Autotator: Semi-Automatic Approach for Accelerating the Chart Image Annotation Process

Junhoe Kim
Seoul National University
Gwanak-gu, Seoul, Republic of Korea
jhkim@hcil.snu.ac.kr

Jinwook Seo
Seoul National University
Gwanak-gu, Seoul, Republic of Korea
jseo@snu.ac.kr

Jaemin Jo
Seoul National University
Gwanak-gu, Seoul, Republic of Korea
jmjo@hcil.snu.ac.kr

ISS'19 Companion,, November 10–13, 2019, Daejeon, Republic of Korea
ACM 978-1-4503-6891-9/19/11.
https://doi.org/10.1145/3343055.3360741

Abstract

Annotating chart images for training machine learning models is tedious and repetitive especially in that chart images often have a large number of visual elements to annotate. We present Autotator, a semi-automatic chart annotation system that automatically provides suggestions for three annotation tasks such as labeling a chart type, annotating bounding boxes, and associating a quantity. We also present a web-based interface that allows users to interact with the suggestions provided by the system. Finally, we demonstrate a use case of our system where an annotator builds a training corpus of bar charts.

Author Keywords

Chart annotation; Data collection; Information extraction; Deep learning; Mixed-initiative interaction.

CCS Concepts

•**Human-centered computing** $\rightarrow$ **Human computer interaction (HCI);** *User interface toolkits; Social tagging systems;* Web-based interaction;

Introduction

Computer vision technologies opened up a new research area in information visualization. Various CNN-based systems have been proposed to extract metadata (e.g., a chart type) [1, 3, 6, 10, 11, 12] or underlying data (e.g., x and y

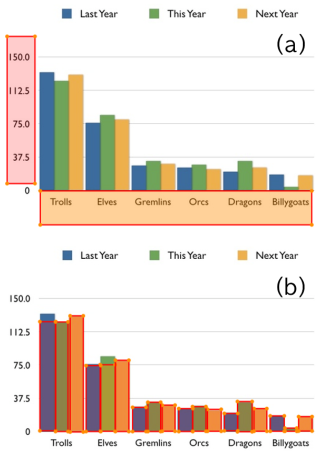

Figure 2: Suggestions (red-bordered boxes) for a bar chart image. Suggested bounding boxes for the X and Y axes (top), and bars (bottom). Note that the suggestions require correction (e.g., the bounding box for the Y-axis should be vertically stretched to include 0.)

coordinates of data points) [1, 3, 13] from chart images. Furthermore, recent advances in computer vision have enabled a new set of interactions between people and visualizations, such as recoloring visualizations or adding interactivity to them. [3, 11]. One of the prerequisites for those intelligent applications is a large annotated dataset for training neural network models, but building a quality dataset requires much human resources. In particular, creating annotations is often more difficult for chart images than natural images, because the chart images have more visual elements to annotate (e.g., axes, labels, and marks) that can even appear repeatedly. As a result, previous studies have used relatively small (~10K) or synthetic datasets [7].

We present Autotator, a semi-automatic chart annotation system that provides suggestions tailored to annotation tasks. We surveyed nine InfoVis papers that used computer vision techniques on chart images and elicited three common tasks. Autotator supports the following three annotation tasks:

T1. **Labelling a chart type** is a task of determining a type of a chart (e.g., *simple*, *stacked*, or *grouped* for a bar chart and *pie* or *donut* for a pie chart) [1, 3, 6, 10, 11, 12, 13].

T2. **Annotating bounding boxes** is a task of drawing a box that includes a specific visual element (e.g., an axis or a bar), if needed, with its label (e.g., X or Y) [1, 2, 10, 11, 12, 13].

T3. **Associating a quantity** is a task of associating a quantity to a chart image (e.g., the number of colors used for bars) [4].

We develop CNN-based predictors for the tasks and train the predictors each time a specific amount (e.g., 100) of new annotated images are collected. As more annotated

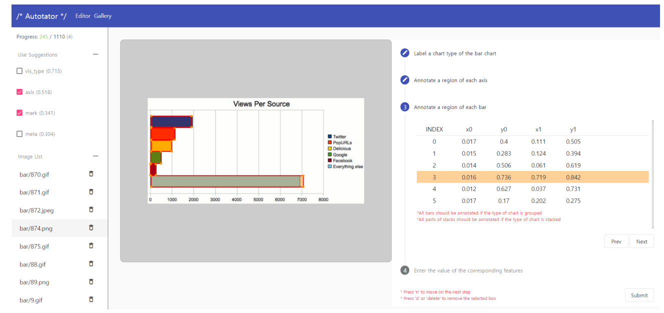

Figure 1: The web-based interface of Autotator. Annotation tasks are displayed on the right side using an accordion interface. An annotator is annotating bounding boxes of bars in a bar chart.

images are provided by human annotators, the predictors are improved to give more accurate suggestions, which may in turn accelerate the annotation process by the human annotators.

Use Case

We present a use case to demonstrate our system, where we assume an annotator, Sara, is annotating a bar chart (see Figure 1). Her first task is to choose the type of the chart among possible choices (*simple*, *grouped*, or *stacked*). Autotator suggests *grouped*, which is correct. She moves on to the next task, just accepting the suggestion. Next, she needs to annotate the X and Y axes of the chart. Autotator gives suggestions for the axes (Figure 2a), but she notices that the bounding box for the Y-axis does not include the bottommost tick label, “0”. She vertically stretches the box by dragging the corner so that the box covers all the tick labels. For the third task, she wants to annotate all bars in the chart. She finds that the predictor correctly detects most of the bars except for a few ones (Figure 2b). To fix the misaligned boxes, she can either 1) redraw a box by selecting

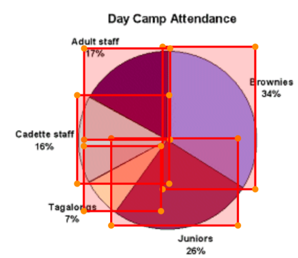

Figure 3: Suggestions (red-bordered boxes) for a pie chart image. Suggested bounding boxes for sectors are mostly aligned.

its the top-left and bottom-right corners or 2) right-click on a pixel in a bar to automatically select a rectangular region of adjacent pixels with similar colors. Then, she confirms that two quantities, the number of colors used for color-encoding and the number of bar groups, are correctly predicted. Finally, she submits the current annotation and moves on to the next chart. Figure 3 demonstrates a similar use case for a pie chart.

Suggesting Annotations

For **T1** and **T3**, we employed existing ResNet-18 pre-trained on ImageNet [5]. For training, we used the softmax cross-entropy loss and the MSE loss for **T1** and **T3**, respectively, using Adam [8] with a learning rate of 1e-3. For **T2**, we employed Faster-RCNN with feature pyramid network [9] pre-trained on COCO train 2017 and tuned the anchor box generator of the network, so that it assigns positive labels to anchors with a more extreme aspect ratio, such as bars in a bar chart. For training, we used the same loss function as the original article using Adam with a learning rate of 1e-4. Each time a specific amount of annotated images are collected, we trained new models in background and switched them with the old ones on the fly.

Conclusion and Future Work

We present Autotator, a semi-automatic chart image annotation system with suggestions provided by CNN-based predictors. Currently, Autotator supports pie and bar charts but can be extended to other charts such as scatterplots. We are also interested in automating the annotation of text labels, for example, through optical character recognition, to broaden the coverage of our system.

REFERENCES

[1] Abhijit Balaji, Thuvaarakkesh Ramanathan, and Venkateshwarlu Sonathi. 2018. Chart-Text: A Fully Automated Chart Image Descriptor. *arXiv preprint arXiv:1812.10636* (2018).

[2] Aaron Baucom and Christopher Echanique. 2013. ScatterScanner: Data extraction and chart restyling of scatterplots. (2013).

[3] Jinho Choi, Sanghun Jung, Deok Gun Park, Jaegul Choo, and Niklas Elmqvist. 2019. Visualizing for the Non-Visual: Enabling the Visually Impaired to Use Visualization. *Computer Graphics Forum* 38, 3 (2019), 249–260. DOI: `http://dx.doi.org/10.1111/cgf.13686`

[4] Daniel Haehn, James Tompkin, and Hanspeter Pfister. 2018. Evaluating Graphical Perception with CNNs. *IEEE transactions on visualization and computer graphics* 25, 1 (2018), 641–650.

[5] 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*. 770–778.

[6] Daekyoung Jung, Wonjae Kim, Hyunjoo Song, Jeong-in Hwang, Bongshin Lee, Bohyoung Kim, and Jinwook Seo. 2017. Chartsense: Interactive data extraction from chart images. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 6706–6717.

[7] 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* (2017).

[8] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[9] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. 2017. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2117–2125.

[10] Jorge Poco and Jeffrey Heer. 2017. Reverse-Engineering Visualizations: Recovering Visual Encodings from Chart Images. In *Computer Graphics Forum*, Vol. 36. Wiley Online Library, 353–363.

[11] Jorge Poco, Angela Mayhua, and Jeffrey Heer. 2017. Extracting and retargeting color mappings from bitmap images of visualizations. *IEEE transactions on visualization and computer graphics* 24, 1 (2017), 637–646.

[12] Manolis Savva, Nicholas Kong, Arti Chhajta, Li Fei-Fei, Maneesh Agrawala, and Jeffrey Heer. 2011. Revision: Automated classification, analysis and redesign of chart images. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. ACM, 393–402.

[13] 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*. Springer, 664–680.