Explicit Bias Discovery in Visual Question Answering Models

Varun Manjunatha[1]*   Nirat Saini[2]   Larry S. Davis[2]

[1]Adobe Research   [2]University of Maryland, College Park

vmanjuna@adobe.com, {nirat@cs, lsd@umiacs}.umd.edu

Abstract

Researchers have observed that Visual Question Answering (VQA) models tend to answer questions by learning statistical biases in the data. For example, their answer to the question "What is the color of the grass?" is usually "Green", whereas a question like "What is the title of the book?" cannot be answered by inferring statistical biases. It is of interest to the community to explicitly discover such biases, both for understanding the behavior of such models, and towards debugging them. Our work address this problem. In a database, we store the words of the question, answer and visual words corresponding to regions of interest in attention maps. By running simple rule mining algorithms on this database, we discover human-interpretable rules which give us unique insight into the behavior of such models. Our results also show examples of unusual behaviors learned by models in attempting VQA tasks.

1. Introduction

In recent years, the problem of Visual Question Answering (VQA) - the task of answering a question about an image has become a hotbed of research activity in the computer vision community. While there are several publicly available VQA datasets[6, 23, 26, 29], our focus in this paper will be on the dataset provided in [6] and [18], which is the largest natural image-question-answer dataset and the most widely cited. Even so, the narrowed-down version of the VQA problem on this dataset is not monolithic - ideally, several different skills are required by a model to answer the various questions. In Figure 1(left) , a question like "What time is it?" requires the acquired skill of being able to read the time on a clock-face, "What is the title of the top book?" requires an OCR-like ability to read sentences, whereas the question "What color is the grass?" can be answered largely using statistical biases in the data itself (because frequently in this dataset, grass is green in color). Many models have attempted to solve the problem of VQA with varying degrees of success, but among them, the vast majority still attempt to solve the VQA task by exploiting biases in the

dataset [25, 37, 2, 17, 7, etc], while a smaller minority address the individual problem types [4, 38, 11, etc].

Keeping the former in mind, in this work, we provide a method to discover and enumerate explicitly, the various biases that are learned by a VQA model. To illustrate, in Figure 1(right), we provide examples of some rules learned by a strong baseline [25]. The model seems to have learned that if a question contains the words {What, time, day} (Eg : "What time of day is it?") and the accompanying image contains the bright sky ( ), the model is likely to answer "afternoon". The model answers "night" to the same question accompanied with an image containing a "night-sky" patch ( ). On the other hand, if it contains a clock face( ), it tends to answer the question with a time in an "HH:MM" format, while a question like "What time of the year?" paired with leafless trees( ) prompts "fall" as the answer. At the core of our method towards discovering such biases is the classical Apriori algorithm [3] which is used to discover rules in large databases - here the *database* refers to the question-words and model responses on the VQA validation set, which can be mined to produce these rules.

Deep learning algorithms reduce training error by learning biases in the data. This is evident from the observation that validation/test samples from the long tail of a data distribution are hard to solve, simply because similar examples do not occur frequently enough in the training set[41, 31, etc]. However, explicitly enumerating these biases in a human-interpretable form is possible only in a handful of problems, such as VQA. VQA is particularly illustrative because the questions and answers are in human language, while the images (and attention maps) can also be interpreted by humans. VQA is also interesting because it is a multi-modal problem - both language and vision are required to solve this problem. The language alone (i.e., an image agnostic model) can generate plausible (but often incorrect) answers to *most* questions (as we show in Section 4.1), but incorporating the image generates more accurate answers. That the language alone is able to produce plausible answers strongly indicates that VQA models implicitly

---

*This work was done while the author was at the University of Maryland.
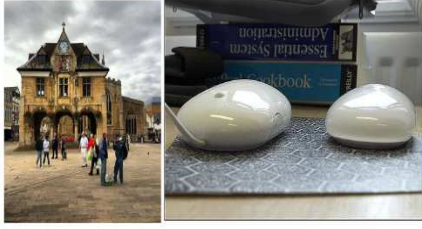
No.	antecedent words	antecedent visual words	consequents
1	what,time,day		afternoon*
2	what,time,day		night*
3	what,time,clock,show		11:30*
4	what,time,year		fall*

Figure 1. On the left, we show examples of two questions from the VQA dataset of [6, 18] where a model would require a "skill" to answer correctly (such as telling the time, or reading the English language), and a third which can be answered using statistical biases in the data itself. On the right, we show examples of statistical biases for a set of questions containing the phrase "What time?" and various visual elements (*antecedents*). Note that each row in this figure represents multiple questions in the VQA validation set. The * next to the answer (or *consequent*) reminds us that it is from the set of answer words. There are several visual words associated with afternoon and night, but we have provided only two for brevity.

.

use simple rules to produce answers - we endeavour in this paper to find an approach that can discover these rules.

Finally, we note that in this work, we do not seek to improve upon the state of the art. We do most of our experiments on the model of Kazemi et. al.[25], which is a strong baseline for this problem. We choose this model because it is simple to train and analyze (Section 3.1). To concretely summarize, our main contribution is to provide a method that can capture macroscopic rules that a VQA model ostensibly utilizes to answer questions. To the best of our knowledge, this is the first detailed work that analyzes the VQA dataset of [18] in this manner.

The rest of this paper is arranged as follows : In Section 2, we discuss related work, specifically those which look into identifying pathological biases in several machine learning problems, and "debugging" VQA models. In Section 3, we discuss details of our method. In Section 4, we provide experimental results and list (in a literal sense) some rules we believe the model is employing to answer questions. We discuss limitations of this method in Section 5 and conclude in Section 6.

2. Background and Related Work

The VQA problem is most often solved as a multi-class classification problem. In this formulation, an image(I) usually fed through a CNN, and a question(Q) fed through a language module like an LSTM [22] or GRU [13], are jointly mapped to an answer category ("yes", "no", "1", "2", etc). Although the cardinality of the set of all answers given a QI dataset is potentially infinite, researchers have observed that a set of a few thousand (typically 3000 or so) most frequently occurring answers can account for over 90% of all answers in the VQA dataset. Further, the evaluation of VQA in [6] and [18] is performed such that an answer receives partial credit if at least one human an-

notator agreed with the answer, even if it might not be the answer provided by the majority of the annotators. This further encourages the use of a classification based VQA system that limits the number of answers to the most frequent ones, rather than an answer generation based VQA system (say, using a decoder LSTM like [39]).

On undesirable biases in machine learning models: Machine learning methods are increasingly being used as tools to calculate credit scores, interest rates, insurance rates, etc, which deeply impact lives of ordinary humans. It is thus vitally important that machine learning models not discriminate on the basis of gender, race, nationality, etc[19, 5, 9]. [36] focus on revealing racial biases in image-based datasets by using adversarial examples. [43] explores data as well as models associated with object classification and visual semantic role labeling for identifying gender biases and their amplification. Further, [8] shows the presence of gender biases while encoding word embeddings, which is further exacerbated while using those embeddings to make predictions. [21] propose an Equalizer model which ensures equal gender probability when making predictions on image captioning tasks.

On debugging deep networks: The seminal work by [28] suggests that the Machine Learning community does not have a good understanding of what it means to interpret a model. In particular, this work expounds *post-hoc interpretability* - interpretation of a model's behavior based on some criteria, such as visualizations of gradients [34] or attention maps [42], *after* the model has been trained. Locally Interpretable Model Agnostic Explanations (LIME), [32] explain a classifier's behavior at a particular point by perturbing the sample and building a linear model using the perturbations and their predictions. A follow up work [33] constructs *Anchors*, which are features such that, in an instance where these features are present, a model's pre-

diction does not change. This work is the most similar prior work to ours, and the authors provide a few results on VQA as well. However, they only assume the existence of a model, and perturb instances of the data, whereas ours assumes the existence of responses to a dataset, but not the model itself. We use standard rule finding algorithms and provide much more detailed results on the VQA problem.

On debugging VQA :[1] study the behavior of models on the VQA 1.0 dataset. Through a series of experiments, they show that VQA models fail on novel instances, tend to answer after only partially reading the question and fail to change their answers across different images. In [2], recognizing that deep models seem to use a combination of identifying visual concepts and prediction of answers using biases learned from the data, the authors develop a mechanism to disentangle the two. However, they do not explicitly find a way to discover such biases in the first place. In [18], the authors introduce a second, more balanced version of the VQA dataset that mitigates biases (especially language based ones) in the original dataset. The resulting balanced dataset is christened VQA 2.0, and is the dataset that our results are reported on. In [24], the authors balance yes/no questions (those which indicate the presence or absence of objects), and propose two new evaluation metrics that compensate for forms of dataset bias.

3. Method

We cast our bias discovery task as an instance of the rule mining problem, which we shall describe below. The connection between discovering biases in VQA and rule mining is as follows : each (Question, Image, Answer) or QI+A triplet can be cast as a transaction in a database, where each word in the question, answer and image patch (or visual word, Section 3.2 and 3.3) is akin to an item. There are now three components to our rule mining operation :

- First, a frequent itemset miner picks out a set of all itemsets which occur at least s times in the dataset where s is the support. Because our dataset has over 200,000 questions (the entire VQA validation set), and the number of items exceeds 40,000 (all question words+all answer words+all visual words), we choose GMiner [14] due to its speed and efficient GPU implementation. Examples of such frequent itemsets in the context of VQA include {what, color, red*}, {what, sport, playing}, where the presence of a * indicates that the word is an answer-word.

- Next, a rule miner Apriori [3] forms all valid association rules $A \rightarrow C$, such that the rule has a support $> s$ and a confidence $> c$, where the confidence is defined as $\frac{|A \cup C|}{|A|}$. Here, the itemset A is called *antecedent* and the itemset C is called *consequent*. We choose $c = 0.2$

unless specified otherwise. An example of an association rule is {what, sport, playing, ▨▨} $\rightarrow$ {tennis*}, which can be interpreted as "If the question contains the words —what, sport, playing— and the accompanying image contains a tennis player, the answer *could* be tennis".

- Finally, a post-processing step removes obviously spurious rules by considering the causal nature of the VQA problem (i.e., only considering rules that obey : Image/Question $\rightarrow$ Answer). For the purpose of the results in Section 4, we query these rules with search terms like {What,sport}.

More concretely, let the ith (Image, Question) pair result in the network predicting the answer a^i. Let the question itself contain the words $\{w_1^i, w_2^i, ..., w_k^i\}$. Further, while answering the question, let the part of the image that the network shows attention towards correspond to the visual code-word v^i (Section 3.2 and 3.3). Then, this QI+A corresponds to the transaction $\{w_1^i, w_2^i, ..., w_k^k, v^i, a^i\}$. By pre-computing and combining question, answer and visual vocabularies, each item in a transaction can be indexed uniquely. This is shown in Figure 2 and explained in greater detail in the following sub-sections.

3.1. Baseline Model

The baseline model we use in this work is from [25], which was briefly a state-of-the-art method, yielding higher performance than other, more complicated models. We choose this model for two reasons : first, its simplicity (in other words, an absence of "bells and whistles") makes it a good test-bed for our method and has been used by other works that explore the behavior of VQA algorithms [30, 16]. The second reason is that the performance of this baseline is within 4% of the state-of-the-art model [37] without using external data or ensembles. We use the implementation of https://github.com/Cyanogenoid/pytorch-vqa. A brief description of this model is as follows : The VQA problem is formulated as a multi-class classification problem (Section 2). The input to the model is an image and a question, while the output is the answer class with the highest confidence (out of 3000 classes). Resnet-152[20] features are extracted from the image and concatenated with the last hidden state of an LSTM[22]. The text and visual features are combined to form attention maps which are fed to the softmax (output) layer through two dense layers. In this work, we focus on the second attention map.

3.2. Visual Codebook Generation

We generate the visual codebook using the classical "feature extraction followed by clustering" technique from [35]. First, we use the bounding-box annotations in MSCOCO[27] and COCO-Stuff[10] to extract 300,000
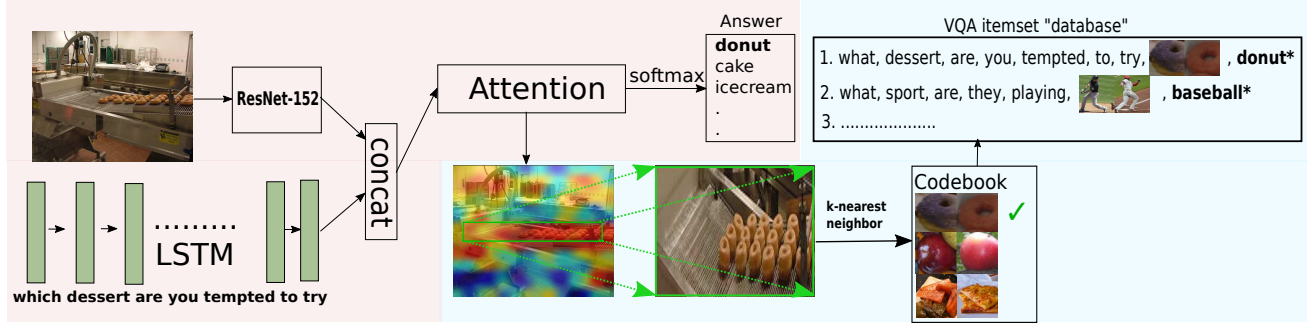
Figure 2. The model from [25] tries to answer the question "Which dessert are you tempted to try?". In doing so, the visual attention focuses on a region of the image which contains donuts. We use the method by [12] to place a bounding box over this region, which maps to a distinct visual word representing *donuts* in our vocabulary. Our database of items thus contains all of the words of the question, the visual word and the answer words. Rules are then extracted using the Apriori algorithm [3]

.

patches from the MSCOCO training set. After resizing each of the patches to 224×224 pixels, we extract ResNet-152[20] features for each of these patches, and cluster them into 1250 clusters using k-means clustering[15]. We note in Figure 3 that the clusters have both expected and unexpected characteristics beyond "objectness" and "stuffness". Expected clusters include dominant objects in the MSCOCO dataset like zebras, giraffes, elephants, cars, buses, trains, people, etc. However, other clusters have textural content, unusual combinations of objects as well as actions. For example, we notice visual words like "people eating", "cats standing on toilets", "people in front of chain link fences", etc, as shown in Figure 3. The presence of these more *eclectic* code-words casts more insight into the model's learning dynamics - we would prefer frequent itemsets containing the visual code-word corresponding to "people eating" than just "people" for a QA pair of *(what is she doing?, eating)*.

3.3. From attention map to bounding box

In this work, we make an assumption that the network focuses on exactly one part of the image, although our method can be easily extended to multiple parts[12]. Following the elucidation of our method in Section 3 and given an attention map, we would like to compute the nearest visual codeword. Doing so requires making the choice of a bounding box that covers enough of the salient parts of the image, cropping and mapping this patch to the visual vocabulary. While there are trainable (deep network based) methods for cropping attention maps [40], we instead follow the simpler formulation suggested by [12], which states that : within an attention-map G, given a percentage ratio τ, find the smallest bounding box B which satisfies :

$$\sum_{p \epsilon B} G(p) \geq \tau \sum_p G(p), \tau \epsilon [0, 1]$$

Since we follow [25] who use a ResNet-152 architecture for visual feature extraction, the attention maps are of size 14×14. It can be shown easily that given a $m \times n$ grid, the number of unique bounding boxes that can be drawn on this grid, i.e., $num\_bboxes = \frac{m \times n \times (m+1) \times (n+1)}{4}$, and when $m = n = 14$, num_bboxes turns out to be 11,025. Because $m(= n)$ is small and fixed in this case, we pre-compute and enumerate all 11,025 bounding boxes and pick the smallest one which encompasses the desired attention, with $\tau = 0.3$. The reason behind a conservatively low choice for τ is that we do not want to crop large regions of the image, which might contain distractor patches. This part of the pipeline is depicted in Figure 4.

3.4. Pipeline Summarized

Now, the pipeline for the experiments (Figure 2) on the VQA dataset including images is as follows. We provide as input to the network - an image and a question. We observe the second attention map and use the method of Section 3.3 to place a tight-fitting bounding-box around those parts of the image that the model attends to. We then extract features on this bounding-box using a ResNet-152 network and perform a k-nearest neighbor search (with $k = 1$) to obtain its nearest visual word from the vocabulary. The words in the question, visual code-word and predicted answer for the entire validation set are provided as the database of transactions to the frequent itemset miner [14], and rules are then obtained using the Apriori algorithm [3].

4. Experiments

4.1. Language only statistical biases in VQA

We show that a large number of statistical biases in VQA are due to language alone. We illustrate this with an obvious example : a language-only model, i.e., one that does not see the image, but still attempts the question, answers about

Figure 3. We show visual code-words generated by the method of Section 3.1. In the first (left-most) column, we notice visual code-words corresponding to objects or patches in MSCOCO, but in the latter two columns (on the right) we notice code-words corresponding to more complex visual concepts like "people eating", "women in bridal-wear" or "black-and-white tennis photographs".
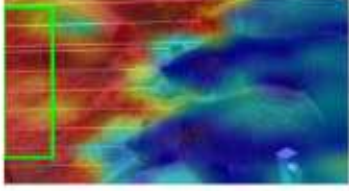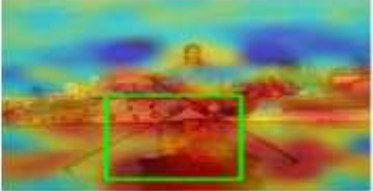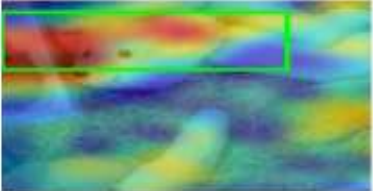

Figure 4. In the first example, critical to answering the question correctly is discovering the presence of a fence (shown in red) in the attention heat-map. The cropping method of [12] places a conservative box over this region, which corresponds to net-like or fence-like visual code-words like a tennis-net or a baseball batting-cage in the visual codebook. Similarly, in the second example, the attention corresponds to a visual code-word which clearly depicts boats, and in the third example, the attention corresponds to the teddy-bear code-word.

43% of the questions correctly on VQA 2.0 validation set and 48% of the questions correctly on VQA 1.0 validation set[18]. However, on a random set of 200 questions from VQA 2.0, we observed empirically that the language-only model answers 88.0% of questions with a *plausibly correct* answer even with a harsh metric of what *plausible* means. Some of these responses are fairly sophisticated as can be seen in Table 1. We note, for example, that questions containing "kind of bird" are met with a species of bird as response, "What kind of cheese" is answered with a type of

Question	Predicted	G.T Ans.
What kind of bird is perched on this branch ?	owl	sparrow
What does that girl have on her face ?	sunglasses	nothing
What kind of cheese is on pizza ?	mozzarella	mozzarella
What is bench made of ?	wood	wood
What brand of stove is in kitchen ?	electric	LG

Table 1. We run a language-only VQA baseline and note that although only 43% of the questions are answered correctly in VQA 2.0 ([18]), a large number of questions (88%) in our experiments are answered with plausibly correct responses. For example, "Sunglasses" would be a perfectly plausible answer to the question "What does that girl have on her face?" - perhaps even more so than the ground-truth answer ("Nothing"). The last example shows an implausible answer provided by the model to the question.

cheese, etc. Thus, the model maps out key words or phrases in the question and *ostensibly* tries to map them through a series of rules to answer words. This strongly indicates that these are biases learned from the data, and the ostensible rules can be mined through a rule-mining algorithm.

4.2. Vision+Language statistical biases in VQA

After applying the method of Section 3, we will examine some rules that have been learned by our method on some popular question types in VQA . Question types are taken from [6] and for the purpose of brevity, only a few instructive rules for each question type are displayed. These question types are : "What is he/she doing?" (Section 4.2.3), "Where?" (Figure 9), "How many?" (Section 4.2.1), "What brand?" (Figure 8), and "Why?"(Section 4.2.2). The tables we present are to be interpreted thus : A question containing the antecedent words paired with an image containing the antecedent visual words can sometimes (but not always) lead to the consequent answer. Two instances of patches mapping to this visual word (Section 3.2) are provided. The presence of an * after the consequent is to remind the reader that the consequent word came from the set of answers.

4.2.1 How many?

This particular instance of the trained VQA model seems to have learned that giraffes have four legs, stop signs have four letters, kitchen stoves have four burners and zebras and giraffes have several (100) stripes and spots respectively (Figure 5). Upon closer examination, we found 33 questions (out of >200k) in the VQA validation set which contain the words {How,many,burners} and the most common answer predicted by our model for these is 4 (which also resembles the ground-truth distribution). However, some of them were along the lines of "How many burners are turned on?", which led to answers different from "4".

4.2.2 Why?

Traditionally, "Why?" questions in VQA are considered challenging because they require a reason based answer. We describe some of the rules purportedly learned by our model for answering "Why?" questions, in Figure 6. Some interesting but intuitive beliefs that the model has learned are that movements cause blurry photographs (why,blurry→movement), outstretching one's arms help in balancing (why,arm→balance) and that people wear helmets or orange vests for the purpose of safety (why,helmet/orange→safety). In many of these cases, no visual element has been picked up by the rule mining algorithm - this strongly indicates that the models are memorizing the answers to the "Why?" questions, and not performing any reasoning. In other words, we could ask the question "Why is the photograph blurry?" to an irrelevant image and obtain "Movement" as the predicted answer.

4.2.3 What is he/she doing?

More interesting are our results on the "What is he/she doing?" category of questions (Figure 7). While common activities like "snowboarding" or "surfing" are prevalant among the answers, we noticed a difference in rules learned for male and female pronouns. For the female pronoun (she/woman/girl/lady), we observed only stereotypical outputs like "texting" even for a very low support, as compared to a more diverse set of responses with the male pronoun. This is likely, a reflection on the inherent bias of the MSCOCO dataset which the VQA dataset of [6, 18] is based on. Curiously, another work by [21] had similar observations for image captioning models also based on MSCOCO.

5. Limitations

While simplicity is the primary advantage of our method, some drawbacks are the following : the exact nature of the rules is limited by the process used to generate the visual vocabulary. In other words, while our method provides a unique insight into the behavior of a VQA model, there surely exist some rules that the models seem to follow which cannot be captured by this method. For example, rules involving colors are difficult to identify because ResNets are trained to be somewhat invariant to colors, so purely color-based visual words are hard to compute. Other examples include inaccurate visual code-words - for example, in rule 4 of Figure 8, the antecedant visual word does show a motorbike, although not a Harley Davidson. Similarly a code-word contains images of scissors and toothbrushes grouped together as part of the (What,brand→Colgate) associate rule (rule 5 of Figure 8).
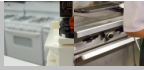
No.	antecedant words	antecedant visual words	consequents	support x 10^{-5}	confidence
1	many,stripe,how		100*	12.1295	0.76
2	many,spot,how		100*	9.79688	0.91
3	many,burner,how		4*	3.73214	0.38
4	many,leg,how		4*	2.33259	0.33
5	how,letter,many		4*	2.33259	0.71

Figure 5. **How many?** : Rule 3-5 show that stoves have 4 burners, giraffes have 4 legs and stop signs have 4 letters. Giraffes and zebras have many (100) spots and stripes, respectively (rules 1-2).

No.	antecedant words	antecedant visual words	consequents	support x 10^{-5}	confidence
1	why		raining*	6.06473	0.31
2	umbrella,why		shade*	6.06473	0.62
3	why,blurry	-	movement*	6.06473	0.46
4	helmet,why	-	safety*	4.66518	0.77
5	why,fence	-	safety*	4.19866	0.47
6	why,wet	-	surfing*	3.73214	0.33
7	arm,why	-	balance*	3.26563	0.47
8	orange,why	-	safety*	2.33259	0.5

Figure 6. **Why?** : Rules that exceeded the support threshold indicate that arms are outstretched for balance (rule 7), umbrellas protect one from rain and provide shade (rules 1-2), and that helmets, fences and (wearing) orange lead to safety (rules 4, 5, 8). The absence of visual words in some of these rules indicates that the model is predicting the answer based on question-words only.

No.	antecedant words	antecedant visual words	consequents	support x 10^{-5}	confidence
1	doing,what,man		surfing*	17.7277	0.64
2	doing,what,man		skateboarding*	13.529	0.81
3	doing,what,man		snowboarding*	6.53125	0.5
4	doing,what,man		playing wii*	2.79911	0.46
5	doing,what,woman		texting*	1.86607	0.4

Figure 7. **What is he/she doing?** : We observed a difference in diversity of rules for male (skateboarding, snowboarding, surfing) and female pronouns (texting) even at very low support. This indicates that the VQA , or more likely, the MSCOCO datasets are unintentionally skewed in terms of gender.
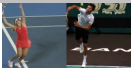
No.	antecedant words	antecedant visual words	consequents	support x 10^{-5}	confidence
1	brand,what		dell*	9.33036	0.41
2	brand,what		wilson*	5.59822	0.57
3	brand,computer,what		apple*	4.66518	0.45
4	brand,what		harley davidson*	4.19866	0.38
5	what,brand		colgate*	3.26563	0.58
6	brand,what		jetblue*	2.33259	0.38

Figure 8. **What brand?** : The VQA model seems to have learned that the Wilson brand is related to tennis, Dell and Apple make laptop computers and that Jetblue is a "brand" of airline.
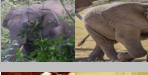
No.	antecedant words	antecedant visual words	consequents	support x 10^{-5}	confidence
1	where		airport*	21.9263	0.61
2	where		zoo*	13.529	0.54
3	where		africa*	9.79688	0.38
4	where		bathroom*	5.59822	0.23
5	where		skate park*	5.1317	0.24
6	bus,where		downtown*	5.1317	0.24

Figure 9. **Where?** : The model of [25] has learned that giraffes can be found in zoos, elephants are from Africa, aircraft can be found in airports and that buses are found in the downtown of a city

6. Conclusion

In this work, we present a simple technique to explicitly discover biases and correlations learned by VQA models. To do so, we store in a database - the words in the question, the response of the model to the question and the portion of the image attended to by the model. Our method then leverages the Apriori algorithm[3] to discover rules from this database. We glean from our experiments that VQA models intuitively seem to correlate *elements* (both textual and visual) in the question and image to answers.

Our work is consistent with prior art in machine learning on fairness and accountability[21], which often shows a skew towards one set of implied factors (like gender), com- pared to others. It is also possible to use the ideas in this work to demonstrate effectiveness of VQA systems - showing dataset biases presented by a frequent itemset and rule miner is a middle-ground between quantitative and qualitative results. Finally, our method is not limited only to VQA , but any problem with a discrete vocabulary. A possible future extension of this work is to track the development of these rules as a function of training time.

References

[1] A. Agrawal, D. Batra, and D. Parikh. Analyzing the behavior of visual question answering models. In *EMNLP*, 2016.

[2] A. Agrawal, D. Batra, D. Parikh, and A. Kembhavi. Don't just assume; look and answer: Overcoming priors for visual question answering. In *CVPR*, 2018.

[3] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *VLDB*, 1994.

[4] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein. Neural module networks. In *CVPR*, 2016.

[5] J. Angwin, J. Larson, S. Mattu, and L. Kirchner. Machine bias: Theres software used across the country to predict future criminals. and its biased against blacks. In *ProPublica*, 2016.

[6] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh. VQA: Visual question answering. In *ICCV*, 2015.

[7] H. Ben-younes, R. Cadène, M. Cord, and N. Thome. MU-TAN: multimodal tucker fusion for visual question answering. In *ICCV*, 2017.

[8] T. Bolukbasi, K. Chang, J. Y. Zou, V. Saligrama, and A. T. Kalai. Man is to computer programmer as woman is to home-maker? debiasing word embeddings. In *NIPS*, 2016.

[9] J. Buolamwini and T. Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on Fairness, Accountability and Transparency, FAT 2018, 23-24 February 2018, New York, NY, USA*, 2018.

[10] H. Caesar, J. Uijlings, and V. Ferrari. Coco-stuff: Thing and stuff classes in context. In *CVPR*, 2018.

[11] P. Chattopadhyay, R. Vedantam, R. R. Selvaraju, D. Batra, and D. Parikh. Counting everyday objects in everyday scenes. In *CVPR*, 2017.

[12] J. Chen, G. Bai, S. Liang, and Z. Li. Automatic image cropping: A computational complexity study. In *CVPR*, 2016.

[13] K. Cho, B. van Merriënboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *EMNLP*, 2014.

[14] K. Chon, S. Hwang, and M. Kim. Gminer: A fast gpu-based frequent itemset mining method for large-scale data. In *Inf. Sci.*, volume 439-440, pages 19–38, 2018.

[15] Y. Ding, Y. Zhao, X. Shen, M. Musuvathi, and T. Mytkowicz. Yinyang k-means: A drop-in replacement of the classic k-means with consistent speedup. In *ICML*, 2015.

[16] S. Feng, E. Wallace, A. G. II, M. Iyyer, P. Rodriguez, and J. Boyd-Graber. Pathologies of neural models make interpretation difficult. In *Empirical Methods in Natural Language Processing*, 2018.

[17] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *EMNLP*, 2016.

[18] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In *CVPR*, 2017.

[19] M. Hardt, E. Price, and N. Srebro. Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems 29*. 2016.

[20] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[21] L. A. Hendricks, K. Burns, K. Saenko, T. Darrell, and A. Rohrbach. Women also snowboard: Overcoming bias in captioning models. In *ECCV*, 2018.

[22] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–1780, 1997.

[23] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, 2017.

[24] K. Kafle and C. Kanan. An analysis of visual question answering algorithms. In *ICCV*, 2017.

[25] V. Kazemi and A. Elqursh. Show, ask, attend, and answer: A strong baseline for visual question answering. 2017.

[26] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. Bernstein, and L. Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. 2016.

[27] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. 2014.

[28] Z. C. Lipton. The mythos of model interpretability. In *Queue*, 2018.

[29] M. Malinowski and M. Fritz. A multi-world approach to question answering about real-world scenes based on uncertain input. In *NIPS*, 2014.

[30] P. K. Mudrakarta, A. Taly, M. Sundararajan, and K. Dhamdhere. Did the model understand the question? In *ACL*, 2018.

[31] W. Ouyang, X. Wang, C. Zhang, and X. Yang. Factors in finetuning deep model for object detection with long-tail distribution. In *CVPR*, 2016.

[32] M. T. Ribeiro, S. Singh, and C. Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *Knowledge Discovery and Data Mining (KDD)*, 2016.

[33] M. T. Ribeiro, S. Singh, and C. Guestrin. Anchors: High-precision model-agnostic explanations. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[34] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 2017.

[35] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *ICCV*, 2003.

[36] P. Stock and M. Cissé. Convnets and imagenet beyond accuracy: Understanding mistakes and uncovering biases. In *ECCV*, 2018.

[37] D. Teney, P. Anderson, X. He, and A. van den Hengel. Tips and tricks for visual question answering. In *CVPR*, 2018.

[38] A. Trott, C. Xiong, and R. Socher. Interpretable counting for visual question answering. In *ICLR*, 2018.

[39] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. 2015.

[40] W. Wang and J. Shen. Deep cropping via attention box prediction and aesthetics assessment. In *ICCV*, 2017.

[41] Y. Wang, D. K. Ramanan, and M. Hebert. Learning to model the tail. In *31st Conference on Neural Information Processing Systems (NIPS)*, December 2017.

[42] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. 2015.

[43] J. Zhao, T. Wang, M. Yatskar, V. Ordonez, and K.-W. Chang. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. In *EMNLP*, 2017.