Supplementary Materials for DVQA: Understanding Data Visualizations via Question Answering

Kushal Kafle[1,*] Brian Price[2] Scott Cohen[2] Christopher Kanan[1]
[1]Rochester Institute of Technology [2]Adobe Research
[1]{kk6055, kanan}@rit.edu [2]{bprice, scohen}@adobe.com

1. Additional details about the dataset

In this section, we present additional details on the DVQA dataset statistics and how it was generated.

1.1. Data statistics

Table 1 extends Table 1 of the main paper on the distribution of questions in the DVQA dataset.

1.2. Variations in question templates

The meaning of different entities in a chart is determined by its title and labels. This allows us to introduce variations in the questions by changing the title of the chart. For example, for a generic title 'Title' and a generic label 'Values', the base-question is: 'What is the value of **L**?'. Depending on the title of the chart, the same question can take following forms:

1. Title: Accuracy of different algorithms, Label: Accuracy ⇒ What is the accuracy of the algorithm **A**?

2. Title: Most preferred objects, Label: Percentage of people ⇒ What percentage of people prefer object **O**?

3. Title: Sales statistics of different items, Label: Units sold ⇒ How many units of the item **I** were sold?

Figure 1 provides an example on how questions can be varied for the same chart by using a different title and different labels.

1.3. Data and visualization generation

In this section, we provide additional details on the heuristics and methods used for generating question-answer pairs.

We aim to design the DVQA dataset such that commonly found visual and data patterns are also more commonly encountered in the DVQA dataset. To achieve this, we downloaded a small sample of bar-charts from Google image search and *loosely* based the distribution of our DVQA dataset on the distribution of downloaded charts. However, some types of chart elements such as logarithmic axes, negative values, etc. that do not occur frequently in the wild are still very important to be studied. To incorporate these in our dataset, we applied such chart elements to a small proportion of the overall dataset. However, we made sure that each of the possible variations was encountered at least 1000 times in the training set.

1.3.1 Distribution of visual styles

To incorporate charts with several appearances and styles in our DVQA dataset, we introduced different types of variations in the charts. Some of them as listed below:

1. Variability in the number of bars and/or groups of bars.

2. Single-column vs. multi-column grouped charts.

3. Grouped bars vs. stacked bars. Stacked bars are further divided into two types: 1) Additive stacking, where bars represent individual values, and 2) Fractional stacking, where each bar represents a fraction of the whole.

4. Presence or absence of grid-lines.

5. Hatching and other types of textures.

6. Text label orientation.

7. A variety of colors, including monochrome styles.

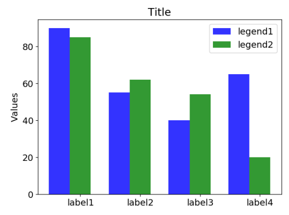
8. Legends placed in a variety of common positions, including legends that are separate from the chart.

9. Bar width and spacing.

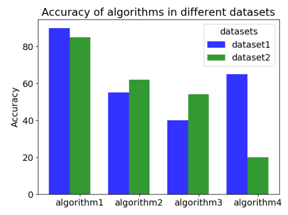
10. Varying titles, labels, and legend entries.

11. Vertical vs. horizontal bar orientation.

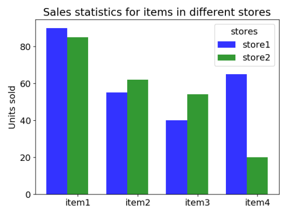
[*]A portion of this research was done while Kushal Kafle was an intern at Adobe Research.



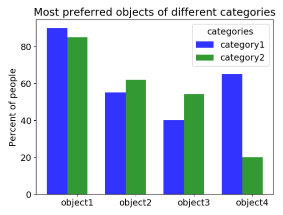
What is the value of *label1* in *legend2*?



What is the accuracy of the algorithm *algorithm1* in the dataset *dataset2*?



How many units of the item *item1* was sold in the store *store2*?



What percentage of people prefer the object *object1* in the category *category2*?

Figure 1: An example showing that different question can be created by using different title and labels in the same chart.

Table 1: Statistics on different splits of dataset based on different question types.

		Total Questions	**Unique Answers**	**Top-2 Answers (in percentage)**
Structure	Train	313,842	10	no: 40.71, yes: 40.71
	Test-Familiar	78,278	10	no: 41.14, yes: 41.14
	Test-Novel	78,988	10	no: 41.00, yes: 41.00
Data	Train	742,896	1038	no: 7.55, yes: 7.55
	Test-Familiar	185,356	1038	no: 7.44, yes: 7.44
	Test-Novel	185,452	538	no: 7.51, yes: 7.51
Reasoning	Train	1,076,391	1076	yes: 8.29, no: 8.26
	Test-Familiar	268,795	1075	no: 8.31, yes: 8.27
	Test-Novel	268,788	577	no: 8.28, yes: 8.22
Overall	Train	2,325,316	1076	yes: 11.74, no: 11.73
	Test-Familiar	580,557	1075	yes: 11.77, no: 11.75
	Test-Novel	581,321	577	no: 11.80, yes: 11.77

In the wild, some styles are more common than others. To reflect this in our DVQA dataset, less common styles, *e.g*. hatched bars, are applied to only a small subset of charts. However, every style-choice appears at least a 1000 times in the training set. In overall, 70% of the charts have vertical bars and the remaining charts have horizontal bars. Among multi-column bar-charts, 20% of the linear and normalized percentage bar-charts are presented as stacked bar-charts and the rest are presented as group bar-charts. In legends we have used two styles that are commonly found in the wild: 1) legend below the chart, and 2) legend to the right of the chart. In 40% of the multi-column charts, legends are positioned outside the bounds of the main chart. Finally, 20% of the charts are hatch-filled with a randomly selected pattern out of six commonly used patterns (stripes, dots, circles, cross-hatch, stars, and grid).

1.3.2 Distribution of data-types

Our DVQA dataset contains three major types of data scales.

- **Linear data.** Bar values are chosen from 1 – 10, in an increment of 1. When bars are not stacked, the axis is clipped at 10. When bars are stacked, the maximum value of the axis is automatically set by the height of the tallest stack. For a small number of charts, values are randomly negated or allowed to have missing values (*i.e*. value of zero which appears as a missing bar).
- **Percentage data.** Bar values are randomly chosen from 10–100, in increments of 10. For a fraction of multi-column group bar charts with percentage data, we normalize the data in each group so that the values add up to 100, which is a common style. A small fraction of bars can also have missing or zero value.
- **Exponential data.** Bar values are randomly chosen in the range of 1 - 10^{10}. The axis is logarithmic.

The majority (70%) of the data in the DVQA dataset is of the linear type (1–10). Among these, 10% of the charts are allowed to have negative. Then, 25% of the data contain percentage scales (10–100), among which half are normalized so that the percentages within each group add up to a 100%. For 10% of both linear and percentage data-type,

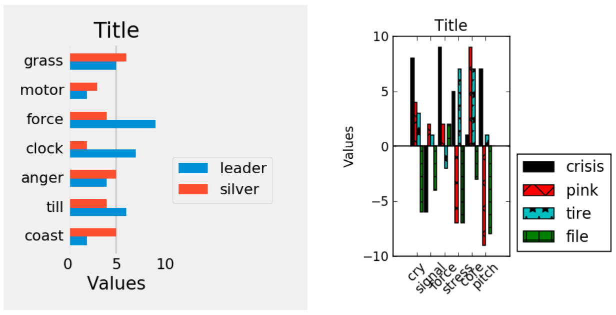


Figure 2: Examples of discarded visualizations due to the bar-chart being smaller than 50% of the total image area.

bars are allowed to have missing (zero) values. The remaining 5% of the data is exponential in nature ranging from 10^0 – 10^{10}.

1.3.3 Ensuring proper size and fit

Final chart images are drawn such that all of them have the same width and height of 448×448 pixels. This was done for the ease in processing and to ensure that the images do not need to undergo stretching or aspect ratio change when being processed using an existing CNN architecture. To attain this, we need to ensure that all the elements in the chart fit in the fixed image size. We have taken several steps to ensure a proper fit. By default, the label texts are drawn without rotation *i.e*. horizontally. During this, if any of the texts overlap with each other, we rotate the text by either 45 or 90 degrees. Another issue is when the labels take up too much space leaving too little space for the actual bar-charts, which often makes them illegible. This is usually a problem with styles that contain large texts and/or charts where legend is presented on the side. To mitigate this, we discard the image if the chart-area is less than half of the entire image-area. Similarly, we also discard a chart if we cannot readjust the labels to fit without overlap despite rotating them. Fig. 2 shows some examples of discarded charts due to poor fit.

1.3.4 Naming colors

For generating diverse colors, we make use of many of the pre-defined styles that are available with the Matplotlib package and also modify it with several new color schemes. Matplotlib allows us to access the RBG face-color of each drawn bar and legend entries from which we can obtain the color of each of the element drawn in the image. However, to ask questions referring to the color of a bar or a legend entry, we need to be able to name it using natural language (*e.g*. 'What does the `red color` represent?). Moreover, simple names such as 'blue' or 'green' alone may not suffice to distinguish different colors in the chart. So, we employ the following heuristic to obtain a color name for a given RGB value.

1. Start with a dictionary of all 138 colors from the CSS3 X11 named colors. Each of the color is accompanied by its RGB value and its common name. The color names contain names such as darkgreen, skyblue, navy, lavender, chocolate, and other commonly used colors in addition to canonical color names such as 'blue', 'green', or 'red'.

2. Convert all the colors to CIE standard L*a*b* color space which is designed to approximate human perception of the color space.

3. Measure color distance between the L*a*b* color of our chart-element and each of the color in the X11 color dictionary. For distance, we use the CIE 2000 delta E color difference measure which is designed to measure human perceptual differences between colors.

4. Choose the color from the X11 colors which has the lowest delta E value from the color of our chart-element.

Table 2: Localization performance of MOM in terms of IOU with the ground truth bounding box.

IOU with ground truth	**Percentage of boxes**
≥ 0.2	73.27
≥ 0.4	56.89
≥ 0.5	46.06
≥ 0.6	32.49
≥ 0.7	18.80
≥ 0.8	6.93
≥ 0.9	0.66
≥ 1.0	0.00

Table 3: Localization performance of MOM in terms of the distance between the center of the predicted and ground truth bounding box.

Distance from the ground truth	**Percentage of boxes**
≤ 1 pixels	0.14
≤ 8 pixels	8.48
≤ 16 pixels	25.77
≤ 32 pixels	52.89
≤ 64 pixels	74.21

2. Analysis of MOM's localization performance

In the main paper, we observed that many predictions made by MOM were close to the ground truth but not exactly the same. This was also corroborated by taking into

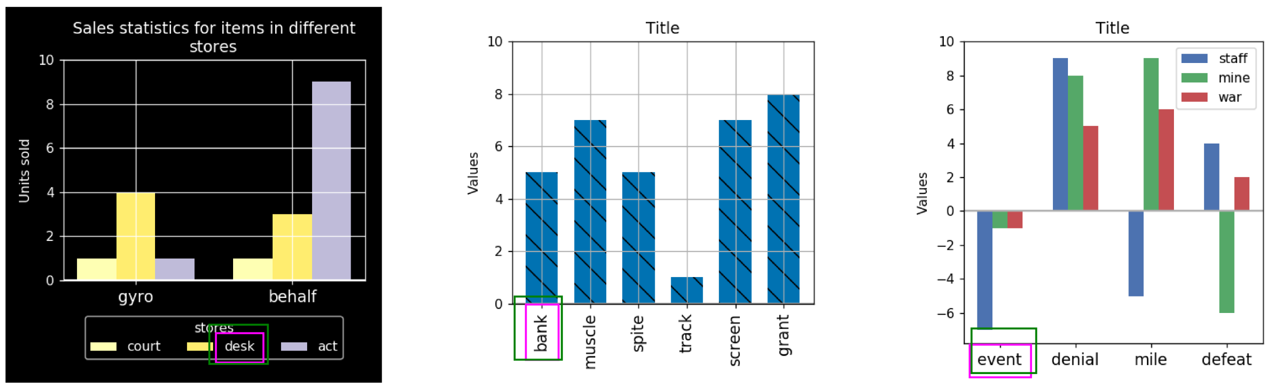


Figure 3: Some examples showing correctly predicted bounding boxes predicted by our MOM model. Magenta shows the ground truth and green shows the predicted bounding box.

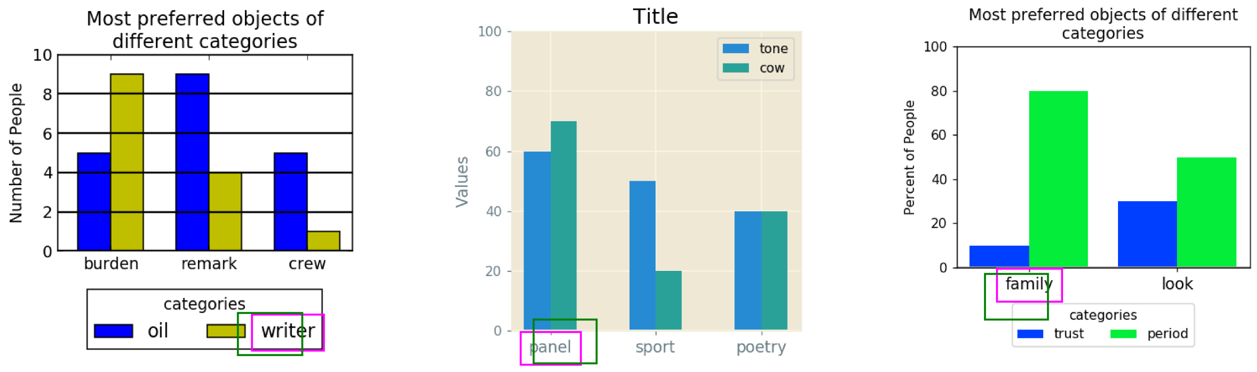


Figure 4: Some examples showing incorrectly predicted bounding boxes predicted by our MOM model. Often the prediction is off by only a few pixels, but the since the OCR requires total coverage, it results in an erroneous prediction. Magenta shows the ground truth and green shows the predicted bounding box.

account the edit-distance between the predicted and ground truth answer strings.

Here we study our hypothesis that this low accuracy is due to poor localization of the predicted bounding boxes. Fig. 3 shows some results from MOM for Test-Familiar split of the dataset in which the bounding boxes are accurately predicted. This shows that the bounding box prediction network works with texts of different orientations and positions. However, Fig. 4 shows some examples where boxes do not 'snap' neatly around the text area but are in the right vicinity. Since the OCR subnetwork in MOM operates only on the features extracted from the predicted bounding box, a poor bounding box would also translate to a poor prediction. To quantify this behavior we conduct two separate studies.

First, we measure the intersection over union (IOU) for predicted and ground truth bounding boxes. Table 2 shows the percentages of boxes that were accurately predicted for various threshold values of IOU.

Next, we measure what percentage of the predicted boxes are within a given distance from the ground truth boxes. The distance is measured as the Euclidean distance between the center x,y co-ordinates for predicted and ground-truth bounding boxes. Result presented in Table 3 shows that more than half of the predicted boxes are within 32 pixels from the ground truth boxes. Note here that the image dimension is 448$\times$448 pixels.

The above experiments show that while many of the predicted bounding boxes are 'near' the ground truth boxes, they do not perfectly enclose the text. Therefore, if the predicted bounding boxes are localized better, which could be achieved with additional fine-tuning of the predicted bounding boxes, we can expect a considerable increase in MOM's accuracy on chart-specific answers.

3. Additional examples

In this section, we present additional examples to illustrate the performance of different algorithms for different types of questions. Fig. 5 shows some example figures with question-answer results for different algorithms and Fig. 6 shows some interesting failure cases.

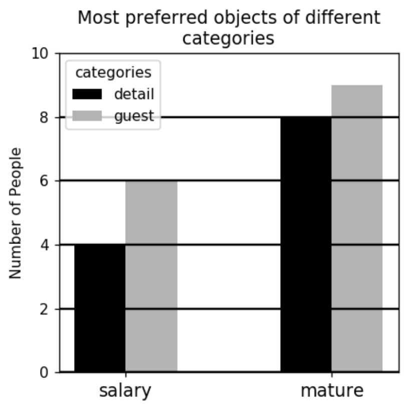
As shown in Fig. 5, SAN-VQA, MOM, and SANDY all perform with high accuracy across different styles for structure understanding questions. This is unsurprising since all the models use the SAN architecture for answering these questions. However, despite the presence of answer-words in the training set (test-familiar split) SAN is incapable of answering questions with chart-specific answers; it always produces the same answer regardless of the question being asked. In comparison, MOM shows some success in decoding the chart-specific answers. However, as explained earlier in section 2, the accuracy of MOM for chart-specific answers also depends on the accuracy of the bounding box prediction due to which its predictions were close but not exact for many questions. As discussed in section 2, although the exact localization of the bounding box was poor, the majority of the predicted bounding boxes were in the vicinity of the ground truth bounding boxes. We believe with additional fine-tuning, *e.g*. regressing for a more exact bounding box based on the features surrounding the initial prediction, could improve the model's performance significantly. Finally, SANDY shows a remarkable success in predicting the chart-specific answers. SANDY's dynamic dictionary converts the task of predicting the answer to predicting the position of the text in the image, making it easier to answer. Once the position is predicted, there are no additional sources of error for SANDY making it less error prone in general.

Similarly, both SAN and MOM are incapable of correctly parsing the questions with chart-specific labels in them. In comparision, SANDY can use the dynamic local dictionary to correctly parse the chart-specific labels showing an improved performance for these questions *e.g*. Fig. 5c, 6c, and 6e.

In Fig. 6, we study some failure cases to better understand the nature of the errors made by current algorithms. One of the most commonly encountered errors for the algorithms that we tested is the error in predicting exact value of the data. Often, predicting these values involve extracting exact measurement and performing arithmetic operations across different values. The results show that the models are able to perform some measurement; the models predict values that are close to the correct answer, *e.g*. predicting smaller values when the bars have smaller height (Fig. 6d) and predicting larger values when the bars are tall (Fig. 6f). In addition, the models are able to make predictions in the accurate data scale *e.g*. For Fig. 6d, the prediction for the value is in percentage scale (0–100) and for Fig. 6e, the prediction is in linear scale (0–10).

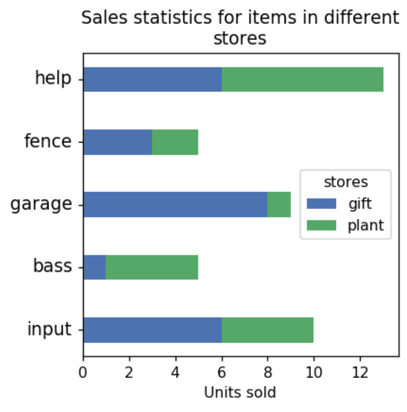
The next class of the commonly encountered errors is the prediction of chart-specific answers. We have already established that the SAN-VQA model completely fails to answer questions with chart-specific answers, which is demonstrated in all the examples in Fig. 5 and 6. Our MOM model also makes errors for several examples as shown in Fig. 6. The errors occur in decoding the OCR (Fig. 6a), predicting the right box (Fig. 6f) or both (Fig. 6d). While our SANDY model shows vastly increased accuracy for these answers, it can make occasional errors for these questions (Fig. 6d).

Example question-answer pairs for different models



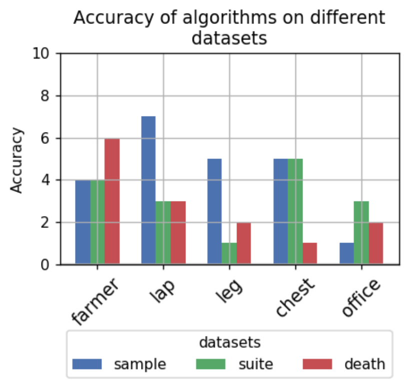
(a)

Q: What is the label of the second bar from the left in each group?
SAN: closet ✗ MOM: guest ✓ SANDY: guest ✓
Q: Is each bar a single solid color without patterns?
SAN: yes ✓ MOM: yes ✓ SANDY: yes ✓



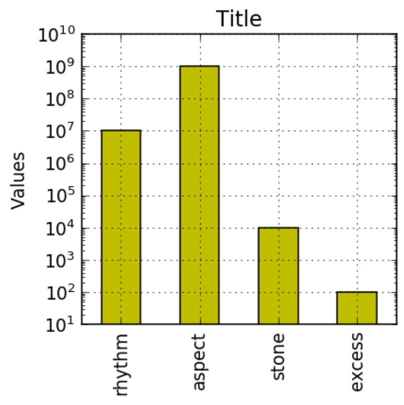
(b)

Q: How many items sold less than 6 units in at least one store?
SAN: four ✓ MOM: four ✓ SANDY: four ✓
Q: Does the chart contain stacked bars?
SAN: yes ✓ MOM: yes ✓ SANDY: yes ✓



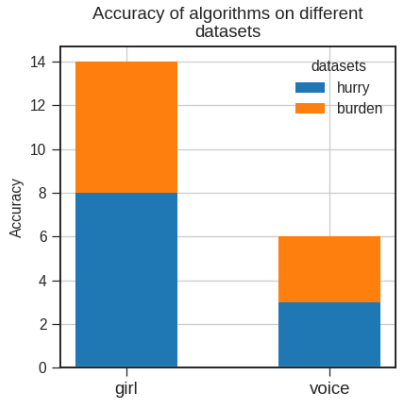
(c)

Q: What is the highest accuracy reported in the whole chart?
SAN: 7 ✓ MOM: 7 ✓ SANDY: 7 ✓
Q: Is the accuracy of the algorithm leg in the dataset suite smaller than the accuracy of the algorithm chest in the dataset sample?
SAN: no ✗ MOM: no ✗ SANDY: yes ✓



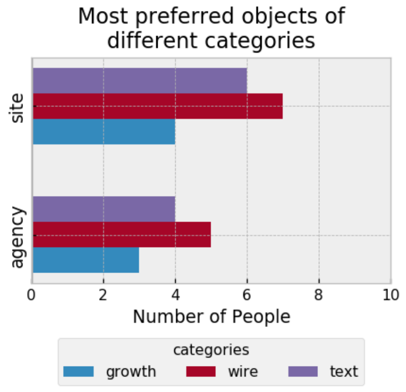
(d)

Q: Which bar has the largest value?
SAN: closet ✗ MOM: aspect ✓ SANDY: aspect ✓
Q: What is the value of the largest bar?
SAN: 10^9 ✓ MOM: 10^9 ✓ SANDY: 10^9 ✓



(e)

Q: How many algorithms have accuracy lower than 3 in at least one dataset?
SAN: zero ✓ MOM: zero ✓ SANDY: zero ✓
Q: Which algorithm has highest accuracy for any dataset?
SAN: closet ✗ MOM: girl ✓ SANDY: girl ✓

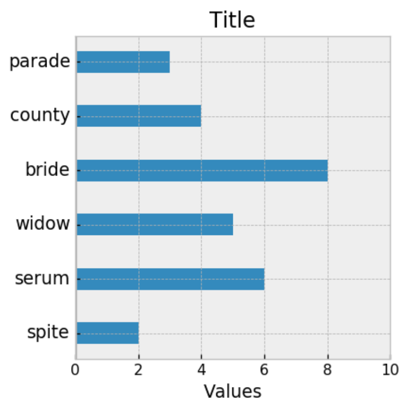


(f)

Q: Which object is preferred by the most number of people summed across all the categories?
SAN: closet ✗ MOM: site ✓ SANDY: site ✓
Q: Are the bars horizontal?
SAN: yes ✓ MOM: yes ✓ SANDY: yes ✓

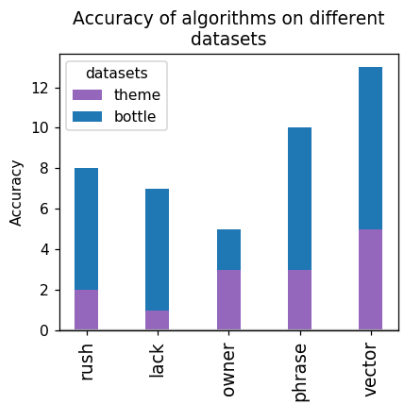
Figure 5: Some example question-answer pair for different algorithms on the Test-Familiar split of the dataset. The algorithms show success in variety of questions and visualizations. However, the SAN model is utterly incapable of predicting chart-specific answers.

Some interesting failure cases



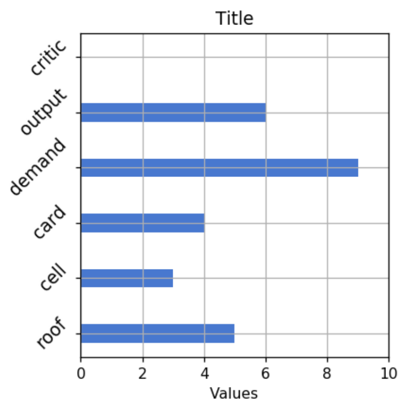
(a)

Q: What is the label of the third bar from the bottom?
SAN: closet ✗ MOM: whidkw ✗ SANDY: widow ✓



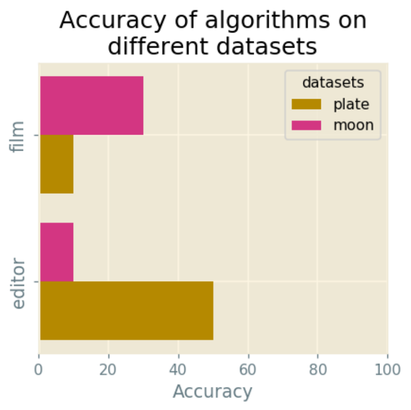
(b)

Q: Which algorithm has the largest accuracy summed across all the datasets?
SAN: closet ✗ MOM: lack ✗ SANDY: vector ✓



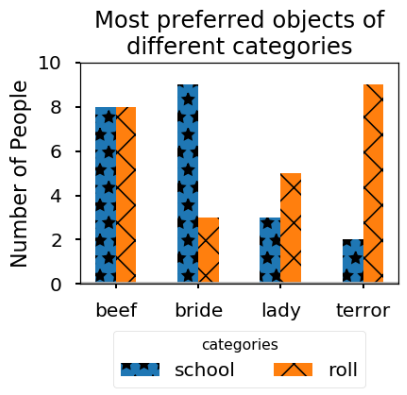
(c)

Q: Is the value of output smaller than demand?
SAN: no ✗ MOM: no ✗ SANDY: yes ✓



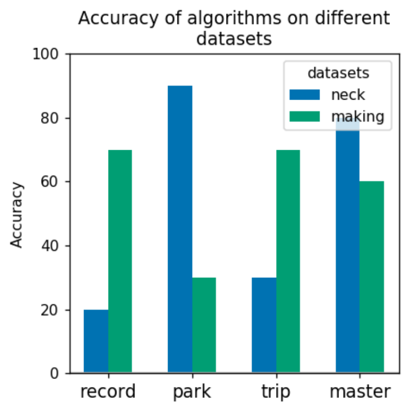
(d)

Q: Which algorithm has the smallest accuracy summed across all the datasets?
SAN: closet ✗ MOM: fil ✗ SANDY: editor ✗
Q: What is the highest accuracy reported in the whole chart?
SAN: 60 ✗ MOM: 60 ✗ SANDY: 60 ✗



(e)

Q: How many total people preferred the object terror across all the categories?
SAN: 10 ✗ MOM: 10 ✗ SANDY: 10 ✗
Q: How many people prefer the object terror in the category roll?
SAN: 1 ✗ MOM: 1 ✗ SANDY: 9 ✓



(f)

Q: What is the highest accuracy reported in the whole chart?
SAN: 90 ✓ MOM: 90 ✓ SANDY: 80 ✗
Q: Which algorithm has the smallest accuracy summed across all the datasets?
SAN: closet ✗ MOM: record ✓ SANDY: park ✗

Figure 6: Some failure cases for different algorithms on the Test-Familiar split of the dataset.