Plot-to-Table Conversion: Fine-Tuning Vision Transformers for Scatter-Plots

Won Park [1]

1. Introduction

The performance of multimodal models, such as GPT-4o, has advanced dramatically. Notably, recent models have demonstrated strong performance in tasks like PlotQA (Methani et al., 2020) and ChartQA (Masry et al., 2022), where they answer questions based on plot images. In 2023, Liu et al. (2023a) proposed DEPLOT, which achieved impressive results by combining a Transformer model for chart-to-table conversion with a large language model (LLM). In 2024, Combined model based on MatCha further set a new state-of-the-art (SOTA) performance on PlotQA (Liu et al., 2023b). However, these models also have limitations. While they are exceptionally effective at converting and answering questions for pie charts, bar charts, and line charts, they struggle significantly with scatterplots, often failing to convert them accurately into tables. Consequently, these models cannot address complex queries such as drawing regression lines or calculating correlations based on scatterplots.

The reason of lower performance of these models on the scatterplots can be explained by bias in the training dataset. Both MatCha and DEPLOT used very few scatterplots in their training and testing datasets. scatterplots were absent from the ChartQA dataset, and the PlotQA dataset consisted predominantly of line charts and bar charts.

This work aims to improve the accuracy of converting scatterplots into tables by proposing a fine-tuned DEPLOT model. Unlike the MatCha model, which focuses on answering various questions about plots, the primary objective of this research is to ensure precise table conversion for scatterplots. Although tasks like PlotQA and ChartQA, which involve answering questions based on plots, can also be addressed using combinations of LLMs as seen in DEPLOT, this study will not explore such applications. Instead, the focus is solely on the scatterplot-to-table conversion task.

To achieve this, we adapted DEPLOT as our baseline model, the state-of-the-art model for the plot-to-table task, specifically for scatterplots. The fine-tuning process used a dataset of 2,500 diverse scatterplot images generated using the `matplotlib` library. This tailored approach is designed to address the unique challenges associated with scatterplot data.

On the test dataset consisting only of scatterplots, the fine-tuned model achieved the highest performance. Compared to the baseline model, it showed a 52.2%p improvement in precision and a 49.5%p improvement in recall. Furthermore, we hypothesize that once combined with an LLM, the fine-tuned model will perform effectively in tasks that involve answering questions based on scatterplots.

2. Related Works

2.1. Scatteract: Automated extraction of data from scatterplots

(Cliche et al., 2017) proposed a model specifically designed for scatterplots. Their approach uses a CNN-based model to extract (x, y) coordinates from scatterplot images and convert them into tables. They used ReInspect method (Stewart et al., 2016) as the object detection method. The model combines GoogleNet CNN layers and LSTM to detect pixel coordinates where points are located. Subsequently, they applied RANSAC regression to map the detected pixel coordinates to plot coordinates, reconstructing the scatterplot data effectively.

2.2. MatCha and DEPLOT

Matcha is a state-of-the-art (SOTA) model for mathematical reasoning tasks involving chart images, capable of generating textual answers. It is based on a Vision Transformer architecture and has achieved SOTA performance on both PlotQA and ChartQA benchmarks.

DePlot fine-tunes the Matcha model to perform the Plot-to-Table task. During its fine-tuning, the question text is fixed as "Generate underlying data table of the figure below:" and the output table is formatted with rows separated by "\n" and columns by "|". For human-written queries, DePlot combined with other large language models (LLMs) such as FlanPaLM or Codex PoT SC outperforms Matcha in generating accurate tables.

The similarity between the generated table and the ground truth is evaluated using the RMS score, a metric proposed by the authors. While DePlot generally performs well on most types of graphs, it exhibits weaknesses when converting scatterplots into tables.

2.3. RMS Scoring Method

Relative Mapping Similarity (RMS) is a metric designed to evaluate the similarity between predicted and target tables. This method treats the tables as unordered collections of mappings rather than sequential data (Liu et al., 2023a). RMS computes pairwise similarities between keys and values in the tables, combining Normalized Levenshtein Distance NL_τ for textual entries and relative distance D_θ for numerical entries. The combined similarity measure, $D_{\tau,\theta}(p,t)$, integrates these two distances to quantify the similarity between entries.

3. Methods

3.1. Challenges and Novel Contributions

We observed that existing models struggled to accurately interpret certain types of charts, particularly scatterplots, when converting them into tables. To address this limitation, we analyzed the shortcomings of these models by conducting case studies with input images and examining their training methods. One key observation from the case studies was that when multiple points shared the same x-coordinate, the models often ignored all but one of these points. This behavior highlighted a significant bias in the datasets used to train models like DEPLOT and MatCha. Most plots in these datasets exhibited functional relationships where each x-value corresponded to a single y-value. In contrast, the models performed poorly on stochastic data or scatterplots, where multiple y-values could exist for the same x-value.

We identified that the most effective way to mitigate this issue was by enhancing the training dataset. While prior research, such as Scatteract (Cliche et al., 2017), utilized 25,000 scatterplot samples, the quality of these datasets was suboptimal. To overcome this limitation, we generated a high-quality dataset of 10,500 scatterplots using the matplotlib library. Of these, 2,000 samples were used for training, 500 for validation, and 100 for testing the fine-tuned model.

This study is significant in that it not only analyzes and addresses the weaknesses of state-of-the-art models for PlotQA and ChartQA tasks but also introduces a model specialized for scatterplot data. This specialization represents a novel contribution to the field of chart-to-table conversion and mathematical reasoning.

3.2. Model construction

Our model adapted pix2struct processor and pix2struct model to train the image to table data. Temperature is set 0 so that the model output is deterministic. Training was begin with adapting pre-trained DEPLOT model. Backpropagation occurs only in the pix2struct model layer. (refer to Figure 1)

3.3. Algorithm and Implementation Details

The proposed method utilizes a fine-tuned DEPLOT model to convert scatterplot images into structured tables. The process is divided into three main stages: preprocessing, model fine-tuning, and test.

Preprocessing Scatterplot images were generated using the `matplotlib.pyplot` library, ensuring a diverse range of scatterplot types, including stochastic data and multiple data points sharing the same x-coordinates. The dataset consists of 10,500 images. Generated dataset is compatible with dataset.Datadict module from Huggingface.

Model Fine-tuning The model was fine-tuned using the training dataset. During fine-tuning, the question text was fixed as *"Generate underlying data table of the figure below:"* to standardize the task. The model was trained to output tables formatted with rows separated by `\n` and columns by `|`.

Test The model's performance is evaluated by F1 score. The score shows the similarity between the generated table and the ground truth.

The implementation was conducted in Python, utilizing PyTorch. The source code and datasets are made publicly available for the reproducibility.

4. Experiments

4.1. Dataset

The dataset was generated using Python's `matplotlib` library. To include diverse types of scatterplots as training data, various properties of scatterplots were modified during the generation process.

Size We used 2,000 image-table pairs for training, 500 for validation, and 100 for test.

Table Characteristics The dataset consists of scatterplots with the following characteristics: randomly distributed scatterplots, scatterplots with weak linear relationships, and scatterplots containing multiple clusters. Some scatterplots have discrete values, while others feature continuous values. On average, each scatterplot contains 16 points, with most having between 0 and 40 points.

The x-axis label and y-axis label were assigned from a set of 50 frequently used labels (e.g., time, length, cost) and their corresponding units (e.g., hours, m, $) drawn from actual datasets. Additionally, 20% of the dataset applied

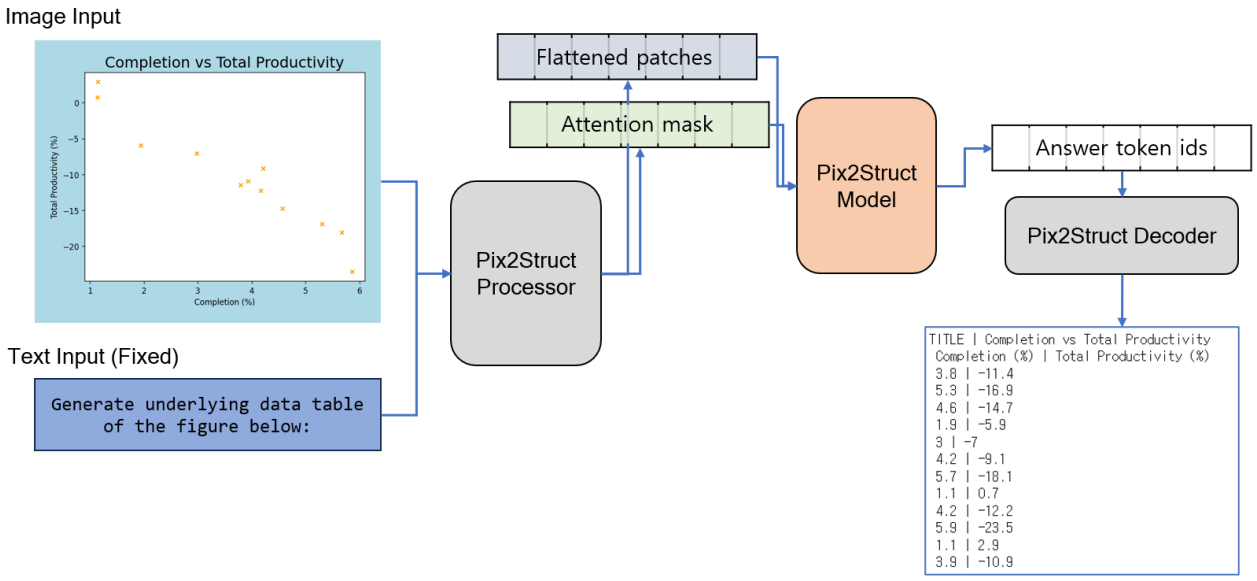

Figure 1. Illustration of the Fine-tuning method. Plot is set as imaged input, and text input is fixed.

completely random strings, such as `xPkqnY1kdh`, as the title, x-label, and y-label.

Chart Characteristics The plot colors were selected from 10 commonly used colors, and 17 distinct plot styles were applied, including circle, x, star, and hollow circle. Approximately 70% of the plots include a grid, while the remaining 30% do not. The resolution (dpi) of the dataset images is distributed as follows: 20% at 50 dpi, 20% at 100 dpi, 40% at 120 dpi, and 20% at 150 dpi.

The output table is formatted with rows separated by `\n` and columns separated by `|`.

Table 1. Scatterplot Data Relationships

Relationship Type	Percentage (%)
Random plot	38.1
Random plot with linearity	52.4
Multiple clusters	9.5

Table 2. Variable Types

Variable Type	Percentage (%)
Discrete	19.1
Continuous	80.9

4.2. Experimental Setup and Design

Computer Resources The experiments were conducted in the Google Colab Pro environment. The model was trained and evaluated on an NVIDIA Tesla A100 GPU, completing the training process in approximately 4 hours. The operating system used was Ubuntu 20.04 LTS, and the total memory available was 80GB. The model was constructed and implemented using the PyTorch framework.

Design The experiment was designed to test the performance of the proposed model on scatterplot-to-table conversion tasks. Training task setup is following:

- **Batch Size**: 12
- **Epochs**: 6
- **Padding**: `"max_length"`
- **Max Length**: 398
- **Train and Validation Metric**: Cross Entropy Loss
- **Weight Decay**: 0.01
- **Gradient Accumulation Steps**: 2
- **FP16**: Enabled
- **Model parameters**: 282M parameters

Cross entropy loss function is adapted for training and validation instead of RMS, since the model needs to compare some tokenized text data on title, x-label, y-label. Training and validation loss per epoch is on the Figure 2. Validation Loss didn't diminished from 5th epochs, so the training was stopped at 6th epoch.

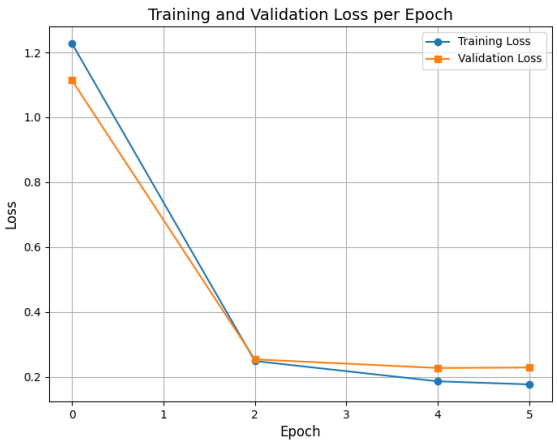

Figure 2. Validation Loss and Training Loss per Epoch

During the test phase, Precision, Recall, and F1 score is adapted to decide accuracy of the model by using. A predicted coordinate was considered correct if it was within a certain Euclidean distance, τ, from the ground truth coordinate.

4.3. Quantitative Results

Precision and Recall was calculated during the test phase. Distance τ is set to 0.5. The test results showed significantly improved performance compared to the baseline. Recall increased from 19.5% to 69.0%, Precision improved from 30.8% to 83.0%, and the F1 Score rose from 22.9% to 74.9%.

DEPLOT Model	F1 Score (%)	Precision (%)	Recall (%)
Baseline (SOTA)	22.9	30.8	19.5
Fine-tuned	**74.9**	**83.0**	**69.0**

Table 3. Performance Comparison of Baseline and Fine-tuned DEPLOT

4.4. Qualitative Results

Visualized qualitative result is on Figure 3, the comparison of GPT-4o, baseline DEPLOT, and fine-tuned DEPLOT. The plot used for qualitative analysis is not randomly generated but is scatterplots representing real-world.

5. Conclusion

Our method was largely successful. Our methods improved scatterplot to table task compared to GPT-4o and DEPLOT baseline model. The model performed well on scatterplots because the DEPLOT pre-trained model maps plot information such as labels, titles, legends, and axis details more effectively than other transformer models.

On the other hand, this model also has limitations. The fine-tuned model cannot handle line charts and bar charts as effectively as DEPLOT. While it is specialized for scatterplot conversion, it exhibits weaker generalization performance. Another limitation is its lack of ability to generate plots with very small values. Since the dataset was generated with rounding applied from the second decimal place, the model struggles to accurately convert chart data with units smaller than 0.01. These weaknesses could potentially be addressed by enhancing the dataset. Moreover, it is necessary to validate the model using real-world datasets to determine if overfitting has occurred.

Further work is needed to generalize these improvements to other mathematical domains. The proposed method could be applied to mathematical problem-solving tasks through integration with large language models (LLMs). As a direction for future research, we suggest investigating which LLMs would be the most effective for this integration.

References

Cliche, M., Rosenberg, D., Madeka, D., and Yee, C. *Scatteract: Automated Extraction of Data from Scatter Plots*, pp. 135–150. Springer International Publishing, 2017. ISBN 9783319712499. doi: 10.1007/978-3-319-71249-9_9. URL http://dx.doi.org/10.1007/978-3-319-71249-9_9.

Liu, F., Eisenschlos, J. M., Piccinno, F., Krichene, S., Pang, C., Lee, K., Joshi, M., Chen, W., Collier, N., and Altun, Y. Deplot: One-shot visual language reasoning by plot-to-table translation, 2023a. URL https://arxiv.org/abs/2212.10505.

Liu, F., Piccinno, F., Krichene, S., Pang, C., Lee, K., Joshi, M., Altun, Y., Collier, N., and Eisenschlos, J. M. Matcha: Enhancing visual language pretraining with math reasoning and chart derendering, 2023b. URL https://arxiv.org/abs/2212.09662.

Masry, A., Long, D. X., Tan, J. Q., Joty, S., and Hoque, E. Chartqa: A benchmark for question answering about charts with visual and logical reasoning, 2022. URL https://arxiv.org/abs/2203.10244.

Methani, N., Ganguly, P., Khapra, M. M., and Kumar, P.

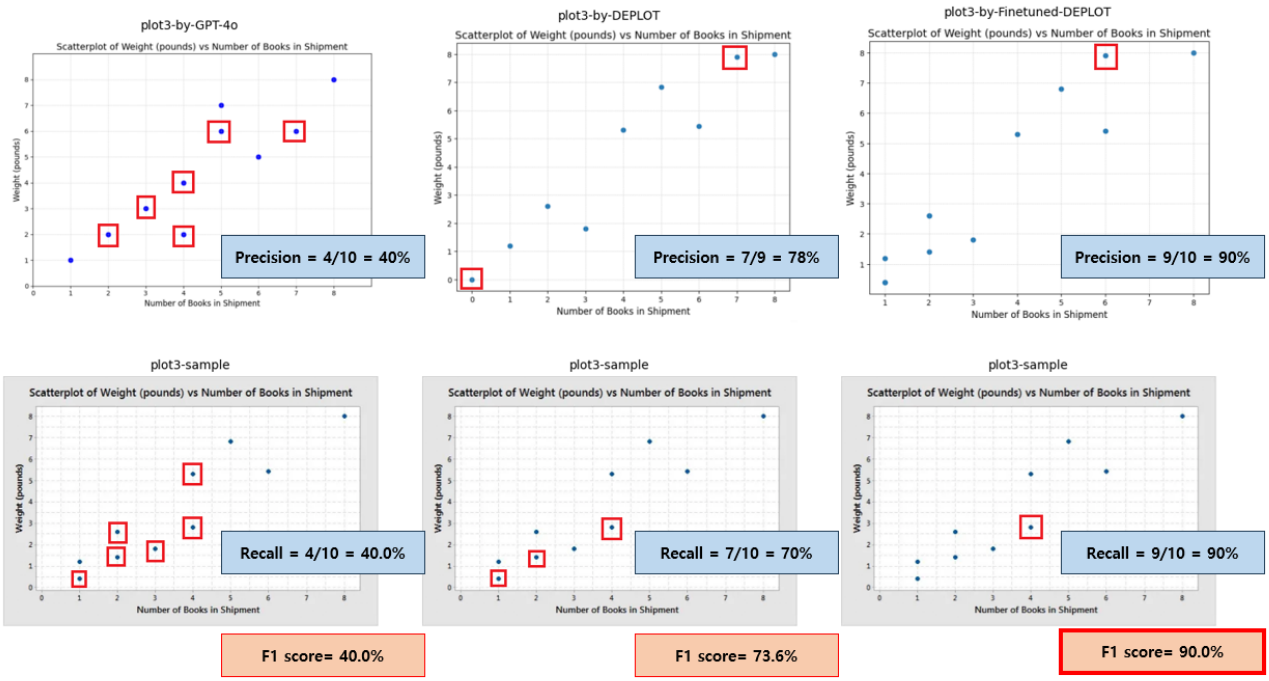

Figure 3. Each graph visualizes tables generated by GPT-4o, DEPLOT, and the fine-tuned DEPLOT. Incorrect points are highlighted with red boxes. fine-tuned DEPLOT (90%) shows better performance than the DEPLOT (73.6%) and GPT-4o (40%) on this scatterplot sample.

Plotqa: Reasoning over scientific plots, 2020. URL https://arxiv.org/abs/1909.00997.

Stewart, R., Andriluka, M., and Ng, A. Y. End-to-end people detection in crowded scenes. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2325–2333, 2016. doi: 10.1109/CVPR.2016.255.

Appendix

Generated Dataset is available on the following URL address:

https://drive.google.com/drive/folders/1L-ammRM4Xd˙N-b7Bbb4NzXnefnH2TMFu?usp=sharing

Fine-tuned model is available on the following URL address: https://drive.google.com/drive/folders/19Z1eESBlPtuy8-PYoRca9nezZdjQdAct?usp=sharing