SciCap: Generating Captions for Scientific Figures

Ting-Yao (Edward) Hsu, C. Lee Giles, Ting-Hao 'Kenneth' Huang
Pennsylvania State University
University Park, PA, USA
{txh357, clg20, txh710}@psu.edu

Abstract

Researchers use figures to communicate rich, complex information in scientific papers. The captions of these figures are critical to conveying effective messages. However, low-quality figure captions commonly occur in scientific articles and may decrease understanding. In this paper, we propose an end-to-end neural framework to automatically generate informative, high-quality captions for scientific figures. To this end, we introduce **SciCap**,[1] a large-scale figure-caption dataset based on computer science arXiv papers published between 2010 and 2020. After pre-processing – including figure-type classification, sub-figure identification, text normalization, and caption text selection – SciCap contained more than two million figures extracted from over 290,000 papers. We then established baseline models that caption graph plots, the dominant (19.2%) figure type. The experimental results showed both opportunities and steep challenges of generating captions for scientific figures.

1 Introduction

Researchers use figures to explain complex concepts or show critical results. In scholarly articles, figure captions are critical to get the message across effectively. Ones that are too generic (*e.g.*, "Results of Experiment A.") or poorly written (*e.g.*, "Relations between X and Y.") represent missed opportunities to explain scientific narratives to readers. Unfortunately, such low-quality captions still occur in published scientific articles. This paper aims to develop automatic figure-captioning models that generate high-quality captions for figures and charts in scientific papers (Figure 1).

Our motivation is two-fold. First, we aim to help researchers write better captions for the figures and charts in their papers. Automatic caption models trained on informative, high-quality captions can suggest better captions. Second, the proposed technology can make scientific charts and figures more accessible to blind or visually impaired readers. Researchers have developed technologies to assist the blind to navigate graphical content, such as data visualization charts (Swaminathan et al., 2014), printed physical maps (Swaminathan et al., 2016), 3D chemical diagrams (Bernareggi et al., 2019), and images on social media (Wu et al., 2017; Salisbury et al., 2017). However, only a few prior works focused on scientific figures. An image-captioning model specialized for scientific figures can improve the narration of scientific articles for the blind even when the original caption is unhelpful.

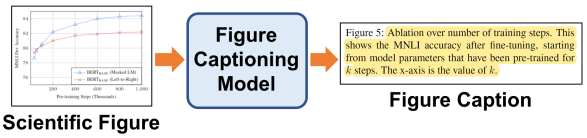


Figure 1: The figure captioning model takes a scientific figure (*e.g.*, a graph plot) as input and generate captions that describes the figure.

To this end, we introduce **SciCap**, a large-scale image-captioning dataset that contains real-world scientific figures and captions. SciCap was constructed using computer science papers collected and released by arXiv. With pre-processing complete – including figure-type classification, subfigure identification, text normalization, and caption text selection – SciCap contained more than two million figures extracted from over 290,000 papers. We then established baseline models that caption graph plots, the dominant (19.2%) figure type. The experimental results showed both exciting opportunities and steep challenges of generating captions for scientific figures.

2 Related Work

One of the few prior works attempting to caption scientific figures was by Chen *et al.* (2019a; 2019b;

[1] SciCap is available at: https://github.com/tingyaohsu/SciCap

2020). They created FigCAP, a caption-figure pair corpus where the figures are synthesized, and used an LSTM model with an attention mechanism to produce captions. FigCAP was built on research that aimed to analyze figure content automatically, including Figure-Seer (Siegel et al., 2016), FigureQA (Kahou et al., 2017), and DVQA (Kafle et al., 2018). DVQA and FigureQA were both made using synthetic figures; FigureSeer contained over 60,000 figures across seven figure types extracted from research papers. Meanwhile, Qian *et al.* (2020) proposed a set of "caption units" (such as Title, Label Name, Min/Max, etc.) that are important to include in a caption of scientific figures; they created a model, FigJAM, to produce such units (Qian et al., 2021). Also relevant is the "data-to-caption" work, which takes a chart's source data table and metadata as input to generate a caption (Obeid and Hoque, 2020; Spreafico and Carenini, 2020). These models generate captions based on data tables, not the figures.

Differences Between Synthetic and Real-World Captions. Most prior work has tried to generate captions for scientific figures using synthetic images and texts (Chen et al., 2019a,b, 2020; Kahou et al., 2017). However, synthetic captions tend to be generic and describe features without conveying higher-level insights, for example, *"This is a line plot. It contains 6 categories. Dark Magenta has the lowest value. Lawn Green has the highest value."* (example from FigCAP.) Human-written captions, on the other hand, tend to highlight the meaningful parts of the figure and bring more context, for example: *"Train loss curve with respect to optimization steps. With prior coarse-tuning on NLI data, convergence becomes much faster and easier."* [example from (Jin et al., 2020)].

3 Constructing SciCap Dataset

This section describes the process that massages real-world figure-caption data into an appropriate easy-to-use format for the NLP community. This data-processing procedure was developed iteratively and empirically.

Step 1: Data Acquisition and Pre-processing. Data acquisition is a fundamental challenge for constructing a public scientific figure-caption dataset. Although there is a vast number of scientific papers, they are not all easy to access. SciCap is based on the arXiv dataset (Clement et al., 2019).[2] The arXiv dataset is licensed under CC-0, which grants remake and republish rights. It contains a repository of 1.7 million articles with relevant features, such as article titles, authors, categories, abstracts, full-text PDFs, and more.

We first downloaded all the scholarly articles from the arXiv dataset and froze the date on Dec 22, 2020 (a total of 1,921,287 papers). SciCap does not include any papers published after this date. We further narrowed our dataset to papers published between 2010 and 2020 in computer science (cs.) and machine learning (stat.ML) topics, which numbered 295,028 papers. We did not use these papers' "source files," which might contain the original LaTeX and figure files. Not all papers come with source files; some source files have complex dependencies that are hard to parse.

Step 2: Figure-Caption Pair Extraction. We then used PDFFigures 2.0 (Clark and Divvala, 2016) to extract the figures from papers in our paper collection. PDFFigures 2.0 is a Scala-based tool created to extract figures, captions, tables, and section titles from scholarly documents, with a focus on the computer science domain. In addition to the figures' images and captions, the tool also extracted all the text snippets inside the figures, such as legends, X-Y labels, and titles. The extracted information can be used to boost the performance of image-captioning models. This step resulted in 295,028 papers and 2,170,719 figures.

Step 3: Figure Type Classification. Given the high diversity in the figure types included in scientific articles, we did not aim to create a single captioning model for all types of figures. Instead, we aimed to create captioning models specialized for one particular figure type. We used an automatic figure type classifier (Siegel et al., 2016) to classify figure type in SciCap. This pre-trained classifier can identify seven types of figures: graph plots, flowcharts (also called node diagrams), equations (also called algorithms), bar plots, scatter plots, tables, and "other." Its reported accuracy is 86% over 60,000 samples (Siegel et al., 2016).

According to the classifier's prediction, out of 2,170,719 figures, 19.2% (416,804) are graph plots, 23.6% (511,984) are tables,[3] 5.9% (127,197) are

[2] arXiv Dataset on Kaggle: https://www.kaggle.com/Cornell-University/arxiv

[3] In this work, tables are not considered to be figures due to drastically different visual features and contents.

equations (including algorithms and pseudo codes), 8.5% (185,398) are flowcharts, 2.0% (44,052) are scatter plots, 4.7% (101,146) are bar charts, and 36.1% (784,138) are "other." In SciCap, we only focus on graph plots, which have the highest classification performance (Siegel et al., 2016) and are also the most common figure type.

Step 4: Removing Figures with Subfigures. Many scientific figures contain subfigures. For example, in our pilot study (Section 3.1), 35.72% of overall scientific figures had subfigures. SciCap focuses on generating captions for single figures, so we removed figures with subfigures from the dataset. We first used handcrafted rules to identify captions that explicitly mention or refer to subfigures [for example, `(a)`, `a)`, `(b)`, `b)`, `(1)`, `1)`, `(2)`, `2)` ... etc.]. Furthermore, we also used FigureSeparator (Tsutsui and Crandall, 2017) to filter figures with subfigures out of our collection. FigureSeparator is a CNN-based model that separates compound figures in the ImageCLEF Medical dataset with 85.9% accuracy.

Of 416,804 graph plots identified in Step 3, the rule-based approach yielded 352,719 graph plots, and the FigureSeparator further narrowed the collection down to 133,543 figures. An estimated 32.04% of the graph plots did not have subfigures.

Step 5: Text Normalization. We used NLTK (Loper and Bird, 2002) for tokenization and converted all the text to lowercase. We also removed the figure numbers, such as "Figure 1:" or "Fig. 1:", and only kept the main caption text. The following two text normalization strategies were then applied:

- **Basic Normalization:** We replaced all the numbers (*e.g.*, `0`, `-0.2`, `3.44%`, `1,000,000`) with `[NUM]`.
- **Advanced Normalization:** We created regular expressions to identify equations in captions and replaced them with `[EQUATION]`. We also replaced all the text spans enclosed by any types of bracket pairs, including `{}`, `[]`, and `()`, with `[BRACKET]`.

Step 6: Target Caption Text Selection. SciCap provides three different data collections, each sampled using different strategies:

- **First Sentence (133,543 Figures):** This collection includes all the figures. For each figure included, this collection only includes the first sentence of the caption.
- **Single-Sentence Caption (94,110 Figures):** This collection includes the complete caption of only the figures with a one-sentence caption. Of the graph plots, 70.47% had a one-sentence caption.
- **Caption with No More than 100 Words (131,319 Figures):** This collection includes the complete caption of only the figures whose captions contained no more than one hundred tokens (punctuation marks included). In this collection, a caption contains 1.66 sentences on average (SD=1.07).

Figure Type Classification (Class = Graph Plot)				
Approach	**P**	**R**	**F**	**Acc**
(Siegel et al., 2016)	.90	.83	.87	.95
Non-Subfigure Figure Classification (For figures labeled as graph plots in Step 3.)				
Approach	**P**	**R**	**F**	**Acc**
Rule-Based	.54	.95	.69	.59
FigureSeparator	.98	.66	.79	.83
Rule-Based+FigureSeparator	.98	.62	.76	.81

Table 1: The tools used to construct SciCap evaluated on 1,926 labeled images. For figure type classification, the overall performance over graph plots was reliable. Regarding identifying the graph plots (as labeled automatically in Step 3) that do not contain subfigures, FigureSeparator achieved an exceptionally high precision.

On average, with advanced normalization (Step 4), a sentence in the "First Sentence" collection contains 23.19 tokens (SD=20.86); a sentence in the "Single-Sentence Caption" collection contains 14.05 tokens (SD=8.15); and a sentence in the "Caption with No More Than 100 Words" collection contains 22.04 tokens (SD=17.44).

Note that we first created the 80/10/10 train/val/test data split for the entire corpus and then proceeded with the caption selection step. This procedure ensured that we used the identical set of figures to construct each collection's test set; the same applied to their training and validation sets.

3.1 Data Analysis and Quality Measurement

To evaluate the quality of our data cleaning and processing pipeline, we randomly sampled 2,000 figures from the original arXiv dataset, and one

author manually labelled each figure's figure type and whether it contained subfigures (Yes/No).[4] Of these 2,000 figures, 1,926 figures had no extraction errors, and were included in our follow-up calculation. As for types, 20.35% of the figures were graph plots, 4.1% were bar charts, and 3.11% were scatter plots.[5] In terms of subfigures, 237 out of 1,926 figures (35.72%) contained subfigures: 33.14% of these figures contained graph plots as subfigures, 5.81% contained bar charts, and 6.83% contained scatter plots.

We used these 1,926 labeled images to evaluate the tools we employed in constructing SCICAP. Table 1 shows the results. For the figure type classification, the overall performance over graph plots were reliable. Regarding identifying the graph plots (as labeled automatically in Step 3) that do not contain subfigures, FigureSeparator had an exceptionally high precision.

4 Experimental Results

To examine the feasibility and challenges of creating an image-captioning model for scientific figures, we established several baselines and tested them using SCICAP. The caption quality was measured by BLEU-4 (Papineni et al., 2002), using the test set of the corresponding data collection as a reference. Figure 2 shows some example outputs.

Baseline Model. We used a classical image-captioning model, CNN+LSTM architecture, as our baseline (Xu et al., 2015). The pre-trained ResNet-101 (He et al., 2016) was used as the image encoder to represent a figure as a 2048-dimension vector. This image vector was then fed into a dense layer to fit the dimension of the word-embedding and the LSTM decoder where the word-embedding and LSTM hidden layer size were all 512. A global attention mechanism was added to the LSTM decoder to better model the context (Luong et al., 2015). The LSTM decoder took the image vector as the initial state and generate captions.

We designed three variations of the baseline models, Vision-only, Vision+Text, and Text-only.

[4]To validate the label quality, we had three graduate students label 100 figures, respectively. On average, they agreed with 97% of our subfigure labels. For the figures without subfigures, they agreed with our figure type labels 82.17% of the time. For the figures with subfigures, they agreed with at least one of our type labels 86.56% of the time.

[5]A figure might contain subfigures of different types (*e.g.*, a bar chart accompanied by a graph plot.) For each figure, we took a multi-class labeling strategy that exhaustively labels all distinct types of its subfigures.

First Sentence						
Subfig Filter		**Norm.**		**#Fig.**	**Vocab Size**	**BLEU-4**
Rule	**FigSep**	**B.**	**A.**			
				416,804	30,776	.0259
✓		✓		352,719	24,355	.0236
✓	✓	✓		133,543	12,666	.0224
✓	✓	✓	✓		11,946	.0219
Single-Sentence Caption Only						
Subfig Filter		**Norm.**		**#Fig.**	**Vocab Size**	**BLEU-4**
Rule	**FigSep**	**B.**	**A.**			
				247,649	21,765	.0291
✓		✓		218,655	17,685	.0228
✓	✓	✓		92,021	9,760	.0234
✓	✓	✓	✓		9,232	.0207
Caption with <= 100 Words						
Subfig Filter		**Norm.**		**#Fig.**	**Vocab Size**	**BLEU-4**
Rule	**FigSep**	**B.**	**A.**			
				395,024	37,885	.0231
✓		✓		341,350	30,316	.0098
✓	✓	✓		132,120	15,642	.0173
✓	✓	✓	✓		14,974	.0172

Table 2: The baseline model's performance on SCICAP, using Vision-Only features. Models trained on the Single-Sentence Caption collection performed the best. The low BLEU-4 scores indicate that more research is needed to reliably generate captions for scientific figures. (The vocabulary sizes were calculated after dropping words with a frequency below 5.)

The text information was the titles, legends, and X-Y labels extracted from the figures (Step 2 in Section 3). Another LSTM was used as a text encoder to encode text information into a vector. For the Vision+Text variation, we concatenated the image vector and the text vector together and fed it into the LSTM decoder for caption generation. The Text-only variation only took the text vector as the feature for the LSTM decoder.

Experimental Setups. We trained the baseline models using an 80/10/10 train/val/test data split. The models were trained by minimizing a cross-entropy loss with a doubly stochastic regularization (Xu et al., 2015) using Adam (Kingma and Ba, 2014). The weights of the pretrained ResNet-101 image encoder were partially frozen so that only convolutional blocks 2 through 4 were fine-tuned throughout the training process (Yosinski et al.,

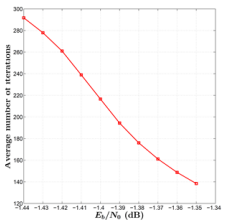
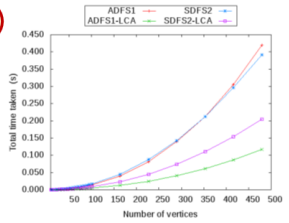
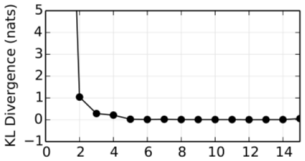
	1	2	3
Figure (Graph Plot)			
Original Caption	Fig. 18. Average number of iterations required to decode the PLDPC-Hadamard code with $r = 8$ and $k = 204, 800$.	Figure 7: Comparison of total time taken and time taken by LCA/LA data structure by the most efficient algorithms for insertion of $m = \binom{n}{2}$ edges for different values of n.	Figure 3: The KL divergence between the values of ϕ^{R} at the current and previous epochs on the thermus dataset.
Normalized Caption *(Before replacing the low-freq tokens with <unk>.)*	average number of iterations required to decode the pldpc-hadamard code with [EQUATION] and [EQUATION] , [NUM] .	comparison of total time taken and time taken by lca/la data structure by the most efficient algorithms for insertion of m = [BRACKET] edges for different values of n .	the kl divergence between the values of \u03c6r at the current and previous epochs on the thermus dataset .
LSTM-Generated Caption	the average number of iterations required for the convergence of algorithm [NUM] . *(BLEU-4 = 0.25)*	the average time to find the <unk> in the <unk> for different values of <unk> . *(BLEU-4 = 0.08)*	the <unk> function <unk> [BRACKET] for [EQUATION] and [NUM] . *(BLEU-4 = 0.00)*

Figure 2: Example outputs of the baseline models trained and tested on the Single-Sentence Caption Only collection. Intensive research will be needed to create models that can caption scientific figures reliably. [Figure sources: (1) (Zhang et al., 2020), (2) (Baswana et al., 2017), and (3) (Brubaker et al., 2015).]

Data Collection	Feature	BLEU-4
First Sentence	Vision Only	.0219
	Vision+Text	.0205
	Text Only	.0213
Single-Sent Caption	Vision Only	.0207
	Vision+Text	.0202
	Text Only	.0212
Caption w/ <=100 words	Vision Only	.0172
	Vision+Text	.0168
	Text Only	.0165

Table 3: The experimental results of models using Vision-Only, Text-Only, and Vision+Text features. Vision-Only and Text-Only features yielded similar performance. (All the subfigure-filtering and text-normalization steps were applied.)

2014). We empirically set the hyper-parameters by observing the performance gain on the validation set. Hyper-parameters ended up being used were a dropout rate of 0.5; a batch size of 16/32; a learning rate of 4e-4 with a decay factor of 0.8 when there was no improvement for 8 epochs. The models were trained until there was no improvement for 20 epochs. We kept the model with the highest BLEU-4 score on the validation set for testing.

Results. We trained the models on each data collection with varying levels of data filtering and text normalization. Table 2 shows the results. Among the three data collections, the models trained on the single-sentence captions performed the best. This might be because the Single-Sentence Caption collection, which is a subset of the First Sentence collection, had the smallest vocabulary size.

Effects of Text Normalization. Our experiments did not show the clear benefits of normalizing text to the resulting BLEU-4 scores. We will explore other methods to normalize text, for example, using advanced techniques to identify equations in text (Mali et al., 2020; Mansouri et al., 2020).

Effects of Text and Vision Features. We also used Vision-Only, Text-Only, and Text+Vision features to develop models (Table 3). Vision-Only and Text-Only features yielded similar performance. Furthermore, the models performed slightly worse when training on combined features.

5 Conclusion and Future Work

This paper introduces SciCap, a large-scale image-captioning dataset that contains real-world scientific figures and captions. SciCap was constructed using more than two million images from over 290,000 papers collected and released by arXiv. We also established several image-captioning baselines, showing the feasibility and challenges of generating captions for scientific figures. In the future, we will explore approaches to improve caption quality, such as taking advantage of large pre-trained language models (Beltagy et al., 2019), or using information in paper's full text to boost performance.

Ethical Considerations

Data Licensing. The arXiv dataset uses the CC0 1.0 Universal (CC0 1.0) Public Domain Dedication license,[6] which grants permission to remix, remake, annotate, and publish the data.

Potential Biases of Language Technologies. We are aware that language technologies trained on a "standard" or mainstream variety of a language (in our case, English) favor the popular variety and harms people using varieties with fewer speakers. For example, standard automatic speech recognition trained on Dutch speeches results in 10-15% higher error rates on Flemish Dutch than on "standard" Dutch (Feng et al., 2021).

Acknowledgments

We thank Chieh-Yang Huang, Hua Shen, and Chacha Chen for helping with the data annotation. We thank Chieh-Yang Huang for the feedback and strong technical support. We also thank the anonymous reviewers for their constructive feedback. This research was partially supported by the Seed Grant (2020) from the College of Information Sciences and Technology (IST), Pennsylvania State University.

References

Surender Baswana, Ayush Goel, and Shahbaz Khan. 2017. Incremental dfs algorithms: a theoretical and experimental study. *arXiv preprint arXiv:1705.02613*.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: Pretrained language model for scientific text. In *EMNLP*.

Cristian Bernareggi, Dragan Ahmetovic, and Sergio Mascetti. 2019. μgraph: Haptic exploration and editing of 3d chemical diagrams. In *The 21st International ACM SIGACCESS Conference on Computers and Accessibility*, pages 312–317. ACM.

Marcus A Brubaker, Ali Punjani, and David J Fleet. 2015. Building proteins in a day: Efficient 3d molecular reconstruction. *arXiv preprint arXiv:1504.03573*.

Charles Chen, Ruiyi Zhang, Sungchul Kim, Scott Cohen, Tong Yu, Ryan Rossi, and Razvan Bunescu. 2019a. Neural caption generation over figures. In *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*, UbiComp/ISWC '19 Adjunct, pages 482–485, New York, NY, USA. ACM.

Charles Chen, Ruiyi Zhang, Eunyee Koh, Sungchul Kim, Scott Cohen, and Ryan Rossi. 2020. Figure captioning with relation maps for reasoning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1537–1545.

Charles Chen, Ruiyi Zhang, Eunyee Koh, Sungchul Kim, Scott Cohen, Tong Yu, Ryan Rossi, and Razvan Bunescu. 2019b. Figure captioning with reasoning and sequence-level training. *arXiv preprint arXiv:1906.02850*.

Christopher Clark and Santosh Divvala. 2016. Pdffigures 2.0: Mining figures from research papers. In *2016 IEEE/ACM Joint Conference on Digital Libraries (JCDL)*, pages 143–152. IEEE.

Colin B. Clement, Matthew Bierbaum, Kevin P. O'Keeffe, and Alexander A. Alemi. 2019. On the use of arxiv as a dataset.

Siyuan Feng, Olya Kudina, Bence Mark Halpern, and Odette Scharenborg. 2021. Quantifying bias in automatic speech recognition. *arXiv preprint arXiv:2103.15122*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Di Jin, Shuyang Gao, Jiun-Yu Kao, Tagyoung Chung, and Dilek Hakkani-tur. 2020. Mmm: Multi-stage multi-task learning for multi-choice reading comprehension. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8010–8017.

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*, pages 5648–5656.

Samira Ebrahimi Kahou, Vincent Michalski, Adam Atkinson, Ákos Kádár, Adam Trischler, and Yoshua Bengio. 2017. Figureqa: An annotated figure dataset for visual reasoning. *arXiv preprint arXiv:1710.07300*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. In *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics*.

[6] CC 1.0: https://creativecommons.org/publicdomain/zero/1.0/

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Parag Mali, Puneeth Kukkadapu, Mahshad Mahdavi, and Richard Zanibbi. 2020. Scanssd: Scanning single shot detector for mathematical formulas in pdf document images. *arXiv preprint arXiv:2003.08005*.

Behrooz Mansouri, Anurag Agarwal, Douglas Oard, and Richard Zanibbi. 2020. Finding old answers to new math questions: the arqmath lab at clef 2020. In *European Conference on Information Retrieval*, pages 564–571. Springer.

Jason Obeid and Enamul Hoque. 2020. Chart-to-text: Generating natural language descriptions for charts by adapting the transformer model. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 138–147.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Xin Qian, Eunyee Koh, Fan Du, Sungchul Kim, and Joel Chan. 2020. A formative study on designing accurate and natural figure captioning systems. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–8.

Xin Qian, Eunyee Koh, Fan Du, Sungchul Kim, Joel Chan, Ryan A Rossi, Sana Malik, and Tak Yeon Lee. 2021. Generating accurate caption units for figure captioning. In *Proceedings of the Web Conference 2021*, pages 2792–2804.

Elliot Salisbury, Ece Kamar, and Meredith Ringel Morris. 2017. Toward scalable social alt text: Conversational crowdsourcing as a tool for refining vision-to-language technology for the blind. In *Fifth AAAI Conference on Human Computation and Crowdsourcing*.

Noah Siegel, Zachary Horvitz, Roie Levin, Santosh Divvala, and Ali Farhadi. 2016. Figureseer: Parsing result-figures in research papers. In *European Conference on Computer Vision*, pages 664–680. Springer.

Andrea Spreafico and Giuseppe Carenini. 2020. Neural data-driven captioning of time-series line charts. In *Proceedings of the International Conference on Advanced Visual Interfaces*, pages 1–5.

Saiganesh Swaminathan, Thijs Roumen, Robert Kovacs, David Stangl, Stefanie Mueller, and Patrick Baudisch. 2016. Linespace: A sensemaking platform for the blind. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 2175–2185. ACM.

Saiganesh Swaminathan, Conglei Shi, Yvonne Jansen, Pierre Dragicevic, Lora A Oehlberg, and Jean-Daniel Fekete. 2014. Supporting the design and fabrication of physical visualizations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3845–3854. ACM.

Satoshi Tsutsui and David J Crandall. 2017. A data driven approach for compound figure separation using convolutional neural networks. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 533–540. IEEE.

Shaomei Wu, Jeffrey Wieland, Omid Farivar, and Julie Schiller. 2017. Automatic alt-text: Computer-generated image descriptions for blind users on a social network service. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, pages 1180–1192. ACM.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks? *arXiv preprint arXiv:1411.1792*.

Peng W Zhang, Francis Lau, and Chiu-W Sham. 2020. Protograph-based low-density parity-check hadamard codes. *arXiv preprint arXiv:2010.08285*.