PAWLS: PDF Annotation With Labels and Structure

Mark Neumann **Zejiang Shen** **Sam Skjonsberg**

Allen Institute for Artificial Intelligence
{markn,shannons,sams}@allenai.org

Abstract

Adobe's Portable Document Format (PDF) is a popular way of distributing view-only documents with a rich visual markup. This presents a challenge to NLP practitioners who wish to use the information contained within PDF documents for training models or data analysis, because annotating these documents is difficult. In this paper, we present PDF Annotation with Labels and Structure (PAWLS), a new annotation tool designed specifically for the PDF document format. PAWLS is particularly suited for mixed-mode annotation and scenarios in which annotators require extended context to annotate accurately. PAWLS supports span-based textual annotation, N-ary relations and freeform, non-textual bounding boxes, all of which can be exported in convenient formats for training multi-modal machine learning models. A PAWLS demo server is available at https://pawls.apps.allenai.org/ [1] and the source code can be accessed at https://github.com/allenai/pawls.

1 Introduction

Scholars of Natural Language Processing technology rely on access to gold standard annotated data for training and evaluation of learning algorithms. Despite successful attempts to create machine readable document formats such as XML and HTML, the Portable Document Format (PDF) is still widely used for read-only documents which require visual markup, across domains such as scientific publishing, law, and government. This presents a challenge to NLP practitioners, as the PDF format does not contain exhaustive markup information, making it difficult to extract semantically meaningful regions from a PDF. Annotating text extracted from PDFs in a plaintext format is difficult, because the extracted text stream lacks any organization or markup, such as paragraph boundaries, figure placement and page headers/footers.

Existing popular annotation tools such as BRAT (Stenetorp et al., 2012) focus on annotation of user provided plain text in a web browser specifically designed for annotation only. For many labeling tasks, this format is exactly what is required. However, as the scope and ability of natural language processing technology goes beyond purely textual processing due in part to recent advances in large language models (Peters et al., 2018; Devlin et al., 2019, *inter alia*), the context and media in which datasets are created must evolve as well.

In addition, the quality of both data collection and evaluation methodology is highly dependent on the particular annotation/evaluation context in which the data being annotated is viewed (Joseph et al., 2017; Läubli et al., 2018). Annotating data directly on top of a HTML overlay on an underlying PDF canvas allows naturally occurring text to be annotated in its original context - that of the PDF itself.

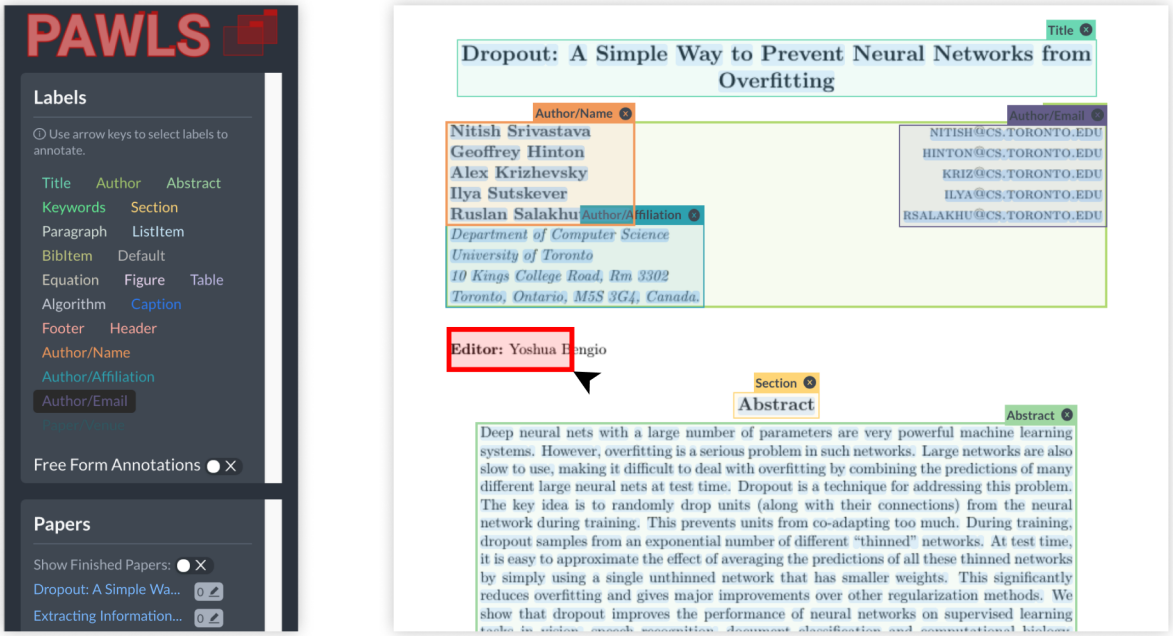
To address the need for an annotation tool that goes beyond plaintext data, we present a new annotation tool called PAWLS (PDF Annotation With Labels and Structure). In this paper, we discuss some of the PDF-specific design choices in PAWLS, including automatic bounding box uniformity, freeform annotations for non-textual image regions and scale/dimension agnostic bounding box storage. We report agreement statistics from an initial round of labelling during the creation of a PDF structure parsing dataset for which PAWLS was originally designed.

2 Design Choices

As shown in Figure 1, the primary operation that PAWLS supports is drawing a bounding box over

[1] Please see Appendix A for instructions on accessing the demo and the demo video.

Proceedings of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations, pages 258–264, August 1st - August 6th, 2021.



1 Control Panel

Annotators can select label types, check labeling status, and add comments for the labeling task.

2 Labeling Canvas

Directly create annotations over PDF documents. To maximize efficiency, annotators create rectangular bounding boxes, with the contained tokens automatically selected and assigned to the label. Freeform bounding boxes without textual content are also supported.

Figure 1: An overview of the PAWLS annotation interface. We show an example of annotating scientific documents in PAWLS, yet the target documents and labeling categories could be easily switched to other domains in a self-hosted version.

a PDF document with the mouse, and assigning that region of the document a textual label. PAWLS supports drawing both freeform boxes anywhere on the PDF, as well as boxes which are associated with tokens extracted from the PDF itself.

This section describes some of the user interface design choices in PAWLS.

2.1 PDF Native Annotation

The primary tenet of PAWLS is the idea that annotators are accustomed to reading and interacting with PDF documents themselves, and as such, PAWLS should render the actual PDF as the medium for annotation. In order to achieve this, annotations themselves must be relative to a rendered PDF's scale in the browser. Annotations are automatically re-sized to fit the rendered PDF canvas, but stored relative to the absolute dimensions of the original PDF document.

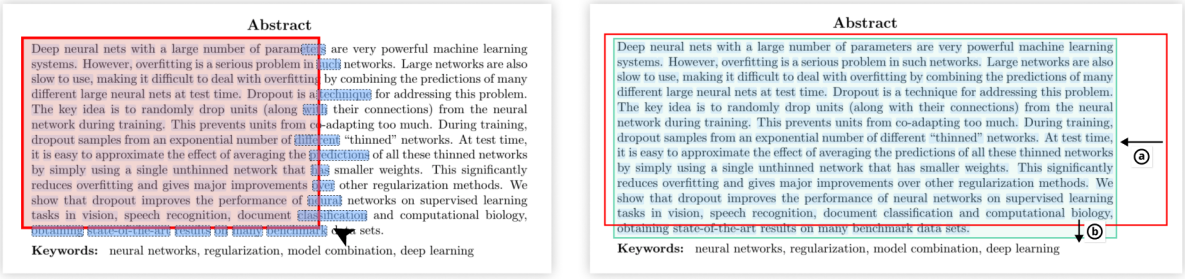
2.2 Annotator Ease of Use

PAWLS contains several features which are designed to speed up annotation by users, as well as minimizing frustrating or difficult interaction experiences. Bounding box borders in PAWLS change depending on the size and density of the annotated span, making it easier to read dense annotations. Annotators can hide bounding box labels using the CTRL key for cases where labels are obscuring the document flow. Users can undo annotations with familiar key combinations (CMD-z) and delete annotations directly from the sidebar. These features were derived from a tight feedback loop with annotation experts during development of the tool.

2.3 Token Parsing

PAWLS pre-processes PDFs before they are rendered in the UI to extract the bounding boxes of every token present in the document. This allows a variety of interactive labelling features described below. Users can choose between different pre-processors based on their needs, such as GROBID [2] and PdfPlumber [3] for digital-born PDFs, or Tesseract [4] for Optical Character Recognition (OCR) in PDFs which have been scanned, or are otherwise low quality. Future extensions to PAWLS will include higher level PDF structure which is general enough to be useful across a range of domains, such as document titles, paragraphs and section

[2] https://github.com/kermitt2/grobid
[3] https://github.com/jsvine/pdfplumber
[4] https://github.com/tesseract-ocr/tesseract



1 Auto detection of the contained tokens

For a user-drawn region bounding box (red), PAWLS automatically detects the contained text, including tokens at the boundaries (highlighted in blue).

2 Box Snapping

For a user-drawn region box (red), PAWLS automatically trims white space (arrow a) and recovers partially labeled word regions (arrow b), generating accurate and normalized region boundaries (green).

Figure 2: An example of visual token selection. When a user begins highlighting a bounding box, PAWLS uses underlying token level boundary information extracted from the PDF to 1) highlight selected textual spans as they are dragged over and 2) normalize the bounding box of a selection to be a fixed padded distance from the maximally large token boundary.

headings to further extend the possible annotation modes, such as clicking on paragraphs or sections.

2.4 Visual Token Selection and Box Snapping

PAWLS pre-processes PDFs before they are served in the annotation interface, giving access to token level bounding box information. When users draw new bounding boxes, token spans are highlighted to indicate their inclusion in the annotation. After the user has completed the selection, the bounding box "snaps" to a normalized boundary containing the underlying PDF tokens. Figure 2 demonstrates this interaction. In particular, this allows bounding boxes to be normalized relative to their containing token positions (having a fixed border), making annotations more consistent and uniform with no additional annotator effort. This feature allows annotators to focus on the content of their annotations, rather than ensuring a consistent visual markup, easing the annotation flow and increasing the consistency of the collected annotations.

2.5 N-ary Relational Annotations

PAWLS supports N-ary relational annotations as well as those based on bounding boxes. Relational annotations are supported for both textual and free-form annotations, allowing the collection of event structures which include non-textual PDF regions, such as figure/table references, or sub-image coordination. For example, this feature would allow annotators to link figure captions to particular figure regions, or relate a discussion of a particular table column in the text to the exact visual region of the column/table itself. Figure 3 demonstrates this interaction mode for two annotations.

Relation Annotation

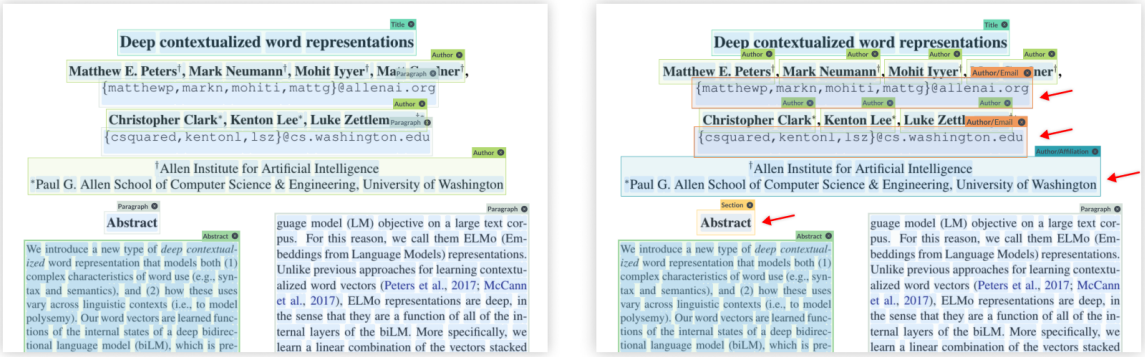
After selecting the five bounding boxes with Shift + Click (step a) , users select a relation label and organise the annotations using a Select View (step b), allowing groups, directed annotations, and multi-entity events.

Figure 3: The n-ary relation annotation modal.

2.6 Command Line Interface

PAWLS includes a command line interface for administrating annotation projects. It includes functionality for assigning labeling tasks to annotators, monitoring the annotation progress and quality (measuring inter annotator agreement), and exporting annotations in a variety of formats. Additionally, it supports pre-populating annotations from model predictions, detailed in Section 2.7.

Annotations in PAWLS can be exported to different formats to support different downstream tasks. The hierarchical structure of user-drawn blocks and PDF tokens is stored in JSON format, linking blocks with their corresponding tokens. For vision-centered tasks (e.g., document layout detection),



1 Annotation pre-population with existing models

We trained a Mask R-CNN model to predict text regions on page images and a text classification model for obtaining the block categories.

2 Gold labels after manual correction

The accurate model predictions can significantly improve the efficiency as only a part of the annotations require changes (highlighed by red arrows).

Figure 4: Annotation pre-population can significantly improve labeling efficiency.

PAWLS supports converting to the widely-used COCO format, including generating jpeg captures of pdf pages for training vision models. For text-centric tasks, PAWLS can generate a table for tokens and labels obtained from the annotated bounding boxes.

2.7 Annotation Pre-population

The PAWLS command line interface supports pre-population of annotations given a set of bounding boxes predictions for each page. Figure 4 illustrates how pre-annotation can help improve the labeling efficiency. In this example, we trained a Mask R-CNN (He et al., 2017) model on the PubLayNet (Zhong et al., 2019b) dataset that can detect content region bounding boxes for the input page image, and a BERT model (Devlin et al., 2018) on the DocBank (Li et al., 2020c) dataset that predicts the textual category for each text region. PAWLS loads the model predictions and automatically corrects the bounding boxes using the block snapping function, and annotators only need to make minor modifications in the box categories to obtain the gold annotations.

This further enables model-in-the-loop type functionality, with annotators correcting model predictions directly on the PDF. Future extensions to PAWLS will include active learning based annotation suggestions as annotators work, from models running as a service.

3 Implementation

PAWLS is implemented as a Python-based web server which serves PDFs, annotations and other metadata stored on disk in the JSON format. The user interface is a Single Page Application implemented using Typescript and relies heavily on the React web framework. PDFs are rendered using PDF.js. PAWLS is designed to be used in a browser, with no setup work required on the behalf of annotators apart from navigating to a web page. This makes annotation projects more flexible as they can be distributed across a variety of crowd-sourcing platforms, used in house, or run on local machines.

PAWLS development and deployment are both managed using the containerization tools Docker and Docker Compose, and multiple PAWLS instances are running on a Google Cloud Platform Kubernetes cluster. Authentication in production environments is managed via Google Account logins, but PAWLS can be run locally by individual users with no authentication.

4 Case Study

PAWLS enables the collection of mixed-mode annotations on PDFs. PAWLS is currently in use for a PDF Layout Parsing project for academic papers, for which we have collected an initial set of gold standard annotations. This dataset consists of 80 PDF pages with 2558 densely annotated bounding boxes of 20 categories from 3 annotators.

Table 1 reports pairwise Inter-Annotator agreement scores, split out into textual and non-textual labels. For textual labels like titles and paragraphs, the agreement is measured via token accuracy: for each word labeled, we compare the label of the belonging block across different annotators. Non-textual labels are used for regions like figures and tables, and they are usually labeled using free-form boxes. Average Precision (AP) score (Lin et al.,

2014), commonly used in Object Detection tasks (e.g., COCO) in computer vision, is adopted to measure the consistency of these boxes labeled by different annotators. As AP calculates the block categories agreement at different overlapping levels, the scoring is not commutative, and an 80 AP scores already suggests a high level of annotation quality.

	Annotator 1	**Annotator 2**	**Annotator 3**
Annotator 1	N/A	94.43 / 86.58	93.28 / 83.97
Annotator 2	94.43 / 86.49	N/A	88.69 / 84.20
Annotator 3	93.28 / 84.67	88.69 / 84.79	N/A

Table 1: The Inter-Annotator Agreement scores for the labeling task. We show the textual / non-textual annotation agreement scores in each cell. The (i, j)-th element in this table is calculated by treating i's annotation as the "ground truth" and j's as the "prediction".

5 Related Work

Many commercial PDF annotation tools exist, such as IBM Watson's smart document understanding feature and TagTog's Beta PDF Annotation tool [5]. PAWLS is open source and freely available. Knowledge management systems such as Protégé (Musen, 2015) support PDFs, but more suited to management of large, evolving corpora and knowledge graph construction than the creation of static datasets.

LabelStudio [6] supports image annotation as well as plaintext/html-based annotation, meaning PDF pages can be uploaded and annotated within their user interface. However, bounding boxes are hand drawn, and the context of the entire PDF is not visible as the pdf pages are viewed as individual images. PDFAnno (Shindo et al., 2018) is the closest tool conceptually to PAWLS, supporting multiple annotation modes and pdf-based rendering. Unfortunately PDFAnno is no longer maintained and PAWLS provides additional functionality, such as pre-annotation.

Several PDF based datasets exist for document parsing, such as DocBank (Li et al., 2020b), PubLeNet (Zhong et al., 2019a) and TableBank (Li et al., 2020a). However, both DocBank and PubLeNet are constructed using weak supervision from Latex parses or Pubmed XML information. TableBank consists of 417k tables extracted from Microsoft Word documents and computer generated PDFs. This approach is feasible for common elements of document structure such as tables, but is not possible for custom annotation labels or detailed figure/table decomposition.

The PAWLS interface is similar to tools which augment PDFs for reading or note taking purposes. Along with commercial tools such as Adobe Reader, SideNoter (Abekawa and Aizawa, 2016) augments PDFs with rich note taking and linguistic annotation overlays, directly on the PDF canvas. ScholarPhi (Head et al., 2020) augments the PDF reading experience with equation overlays and definition modals for symbols.

As a PDF specific annotation tool, PAWLS adds to the wider landscape of annotation tools which fulfil a particular niche. SLATE (Kummerfeld, 2019) provides a command line annotation tool for expert annotators; (Mayhew and Roth, 2018) provides an annotation interface specifically designed for cross-lingual annotation in which the annotators do not speak the target language.

Textual annotation tools such as BRAT (Stenetorp et al., 2012), Pubtator (Wei et al., 2013, 2012), Knowtator (Ogren, 2006), or TextANno (Yimam et al., 2014) are recommended for annotations which do not require full PDF context, or for which extension to multi-modal data formats is not possible or likely. We view PAWLS as a complimentary tool to the suite of text based annotation tools, which support more advanced types of annotation and configuration, but deal with annotation on extracted text removed from its originally published format.

In particular, we envisage scholarly document annotation as one of the key use cases for PAWLS, as PDF is a widely used format in the context of scientific publication. Several recently published datasets leave document structure parsing or multimodal annotation to future work. For example, the SciREX dataset (Jain et al., 2020) use the text-only LaTeX source of ArXiv papers for dataset construction, leaving Table and Figure extraction to future work. Multiple iterations of the Evidence Inference dataset (Lehman et al., 2019; DeYoung et al., 2020) use textual descriptions of interventions in clinical trial reports; answering inferential questions using figures, tables and graphs may be a more natural format for some queries.

[5] https://www.tagtog.net/#pdf-annotation

[6] https://labelstud.io/

6 Conclusion

In this paper, we have introduced a new annotation tool, PAWLS, designed specifically with PDFs in mind. PAWLS facilitates the creation of multimodal datasets, due to its support for mixed mode annotation of both text and image sub-regions on PDFs. Additionally, we described several user interface design choices which improve the resulting annotation quality, and conducted a small initial annotation effort, reporting high annotator agreement. PAWLS is released as an open source project under the Apache 2.0 license.

Acknowledgement

We thank the anonymous reviewers for their comments and suggestions, and we thank Doug Downy, Kyle Lo, Lucy Lu Wang for the helpful discussions. This project is supported in part by NSF Grant OIA-2033558.

References

Takeshi Abekawa and Akiko Aizawa. 2016. SideNoter: Scholarly paper browsing system based on PDF restructuring and text annotation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, pages 136–140, Osaka, Japan. The COLING 2016 Organizing Committee.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jay DeYoung, Eric Lehman, Ben Nye, Iain J. Marshall, and Byron C. Wallace. 2020. Evidence inference 2.0: More data, better models.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969.

Andrew Head, Kyle Lo, Dongyeop Kang, Raymond Fok, Sam Skjonsberg, Daniel S. Weld, and Marti A. Hearst. 2020. Augmenting scientific papers with just-in-time, position-sensitive definitions of terms and symbols. *ArXiv*, abs/2009.14237.

Sarthak Jain, Madeleine van Zuylen, Hannaneh Hajishirzi, and Iz Beltagy. 2020. SciREX: A challenge dataset for document-level information extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7506–7516, Online. Association for Computational Linguistics.

Kenneth Joseph, Lisa Friedland, William Hobbs, David Lazer, and Oren Tsur. 2017. ConStance: Modeling annotation contexts to improve stance classification. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1115–1124, Copenhagen, Denmark. Association for Computational Linguistics.

Jonathan K. Kummerfeld. 2019. SLATE: A super-lightweight annotation tool for experts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 7–12, Florence, Italy. Association for Computational Linguistics.

Samuel Läubli, Rico Sennrich, and Martin Volk. 2018. Has machine translation achieved human parity? a case for document-level evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4791–4796, Brussels, Belgium. Association for Computational Linguistics.

Eric Lehman, Jay DeYoung, Regina Barzilay, and Byron C Wallace. 2019. Inferring which medical treatments work from reports of clinical trials. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 3705–3717.

Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, Ming Zhou, and Zhoujun Li. 2020a. TableBank: Table benchmark for image-based table detection and recognition. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1918–1925, Marseille, France. European Language Resources Association.

Minghao Li, Yiheng Xu, Lei Cui, Shaohan Huang, Furu Wei, Zhoujun Li, and M. Zhou. 2020b. Docbank: A benchmark dataset for document layout analysis. *ArXiv*, abs/2006.01038.

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

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

Stephen Mayhew and Dan Roth. 2018. TALEN: Tool for annotation of low-resource ENtities. In *Proceedings of ACL 2018, System Demonstrations*, pages

80–86, Melbourne, Australia. Association for Computational Linguistics.

M. Musen. 2015. The protégé project: a look back and a look forward. *AI matters*, 1 4:4–12.

Philip V. Ogren. 2006. Knowtator: A protégé plug-in for annotated corpus construction. In *HLT-NAACL*.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Hiroyuki Shindo, Yohei Munesada, and Y. Matsumoto. 2018. Pdfanno: a web-based linguistic annotation tool for pdf documents. In *LREC*.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. brat: a web-based tool for NLP-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107, Avignon, France. Association for Computational Linguistics.

Chih-Hsuan Wei, Hung-Yu Kao, and Zhiyong Lu. 2012. Pubtator: A pubmed-like interactive curation system for document triage and literature curation. *BioCreative 2012 workshop*, 05.

Chih-Hsuan Wei, Hung-Yu Kao, and Zhiyong Lu. 2013. Pubtator: a web-based text mining tool for assisting biocuration. *Nucleic Acids Research*, 41.

Seid Muhie Yimam, Chris Biemann, Richard Eckart de Castilho, and Iryna Gurevych. 2014. Automatic annotation suggestions and custom annotation layers in webanno. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 91–96.

Xu Zhong, J. Tang, and Antonio Jimeno-Yepes. 2019a. Publaynet: Largest dataset ever for document layout analysis. *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1015–1022.

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

A Accessing the Demo

A demo video for PAWLS usage is available at https://youtu.be/TB4kzh2H9og, and the demo server can be accessed at https://pawls.apps.allenai.org/. We demonstrate PAWLS' key functionalities using the scientific document labeling task as an example—the label spaces and exemplar documents are configured accordingly—but they can be easily switched to adapt to other types of documents like financial or legal reports. To fully present the capability of PAWLS, no pre-annotation function is used. THe authors demonstrated documents are

Production deployments of PAWLS use Google Authentication to allow users to log in. The demo server is configured to allow access to all non-corp gmail accounts, e.g any email address ending in "@gmail.com". For this public demo, no personal information and annotations will be collected from this server, as it is read-only. Please feel free to create a one-off account if you prefer not to use a personal gmail. If you cannot log in, please try again using an incognito window which will ensure gmail cookies are not set.