# MSG-Chart: Multimodal Scene Graph for ChartQA

Yue Dai
yue.dai@research.uwa.edu.au
The University of Western Australia
Perth, Western Australia, Australia

Soyeon Caren Han*
The University of Melbourne
Melbourne, Australia
caren.han@unimelb.edu.au

Wei Liu
The University of Western Australia
Perth, Australia
wei.liu@uwa.edu.au

## Abstract

Automatic Chart Question Answering (ChartQA) is challenging due to the complex distribution of chart elements with patterns of the underlying data not explicitly displayed in charts. To address this challenge, we design a joint multimodal scene graph for charts to explicitly represent the relationships between chart elements and their patterns. Our proposed multimodal scene graph includes a visual graph and a textual graph to jointly capture the structural and semantical knowledge from the chart. This graph module can be easily integrated with different vision transformers as inductive bias. Our experiments demonstrate that incorporating the proposed graph module enhances the understanding of charts' elements' structure and semantics, thereby improving performance on publicly available benchmarks, ChartQA and OpenCQA.[1]

## CCS Concepts

• **Computing methodologies** → Visual content-based indexing and retrieval; *Information extraction.*

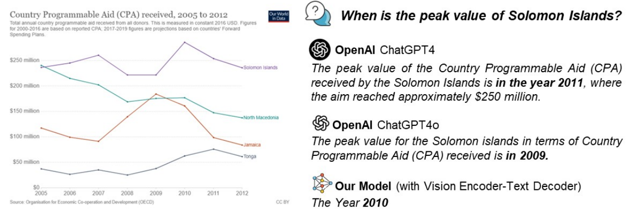
## Keywords

Chart Question Answering, Scene Graph, Multimodal Learning

**ACM Reference Format:**
Yue Dai, Soyeon Caren Han, and Wei Liu. 2024. MSG-Chart: Multimodal Scene Graph for ChartQA. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24), October 21–25, 2024, Boise, ID, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3627673.3679967

## 1 Introduction

Chart Question Answering (ChartQA) involves answering questions presented in natural languages based on information from a chart. As a subset of Visual Question Answering (VQA), ChartQA presents unique challenges distinct from those encountered with natural images. Firstly, a chart contains numerous elements and various element types within a single image. Although the distribution of these elements can be complex, charts are generally highly structured. For instance, y axis labels are normally located at the left side of the image while x axis labels are at the bottom. Secondly, charts often include extensive text and numerical data, understanding these features is crucial for accurate question answering. While recognizing the underlying text of an object is enough for data extraction tasks, more complex reasoning requires understanding the relationships between elements. For example, in Fig. 1, the model must recognise that 'Solomon Islands' is not just a label but specifically the label for the purple line. This highlights the importance of comprehending the structure and the relationships within charts.



***When is the peak value of Solomon Islands?***

**OpenAI** ChatGPT4
*The peak value of the Country Programmable Aid (CPA) received by the Solomon Islands is **in the year 2011**, where the aim reached approximately $250 million.*

**OpenAI** ChatGPT4o
*The peak value for the Solomon islands in terms of Country Programmable Aid (CPA) received is **in 2009.***

**Our Model** (with Vision Encoder-Text Decoder)
*The Year **2010***

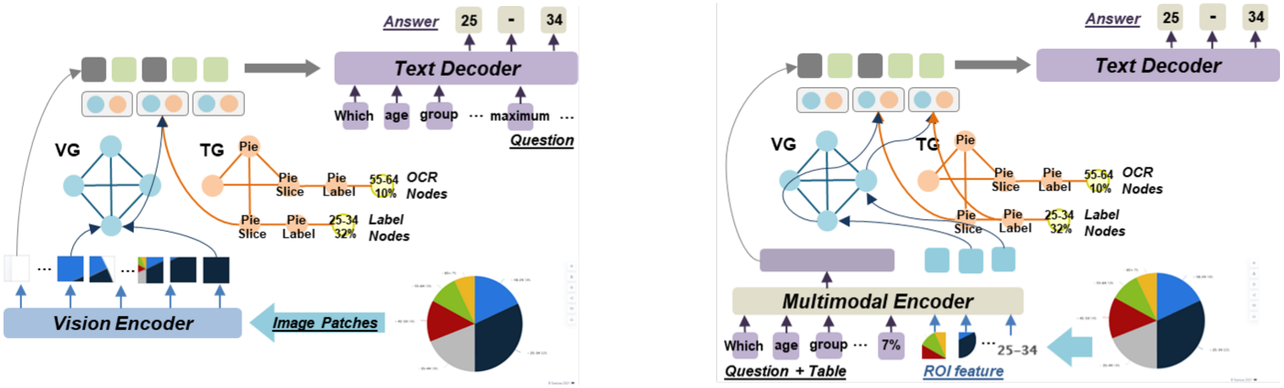
**Figure 1: Cutting-Edge LLMs and Our MSG-Chart**

Although various models have been proposed to address ChartQA, they have not effectively tackled the challenges above. [12] introduced DePlot for chart-to-table conversion. The converted textual data table from the chart is then fed to Flan-PaLM [2] for question answering. Despite the strong mathematical reasoning capabilities of Large Language Models (LLMs), this approach struggles with questions related to visual attributes, such as colour, due to the lack of visual information understanding in textual tables. Masry et al. [15] and Tang et al. [22] employ VL-T5 [1] for ChartQA and Chart Captioning, utilizing Mask R-CNN [5] or Faster R-CNN [19] to extract Region of Interest (ROI) features of objects, combined with textual data tables or scene graphs as input. However, these methods lack explicit spatial relationships between ROI features, and the connections between visual features and their underlying value remain unclear. More recently, pre-trained multimodal language models [13, 14] have been proposed. Their patch-based image inputs can result in chart elements being split across different patches, losing object-wise information and making it challenging for the model to capture the full pattern of an intact object. This limitation is also present in the latest LLMs, such as GPT-4 and GPT-4o, as demonstrated in Fig. 1. While these models generate detailed answers, they struggle to identify peak values and correctly estimate values not explicitly listed on the y-axis labels. Instead of implicitly learning structural and semantic information from charts using pre-trained models, we explicitly use graphs to capture the structural and semantic information from charts. To do so, we propose the multimodal scene graph for chart understanding with visual and textual graphs. The visual scene graph captures the spatial relation in terms of visual features, and the textual scene graph captures the semantic knowledge using textual features. Inspired by [4, 11], we integrated the graph representation as the inductive bias to the backbone model. Our contributions are: 1) We propose new multimodal scene graphs for chart understanding

[*] Corresponding author.
[1] Code available at https://github.com/adlnlp/MSG-Chart

*CIKM '24, October 21–25, 2024, Boise, ID, USA*

ACM ISBN 979-8-4007-0436-9/24/10
https://doi.org/10.1145/3627673.3679967



(a) Vision Encoder-Text Decoder (Backbone: UniChart [14])

(b) Multimodal Encoder-Text Decoder (Backbone: VL-T5[1])

Figure 2: Two Graph Integration Architectures (VG: Visual Graph, TG: Textual Graph)

to represent the structural and semantic relation. 2) The proposed joint multimodal scene graph can be adapted to diverse chart understanding backbones. 3) Our multimodal scene graph can enhance the performance of the diverse ChartQA backbone models.

# 2 Methodology

## 2.1 Graph Construction

Figure 2 presents the proposed graph-based model on two backbone frameworks: 1) Vision Encoder-Text Decoder (UniChart [14]) and 2) Multimodal Encoder-Text Decoder (VL-T5 [1]) The proposed model includes two undirected graphs: visual and textual graphs. The visual graph is initialised using the hidden states from the encoder, capturing structural information related to visual aspects. The textual graph uses chart elements' labels and OCR features, capturing semantic knowledge.

**Visual Graph** ($G_V$) is a fully connected graph designed to capture the spatial relations of objects from a visual aspect. Each node represents an object detected by the Mask R-CNN [5] fine-tuned by Masry et al. [15]. To maintain the spatial relationships between objects, inspired by the basis decomposition method introduced in R-GCN [20], we assign each edge $e$ a coefficient $a_e = \exp(-d)$, where $d$ represents the smallest Euclidean distance between the bounding boxes of two objects. This coefficient prioritises closer neighbours of a node by assigning them larger weights. With each object's bounding box, we can determine the alignment between image patches and corresponding objects. After that, each node is initialised as $V_o = \frac{1}{|P_o|}\sum_{p\in P_o}^{P_o} H_{e_p}$, where $P_o$ represents the set of patches that the object $o$ is located, and $H_e$ is the last hidden states from the encoder of the backbone model. For the vision encoder backbone in Figure 2a, the blue visual node at the bottom represents the dark blue pie slice in the chart image. It is initialised by the mean of hidden states from the corresponding patches. For the multimodal encoder backbone in Figure 2b, each ROI represents an intact object so the same visual node is directly initialised by the hidden states representing the dark blue pie slice.

**Textual graph** ($G_t$) is constructed with textual features based on chart elements. The graph has two types of nodes: label nodes and OCR nodes. Following scene graphs [21, 23, 24], each label node represents an object's label predicted by Mask R-CNN. Label nodes are connected based on the chart semantics. The x/y axis title connects to the x/y axis labels representing x/y axis values. X/Y axis labels connect to lines, bars, or dot lines if their bounding box values overlap on longitude/latitude, indicating value ranges of the latter. Legend labels connect to the closest legend marker to show the former, which is the label of chart elements with the same colour as the legend marker. Pie charts connect to all slices, representing the whole-part relationship, while pie labels connect the nearest pie slice to show the former is the latter's label. OCR nodes represent the OCR text extracted from non-shape objects such as the title and x-axis label. Following Methani et al. [16], we utilise the open-sourced Optical Character Recognition (OCR) model Tesseract[2] to extract texts from the cropped image of each detected object. These OCR nodes are linked to their corresponding label nodes. This allows the textual graph to capture not only the OCR texts from charts but also the relationships between the underlying data and objects. All nodes in the semantic graph are initialised with the Mean of BERT [3] embeddings of the text they represent. Given the graphs $G_v$, $G_t$, visual features $V$, and textual features $T$ that have been projected to the same space as $V$, we obtain graph representations $H_v = G_v(V)$ and $H_t = G_t(T)$ using GCN [8]. Afterwards, we concatenate the graph representations of the same object to combine the structural and semantic information, for example in Fig. 2, the nodes from visual and textual graphs representing the dark blue pie slice and concatenated together. We then utilise the multilayer perceptron with the ReLU activation function to project the concatenation into the space of the hidden states of the backbone model as $H_G$.

## 2.2 Graph Integration and ChartQA

Inspired by [11], we inject the graph representation as the inductive bias to the backbone model. We ensure that our proposed multimodal scene graph module functions can be a general plug-in component without substantially modifying the backbone model.

[2] https://github.com/tesseract-ocr/tesseract

Hence, we integrate the graph module between the encoder and text decoder. We tested our methods on the following two types of backbone models, Vision Encoder-Text Decoder (UniChart[14]) and Multimodal Encoder-Text Decoder (VL-T5[1]).

**Vision Encoder-Text Decoder**: UniChart[14] is a state-of-the-art open-source Vision Encoder-Text Decoder model for chart comprehension. UniChart utilises the Donut [7] encoder as the image encoder and the BART [10] decoder as the text decoder. The models with this architecture take sequential image patches as encoder input and question tokens as input for the decoder. To inject entity-related structural and semantical information into the model, we fuse the same node representation to the patches where the object is located. For example, in Figure 2a, the updated node representation of the dark blue slice from the graph module should be fused with the hidden states where the patches represent the same object. Specifically, we create a bias representation $H_b$ based on graph representation $H_G$. For an index $I$ and an object set $O_i$, where the bounding box of every object $o \in O_i$ intersects with patch $p_i$, the feature of $H_{b_i}$ is assigned by the mean of $H_{G_{O_i}}$. Formally,

$$H_{b_i} = \begin{cases} \frac{1}{|O_i|} \sum_{i=1}^{O_i} H_{G_i} & \text{if} \quad |O_i| > 0 \\ 0 & \text{if} \quad |O_i| = 0 \end{cases} \tag{1}$$

**Multimodal Encoder-Text Decoder**: The second backbone model is VL-T5, a widely used Multimodal Enocder-Text Decoder model employed in prior research for ChartQA and Chart Summarisation tasks [6, 15, 22]. VL-T5 takes both textual tokens and ROI features as input. Since each ROI feature represents one object, we fuse the node representations with their corresponding hidden states. In the case of Figure 2b, the updated node representation of the dark blue pie slice will be fused with the second visual token (blue rectangle) from the encoder. The new representation $H_b$ is a special case of Eq. 1 where $|O_i| = 1$. Since VL-T5 is pre-trained with 36 object regions, we select the top 36 objects detected by Mask R-CNN ranked by confidence score. If fewer than 36 objects are detected, we pad the input with zeros to a fixed length of 36. The original hidden states from the encoder $H_e$ are updated by adding the bias $H_b$, resulting in $\widetilde{H_e} = H_e + H_b$. The new hidden states are then passed to the decoder for answer generation.

Given an image $I$, texts $x$, and label $y$, we jointly train the parameters of the backbone model $\theta_m$, visual and textual graphs $\theta_v, \theta_t$ by minimizing the negative log-likelihood:

$$L_{\theta_m,\theta_v,\theta_t} = -\sum_{j=1}^{|y|} \log P_{\theta_m,\theta_v,\theta_t}(y_j | y_{<j}, x, I) \tag{2}$$

# 3 Experiment

We tested our proposed model on ChartQA and OpenCQA: **ChartQA** is currently the most challenging Chart Question Answering dataset. The questions in this dataset are either generated by a fine-tuned T5 [18] model or through human annotation, resulting in two sets: augmentation and human. The questions in the human set are more difficult as they emphasise logical and visual reasoning. Following [13, 15, 16], we use relaxed accuracy, which requires an exact match for textual answers and allows a 5% tolerance for numerical answers. **OpenCQA** [6] is an open-ended ChartQA dataset. Unlike previous ChartQA datasets, where the answers are mainly words or phrases, the answers in OpenCQA are explanatory texts, with an average answer length of 56 tokens. We use BLEU4 [17] following [6, 14]. We scaled the range from 0 to 100, which was consistent with previous research.

We have different text input settings for different models and datasets. **ChartQA** For VL-T5, we use the flattened ground truth table and question tokens as text inputs. This is to ensure fair comparison with the performance in [15], because the model they used to generate table from chart automatically is not available. For UniChart, we only use questions as text input. **OpenCQA** The original work in [6] has three settings. We use the setting where chart images, questions and OCR texts are used as input. We use this setting for VL-T5 as it's the one used in comparison with UniChart paper. For UniChart, the text input is the question.

All experiments were conducted using a single Nvidia A100 (40G) GPU, with the random seed set to 42 for reproducibility. We use AdamW as the optimiser. The settings for VL-T5 on both datasets follow the original papers [15] and [6], with the exception that the batch size for ChartQA is 24 and for OpenCQA is 12 due to GPU memory limitations. The settings for UniChart follow [14], with the model trained using mixed precision consistent with the provided code. The batch sizes for both datasets are set to 8 due to GPU memory constraints. Additionally, the number of training epochs for OpenCQA is set to 20, different from the original paper, because we observed overfitting with 20 epochs. The GCNs for both the visual and textual graphs consist of 2 layers, and the hidden state dimensions matched the backbone models: 1024 for UniChart and 768 for VL-T5. The dropout rate is set to 0.2.

| | | | | ChartQA | | OpenCQA |
|---|---|---|---|---|---|---|
| | GT | Graph | aug. | human | avg. | BLEU |
| Pix2Struct [9] | × | × | 81.60 | 30.50 | 56.00 | - |
| Matcha[3] [13] | × | × | 90.20 | 38.20 | 64.20 | - |
| Matcha[4] | × | × | 86.64 | 36.96 | 61.80 | - |
| Matcha[5] | × | × | 81.28 | 28.16 | 54.72 | - |
| UniChart[3] [14] | × | × | 88.56 | 43.92 | 66.24 | 14.88 |
| UniChart[4] | × | × | 82.32 | 34.48 | 58.40 | 8.76 |
| UniChart[5] | × | × | 82.00 | 30.80 | 56.40 | 10.86 |
| **UniChart (Ours)** | × | ✓ | **85.36** | **37.44** | **61.4** | **11.97** |
| VL-T5 [6, 15] | ✓ | × | - | - | 59.12 | 14.73 |
| **VL-T5 (Ours)** | ✓ | ✓ | **92.4** | **38** | **64.8** | **17.14** |

**Table 1: Overall Performance on ChartQA (GT: Golden Table)**

# 4 Results

## 4.1 Overall Performance

We present the results of our models compared with previous baselines in Table 1. Note that the performance of UniChart differs from the results reported in the original paper. The publicly available code or the given huggingface checkpoint[3] provided by [14] are unable to reproduce the same performance, which yields an accuracy of 58.4 instead of the reported 66.24. This discrepancy may be due to differences in the environment, such as GPU configuration and

[3] Result from original paper
[4] Result from checkpoint in huggingface
[5] Result from model fine-tuned by ourselves
[3] https://huggingface.co/ahmed-masry/unichart-base-960

random seed. For a fair comparison, we fine-tuned the model in our environment, which resulted in a 56.4 overall accuracy. Similarly, for UniChart on OpenCQA, the result from the provided checkpoint is 8.76 rather than the reported 14.88 from the paper, and the result fine-tuned by ourselves is 10.86. Under the same training environment, our proposed graph module enhances the performance of both UniChart and VL-T5 on both datasets. Specifically, UniChart demonstrates at least a 3% improvement in accuracy on ChartQA and a 1.1-point increase in BLEU score on OpenCQA. For VL-T5, the accuracy on ChartQA increases by 5.68%, and the BLEU score on OpenCQA rises by 2.41 points. The trend of the produced result shows that integrating our components with any state-of-the-art chartQA framework is feasible. It also proves the superiority of our proposed graph model, which consistently improves overall performance in diverse types of chartQA tasks.

## 4.2 Ablation Studies

| | | | | ChartQA | | OpenCQA |
|---|---|---|---|---|---|---|
| | VG | TG | aug. | human | avg. | BLEU |
| UniChart | ✓ | ✓ | **85.36** | **37.44** | **61.4** | **11.97** |
| | ✓ | × | 84.88 | 35.84 | 60.36 | 11.42 |
| | × | ✓ | 82.8 | 34.56 | 58.68 | 11.76 |
| | × | × | 82.32 | 34.48 | 58.40 | 8.76 |
| VL-T5 | ✓ | ✓ | **92.4** | **38** | **64.8** | 17.14 |
| | ✓ | × | 91.12 | 36.08 | 63.6 | 16.99 |
| | × | ✓ | 90.56 | 37.36 | 63.96 | **18.14** |
| | × | × | - | - | 59.12 | 14.73 |

**Table 2: Performance comparison w.r.t. different graph setting (VG: Visual Graph, TG: Textual Graph)**

**Effect of the VG and TG:** We conducted ablation studies with different graph settings. As shown in Table 2, performances are consistently better when both graphs are employed. For UniChart and VL-T5, omitting the visual graph leads to the most significant performance drop on the augmented test set of ChartQA, with decreases of 2.56% for UniChart and 1.84% for VL-T5. This aligns with the findings of [13] that the augmented set contains more extractive questions, while the human set includes more complex reasoning questions. This demonstrates that our visual graph enhances the models' structural understanding of charts. When only the visual graph is used, the performance on the human set increases by 1.36% for UniChart. However, when both the visual and textual graphs are utilised, the performance on the human set improves by 2.96%, indicating that the textual graph significantly enhances the semantic understanding of charts. However, since complex math reasoning questions also require sophisticated structural understanding, using textual graphs alone doesn't necessarily result in the second-best performance (results with underlining) on the human set, as seen with VL-T5. Nonetheless, when both graphs are used, the performance on the human set is consistently the best for both models.

**Effect of the proposed relation-based graphs:** To evaluate whether the proposed relation in graphs can effectively capture semantic information in charts, we replaced the proposed edges in textual graphs with fully connected edges. The results in Table 3 reveal that fully connected graphs significantly decrease performance on the human set of ChartQA, with a 4% drop for UniChart and a 3.2% drop for VL-T5. This difference in the human test set indicates that a carefully designed scene graph enhances the backbone model's ability to understand the semantics of charts. OpenCQA does not have a significant gap between those with diverse edge types, and it is expected because OpenCQA includes more explanatory and descriptive text to answer questions.

| | | | ChartQA | | OpenCQA |
|---|---|---|---|---|---|
| | TG Relation | aug. | human | avg. | BLEU |
| UniChart | designed rules | **85.36** | **37.44** | **61.4** | **11.97** |
| | fully connected | 84.24 | 33.44 | 58.84 | 11.40 |
| VL-T5 | designed rules | **92.4** | **38** | **64.8** | 17.14 |
| | fully connected | 91.68 | 34.8 | 63.24 | **17.92** |

**Table 3: Performance comparison w.r.t. designed semantic relations and fully connected graph (TG: Textual Graph)**

| | | | | ChartQA | | OpenCQA |
|---|---|---|---|---|---|---|
| | G | Text | aug. | human | avg. | BLEU |
| UniChart | ✓ | label+OCR | **85.36** | **37.44** | **61.4** | **11.97** |
| | × | × | 82.32 | 34.48 | 58.4 | 8.76 |
| | × | label | 83.52 | 35.04 | 59.28 | 11.46 |
| | × | OCR | 84 | 33.68 | 58.84 | 11.15 |
| VL-T5 | ✓ | label+OCR | **92.4** | **38** | **64.8** | 17.14 |
| | × | × | - | - | 59.12 | 14.73 |
| | × | label | 92.32 | 36.08 | 64.2 | 15.05 |
| | × | OCR | 90.56 | 36.08 | 63.32 | **17.78** |

**Table 4: Performance comparison with the presence of graph and textual feature only (G: Graph)**

**Effect of textual features** We conducted an experiment where the graphs were removed, and the textual features were directly fused into the model. We fused either the object's label or the OCR textual feature with the visual features from the encoder. The results are shown in Table 4. The performances with the presence of the graphs are better compared to solely injecting textual features. Depending on the different textual features, the improvement gained from using graphs is at least 2.12% for UniChart and 0.6% for VL-T5 on ChartQA. Notably, fusing the label feature yields higher accuracy than the OCR feature, likely due to the noisy text introduced by the OCR error during text detection.

# 5 Conclusion

This research proposes a novel multimodal scene graph, including a visual graph and a textual graph, to capture the structure and semantic information from charts. The graph module can be easily integrated into different types of chartQA frameworks. Through experiments, we show that the graph module can improve the understanding of charts and consequently perform better on ChartQA tasks. We hope that this multimodal chart-based scene graph can be a useful stepstone to improve the performance of chart understanding and retrieval.

## References

[1] Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. 2021. Unifying Vision-and-Language Tasks via Text Generation. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, 1931–1942. http://proceedings.mlr.press/v139/cho21a.html

[2] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2024. Scaling Instruction-Finetuned Language Models. *Journal of Machine Learning Research* 25, 70 (2024), 1–53. http://jmlr.org/papers/v25/23-0870.html

[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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, 4171–4186. https://doi.org/10.18653/V1/N19-1423

[4] Caren Han, Siqu Long, Siwen Luo, Kunze Wang, and Josiah Poon. 2020. VICTR: Visual Information Captured Text Representation for Text-to-Vision Multimodal Tasks. In *Proceedings of the 28th International Conference on Computational Linguistics*. 3107–3117.

[5] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. 2017. Mask R-CNN. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. IEEE Computer Society, 2980–2988. https://doi.org/10.1109/ICCV.2017.322

[6] Shankar Kantharaj, Xuan Long Do, Rixie Tiffany Ko Leong, Jia Qing Tan, Enamul Hoque, and Shafiq R. Joty. 2022. OpenCQA: Open-ended Question Answering with Charts. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (Eds.). Association for Computational Linguistics, 11817–11837. https://doi.org/10.18653/V1/2022.EMNLP-MAIN.811

[7] Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. 2022. OCR-Free Document Understanding Transformer. In *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXVIII (Lecture Notes in Computer Science, Vol. 13688)*, Shai Avidan, Gabriel J. Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (Eds.). Springer, 498–517. https://doi.org/10.1007/978-3-031-19815-1_29

[8] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=SJU4ayYgl

[9] Kenton Lee, Mandar Joshi, Iulia Raluca Turc, Hexiang Hu, Fangyu Liu, Julian Martin Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. 2023. Pix2Struct: Screenshot Parsing as Pretraining for Visual Language Understanding. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA (Proceedings of Machine Learning Research, Vol. 202)*, Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (Eds.). PMLR, 18893–18912. https://proceedings.mlr.press/v202/lee23g.html

[10] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault (Eds.). Association for Computational Linguistics, 7871–7880. https://doi.org/10.18653/V1/2020.ACL-MAIN.703

[11] Jinyang Li, Binyuan Hui, Reynold Cheng, Bowen Qin, Chenhao Ma, Nan Huo, Fei Huang, Wenyu Du, Luo Si, and Yongbin Li. 2023. Graphix-T5: Mixing Pre-trained Transformers with Graph-Aware Layers for Text-to-SQL Parsing. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, Brian Williams, Yiling Chen, and Jennifer Neville (Eds.). AAAI Press, 13076–13084. https://doi.org/10.1609/AAAI.V37I11.26536

[12] Fangyu Liu, Julian Martin Eisenschlos, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Wenhu Chen, Nigel Collier, and Yasemin Altun. 2023. DePlot: One-shot visual language reasoning by plot-to-table translation. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, 10381–10399. https://doi.org/10.18653/V1/2023.FINDINGS-ACL.660

[13] Fangyu Liu, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Yasemin Altun, Nigel Collier, and Julian Martin Eisenschlos. 2023. MatCha: Enhancing Visual Language Pretraining with Math Reasoning and Chart Derendering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, 12756–12770. https://doi.org/10.18653/V1/2023.ACL-LONG.714

[14] Ahmed Masry, Parsa Kavehzadeh, Do Xuan Long, Enamul Hoque, and Shafiq Joty. 2023. UniChart: A Universal Vision-language Pretrained Model for Chart Comprehension and Reasoning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*. Association for Computational Linguistics, 14662–14684. https://aclanthology.org/2023.emnlp-main.906

[15] Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq R. Joty, and Enamul Hoque. 2022. ChartQA: A Benchmark for Question Answering about Charts with Visual and Logical Reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.). Association for Computational Linguistics, 2263–2279. https://doi.org/10.18653/V1/2022.FINDINGS-ACL.177

[16] Nitesh Methani, Pritha Ganguly, Mitesh M. Khapra, and Pratyush Kumar. 2020. PlotQA: Reasoning over Scientific Plots. In *IEEE Winter Conference on Applications of Computer Vision, WACV 2020, Snowmass Village, CO, USA, March 1-5, 2020*. IEEE, 1516–1525. https://doi.org/10.1109/WACV45572.2020.9093523

[17] Matt Post. 2018. A Call for Clarity in Reporting BLEU Scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers, WMT 2018, Belgium, Brussels, October 31 - November 1, 2018*, Ondrej Bojar, Rajen Chatterjee, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno-Yepes, Philipp Koehn, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana L. Neves, Matt Post, Lucia Specia, Marco Turchi, and Karin Verspoor (Eds.). Association for Computational Linguistics, 186–191. https://doi.org/10.18653/V1/W18-6319

[18] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.* 21 (2020), 140:1–140:67. http://jmlr.org/papers/v21/20-074.html

[19] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett (Eds.). 91–99. https://proceedings.neurips.cc/paper/2015/hash/14bfa6bb14875e45bba028a21ed38046-Abstract.html

[20] Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling Relational Data with Graph Convolutional Networks. In *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings (Lecture Notes in Computer Science, Vol. 10843)*, Aldo Gangemi, Roberto Navigli, Maria-Esther Vidal, Pascal Hitzler, Raphaël Troncy, Laura Hollink, Anna Tordai, and Mehwish Alam (Eds.). Springer, 593–607. https://doi.org/10.1007/978-3-319-93417-4_38

[21] Tripti Shukla, Paridhi Maheshwari, Rajhans Singh, Ankita Shukla, Kuldeep Kulkarni, and Pavan K. Turaga. 2023. Scene Graph Driven Text-Prompt Generation for Image Inpainting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023 - Workshops, Vancouver, BC, Canada, June 17-24, 2023*. IEEE, 759–768. https://doi.org/10.1109/CVPRW59228.2023.00083

[22] Benny J. Tang, Angie Boggust, and Arvind Satyanarayan. 2023. VisText: A Benchmark for Semantically Rich Chart Captioning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, 7268–7298. https://doi.org/10.18653/V1/2023.ACL-LONG.401

[23] Cantao Wu, Yi Cai, Liuwu Li, and Jiexin Wang. 2023. Scene Graph Enhanced Pseudo-Labeling for Referring Expression Comprehension. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, 11978–11990. https://doi.org/10.18653/V1/2023.FINDINGS-EMNLP.802

[24] Kanghoon Yoon, Kibum Kim, Jinyoung Moon, and Chanyoung Park. 2023. Unbiased Heterogeneous Scene Graph Generation with Relation-Aware Message Passing Neural Network. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, Brian Williams, Yiling Chen, and Jennifer Neville (Eds.). AAAI Press, 3285–3294. https://doi.org/10.1609/AAAI.V37I3.25435