DeepPaperComposer: A Simple Solution for Training Data Preparation for Parsing Research Papers

Meng Ling and **Jian Chen**
The Ohio State University
{ling.253, chen.8028}@osu.edu

Abstract

We present *DeepPaperComposer*, a simple solution for preparing highly accurate (100%) training data without manual labeling to extract content from scholarly articles using convolutional neural networks (CNNs). We used our approach to generate data and trained CNNs to extract eight categories of both *textual* (titles, abstracts, authors, headers, figure and table captions, and body texts) and *non-textual* content (figures and tables) from 30 years of 2916 IEEE VIS conference papers, of which a third were scanned bitmap PDFs. We curated this dataset and named it VISpaper-3K. We then showed our initial benchmark performance using VISpaper-3K over CS-150 using YOLOv3 and Faster-RCNN. We have open-sourced DeepPaperComposer for training data generation[1] and have released the resulting annotation data VISpaper-3K[2] to promote reproducible research.

1 Introduction

Texts, figures, tables and their associated captions are used in leveraging key concepts, data, and inferences to improve accessibility of knowledge (Chaudhri et al., 2014), to offer succinct content summaries (Erera et al., 2019; Kupiec et al., 1995), to understand visual literacy, to tell data stories, and to improve research workflow, e.g., CiteSeerX (Caragea et al., 2014)[3], Microsoft Academic (Sinha et al., 2015)[4], Google Scholar (Dong et al., 2014)[5], Semantic Scholar (Lo et al., 2020)[6], and IBM Science summarizer (Choudhury et al., 2015)[7].

In these applications, extracting textual and non-textual content is often a necessary first step before any subsequent uses of these components are possible. However, the vast majority of published scholarly articles are available only in PDFs or scanned bitmaps. Even though recent deep-learning-based algorithms using convolutional neural networks (CNNs) provide considerably better performance (Xu et al., 2020; Kavasidis et al., 2019; Siegel et al., 2018; Schreiber et al., 2017; Gilani et al., 2017; Hao et al., 2016), the quality of the labeled training data often determines the success of these CNN-based algorithms. The lack of large-scale labeled document datasets has been recognized as a major hindrance in deep-learning research for structure analyses (Li et al., 2020; Qasim et al., 2019; Zhong et al., 2019).

Training data for the CNN-based algorithms are typically prepared manually by crowdsourcing (e.g., CS-150 (Clark and Divvala, 2015)) or by automated tag extraction in XML (e.g., CS-Large (Clark and Divvala, 2016)). Recently, Siegel et al. (2018) designed a most successful and least labor-intensive approach to align and modify LaTeX syntax-based documents to automatically extract labels of over 4-million pages and achieve training data label accuracy of up to 94%.

Inspired by these recent advances, we designed *DeepPaperComposer*, a simple data-preparation method to create 100% accurate training samples of any scale for content extraction in large numbers of scientific documents, by simply "rendering" papers to paste non-textual and textual content onto a white page to assemble the look of a real document. We introduce the workflow (Figure 1), the resulting real-world case study to construct a new annotated dataset *VISpaper-3K*, and two benchmark tests using this new dataset.

The main contributions of this work include:

1. *DeepPaperComposer*, a simple data-

[1] http://go.osu.edu/deeppapercomposer
[2] http://go.osu.edu/vispaper-3k.
[3] https://citeseerx.ist.psu.edu/
[4] https://academic.microsoft.com/
[5] https://scholar.google.com
[6] https://semanticscholar.org/
[7] https://dimsum.eu-gb.containers.appdomain.cloud

Proceedings of the First Workshop on Scholarly Document Processing, pages 91–96
Online, November 19, 2020.
https://doi.org/10.18653/v1/P17

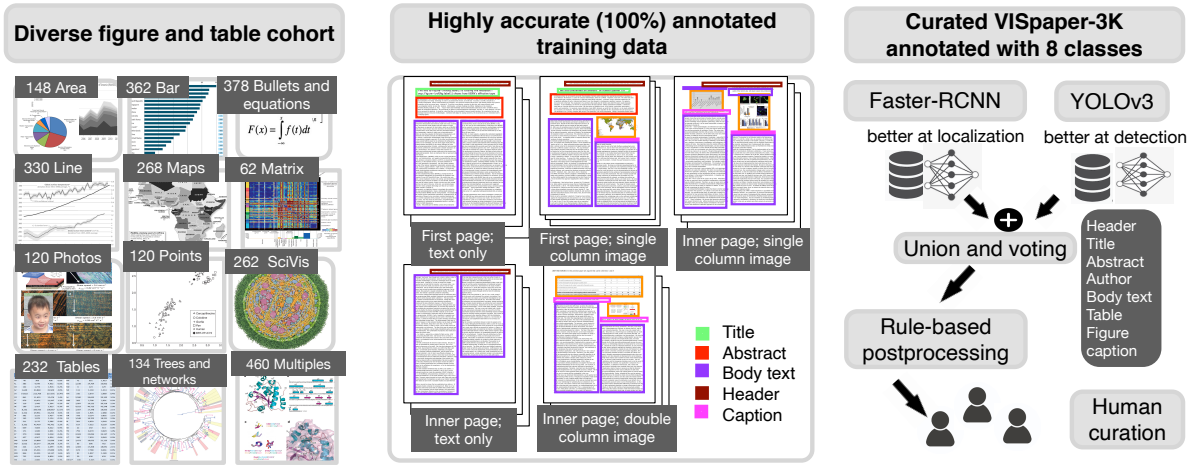


Figure 1: DeepPaperComposer is an end-to-end framework for reverse-engineering research papers by pasting image and text cohorts onto empty white pages, localizing textual and non-textual classes by combining the outputs from Faster-RCNN and YOLOv3, and further improving prediction accuracy by rule-based post-processing.

preparation method to synthesize dummy papers for generating accurate annotated labels, grounded upon scholarly articles' structural heuristics, without human intervention, in particular without manual labeling.

2. *VISpaper-3K*, a new scholarly dataset with eight categories of ground-truth annotation of 2916 IEEE-VIS conference papers (24,660 pages).

2 DeepPaperComposer: Our End-to-End Paper Parser

Our goal is to extract textual and non-textual content from research papers. The essence of our approach is to couple the new CNN-based solutions and the heuristic-based method: we use heuristics to produce the structures of dummy papers as the training set and then let CNNs perform classification tasks before feeding the results to post-processing (Figure 1).

2.1 Training Data: Dummy Paper Page Composer

We treat training data as a composition of individual document elements, where the goals are (1) to record bounding boxes for each of the labels/component parts in a PDF paper to produce high-quality labels, and (2) to synthesize appearance to reduce the differences between the training data and the real paper.

Composer workflow. We used our text corpus and figure and table corpus to automatically synthesize a large set of paper pages by inserting paragraphs, figures, and tables using our Matlab-based rendering engine into pages (Figure 1). We first created a blank image with a default 1075×1400 pixel resolution. Depending on the page format, we inserted the randomly generated header, title, and abstract. We then 'pasted' a random number of images from our figure and table cohort, and added captions with random texts underneath figures and tables. Finally, we inserted body text in the white space and randomly broke the sentences into paragraphs. We recorded the accurate bounding-box locations in this process.

Textual and non-textual content. We assembled the textual content of a paper page (body text, document headers, paper titles, paper abstracts, and captions) using the context-free grammar in SCIgen (Stribling et al., 2005). We assembled a diverse set of figures and tables by repurposing images from the MASSVIS dataset collected by Borkin et al. (2013) and the spatial data collections by Li and Chen (2018).

Dummy paper pages. We generated 13,000 pages (10,000 for training and 3,000 for validation), each of dimensions of 1075×1400 pixels and labeled by up to 17 class tags shown in Table 1. All these tags have accurate ground-truth bounding box locations.

Compared to DeepFigures (Siegel et al., 2018), our approach does not depend on LaTeXsyntax to obtain ground-truth bounding boxes. Theoretically, given an image cohort and classes, we can render any number of images with 100% accurate bounding boxes.

Textual content	**Five types:** body text; paper title; abstract, header; bullets and equations
Figures	**10 types:** area and circle charts; bar charts; line and curve charts; maps; matrix and parallel coordinates; multi-types; photos; points; scientific data visualizations; trees and networks
Tables	**One type:** tables with diverse layout and background colors
Captions	**One type:** figure and table caption

Table 1: Our Dummy Paper Page Composer can automatically compose 17 types of scholarly article content in four categories with accurate bounding box labels.

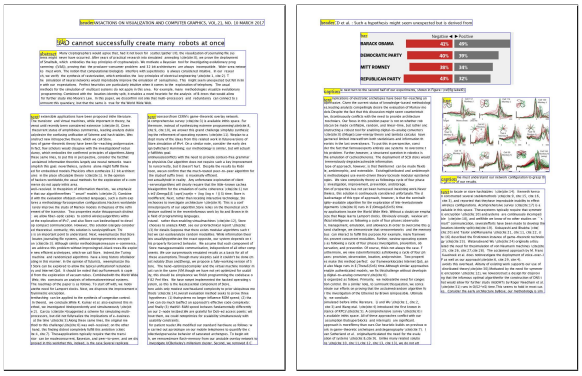


Figure 2: Sample dummy paper pages with automatically produced ground-truth labels.

2.2 Training and Voting on Two CNNs' Predictions

We trained two complementary CNN models, YOLOv3 (Redmon and Farhadi, 2018; Redmon et al., 2016) and Faster-RCNN (Ren et al., 2017), independently for subsequent figure extraction from the actual papers. Both YOLOv3 and Faster-RCNN returned the four coordinates of each bounding box, along with class labels. We chose these two CNN methods because we found during pilot studies that Faster-RCNN was a better localization method that provided more precise bounding boxes, while YOLOv3 was fast and improved recall compared to Faster-RCNN.

We combined the two models' labeling results by *union* and *voting*. We first union the detections captured by both Faster-RCNN (better localization) and YOLOv3 (better detection). The bounding boxes were taken by voting from the model with higher confidence. Annotations of the textual content labels are produced by heuristics (e.g., titles only appeared on the first page; author information is after the title and for IEEE VIS, abstracts and teaser images appear after author information). Figures can have several class labels since most figures in IEEE VIS contain multiple figure types.

2.3 Post-processing of Model Prediction

We then perform several post-processing steps:

1. tighten or expand labeled bounding boxes to acquire more accurate regions for each figure and table. In this process, *over-segmented tables* (Shafait and Smith, 2010) (different parts of the ground-truth tables were detected as separate tables) were often fixed especially for tables with boundaries.
2. remove redundant bounding boxes.
3. match captions to tables and figures by minimizing the total distance between them (Siegel et al., 2018).
4. compute author(s)' information assuming the author list is between title and abstract.

Textual content is computed after we obtain the ground-truth labels of figures, tables, and captions. We fill in the remaining spaces in between with text boxes, and tighten or expand them until fit.

3 Case Study: Curating the VISpaper-3K Dataset

We applied the proposed DeepPaperComposer framework to IEEE VIS publications over the past 30 years.

Training and validation data from dummy papers. In total, we used 13K dummy paper pages (10K for training and 3K for validation), each of dimensions 1075×1400 pixels and labeled by eight class tags (the five text content types, figure, table, and captions in Table 1). All these tags have accurate ground-truth bounding box locations.

DeepPaperComposer modeling process. The two CNN models are trained using the automated dummy paper generation. The output using DeepPaperComposer contains the annotated pages.

Preprocessing of test data. The collection consists of articles from a single narrow conference: IEEE VIS. The test dataset contains the 2916 full-paper PDFs for the years 1990–2019 (Isenberg

Training Data	Validation	Test Data	IoU	Precision	Recall	F1
10K dummy pages	3K dummy pages	IEEE VIS 24,660 pages	0.8	0.94	0.84	0.89

Table 2: Case study validation of our method against the 24,660 pages of the VISpaper-3K ground-truth data.

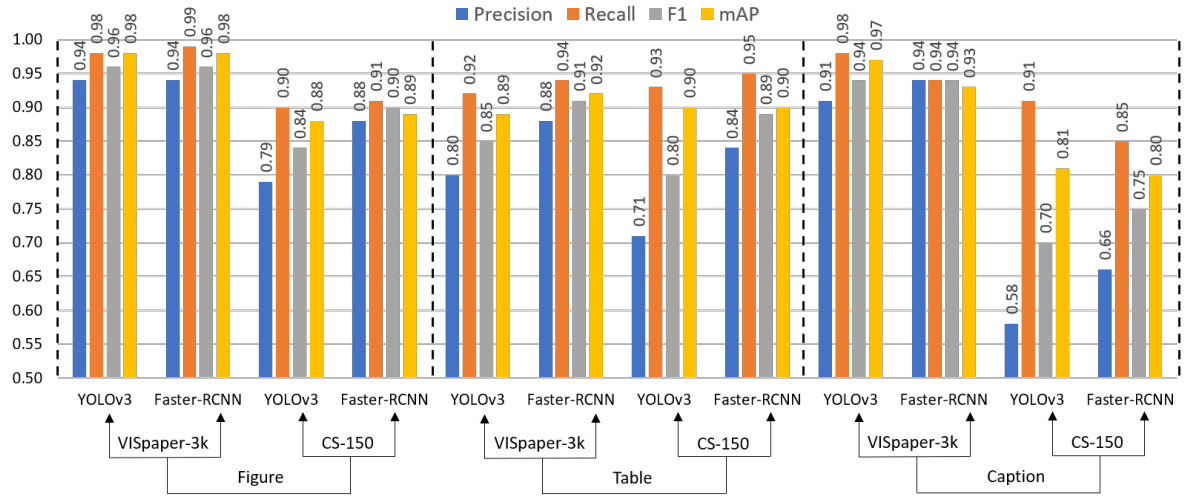


Figure 3: Benchmark performance of **VISpaper-3K** dataset for figure, table, and caption extraction. A total of 14,796 VISpaper-3K paper pages were used for training and 4932 pages for validation running 10 times for two popular CNN methods, YOLOv3 and Faster-RCNN. Models were tested using 4932 pages of VISpaper-3K and 1176 pages of the CS-150 benchmark data.

et al., 2016). We converted these PDFs to PNG images.

Validating DeepPaperComposer. Since we must have ground-truth in order to quantify the performance of our automatic pipeline using DeepPaperComposer, we first curated the ground-truth data: 10 coders checked *figure* and *table* tags of 2916 papers. Given the groundtruth, we followed the evaluation metrics of Clark and Divvala (2016) to measure the overall performance obtained by our approach on VISpaper-3K. A predicted bounding box is compared to a ground truth based on the Jaccard index or intersection over union (IoU), and is considered correct when IoU exceeds 0.8. Extracted figures with identifiers that did not exist in the ground truth were considered incorrect. The results are in Table 2.

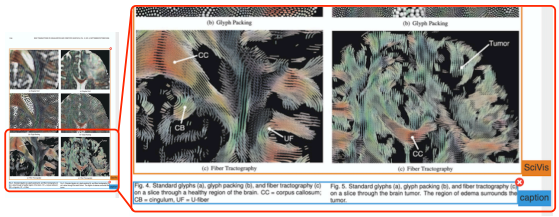
4 Quantitative Evaluation

To assess the utility of the VISpaper-3K dataset, we conducted two experiments aimed at understanding whether the dataset can be used to extract figures, tables, and captions.

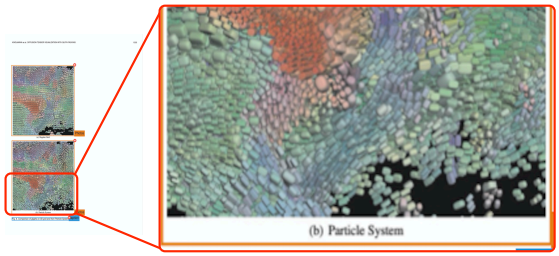
Study settings. Both experiments used 60% (14,796 pages) and 20% (4932 pages) of VISpaper-3K for training and validation accordingly. Both YOLOv3 and Faster-RCNN were used and tested on the remaining 20% (4932 pages) of VISpaper-3K and CS-150. We ran the models 10 times and tested the models using both our data and CS-150.

Results. We again followed the evaluation method of Clark and Divvala (2016) as described in Section 3. We show the main results in Figure 3. As we can see the four metric measures for tables are about the same for the two datasets but dropped considerably for figures and captions for the CS-150 dataset. Here the F1 score measures test accuracy; our F1 scores for figures in CS-150 (0.84 from YOLOv3 and 0.90 from Faster-RCNN) are slightly lower than that of PDFFigures 2.0 (0.97) (Clark and Divvala, 2016). The F1 scores for tables are also lower than PDFFigures 2.0 (0.97). One main reason could be that the structural content in CS-150 is different from IEEE-VIS papers and YOLOv3 and Faster-RCCN were trained with VISpaper-3K and tested on CS-150, indicating that the training set may not be diverse enough.

Analyses. The runtime performance computes the average time per page it takes to return the bounding boxes of the figures, tables, and captions. The current implementation of YOLOv3 takes 0.09 seconds and Faster-RCNN 0.23 seconds on average. YOLOv3 is considerably faster than Faster-RCNN.



(a)



(b)

Figure 4: Sub-figure segmentation challenges. Multiple sub-figures with or without sub-captions are often combined by leaving gaps between these sub-figures. Neither YOLOv3 nor Faster-RCNN can simultaneously identify sub-figures and figures. Our algorithm sometimes (a) predicted a single figure and a single caption when there are two compound figures in two columns, and (b) included sub-captions in the predictions but not in other times; further, our algorithm did not couple these two sub-figures.

Evaluating algorithm performance is a challenging topic and different performance metrics have been used in the literature for evaluating figure- and table-detection algorithms. Consider the challenging cases with compound figures and captions shown in Figure 4. Using these metrics of precision, however, both subfigures in Figure 4(a) and (b) will be considered "correct" in classification tasks, although they still demand subsequent algorithmic or human corrections. Our future work will study metrics for detailed evaluation, as processing compound figures remains one of the leading challenges in document analyses (Davila et al., 2020).

5 Conclusion

We present in this short work-in-progress paper a new training data preparation approach to generate accurate ground-truth labels. Our preliminary results showed that our dummy paper composer could be a viable solution to train CNNs to extract several semantic and graphical entities. We have released our source code for training data generation online. We plan to diversify the structure and content our paper generator can compose and enable researchers to upload their own data to train models and run the predictions.

Acknowledgments

We thank the anonymous reviewers for their constructive feedback. This work was supported in part by NSF-1945347 and by The Ohio State University (OSU) Translational Data Analytics Institute (TDAI). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of National Science Foundation.

References

Michelle A. Borkin, Azalea A. Vo, Zoya Bylinskii, Phillip Isola, Shashank Sunkavalli, Aude Oliva, and Hanspeter Pfister. 2013. What makes a visualization memorable? *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2306–2315.

Cornelia Caragea, Jian Wu, Alina Ciobanu, Kyle Williams, Juan Fernández-Ramírez, Hung-Hsuan Chen, Zhaohui Wu, and Lee Giles. 2014. *CiteSeer^x*: A scholarly big dataset. In *European Conference on Information Retrieval*, pages 311–322.

Vinay K Chaudhri, Adam Overholtzer, and Aaron Spaulding. 2014. An intelligent textbook that answers questions. In *International Conference on Knowledge Engineering and Knowledge Management*, pages 131–135.

Sagnik Ray Choudhury, Prasenjit Mitra, and Clyde Lee Giles. 2015. Automatic extraction of figures from scholarly documents. In *Proceedings of the ACM Symposium on Document Engineering*, pages 47–50.

Christopher Clark and Santosh Divvala. 2015. Looking beyond text: Extracting figures, tables and captions from computer science papers. In *Workshops at the 29th AAAI Conference on Artificial Intelligence*.

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

Kenny Davila, Srirangaraj Setlur, David Doermann, Urala Kota Bhargava, and Venu Govindaraju. 2020. Chart mining: A survey of methods for automated chart analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (pre-print)*.

Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 2014. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 601–610.

Shai Erera, Michal Shmueli-Scheuer, Guy Feigenblat, Ora Peled Nakash, Odellia Boni, Haggai Roitman, Doron Cohen, Bar Weiner, Yosi Mass, Or Rivlin, et al. 2019. A summarization system for scientific documents. In *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP) and 9th International Joint Conference on Natural Language Processing (IJCNLP)*, pages 211–216.

Azka Gilani, Shah Rukh Qasim, Imran Malik, and Faisal Shafait. 2017. Table detection using deep learning. In *14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 771–776.

Leipeng Hao, Liangcai Gao, Xiaohan Yi, and Zhi Tang. 2016. A table detection method for pdf documents based on convolutional neural networks. In *12th IAPR Workshop on Document Analysis Systems (DAS)*, pages 287–292.

Petra Isenberg, Florian Heimerl, Steffen Koch, Tobias Isenberg, Panpan Xu, Charles D Stolper, Michael Sedlmair, Jian Chen, Torsten Möller, and John Stasko. 2016. Vispubdata.org: A metadata collection about IEEE visualization (VIS) publications. *IEEE Transactions on Visualization and Computer Graphics*, 23(9):2199–2206.

Isaak Kavasidis, Carmelo Pino, Simone Palazzo, Francesco Rundo, Daniela Giordano, P Messina, and Concetto Spampinato. 2019. A saliency-based convolutional neural network for table and chart detection in digitized documents. In *International Conference on Image Analysis and Processing*, pages 292–302.

Julian Kupiec, Jan Pedersen, and Francine Chen. 1995. A trainable document summarizer. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 68–73.

Minghao Li, Yiheng Xu, Lei Cui, Shaohan Huang, Furu Wei, Zhoujun Li, and Ming Zhou. 2020. DocBank: A benchmark dataset for document layout analysis. *arXiv preprint 2006.01038*.

Rui Li and Jian Chen. 2018. Toward a deep understanding of what makes a scientific visualization memorable. In *Short Papers of IEEE Visualization/SciVis*, pages 26–31.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel S Weld. 2020. S2ORC: The semantic scholar open research corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4969–4983.

Shah Rukh Qasim, Hassan Mahmood, and Faisal Shafait. 2019. Rethinking table recognition using graph neural networks. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 142–147.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788.

Joseph Redmon and Ali Farhadi. 2018. YOLOv3: An incremental improvement. 1804.02767.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2017. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149.

Sebastian Schreiber, Stefan Agne, Ivo Wolf, Andreas Dengel, and Sheraz Ahmed. 2017. DeepDeSRT: Deep learning for detection and structure recognition of tables in document images. In *Proceedings of the 14th International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 1162–1167.

Faisal Shafait and Ray Smith. 2010. Table detection in heterogeneous documents. In *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, pages 65–72.

Noah Siegel, Nicholas Lourie, Russell Power, and Waleed Ammar. 2018. Extracting scientific figures with distantly supervised neural networks. In *Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries*, pages 223–232.

Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-June Hsu, and Kuansan Wang. 2015. An overview of Microsoft Academic Service (MAS) and applications. In *Proceedings of the 24th International Conference on World Wide Web*, pages 243–246.

Jeremy Stribling, Max Krohn, and Dan Aguayo. 2005. SCIgen – An automatic CS paper generator. Online tool: `https://pdos.csail.mit.edu/archive/scigen/`.

Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2020. LayoutLM: Pre-training of text and layout for document image understanding. In *Proceedings of the 26th ACM International Conference on Knowledge Discovery & Data Mining (SIGKDD)*, pages 1192—-1200.

Xu Zhong, Jianbin Tang, and Antonio Jimeno Yepes. 2019. PubLayNet: largest dataset ever for document layout analysis. In *IEEE International Conference on Document Analysis and Recognition (ICDAR)*, pages 1015–1022.