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

Simon Tannert
IMS, Stuttgart
tannersn@ims.uni-stuttgart.de

Marcelo Feighelstein
Technion, Haifa

Jasmina Bogojeska
ZHAW, Zurich

Joseph Shtok
IBM Research, Haifa
josephs@il.ibm.com

Assaf Arbelle
IBM Research, Haifa

Peter W. J. Staar
IBM Research, Zurich

Anika Schumann
IBM Research, Zurich

Jonas Kuhn
IMS, Stuttgart

Leonid Karlinsky
MIT-IBM Research Lab, Cambridge

Abstract

In this paper, we present FlowchartQA, a new and unique large-scale benchmark for visual question answering (VQA) over flowcharts. FlowchartQA comprises close to 1M flowchart images and 6M question-answer pairs, covering various aspects of geometric and topological information contained in the charts. The questions have been carefully balanced to minimize biases. To accompany the proposed benchmark, we present a baseline model and perform comprehensive ablation studies and qualitative analyses to provide a solid foundation for future work. Our experimental results reveal interesting findings and demonstrate the potential of FlowchartQA as a testbed for flowchart understanding, which has been previously absent in the community.

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 and 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 / datset offers any large scale data for training / evaluating flowchart understanding models.

Therefore, inspired by recent advances and successes in addressing vision-language 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 two types of questions over flowcharts, geometric and topological. The code 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. To tackle this problem, we present a baseline model that leverages advanced neural architectures, such as transformers and attention mechanisms. The model is designed to integrate both textual and visual modalities of the input data. The effectiveness of the proposed model is demonstrated through evaluation and ablation experiments.

Proceedings of the 1st Workshop on Linguistic Insights from and for Multimodal Language Processing, pages 34–46
September 22, 2023. ©2023 Association for Computational Linguistics

The man 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 questions that can potentially be adapted to generate data relevant for a specific target task.

3. A neural baseline approach for the multiple choice visual QA task over flowcharts: based on text transformers and a combination of text and visual transformers.

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 Minh (2020) integrate BERT (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 baseline.

2.3 Charts Analysis and QA

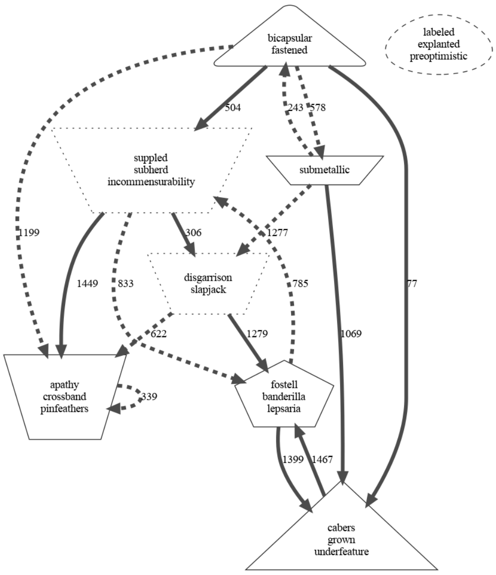
Related to QA on flowcharts is the task of regular chart analysis. Early works addressing automatic chart classification and data extraction (Savva et al., 2011; Al-Zaidy and Giles, 2015), used classical computer vision techniques, such as codebooks obtained by clustering normalized image patches, connected 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 and 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 lines of work (Liu et al., 2019; Zhou et al., 2021) propose 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 the baseline model 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.



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]. 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 that cover a large number of graph properties at different scales as well as node properties and the relations between them. These include 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`

[1] https://graphviz.org/

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

Table 1: Questions by question type

`maximum degree in the graph?`). We categorize the questions into two categories, *geometric* and *topological*, based on the knowledge required 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.

3.4 Balancing the Dataset

Due to 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

[2] https://networkx.org/

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 (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 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.

3.5 Real-World Test Set

In order to test our dataset and model further, we also create and provide a small test set from real-world flowcharts. We use a collection of Business Process Model and Notation (BPMN) diagrams[3] which contains user generated diagrams for four different tasks. The data for each task comprises a description of the process to be modeled, multiple diagrams created by users as well as a reference solution.

We generate questions and answers from the task descriptions and node labels using the method from Shakeri et al. (2020). Following the idea in Reddy et al. (2021), we fine-tune BART (Lewis et al., 2020) on the Natural Questions dataset (Kwiatkowski et al., 2019) and extract entities found in the node labels in order to be able to generate a question and answer given a task description and node label as generation cue.

All generated question-answer pairs were manually checked and instances that contain spelling mistakes or syntax errors were removed. The remaining questions and answers were subsampled to reduce the number of duplicates. Using this method, we collect a total of 266 questions over 166 images which we use to evaluate the model we fine-tuned on our synthetic dataset. Unlike the geometric and topological questions in the synthetic dataset, the questions generated from the task descriptions require understanding of the semantics of the flowchart. An example from our real-world test set can be seen in Section D.

4 Baseline Method

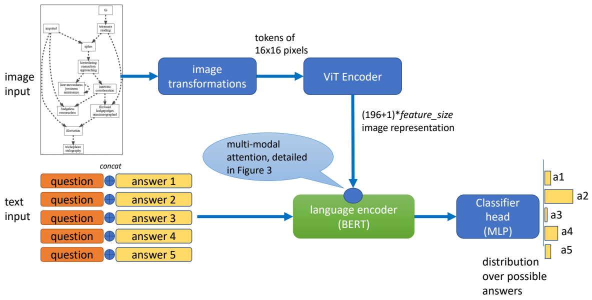


Figure 2: Architecture of our multi-modal baseline. The cross-attention is described in Fig. 3

We fine-tune a multi-modal transformer neural network for multiple choice question answering (cf. Figure 2) to establish baseline performance on our

[3] https://github.com/camunda/bpmn-for-research/

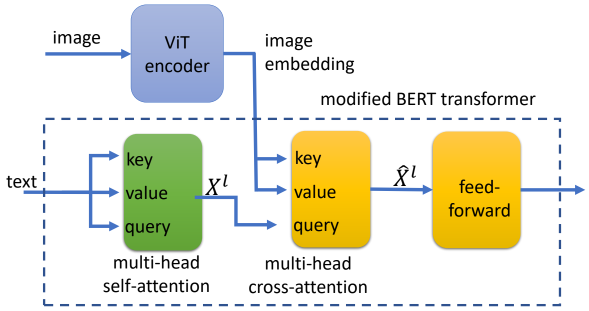


Figure 3: Cross-attention mechanism. A multi-head cross-attention layer is added to each layer of the text classifier to allow it to attend to the features of the visual encoder. The figure depicts the integration into a single layer of the textual encoder.

datasets. Each answer candidate is concatenated with the question and separately encoded by our model using Bert (Devlin et al., 2019). Visual features are extracted from the flowchart image using the Vision Transformer (ViT) (Dosovitskiy et al., 2021) which BERT can attend to during encoding using cross-attention (cf. Figure 3). After encoding, we obtain a probability distribution over the answer candidates using a linear layer.

We also test a variant of this model which does not have access to the flowchart images to test for biases in the questions answer which we refer to as text-only in the results.

4.1 Implementation Details

We use the huggingface library (Wolf et al., 2020) for implementations of the transformer model. The textual encoder model is initialized with pre-trained Bert weights[4] and the visual encoder with pre-trained Vision Transformer weights[5] Each image is rescaled is 224x224 pixels and visual features are extracted from a grid of 14x14 patches. We train our baseline system 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 a single NVIDIA RTX A6000 GPU.

5 Results

5.1 QA on Synthetic Dataset

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 and Figure 5, where numbers on the horizontal axes refer to the questions in the geometric category in Table 1.

Question type	Model (Accuracy)		
	Random	Text-only	Multi-modal
geometric	30.91	33.19	71.65
topological	33.22	35.63	74.87
overall	32.58	34.96	73.98

Table 2: Results of the baseline systems by question type

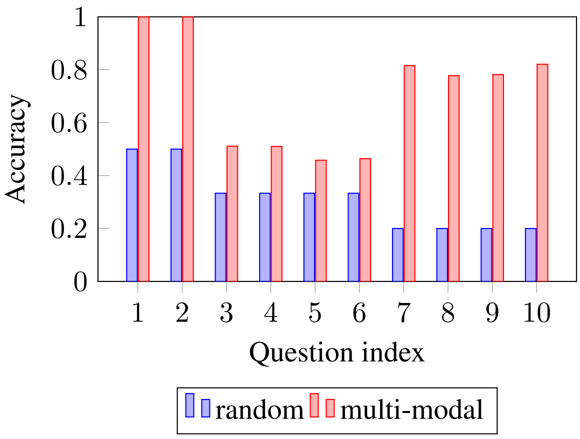


Figure 4: Accuracy of the best performing multi-modal model on the geometric questions.

Figure 6 shows the results of our best-performing multi-modal model in terms of different graph properties. Looking at model performance based on node or edge count, shows that accuracy decreases as the node or edge count increases and the flowcharts become more complex. The effect is more pronounced for the node counts because there are more questions about nodes than there are about edges. The same effect can be observed for the edge label type which does not have a large influence on model performance.

The biggest influence on performance can be observed for the diameter of graphs which drops off significantly for higher diameters as they require better understanding of the graph topology and reasoning capabilities.

[4] https://huggingface.co/bert-base-uncased

[5] https://huggingface.co/google/vit-base-patch16-224-in21k

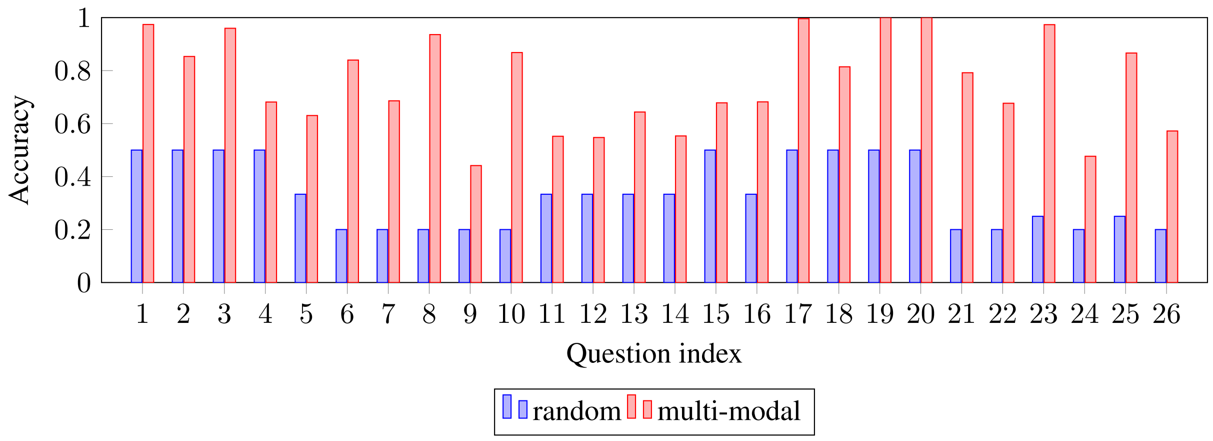


Figure 5: Accuracy of the best performing multi-modal model on the topological questions.

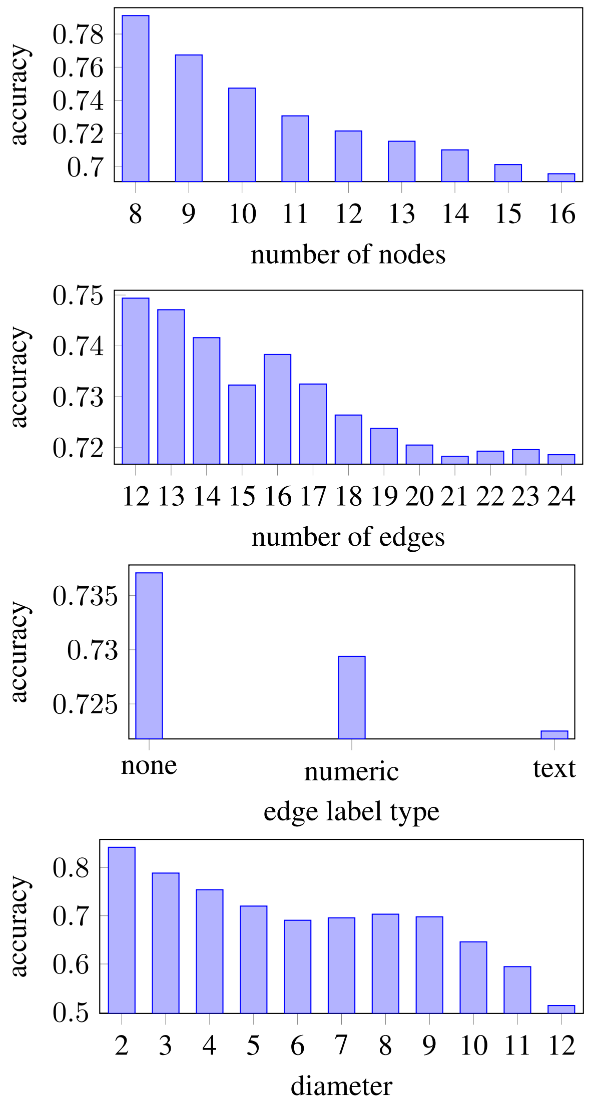


Figure 6: Accuracy of the best-performing multi-modal model on different subsets of the test set. Based on the number of nodes in the flowchart graphs, the number of edges, number of connected components and the diameter of graph with a single connected component.

Dataset	Random	Text-only	Multi-modal	
			frozen	unfrozen
FlowchartQA	32.82	34.96	57.98	73.98
real-world	20.00	21.05	20.68	26.35

Table 3: Results of our model fine-tuned on FlowchartQA, evaluated on the test split of FlowchartQA and our test set of real-world BPMN diagrams (Accuracy).

Question type	Frozen layers (Accuracy)			
	both	visual	textual	unfrozen
geometric	47.14	49.45	57.03	71.65
topological	62.10	62.94	69.21	74.87
overall	57.98	59.22	65.85	73.98

Table 4: Multi-modal model ablation study

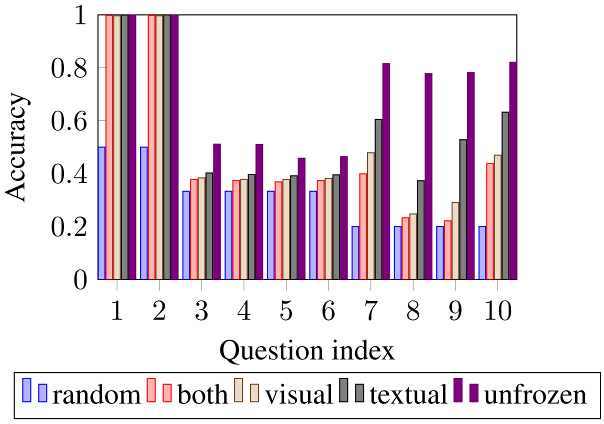


Figure 7: Accuracy of the multi-modal model on the geometric questions with different parts of the model frozen during fine-tuning.

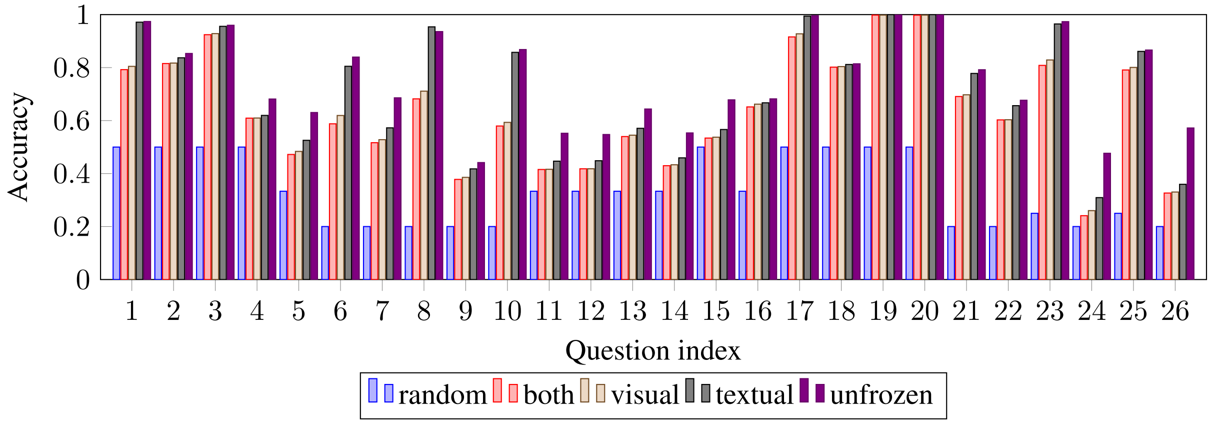


Figure 8: Accuracy of the multi-modal model on the topological questions with different parts of the model frozen during fine-tuning.

5.2 QA on Real-World Test Set

We also evaluate our model on the real-world test set without additional fine-tuning due to the small size of the dataset. The accuracy The results for the real-world dataset in Table 3 show that the model that was fine-tuned with the visual encoder unfrozen leads to an improvement over the random baseline and the text-only model. The model with both visual and textual encoder unfrozen during fine-tuning shows the largest improvement, indicating that it was able to learn generalizable knowledge that transfers to the semantic questions and different visual style of the real-world test set.

6 Ablation Study

We test the influence of keeping different layers of our joint networks frozen during fine-tuning. In the multi-modal baseline, the visual encoder is initialized with pre-trained ViT weights and the text encoder is initialized with Bert weights (cf. Section 4.1). We test the performance of the model while keeping either the visual encoder, the textual encoder or both frozen during fine-tuning. Note that cross-attention and output layers are being trained in all settings because they are not initialized from pre-trained weights.

The results for the multi-modal baseline in Table 4 show that the pre-trained models already exhibit strong baseline performance even when both visual and textual encoder are kept frozen. Fine-tuning the textual encoder only yields a minor improvement in all categories while fine-tuning the visual encoder leads to a stronger improvement over the random baseline. The pre-trained ViT model of the visual encoder was trained on ImageNet-21k (Ridnik et al., 2021) which consists of images depicting natural scenes and seems to benefit from fine-tuning on our graph images. The best performance is observed with both the visual and textual encoder are fine-tuned. This is most notable in the geometric question type which requires spatial reasoning with multiple nodes. Figure 8 breaks down the performance over the different geometric questions. Questions 5.-8. (cf. Table 1) require identifying the top-, bottom-, left- or rightmost node, which benefit noticeably from fine-tuning of the visual encoder. Questions 1.-4. are binary but require identifying and reasoning over two nodes which makes them conceptually more difficult.

7 Conclusions

In conclusion, this paper presents a new benchmark for visual QA over flowcharts, which includes close to 1M synthetic flowchart images and 6M question-answer pairs. The benchmark has been carefully balanced to mitigate biases that could enhance random guess performance. We also provide a baseline model to evaluate the benchmark and demonstrate its performance through both quantitative and qualitative results.

However, the results obtained from the baseline model, which utilizes state-of-the-art computer vision tools, suggest that the QA task on FlowchartQA remains a challenging problem. This presents an interesting opportunity for further exploration by the computer vision community.

Future work directions may include addressing additional tasks, such as the extraction of flowchart components, domain adaption (e.g., biology, chemistry, law, etc.), and extending the tasks and analysis to few-shot or zero-shot question types.

References

Rabah A. Al-Zaidy and C. Lee Giles. Automatic extraction of data from bar charts. *Proceedings of the 8th International Conference on Knowledge Capture, K-CAP 2015*, 2015. doi: 10.1145/2815833.2816956.

Rabah A. Al-Zaidy, Sagnik Ray Choudhury, and C. Lee Giles. Automatic summary generation for scientific data charts. *AAAI Workshop - Technical Report*, WS-16-01 -:658–663, 2016.

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

Ali Furkan Biten, Ron Litman, Yusheng Xie, Srikar Appalaraju, and R. Manmatha. Latr: Layout-aware transformer for scene-text vqa., 2021. URL http://arxiv.org/abs/2112.12494.

Mathieu Cliche, David Rosenberg, Dhruv Madeka, and Connie Yee. Scatteract: Automated Extraction of Data from Scatter Plots. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10534 LNAI(1):135–150, 2017. ISSN 16113349. doi: 10.1007/978-3-319-71249-9{_}9.

Wenjing Dai, Meng Wang, Zhibin Niu, and Jiawan Zhang. Chart decoder: Generating textual and numeric information from chart images automatically. *Journal of Visual Languages and Computing*, 48:101–109, 10 2018. ISSN 1045926X. doi: 10.1016/j.jvlc.2018.08.005.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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)*, pages 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, 2021. URL http://arxiv.org/abs/2010.11929.

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

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.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, July 2020. URL https://aclanthology.org/2020.acl-main.703.

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

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

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

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

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

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

Jorge Poco and Jeffrey Heer. Reverse-Engineering Visualizations: Recovering Visual Encodings from Chart Images. *Computer Graphics Forum*, 36(3):353–363, 2017. ISSN 14678659. doi: 10.1111/cgf.13193.

Revanth Gangi Reddy, Xilin Rui, Manling Li, Xudong Lin, Haoyang Wen, Jaemin Cho, Lifu Huang, Mohit Bansal, Avirup Sil, Shih-Fu Chang, Alexander Schwing, and Heng Ji. Mumuqa: Multimedia multi-hop news question answering via cross-media knowledge extraction and grounding, 2021. URL https://arxiv.org/abs/2112.10728.

Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik. ImageNet-21K Pretraining for the Masses. 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/98f13708210194c475687be6106a3b84-Abstract-round1.html.

Manolis Savva, Nicholas Kong, Arti Chhajta, Fei Fei Li, Maneesh Agrawala, and Jeffrey Heer. ReVision: Automated classification, analysis and redesign of chart images. *UIST'11 - Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, (April):393–402, 2011. doi: 10.1145/2047196.2047247.

Siamak Shakeri, Cicero Nogueira dos Santos, Henghui Zhu, Patrick Ng, Feng Nan, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. End-to-end synthetic data generation for domain adaptation of question answering systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5445–5460, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.439. URL https://aclanthology.org/2020.emnlp-main.439.

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

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

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6.

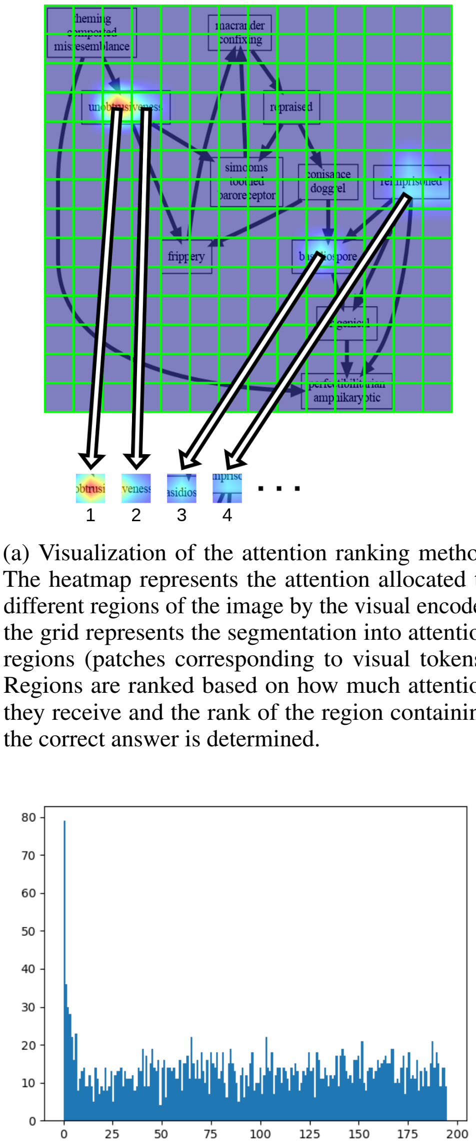
Zhou Yu, Jun Yu, Jianping Fan, and Dacheng Tao. Multi-modal Factorized Bilinear Pooling with Co-attention Learning for Visual Question Answering. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1839–1848. IEEE, 2017. ISBN 978-1-5386-1032-9. doi: 10.1109/ICCV.2017.202. URL http://ieeexplore.ieee.org/document/8237464/.

Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. Deep Modular Co-Attention Networks for Visual Question Answering. pages 6281–6290, 2019. URL https://openaccess.thecvf.com/content_CVPR_2019/html/Yu_Deep_Modular_Co-Attention_Networks_for_Visual_Question_Answering_CVPR_2019_paper.html.

Zhou Yu, Yuhao Cui, Jun Yu, Meng Wang, Dacheng Tao, and Qi Tian. Deep Multimodal Neural Architecture Search. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 3743–3752. Association for Computing Machinery, 2020. ISBN 978-1-4503-7988-5. URL https://doi.org/10.1145/3394171.3413977.

Fangfang Zhou, Yong Zhao, Wenjiang Chen, Yijing Tan, Yaqi Xu, Yi Chen, Chao Liu, and Ying Zhao. Reverse-engineering bar charts using neural networks. *Journal of Visualization*, 24(2):419–435, 2021. ISSN 18758975. doi: 10.1007/s12650-020-00702-6. URL https://doi.org/10.1007/s12650-020-00702-6.

A Appendix



(a) Visualization of the attention ranking method. The heatmap represents the attention allocated to different regions of the image by the visual encoder, the grid represents the segmentation into attention regions (patches corresponding to visual tokens). Regions are ranked based on how much attention they receive and the rank of the region containing the correct answer is determined.

(b) Distribution of the rank of the image region containing the node that correctly answers the question `"What is the node with the maximum degree in the graph?"`

Figure 9: Visual attention analysis

B Quantitative Attention Analysis

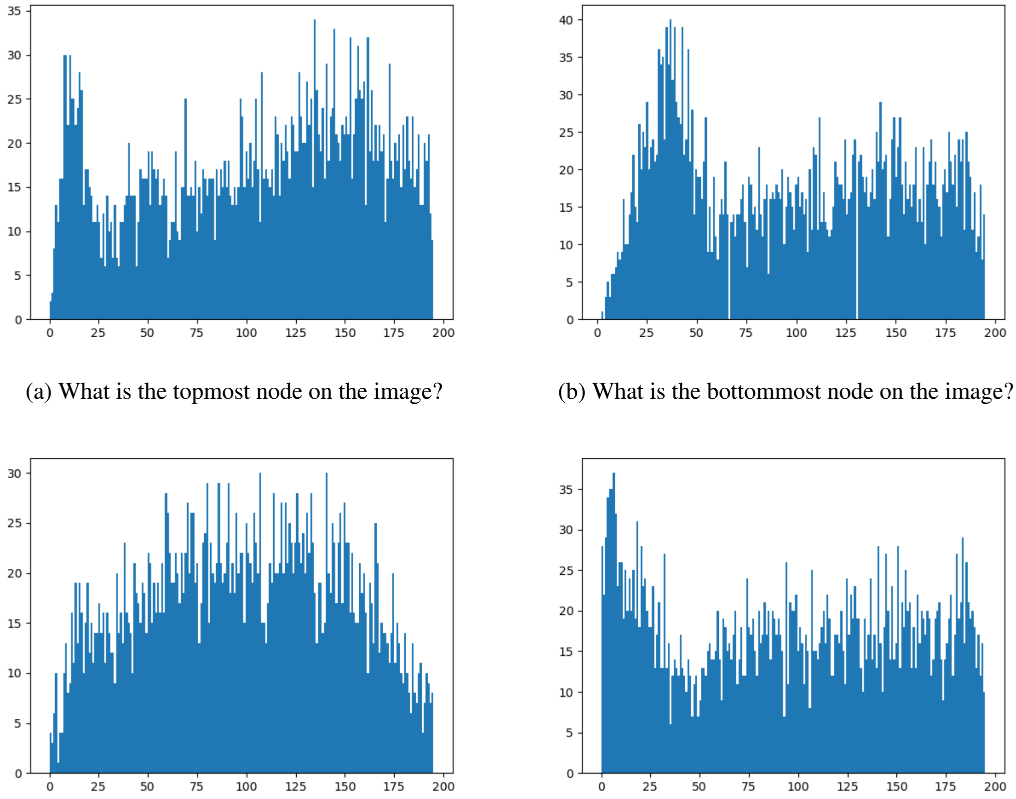
For questions that are answered with a node label, we analyze how much attention the region that contains the correct node receives. We rank the regions of the image by how much attention they receive for all instances where they have been answered correctly by the unfrozen multi-modal baseline model. We then determine how much attention the node that answers the question correctly receives by determining the rank of the region that contains the center of the respective node. The rank distribution for `"What is the node with the maximum degree in the graph?"` can be seen in Figure 9b, the rank distributions for questions: `"What is the {topmost, bottommost, leftmost, rightmost} node on the image?"` is shown in Figure 10. In all these figures, ideally, we would like the correct answer to receive the smallest rank possible (aka top rank). The maximal possible rank corresponds to the number of image tokens, $14 \times 14 = 196$ in our case. It is also worth mentioning that model's attention can have multiple uses, sometimes a model can devote a higher attention to a certain region for inhibitory purposes, in other words - to rule out certain options.

The distribution in Figure 9b shows a peak for the top ranks of the attention distribution, indicating that in many instances, the cross-attention allocates most attention on the region that contains the node that answers the question.

The distributions in Figure 10 show that with the exception of `"What is the leftmost node on the image"`, there are also clear peaks near the top ranks of the attention distribution. When we compare this to the relative performance of the models that have layers frozen during fine-tuning (cf. Figure 8), we can see that the model that was fine-tuned with the visual encoder unfrozen yields lower accuracy on `"What is the leftmost node on the image"` (questions number 6) than the other three questions (5, 7 and 8). When we fine-tune with all parts of the model unfrozen the performance degradation vanishes and the other parts of the network make up for a weaker representation in the visual encoder but the effect can still be seen in the visual attention.

C Visual Attention Heatmaps

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 11.



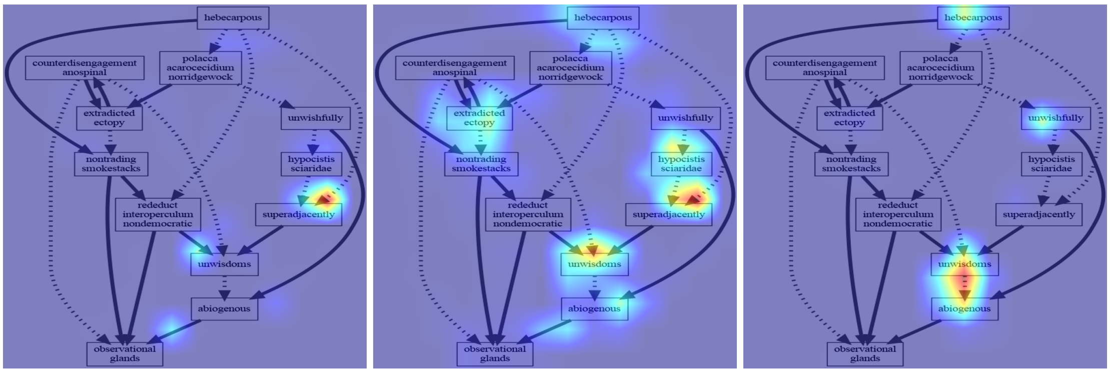
(a) What is the topmost node on the image?

(b) What is the bottommost node on the image?

(c) What is the leftmost node on the image?

(d) What is the rightmost node on the image?

Figure 10: Distribution of the rank of the image region containing the node that answers the question `"What is the {topmost, bottommost, leftmost, rightmost} node on the image?"`



(a) Q: Is this an undirected graph? A: No

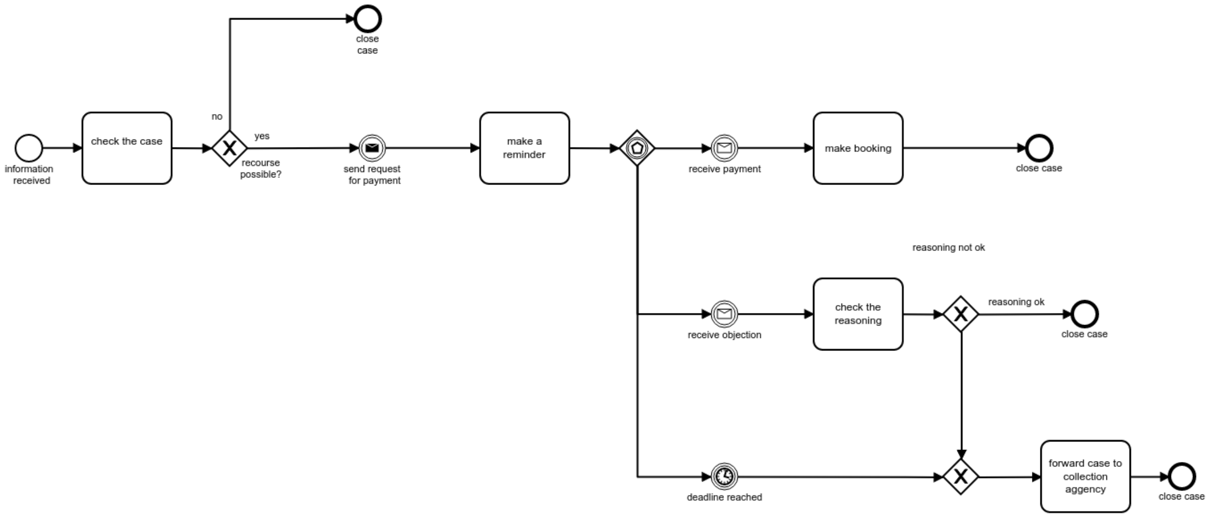
(b) Q: Is there a node directly connected to itself? A: No

(c) Q: Is <abiogenous> a direct successor of <unwisdoms>? A: Yes

Figure 11: Cross-attention visualization of the multi-modal model.

D Real-World Test Set Example

Figure 12 shows one of the figures from the camunda BPMN dataset that we used to generate the real-world test set as well as the inputs and outputs of our QA generation model. We use the instruction text and a randomly selected node label to generate a question and answer using the model described in Section 3.5. Four negative answer choices are sampled from the node labels of the same diagram.



`instructions` exercise 3

Recourse

Please model the following process:

If an insurant could be possibly subrogated against, I get information about that. I check that case and if the possibility is really there, I send a request for payment to the insurant and make me a reminder. If recourse is not possible, I close the case.

When we receive the money, I make a booking and close the case. If the insurant disagrees with the recourse, I'll have to check the reasoning of that. If he is right, I simply close the case. If he is wrong, I forward the case to a collection agency.

It the deadline for disagreement is reached and we haven't received any money, I forward the case to the collection agency as well.

Background information:

Insurants can be forced to pay back money they received from the insurance company for different reasons. This is called recourse. Here the clerk describes how this process works.

`node label` check the reasoning

`question` when do i check the reasoning of a case?

`answers`
1. if the deadline for disagreement is reached
2. **if the insurant disagrees with the recourse**
3. if recourse is not possible or money is received
4. if we receive the money
5. if the insurant could be possibly subrogated against

Figure 12: Example BPMN diagram from https://github.com/camunda/bpmn-for-research used in the real-world test set together with the instructions provided with the dataset. The extracted node label was used together with the instructions to generate a question and answer. The negative answers were selected from the remaining node labels of the same figure. The correct answer is higlighted in bold.