

PlotQA: Reasoning over Scientific Plots

Nitesh Methani *

Pritha Ganguly*

Mitesh M. Khapra

Pratyush Kumar

Department of Computer Science and Engineering
 Robert Bosch Centre for Data Science and AI (RBC-DSAI)
 Indian Institute of Technology Madras, Chennai, India
 {nmethani, prithag, miteshk, pratyush}@cse.iitm.ac.in

Abstract

Existing synthetic datasets (FigureQA, DVQA) for reasoning over plots do not contain variability in data labels, real-valued data, or complex reasoning questions. Consequently, proposed models for these datasets do not fully address the challenge of reasoning over plots. In particular, they assume that the answer comes either from a small fixed size vocabulary or from a bounding box within the image. However, in practice, this is an unrealistic assumption because many questions require reasoning and thus have real-valued answers which appear neither in a small fixed size vocabulary nor in the image. In this work, we aim to bridge this gap between existing datasets and real-world plots. Specifically, we propose PlotQA with 28.9 million question-answer pairs over 224,377 plots on data from real-world sources and questions based on crowd-sourced question templates. Further, 80.76% of the out-of-vocabulary (OOV) questions in PlotQA have answers that are not in a fixed vocabulary. Analysis of existing models on PlotQA reveals that they cannot deal with OOV questions: their overall accuracy on our dataset is in single digits. This is not surprising given that these models were not designed for such questions. As a step towards a more holistic model which can address fixed vocabulary as well as OOV questions, we propose a hybrid approach: Specific questions are answered by choosing the answer from a fixed vocabulary or by extracting it from a predicted bounding box in the plot, while other questions are answered with a table question-answering engine which is fed with a structured table generated by detecting visual elements from the image. On the existing DVQA dataset, our model has an accuracy of 58%, significantly improving on the highest reported accuracy of 46%. On PlotQA, our model has an accuracy of 22.52%, which is significantly better than state of the art models.

1. Introduction

Data plots such as bar charts, line graphs, scatter plots, etc. provide an efficient way of summarizing numerical information. Recently, in [13, 12] two datasets containing plots and deep neural models for question answering over the generated plots have been proposed. In both the datasets, the plots are synthetically generated with data values and labels drawn from a custom set. In the FigureQA dataset [13], all questions are binary wherein answers are either Yes or No, (see Figure 1a for an example). The DVQA dataset [12], generalizes this to include questions which can be answered either by (a) fixed vocabulary of 1000 words, or (b) extracting text (such as tick labels) from the plot. An example question could seek the numeric value represented by a bar of a specific label in a bar plot (see Figure 1b). Given that all data values in the DVQA dataset are chosen to be integers and from a fixed range, the answer to this question can be extracted from the appropriate tick label.

While these datasets have initiated the research questions on plot reasoning, realistic questions over plots are much more complex. For instance, consider the question in Figure 1c, where we are to compute the average of floating point numbers represented by three bars of a color specified by the label. The answer to this question is neither in a fixed vocabulary nor can it be extracted from the plot itself. Answering such questions requires a combination of perception, language understanding, and reasoning, and thus poses a significant challenge to existing systems. Furthermore, this task is harder if the training set is not synthetic, but instead is sourced from real-world data with large variability in floating-point values, large diversity in axis and tick labels, and natural complexity in question templates.

To address this gap between existing datasets and real-world plots, we introduce the PlotQA¹ dataset with 28.9 million question-answer pairs grounded over 224,377 plots. PlotQA improves on existing datasets on three fronts. First,

*The first two authors have contributed equally

¹The dataset can be downloaded from bit.ly/PlotQA

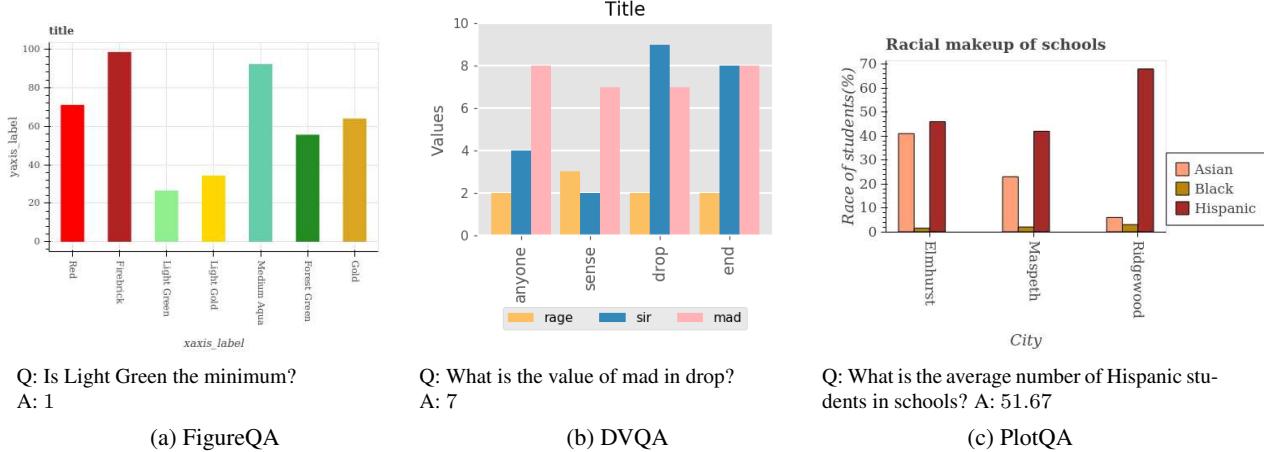


Figure 1: A sample {plot, question, answer} triplet from FigureQA, DVQA, and PlotQA (our) datasets.

Answer Type	Question Type		
	Structure	Data Retrieval	Reasoning
Yes/No	Does the graph contain grids?	Does the price of diesel in Barbados monotonically increase over the years?	Is the difference between the price of diesel in Angola in 2002 and 2004 greater than the difference between the price of diesel in Lebanon in 2002 and 2004?
Fixed vocabulary	How are the legend labels stacked?	What is the label or title of the X-axis?	In how many years, is the price of diesel greater than 0.6 units?
Open vocabulary	-	What is the price of diesel in Lebanon in the year 2008?	What is the ratio of the price of diesel in Lebanon in 2010 to that in 2014?

Table 1: Sample questions for 9 different question-answer types in PlotQA. The example questions are with respect to the plot in Figure 2b. Note that there are no open vocabulary answers for *Structural Understanding* questions.

roughly 80.76% of the questions have answers which are not present in the plot or in a fixed vocabulary. Second, the plots are generated from data sourced from World Bank, government sites, etc., thereby having a large vocabulary of axis and tick labels, and a wide range in data values. Third, the questions are complex as they are generated based on 74 templates extracted from 7,000 crowd-sourced questions asked by workers on a sampled set of 1,400 plots. Questions are categorized into 9 (=3x3) cells based on the question type: ‘Structural Understanding’, ‘Data Retrieval’, or ‘Reasoning’ and the answer type: ‘Yes/No’, ‘From Fixed Vocabulary’, or ‘Out Of Vocabulary (OOV)’ (see Table 1).

We first evaluate three state of the art models on PlotQA, *viz.*, SAN-VQA[36], Bilinear attention network (BAN) [16] and LoRRA [33]. Note that, by design none of these models are capable of answering OOV questions. In particular, SAN-VQA and BAN treat plot reasoning as a classification task and expect the answer to lie in a small vocabulary whereas in our dataset the answer vocabulary is prohibitively large ($\sim 5M$ words). Similarly, LoRRA assumes that the answer is present in the image itself as a text and the task is to just extract this region containing the text followed by OCR (optical character recognition). Again, such

a model will be unable to answer questions such as the one shown in Figure 1c, which form a significant segment of real-world use-cases and our dataset. As a result, these models give an accuracy of less than 8% on our dataset. On the other hand, existing models (in particular, SAN) perform well on questions with answers from a fixed vocabulary, which was the intended purpose of these models.

Based on the above observations, we propose a hybrid model with a binary classifier which given a question decides if the answer would lie in a small top- k vocabulary or if the answer is OOV. For the former, the question is passed through a classification pipeline which predicts a distribution over the top- k vocabulary and selects the most probable answer. For the latter (arguably harder questions), we pass the question through a pipeline of four modules: Visual element detection, Optical character recognition, Extraction into a structured table, and Structured table question answering. This proposed hybrid model significantly outperforms the existing models and has an aggregate accuracy of 22.52% on the PlotQA dataset. We also evaluate our model on the DVQA dataset where it gives an accuracy of 58%, improving on the best-reported result of SANDY [12] of 46%. In summary, we make two major contributions:

- (1) We propose PlotQA dataset with plots on data sourced from the real-world and questions based on templates sourced from manually curated questions. The dataset exposes the need to train models for questions that have answers from an Open Vocabulary.
- (2) We propose a hybrid model with perception and QA modules for questions that have answers from an Open Vocabulary. This model gives the best performance not only on our dataset but also on the existing DVQA dataset.

2. Related Work

Datasets: Over the past few years several large scale datasets for Visual Question Answering have been released. These include datasets such as COCO-QA [28], DAQUAR [23], VQA [1, 7] which contain questions asked over natural images. On the other hand, datasets such as CLEVR [11] and NVLR [35] contain complex reasoning based questions on synthetic images having 2D and 3D geometric objects. There are some datasets [14, 15] which contain questions asked over diagrams found in text books but these datasets are smaller and contain multiple-choice questions. FigureSeer [31] is another dataset which contains images extracted from research papers but this is also a relatively small (60,000 images) dataset. Further, FigureSeer focuses on answering questions based on line plots as opposed to other types of plots such as bar charts, scatter plots, *etc.* as seen in FigureQA [13] and DVQA [12]. There is also the recent TextVQA [33] dataset which contains questions which require models to read the text present in natural images. This dataset does not contain questions requiring numeric reasoning. Further, the answer is contained as a text in the image itself. Thus, *no* existing dataset contains plot images with complex questions which require reasoning and have answers from an Open Vocabulary.

Models: The availability of the above mentioned datasets has facilitated the development of complex end-to-end neural network based models ([36], [22], [37], [25], [30], [12], [33]). These end-to-end networks contain (a) encoders to compute a representation for the image and the question, (b) attention mechanisms to focus on important parts of the question and image, (c) interaction components to capture the interactions between the question and the image, (d) OCR module to extract the image specific text and (e) a classification layer for selecting the answer either from a fixed vocabulary or from a OCR appended vocabulary.

3. The PlotQA dataset

In this section, we describe the PlotQA dataset and the process to build it. Specifically, we discuss the four main stages, *viz.*, (i) curating data such as year-wise rainfall statistics, country-wise mortality rates, *etc.*, (ii) creating different types of plots with a variation in the number of ele-

ments, legend positions, fonts, *etc.*, (iii) crowd-sourcing to generate questions, and (iv) extracting templates from the crowd-sourced questions and instantiating these templates using appropriate phrasing suggested by human annotators.

3.1. Data Collection and Curation

We considered online data sources such as World Bank Open Data, Open Government Data, Global Terrorism Database, *etc.* which contain statistics about various indicator variables such as fertility rate, rainfall, coal production, *etc.* across years, countries, districts, *etc.* We crawled data from these sources to extract different variables whose relations could then be plotted (for example, rainfall v/s years across countries, or movie v/s budget, or carbohydrates v/s food.item). There are a total of 841 unique indicator variables (CO2 emission, Air Quality Index, Fertility Rate, Revenue generated, etc.) with 160 unique entities (cities, states, districts, countries, movies, food, *etc.*). The data ranges from 1960 to 2016, though not all indicator variables have data items for all years. The data contains positive integers, floating point values, percentages, and values on a linear scale. These values range from 0 to 3.50e+15.

3.2. Plot Generation

We included 3 different types of plots in this dataset, *viz.*, bar plots, line plots, and scatter plots. Within bar plots, we have grouped them by orientation as either horizontal or vertical. Figure 2 shows one sample of each plot type. Each of these plot types can compactly represent 3-dimensional data. For instance, in Figure 2b, the plot compares the diesel prices across years for different countries. To enable the development of supervised modules for various sub-tasks, we provide bounding box annotations for legend boxes, legend names, legend markers, axes titles, axes ticks, bars, lines, and title. By using different combinations of indicator variables and entities (years, countries, *etc.*) we created a total of 224, 377 plots.

To ensure variety in the plots, we randomly chose the following parameters: grid lines (present/absent), font size, notation used for tick labels (scientific-E notation or standard notation), line style (solid, dashed, dotted, dash-dot), marker styles for marking data points (asterisk, circle, diamond, square, triangle, inverted triangle), position of legends (bottom-left, bottom-centre, bottom-right, center-right, top-right), and colors for the lines and bars from a set of 73 colors. The number of discrete elements on the *x*-axis varies from 2 to 12 and the number of entries in the legend box varies from 1 to 4.

3.3. Sample Question Collection by Crowd-sourcing

As the source data of PlotQA dataset is significantly richer in comparison to FigureQA and DVQA, we found

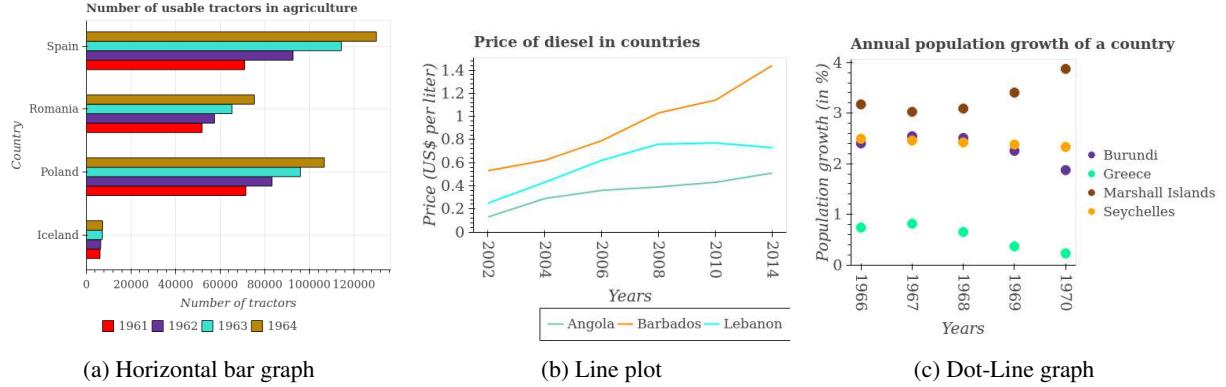


Figure 2: Sample plots of different types in the PlotQA dataset.

Answer (A) Type	Question (Q) Type		
	Structure	Data Retrieval	Reasoning
Yes/No	36.99%	5.19%	2.05%
Fixed vocabulary	63.01%	18.52%	15.92%
Open vocabulary	0.00%	76.29%	82.03%

Table 2: Distribution of Q and A types in PlotQA.

it necessary to ask a larger set of annotators to create questions over these plots. However, creating questions for all the plots in our dataset would have been prohibitively expensive. We sampled 1,400 plots across different types and asked workers on Amazon Mechanical Turk to create questions for these plots. We showed each plot to 5 different workers resulting in a total of 7,000 questions. We specifically instructed the workers to ask complex reasoning questions which involved reference to multiple plot elements in the plots. We paid the workers USD 0.1 for each question.

3.4. Question Template Extraction & Instantiation

We manually analyzed the questions collected by crowdsourcing and divided them into a total of 74 templates. These templates were divided into 3 question categories. These question categories along with a few sample templates are shown below. See Table 2 for statistics of different question and answer types in our dataset (please refer to the Supplementary material for further details).

Structural Understanding: These are questions about the overall structure of the plot and do not require any quantitative reasoning. Example: “How many bars are there?”.

Data Retrieval: These questions seek data item for a single element in the plot. Example: “What is the number of tax payers in Myanmar in 2015?”.

Reasoning: These questions either require numeric reasoning over multiple plot elements or a comparative analysis of different elements of the plot, or a combination of both to answer the question. Example: “What is the median ba-

nana production?”. We abstracted the questions into templates such as “In how many <plural form of X_label>, is the <Y_label> of/in <legend_label> greater than the average <Y_label> of/in <legend_label> taken over all <plural form of X_label>?”. We could then generate multiple questions for each template by replacing X_label, Y_label, legend_label, etc. by indicator variables, years, cities etc. from our curated data. However, this was a tedious task requiring a lot of manual intervention. For example, consider the indicator variable “Race of students” in Figure 1c. If we substitute this indicator variable as it is in the above template, it would result in a question, “In how many cities, is the race of the students(%) of Asian greater than the average race of the students (%) of Asian taken over all cities?”, which sounds unnatural. To avoid this, we asked in-house annotators to carefully paraphrase these indicator variables and question templates. The paraphrased version of the above example was “In how many cities, is the percentage of Asian students greater than the average percentage of Asian students taken over all cities?”. Using this semi-automated process we generated a total of 28,952,641 questions. This approach of creating questions on real-world plot data with carefully curated question templates followed by manual paraphrasing is a key contribution of our work. The resultant PlotQA dataset is much closer to the real-world challenge of reasoning over plots, significantly improving on existing datasets. Table 3 summarizes the differences between PlotQA and existing datasets (FigureQA, DVQA). Note that (a) the number of unique answers in PlotQA is very large, (b) the questions in PlotQA are much longer, and (c) the vocabulary of PlotQA is more realistic than FigureQA or DVQA.

4. Proposed Model

Existing models for VQA are of two types: (i) read the answer from the image (as in LoRRA) or (ii) pick the answer from a fixed vocabulary (as in SAN and BAN). Such

Datasets	#Plot types	#Plot images	#QA pairs	Vocabulary	Avg. question length	#Templates	#Unique answers	Open vocab.
FigureQA	4	180,000	2,388,698	100 colours from X11 colour set	7.5	15 (no variations)	2	Not present
DVQA	1	300,000	3,487,194	1K nouns from Brown corpus	12.30	26 (without paraphrasing)	1576	Not present
PlotQA	3	224,377	28,952,641	Real-world axes variables and floating point numbers	43.54	74 (with paraphrasing)	5,701,618	Present

Table 3: Comparison between the existing datasets (FigureQA and DVQA) and our proposed dataset (PlotQA).

models work well for datasets such as DVQA where indeed all answers come from a fixed vocabulary (global or plot specific) but are not suited for PlotQA with a large number of OOV questions. Answering such questions involves various sub-tasks: (i) detect all the elements in the plot (bars, legend names, tick labels, etc), (ii) read the values of these elements, (iii) establish relations between the plot elements, e.g., creating tuples of the form {country=Angola, year=2006, price of diesel = 0.4 }, and (iv) reason over this structured data. Expecting a single end-to-end model to be able to do all of this is unreasonable. Hence, we propose a multi-staged pipeline to address each of the sub-tasks.

We further note that for simpler questions which do not require reasoning and can be answered from a small fixed size vocabulary, such an elaborate pipeline is an overkill. As an illustration consider the question “How many bars are there in the image?”. This does not require reasoning and can be answered based on visual properties of the image. For such questions, we have a simpler *QA-as-classification* pipeline. As shown in Figure 3, our overall model is thus a hybrid model containing the following elements: (i) a binary classifier for deciding whether the given question can be answered from a small fixed vocabulary or needs more complex reasoning, and (ii) a simpler *QA-as-classification* model to answer questions of the former type, and (iii) a multi-staged model containing four components as described below to deal with complex reasoning questions.

4.1. Visual Elements Detection (VED)

The data bearing elements of a plot are of 10 distinct classes: the title, the labels of the x and y axes, the tick labels (e.g., countries) on the x and y axis, the data markers in the legend box, the legend names, and finally the bars and lines in the graph. Following existing literature ([4], [12]), we refer to these elements as the *visual elements* of the graph. The first task is to extract all these visual elements by drawing bounding boxes around them and classifying them into the appropriate class. To this end, we can either apply object detection models such as Fast-RCNN [6], YOLO [27], SSD [21], Mask-RCNN [9], etc. Upon comparing all methods, we found that Faster R-CNN [29] model along with Feature Pyramid Network(FPN) [20] performed

the best and hence we used it as our VED module.

4.2. Object Character Recognition (OCR)

Some of the visual elements such as title, legends, tick labels, etc. contain numeric and textual data. For extracting this data from within these bounding boxes, we use a state-of-the-art OCR model [34]. We crop the detected visual element to its bounding box, convert the cropped textual image into gray-scale, resize and deskew it, and then pass it to an OCR module. Existing OCR modules perform well for machine-written English text, and indeed we found that a pre-trained OCR module² works well on our dataset.

4.3. Semi-Structured Information Extraction (SIE)

The next stage of extracting the data into a semi-structured table is explained with the help of the plot image in Figure 3. The desired output of SIE is a table where the rows correspond to the ticks on the x -axis (2002, 2003, 2004, 2005), the columns correspond to the different elements listed in the legend (Bulgaria, Cuba) and the i,j -th cell contains the value corresponding to the x -th tick and the y -th legend. The values of the x -tick labels and the legend names are available from the OCR module. The mapping of legend name to legend marker or color is done by associating a legend name to the marker or color whose bounding box is closest to the bounding box of the legend name. Similarly, we associate each tick label to the tick marker whose bounding box is closest to the bounding box of the tick label. For example, we associate the legend name Cuba to the color “Purple” and the tick label 2004 to the corresponding tick mark on the x -axis. With this we have the 4 row and 2 column headers. To fill the 8 values in the table, there are again two smaller steps. First, we associate each of the 8 bounding boxes of the 8 bars to their corresponding x -ticks and legend names. A bar is associated with an x -tick label whose bounding box is closest to the bounding box of the bar. To associate a bar to a legend name, we find the dominant color in the bounding box of the bar and match it with a legend name corresponding to that color. Second, to find the value represented by each bar, we extract the height of the bar using bounding box information and then search for

²<https://github.com/tesseract-ocr/tesseract>

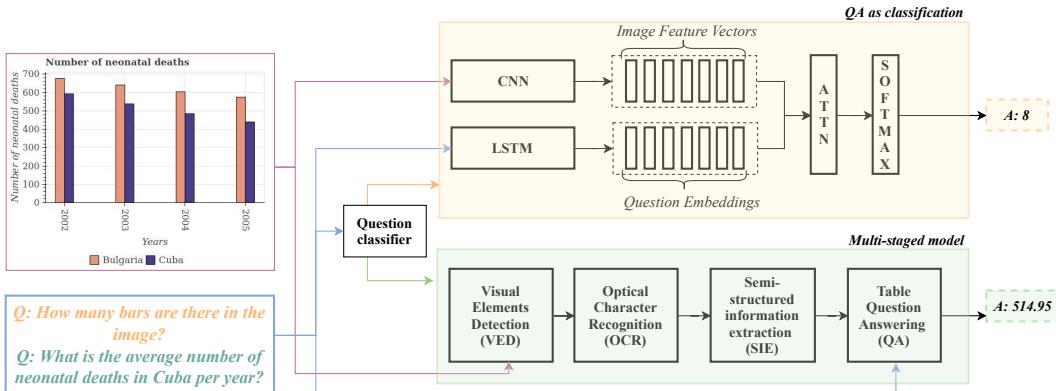


Figure 3: Our proposed model containing (i) a question classifier for deciding whether the question can be answered from a fixed vocabulary (orange) or needs more complex reasoning (green), (ii) *QA-as-classification* model to answer questions of the former type, and (iii) *multi-staged model* as a pipeline of perception and QA modules for answering complex questions.

the y -tick labels immediately above and below that height. We then interpolate the value of the bar based on the values of these bounding ticks. With this we extracted the information from the plot into a semi-structured table.

4.4. Table Question Answering (QA)

The final stage of the pipeline is to answer questions on the semi-structured table. As this is similar to answering questions from the WikiTableQuestions dataset [26], we adopt the same methodology as proposed in [26]. In this method, the table is converted to a knowledge graph and the question is converted to a set of candidate logical forms by applying compositional semantic parsing. These logical forms are then ranked using a log-linear model and the highest ranking logical form is applied to the knowledge graph to get the answer. Note that with this approach the output is computed by a logical form that operates on the numerical data. This supports complex reasoning questions and also avoids the limitation of using a small answer vocabulary for multi-class classification as is done in existing work on VQA. There are recent neural approaches for answering questions over semi-structured tables such as [24, 8]. Individually these models do not outperform the relatively simpler model of [26], but as an ensemble they show a small improvement of only (1-2%). To the best of our knowledge, there is only one neural method [19] which outperforms [26], but the code for this model is not available which makes it hard to reproduce the results.

5. Experiments

In this section we detail the data splits, baseline models, hyperparameter tuning and evaluation metrics.

Dataset Split	#Images	#QA pairs
Train	157,070	20,249,479
Validation	33,650	4,360,648
Test	33,657	4,342,514
Total	224,377	28,952,641

Table 4: PlotQA Dataset Statistics

5.1. Train-Valid-Test Splits

By using different combinations of 841 indicator variables and 160 entities (years, countries, *etc*), we created a total of 224,377 plots. Depending on the context and type of the plot, we instantiated the 74 templates to create meaningful {question, answer} pairs for each of the plots. We created train (70%), valid (15%) and test (15%) splits (Table 4). The dataset and the crowd-sourced questions can be downloaded from the link: bit.ly/PlotQA.

5.2. Models Compared

- **IMG-only:** This is a simple baseline where we just pass the image through a VGG19 and use the embedding of the image to predict the answer from a fixed vocabulary.
- **QUES-only:** This is a simple baseline where we just pass the question through a LSTM and use the embedding of the question to predict the answer from a fixed vocabulary.
- **SAN** [36]: This is an encoder-decoder model with a multi-layer stacked attention [2] mechanism. It obtains a representation for the image using a deep CNN and a representation for the query using LSTM. It then uses the query representation to locate relevant regions in the image and uses this to pick an answer from a fixed vocabulary.
- **SANDY** [12]: This is the best performing model on the DVQA dataset and is a variant of SAN. Unfortunately, the

code for this model is not available and the description in the paper was not detailed enough for us to reimplement it.³ Hence, we report the numbers for this model only on DVQA (from the original paper).

- **LoRRA** [33]: This is the recently proposed model on the TextVQA dataset. It concatenates the image features extracted from pre-trained ResNet-152 [10] model with the region based features extracted from Faster-RCNN [5] model. It then reads the text present in the image using a pre-trained OCR module and incorporates an attention mechanism to reason about the image and the text. Finally, it does multi-class classification where the answer either comes from a fixed vocabulary or is copied from the text in the image.

- **BAN** [16]: This model exploits bilinear interactions between two groups of input channels, *i.e.*, between every question word (GRU [3] features) and every image region (pre-trained Faster-RCNN [29] object features). It then uses low-rank bilinear pooling [17] to extract the joint distribution for each pair of channels. BAN accumulates 8 such bilinear attention maps which are then fed to a two-layer perceptron classifier to get the final joint distribution over answers from a fixed vocabulary.

- **Our Model:** This proposed model shown in Figure 3 with two model paths. The training data for the binary classification is generated by comparing the performance of the individual models: For a given question, the label is set to 1 if the performance of QA-as-classification model is better than the multi-stage pipeline, and 0 otherwise. We use an LSTM to represent the input question and then perform binary classification on this representation.

5.3. Training Details

SAN: We used an existing implementation of SAN⁴ for the initial baseline results. Image features are extracted from the last pooling layer of VGG19 network. Question features are the last hidden state of the LSTM. Both the LSTM hidden state and 512-d image feature vector at each location are transferred to a 1024-d vector by a fully connected layer, and added and passed through a non-linearity (tanh). The model was trained using Adam [18] with an initial learning rate of 0.0003 and a batch size of 128 for 25,000 iterations.

Our model: The binary question classifier in the proposed model contains a 50-dimensional word embedding layer followed by an LSTM with 128 hidden units. The output of the LSTM is projected to 256 dimensions and this is then fed to the output layer. The model is trained for 10 epochs using RMSProp with an initial learning rate of 0.001. Accuracy on the validation set is 87.3%. Of the 4 stages of the multi-stage pipeline, only two require training, *viz.*, Visual Elements Detection (VED) and Table Question Answering

³We have contacted the authors and while they are helpful in sharing various details, they do not have access to the original code now.

⁴<https://github.com/TingAnChien/san-vqa-tensorflow>

(QA). As mentioned earlier, for VED we train a variant of Faster R-CNN [20] with FPN using the bounding box annotations available in PlotQA. We trained the model with a batch size of 32 for 200,000 steps. We used RMSProp with an initial learning rate of 0.004. For Table QA, we trained the model proposed in [26] using questions from our dataset and the corresponding ground truth tables.

5.4. Evaluation Metric

We used accuracy as the evaluation metric. Specifically, for textual answers (such as India, CO₂, etc.) the model’s output was considered to be correct only if the predicted answer exactly matches the true answer. However, for numeric answers with floating point values, an exact match is a very strict metric. We relax the measure to consider an answer to be correct as if it is within 5% of the correct answer.

5.5. Human Accuracy on PlotQA dataset

To assess the difficulty of the PlotQA dataset, we report human accuracy on a small subset of the Test split of the dataset. With the help of in-house annotators, we were able to evaluate 5,860 questions grounded in 160 images. Human accuracy on this subset is found to be 80.47%. We used the evaluation metric as defined in section 5.4. Most human errors were due to numerical precision as it is difficult to find the exact value from the plot even with a 5% margin.

6. Observations and Results

1. Evaluating models on PlotQA dataset (Table 6):

The baselines IMG-only and QUES-only performed poorly with an accuracy of 4.84% and 5.35% respectively. Existing models (SAN, BAN, LoRRA) perform poorly on this dataset. In particular, BAN and LoRRA have an abysmal accuracy of less than 1%. This is not surprising given that both models are not designed to answer OOV questions. Further, the original VQA tasks for which BAN was proposed does not have any complex numerical reasoning questions as found in PlotQA. Similarly, LoRRA was designed only for text based answers and not for questions requiring numeric reasoning. Note that we have used the original code [32] released by the authors of these models. Given the specific focus and limited capabilities of these existing models it may even seem unfair to evaluate these models on our dataset but we still do so for the sake of completeness and to highlight the need for better models. Lastly, our model gives the best performance of 22.52% on the PlotQA dataset. Supplementary material contains details on the performance of each question type (structural, data retrieval, reasoning) and each answer type (binary, fixed vocabulary, OOV). We acknowledge that the accuracy is significantly lower than human performance. This establishes that the dataset is challenging and raises open questions on models for visual reasoning.

2. Analysis of the pipeline We analyze the performance of VED, OCR and SIE modules in the pipeline.

VED: Table 7 shows that the VED module performs reasonably well at an Intersection Over Union (IOU) of 0.5. For higher IOUs of 0.75 and 0.9, the accuracy falls drastically. For instance, at IOU of 0.9, dotlines are detected with an accuracy of under 20%. This brings out an interesting difference between this task and other instance segmentation tasks where the margin of error is higher (where IOU of 0.5 is accepted). A small error in visual element detection as indicated by mAP scores of 75% is considered negligible for VQA tasks, however for PlotQA small errors can cause significantly misaligned table generation and subsequent QA. We illustrate this with an example given in Figure 4. The predicted red box having an IOU of 0.58 estimates the bar size as 760 as opposed to ground truth of 680, significantly impacting downstream QA accuracy.

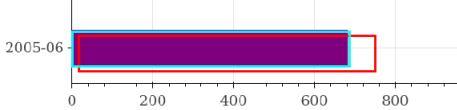


Figure 4: Ground-truth (cyan) and predicted (red) boxes.

OCR: We evaluate the OCR module in standalone/oracle mode and pipeline mode. In the oracle mode, we feed ground truth boxes to the OCR model whereas in the pipeline model we perform OCR on the output of the VED module. We observe only a small drop in performance from 97.06% (oracle) to 93.10% (after VED), which indicates that the OCR module is robust to the reduction in VED module’s accuracy at higher IOU as it does not depend on the class label or the exact position of bounding boxes.

SIE: We now evaluate the performance of the SIE module. We consider each cell in the table to be a tuple of the form {row header, column header, value } (e.g., {Poland, 1964, 10000 tractors}). We consider all the tuples extracted by the SIE module with the tuples present in the ground truth table to compute the F1-score. Even though Table 7 suggests that the VED model is very accurate with a mAP@0.5 of 96.43%, we observe that the F1-score for table extraction is only 0.68. This indicates that many values are not being extracted accurately due to the kind of errors shown in Figure 4 where the bounding box has a high overlap with the true box. We thus need better plot VED modules which can predict tighter bounding boxes (higher mAP at IOU of 0.9) around the plot’s visual and textual elements. Inaccurate VED module generates erroneous tables which further affects the downstream QA accuracy.

- In summary, a highly accurate VED for structured images is an open challenge to improve reasoning over plots.

3. Evaluating new models on the existing DVQA dataset

Model	DVQA (TEST)	DVQA (TEST-NOVEL)
SAN	32.1%	30.98%
SANDY-OCR	45.77%	45.81%
Our Model	57.99%	59.54%

Table 5: Accuracy of different models on DVQA dataset.

Models	IMG	QUES	BAN	LoRRA	SAN	Our Model
Accuracy	4.84	5.35	0.01	0.02	7.76	22.52

Table 6: Accuracy (in %) of different models on PlotQA.

Class	AP@0.5	AP@0.75	AP@0.9
Title	100.00%	78.83%	0.22%
Bar	95.84%	94.30%	85.54%
Line	72.25%	62.04%	37.65%
Dotline	96.30%	95.14%	18.07%
X-axis Label	99.99%	99.99%	99.09%
Y-axis Label	99.90%	99.90%	99.46%
X-tick Label	99.92%	99.74%	96.04%
Y-tick Label	99.99%	99.97%	96.80%
Legend Label	99.99%	99.96%	93.68%
Legend Preview	99.95%	99.94%	96.30%
mAP	96.43%	92.98%	72.29%

Table 7: VED Module’s Accuracy on PlotQA dataset

(Table 5): The proposed model performs better than the existing models (SAN and SANDY-OCR) establishing a new SOTA result on DVQA. The higher performance of the proposed hybrid model in comparison to SAN (in contrast to the PlotQA results) suggests that the extraction of the structured table is more accurate on the DVQA dataset. This is because of the limited variability in the axis and tick labels and shorter length (one word only) of labels.

7. Conclusion

We introduce the PlotQA dataset to reduce the gap between existing synthetic plot datasets and real-world plots and question templates. Analysis of existing VQA models on PlotQA reveals that they perform poorly for Open Vocabulary questions. This is not surprising as these models were not designed to handle complex questions which require numeric reasoning and OOV answers. We propose a hybrid model with separate pipelines for handling (i) simpler questions which can be answered from a fixed vocabulary and (ii) complex questions with OOV answers. For OOV questions, we propose a pipelined approach that combines visual element detection and OCR with QA over tables. The proposed model gives state-of-the-art results on both the DVQA and PlotQA datasets. Further analysis of our pipeline reveals the need for more accurate visual element detection to improve reasoning over plots.

References

- [1] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. VQA: visual question answering. In *ICCV*, 2015.
- [2] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.
- [3] K. Cho, B. van Merriënboer, Ç. Gülcəhre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1724–1734, 2014.
- [4] M. Cliche, D. S. Rosenberg, D. Madeka, and C. Yee. Scatteract: Automated extraction of data from scatter plots. In *ECML PKDD*, 2017.
- [5] R. Girshick, I. Radosavovic, G. Gkioxari, P. Dollár, and K. He. Detectron. <https://github.com/facebookresearch/detectron>, 2018.
- [6] R. B. Girshick. Fast R-CNN. *CoRR*, abs/1504.08083, 2015.
- [7] 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.
- [8] T. Haug, O. Ganea, and P. Grnarova. Neural multi-step reasoning for question answering on semi-structured tables. In *Advances in Information Retrieval - 40th European Conference on IR Research, ECIR 2018, Grenoble, France, March 26-29, 2018, Proceedings*, pages 611–617, 2018.
- [9] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick. Mask R-CNN. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2980–2988, 2017.
- [10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778, 2016.
- [11] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. B. Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, 2017.
- [12] K. Kafle, S. Cohen, B. L. Price, and C. Kanan. DVQA: understanding data visualizations via question answering. *CoRR*, abs/1801.08163, 2018.
- [13] S. E. Kahou, A. Atkinson, V. Michalski, Á. Kádár, A. Trischler, and Y. Bengio. Figureqa: An annotated figure dataset for visual reasoning. *CoRR*, abs/1710.07300, 2017.
- [14] A. Kembhavi, M. Salvato, E. Kolve, M. J. Seo, H. Hajishirzi, and A. Farhadi. A diagram is worth a dozen images. In *ECCV*, 2016.
- [15] A. Kembhavi, M. J. Seo, D. Schwenk, J. Choi, A. Farhadi, and H. Hajishirzi. Are you smarter than a sixth grader? textbook question answering for multimodal machine comprehension. In *CVPR*, 2017.
- [16] J. Kim, J. Jun, and B. Zhang. Bilinear attention networks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pages 1571–1581, 2018.
- [17] J. Kim, K. W. On, W. Lim, J. Kim, J. Ha, and B. Zhang. Hadamard product for low-rank bilinear pooling. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.
- [18] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.
- [19] J. Krishnamurthy, P. Dasigi, and M. Gardner. Neural semantic parsing with type constraints for semi-structured tables. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1516–1526, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics.
- [20] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie. Feature pyramid networks for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 936–944, 2017.
- [21] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg. SSD: single shot multibox detector. In *ECCV*, 2016.
- [22] J. Lu, J. Yang, D. Batra, and D. Parikh. Hierarchical question-image co-attention for visual question answering. In *NIPS*, 2016.
- [23] M. Malinowski and M. Fritz. A multi-world approach to question answering about real-world scenes based on uncertain input. In *NIPS*, 2014.
- [24] A. Neelakantan, Q. V. Le, M. Abadi, A. McCallum, and D. Amodei. Learning a natural language interface with neural programmer. *CoRR*, abs/1611.08945, 2016.
- [25] H. Noh and B. Han. Training recurrent answering units with joint loss minimization for VQA. *CoRR*, abs/1606.03647, 2016.
- [26] P. Pasupat and P. Liang. Compositional semantic parsing on semi-structured tables. In *ACL*, 2015.
- [27] J. Redmon and A. Farhadi. YOLO9000: better, faster, stronger. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 6517–6525, 2017.
- [28] M. Ren, R. Kiros, and R. S. Zemel. Image question answering: A visual semantic embedding model and a new dataset. *CoRR*, abs/1505.02074, 2015.
- [29] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 91–99, 2015.
- [30] A. Santoro, D. Raposo, D. G. T. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap. A simple neural network module for relational reasoning. In *NIPS*.

- [31] N. Siegel, Z. Horvitz, R. Levin, S. K. Divvala, and A. Farhadi. Figureseer: Parsing result-figures in research papers. In *ECCV*, 2016.
- [32] A. Singh, V. Natarajan, Y. Jiang, X. Chen, M. Shah, M. Rohrbach, D. Batra, and D. Parikh. Pythia-a platform for vision & language research. In *SysML Workshop, NeurIPS*, volume 2018, 2018.
- [33] A. Singh, V. Natarajan, M. Shah, Y. Jiang, X. Chen, D. Batra, D. Parikh, and M. Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.
- [34] R. Smith. An overview of the tesseract ocr engine. In *ICDAR*, 2007.
- [35] A. Suhr, M. Lewis, J. Yeh, and Y. Artzi. A corpus of natural language for visual reasoning. In *ACL*, 2017.
- [36] Z. Yang, X. He, J. Gao, L. Deng, and A. J. Smola. Stacked attention networks for image question answering. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 21–29, 2016.
- [37] Z. Yang, X. He, J. Gao, L. Deng, and A. J. Smola. Stacked attention networks for image question answering. In *CVPR*, 2016.