Towards Automatic Formal Feedback on Scientific Documents

Louise Bloch[1,2,3,*], **Johannes Rückert**[1,*], **Christoph M. Friedrich**[1,2]
[1]Department of Computer Science, University of Applied Sciences and Arts Dortmund,
Emil-Figge-Str. 42, 44227 Dortmund, Germany,
[2]Institute for Medical Informatics, Biometry and Epidemiology (IMIBE),
45147 Essen, Germany,
[3]Institute for Artificial Intelligence in Medicine (IKIM),
University Hospital Essen, 45147 Essen, Germany,
[*]These authors contributed equally to the work,
Correspondence: christoph.friedrich@fh-dortmund.de

Abstract

This paper introduces IPPOLIS Write, an open source, web-based tool designed to provide automated feedback on the formal aspects of scientific documents. Aimed at addressing the variability in writing and language skills among scientists and the challenges faced by supervisors in providing consistent feedback on student theses, IPPOLIS Write integrates several open source tools and custom implementations to analyze documents for a range of formal issues, including grammatical errors, consistent introduction of acronyms, comparison of literature entries with several databases, referential integrity of figures and tables, and consistent link access dates.

IPPOLIS Write generates reports with statistical summaries and annotated documents that highlight specific issues and suggest improvements while also providing additional background information where appropriate. To evaluate its effectiveness, a qualitative assessment is conducted using a small but diverse dataset of bachelor's and master's theses sourced from arXiv. Our findings demonstrate the tool's potential to enhance the quality of scientific documents by providing targeted and consistent feedback, thereby aiding both students and professionals in refining their document preparation skills.

1 Introduction

Feedback on scientific documents, for example within a peer-review process, usually and understandably focuses on the discipline-specific content first and foremost. Writing, language, and other formal aspects are a secondary focus and often only commented upon when glaring or repeating issues are present. And while every scientist is expected to have a good grasp of their discipline-specific content, experiences and skills in the areas of writing and language vary greatly among the scientific community, making such feedback less consistent and more subjective (Shashok, 2008; Wei and Liu, 2024).

For students writing their first scientific documents, such feedback on formal issues is especially useful. These students often make similar mistakes, and supervisors are faced with the task of repeatedly providing feedback about the same issues or focusing feedback on the more important areas, which are usually related to the discipline-specific content and not so much to writing, language, and other formal aspects. This is especially problematic in study courses where writing is not the main way to communicate results or solve tasks.

Existing tools for automated scientific document analysis either focus on analyzing the contents of the documents with regard to their accuracy and veracity, work only for certain document formats, or provide feedback or corrections for specific aspects only. In addition, most of the tools are commercial and closed source. Some of the existing tools are introduced in Section 2.

In this paper, we introduce a web-based open source software[1], which aims to combine a number of existing open source tools and libraries with custom implementations into a single application for analyzing scientific documents under formal aspects pertaining to document structure, readability, literature, referential integrity, tables, and figures. Based on a number of independent document analyzers, it generates reports with statistics and annotated documents with feedback.

2 Related Work

Various tools provide feedback on scientific manuscripts, each with a distinct focus or supported input formats. A detailed review on automated paper review systems (Lin et al., 2023) explained the underlying concepts, recent tools, and challenges.

[1]https://gitlab.com/ippolis_wp3/write, Accessed: 2025-06-05

Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications, pages 334–344

In (Lu and Liu, 2014), a tool was presented which validates the formal compliance of dissertations submitted as DOCX documents with given templates. The tool checks line spacing, font, font size, alignment style, and other formal aspects of DOCX documents. As a pre-processing step, the documents were converted to the eXtensible Markup Language (XML) format. An experiment on 50 dissertations compared automated annotations with manual ones, yielding a 94.5 % detection rate and a 3.7 % false detection rate.

The IEEE PDF eXpress®[2] was developed to validate the consistency of IEEE-related conference and journal submissions in the Portable Document Format (PDF) with respective guidelines. Among other aspects, the proprietary tool checks page margins and the copyright footer.

The ACL pubcheck tool[3] performs similar checks for ACL venues. It detects common formatting issues related to the ACL template in PDF documents. These issues include font inconsistencies, improper author formatting, margin violations, and outdated citations.

TeXtidote[4] is a tool that detects formal issues in LaTeX and Markdown. It checks the style (e.g., proper title formatting, reference capitalization, caption punctuation), citations and references (e.g., consistent citation commands, reference summaries), figures (e.g., presence of captions and references), document structure (e.g., singular subsections, valid section order, stacked headings, and short sections), and hard-coding (e.g., relative paths for figures, hard-coded section/figure/table references, manual line and page breaks). Spelling, grammar, and punctuation errors were detected using the LanguageTool (Naber, 2003), an open source proofreading software based on rule-based correction algorithms and Machine Learning (Breneis, 2018).

Penelope AI[5] is a proprietary Artificial Intelligence (AI)-based tool that checks whether DOCX manuscripts meet configurable journal requirements. The tool performs validations, including the availability, position, and title of the ethical approval statement, along with the necessary declarations. It checks the formatting and completeness of the title page and abstract, as well as the presence of pre-defined sections. Figures and tables are verified for correct integration, proper positioning, logical order, and accurate referencing. The manuscripts are evaluated for accurate referencing styles, proper citation order, and completeness of reference lists. Compliance with journal-specific limits on words, references, tables, and figures is checked. Endnote citations, metadata completeness, page numbers, and line spacing are annotated.

The proprietary YesNoError tool[6] focuses more on content-related errors than formal feedback. It was designed to process PDF, DOCX, and LaTeX files and validates the methodological process, statistical correctness, and interpretational comprehensibility using OpenAI's o1 model. The analyzers detect issues including mathematical (e.g., arithmetic operations, bracket mismatches), methodological (e.g., study design, sample sizes, statistical tests), literature (e.g., citation and reference consistency), and logical errors (e.g., consistency of statements, conclusions, and argument flow).

Unlike these tools, our tool is open source, supports various file formats, is configurable for different formatting requirements in different disciplines, and emphasizes a broad range of comprehensive formal feedback.

3 Methods and Materials

IPPOLIS Write is a web-based tool that analyzes documents provided by its users based on the configured analysis profile and generates feedback and statistics as annotations directly in the original document and as a report which can be viewed in the web interface. An overview of the general workflow is shown in Figure 1 and is described in more detail in the following sections.

3.1 Cloud Share Link

The documents to be analyzed, analysis progress information and analysis results are stored in a cloud share. Many popular cloud providers are supported through the Web-based Distributed Authoring and Versioning (WebDAV) (Whitehead and Goland, 1999) protocol. Users can utilize their own cloud provider by generating a cloud share link with full read and write permission, allowing the tool to not only read the documents to be analyzed but also save analysis progress information

[2] https://www.ieee.org/conferences/publishing/pdfexpress.html, Accessed: 2025-06-05

[3] https://github.com/acl-org/aclpubcheck, Accessed: 2025-06-05

[4] https://github.com/sylvainhalle/textidote, Accessed: 2025-06-05

[5] https://www.penelope.ai/, Accessed: 2025-06-05

[6] https://yesnoerror.com/whitepaper, Accessed: 2025-06-05

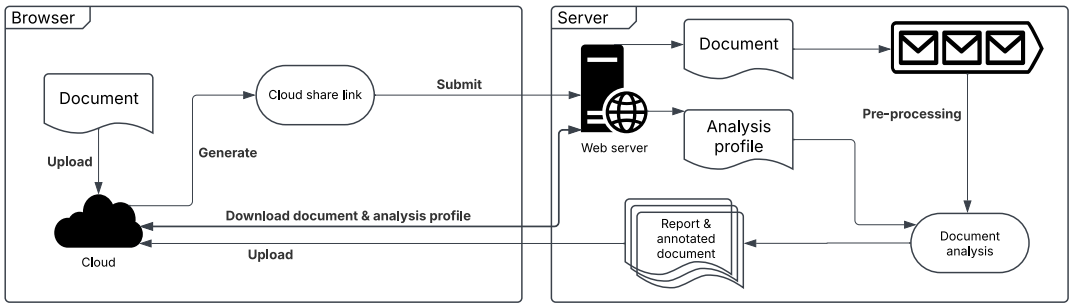

Figure 1: Overview of the document analysis pipeline. Users share a document with the tool through a cloud share link, the document is converted and analyzed on the server, and the annotated document and report are saved in the user's cloud share and can be viewed on the tool's website.

and analysis results which can later be displayed on the website. That way, all data is always in the hands of the users. For convenience, a direct upload of documents is planned.

3.2 Analysis Profile

The analysis profile allows configuring, as well as disabling and enabling analysis options. Its goal is to make the tool useful for different disciplines, study courses, formal requirements, and phases in the writing process.

3.3 Pre-processing

After a document in one of the supported formats (PDF, DOCX, LaTeX, BibTeX, RIS, or ZIP) has been submitted, different pre-processing steps are performed to prepare it for analysis. The following sections outline this process for different input formats.

3.3.1 PDF

Although metadata such as the structure can be included in PDFs using PDF tags, this feature is rarely used (Schmitt-Koopmann et al., 2022) making the documents inaccessible to computer systems. IPPOLIS Write combines Poppler[7] to extract the content, positions, and fonts of texts with document layout detection to determine the functional role of each element. Document layout detection based on You Only Look Once version 8 (YOLOv8) (Jocher et al., 2023) were trained on the DocLayNet (Pfitzmann et al., 2022) and ArxivFormula[8] datasets. GeneRation Of BIbliographic Data (GROBID) v0.8.1[9] is used to identify citations and references and to convert bibliographic information into the BibTeX format.

3.3.2 DOCX

Apache Poor Obfuscation Implementation (POI) v5.2.5[10] is used to extract the content, structure, and formatting from DOCX documents.

3.3.3 LaTeX/ZIP

TeXtidote is used for pre-processing LaTeX files, where the text and a mapping to the original LaTeX file is extracted from the document. ZIP archives are supported to allow users to upload LaTeX projects, which usually consist of at least a BibTeX file in addition to one or more LaTeX source files. The archives are extracted and both LaTeX and BibTeX files are analyzed separately.

3.3.4 BibTeX

For literature analysis, the tool focuses on the BibTeX format. The tool uses pybtex v0.24.0[11], and jbibtex v1.0.20[12] to extract literature information and identify invalid entries and fields. Bibtool v2.68+ds-1[13] is used to identify literature entries cited in a LaTeX document, and bibutils v7.2-1[14] converts Research Information System Format (RIS) files to BibTeX.

3.4 Document Analysis

The actual analysis is performed by dozens of independent analyzers based on the configuration in

[7] https://poppler.freedesktop.org/, Accessed: 2025-06-05

[8] https://github.com/microsoft/ArxivFormula, Accessed: 2025-06-05

[9] https://github.com/kermitt2/grobid, Accessed: 2025-06-05

[10] https://poi.apache.org/, Accessed: 2025-06-05

[11] https://pybtex.org/, Accessed: 2025-06-05

[12] https://github.com/jbibtex/jbibtex, Accessed: 2025-06-05

[13] https://github.com/ge-ne/bibtool, Accessed: 2025-06-05

[14] https://ctan.org/pkg/bibutils, Accessed: 2025-06-05

the analysis profile and the artifacts produced during pre-processing. The analyzers are summarized in nine categories. These are general/formal, document structure, language, readability, literature, reproducibility, referential integrity, images, and tables. The analyzers refer to different input formats and provide analysis results, such as report statistics and annotations. More information on analyzers, their implementations, advantages and disadvantages is described in Section 4. A complete list of implemented and planned analyzers can be found in the Git repository[15]. All analyzers can be enabled or disabled for individual requirements via the analysis profile. During analysis, users are informed about the current analysis progress in the web interface. This is entirely asynchronous, and the analysis is not canceled if the user closes the website.

3.5 Report and Annotated Document

The results, statistics and annotations produced by the analyzers are converted into a report and, for some document types, an annotated document which is saved in the user-provided cloud share. For DOCX documents, annotations are embedded as comments, implemented using Apache POI. PDF documents were annotated using iText7 v8.0.3[16]. Optional Content Group (OCG) layers are utilized to conveniently enable or disable annotations directly within the PDF document. The report contains statistics and gamification elements, such as comparisons to analysis results of earlier versions of the document.

3.6 Dataset

To present and evaluate the analyzers, experiments were performed on a dataset containing bachelor and master theses in PDF format. The dataset was extracted from the arXiv (Ginsparg, 1994, 2011) preprint server, which was searched for the terms "master thesis" and "bachelor thesis" in the field of computer science (arXiv category: CS.*) on 2025-03-11. This query leads to 492 master theses and 91 bachelor theses. One master thesis cannot be downloaded, and one bachelor thesis is an invalid PDF document. Documents with fewer than 20 pages were excluded, as theses are typically longer, leading to 451 master theses and 79 bachelor theses. It was decided to concentrate on more recent theses, which were submitted in arXiv after the 2023-01-01, leading to 110 master's and 39 bachelor's theses. One bachelor thesis and four master theses not written in English or German, the languages currently supported by the tool, were excluded. The four master theses were written in Persian, French, partially in Greek, and partially in Japanese. The bachelor thesis was written in Indonesian. The resulting 106 master's and 38 bachelor's theses are randomly split into a 50 % training, and a 50 % test set. The most frequently arXiv primary category was cs.LG (Machine Learning; 14.58 %) followed by cs.CL (Computation and Language; 11.81 %). An overview of the dataset is given in Table 1 and a list containing the arXiv IDs as well as a script to download the PDF documents is published in a git repository[17]. The dataset includes documents created in LaTeX and Word, featuring a wide variety of templates.

Dataset	# documents	Avg. # pages
Bachelor train	19	66.63
Bachelor test	19	61.84
Master train	53	73.64
Master test	53	76.83
$\sum$	144	72.33

Table 1: Distribution of the primary arXiv categories across the training and test set.

4 Results

The tool was evaluated on the test set. None of the theses were used to develop or optimize the tool prior to evaluation. All documents in the test set were processed without problems during the analysis. In the following sections, an overview of the currently implemented PDF analyzers is given, along with examples from the test set, showing which annotations are generated most frequently, in which areas the tool works well, and for which aspects incorrect annotations are generated most often. In addition, common reasons for formatting violations are explained.

Figure 2 presents boxplots depicting the average number of annotations per page for each annotation category in the test set. The overall number of annotations per page differs between 2.17 and 16.08. Master's theses exhibit a slightly lower number of

[15] https://gitlab.com/ippolis_wp3/write, Accessed: 2025-06-05

[16] www.itextpdf.com, Accessed: 2025-06-05

[17] https://gitlab.com/ippolis_wp3/bea-2025-ippolis-write-dataset. Accessed: 2025-06-05

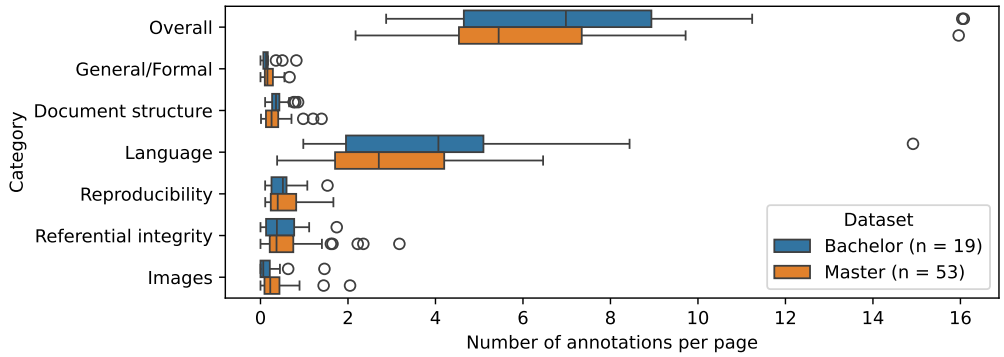

Figure 2: Boxplot showing the average number of annotations per page across annotation categories in the test set.

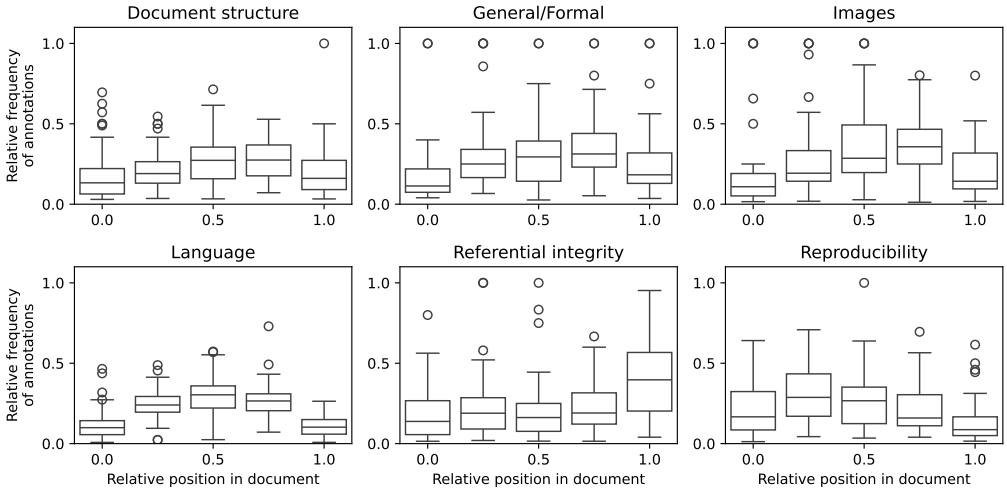

Figure 3: Boxplots showing annotation distribution by category across page positions in the test set.

annotations per page. This indicates that master students have more experience writing scientific theses. The most frequent annotations were language annotations that differed between 0.38 and 14.92 annotations per page. The remaining categories are detected less frequently with a maximum of 3.18 annotations per page, which was reached for a master thesis and the referential integrity category.

Figure 3 illustrates the distribution of annotation frequencies across document pages for each category. The plots show raw annotations that have not been validated. In the document structure, images, and language categories, annotations are more frequent in the middle sections of the documents. One possible explanation is that the middle of the document contains the main body of text along with most of the figures. General/formal and reproducibility annotations exhibit similar patterns, although general/formal annotations tend to accumulate slightly more in the later sections, while reproducibility annotations are more frequent in the earlier sections. Referential integrity annotations accumulate at the end of the documents, indicating a large number of annotations in the reference sections, and thus references that were not recognized in the text.

A collage of several screenshots showing individual annotations for the master's thesis (Singh, 2024) is visualized in Figure 4.

4.1 General/Formal

The analyzers that provide general and formal feedback ensure a well-organized presentation, including the identification of changing fonts, line spacing, and text alignments, as well as texts that exceed page margins. In addition, incorrect decimal and thousands separators are recognized, as well as missing punctuation marks at the end of captions.

The test set did not reveal any issues related to font changes. One reason for this might be that the documents are submitted theses, which are often

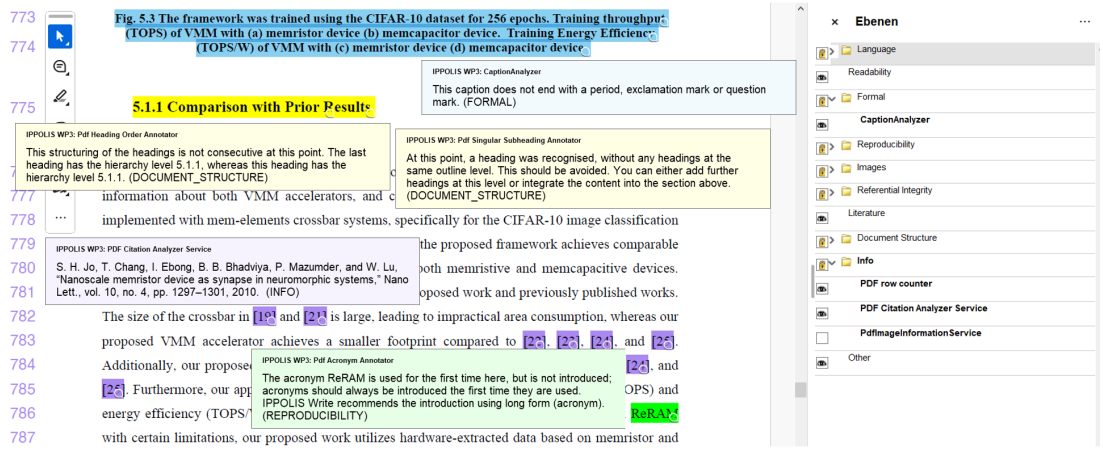

Figure 4: Collage of several screenshots showing individual annotations for the master's thesis (Singh, 2024) from the test set. The OCG layers are shown on the right-hand side, which can be used to deactivate individual annotators.

revised multiple times. Occasional page margin violations are reliably identified, especially in LaTeX-generated PDFs. These violations often result from incorrect hyphenation settings, improperly formatted equations, or misaligned tables. When such issues are detected, IPPOLIS Write offers feedback, outlines common causes, and suggests ways to prevent them. At the moment, it does not detect figures that exceed page margins, which also occurs in LaTeX-based PDF documents but a future extension is planned for this feature.

In addition, many documents had missing punctuation marks in the captions. For this analyzer, it was observed that the layout detection described in Section 3.3.1 occasionally fails to detect captions correctly, depending on the template used.

4.2 Document Structure

All analyzers in this category are designed to maintain the structural integrity of the manuscript. These include analyses on heading structure (e.g., inconsistent heading numbers and orders, lowercase headings, missing heading layer), page numbers (e.g., missing and inconsistent page numbers), section content (e.g., empty sections), as well as table and figure captions (e.g., availability, length, and consistent positions).

Problems most often and reliably detected in the dataset are empty sections (i.e., a section heading is immediately followed by a subsection heading), and sections without another section at the same level. Due to the difficulty of extracting the document layout from the PDF, captions are sometimes not identified correctly. For unusual formatting (e.g., small caps), section headings are sometimes not identified correctly. Tables and equations included as images are currently identified and analyzed as images.

4.3 Language

The language-based analyses include the investigation of the spelling, grammar, and punctuation (implemented using the LanguageTool[18]) as well as the vocabulary (occurring nouns, verbs, and n-grams). In addition, the detection of filler words and judgmental words (both adapted from the Readability Analysis Tool (Holdorf, 2016) and the angry-reviewer tool[19]), hype terms (Millar et al., 2023), and ChatGPT phrases[20] are implemented. First-person pronouns are detected as they may diminish the objectivity and neutrality of texts, particularly in specific languages and research fields. Duplicated sentences are annotated to prevent redundancies that could reduce the reader's attention.

Language problems most often detected in the analyzed theses are grammar errors, frequent use of first-person pronouns, and the use of judgmental words. Most incorrect language annotations are generated because of a lack of context awareness, e.g., inside mathematical expressions, or when a word has several meanings (e.g. "clearly").

4.4 Readability

The readability analysis focuses on identifying long sentence and reporting several readability scores

[18] https://languagetool.org/, Accessed: 2025-06-05

[19] https://github.com/anufrievroman/Angry-Reviewer, Accessed: 2025-06-05

[20] https://www.twixify.com/post/most-overused-words-by-chatgpt, Accessed: 2025-06-05

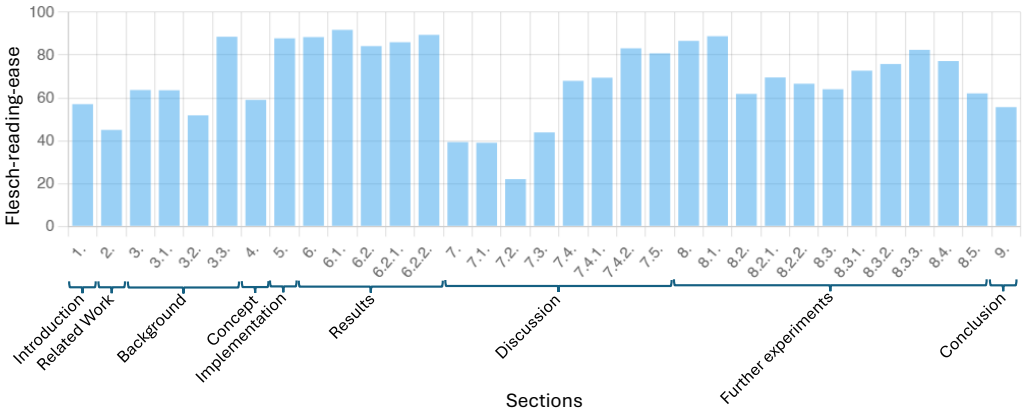

Figure 5: Flesch-reading-ease for document (Lachner, 2021) from the test set.

(e.g., the Coleman-Liau index (Coleman and Liau, 1975), or the Flesch-reading-ease (Flesch, 1948)). The readability scores are implemented to identify changes in language quality across the document. It was anticipated that more theoretical sections, mostly at the beginning of a thesis such as the introduction, related work, or methods sections, would have higher complexity compared to sections focusing on applied content (e.g., implementations or evaluations). A similar pattern was found for the Flesch-reading-ease across one test bachelor thesis (Lachner, 2021) which is visualized in Figure 5. The Flesch-reading-ease considers the average length of sentences and the average number of syllables per word. Higher values indicate easier text comprehension. The example exhibits lower values and thus more difficult text comprehension in the introduction, related work, background, and concept sections, followed by higher values in the implementation and results sections. The discussion section shows mixed scores, while the further experiments and conclusion sections display moderate values. Additional readability scores can be added in future versions of the tool.

4.5 Literature

Literature annotations are generated for citations with incomplete information, incorrect citation types, and published arXiv preprints. Information is retrieved from SemanticScholar (Ammar et al., 2018; Kinney et al., 2023; Lo et al., 2020), DBLP (Ley, 2002), OpenAlex (Priem et al., 2022), and CrossRef (Rachael, 2014).

The extraction of citations and reference lists from PDF documents using GROBID and additional extraction using regular expressions shows acceptable results. However, some citations remain incorrectly identified, independent of the citation style used. In addition, some false-positive detections occur within equations or references to figures or captions. One reason for the incorrect detection is that GROBID was mainly trained on scientific articles, and the formatting differs from theses. The detection accuracy of published arXiv preprints was investigated in more detail in previous work (Bloch et al., 2023).

4.6 Reproducibility

Reproducibility-based analyses include verifying the availability and consistency of hyperlink access dates, detecting URLs that are not properly linked, and identifying acronyms that are used without being introduced.

During the evaluation, it was observed that many documents lack access dates when mentioning URLs. These are often detected reliably, with some exceptions for date formats that omit the day. Missing hyperlinks, i.e. links that are not clickable, are reliably detected.

Almost all documents in the test set contain acronyms that are not introduced properly when first mentioned or employ inconsistent patterns to introduce them (e.g., Long Form (LF) vs. LF (Long Form)). Currently, the tool sometimes mistakenly identifies certain names, such as those mentioned in references (e.g., LeCun), as well as parts of equations (e.g., $A - B$), as acronyms, resulting in incorrect annotations. These issues can be addressed by improving reference detection and recognizing mathematical equations more accurately. The tool currently does not check the availability of acronym indexes, which could serve as an alternative to introducing acronyms upon first use.

4.7 Referential Integrity

The referential integrity of a document is important for the clarity and comprehensibility of scientific texts. IPPOLIS Write ensures updated table of contents, as well as complete and consistent references for tables, figures, sections and references.

In the test set, some PDF documents produced with DOCX and LaTeX suffer from outdated tables of content. For LaTeX documents, the page number of the bibliography section is often incorrect. DOCX documents are affected by inconsistent page numbers and section numbers, often due to manually created or outdated tables of contents. The tool identifies such problems reliably, with some issues when identifying page numbers.

Few figures and tables in the test set have missing captions, which were reliably identified. In contrast, many figures or tables are never or distantly mentioned in the text. In addition, inconsistent reference types were sometimes identified. Similarly to previous observations, occasionally incorrect annotations were produced, resulting mostly from incorrect layout detection, line breaks, or page breaks.

Citations are cross-checked with the literature index based on the previously mentioned GROBID software. The identification and mapping of citations show some issues especially for multiple consecutive references, as well as references in tables. One reason is that the formatting of theses differs from the documents, which were used to train and optimize GROBID.

4.8 Images

The tool examines whether the resolution of figures is sufficient for print publications and ensures they are not distorted during document creation. Furthermore, experiments were conducted to validate the images in more detail. These analyses include the detection of missing axis labels, tick marks, the number of colors, and image artifacts and were implemented using Vision-Language Models (VLMs) (Rückert et al., 2025). These features will be integrated into the tool in the future.

One issue present in almost every analyzed thesis is the inclusion of images with low quality (less than 300 DPI), with some of these including images of clearly poor quality with less than 100 DPI. During the implementation and test of the image analysis features, it became apparent that this is not always due to the quality of the original image, but often unintentional image compression during document compilation or conversion can lead to poor image quality in the final PDF.

4.9 Tables

At the moment, tables are only checked for valid decimal and thousands separators by the tool. Analyzers regarding the validation of the table structure and the availability of units in table columns are planned. Table captions are analyzed as described in Section 4.2.

5 Discussion

The IPPOLIS Write tool is a web-based open source tool that provides automated feedback on formal aspects of scientific theses and papers to help students, but also researchers fulfill formal aspects of research theses and scientific papers. In comparison to previously developed tools, it is able to process a large number of document formats (PDF, DOCX, LaTeX, BibTeX, RIS, ZIP) and imposes no restriction on used templates. In addition, feedback is provided on a wide variety of formal aspects which are consolidated into nine categories (formal/general, document structure, language, readability, literature, reproducibility, referential integrity, images, and tables). Data privacy is maintained by temporarily sharing documents with the software through a cloud share, while all analysis results are directly stored in the cloud share. This solution enables the implementation of a gamification element, which can motivate users without having to create a user account. The analysis pipeline includes pre-processing of the documents to convert them into machine-readable data and document analyzers, which can be manually configured via the analysis profile. This profile makes it possible to customize the tool to individual requirements, for example, in different departments. The analysis results are converted into a report, and, for some document types, an annotated document was generated. The annotated document makes it easier for the user to quickly understand the annotations and correct the document.

The tool was validated using a PDF dataset from the arXiv preprint server containing bachelor's and master's theses with diverse formatting. The qualitative evaluation investigates which formal issues are identified most frequently, in which areas the tool works well, and for which aspects incorrect annotations are generated. The tool successfully

processes all theses and identified various issues in the test set. These include violations of the page margins, especially in LaTeX-based PDFs, empty sections, and sections without sibling sections at the same hierarchical level. Furthermore, various grammatical errors, missing access dates for URLs, and acronyms with missing or inconsistently introduced long forms were correctly identified. Some documents had outdated tables of content and figures or tables that were never mentioned in the text. Many documents suffered from low image quality, in part with resolutions smaller than 100 DPI. The high frequency of annotations found for theses published on arXiv illustrates the value of the tool.

The most incorrect annotations were identified as being caused by inaccurate layout detection, among other things leading to the detection of non-introduced acronyms or grammatical errors in mathematical equations or citations. This problem will be addressed in future releases by using improved layout analysis. As document layout analysis remains an active area of research (Gemelli et al., 2024), with demonstrated potential to enhance the robustness of Large Language Models (LLMs) for document understanding (Sciuss-Bertrand et al., 2024; Lamott et al., 2024), further advancements in this domain are anticipated in the near future. Additional problems, which will be addressed in future developments, will be improved context awareness during the detection of judgmental words, as well as improved detection accuracy for citations and references using LLMs.

Overall, the results demonstrate that IPPOLIS Write can identify various issues in academic theses, offering valuable formal feedback to assist students and researchers in revising their theses and papers.

6 Conclusion

In summary, this paper introduces IPPOLIS Write, a web-based tool that automatically provides feedback on the formal aspects of scientific theses and papers, assisting both students and researchers in meeting these requirements. IPPOLIS Write covers the most common document formats and a wide variety of formal aspects. In addition, it can be customized to meet the requirements of various disciplines. Data privacy is maintained by temporarily sharing documents through a cloud share. The tool was qualitatively evaluated on a diverse test set containing bachelor and master theses. The validation shows that IPPOLIS Write detects a wide variety of issues in these documents. Incorrect annotations are mostly caused by inaccurate document layout detection, which is related to the high number of templates used in the dataset.

Limitations

IPPOLIS Write is not a finished product and in some areas lacks in robustness and consistency. Most issues stem from the difficulty of extracting information systematically from documents. Handling subtle differences introduced by diverse document creation tools, layouts, fonts, and formatting has consumed more development time than anticipated, and new documents still sometimes reveal new problems. Better pipelines based on new technologies may alleviate this issue in the future. Another limitation is the lack of a systematic quantitative evaluation of the generated annotations including a manual analysis of false positives and false negatives, as well as a user study, all of which would be useful to provide stronger evidence of the tool's impact. The evaluation dataset is currently limited to student theses from the computer science field. A more varied dataset could help expand and generalize the evaluation results. The tool is meant as a learning resource and does not provide immediate corrections but only suggestions which have to be manually applied, this limitation is an intentional design decision. Limitations that could be addressed in future development iterations of the tool include annotations available directly in the browser (currently only the Adobe PDF viewer fully supports the annotations), marking suggestions as solved/irrelevant for future document analyses, Optical Character Recognition (OCR) pipelines for images, better extraction of bibliography information from PDF, and support for additional languages beyond English and German.

Acknowledgments

This work is part of the BMBF-funded project "Intelligente Unterstützung projekt- und problemorientierter Lehre und Integration in Studienabläufe" (IPPOLIS) (Funding code: 16DHBKI050).

The work of Louise Bloch was partially funded by a PhD grant from University of Applied Sciences and Arts Dortmund, Dortmund, Germany.

References

Waleed Ammar, Dirk Groeneveld, Chandra Bhagavatula, Iz Beltagy, Miles Crawford, Doug Downey, Jason Dunkelberger, Ahmed Elgohary, Sergey Feldman, Vu Ha, Rodney Kinney, Sebastian Kohlmeier, Kyle Lo, Tyler Murray, Hsu-Han Ooi, Matthew Peters, Joanna Power, Sam Skjonsberg, Lucy Lu Wang, Chris Wilhelm, Zheng Yuan, Madeleine van Zuylen, and Oren Etzioni. 2018. Construction of the literature graph in semantic scholar. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2018): Human Language Technologies*, volume 3, pages 84 – 91, New Orleans - Louisiana. Association for Computational Linguistics. Industry Papers.

Louise Bloch, Johannes Rückert, and Christoph M. Friedrich. 2023. PreprintResolver: Improving citation quality by resolving published versions of ArXiv preprints using literature databases. In *Linking Theory and Practice of Digital Libraries*, pages 47 – 61, Cham. Springer Nature Switzerland.

Markus Brenneis. 2018. Development of neural network based rules for confusion set disambiguation in LanguageTool. In *SKILL 2018 - Studierendenkonferenz Informatik*, pages 181 – 192, Bonn. Gesellschaft für Informatik e.V.

Meri Coleman and Ta Lin Liau. 1975. A computer readability formula designed for machine scoring. *Journal of Applied Psychology*, 60:283 – 284.

Rudolph Flesch. 1948. A new readability yardstick. *Journal of Applied Psychology*, 32(3):221 – 233.

Andrea Gemelli, Simone Marinai, Lorenzo Pisaneschi, and Francesco Santoni. 2024. Datasets and annotations for layout analysis of scientific articles. *International Journal on Document Analysis and Recognition (IJDAR)*, 27(4):683 – 705.

Paul Ginsparg. 1994. First steps towards electronic research communication. *Computers in Physics*, 8(4):390 – 396.

Paul Ginsparg. 2011. ArXiv at 20. *Nature*, 476(7359):145 – 147.

Matthias Holdorf. 2016. Computer support for the analysis and improvement of the readability of IT-related texts. Master's thesis, Master Thesis at Department of Informatics - Technische Universität München (TUM), Munich, Germany.

Glenn Jocher, Jing Qiu, and Ayush Chaurasia. 2023. Ultralytics YOLO. `https://github.com/ultralytics/ultralytics`, Accessed: 2025-06-05.

Rodney Michael Kinney, Chloe Anastasiades, Russell Authur, Iz Beltagy, Jonathan Bragg, Alexandra Buraczynski, Isabel Cachola, Stefan Candra, Yoganand Chandrasekhar, Arman Cohan, Miles Crawford, Doug Downey, Jason Dunkelberger, Oren Etzioni, Robert Evans, Sergey Feldman, Joseph Gorney, David W. Graham, F.Q. Hu, Regan Huff, Daniel King, Sebastian Kohlmeier, Bailey Kuehl, Michael Langan, Daniel Lin, Haokun Liu, Kyle Lo, Jaron Lochner, Kelsey MacMillan, Tyler Murray, Christopher Newell, Smita Rao, Shaurya Rohatgi, Paul L Sayre, Zejiang Shen, Amanpreet Singh, Luca Soldaini, Shivashankar Subramanian, A. Tanaka, Alex D Wade, Linda M. Wagner, Lucy Lu Wang, Christopher Wilhelm, Caroline Wu, Jiangjiang Yang, Angele Zamarron, Madeleine van Zuylen, and Daniel S. Weld. 2023. The semantic scholar open data platform. Preprint. *Preprint*, arXiv:2301.10140v1.

Michael Lachner. 2021. Linear and non-linear machine learning attacks on physical unclonable functions. Master's thesis, Master Thesis at Institute for Computer Science - Ludwig Maximilian University Munich (LMU), Munich, Germany.

Marcel Lamott, Yves-Noel Weweler, Adrian Ulges, Faisal Shafait, Dirk Krechel, and Darko Obradovic. 2024. Lapdoc: Layout-aware prompting for documents. In *Document Analysis and Recognition - ICDAR 2024*, pages 142 – 159, Cham. Springer Nature Switzerland.

Michael Ley. 2002. The DBLP computer science bibliography: Evolution, research issues, perspectives. In *String Processing and Information Retrieval*, pages 1 – 10, Berlin, Heidelberg. Springer Berlin Heidelberg.

Jialiang Lin, Jiaxin Song, Zhangping Zhou, Yidong Chen, and Xiaodong Shi. 2023. Automated scholarly paper review: Concepts, technologies, and challenges. *Information Fusion*, 98:101830.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld. 2020. S2ORC: The semantic scholar open research corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*, pages 4969 – 4983.

Xin Lu and Jing Liu. 2014. An XML-based model method for review of academic dissertation format. In *Proceeding os the seventh International Symposium on Computational Intelligence and Design (ISCID 2014)*, volume 2, pages 174 – 178.

Neil Millar, Bojan Batalo, and Brian Budgell. 2023. Promotional language (hype) in abstracts of publications of national institutes of health–funded research, 1985-2020. *JAMA Network Open*, 6(12):e2348706 – e2348706.

Daniel Naber. 2003. A rule-based style and grammar checker. Master's thesis, Diploma thesis at Technical Faculty, University of Bielefeld, Bielefeld, Germany.

Birgit Pfitzmann, Christoph Auer, Michele Dolfi, Ahmed S. Nassar, and Peter Staar. 2022. DocLayNet: A large human-annotated dataset for document-layout segmentation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2022)*, pages 3743 – 3751.

Jason Priem, Heather A. Piwowar, and Richard Orr. 2022. OpenAlex: A fully-open index of scholarly works, authors, venues, institutions, and concepts. Preprint. *Preprint*, arXiv:2205.01833v2.

Lammey Rachael. 2014. CrossRef developments and initiatives: an update on services for the scholarly publishing community from CrossRef. *Science Editing*, 1(1):13 – 18.

Johannes Rückert, Louise Bloch, and Christoph M. Friedrich. 2025. Evaluating compliance with visualization guidelines in diagrams for scientific publications using large vision language models. In *Accepted at the International Conference on Document Analysis and Recognition (ICDAR 2025)*.

Felix M. Schmitt-Koopmann, Elaine M. Huang, and Alireza Darvishy. 2022. Accessible PDFs: Applying artificial intelligence for automated remediation of STEM PDFs. In *Proceedings of the 24th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS 2022)*, ASSETS '22, New York, NY, USA. Association for Computing Machinery.

Anna Scius-Bertrand, Atefeh Fakhari, Lars Vögtlin, Daniel Ribeiro Cabral, and Andreas Fischer. 2024. Are layout analysis and ocr still useful for document information extraction using foundation models? In *Document Analysis and Recognition - ICDAR 2024*, pages 175 – 191, Cham. Springer Nature Switzerland.

Karen Shashok. 2008. Content and communication: How can peer review provide helpful feedback about the writing? *BMC Medical Research Methodology*, 8(1):3.

Ankur Singh. 2024. Mem-elements based neuromorphic hardware for neural network application. *Preprint*, arXiv:2403.03002v1.

Yuzhu Wei and Donghong Liu. 2024. Incorporating peer feedback in academic writing: a systematic review of benefits and challenges. *Frontiers in Psychology*, 15:1506725.

E. James Whitehead and Yaron Y. Goland. 1999. WebDAV. In *Proceedings of the Sixth European Conference on Computer Supported Cooperative Work (ECSCW '99) 12–16 September 1999, Copenhagen, Denmark*, pages 291 – 310, Dordrecht. Springer Netherlands.