FlowchartQA

The First Large-Scale Benchmark for Reasoning over Flowcharts

Simon Tannert[1] Marcelo Feighelstein [2] Jasmina Bogojeska[3] Joseph Shtok[4]
Assaf Arbelle[4] **Peter Staar** [3] **Anika Schumann**[3] **Jonas Kuhn**[1] **Leonid Karlinsky** [5]

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 and qualitative analysis in order to establish a good basis for future work.

ACM Reference Format:
Simon Tannert[1] Marcelo Feighelstein [2] Jasmina Bogojeska[3] Joseph Shtok[4] , **Assaf Arbelle**[4] **Peter Staar** [3] **Anika Schumann**[3] **Jonas Kuhn**[1] **Leonid Karlinsky** [5]. 2022. FlowchartQA: The First Large-Scale Benchmark for Reasoning over Flowcharts. In *Proceedings of DI@KDD'22*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

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.

Inspired by recent advances and successes in addressing language-vision problems, we introduce FlowchartQA – a first of its kind benchmark for question answering on flowcharts.

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

DI@KDD'22, August 14–18, 2022, Washington DC, U.S.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

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 [17, 16, 15] , but was recently applied for documents with figures and diagrams. Among the first and important works is FigureQA [6], 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 [5]; 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 [10], 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 [13] recently were used in computer vision as alternatives to CNNs and have been used extensively for vision tasks such as the Vision Transformer (ViT) [4] In particular, they find applications in VQA domain: [2] use layout-aware transformers to answer questions by utilizing the scene text in the image, and [11] integrate BERT (a transformer-based language network [3]) for embedding text with convolutional models to represent images.

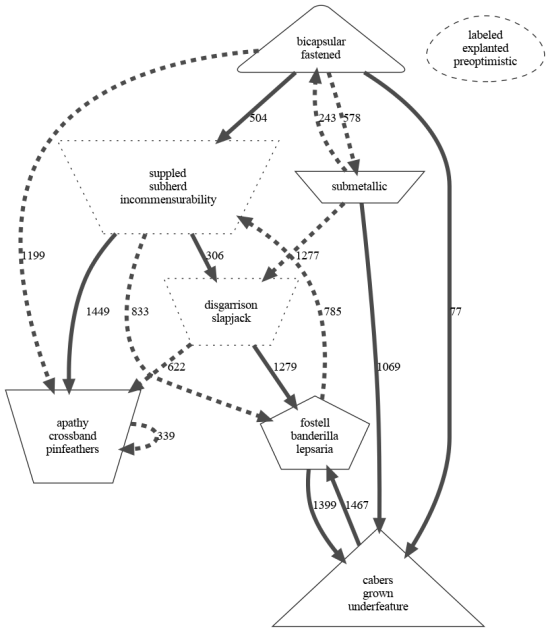
Another use of a language based model was shown in [9], where the GPT2 model [1] has been used as the decoder to facilitate image captioning tasks. This and other multi-modal architectures

integrating Transformers for combined Vision-Language tasks [12, 8, 7] 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.

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 different domains.



Question	Answer	Answer candidates
How many nodes are in the graph?	8	6, 3, 12, 8, 9
Do all nodes have the same style?	No	Yes, No
Is <submetallic> below <bicapsular\nfastened> on the image?	Yes	Yes, No

Figure 1: Example flowchart image with QA annotations

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[1].

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, questions that require a numerical answer and questions that can be answered with a node label. We categorize the questions into three categories, *geometric*, *topologic* and *semantic* based on the knowledge they require to answer them. The full list of questions can be seen in Table 1. The generated graph is loaded into networkx[2] which allows us to analyze its topology and answer the questions.

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 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 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 the same question in the dataset.

[1]https://graphviz.org/
[2]https://networkx.org/

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

Table 1: Questions by question type

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 585,179 questions and 99,139 images for testing. It contains directed and undirected graphs with 8 to 16 nodes and 12 to 24 edges. Nodes 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.

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 [3] 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

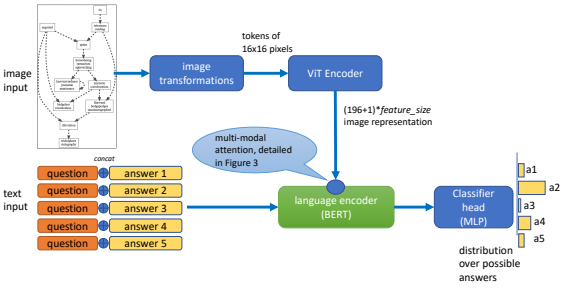


Figure 2: Architecture of the image-based baseline. The multi-modal attention is described in Figure 3

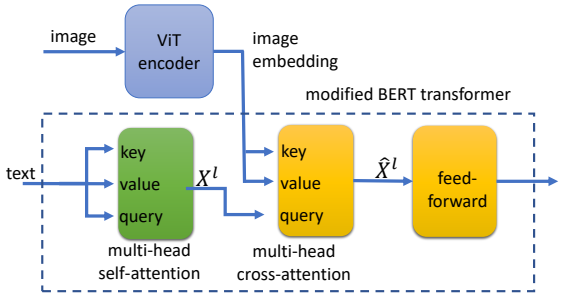


Figure 3: Cross-attention mechanism. A multi-head cross-attention layer is added to each layer of the text classifier to allow it to attend to the features of the visual encoder. The figure depicts the integration into a single layer of the textual encoder.

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 [4] 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 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. To allow the model to learn spatial features, we add sinusoidal coordinate embeddings representing the position and size of each node [12]. The final graph representation is obtained from the `[CLS]` embedding of each node (please refer to the [3] for details).

4.4 Implementation Details

We use the huggingface transformers library [14] 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.

Question type	Model (Accuracy)			
	Random	Text-only	Image-based	Graph-based
geometric	26.38	29.17	63.05	86.37
topological	31.27	32.61	75.65	72.45
semantic	40.33	44.33	75.81	79.51
overall	32.82	34.96	72.89	77.42

Table 2: Results of the baseline systems by question type

6 CONCLUSIONS

In this paper we have proposed FlowchartQA – a new, and first of its kind, large scale QA benchmark for reasoning over flowcharts. It is automatically and rigorously balanced to reduce biases that would allow significantly deviating from chance performance if attempting to answer the questions without regarding the respective flowcharts.

Our findings motivate interesting vision tasks to be explored further by the computer vision community. Despite clearly far from random performance reported for our best baselines using some of the latest computer vision tools, such as vision transformers and proposed text-vision and text-graph cross-attention scheme, our reported results indicate that the flowcharts QA task on FlowchartQA is yet far from being solved.

[3]https://huggingface.co/bert-base-uncased
[4]https://huggingface.co/google/vit-base-patch16-224-in21k

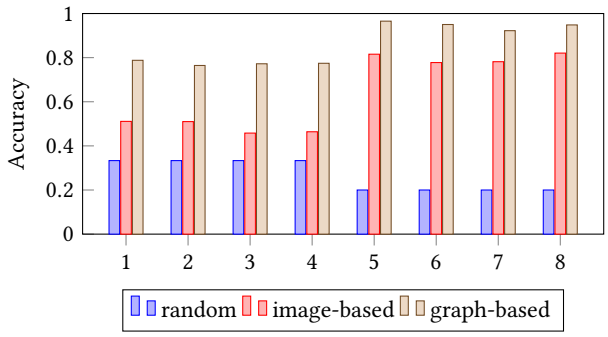


Figure 4: Accuracy of the best performing models on the geometric questions.

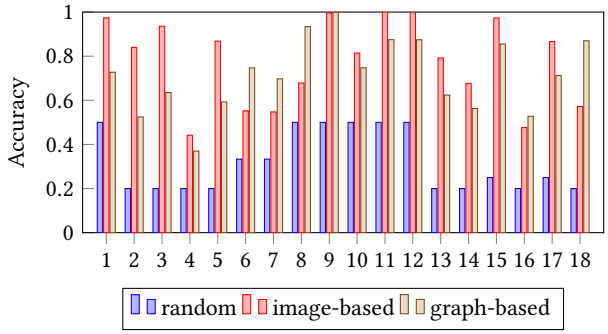


Figure 5: Accuracy of the best performing models on the topological questions.

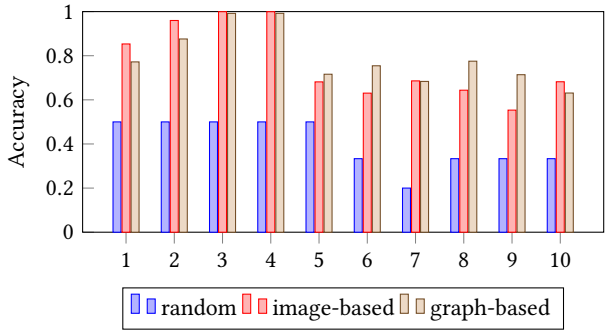


Figure 6: Accuracy of the best performing models on the semantic questions.

Additional future work directions to extend the proposed benchmark include: introducing additional tasks (e.g. flowchart components detection and segmentation), introducing domain specialization by generating chart styles and content specific to certain knowledge domains (e.g. biology, chemistry, computer science, etc.), and extending the tasks and analysis to few-shot or zero-shot (completely unseen) question types.

ACKNOWLEDGEMENTS

This work is supported by IBM Research AI through the IBM AI Horizons Network.

REFERENCES

[1] Radford Alec, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. In *OpenAI blog*.

[2] Ali Furkan Biten, Ron Litman, Yusheng Xie, Srikar Appalaraju, and R. Manmatha. 2021. Latr: layout-aware transformer for scene-text vqa. (2021). http://arxiv.org/abs/2112.12494.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. NAACL-HLT 2019. Association for Computational Linguistics, Minneapolis, Minnesota, (June 2019), 4171–4186. DOI: 10.18653/v1/N19-1423.

[4] Alexey Dosovitskiy et al. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. (June 3, 2021). arXiv: 2010.11929 [cs]. Retrieved Mar. 1, 2022 from http://arxiv.org/abs/2010.11929.

[5] Kushal Kafle, Brian Price, Scott Cohen, and Christopher Kanan. 2018. DVQA: Understanding Data Visualizations via Question Answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (June 2018).

[6] Samira Ebrahimi Kahou, Vincent Michalski, Adam Atkinson, Akos Kadar, Adam Trischler, and Yoshua Bengio. 2018. FigureQA: An Annotated Figure Dataset for Visual Reasoning. (Feb. 22, 2018). arXiv: 1710.07300 [cs]. Retrieved Mar. 9, 2021 from http://arxiv.org/abs/1710.07300.

[7] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, and Lijuan Wang et al. 2020. Oscar: object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision*, 121–137.

[8] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in neural information processing systems* number 32.

[9] Ziyang Luo, Yadong Xi, Rongsheng Zhang, and Jing Ma. 2022. Vc-gpt: visual conditioned gpt for end-to-end generative vision-and-language pre-training. In *arXiv:2201.12723*.

[10] Nitesh Methani, Pritha Ganguly, Mitesh M. Khapra, and Pratyush Kumar. 2020. PlotQA: Reasoning over Scientific Plots. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. (Mar. 2020).

[11] Tung Le; Nguyen Tien Huy; Nguyen Le Minh. 2020. Integrating transformer into global and residual image feature extractor in visual question answering for blind people. In *International Conference on Knowledge and Systems Engineering*.

[12] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. 2020. Vl-bert: pre-training of generic visual-linguistic representations. In *ICLR*.

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc. Retrieved Mar. 1, 2022 from https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[14] Thomas Wolf et al. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, Online, (Oct. 2020), 38–45. DOI: 10.18653/v1/2020.emnlp-demos.6.

[15] Zhou Yu, Yuhao Cui, Jun Yu, Meng Wang, Dacheng Tao, and Qi Tian. 2020. Deep Multimodal Neural Architecture Search. In *Proceedings of the 28th ACM International Conference on Multimedia*. Association for Computing Machinery, New York, NY, USA, (Oct. 12, 2020), 3743–3752. ISBN: 978-1-4503-7988-5. Retrieved Mar. 1, 2022 from https://doi.org/10.1145/3394171.3413977.

[16] Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. 2019. Deep Modular Co-Attention Networks for Visual Question Answering. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 6281–6290. Retrieved Mar. 1, 2022 from https://openaccess.thecvf.com/content%5C_CVPR%5C_2019/html/Yu%5C_Deep%5C_Modular%5C_Co-Attention%5C_Networks%5C_for%5C_Visual%5C_Question%5C_Answering%5C_CVPR%5C_2019%5C_paper.html.

[17] Zhou Yu, Jun Yu, Jianping Fan, and Dacheng Tao. 2017. Multi-modal Factorized Bilinear Pooling with Co-attention Learning for Visual Question Answering. In *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017 IEEE International Conference on Computer Vision (ICCV). IEEE, Venice, (Oct. 2017), 1839–1848. ISBN: 978-1-5386-1032-9. DOI: 10.1109/ICCV.2017.202.