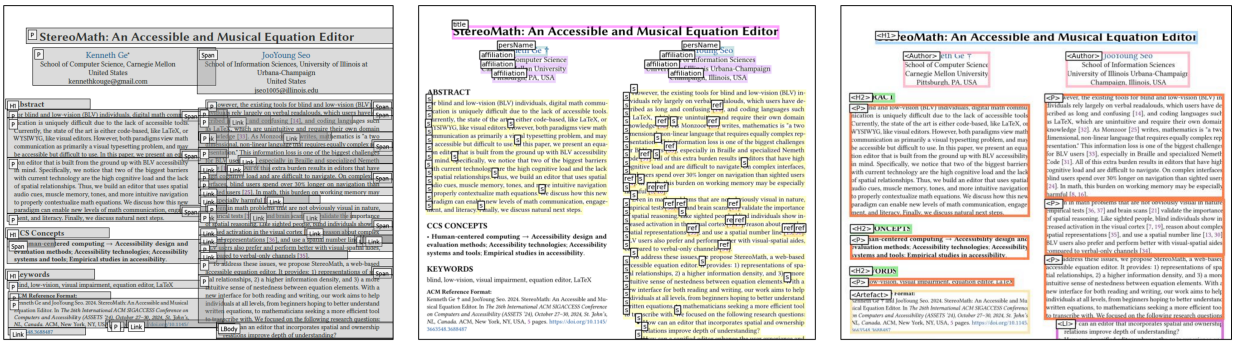
We Write Our Research Papers in WYSIWYM. Why Do We Tag Our PDFs in WYSIWYG?

Peya Mowar
pmowar@cs.cmu.edu
Carnegie Mellon University
Pittsburgh, PA, USA

Aaron Steinfeld
steinfeld@cmu.edu
Carnegie Mellon University
Pittsburgh, PA, USA

Jeffrey P. Bigham
jbigham@cs.cmu.edu
Carnegie Mellon University
Pittsburgh, PA, USA



(a) Adobe Acrobat Auto-Tagging

(b) GROBID Metadata Generation (used in SciA11y and Paper to HTML)

(c) WYSIWYM Tagging (Ours)

Figure 1: Metadata generated by (a) Adobe Acrobat, (b) the GROBID machine learning pipeline (used in SciA11y [16] and Paper to HTML [15]), and (c) our WYSIWYM (What You See Is What You Mean) tagging approach. Both Acrobat and GROBID rely solely on visual features of the PDF, *i.e.,* adopting a WYSIWYG (What You See Is What You Get) approach, producing cluttered or inaccurate metadata. In contrast, our WYSIWYM tags are refined using additional context from source documents (e.g., LaTeX files), resulting in cleaner and more semantically accurate results.

ABSTRACT

Most academic research is ultimately disseminated through documents in the PDF format. This format has advantages in flexibility and portability, but presents challenges for accessibility that have stubbornly resisted solutions despite decades of attempts. Tagging PDFs is hard to automate because tags are currently generated visually, not semantically, which makes the output cluttered and manual correction tedious and error-prone. Ironically, this semantic structure already exists during authoring, especially in WYSIWYM authoring tools like LaTeX, but is discarded during PDF rendering. This raises an obvious question, can we use this lost semantic information to better automate tagging in PDFs? In this paper, we refine generated metadata using the semantics in the source documents of research papers. We demonstrate that the metadata generated by our method already surpasses what authors currently submit to ACM ASSETS on many criteria. Our approach represents a concrete step toward finally automating tagging and reading order in PDFs, directly improving the accessibility of our research papers.

ACM Reference Format:
Peya Mowar, Aaron Steinfeld, and Jeffrey P. Bigham. 2025. We Write Our Research Papers in WYSIWYM. Why Do We Tag Our PDFs in WYSIWYG?. In *The 27th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '25), October 26–29, 2025, Denver, CO, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3663547.3759730

1 INTRODUCTION

Most academic research is disseminated through documents in the Portable Document Format (PDF), primarily because this format preserves formatting across many different platforms. However, persistent challenges remain in ensuring that documents prepared in this format consistently include the metadata necessary to make them accessible to people with disabilities [2, 3]. The PDF format is used almost exclusively in the academic community (even at ACM ASSETS), despite its shortcomings in accessibility.

Making a PDF accessible requires three primary kinds of metadata, which directly translate to how assistive technology conveys the content: *(i)* tagging individual sections of the PDF content according to their type and location on the

ASSETS '25, October 26–29, 2025, Denver, CO, USA

ACM ISBN 979-8-4007-0676-9/2025/10
https://doi.org/10.1145/3663547.3759730

page (*e.g.*, heading, paragraph, figure, etc.), *(ii)* defining an appropriate reading order, and *(iii)* providing content-specific metadata (*e.g.*, alternative text description for images or structured information for tables and lists). In this paper, we focus on the first two categories of metadata (tags and reading order), primarily because these are most tractable to be "solved" by machine learning and yet even PDFs labeled by authors still show significant problems on these dimensions.

The tools available for providing this necessary metadata (often called accessibility remediation) are tedious to use and it is difficult to know if you have made your PDF accessible correctly. Although a number of tools exist for PDF remediation [1, 4, 12], they mostly work in the same way – the tool visualizes each page of the PDF document including any existing metadata information and the user must manually provide (or fix) the metadata. Because the metadata is only visible via specialized remediation tools and assistive technologies, it often goes unchecked or even accidentally removed during various processing steps during publication.

In this paper, we leverage the multiple representations of papers that are naturally created during the authoring process to automatically obtain this metadata. In particular, we use both the source document of the paper (LaTeX) and its visual rendering (PDF) to create highly accurate tags in the correct reading order. First, we run an object detection model on the visual rendering, and an optical character recognition on each segment to determine what text it contains. We then use the text in each segment as an index into the source document, which we use to find related metadata (*e.g.*, section heading levels), and to influence the reading order where visual layout is unclear (common in two column formats).

We believe our method is starting to approach the quality level needed to perform tagging and reading order remediation automatically. While we advise being appropriately cautious, we are not the first to recognize that AI may be starting to get to the point where it may obviate the need for human intervention and that this may allow us to finally solve long-standing problems in accessibility [14]. In the particular instance of PDF tagging and reading order, we believe this is likely to become nearly solved because the information necessary to do this well is already available when producing PDFs, it just exists in disparate sources.

2 RELATED WORK

Making academic paper PDFs accessible has remained a persistent challenge over the past decade [3, 7]. This is partly due to the human challenges in making research papers accessible, and partly due to the poor usability of existing PDF remediation tools.

Human Challenges in Creating Accessible Academic Papers. As an academic community, we have adopted PDF as the standard format for publishing research due to its consistency and portability, yet this choice significantly undermines accessibility. The burden of PDF accessibility is often placed on the authors, who are expected to use lacking (and often expensive!) tools to navigate the complex and time-consuming process of remediation [9], making it difficult for them to do a good job (if at all). Prior work explored various strategies including requesting, mandating, or distributing the responsibility among authors and other stakeholders in the publishing process (*e.g.*, publishers, volunteers, paid agencies) [2]. Still, tired of waiting for better tools, they found getting rid of the PDF format was the only viable solution. Consequently, SciA11y [16] and Paper to HTML [15] convert PDFs, while ar5iv [5] and the ACM Digital Library convert source documents, into an accessible HTML format. Yet, the uptake of alternative formats in scholarly communication remains limited; even ASSETS still requires PDF submissions for review.

Tooling for PDF Accessibility Remediation. Despite the availability of many tools [1, 4], authors still struggle to make PDFs accessible. Even newer tools designed to simplify the process come with a steep learning curve [12]. The challenge lies in the nature of PDFs themselves: they are based on visual formatting, which makes it hard to extract their underlying structure. Thus, automated remediation tools often create incorrect tag hierarchies, requiring human review and correction. These human-in-the-loop workflows still rely on visual tagging, making them inaccessible to blind authors [9]. Given these challenges in retroactively fixing PDFs, the LaTeX tagged PDF [10] project aims to automatically produce tagged PDFs directly from LaTeX source files. While this is a promising long-term solution, the authors document years of intensive engineering effort to make it viable. In contrast, we argue that source files can be leveraged for PDF remediation today, and our work is the first to explore this approach.

3 METHODOLOGY

Our approach (see Figure 2) processes the PDF visually to generate content regions, refines them by taking additional semantic context from the source documents, and ultimately produces content tags and a coherent reading order.

3.1 Visual Rendering (PDF) Processing

Identifying Content Regions. We use the Ultralytics YOLOv11 model [6] to identify content regions and their respective types in page images extracted from PDFs. We perform inference on a fine-tuned version[1] of the model on the DocLayNet dataset [11] to obtain content region bounding boxes, labels, and confidence scores for each page of the PDF.

Extracting Text from Regions. We extract the text contained on the page using PyTesseract v0.3.13[2], a wrapper for Google's Tesseract OCR Engine [13]. PyTesseract returns word-level bounding boxes for all detected text on the page, along with the corresponding recognized text. We map the word-level bounding boxes to the corresponding predicted content region, and aggregate the recognized text within each region. If textual

[1] https://github.com/moured/YOLOv11-Document-Layout-Analysis
[2] https://pypi.org/project/pytesseract/

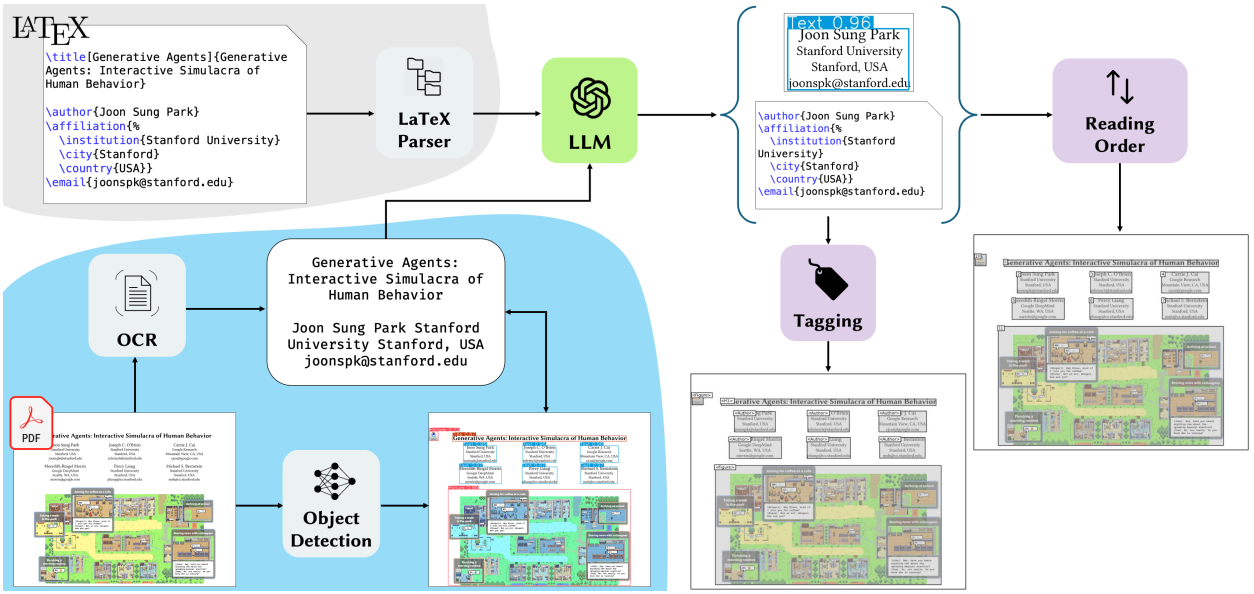


Figure 2: Our proposed method comprises of four steps: (i) processing the visual rendering of PDFs (in blue background), (ii) parsing the source document (in gray background), (iii) aligning the visual and structural representations using an LLM, and (iv) generating the metadata (tags and reading order).

content exists outside the current set of content regions, we treat it as a standalone `text` content region.

3.2 Source Document Parsing

Linearizing the Semantic Source. The source document(s) are linearized to form the semantic context for further remediation of the generated content regions. Our parser processes the `main.tex` file and recursively incorporates content from all included files (e.g., `<section>.tex` and `main.bbl`).

Chunking the Semantic Context. We pre-process the semantic context to remove comments, package imports, and other non-visual metadata. The cleaned document is then segmented into "chunks" – snippets of LaTeX code that are expected to render distinct visual content regions on a page. These chunks may include elements such as the title (`\title{}`), authors (`\author{}`), section headings (`\section{}` or `\ipstart{}`), lists `\begin{itemize}` or `\begin{enumerate}`), references (`@article{}`), footnotes (`\footnote{}`), and other meaningful constructs. We use hand-crafted regular expressions to split the source context.

3.3 Aligning Visual and Source Representations

Mapping Visual Content to Semantic Chunks. We align the OCR text output with the corresponding semantic chunk for each content region. First, we sort and filter the top 10 semantic chunks based on their forward and reverse word overlap scores with the OCR content. This list is then further filtered by prompting GPT-4o, a Large Language Model (LLM). We keep an LLM in the loop to interpret noisy OCR text, recover structure from ambiguous syntax, and handle edge cases where heuristics don't generalize.

Refining Mappings and Regions. The LLM agent returns an initial mapping of semantic chunks to content regions, identifying which snippets most plausibly render each visual element. We further refine this mapping through two adjustments:

- **Splitting or Merging Content Regions**: When one content region maps to multiple semantic chunks, distinct elements (*e.g.*, a section and a subsection header) have been erroneously merged into one content region. Conversely, when multiple content regions correspond to a single semantic chunk, a coherent visual element, such as a paragraph or list, has been incorrectly fragmented into several bounding boxes. We use bounding box area heuristics to correct such cases.
- **Mapping Figures to Semantic Chunks**: Since our mapping is based on text overlap, it does not associate visual elements with their corresponding semantic chunks (*i.e.*, `\begin{figure}` blocks). We identify content regions labeled as `Picture` in DocLayNet taxonomy and associate them with the nearest text region that has already been assigned to a figure chunk (typically the caption). We use a visual proximity heuristic (threshold: 4% page height) to determine this mapping.

3.4 Metadata Generation

Generating Content Tags. After obtaining the mapped semantic chunk for each content region, we use a combination of

regular expressions against the semantic chunk to correct or refine the visual label and generate the appropriate content tags. A content region not mapped to any semantic chunk is treated as an <Artifact>. Our current implementation produces 12 WYSIWYM tags: <H1:H3>, <Paragraph>, <Figure>, <Table>, <List>, <Artifact>, <Caption>, <Author>, <FE_Note>, and <BibEntry>. While our content region detector (Section 3.1) is trained on the DocLayNet taxonomy, our method produces more granular and semantically meaningful tags, often also correcting classification errors in the process.

Identifying Reading Order. We use the semantic chunk associated with each content region as an anchor for recovering the intended reading order of the PDF. Specifically, we sort content regions on each page based on the position of their corresponding semantic chunk in the semantic context file. This ensures the reading order follows the author intent: for example, reading footnotes at their mention, paragraphs spanning columns uninterrupted, associating figures with their captions, and skipping artifacts.

4 EXPERIMENTS

We performed a manual evaluation of our proposed system by assembling a dataset of ACM research papers along with their source files. We compare our approach with two baselines: *(i)* Adobe Acrobat Pro Auto-Tagging, the most widely used PDF remediation software, and *(ii)* the higher standard of author-submitted PDFs to ASSETS [8].

Dataset. We targeted papers published at the ACM ASSETS over the past six years, as the venue requires PDF accessibility tagging and recent papers are likely prepared with the latest remediation tools. We searched for them on arXiv and filtered for entries including source documents. For each paper, we downloaded the LaTeX source from arXiv and the corresponding PDF from the ACM Digital Library to obtain the authors' final submitted, tagged versions. We discarded any entries where the arXiv version differed from the ACM version. We collected 10 PDFs comprising 121 pages and over 2000 visual segments that required tagging.

Accessibility Criteria. We evaluated how well our generated metadata fulfills PDF accessibility criteria, as defined in prior work [12]. Based on our collected dataset and predicted tags, we selected a subset of the recommended criteria: tagging of all content, headings, heading levels, figures, tables, lists, captions, and correct reading order.

Results. The metadata generated by our approach surpasses both Adobe Acrobat and the authors on several accessibility criteria (see Table 1): reading order, tagging tables and captions. It identifies notes and captions, which were often incorrectly tagged as <Paragraph> by both Acrobat and the authors. Sample positive outputs from our method are shown in Figure 3.

We further analyzed our predicted tags to identify failure cases in our approach. Occasionally, our word overlap-based heuristics fail to identify corresponding semantic chunks for

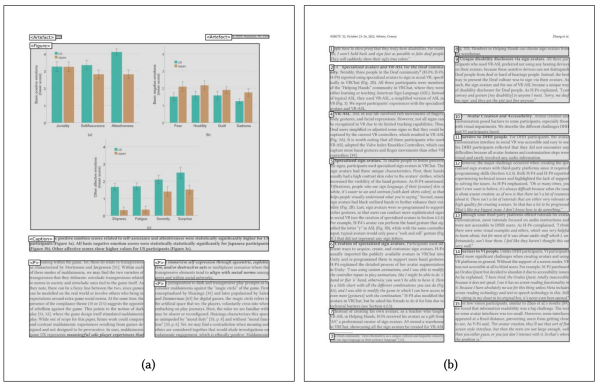


Figure 3: Positive example outputs from our method: (a) Our approach accurately classifies figures, captions, and artifacts; (b) It correctly predicts footnote order while skipping irrelevant artifacts.

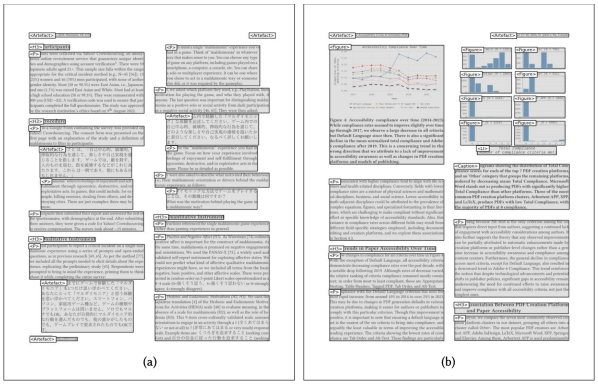


Figure 4: Negative example outputs from our method: (a) The method fails to tag paragraphs containing only Japanese text; (b) It inaccurately segments figures, occasionally merging them with captions or splitting them into sub-figure bounding boxes.

headings, resulting in incorrect tagging as <Artifact>. Further, our current implementation did not process content in other languages. Finally, our refinements in Section 3.3 were not always able to correct content region boundaries, especially when dealing with non-textual content, such as figures or sub-figures (see Figure 4).

Limitations. Our current implementation of this work presents several limitations which suggest avenues for future work. First, our method relies on heuristics to map and refine multiple representations, which struggle to generalize in observed failure cases. Future work could focus on building a large accessibility metadata corpus to support the training of auto-tagging approaches. Second, our approach can be refined to generate richer semantic metadata, including alt-texts for figures and equations and structured tagging for table and list elements, from the source documents. For example, we could

Criteria	Adobe Acrobat Auto-Tags [%]	What Authors Submitted [%]	WYISWYM Tags (Ours) [%]
All Content Tagged	100.0	100.0	100.0
Reading Order	89.66	95.419	**96.26**
Headings Tagged	90.47	94.78	82.68
Headings Tagged + Level	85.28	**94.35**	81.82
Tables Tagged	68.18	50.0	**95.0**
Lists Tagged	60.34	85.36	75.61
Figures Tagged	52.24	**100.0**	84.0
Captions Tagged	0.0	6.45	**95.65**
Average Score	71.79	78.21	**88.75**

Table 1: Comparison of PDF accessibility tagging between metadata generated by Acrobat's auto-tagging feature, what authors submitted and our generated metadata. To compare the results, we manually evaluate 10 randomly selected papers from ACM ASSETS (2019-2024).

further chunk the `\itemize` command into list items defined by `\item`. Third, while we demonstrate our approach works for LaTeX source documents, it can also be extended for WYSIWYG authoring tools like MS Word, which preserve a linear reading order. Finally, our work does not endorse the continued dominance of PDFs in academic publishing; formats like HTML and ePub offer far better accessibility support. However, given its pervasiveness in current research dissemination practices, we believe improving accessibility through better authoring and PDF remediation tools remains necessary, even as the community moves toward more accessible standards.

5 CONCLUSION

In this paper, we introduce a novel approach that combines visual tagging of PDFs with semantic cues from the source document, enabling more accurate and robust metadata generation. Our method offers what we believe to be the most reliable solution to date for automated PDF tagging and surpasses the quality of manual tagging provided by authors in our collected dataset. This work represents a concrete first step toward the long-standing goal of accurately and automatically tagging PDF documents.

REFERENCES

[1] Adobe Inc. 2023. *Adobe Acrobat Pro DC Accessibility Features.* Adobe Systems Incorporated. https://helpx.adobe.com/acrobat/using/create-verify-pdf-accessibility.html.

[2] Jeffrey P. Bigham, Erin L. Brady, Cole Gleason, Anhong Guo, and David A. Shamma. 2016. An Uninteresting Tour Through Why Our Research Papers Aren't Accessible. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI EA '16)*. Association for Computing Machinery, New York, NY, USA, 621–631. https://doi.org/10.1145/2851581.2892588

[3] Erin Brady, Yu Zhong, and Jeffrey P Bigham. 2015. Creating accessible PDFs for conference proceedings. In *Proceedings of the 12th International Web for All Conference*. 1–4.

[4] Equidox Software Company. 2025. Equidox PDF Accessibility Software. https://equidox.co/pdf-solutions/pdf-accessibility-software/. Accessed: 2025-04-17.

[5] Charles Frankston, Jonathan Godfrey, Shamsi Brinn, Alison Hofer, and Mark Nazzaro. 2024. HTML papers on arXiv–why it is important, and how we made it happen. *arXiv preprint arXiv:2402.08954* (2024).

[6] Glenn Jocher and Jing Qiu. 2024. *Ultralytics YOLO11.* https://github.com/ultralytics/ultralytics

[7] Anukriti Kumar and Lucy Lu Wang. 2024. Uncovering the New Accessibility Crisis in Scholarly PDFs. *arXiv preprint arXiv:2410.03022* (2024).

[8] Jonathan Lazar, Elizabeth F Churchill, Tovi Grossman, Gerrit van Der Veer, Philippe Palanque, John" Scooter" Morris, and Jennifer Mankoff. 2017. Making the field of computing more inclusive. *Commun. ACM* 60, 3 (2017), 50–59.

[9] Rachel Menzies, Garreth W Tigwell, and Michael Crabb. 2022. Author reflections on creating accessible academic papers. *ACM Transactions on Accessible Computing* 15, 4 (2022), 1–36.

[10] Frank Mittelbach, Ulrike Fischer, David Carlisle, and Joseph Wright. 2024. Automatically producing accessible and reusable PDFs with LATEX. In *Proceedings of the ACM Symposium on Document Engineering 2024*. 1–4.

[11] Birgit Pfitzmann, Christoph Auer, Michele Dolfi, Ahmed S Nassar, and Peter W J Staar. 2022. DocLayNet: A Large Human-Annotated Dataset for Document-Layout Analysis. (2022). https://doi.org/10.1145/3534678.353904

[12] Felix M. Schmitt-Koopmann, Elaine M. Huang, Hans-Peter Hutter, and Alireza Darvishy. 2025. Towards More Accessible Scientific PDFs for People with Visual Impairments: Step-by-Step PDF Remediation to Improve Tag Accuracy. In *Proceedings of the ACM Conference on Human Factors and Computing (CHI 2025)*. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.48550/arXiv.2503.22216

[13] Ray Smith. 2007. An overview of the Tesseract OCR engine. In *Ninth international conference on document analysis and recognition (ICDAR 2007)*, Vol. 2. IEEE, 629–633.

[14] Gregg Vanderheiden and Crystal Yvette Marte. 2024. Will AI allow us to dispense with all or most accessibility regulations?. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI EA '24)*. Association for Computing Machinery, New York, NY, USA, Article 571, 9 pages. https://doi.org/10.1145/3613905.3644059

[15] Lucy Lu Wang, Jonathan Bragg, and Daniel S Weld. 2023. Paper to html: A publicly available web tool for converting scientific pdfs into accessible html. *ACM SIGACCESS Accessibility and Computing* 134 (2023), 1–1.

[16] Lucy Lu Wang, Isabel Cachola, Jonathan Bragg, Evie Yu-Yen Cheng, Chelsea Haupt, Matt Latzke, Bailey Kuehl, Madeleine van Zuylen, Linda Wagner, and Daniel S Weld. 2021. Improving the accessibility of scientific documents: Current state, user needs, and a system solution to enhance scientific PDF accessibility for blind and low vision users. *arXiv preprint arXiv:2105.00076* (2021).