FlowchartQA: The First Large-Scale Benchmark for Reasoning over Flowcharts

Simon Tannert¹  Marcelo Feighelstein²  Jasmina Bogojeska³  Joseph Shtok⁴  
Assaf Arbelle⁴  Peter Staar³  Anika Schumann³  Jonas Kuhn¹  Leonid Karlinsky⁵

¹Institute for Natural Language Processing, University of Stuttgart, Germany  
²Data Science Research Center, University of Haifa, Israel  
³IBM Research, Rüschlikon, Switzerland ⁴IBM Research, Haifa, Israel  
⁵MIT-IBM AI-Watson Lab

Abstract

Flowcharts are a very popular type of diagram in many kinds of documents, conveying large amounts of useful information and knowledge (e.g. on processes, workflows, or causality). In this paper, we propose FlowchartQA – a novel, and first of its kind, large scale benchmark with close to 1M flowchart images and 6M question-answer pairs. The questions in FlowchartQA cover different aspects of geometric, topological, and semantic information contained in the charts, and are carefully balanced to reduce biases. We accompany our proposed benchmark with a comprehensive set of baselines based on text-only, image and graph, together with a set of qualitative analysis and comprehensive ablation studies (provided in the supplementary material) in order to establish a good basis for future work. We report some interesting findings following our baseline experiments, and believe that FlowchartQA will provide the community with a (currently absent) testbed for flowchart understanding.

1 Introduction

Flowcharts and other graph-like charts are very valuable sources of information used to intuitively communicate complex processes, guidelines, workflows, systems and algorithms. They contain text, use various shapes, such as rectangles, ovals, diamonds, and can have directed edges to define sequence or flow, or undirected edges to define relations. Since they are easy to understand by both technical and non-technical people, they are widely used in numerous fields such as science, education, engineering, manufacturing, healthcare, finance, sales and marketing. Machine understanding of such rich visual information would enable easy, focused access to a large amount of relevant valuable data for automated knowledge extraction systems. However, we found that no currently available benchmark / dataset offers any large scale data for training / evaluating flowchart understanding models.

Therefore, inspired by recent advances and successes in addressing language-vision problems and the importance that datasets like FigureQA [Kahou et al. [2018]], PlotQA [Methani et al. [2020]], and DVQA [Kafle et al. [2018]] played for developing and evaluating many state-of-the-art approaches for other types of charts (bar, pie, line and scatter plots), we introduce FlowchartQA – a first of its kind benchmark for question answering on flowcharts. It is a large synthetic corpus of 6M question-answer pairs corresponding to 1M flowchart images with corresponding ground truth annotations, created

to enable systematic research and development of methods for machine comprehension for this important chart type. More specifically, the final FlowchartQA dataset contains a grayscale plot image of the graph along with all the metadata providing the node positions and labels, the edge positions and labels, the question, the answer and a multiple choice answer. FlowchartQA contains three types of questions over graph charts, namely, geometric, topological and semantic. The code used for generating the flowchart images and ground truth data will also be published.

Another focus of this work is the problem of visual QA over flowcharts. For this purpose we propose, evaluate and ablate three baselines that utilize powerful neural architectures and concepts, such as transformers and attention, two of which are designed to also make use of the different modalities of the input data comprising text, image and graph. One of the considered approaches such as the text transformers use solely the text input, namely the question and the possible answers, to accurately evaluate the biases of the questions and answers in the dataset. The second one combines a text and a vision transformer to be able to ingest these two input modalities. We further consider an transformer-based approach that has access to the available ground truth information on the graph along with the QA text.

The main contributions of our paper are:

1. Large flowchart dataset with ground truth and QA annotations;
2. Code for controlled generation of diverse graph charts coupled with various question types that can potentially be adapted to generate data relevant for a specific target task;
3. Three neural baseline approaches for the multiple choice visual QA task over flowcharts: based on text transformers and a combination of text and visual transformers.

The rest of the paper is organized as follows. We start by providing an overview of the relevant related work. Then we describe the benchmark flowchart QA dataset and give the details of the generation process. After this we provide the details of the considered neural network approaches for addressing the visual QA over flowcharts problem. We also describe, analyze and discuss the results from the experimental evaluation and round up the paper with a conclusion.

2 Related Work

2.1 Visual QA datasets and algorithms

Generally, visual question-answering (VQA) was developed for natural images [Yu et al. 2017, 2019, 2020], but was recently applied for documents with figures and diagrams. Among the first and important works is FigureQA [Kahou et al. 2018], addressing the task of analysing different types of charts in the documents, by introducing a large synthetic chart dataset for training. This work uses CNN and LSTM architectures to encode image and text and a classifier for (binary) question answers based on these representations.

Another synthetic dataset, focusing on the bar charts, was introduced in DVQA [Kafle et al. 2018]; this work also introduced a neural model for question answering on charts, involving again CNN and LSTM and relying on high-quality OCR; in particular it enables to extract tabular data by appropriate sets of questions. Recently, PlotQA [Methani et al. 2020], brought the synthetic graphics closer to real world by using real tabular data to generate the figures for training.

2.2 Multi-modal Transformer-based VQA architectures

Transformers [Vaswani et al. 2017] recently were used in computer vision as alternatives to CNNs and have been used extensively for vision tasks such as the Vision Transformer (ViT) [Dosovitskiy et al. 2021]. In particular, they find applications in VQA domain: [Biten et al. 2021] use layout-aware transformers to answer questions by utilizing the scene text in the image, and [Mnih 2020] integrate BERT (a transformer-based language network [Devlin et al. 2019]) for embedding text with convolutional models to represent images.
Another use of a language based model was shown in Luo et al. [2022], where the GPT2 model Alec et al. [2019] has been used as the decoder to facilitate image captioning tasks. This and other multi-modal architectures integrating Transformers for combined Vision-Language tasks Su et al. [2020], Lu et al. [2019], Li et al. [2020] have also shown great benefits of such multi-modal Vision-Language models for visual reasoning and question answering. Following this line of research, we use ViT for producing visual representations of the flowchart images in our baselines.

### 2.3 Charts analysis and QA

Related to QA on flowcharts is the task of regular chart analysis. Early works addressing automatic chart classification and data extraction Savva et al. [2011], Al-Zaidy and Giles [2015], used classical computer vision techniques, such as codebooks obtained by clustering normalized image patches, connected components (for bars), Hough transform (for pies) and OCR, Al-Zaidy and Giles [2015] was extended in Al-Zaidy et al. [2016] to include chart summarization based on the extracted data. More recently, Poco and Heer [2017], Dai et al. [2018], Cliche et al. [2017] have presented hybrid neural-algorithmic pipelines, performing detection of the graphical objects with following extraction of numerical and textual information using OCR, Computer Vision techniques and rules; our approach belongs to this group of methods in terms of its general design. Other line of works Liu et al. [2019], Zhou et al. [2021] proposes an end-to-end analysis of the charts by a neural network. Zhou et al. [2021] develops an encoder-decoder architecture an attention mechanism for direct data extraction from bar charts by an RNN. Scatter plots are treated in Cliche et al. [2017] by using bounding boxes proposals of a detector for the points, tick marks and values. In Liu et al. [2019] a standard object detector is equipped with a relation network to address the connections between the different chart elements, such as the individual bars, the legend entries and the numerical and label axes; this model is able to produce bar heights and angles of pie segments (for single pie chart), and to match them against the legend entries. In contrast, in baselines presented in this paper we take the more generic approach, learning to answer questions about flowcharts without explicitly modeling the structure of nodes and edges and the graphical variations.

### 3 Dataset

We introduce a large, novel, synthetic dataset for question answering and reasoning on flowcharts. Our dataset comprises images of flowcharts together with annotations of the underlying data, the bounding boxes and outline polygons of nodes and edges, textual labels and the adjacency matrix of the depicted graph. We also provide questions, answers and multiple choice answer candidates, covering a large number of graph properties.

The dataset creation process is fully automatic which allows us to create large-scale datasets and parameterized so the creation process can be adapted to various different domains. Graphs can be directed or undirected, contain different numbers of nodes and edges, various node and edge styles and textual or numeric edge labels. We generate questions and corresponding answers for each graph from a rich set of templates which can be extended for domain adaptation. The final output contains a grayscale plot image of the graph along with all the metadata providing the node positions and labels, the edge positions and labels, the question, the answer and multiple choice answers. In the following we will describe the generation steps in more detail.

#### 3.1 Graph generation

The first step is the generation of a graph which can be parameterized in multiple ways. Among others, we control for the maximum number of nodes and edges in the graph, the maximum degree of each node and whether edges are directed or undirected. Edges can have textual or numeric labels or be unlabeled and nodes and edges can have different styles.
To generate a graph, a random number of nodes is generated within the selected range and node labels are drawn from the provided vocabulary. Edges are then randomly added to the set of nodes according to the constraints given by the generation parameters and edge labels are generated.

The generated graph is laid out and rendered using the graphviz dot engine. We obtain two different versions of the image during rendering, a colored image on which nodes are colored red and edges green and a gray scale image which serves as final output.

3.2 Ground truth data

Precise node bounding boxes can be obtained directly as an artefact of the rendering process. Getting ground truth data for edges is more challenging, as they may be curved and intersecting other edges and nodes. From graphviz, we obtain polygons roughly enclosing the edges; for exact binary images depicting the edges we additionally render the flowchart images in color and extract the edgemaps. We provide the bounding boxes obtained from the graph rendering process as ground truth in the dataset.

3.3 QA generation

For each graph, we generate questions and answers for a large number of question templates. There are binary questions (e.g., Is <node> in the graph?, Do all nodes have the same shape?, Is this a directed graph?), questions with a numerical answer (e.g., How many nodes are in the graph?, What is the eccentricity of <node>?, How many strongly connected components are in the graph?) and questions that can be answered with a node label (e.g., What is the leftmost node on the image?, What is the node with the maximum degree in the graph?). We categorize the questions into three categories, geometric, topological and semantic based on the knowledge they require to answer them.

---

1https://graphviz.org/
The full list of questions can be seen in Table 1. The generated graph is loaded into networkx\textsuperscript{2} which allows us to analyze its topology and answer the questions.

<table>
<thead>
<tr>
<th>Question</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Is &lt;&gt; above &lt;&gt; on the image?</td>
<td>geometric</td>
</tr>
<tr>
<td>2. Is &lt;&gt; below &lt;&gt; on the image?</td>
<td>geometric</td>
</tr>
<tr>
<td>3. Is &lt;&gt; to the left of &lt;&gt; on the image?</td>
<td>geometric</td>
</tr>
<tr>
<td>4. Is &lt;&gt; to the right of &lt;&gt; on the image?</td>
<td>geometric</td>
</tr>
<tr>
<td>5. What is the bottommost node on the image?</td>
<td>geometric</td>
</tr>
<tr>
<td>6. What is the leftmost node on the image?</td>
<td>geometric</td>
</tr>
<tr>
<td>7. What is the rightmost node on the image?</td>
<td>geometric</td>
</tr>
<tr>
<td>8. What is the topmost node on the image?</td>
<td>geometric</td>
</tr>
<tr>
<td>9. Are there any two inverted edges?</td>
<td>topological</td>
</tr>
<tr>
<td>10. How many edges are in the graph?</td>
<td>topological</td>
</tr>
<tr>
<td>11. How many nodes are in the graph?</td>
<td>topological</td>
</tr>
<tr>
<td>12. How many steps are in the shortest path between &lt;&gt; and &lt;&gt;?</td>
<td>topological</td>
</tr>
<tr>
<td>13. How many strongly connected components are in the graph?</td>
<td>topological</td>
</tr>
<tr>
<td>15. Is &lt;&gt; a direct successor of &lt;&gt;?</td>
<td>topological</td>
</tr>
<tr>
<td>16. Is &lt;&gt; in the graph?</td>
<td>topological</td>
</tr>
<tr>
<td>17. Is there a node directly connected to itself?</td>
<td>topological</td>
</tr>
<tr>
<td>18. Is there a path starting from &lt;&gt; and ending at &lt;&gt; using &lt;&gt;?</td>
<td>topological</td>
</tr>
<tr>
<td>19. Is this a directed graph?</td>
<td>topological</td>
</tr>
<tr>
<td>20. Is this an undirected graph?</td>
<td>topological</td>
</tr>
<tr>
<td>21. What is the diameter of the graph?</td>
<td>topological</td>
</tr>
<tr>
<td>22. What is the eccentricity of &lt;&gt;?</td>
<td>topological</td>
</tr>
<tr>
<td>23. What is the maximum degree of nodes in the graph?</td>
<td>topological</td>
</tr>
<tr>
<td>24. What is the node with the maximum degree in the graph?</td>
<td>topological</td>
</tr>
<tr>
<td>25. What is the radius of the graph?</td>
<td>topological</td>
</tr>
<tr>
<td>26. What is the state reached if &lt;&gt; is equal to &lt;&gt;?</td>
<td>topological</td>
</tr>
<tr>
<td>1. Can we reach &lt;&gt; if &lt;&gt; is equal to &lt;&gt;?</td>
<td>semantic</td>
</tr>
<tr>
<td>2. Can we start from any node and arrive at any other node in the graph removing edge &lt;&gt;?</td>
<td>semantic</td>
</tr>
<tr>
<td>3. Do all nodes have the same shape?</td>
<td>semantic</td>
</tr>
<tr>
<td>4. Do all nodes have the same style?</td>
<td>semantic</td>
</tr>
<tr>
<td>5. Do we directly reach &lt;&gt; if &lt;&gt; is equal to &lt;&gt;?</td>
<td>semantic</td>
</tr>
<tr>
<td>6. Does &lt;&gt; connect &lt;&gt; with &lt;&gt;?</td>
<td>semantic</td>
</tr>
<tr>
<td>7. How many neighbors can be reached starting from &lt;&gt;?</td>
<td>semantic</td>
</tr>
<tr>
<td>8. Is &lt;&gt; connected to &lt;&gt;?</td>
<td>semantic</td>
</tr>
<tr>
<td>9. Is &lt;&gt; directly connected to &lt;&gt;?</td>
<td>semantic</td>
</tr>
<tr>
<td>10. Is it shorter to get from &lt;&gt; to &lt;&gt; if we go through &lt;&gt; than if we go through &lt;&gt;?</td>
<td>semantic</td>
</tr>
</tbody>
</table>

Table 1: Questions by question type

3.4 Balancing the Dataset

Due to the randomness in the generation process, the resulting dataset can be imbalanced in several ways. Some questions like How many strongly connected components are in the graph? are based on features we do not directly control for and will have a different amount of instances per distinct answer. Binary questions have only two answer types while questions that can be answered with a node label have many distinct answers with few instances each.

In order to balance the dataset, we sub-sample the questions and answers in several ways:

1. For questions with a relatively small number of distinct answers (i.e. questions which are not asking for a node label), we subsample the number of instances of each distinct answer

\textsuperscript{2}https://networkx.org/
to match the one with the least instances. In a second step we subsample the number of instances of each question to the question with the least instances.

2. For questions with many distinct answers (i.e. questions which are answered with a node label), subsample distinct answers until the number of instances matches the question with the least number of instances.

After balancing the dataset, we generate negative answer (i.e. wrong) candidates for multiple-choice question answering. Depending on the question type, we use one of two strategies to sample difficult to answer candidates.

- For questions where the answer is a node label, pick up to n-1 node labels from the same graph.
- For all other questions, sample up to n-1 answers from the space of all answers for that question in the dataset.

Using this strategy, we create a benchmark dataset of 5,964,647 questions and 992,057 images for training, 610,309 questions and 99,284 images for validation and 99,139 images for testing. It contains directed and undirected graphs with 8 to 16 nodes and 12 to 24 edges. Node styles are either solid rectangles or two or three randomly selected different node styles. Node labels contain one to three words sampled randomly from the vocabulary. Edges are either solid lines or randomly drawn from two different node styles. Edge labels can be empty, numeric or textual in which case they are represented by a single word drawn from the vocabulary.

The number of generated images is evenly distributed across all parameters and the vocabularies of the train, val and test splits are disjunct. We generate up to four negative answers for each question. An example of an image with QA annotations can be seen in Figure 1.

3.5 Additional real-world test set

In order to test our dataset and models further, we also create and provide a small test set from real-world flowcharts. We generate questions and answers from the task descriptions and node labels using the method from Reddy et al. [2021] and Shakeri et al. [2020]. A full description of the dataset creation process and experimental results on this dataset are included in the supplementary material.

4 Baseline methods

We implement three models using different input modalities to establish baseline performance for the visual QA task on our dataset.

4.1 Text-only baseline

For our first baseline we fine-tune a transformer network which only uses the question and answer candidates as inputs. Each answer candidate is concatenated separately with the question and encoded by our model for which we use the Bert [Devlin et al. [2019] model architecture. After encoding, we obtain a probability distribution over the answer candidates using a linear layer. We include this model to disclose to what extent answering the questions truly requires seeing the flowchart; we use it as a sanity check for biases in the questions and answers of the dataset.

4.2 Image-based baseline

The image-based baseline (Figure 2) uses two input modalities, namely the image and the question and answer candidates. Each image is rescaled to 224x224 pixels and a visual embedding is extracted from a grid of 14x14 patches using the Vision Transformer [Dosovitskiy et al. [2021] model. The transformer architecture diagram is given in Figure 3. We ingest the multi-modal input by utilizing the the same text encoder model as in the text-only baseline but allow it to attend to the
Figure 2: Architecture of the image-based baseline. The multi-modal attention is described in Fig. 3.

Figure 3: Cross-attention mechanism. A multi-head cross-attention layer is added to each layer of the text classifier to allow it to attend to the features of the visual encoder. The figure depicts the integration into a single layer of the textual encoder.

image features to answer the question. For that purpose, we add a multi-head dot-product attention layer (Figure 3) after each self-attention layer in the text classification model.

4.3 Graph-based baseline

Our graph-based baseline has access to the underlying graph as well as the question and answer candidates. We use another transformer model, accepting graph nodes and edges converted into tokens, to represent the graph structure and combine it with the text input using the same text transformer model with cross attention (Figure 3) as in the image-based baseline. Each node is represented by its label and the labels of the nodes that can be reached from it as context. To allow the model to learn spatial features, we add sinusoidal coordinate embeddings representing the position and size of each node [Su et al., 2020]. The final graph representation is obtained from the [CLS] embedding of each node (please refer to the Devlin et al. [2019] for details).

4.4 Implementation Details

We use the huggingface transformers library [Wolf et al., 2020] for implementations of the transformer models. The textual encoder models are initialized with pre-trained Bert weights[3] and the visual encoder with pre-trained Vision Transformer weights[4]. We train all of our baseline systems on the training split for up to three epochs and check performance on a random sample of ten percent of the

validation split five times per epoch for early stopping. Training stops early if no improvement is observed in the last three validation runs. Each model was trained with cross entropy loss and Adam optimizer with a learning rate of $10^{-5}$ and a batch size of 256 on an NVIDIA RTX A6000 GPU.

5 Results

The results on the best model configurations can be seen in Table 2 and detailed results for individual questions by question type in Figure 4, Figure 5, and Figure 6, where numbers on the horizontal axes refer to the questions in the geometric category in Table 1.

<table>
<thead>
<tr>
<th>Question type</th>
<th>Model (Accuracy)</th>
<th>Random</th>
<th>Text-only</th>
<th>Image-based</th>
<th>Graph-based</th>
</tr>
</thead>
<tbody>
<tr>
<td>geometric</td>
<td></td>
<td>26.38</td>
<td>29.17</td>
<td>63.05</td>
<td>86.37</td>
</tr>
<tr>
<td>topological</td>
<td></td>
<td>31.27</td>
<td>32.61</td>
<td>75.65</td>
<td>72.45</td>
</tr>
<tr>
<td>semantic</td>
<td></td>
<td>40.33</td>
<td>44.33</td>
<td>75.81</td>
<td>79.51</td>
</tr>
<tr>
<td>overall</td>
<td></td>
<td>32.82</td>
<td>34.96</td>
<td>72.89</td>
<td>77.42</td>
</tr>
</tbody>
</table>

Table 2: Results of the baseline systems by question type

Figure 4: Accuracy of the best performing models on the geometric questions.

Figure 5: Accuracy of the best performing models on the topological questions.

We attempt to visualize the distribution of question specific attention on the image by aggregating the cross-attention weights and projecting them back onto the image. To do so, we average cross-attention weights across all heads of each layer and multiply the averaged attention weights of all layers. Lastly, we take the attention weights for the [CLS] token and normalize the distribution before projecting the weights back on the original image. An example for visualizations on different questions can be found in Figure 7. Additional cross-attention visualizations can be found in the supplementary material.
6 Limitations

One major limitation of the dataset in its current form is the lack of meaning in the flowcharts which are randomly generated with labels drawn from a large vocabulary. This also entails that the semantic question category currently is underdeveloped and will only become more meaningful with additional work towards domain adaptation.

Large, neural machine learning models in computer vision pre-trained for image feature extraction are generally trained on natural images. We expect this dataset to be helpful as pre-training task for downstream tasks on flowcharts but we did not test it as such.

7 Conclusions

We have proposed a first QA benchmark for reasoning over flowcharts, comprises close to 1M synthetic flowchart images and 6M question/answer pairs. It is automatically balanced to avoid various biases that would boost the random guess performance. Also, we have provided a collection of baselines probing the different aspects of the dataset (via text-only, image-based, and graph-based analysis), including quantitative and qualitative results. Via the baselines, we have found that the visual information can be used to answer the questions, significantly outperforming the random baseline. Also, underlying graph and the spatial information are shown to facilitate answers to semantic and the geometric questions.

While making a significant gap from random-guess performance, reported results on our baselines (involving some of the latest computer vision tools) indicate that the flowcharts QA task on Flowchart-sQA is yet far from being solved. This poses an interesting challenge to be explored further by the computer vision community.

Additional future work directions can include: addressing additional tasks, like extraction of flowchart components, introducing specialization to specific domain (e.g. biology, chemistry, law, etc.) and extending the tasks and analysis to few-shot or zero-shot (completely unseen) question types.
References


Samira Ebrahimi Kahou, Vincent Michalski, Adam Atkinson, Akos Kadar, Adam Trischler, and Yoshua Bengio. FigureQA: An Annotated Figure Dataset for Visual Reasoning, 2018. URL http://arxiv.org/abs/1710.07300


Xiaoyi Liu, Diego Klabjan, and Patrick N. Bless. Data extraction from charts via single deep neural network. arXiv, 2019. ISSN 23318422.


Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 6
   (c) Did you discuss any potential negative societal impacts of your work? [No]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]

3. If you ran experiments (e.g. for benchmarks)...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See supplementary material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 4.4 for training details such as pre-trained parameters and hyperparameters. The training data is detailed at the end of Section 3.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] We did not run the experiments multiple times due to compute constraints.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 4.4.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Section 4.4 as well as the footnotes in Section 3 and 4
   (b) Did you mention the license of the assets? [No] All assets used were published under licenses that allow them to be used for research. The licenses can be found under the respective links.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] The URL to the dataset introduced in this work is included in the supplementary material.
   (d) Did you discuss whether and how consent was obtained from people whose data you’re using/curating? [Yes] See supplementary material
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] See supplementary material

5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]