PDFAnno: a Web-based Linguistic Annotation Tool for PDF Documents

Hiroyuki Shindo[1], Yohei Munesada, Yuji Matsumoto[1]
[1]Graduate School of Information and Science
Nara Institute of Science and Technology
8916-5, Takayama, Ikoma, Nara, 630-0192, Japan
shindo@is.naist.jp, y.munesada@gmail.com, matsu@is.naist.jp

Abstract

We present PDFAnno, a web-based linguistic annotation tool for PDF documents. PDF has become widespread standard for various types of publications, however, current tools for linguistic annotation mostly focus on plain-text documents. PDFAnno offers functions for various types of linguistic annotations directly on PDF, including named entity, dependency relation, and coreference chain. Furthermore, for multi-user support, it allows simultaneous visualization of multi-user's annotations on the single PDF, which is useful for checking inter-annotator agreement and resolving annotation conflicts. PDFAnno is freely available under open-source license at https://github.com/paperai/pdfanno.

Keywords: text annotation, annotation tool, pdf

1. Introduction

Gold standard annotations for texts are a prerequisite for training and evaluation of statistical models in Natural Language Processing (NLP). Since human annotation is known as one of the most costly and time-consuming tasks in NLP, an easy-to-use and easy-to-manage annotation tool is highly required for cost effective development of gold standard data.

Currently, general-purpose linguistic annotation tools such as BRAT (Stenetorp et al., 2012) and WebAnno (Yimam et al., 2013) only support text documents. Some commercial software packages provide annotation functions for PDF, however, they lack a function of *relation* annotation suitable for dependency relation and coreference chain.

Since PDF has become widespread standard for many publications, a linguistic annotation tool for PDF is strongly desired for knowledge extraction from PDF documents. For example, previous work has developed an annotated corpus for coreference resolution on scientific papers (Panot et al., 2014; Schafer et al., 2012; Steven et al., 2008). In their work, PDF articles are converted to plain-text format using OCR software, then import them to a text annotation tool. As pointed out in the literature, OCR errors are present in the data and they need to clean up the text by viewing the associated PDF file. This motivates us to develop a new annotation tool that can directly annotate on PDF.

There are two types of annotation processes for creating an annotated text from a PDF file as shown in Figure 1. One is to convert the PDF into plain text or HTML format, then annotate it using a text annotation tool, as in the previous work. Another one is to annotate the PDF directly, then convert the annotated PDF into plain text format. Even if annotated plain-text is eventually necessary, the latter one has at least two benefits. First, PDF is often much more readable for annotators than plain text since it is well-structured with sections and paragraphs. This helps us maintain annotation quality and consistency. Second, the annotations become insensitive to OCR errors. That means, if high-quality OCR software was developed later, we can switch the OCR software for converting it to plain text without modifying annotations.

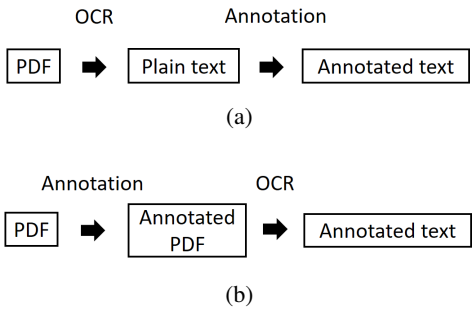


Figure 1: Annotation flows for PDF documents. (a) convert PDF into a plain text, then annotate it. (b) annotate PDF file, then convert it into a text file.

In this work, we present PDFAnno, a general-purpose linguistic annotation tool for PDF documents. PDFAnno offers functions for various types of annotations in a web browser, including named entity, dependency relation, and coreference chain. It requires no installation effort and can be used offline. Furthermore, for multi-user support, it allows simultaneous visualization of multiple annotations on the single PDF, which is useful for checking inter-annotator agreement and resolving annotation conflicts. We also implement a server-side program which converts annotated PDF to XML format by using our PDF parser. The automatic parsed results can be visualized as annotations in the PDFAnno viewer.

We show two case studies of PDFAnno: relation annotation for materials science papers and coreference annotation for ACL anthology papers. In both cases, we observe that the PDF-based annotation has a clear advantage over the text-based annotation in terms of annotation usability.

2. Related Work

In NLP community, a number of annotation tools for text documents have been developed so far (Bontcheva et al., 2010; Muller and Strube, 2006; Stenetorp et al., 2012; Yimam et al., 2013).

BRAT (Stenetorp et al., 2012) is a well-known web-based tool for linguistic annotation and visualization. It is implemented using a client-server architecture in Python and supports rich structured annotation for a variety of NLP tasks. However, it is targeted to annotate text documents.

GATE Teamware (Bontcheva et al., 2010) is a web-based management platform for collaborative text annotation and curation. It is mostly web-based, but the annotation is carried out with the local software.

WebAnno (Yimam et al., 2013) is also a web-based annotation tool which supports a wide range of linguistic annotations. WebAnno has unique characteristics in that it has advanced features for project and user management with monitoring tools. It supports various types of text format including plain text and CoNLL format. However, since WebAnno visualization frontend is built on BRAT, it is also impossible to make annotation directly on PDF.

For PDF annotation, there are many commercial products such as Adobe Acrobat, PDF Annotator[1], and A.nnotate[2], which basically support text highlighting and adding notes and comments on PDF. However, these tools are not intended to be used for linguistic annotation, thus these lack annotation types suitable for linguistic phenomena such as dependency relation and coreference chain. On the other hand, PDFAnno supports such *relation* annotation and multi-user annotation. Furthermore, it is open-source and extensible with annotation API for external programs.

3. Features

3.1. User Interface

Figure 2 shows a screenshot of PDFAnno user-interface. PDFAnno is a browser-based application and built entirely using standard web technologies. For rendering a PDF document, we use PDF.js[3], a web-based PDF viewer built with HTML5. PDF.js is a default built-in PDF viewer in Firefox, thus it offers a familiar environment to annotators for PDF operations such as zoom, search, and print.

We implemented annotation layers on PDF.js with JavaScript. Currently, PDFAnno supports three types of annotations: span, rectangle, and relation. The use cases of these annotations are shown later.

Span Span is the most basic type of annotation to mark text spans in PDF. For each span, users can assign a text label. For part-of-speech annotation, annotators mark a text span by selecting it with the mouse dragging and assign a part-of-speech tag. Similarly, the span annotation can be used for named entities. PDFAnno enables autocompletion for text label fields, thus annotators can fill in long words by typing only a few characters. In the implementation, the span is preserved as the position: (x, y, width, height) and the page number where x and y describe the coordinates of the top left point of the span, and width and height describe its dimensions.

Rectangle Rectangle is a type of annotation to select a region in PDF. This is intended to be used for annotation of non-text objects such as tables and figures. This is not directly related with text annotations, however, we provide the rectangle function for creating training data for region detection of figures and tables, which is useful for knowledge extraction from scientific papers.

Relation Relation is a type of annotation to make a connection between annotated objects. PDFAnno provides three kinds of binary relations: one-way, two-way and undirected arrows. The one-way arrow can be used for annotation of word and named entity dependencies, the two-way arrow used for bidirectional relation between objects, and the undirected arrow used for coreference chain and grouping multiple annotations. As in the case of span annotation, users can assign a text label to each relation.

In PDFAnno, an identifier (ID) is assigned to each annotation object. The relation is preserved as a pair of annotation IDs and its direction.

3.2. System Architecture

The overall system architecture of PDFAnno is shown in Figure 3. PDFAnno is a simple client-side application in a web browser. It loads PDF.js for rendering the user-specified PDF, then provides functions for adding annotation layers on PDF.js. For multi-user annotation, we assume to use an online storage service to share PDF documents and annotation files between annotators. Every user who has a permission to access the common online storage can load the shared annotation files with PDFAnno.

Our system architecture contrasts with that of BRAT and WebAnno in that most annotation functions and settings in PDFAnno can be accessed and controlled on a client-side. In BRAT and WebAnno, the server fully manages datasets and user account settings. However, we believe that an online storage service can be substituted for most of such server-side functions.

3.3. PDF to XML Converter

While the annotation functions in PDFAnno require no communication with the server, we optionally provide server-side programs for parsing and converting the annotated PDFs into XML. The server-side programs first extract text and positional information from the PDF with Apache PDFBox[4], then convert it to XML format with the user's annotation information. Currently the PDF to XML conversion is performed with our rule-based method, however, we plan to replace it with machine learning approach to reduce the conversion errors.

3.4. Annotation File

In PDFAnno, user's annotation is preserved separately from the original PDF file, and downloadable anytime as a text file following TOML format[5]. Compared with JSON and YAML format, TOML is easy to read and easy to write by

[1] https://www.pdfannotator.com/

[2] http://a.nnotate.com/

[3] https://mozilla.github.io/pdf.js/

[4] https://pdfbox.apache.org/

[5] https://github.com/toml-lang/toml

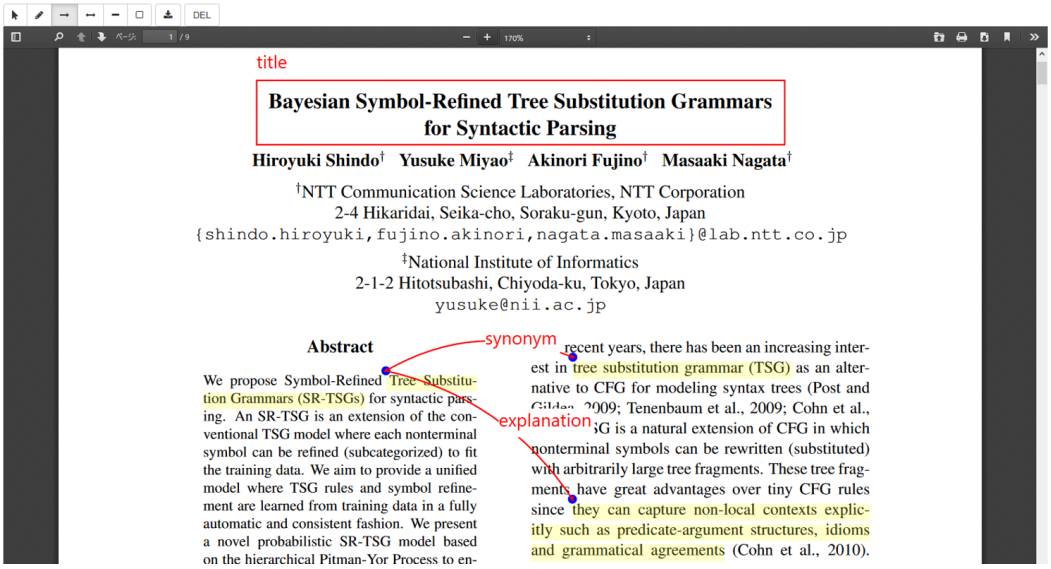


Figure 2: Screenshot of the PDFAnno user-interface, showing example annotations of text span, rectangle, and relation.

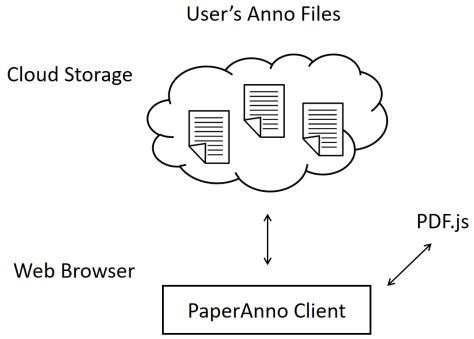


Figure 3: System architecture of PDFAnno.

```
[1]
type = "span"
page = 1
position = [[95.818,252.977,181.761,10.909]]
label = "label-1"

[2]
type = "span"
page = 2
position = [[323.863,213.988,230.715,11.590]]
label = "label-2"
```

Figure 4: Example annotation file for PDFAnno.

both human and computers. Figure 4 shows an example annotation file (anno file) with two spans, one rectangle and one relation. Span is represented as a page number, positions, and its label, while relation contains a page number, connection type, two identified spans and its label.

3.5. Support for Multi-User Annotation

For multi-user annotation, PDFAnno user-inferface accepts to load multiple annotation files and renders these annotations on the single PDF with distinct colors one another. Figure 5 shows an example of rendering multiple annotations. A user can perform annotation work while referring to other annotation file, which helps users check inter-annotator agreement and resolve annotation conflicts.

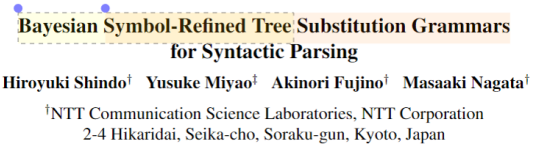


Figure 5: Example of rendering multiple annotations on a single PDF. In the example, one annotator marked "Bayesian Symbol-Refined Tree" as a span but another annotator marked "Symbol-Refined Tree Substitution Grammars".

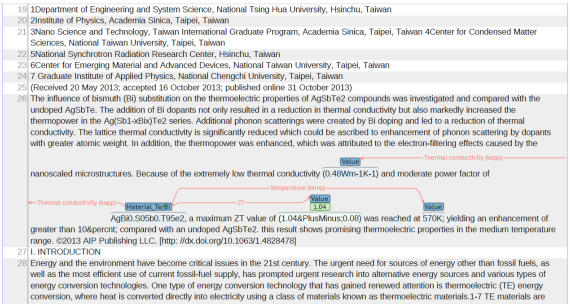
4. Case Studies

4.1. Information Extraction from Scientific Papers

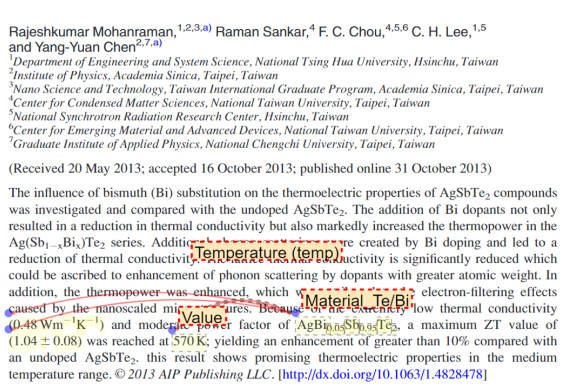
PDFAnno is well-suited for creating gold annotation data for information extraction (IE) from scientific papers. To test the effectiveness of PDFAnno, we conducted an annotation experiment for IE from scientific papers. In particular, we asked experts in materials science to annotate selected papers in their field with material names and their physical quantities such as temperature and thermal conductivity. In our annotation guideline, material names and their physical quantities are marked as *span*, and these are connected with each other as *relation*.

We provided the selected papers as both plain-text format and PDF to the annotators, and compared text-based and PDF-based annotation with respect to annotation time and quality. For text-based annotation, the plain texts were extracted from PDFs with Poppler[6] and annotation work was performed using WebAnno (Yimam et al., 2013) . For PDF-based annotation, we provided our PDFAnno to the annotators. Figure 6 shows the screenshots of WebAnno and PDFAnno for scientific papers.

[6]https://poppler.freedesktop.org/



(a) Text-based annotation using WebAnno.



(b) PDF-based annotation using PDFAnno.

Figure 6: Comparison of text-based and PDF-based annotation on materials science paper.

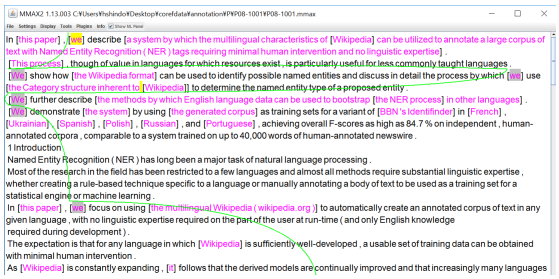
As we expected, we observed that the annotators completed the annotation work much more rapidly in PDF-based annotation compared with text-based one. The main reason is that the structure of section and paragraph of papers is lost in plain-text format, which makes annotators read and understand the paper much more difficult than the original PDF. Actually, every annotator performed the work of text-based annotation while referring to the original PDF. On the other hand, such problems did not occur in the PDF-based annotation.

Although further research and evaluation is required to establish the benefits of PDFAnno, we view that the direct annotation on PDF with PDFAnno is a promising direction for annotating scientific papers. The PDFs and annotation files in this experiment are freely available at PDFAnno website.

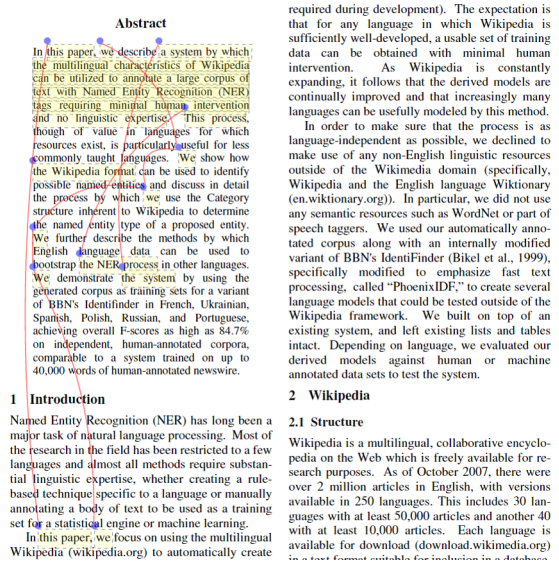
4.2. Coreference Annotation

To further test the effectiveness of PDFAnno, we annotated selected papers from ACL anthology with coreference chains. Since previous work (Schafer et al., 2012) on coreference annotation for ACL papers has released gold annotated data as MMax2 format, we actually transferred the gold annotations to PDF using PDFAnno. In the previous work (Schafer et al., 2012), the texts are extracted from PDF using a commercial OCR software, then coreference chains are annotated on the texts using MMax2 annotation tool (Muller and Strube, 2006). Figure 7 shows an example of coreference annotation on P08-1001 paper using MMax2 and PDFAnno.

The main problem of text-based annotation is that there exists multiple versions of text corpus extracted from PDFs.



(a) Text-based coreference annotation using MMax2.



(b) PDF-based coreference annotation using PDFAnno.

Figure 7: Examples of coreference annotations on plain-text and PDF for P08-1001 paper.

For example, two versions of ACL anthology reference corpus have been released (Steven et al., 2008). In addition, the previous work uses their own OCR software for text extraction from PDF since the ACL reference corpus converted with PDFBox contained many extraction errors. Therefore, it is difficult to establish consistency with multiple annotations on ACL papers due to the differences between text extraction tools. As in the case of materials science papers, PDF is more readable and easy to annotate with coreference chain. We have noticed that the main drawback of PDF annotation is that the line space is not adjustable, which makes annotation work difficult when there are too many annotation objects on PDF. We will leave the problem of rendering many annotation objects for future work.

5. Conclusions

We present PDFAnno, a web-based linguistic annotation tool for PDF documents. PDFAnno provides functions for a variety of linguistic annotation for PDF documents. It is a simple and easy-to-use client-side browser application. Furthermore, it allows simultaneous visualization of multi-user's annotations on the single PDF, which is useful for checking inter-annotator agreement and resolving annotation conflicts. Future work involves adding more advanced project management system and XML/HTML support.

6. Acknowledgements

This work was partly supported by JSPS KAKENHI Grant Number 15K16053 and JST CREST Grant Number JPMJCR1513, Japan..

7. Bibliographical References

Bontcheva, K., Cunningham, H., Roberts, I., and Tablan, V. (2010). Web-based collaborative corpus annotation: Requirements and a framework implementation. In *New Challenges for NLP Frameworks workshop at LREC*, pages 197–214.

Muller, C. and Strube, M. (2006). Multi-level annotation of linguistic data with MMAX2. In *Corpus Technology and Language Pedagogy: New Resources, New Tools, New Methods*.

Panot, C., Akiko, A., and Yuka, T. (2014). Corpus for Coreference Resolution on Scientific Papers. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC)*, pages 26–31.

Schafer, U., Spurk, C., and Steffen, J. (2012). A Fully Coreference-annotated Corpus of Scholarly Papers from the ACL Anthology. In *Proceedings of the International Conference on Computational Linguistics (COLING) 2012*, pages 1059–1070.

Stenetorp, P., Pyysalo, S., Topic, G., Ohta, Tomoko Ananiadou, S., and Tsujii, J. (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 (EACL)*, pages 102–107.

Steven, B., Robert, D., Bonnie, D., Bryan, G., Mark, J., Min-Yen, K., Dongwon, L., Brett, P., Dragomir, R., and Yee Fan, T. (2008). The ACL Anthology Reference Corpus: A Reference Dataset for Bibliographic Research in Computational Linguistics. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC)*, pages 1755–1759.

Yimam, S. M., Gurevych, I., Eckart de Castilho, R., and Biemann, C. (2013). WebAnno: A Flexible, Web-based and Visually Supported System for Distributed Annotations. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations (ACL)*, pages 1–6.