified in Section 3.2.1, **QA** is the concatenation of question and answer, AS_{VQA} is answer scorer model trained with VQA and $AS_{SNLI-VE}$ with SNLI-VE.

5 Experiments and Analysis

5.1 ***VEQA*** Configurations

To get a better understanding of the individual components of ***VEQA***, this work tests ***VEQA*** in 4 different configurations (with two configurations of the hypothesis generator and answer score each). We describe the configurations for both the components below:

We test ***VEQA*** performance with regard to the hypothesis generator in two configurations - (1) With a simple concatenation of question and answer (**QA**), (2) With the generation mechanism specified in Section 3.2.1 (**HG**). Our major reason to pursue this direction is to examine the importance of proper sentence structure for using VE for VQA. In **QA** configuration, a simple concatenation of question and answer most often does not result in a properly-formed natural language answer. On the other hand, **HG** configuration appropriately merges the question with the answer to form a natural-sounding sentence(s).

For our answer scorer, we train our model on two different datasets - SNLI-VE [18] and VQA v1 [15]. Given that SNLI-VE was specifically curated for the entailment task, the data annotations are more representative of the entailment logic. Additionally, there is an equal distribution of entailment, contradiction and neutral examples, which does not cause any label imbalance. However, due to known issues with the neutral labels [19], we only consider the entailment and contradiction data examples. On the other hand, we also train ***VEQA*** on VQA dataset modified to fit the premise-hypothesis format. We believe training on this data is sensible since it causes minimum data drift between training and test datasets.

5.2 Experimental Setup

We train our ***VEQA*** model over 10 epochs with a maximum possible batch size of 8 owing to memory issues. To emulate real-world QA Systems, we limit our number of answer choices to 4. Hence, in addition to our ground truth answer, we randomly select 3 answers from the answer choice pool. Furthermore, we randomly allocate their positions in the ground truth answer vector.

We employ FasterRCNN [20] image features for our input images. Since LXMERT also asks for their corresponding bounding box annotations, we utilize the four-coordinate format bounding box annotations provided by FasterRCNN. We employ the LXMERT tokenizer, which uses byte pair encoding under the hood for tokenizing the hypothesis text.

For the training process, we employ ADAM optimizer with a learning rate of 10^{-5}. Since our VE task is a binary labeling tasks, we employ binary cross entropy loss.

5.3 Results and Discussion

Table 1 shows the results of our experiments, highlighting the performances of the four configurations specified in Section 5.1. We employ four metrics to analyze the performance of our models. Accuracy@k displays the top-k accuracy, which is the percentage of examples that ***VEQA*** correctly predicts in its top k predicted answers. Additionally, we also employ recall and precision measures.

We see that $HG + AS_{VQA}$ performs the best with the highest top-1 accuracy along with the highest recall and precision values. $QA + AS_{SNLI-VE}$ performs the best in terms of Accuracy@2. If we only look at the results with respect to the answer scorer configurations($AS_{SNLI-VE}$ vs. AS_{VQA}), the top 1 accuracy value for AS_{VQA} is consistently higher than that for $AS_{SNLI-VE}$ across both the hypothesis generator configurations. Furthermore, the increase in values from top1 to top2 accuracy is significant with AS with $AS_{SNLI-VE}$. However, the difference is not as apparent with Accuracy@2 values for AS_{VQA}. This observation holds weight since a big jump from Accuracy@1 to Accuracy@2 highlights an inherent model behavior to not place high confidence in a single answer observation, but to assign fairly uniform confidence to more than one choice - in other words, the model may be trying to play safe. These two observations combined highlight the robustness of the VQA configuration. However, this is counter-intuitive since the SNLI-VE dataset is carefully crafted to incorporate the entailment mechanism. Although this observation does confirm our hypothesis that the right answer always entails and the wrong ones are contradictory is correct and our results prove this fact since our reformatted VQA dataset with sharp 1.0 vs. 0 scoring for correct and wrong answers works well.

When the hypothesis generation modes are compared, our observations differ across the snail and via answer scorer variants. With $AS_{SNLI-VE}$, we see that the QA configuration seems to perform better than the HG mode, which unleashes an interesting insight into the internal working of the model - the inherent structure of the hypothesis sentence (question and answer joint representation) may not be as important as we think, instead, it is the information that is more important. However, with AS_{VQA}, we see the opposite trend. This uncovers two very important observations - answer scorer trained on SNLI-VE could be more adept at understanding the cross-modal nuances between the text and image, which lightens a single-handed dependency on either of the modalities, thereby rendering the structure of the text not as important as the information itself. On the same note, our second observation is that the VQA-trained model could have the tendency to tap into some linguistic biases, leading to a higher dependency on the sentence structure as supported by the higher performance in HQ configuration than QA.

6 Conclusion

We propose ***VEQA*** a visual question-answering framework that reforms the visual entailment. Through our various configurations, we learn about the inherent nature of the visual question-answering tasks in themselves and about the cross-modal understanding capabilities of the entailment task in a multimodal setting. Through our experiment and analysis, we show that the source images indeed entail the answer and contradict the wrong ones, which forms the basis for potentially simplifying the visa question-answering pipeline.

Future work can consider expanding the question answering to open-domain question answering. Furthermore, one can also consider modeling the entailment probabilities of a question alone conditioned on the image. This enables us to disentangle the answers and makes it possible to extend this work to open-domain question-answering areas.

References

[1] Peng Zhang, Yash Goyal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Yin and Yang: Balancing and answering binary visual questions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[2] Alan Ross Anderson and Nuel D. Belnap. *Entailment: The Logic of Relevance and Neccessity, Vol. I.* Princeton University Press, 1975.

[3] Roy Bar-Haim, Ido Dagan, and Idan Szpektor. *Benchmarking Applied Semantic Inference: The PASCAL Recognising Textual Entailment Challenges*, pages 409–424. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[4] Ning Xie, Farley Lai, Derek Doran, and Asim Kadav. Visual entailment task for visually-grounded language learning. *arXiv preprint arXiv:1811.10582*, 2018.

[5] Harsh Trivedi, Heeyoung Kwon, Tushar Khot, Ashish Sabharwal, and Niranjan Balasubramanian. Repurposing entailment for multi-hop question answering tasks, 2019.

[6] Manpreet Kaur and Dipti Srivastava. Text summarization using partial textual entailment based graphs. In *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, pages 366–374. IEEE, 2019.

[7] Naveen Saini, Sriparna Saha, Pushpak Bhattacharyya, and Himanshu Tuteja. Textual entailment–based figure summarization for biomedical articles. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 16(1s):1–24, 2020.

[8] Daniel Z Korman, Eric Mack, Jacob Jett, and Allen H Renear. Defining textual entailment. *Journal of the Association for Information Science and Technology*, 69(6):763–772, 2018.

[9] Shailja Gupta, Sachin Lakra, and Manpreet Kaur. Sentiment analysis using partial textual entailment. In *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, pages 51–55. IEEE, 2019.

[10] Asma Ben Abacha and Dina Demner-Fushman. A question-entailment approach to question answering. *BMC Bioinformatics*, 20(1), oct 2019.

[11] Aarthi Paramasivam and S. Jaya Nirmala. A survey on textual entailment based question answering. *Journal of King Saud University - Computer and Information Sciences*, 2021.

[12] Aarthi Paramasivam and S Jaya Nirmala. A survey on textual entailment based question answering. *Journal of King Saud University-Computer and Information Sciences*, 2021.

[13] Qingyi Si, Zheng Lin, Mingyu Zheng, Peng Fu, and Weiping Wang. Check it again: Progressive visual question answering via visual entailment. *arXiv preprint arXiv:2106.04605*, 2021.

[14] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019.

[15] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: Visual Question Answering. In *International Conference on Computer Vision (ICCV)*, 2015.

[16] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2014.

[17] Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models, 2015.

[18] Ning Xie, Farley Lai, Derek Doran, and Asim Kadav. Visual entailment: A novel task for fine-grained image understanding. *arXiv preprint arXiv:1901.06706*, 2019.

[19] Virginie Do, Oana-Maria Camburu, Zeynep Akata, and Thomas Lukasiewicz. e-snli-ve: Corrected visual-textual entailment with natural language explanations. 2020.

[20] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2015.