

Accessible PDFs: Applying Artificial Intelligence for Automated Remediation of STEM PDFs

Felix M. Schmitt-Koopmann
University of Zurich, Department of
Informatics and Zurich University of
Applied Science, Institute of Applied
Information Technology
fschmitt@ifi.uzh.ch

Prof. Dr. Elaine M. Huang
University of Zurich, Department of
Informatics
huang@ifi.uzh.ch

Prof. Dr. Alireza Darvishy
Zurich University of Applied Science,
Institute of Applied Information
Technology
alireza.darvishy@zhaw.ch

ABSTRACT

People with visual impairments use assistive technology, e.g., screen readers, to navigate and read PDFs. However, such screen readers need extra information about the logical structure of the PDF, such as the reading order, header levels, and mathematical formulas, described in readable form to navigate the document in a meaningful way. This logical structure can be added to a PDF with tags. Creating tags for a PDF is time-consuming, and requires awareness and expert knowledge. Hence, most PDFs are left untagged, and as a result, they are poorly readable or unreadable for people who rely on screen readers. STEM documents are particularly problematic with their complex document structure and complicated mathematical formulae. These inaccessible PDFs present a major barrier for people with visual impairments wishing to pursue studies or careers in STEM fields, who cannot easily read studies and publications from their field. The goal of this Ph.D. is to apply artificial intelligence for document analysis to reasonably automate the remediation process of PDFs and present a solution for large mathematical formulae accessibility in PDFs. With these new methods, the Ph.D. research aims to lower barriers to creating accessible scientific PDFs, by reducing the time, effort, and expertise necessary to do so, ultimately facilitating greater access to scientific documents for people with visual impairments.

CCS CONCEPTS

• **Human-centered computing** → Accessibility; Accessibility systems and tools; Accessibility; Accessibility technologies; • **Applied computing** → Document management and text processing; Document capture; Document analysis.

KEYWORDS

Accessibility, PDF Accessibility, Tagged PDF, PDF/UA, Math Viewer, Document Analysis, Formula Recognition, Page Object Detection, Reading Order

ACM Reference Format:

Felix M. Schmitt-Koopmann, Prof. Dr. Elaine M. Huang, and Prof. Dr. Alireza Darvishy. 2022. Accessible PDFs: Applying Artificial Intelligence



This work is licensed under a Creative Commons Attribution International 4.0 License.

ASSETS '22, October 23–26, 2022, Athens, Greece

© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-9258-7/22/10.
<https://doi.org/10.1145/3517428.3550407>

for Automated Remediation of STEM PDFs. In *The 24th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '22)*, October 23–26, 2022, Athens, Greece. ACM, New York, NY, USA, 6 pages.
<https://doi.org/10.1145/3517428.3550407>

1 INTRODUCTION

Since 1998, the US Rehabilitation Act section 508 [1] requires US Federal departments and agencies to make electronic and information technology accessible to people with disabilities. Additionally, the 2008 United Nations Convention on the Rights of Persons with Disabilities [2], and the 2019 European Accessibility Act [3], require that critical products and services be usable by people with disabilities. The members of the European Union must implement these requirements by 2025. One element of these acts is document accessibility.

The Portable Document Format (PDF) is the most popular document format, especially for scientific papers. Adobe created it in 1993, and since 2008 it is an open format managed by the PDF Association [4]. The PDF format was developed to display documents independent of the software and hardware used, which is one of the reasons for the format's popularity. In 2012, the PDF Association introduced the ISO 14289 standard, which is better known as the PDF/Universal Accessibility (UA) standard. It specifies that a PDF must be tagged to be accessible with assistive tools, such as screen readers. The tags contain information about the logical structure of the PDF, e.g. what is the header and which header level is it. This logical structure allows screen readers to correctly process content objects, such as headers, tables, and lists, and read the objects in the correct reading order. However, most PDFs do not meet the UA standard, and therefore are not easily readable for people with visual impairments who rely on screen readers.

Different tools exist to create accessible PDFs. These tools can be separated into two groups. The first group allows the tagging of existing PDFs, a process known as PDF remediation. The second group supports the generation of tags during the creation of the PDF e.g., with special add-ins for Microsoft Word or Microsoft PowerPoint [5]. In this Ph.D. project, we investigate PDF remediation, because PDF remediation can be applied to all PDFs and it is not software specific. For PDF remediation, the creator of the PDF can use programs such as Adobe Acrobat Pro [6], PAVE [7], [8], and others to tag their PDFs.

Nevertheless, most authors do not use these tools to create accessible PDFs. Research has shown there are three main reasons why many PDFs contain no tags [9]. The first reason is that authors lack awareness about accessible PDFs and do not know this problem exists. A second related reason is that PDF remediation requires

expert knowledge. Besides having to obtain expensive software to create accessible PDFs, the author also needs to be familiar with the guidelines to create accessibility tags. Thirdly, the creation of accessibility tags is time-consuming. Even an accessibility expert needs hours to make a document with a complex structure accessible. At present, however, there are no viable alternatives to expert manual tagging for complex documents. Existing remediation tools, such as Adobe Acrobat Pro, have automated tagging options, but their algorithms only work with simple documents, such as documents with basic linear structure and consisting primarily of text. When applied to more complex documents, e.g. documents with multi-column text, figures, lists, and mathematical formulae, many of the resulting tags are incorrect, potentially resulting in a reading order that jumps between two columns, headers not being detected, or through a particular Adobe bug, images being modified by the automatic addition of alternative text [10].

STEM documents are a challenge for PDF remediation because most of them necessarily have a complex document structure. In particular, most STEM documents contain mathematical formulae, and being able to access them is critical for understanding the content of the document. However, mathematical formulae are currently not addressed in the PDF 1.7 (ISO 32000-1:2008) standard [11] or the PDF/UA (ISO 14289-1:2014) standard [12]. Commonly, mathematical formulae are tagged by adding alternative text to the formula. This works well for small formulae, but for larger formulae the alternative text can get very long. For example, the solution of quadratic equations consists of 23 words. This results in a high mental load on the reader because it is not possible for them to focus selectively on parts of the formula or understand the broad structure of the formula before examining its details, due to the linear nature of the screen reader's presentation of the alternative text. This limitation is not merely inconvenient; it presents substantial disadvantages for people with visual impairments working or studying in STEM fields by restricting their access to valuable information. For these reasons, adding alternative text to mathematical formulae is not sufficient; better solutions are necessary to improve the accessibility of mathematical formulae in PDFs.

Hence, the goal of this Ph.D. project is to leverage artificial intelligence (AI) approaches to create a reasonably automated method to tag PDFs from the STEM field with a new solution for large mathematical formulae in PDFs. We will develop methods to automate parts of the PDF remediation process and integrate them into the existing PDF remediation tool PAVE to create PAVE 2.0. These new methods should allow authors who are not experts in the field of accessibility to tag PDFs in a manner that is substantially faster and easier than expert manual tagging. In addition to the PDF remediation methods, we plan to develop a new method to tag large formulae in PDFs to make them simpler to understand for users who rely on screen readers. We plan to evaluate our methods in several studies with users, to investigate the advantages and disadvantages of our methods compared to existing solutions.

With this project, we want to support people with visual impairment by leveraging AI approaches to increase the accessibility of mathematical formulae in PDFs with screen readers. Moreover, with PAVE 2.0 we aim to simplify the PDF remediation process for STEM documents, which could in turn increase the number of accessible STEM documents available. We hope this will in turn

have the effect of helping to reducing barriers for visual impairment to engage in studies or careers in STEM fields through increased access to scientific information.

In Chapter 2 we present related work of PDF remediation and mathematical formulae in PDFs. Chapter 3 presents the problem of large mathematical formulae in detail along with a possible solution. Chapter 4 presents technical details of PDF remediation with deep learning models. Chapter 5 presents our planned studies to explore the influence of our methods on accessibility. Chapter 6 presents the actual Ph.D. project stand and the time schedule.

2 RELATED WORK

In recent years, web accessibility has received substantial attention and many scholars, researchers, and practitioners are aware of the web content accessibility guidelines WCAG 2.1 [13]. However, fewer are aware of the related topic of PDF accessibility. Awareness of accessible PDFs is growing and there are many conferences that promote or encourage the use of accessible PDFs, but at present the process is still costly and time-consuming.

Research [9], [10] has shown that a major issue of PDF accessibility for STEM documents is the lack of good PDF remediation tools. There are two major options for improving PDF remediation tools. First, the development of new methods to automatically tag the documents with a higher accuracy, as we plan to do. Secondly, by enhancing the user interface and redesigning the tasks the user must do. One of the newest enhanced user interfaces is Ally [14] (not publicly available). Ally utilizes best practices from other HCI research, which allows Ally to speed up and to increase the accuracy of the implemented PDF remediation tasks. Nevertheless, the PDF remediation process is still time-consuming due to the significant manual work required by the author, as no solution for mathematical formulae has been presented.

Another tool that simplifies the PDF remediation process with an enhanced user interface is the PDF Accessibility and Validation Engine (PAVE). To the best of our knowledge, PAVE is the only free available web-based application. It allows the identification and correction of accessibility issues in PDF documents. Similar to Ally, it contains no sophisticated automated tagging of documents to reduce the manual work of the author or a solution for mathematical formulae.

The most popular, but fee-based, PDF remediation tool is Adobe Acrobat Pro [6]. It allows the user to do all sorts of PDF remediation tasks and it contains an auto-tagger to automatically tag PDFs. However, research showed that the user interface is not user-friendly [9] and the auto-tagger has problems with the complex document structures of STEM documents [10], which reduces significantly the value for the user. Due to its popularity, it is the baseline tool for PDF remediation research. Moreover, other tools build upon Adobe Acrobat Pro with plug-ins to improve the accessibility suite of Adobe Acrobat Pro.

One such plug-in is the fee-based PDF remediation tool CommonLook PDF [15]. It aims to simplify the PDF remediation process via an improved user interface and an enhanced accessibility checker, that fulfils the requirements of Section 508. Besides the PDF remediation tool, they provide the service of finalizing the PDF remediation process by an expert. Recently, the creators presented

CommonLook AI Cloud 2.0 [16], which uses AI to speed up the process. Nevertheless, they focus on documents from the industry and the government and not from the STEM field, with their special document structure and mathematical formulae.

To the best of our knowledge, nobody develops methods for PDF remediation of STEM documents with AI. However, automated PDF remediation requires steps involving document analysis, a common deep learning task that we address in Subsection 2.1.

To the best of our knowledge, there exists no research or solution for large mathematical formulae in PDFs. As a result, we cannot present any related work. We present in detail the problem and a possible solution in Section 3.

2.1 Deep Learning for PDF remediation

We have identified four steps that must be completed during PDF remediation. Firstly, the user identifies structures, such as headers, tables, and lists. Secondly, the user analyzes these objects. This could entail determining the level of the header, translating the formula into text, recognizing the table structure, and more. Thirdly, the user identifies the correct reading order. Fourthly, the user adds the tags with the information gained from the other steps. The first three steps are document analysis steps. Advanced optical character recognition (OCR) systems, like InftyReader [17], can detect document structure elements and mathematical formulae in addition to detecting the text. However, these systems are limited by the rule-based approach they are using. In a recent evaluation of the Infty system on a large dataset of real formulae drawn from 60'000 scientific papers in [18], this system achieved only a BLEU score of 67%. With such a score, many parts of a formula still need to be corrected manually, which greatly limits the benefit of an automated tagging system of mathematical formulae. As a result, we want to exceed these limitations with deep learning approaches.

To our knowledge, no comprehensive document analysis system for PDFs using a deep learning approach exists that enables automatic tagging of PDFs. However, there are many deep learning models that address aspects of document analysis and recognition tasks that we can leverage in our research. There exist deep learning models for detecting tables in documents [19]–[21], mathematical formula detection and recognition [22]–[24], document structure detection systems [25]–[28], and more. The technical details of our PDF remediation method are presented in Chapter 4 and the evaluation of our methods in Chapter 5.1.

3 MATHEMATICAL FORMULAE IN PDFS

Accessible mathematical formulae are currently not specifically addressed in the PDF 1.7 (ISO 32000-1:2008) standard [11] or the PDF/UA (ISO 14289-1:2014) standard [12]. As a result, there is no accepted standard way for tagging mathematical formulae. The most common way is to tag mathematical formulae by adding alternative text. The use of MathSpeak rules [29] is recommended for creating the alternative texts for mathematical formulae. For example, the alternative text for Equation 1 (below) with MathSpeak is “x equals start fraction minus b plus-or-minus square root of b squared minus 4 a c end root over 2 a end fraction”. There are two main issues with this method. First, the exact alternative text can vary if the author is not a MathSpeak expert. Second, the complexity

of the alternative text means that even a small formula such as this one can result in substantial cognitive load on the person with a visual impairment.

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad (1)$$

For other formats such as websites, there are solutions with “math viewers,” such as the math viewer provided with JAWS [30] or the similar access8math add-on for NVDA [31]. The purpose of a math viewer is to present a mathematical formula in a more meaningful way than just plain text. For example, JAWS converts the formula into a tree structure, allowing a user to selectively read parts of the formula and to understand the structure of the formula more easily. The user can navigate with in a formula with 4 actions: First, the user can step into the current part of the formula. Second, the user can step out of the current part of the formula. Third, the user can go to the next right element. Fourth, the user can go to the next left element. A potential exploration of Equation 1 is presented in Table 1.

The math viewer concept of JAWS could be a possible approach for large mathematical formulae in PDFs, but we could not find any research about math viewers or if the presented concept helps people to understand mathematical formulae better. Hence, we plan to evaluate the math viewer concept of JAWS as a possible PDF math viewer approach in a user study, as described in Chapter 5.2. Depending on the results, we will change the presented math viewer concept to improve the understanding of mathematical formulae. Our final PDF math viewer should be web-based, so no local software is required. This could be achieved by working with links in the PDF. The resulting PDF math viewer will be compared in a user study with the existing method of alternative texts (see Chapter 5.2).

4 TECHNICAL DETAILS OF PDF REMEDIATION

Document analysis with PDFs is challenging because the raw PDF file depends on the software used to create the PDF. As a result, visually similar PDFs can have different raw PDF files. Due to the large variation, analysis is challenging and sometimes impossible. Therefore, we use images of each page for the document analysis steps instead of the raw PDF file. Another advantage of using images is that image processing is a common research field of deep learning and accordingly, many methods exist. An overview of the tagging pipeline is shown in Figure 1.

Because the creation of a complete document analysis system to tag a PDF would exceed the scope of this Ph.D. project, we decided to build our system upon the existing PDF remediation tool PAVE. The goal is to automate and improve the most important parts of PAVE to speed up the PDF remediation process and automatically tag mathematical formulae. We have identified three elements of the tagging pipeline necessary for achieving this goal.

The first element is the detection of the different logical content parts in a document. Detecting the correct logical content part is crucial for PDF remediation and can speed up the tagging process most. We will detect the different logical content parts by using a Page Object Detection (POD) model. Most page object detection models do not detect formulae in the text due to the lack of large and

Table 1: Example of the workwise of the math viewer at the example of Equation 1.

Action	Output
open math viewer	$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$
step into	x
next right	$=$
next right	$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$
step into	$-b \pm \sqrt{b^2 - 4ac}$
step into	$-$
next right	b
next right	\pm
next right	$b^2 - 4ac$
step out	$-b \pm \sqrt{b^2 - 4ac}$
next right	$2a$

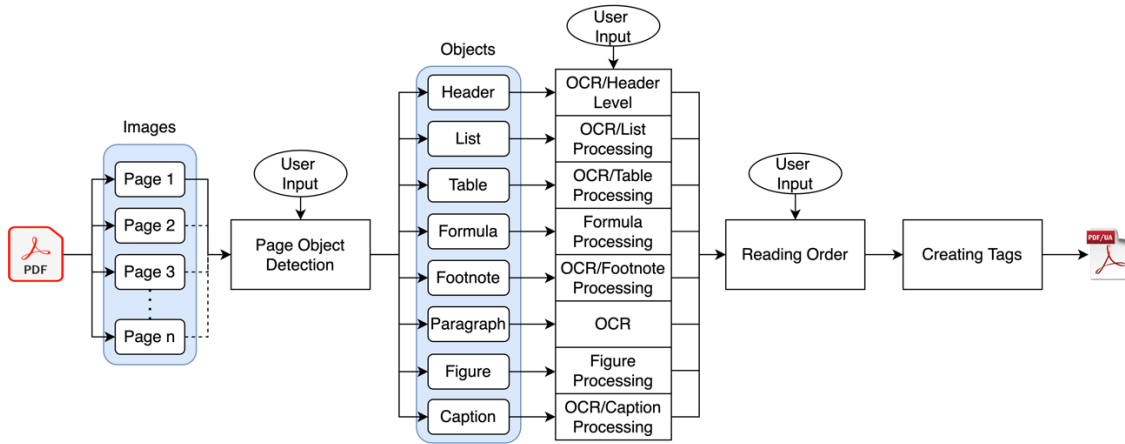


Figure 1: Overview of the tagging pipeline.

high-quality labeled datasets on which to train them. To address this, we developed FormulaNet (publication in preparation) a new POD dataset with formula labels. Currently, we develop our POD model based on the object detection model Generalized Focal Loss V2 [32].

The second element is the recognition of mathematical formulae based on images. This means the input for the model is an image of the mathematical formula, and the output is the mathematical formula as text in a markup language like MathML [33]. Mathematical formula recognition in documents is an unsolved problem for two main reasons. First, the detection of formulae within a PDF document is still challenging, especially when the formulae are embedded in the text [22], [34]. Second, current promising formula recognition results have been achieved under ideal conditions [35], which means perfectly snapped and low style variation of the input image. However, our formula recognition model must handle less perfect input images because the POD model will not provide such perfect inputs for the formula recognition model. The latest challenges of formula recognition showed that the end-to-end models

achieve better results than the two-step approach [34]. Hence, we plan to develop an end-to-end model based on [35].

The third element we want to improve is the reading order. The reading order allows screen readers to guide the user through the content of a PDF in the correct logical order and as a result, it has a large impact on the user experience with screen readers. If the POD works correctly, the reading order should be determined by rules. Experiments will show if simple left-to-right and top-to-bottom heuristics allow determining the reading order by using the POD information. If simple rules are not enough, we will train another deep learning model to detect the reading order. One of the most promising models is the LayoutReader [36], which uses a sequence to sequence model and builds upon LayoutLM [37].

We plan to implement our methods for POD, formula recognition, and reading order into PAVE, which we will call PAVE 2.0. The planned studies (see Section 5.1) will show if PAVE 2.0 has the ability to speed up the tagging process and improve the user experience.

5 EVALUATION OF OUR METHODS

5.1 PDF Remediation

We plan to conduct two studies to analyze the tag quality and user experience of our automated tagging pipeline. The first study (Section 5.1.1) investigates the tag quality of our method compared to existing tools. The second study (Section 5.1.2) will investigate the authors' user experience with our automated method compared to the automated tagging method.

5.1.1 Experimental study: tag quality of our method compared to existing methods. This study will investigate the quality of automated generated tags. The experiment will allow us to answer three questions. First, what are the challenges for automated tagging methods? Second, what are the strength and weaknesses of PAVE 2.0? Third, does it improve PAVE, and what are the improvements? We will compare the results of our automated tagging pipeline with existing automated tagging tools such as the auto tagger from Adobe. We plan to analyze the tagging quality with two document collections. First, we will collect 20 STEM documents that address specific challenging accessibility issues. This should show how the automated tagging pipelines handle difficult issues. However, this collection will not be representative of all STEM documents. Therefore, we will create the second collection by randomly sampling 20 STEM documents from arXiv.org [38] from the year 2022. This will allow us to investigate what the average results that can be expected are and how our method performs in comparison with other auto taggers.

We plan to evaluate the results manually with a predefined examination sheet. The examination sheet will be grouped into 3 groups. The first group will investigate if the different contents of the PDF are detected and grouped correctly. The second group investigates if the different objects are processed correctly. The last group will evaluate the reading order detected.

5.1.2 User study: user experience of our method compared to existing methods. This study will investigate the user experience of our method. We plan to do this study with 25 authors of STEM documents. They should have different levels of experience (no experience, medium experience, expert) with PDF accessibility. The task will be to remediate 5 short STEM documents with the PAVE, PAVE 2.0, and Adobe Acrobat Pro tools. 4 of the 5 documents are randomly selected STEM documents from arXiv.org, and 1 of them is prepared with challenging remediation tasks. They will watch a short introduction video about the remediation tool they will use, and then have 30 minutes to make the PDF accessible. We will record the screen as they do so, while tracking the time they need to solve different remediation tasks. At the end, the study participants will fill out a survey about their experience with the PDF remediation tool and what parts of the document are accessible or not. Additionally, we will analyze the accessibility of the resulting documents with the examination sheet from the previous experiment (see Chapter 5.1.1).

5.2 Math Viewer

We plan two studies to assess the math viewer. The first study (see Section 5.2.1) will investigate the user experience of the existing

math viewer from JAWS. The second study (see Section 5.2.2) will explore the impact of our PDF math viewer on the user experience.

5.2.1 User study: user experience with JAWS math viewer. This user study will compare the math viewer concept from JAWS with a plain description of the formula. We do not compare the math viewer plug-in access8math for NVDA with the math viewer from JAWS, because their concept is the same. We will recruit approximately 10 people with visual impairments who use screen readers for the study. For the task, we will select 20 formulae of different lengths from STEM documents. For each formula we will create questions to assess the reader's understanding of the formula. Participants will first get a short introduction of the math viewer. Then, they will use the math viewer from JAWS for 10 formulae and the alternative text for 10 other formulae. The study participants have up to 5 minutes per formula to answer the questions and we will record the screen, their input, and the output of the math viewer. We will analyze the results with respect to the speed and correctness of their responses. Participants will also fill out a survey about their experience with the math viewer and the alternative text which we will analyze to understand their comfort and satisfaction with the approaches. This study will help us to understand the advantages and disadvantages of the math viewer concept of JAWS and what the advantages and disadvantages are compared to alternative text.

5.2.2 User study: user experience of PDF math viewer. After evaluating JAWS math viewer concept in the previous study, we will modify the math viewer concept if needed and we will develop a method to integrate the math viewer concept into PDF. This second user study will investigate the user experience of our implementation of the PDF math viewer compared with alternative text in STEM documents. Therefore, we will prepare 5 STEM documents with our PDF math viewer and with alternative text of the formulae. We will again need 10 people with visual impairments who use screen readers for the study. They will get the STEM documents with or without the math viewer. They will need to answer questions related to mathematical formulae in the documents. Again, we will record the screen. At the end, the study participants will fill out a survey about their experience with the math viewer and the alternative text. We will also compare the results with respect to time and correct answers. This study will show whether our PDF math viewer helps to speed up the understanding of mathematical formulae and improves the correct understanding of mathematical formulae.

6 PH.D. PROJECT OVERVIEW

This Ph.D. project is planned to be completed in 4 years. As presented in the previous chapters, the project can be divided into two work packages. The first work package is the automated PDF remediation presented in Chapter 4. The development of this package should be completed within 2 years and 3 months. It can be divided into six sub-packages (POD dataset, POD Model, Formula Recognition, Reading Order, Pave 2.0, and studies). The first of these sub-packages (POD dataset) is complete, and the POD model and Formula Recognition sub-packages are in progress. The second work package, the math viewer, will require approximately 1 year

to complete. It consists of the two planned studies and the development of the PDF Math Viewer. This work package is still in the planning stage. The last 9 months are designated for writing the dissertation and as a reserve time for finishing studies or making additional improvements to the PDF remediation pipeline or the math viewer.

REFERENCES

- [1] “29 U.S. Code §794d - Electronic and information technology,” *LII / Legal Information Institute*. <https://www.law.cornell.edu/uscode/text/29/794d> (accessed Jun. 15, 2022).
- [2] “Convention on the Rights of Persons with Disabilities – Articles | United Nations Enable,” 2008. [https://www.un.org/development/desa/disabilities/convention-on-the-rights-of-persons-with-disabilities-2.html](https://www.un.org/development/desa/disabilities/convention-on-the-rights-of-persons-with-disabilities/convention-on-the-rights-of-persons-with-disabilities-2.html) (accessed May 24, 2022).
- [3] Directive (EU) 2019/882 of the European Parliament and of the Council of 17 April 2019 on the accessibility requirements for products and services (Text with EEA relevance), vol. 151. 2019. Accessed: May 24, 2022. [Online]. Available: <http://data.europa.eu/eli/dir/2019/882/oj/eng>
- [4] “PDF Association.” <https://www.pdfa.org> (accessed May 24, 2022).
- [5] “ICT-Accessibility: Accessibility Plug-ins.” <https://accessibility.zhaw.ch/indexea25.html> (accessed Jun. 20, 2022).
- [6] “Adobe Acrobat Pro DC download | Adobe Acrobat.” <https://www.adobe.com/acrobat/acrobat-pro.html> (accessed Feb. 23, 2022).
- [7] L. Doblies, D. Stolz, A. Darvishy, and H.-P. Hutter, “PAVE: a web application to identify and correct accessibility problems in PDF documents,” 2014, pp. 185–192. doi: 10.1007/978-3-319-08596-8.
- [8] “PAVE – PDF-Barrierefreiheit Überprüfen und Verbessern.” <https://pave-pdf.org/> (accessed Feb. 23, 2022).
- [9] A. Jembu Rajkumar, J. Lazar, J. B. Jordan, A. Darvishy, and H.-P. Hutter, *PDF Accessibility of Research Papers: What Tools are Needed for Assessment and Remediation?* 2020. Accessed: May 24, 2022. [Online]. Available: <http://hdl.handle.net/10125/64254>
- [10] J. P. Bigham, E. L. Brady, C. Gleason, A. Guo, and D. A. Shamma, “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*, New York, NY, USA, Mai 2016, pp. 621–631. doi: 10.1145/2851581.2892588.
- [11] “ISO 32000-1:2008,” *PDF 1.7*. <https://www.iso.org/standard/51502.html> (accessed Jun. 08, 2022).
- [12] “ISO 14289-1:2014,” *PDF/UA-1*. <https://www.iso.org/standard/64599.html> (accessed Jun. 08, 2022).
- [13] “Web Content Accessibility Guidelines (WCAG) 2.1” <https://www.w3.org/TR/WCAG21/> (accessed Jun. 07, 2022).
- [14] D. Pradhan et al., “Development and Evaluation of a Tool for Assisting Content Creators in Making PDF Files More Accessible,” *ACM Trans. Access. Comput.*, vol. 15, no. 1, pp. 1–52, Mar. 2022, doi: 10.1145/3507661.
- [15] “CommonLook PDF | PDF Accessibility and Compliance,” Dec. 31, 2018. <https://commonlook.com/accessibility-software/pdf/> (accessed Jun. 08, 2022).
- [16] “AI Cloud | PDF Accessibility and Compliance.” <https://commonlook.com/accessibility-software/ai-cloud/> (accessed Jun. 08, 2022).
- [17] M. Suzuki, F. Tamari, R. Fukuda, S. Uchida, and T. Kanahori, “INFTY: an integrated OCR system for mathematical documents,” in *Proceedings of the 2003 ACM symposium on Document engineering*, New York, NY, USA, Nov. 2003, pp. 95–104. doi: 10.1145/958220.958239.
- [18] Y. Deng, A. Kanervisto, J. Ling, and A. M. Rush, “Image-to-Markup Generation with Coarse-to-Fine Attention,” in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, Sydney, NSW, Australia, Aug. 2017, pp. 980–989.
- [19] B. Coüasnou and A. Lemaitre, “Recognition of Tables and Forms,” in *Handbook of Document Image Processing and Recognition*, D. Doermann and K. Tombre, Eds. London: Springer London, 2014, pp. 647–677. doi: 10.1007/978-0-85729-859-1_20.
- [20] S. Schreiber, S. Agne, I. Wolf, A. Dengel, and S. Ahmed, “DeepDeSRT: Deep Learning for Detection and Structure Recognition of Tables in Document Images,” in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, Kyoto, Nov. 2017, pp. 1162–1167. doi: 10.1109/ICDAR.2017.192.
- [21] S. A. Siddiqui, M. I. Malik, S. Agne, A. Dengel, and S. Ahmed, “DeCNT: Deep Deformable CNN for Table Detection,” *IEEE Access*, vol. 6, pp. 74151–74161, 2018, doi: 10.1109/ACCESS.2018.2880211.
- [22] D. Anitei, J. A. Sánchez, J. M. Fuentes, R. Paredes, and J. M. Benedí, “ICDAR 2021 Competition on Mathematical Formula Detection,” in *Document Analysis and Recognition – ICDAR 2021*, Cham, 2021, pp. 783–795. doi: 10.1007/978-3-030-86337-1_52.
- [23] M. Z. Afzal, K. A. Hashmi, A. Pagani, M. Liwicki, and D. Stricker, “DeHyFoNet: Deformable Hybrid Network for Formula Detection in Scanned Document Images,” Jan. 2022, doi: 10.20944/preprints202201.0090.v1.
- [24] Y. Zhong et al., “1st Place Solution for ICDAR 2021 Competition on Mathematical Formula Detection,” *ArXiv210705534 Cs*, Jul. 2021, Accessed: Feb. 23, 2022. [Online]. Available: <http://arxiv.org/abs/2107.05534>
- [25] C. Clausner, A. Antonacopoulos, and S. Pletschacher, “ICDAR2017 Competition on Recognition of Documents with Complex Layouts - RDCL2017,” in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, Nov. 2017, vol. 01, pp. 1404–1410. doi: 10.1109/ICDAR.2017.229.
- [26] C. Clausner, A. Antonacopoulos, and S. Pletschacher, “ICDAR2019 Competition on Recognition of Documents with Complex Layouts - RDCL2019,” in *2019 International Conference on Document Analysis and Recognition (ICDAR)*, Sep. 2019, pp. 1521–1526. doi: 10.1109/ICDAR.2019.00245.
- [27] B. Meier, T. Stadelmann, J. Stampfli, M. Arnold, and M. Cieliebak, “Fully Convolutional Neural Networks for Newspaper Article Segmentation,” in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, Kyoto, Nov. 2017, pp. 414–419. doi: 10.1109/ICDAR.2017.75.
- [28] T. Palfray, D. Hebert, S. Nicolas, P. Tranouez, and T. Paquet, “Logical segmentation for article extraction in digitized old newspapers,” in *Proceedings of the 2012 ACM symposium on Document engineering - DocEng '12*, Paris, France, 2012, p. 129. doi: 10.1145/2361354.2361383.
- [29] “MathSpeak and MathSpeak Rules | SeeWriteHear,” Jun. 09, 2021. <https://www.seewritehear.com/learn/mathspk-and-mathspeak-rules/> (accessed May 19, 2022).
- [30] “Accessing Math Content with JAWS and Fusion – Freedom Scientific.” <https://www.freedomscientific.com/training/teachers/accessing-math-content-with-jaws-and-fusion/> (accessed May 18, 2022).
- [31] “Access8Math.” <https://addons.nvda-project.org/addons/access8math.en.html> (accessed Jun. 08, 2022).
- [32] X. Li, W. Wang, X. Hu, J. Li, J. Tang, and J. Yang, “Generalized Focal Loss V2: Learning Reliable Localization Quality Estimation for Dense Object Detection,” in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2021, pp. 11627–11636. doi: 10.1109/CVPR46437.2021.01146.
- [33] “Mathematical Markup Language (MathML) Version 3.0 2nd Edition.” <https://www.w3.org/TR/MathML/> (accessed May 18, 2022).
- [34] M. Mahdavi, R. Zanibbi, H. Mouchere, C. Viard-Gaudin, and U. Garain, “ICDAR 2019 CROHME + TFD: Competition on Recognition of Handwritten Mathematical Expressions and Typeset Formula Detection,” in *2019 International Conference on Document Analysis and Recognition (ICDAR)*, Sydney, Australia, Sep. 2019, pp. 1533–1538. doi: 10.1109/ICDAR.2019.00247.
- [35] Z. Wang and J.-C. Liu, “Translating Math Formula Images to LaTeX Sequences Using Deep Neural Networks with Sequence-level Training,” *Int. J. Doc. Anal. Cognit. IJ DAR*, vol. 24, no. 1–2, pp. 63–75, Jun. 2021, doi: 10.1007/s10032-020-00360-2.
- [36] Z. Wang, Y. Xu, L. Cui, J. Shang, and F. Wei, “LayoutReader: Pre-training of Text and Layout for Reading Order Detection,” in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Online and Punta Cana, Dominican Republic, 2021, pp. 4735–4744. doi: 10.18653/v1/2021.emnlp-main.389.
- [37] Y. Xu, M. Li, L. Cui, S. Huang, F. Wei, and M. Zhou, “LayoutLM: Pre-training of Text and Layout for Document Image Understanding,” in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Aug. 2020, pp. 1192–1200. doi: 10.1145/3394486.3403172.
- [38] “arXiv.org e-Print archive.” <https://arxiv.org/> (accessed Feb. 23, 2022).