CiteAssist: A System for Automated Preprint Citation and BibTeX Generation

Lars Benedikt Kaesberg
Georg-August University / Göttingen
l.kaesberg@stud.uni-goettingen.de

Terry Ruas
Georg-August University / Göttingen
ruas@uni-goettingen.de

Jan Philip Wahle
Georg-August University / Göttingen
wahle@uni-goettingen.de

Bela Gipp
Georg-August University / Göttingen
gipp@uni-goettingen.de

Abstract

We present *CiteAssist*, a system to automate the generation of BibTeX entries for preprints, streamlining the process of bibliographic annotation. Our system extracts metadata, such as author names, titles, publication dates, and keywords, to create standardized annotations within the document. CiteAssist automatically attaches the BibTeX citation to the end of a PDF and links it on the first page of the document so other researchers gain immediate access to the correct citation of the article (see Figure 1). This method promotes platform flexibility by ensuring that annotations remain accessible regardless of the repository used to publish or access the preprint. The annotations remain available even if the preprint is viewed externally to CiteAssist. Additionally, the system adds relevant related papers based on extracted keywords to the preprint, providing researchers with additional publications besides those in related work for further reading. Researchers can enhance their preprints organization and reference management workflows through a free and publicly available web interface[1].

1 Introduction

A common challenge in academic research is tracing citational information for preprints. Many researchers experience a low citational impact because their citational information is not accessible easily together with the preprint. This phenomenon is often evident on platforms like ResearchGate, where numerous articles are downloaded hundreds of times but receive only a few citations (Bollmann and Elliott, 2020). This discrepancy between reads and citations highlights a difference in how information is consumed and credited (a pattern also observed in contexts such as social media impressions versus likes) and specifically in citation analysis (Meho, 2006; Wahle et al., 2023, 2024).

(a) The first preprint page with a generated annotation button.

Citation for this Paper

Click here for the Online Version

```
@article{citeassist,
  author={Example Example},
  title={CiteAsisst: A System for Automated Preprint Citation and BibTeX
  Generation},
  journal={ACS},
  volume={1},
  number={1},
  pages={13},
  year={2024},
  month={01}
}
```

(b) A generated citation box on the last page of the preprint.

Figure 1: Document enhanced with CiteAssist. A detailed version available in Appendix C).

An imbalance between reading and citing papers can have many reasons: researchers may judge a focal work as irrelevant to them, not exciting, not sound, or simply forgetting. Yet, a key reason is the accessibility of citational information (Bollmann and Elliott, 2020). If citation details are consistent and easily accessible, citations are generally higher (Ale Ebrahim et al., 2013; Bollmann and Elliott, 2020). Long review processes and publication timelines can slow scientific progress and hinder researchers from proposing state-of-the-art techniques. Thus, preprints are often used in modern scientific research. Public available preprints promote the rapid exchange of ideas and results before peer-reviewed publications are available, accelerating the spread of knowledge and driving scientific progress (Puebla et al., 2022). With the fast growth in the number of preprints, researchers are faced with an ever-growing sea of scientific information (Abdill and Blekhman, 2019a). The sheer volume of preprints make efficient retrieval challenging.

---

[1] https://preprint.larskaesberg.de

So far, there is no easy way to annotate preprints with citational information. While some repositories allow for archiving preprints (e.g., Zenodo, figshare, arXiv), the citation details of a paper are often only available through external infrastructure (e.g., publishers, scholarly search engines). To the best of our knowledge, no publicly available system provides the citation details of a paper within the PDF itself in simple format. Furthermore, we lack a common way to add recommended literature on related projects within preprints for further reading about a topic over the scope of related work sections. Those could be related ideas from other areas, books, or talks that complement a paper but are not directly discussed in related work.

We present CiteAssist, a system designed to simplify the creation and distribution of citation annotations by automatically generating and appending annotations to the preprint. This is achieved either by generating a new PDF with the annotation or by generating a LaTeX file that can be integrated directly into the source document. CiteAssist provides researchers with a convenient way to enhance their preprint by directly attaching citation information to it, improving the reference management workflow. These annotations capture metadata such as author names, publication details, and abstracts, enabling researchers to create bibliographic records for their preprints. The system also allows users to add custom metadata (e.g., keywords) to the BibTeX annotation. By automating this process, researchers can save time that would otherwise be spent on manual annotation. Our system is platform-independent as it incorporates all citation details within the paper. The annotation is attached to the preprint; thus, it can be published at any preprint repository. Unlike platform-specific solutions, our system accommodates various sources, making it accessible and useful to researchers in various fields. Figure 1 (a) shows the first page containing the button to the BibTex entry and (b) the generated annotation itself. In addition, CiteAssist suggests related papers at the end of an article to complement a traditional "related work" section. Related work sections focus on considering the proposed work and crediting previous ideas. However, they are bound to the researcher's awareness of these works, which may include *citation amnesia*, i.e., not citing good and relevant work (Singh et al., 2023; Wahle et al., 2024). CiteAssist mitigates this issue by suggesting additional literature on poten-

tially forgotten or overlooked works. Researchers can manually add literature they wish to suggest to readers as additional resources.

2 Related Work

2.1 Preprint Platforms

Multiple platforms allow researchers to publish their preprints to repositories, e.g., arXiv, bioRxiv, ResearchGate, etc (Abdill and Blekhman, 2019b; Puebla et al., 2022). While these platforms support scholarly communication, they rely on external infrastructure that may not be accessible or familiar to all researchers. Some may prefer to upload their work to personal websites or may not know how to use LaTeX. Our system bridges these gaps, making preprint submission and formatting easier by providing a simplified interface that allows researchers to enter citation details directly into the paper.

2.2 Annotation Generation Tools

Annotation generation tools automatically extract metadata from scholarly articles and generate annotations. In this section, we discuss three relevant annotation generation tools relevant to CiteAssist, namely CERMINE (Tkaczyk et al., 2015), Neural ParsCit (Prasad et al., 2018), and Grobid (Lopez, 2008–2024).

CERMINE is a tool for extracting metadata from scholarly articles. It employs machine learning techniques like SVM, CRF, and clustering to parse PDF documents and extract structured information, including authors, titles, abstracts, and references (Tkaczyk et al., 2015). CERMINE can extract metadata from diverse scholarly articles, even with complex layouts and formatting. It generates structured XML output, creating annotations in various formats, such as BibTeX or CSL (Tkaczyk et al., 2015). Unlike CERMINE, which primarily extracts metadata for structured XML output, our system generates BibTeX annotations for scientific citation and proactively suggests related papers, increasing the visibility of preprints in the scientific literature. CiteAssist integrates related literature, which enhances the annotation process and the researcher's ability to explore related works. Additionally, while CERMINE outputs structured XML, our system directly attaches the generated BibTeX annotation to the paper, providing a more streamlined workflow for researchers.

Neural ParsCit focuses on extracting citation information from scientific articles. While its pri-

mary purpose is to parse and analyze citations, Neural ParsCit can also generate annotations using the extracted information. It identifies and extracts key metadata elements, including authors, titles, and publication venues. The tool focuses on citations, ensuring accurate extraction of citation-related metadata (e.g., identifying a citation string such as "Doe, J., Smith, A., & Johnson, K. (2020). Title of the article. 10(2), 123-134"). However, Neural ParsCit does not capture information like the publication date or journal (Tkaczyk et al., 2015). Therefore, additional processing or integration with other tools may be required to generate annotations for preprints (Prasad et al., 2018). In contrast, our tool provides a complete solution for Bib-TeX annotation generation and the integration of related papers. Unlike Neural ParsCit, CiteAssist leverages user input and keyword-based searches to generate BibTeX citations that are directly linked in the paper PDF and to include relevant literature.

Grobid provides an API for extracting metadata from papers, using both a small CRF model for quick results and a larger BidLSTM-CRF model for higher accuracy, though at greater computational cost (Lopez, 2008–2024). While it shares capabilities with CERMINE, like extracting titles, authors, and abstracts, Grobid does not capture journal information (Lopez, 2008–2024). Although Grobid's self-hostable API via Docker makes it an excellent component for CiteAssist to leverage for more advanced data extraction directly from the PDF text, it lacks the advanced PDF enhancement features that CiteAssist offers, such as direct annotation with BibTeX and the suggestion of related papers.

3 System Architecture

The system architecture of CiteAssist consists of a frontend, backend, and database. The frontend handles the user interaction and allows users to upload a PDF file, which is parsed by our method and in the backend with Grobid. We merge and display the results to the user for correction, if necessary. Furthermore, users can add related papers, either provided by the backend via keyword matching or with a provided DOI or arXiv id. The system generates a new PDF or LaTeX file with an attached BibTeX annotation and related papers. After user approval, these results are uploaded to the backend, where the PDF is stored locally and metadata forwarded to the database. We refer to Appendix A for a more detailed system architecture description.

4 Preprint Annotation Workflow

This section shows a practical example of how researchers can annotate preprints using CiteAssist. The workflow begins when a user uploads a preprint in PDF format to the CiteAssist homepage via a drag-and-drop interface or a file selection dialogue as seen in Figure 2. Users are then asked to review (Figure 3), edit potentially incorrect information, or add other relevant metadata (e.g., keywords). Users can select relevant suggested keywords and add new ones if desired. The system enhances the compatibility of these keywords through lemmatization. With refined keywords, the backend searches the database for related preprints and displays these on the frontend for user review and selection, as seen in Figure 5. CiteAssist compiles this information and generates a BibTeX annotation containing the edited metadata and selected related papers (Figure 11). The compiled annotation is attached at the end of the preprint. On the title page of the uploaded article, it adds a button that links to the annotation at the bottom, as shown in Figure 10. Users can download the final annotated PDF version or generate LaTeX files to include it directly in the source document. With user consent, the backend stores the annotated PDFs for future access. Users can view these files online, share them with colleagues, or download them again.

5 BibTeX Annotation Generation

We combine the PDF-LIB [2] library, Grobid, and our custom information extraction method to extract BibTeX annotation information. Our process can be divided into five steps:

1. **PDF Parsing:** The PDF file is parsed using the PDF-LIB library, which allows us to access the document's metadata (e.g., title, author, creation date) if available and saved in the PDF.

2. **Text Extraction:** Our custom function extracts the text content from the PDF file. This includes the text from the first page, which contains information such as the title, author, and more. The rest of the text is used for keyword extraction.

3. **Metadata Extraction:** We extract various metadata elements necessary for the BibTeX annotation. These elements include:

   - **Article Type:** The publication category for the paper (e.g., proceedings, journal). Set to "article" as a default.

---

[2] https://github.com/Hopding/pdf-lib

- **Author:** Author name obtained from the PDF metadata or extracted from the text.
- **Title:** Title of the article from the first page's text. If unavailable, the PDF's title or the filename is used as a default.
- **Pages:** The total number of pages in the PDF document.
- **Date:** Publication date from the PDF metadata. If unavailable, the current date is used.

4. **Grobid:** Simultaneously, the frontend sends the PDF to the backend, which then forwards it to Grobid to extract relevant information, including the title, author, publication date, and keywords.

5. **Constructing the BibTeX Annotation:** We merge the information extracted from Grobid and our own. In cases where neither alternative provides the information, we resort to a fallback value that can be edited by the user. Next, we construct the BibTeX annotation following the required format from the extracted metadata. The annotation includes key-value pairs for elements such as author, title, journal, volume, pages, year, and DOI.

The combination of `PDF-LIB`s metadata extraction, Grobid, and our custom metadata extraction function allows us to generate BibTeX annotations for preprints shown in Figure 1b (a figure outlining the process can be found in Figure 9).

6 Integration of Related Papers

The integration of related papers is designed to augment the research experience by providing users with access to literature that, while not directly cited as related work, can indicate the reader to other possible related materials. We use keyword-based searches to retrieve a selection of papers that serve as complementary references, offering broader insights into the subject area without overlapping with the core related work already cited. This task involves the following functionalities:

- **Keyword Extraction:** The preprint's whole text is processed by our algorithm to extract the five most relevant keywords. This is done by removing stop words, applying lemmatization, and selecting based on frequency of occurrence. This functionality is currently only available for English texts. If the author has provided keywords, Grobid will also extract these.
- **User Input:** Users provide additional keywords or tags associated with the preprint being annotated. These keywords act as search criteria for finding related papers.
- **Search Query:** The system applies lemmatization to the provided keywords, converting them to their base forms for better matching. This improves the effectiveness of the query in retrieving related papers. The query returns the five most relevant papers, ranked by the number of matching keywords.
- **Selection and Integration:** Users can choose specific preprints from the provided suggestions to add to their annotation. They can also add new papers using either a DOI or arXiv ID. These selected papers are integrated into the annotation generation process.

Integrating related papers into the annotation enhances the paper by providing users access to related resources that contribute insights, broader the existing related work section and further context to their initial findings, and save researchers time and effort from authors and readers.

7 Final Considerations

In this work, we introduced CiteAssist, a system designed to streamline the creation and distribution of BibTeX annotations within preprints. CiteAssist standardizes the reference management process with the help of automatic generation and integration of annotations, either through a newly generated PDF or directly within a LaTeX source document. A button on the title page linking to the annotation improves the usability and accessibility of the citation information of the paper. Our system enhances the usability of preprints by embedding citation details and ensures that the annotations are readily accessible, regardless of the publication repository used. Additionally, CiteAssist offers suggestions for related papers, enriching the scope of resources available to researchers about a given publication. CiteAssist's development reflects the growing importance of digital tools in academic research. By allowing researchers to easily include their citational information into preprints, we see a direct impact on giving authors proper credit and to be cited correctly with the publication venue as opposed to the preprint server. Future enhancements could include recommending similar papers based on semantic similarity and providing a more comprehensive representation of the paper's content.

Limitations

Although CiteAssist has enhanced the preprint annotation process, it still has room for improvement. Implementing stopword removal for several languages during keyword extraction may improve the relevancy of the discovered keywords. Furthermore, integrating language models to extract keywords directly can take advantage of recent natural language processing advances (e.g., large context length (Ma et al., 2024)) to produce more accurate results. Incorporating large language models into CiteAssist can improve metadata extraction accuracy and cover a wider range of document layouts and formats, increasing the tool's overall efficiency. We plan to improve CiteAssist's UI and usability through user feedback sessions. By gathering insights directly from users, we will make iterative improvements to enhance the system's intuitiveness and functionality, ensuring it effectively meets researchers' needs.

References

Richard J Abdill and Ran Blekhman. 2019a. Meta-research: Tracking the popularity and outcomes of all biorxiv preprints. *eLife*, 8:e45133.

Richard J Abdill and Ran Blekhman. 2019b. Meta-research: Tracking the popularity and outcomes of all biorxiv preprints. *eLife*, 8:e45133.

Nader Ale Ebrahim, Hadi Salehi, Mohamed Embi, Farid Tanha, Hossein Gholizadeh, Seyed Mohammad Motahar, and Ali Ordi. 2013. Effective strategies for increasing citation frequency. *International Education Studies*, 6:93–99.

Marcel Bollmann and Desmond Elliott. 2020. On forgetting to cite older papers: An analysis of the ACL Anthology. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7819–7827, Online. Association for Computational Linguistics.

Patrice Lopez. 2008–2024. Grobid. `https://github.com/kermitt2/grobid`.

Xuezhe Ma, Xiaomeng Yang, Wenhan Xiong, Beidi Chen, Lili Yu, Hao Zhang, Jonathan May, Luke Zettlemoyer, Omer Levy, and Chunting Zhou. 2024. Megalodon: Efficient llm pretraining and inference with unlimited context length.

L. I. Meho. 2006. The rise and rise of citation analysis.

Jakob Nielsen. 1994. Enhancing the explanatory power of usability heuristics. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '94, page 152–158, New York, NY, USA. Association for Computing Machinery.

Animesh Prasad, Manpreet Kaur, and Min-Yen Kan. 2018. Neural parscit: A deep learning based reference string parser. *International Journal on Digital Libraries*, 19:323–337.

Iratxe Puebla, Jessica Polka, and Oya Rieger. 2022. *Preprints: Their Evolving Role in Science Communication.* Against the Grain (Media), LLC.

Janvijay Singh, Mukund Rungta, Diyi Yang, and Saif Mohammad. 2023. Forgotten knowledge: Examining the citational amnesia in NLP. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6192–6208, Toronto, Canada. Association for Computational Linguistics.

Dominika Tkaczyk, Pawel Szostek, Mateusz Fedoryszak, Piotr Jan Dendek, and Lukasz Bolikowski. 2015. Cermine: automatic extraction of structured metadata from scientific literature. *International Journal on Document Analysis and Recognition (IJDAR*, 18(4).

Jan Wahle, Terry Ruas, Mohamed Abdalla, Bela Gipp, and Saif Mohammad. 2023. We are Who We Cite: Bridges of Influence Between Natural Language Processing and Other Academic Fields. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12896–12913. Association for Computational Linguistics.

Jan Philip Wahle, Terry Ruas, Mohamed Abdalla, Bela Gipp, and Saif M. Mohammad. 2024. Citation Amnesia: NLP and Other Academic Fields Are in a Citation Age Recession. volume abs/2402.12046.

A Detailed System Architecture

The system architecture of CiteAssist is designed to provide researchers with a responsive and user-friendly experience. It consists of a frontend, a backend, and a database, each component fulfilling specific functionalities to enable efficient and semi-automatic annotation generation.

A.1 Frontend and UI/UX Design

Our system's front end handles the user's interaction with the system. Researchers can upload their preprint in PDF format, and the system extracts metadata such as the preprint's title, author, year, and page numbers to generate the BibTeX annotation (see Figure 2). This process is done both in the frontend and in the back end with the help of Grobid, and then results are merged to achieve the most complete set of automatically extracted metadata. The data is displayed to the user, offering an overview of the preprint's details (see Figure 3). To guarantee accuracy, users can edit this extracted data as needed. Also, users can directly copy and paste a BibTeX they already have (for example from their venue) and the tool extracts the necessary metadata from that BibTeX. Once verified, the frontend sends the metadata, along with the

user-specified keywords, to the backend.

Beyond metadata extraction, the frontend is responsible for generating the final PDF file. It receives related papers from the backend and integrates them with the BibTeX annotation into a new PDF document. The resulting PDF combines the preprint with the annotation and related literature. The system can also create a LaTeX file with the annotation that can be imported into the LaTeX source document. This makes it convenient to submit to source code-based repositories such as arXiv.

Design Principles: We dedicated our UI/UX design to user-centric principles. To support our design decisions, we looked at reputable UI/UX guidelines, such as those proposed by Nielsen (1994). Adopting these universally acknowledged design principles ensures our tool meets industry standards. We prioritized:

- **Simplicity:** We aimed for a clean and uncluttered interface, prioritizing simplicity to ensure ease of use. To achieve this, we employed a minimalist design approach, using a monochromatic color scheme and limiting the number of interactive elements visible at any one time. This approach reduces cognitive load and allows users to focus on task-relevant information.

- **Consistency:** We maintained consistent visual and interaction design throughout the tool to provide a cohesive user experience. We adopted a standard color palette and uniform typography across all pages. Navigation elements are consistently placed in the same locations across screens.

- **Intuitiveness:** We focused on creating an intuitive user interface, where users can easily understand the tool's functionalities without extensive instructions. Labels are descriptive and positioned adjacent to their respective controls, tooltips are provided for complex functions.

- **Responsive Design:** We adopted a responsive design approach, ensuring that the UI adapts seamlessly to different screen sizes and devices. We used a flexible grid layout to adjust the layout dynamically, providing an optimal viewing experience on tablets, phones, and desktops.

User Interface Components: The user interface components are designed to address user tasks

through distinct components.

- **Upload Interface (Figure 2):** Users are presented with an interface to upload their preprint PDF files. The upload interface includes drag-and-drop functionality and instructions to guide users through the process.

- **Metadata Editing (Figure 3):** Users have the ability to edit the extracted metadata of the preprint, such as title, author, etc. A user-friendly form-based interface is employed, allowing users to make necessary modifications easily.

- **Keyword Input (Figure 3):** Users can input relevant keywords or tags associated with the preprint to search for related papers. The keyword input interface supports both manual entry and suggestions based on extracted keywords.

- **Related Papers Display (Figure 5):** The interface presents related papers retrieved based on the user's keywords. Users have the option to delete papers they consider irrelevant, or to add related papers with their Digital Object Identifier (DOI) or arXiv id.

User Experience Enhancements: We took steps to enhance user interaction with our system. This included:

- **Ease of Navigation:** The system provides seamless navigation between its different sections. To enhance ease of navigation, clear and intuitive navigation elements, such as a persistent top menu, have been implemented.

- **Feedback and Error Handling:** The tool provides error messages to guide users in case of incorrect inputs. Clear error messages, tooltips, and validation checks have been implemented to assist users in understanding and resolving issues effectively.

- **Visual Appeal:** The UI utilizes visually appealing design elements, such as appropriate color schemes, typography, and layout, to provide an engaging and aesthetically pleasing experience for users.

A.2 Backend

The backend component of our system searches for related preprints based on provided keywords. The primary role of the backend is to accept new preprint metadata from the frontend and store them in the database for future reference. It acts as a

bridge, receiving the preprint information and passing it on to the database without directly processing the preprint content.

In addition, the backend supports querying the database of previously uploaded preprints for related works based on keywords provided by the user. When the frontend sends a request for related preprints, the backend performs the necessary database searches to identify relevant preprints.

The backend component of our tool also stores the actual PDF files of the preprints. This storage capability is crucial, as it allows for an easy retrieval of the original documents. When a user requests to view or download a specific preprint, the backend serves the corresponding PDF file back to the frontend.

A.3 Database

The database component of our system is responsible for storing and managing information about preprints. It serves as a central repository for preprint metadata, enabling efficient data storage and retrieval. The database stores information about each preprint, including its title, author, DOI, URL to the publication, year of publication, and keywords associated with it.

To ensure secure and reliable data storage, our system uses PostgreSQL as the database management system. PostgreSQL offers robust features such as data integrity, efficient indexing, and transaction support, making it well-suited for managing preprint information.

B CiteAssist Screenshots
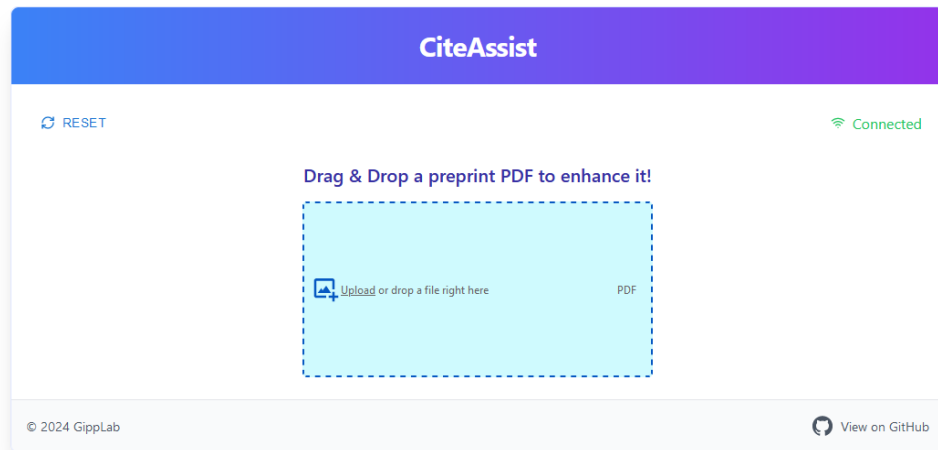
Figure 2: CiteAssist Homepage.

Figure 3: CiteAssist Edit Metadata Page.

Figure 4: CiteAssist Advanced Edit Metadata Page.

Figure 5: CiteAssist Related Paper Page.

Figure 6: CiteAssist Parse BibTeX Page.

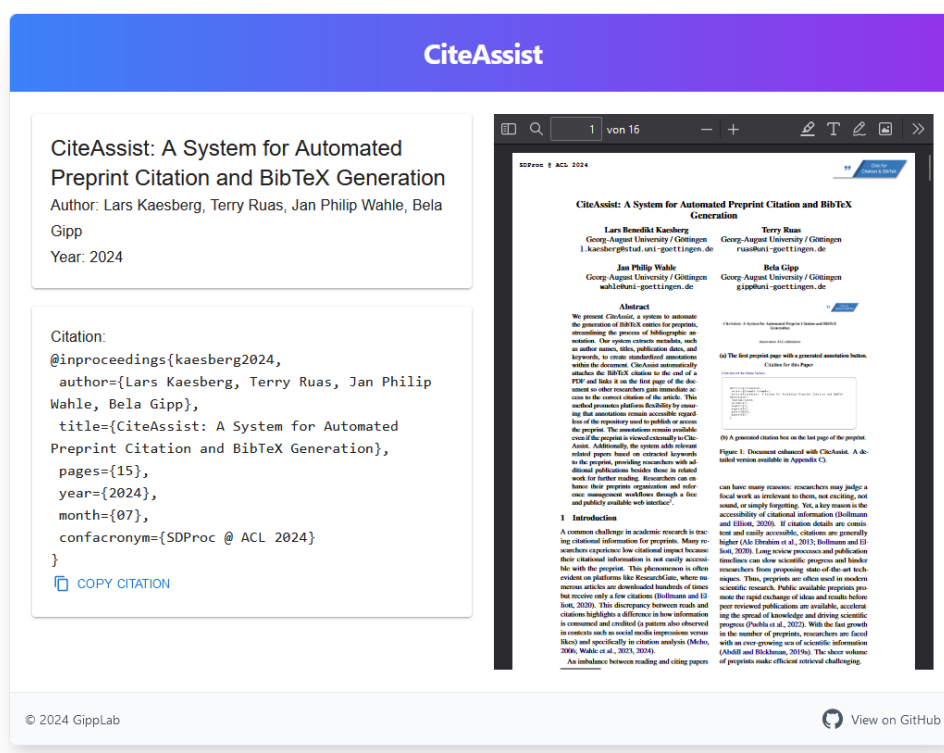

Figure 7: CiteAssist Generate Page.

Figure 8: CiteAssist Preprint Webview.

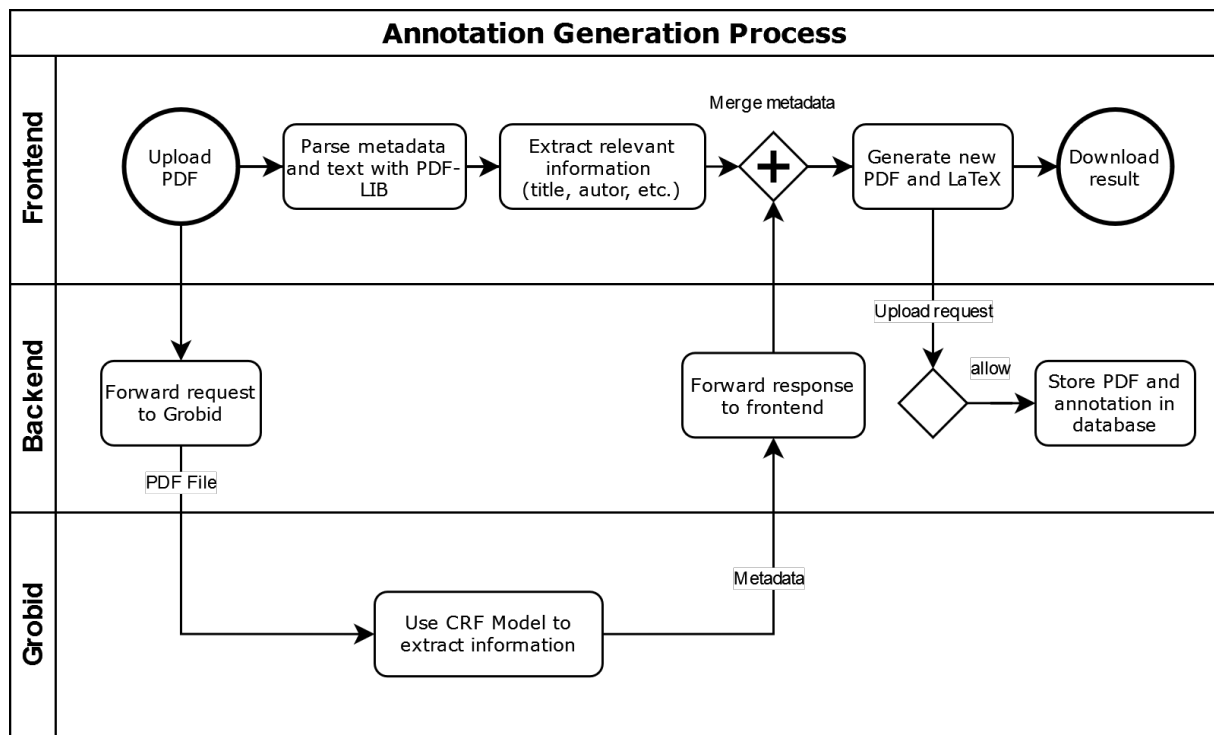## C CiteAssist Enhanced Document

Figure 9: Generation process for a CiteAssist annotation.
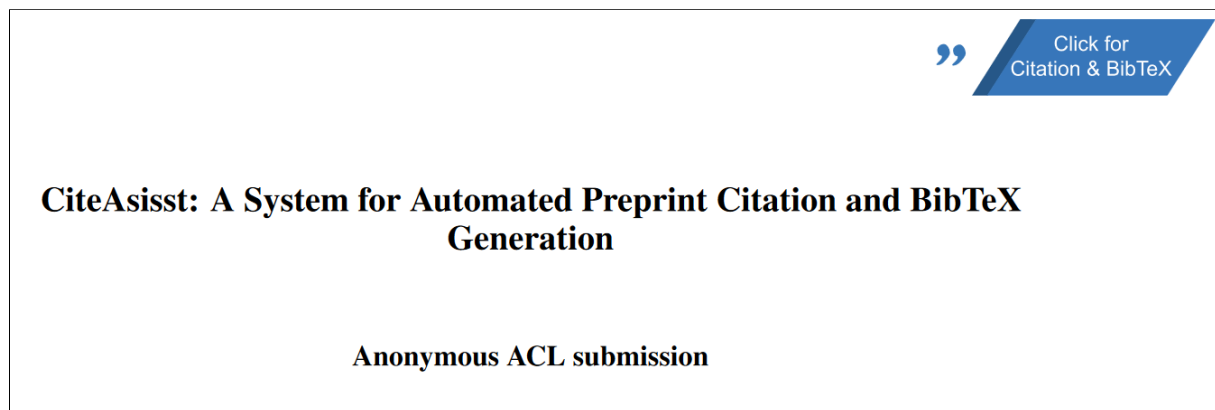
Figure 10: The first preprint page with a generated annotation button.

Citation for this Paper

Click here for the Online Version

```
@article{citeassist,
 author={Example Example},
 title={CiteAsisst: A System for Automated Preprint Citation and BibTeX
Generation},
 journal={ACS},
 volume={1},
 number={1},
 pages={13},
 year={2024},
 month={01}
}
```

Figure 11: A generated citation box on the last page of the preprint.

AI Usage Card for *Enhanced Preprint Generator*

CORRESPONDENCE(S) Lars Benedikt Kaesberg	**CONTACT(S)** l.kaesberg@stud.uni-goettingen.de	**AFFILIATION(S)** Georg-August University Göttingen
	PROJECT NAME CiteAssist	**KEY APPLICATION(S)** automate preprint generation
MODEL(S) chat-gpt	**DATE(S) USED** 2023-06-07	**VERSION(S)** 3.5

IDEATION	**GENERATING IDEAS, OUTLINES, AND WORKFLOWS** Not used	**IMPROVING EXISTING IDEAS** Not used
	FINDING GAPS OR COMPARE ASPECTS OF IDEAS Not used	
LITERATURE REVIEW chat-gpt	**FINDING LITERATURE** search for similar work/tools	**FINDING EXAMPLES FROM KNOWN LITERATURE** Not used
	ADDING ADDITIONAL LITERATURE FOR EXISTING STATEMENTS AND FACTS Not used	**COMPARING LITERATURE** Not used

METHODOLOGY	**PROPOSING NEW SOLUTIONS TO PROBLEMS** Not used	**FINDING ITERATIVE OPTIMIZATIONS** Not used
	COMPARING RELATED SOLUTIONS Not used	
EXPERIMENTS	**DESIGNING NEW EXPERIMENTS** Not used	**EDITING EXISTING EXPERIMENTS** Not used
	FINDING, COMPARING, AND AGGREGATING RESULTS Not used	

WRITING chat-gpt	**GENERATING NEW TEXT BASED ON INSTRUCTIONS** Not used	**ASSISTING IN IMPROVING OWN CONTENT** rewrite own text to improve writing style
	PARAPHRASING RELATED WORK Not used	**PUTTING OTHER WORKS IN PERSPECTIVE** Not used
PRESENTATION	**GENERATING NEW ARTIFACTS** Not used	**IMPROVING THE AESTHETICS OF ARTIFACTS** Not used
	FINDING RELATIONS BETWEEN OWN OR RELATED ARTIFACTS Not used	

CODING	GENERATING NEW CODE BASED ON DESCRIPTIONS OR EXISTING CODE Not used	REFACTORING AND OPTIMIZING EXISTING CODE Not used
	COMPARING ASPECTS OF EXISTING CODE Not used	
DATA	SUGGESTING NEW SOURCES FOR DATA COLLECTION Not used	CLEANING, NORMALIZING, OR STANDARDIZING DATA Not used
	FINDING RELATIONS BETWEEN DATA AND COLLECTION METHODS Not used	
ETHICS	WHAT ARE THE IMPLICATIONS OF USING AI FOR THIS PROJECT? undefined	WHAT STEPS ARE WE TAKING TO MITIGATE ERRORS OF AI FOR THIS PROJECT? undefined
	WHAT STEPS ARE WE TAKING TO MINIMIZE THE CHANCE OF HARM OR INAPPROPRIATE USE OF AI FOR THIS PROJECT? undefined	THE CORRESPONDING AUTHORS VERIFY AND AGREE WITH THE MODIFICATIONS OR GENERATIONS OF THEIR USED AI-GENERATED CONTENT Yes

AI Usage Card v1.0   https://ai-cards.org   PDF | BibTeX

CiteAssist
CITATION SHEET

Generated with citeassist.uni-goettingen.de

BibTeX Entry

```
@inproceedings{kaesberg2024,
  author={Kaesberg, Lars and Ruas, Terry and Wahle, Jan Philip and Gipp, Bela},
  title={CiteAssist: A System for Automated Preprint Citation and BibTeX Generation},
  address={Bangkok, Thailand},
  booktitle={Proceedings of the Fourth Workshop on Scholarly Document Processing (SDP
      2024)},
  pages={105--119},
  publisher={Association for Computational Linguistics},
  url={https://aclanthology.org/2024.sdp-1.10/},
  year={2024},
  month={08},
  topic={nlp}
}
```

Online Access

Official Publication   https://aclanthology.org/2024.sdp-1.10/

CiteAssist   https://citeassist.uni-goettingen.de/preprint/6c9fbf87-af1f-4b10-9428-8960172a2844