Expert Systems With Applications 333 (2027) 133839

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

The DocLap integrated system for understanding multi-page scholarly documents

Lorenzo Massai *, Simone Marinai

DINFO - University of Florence, via S. Marta, 3, Firenze, Italy

ARTICLE INFO

Keywords:
Document layout analysis
Large language models
Natural language processing
Ontologies
Question answering
Retrieval augmented generation
Visual document understanding

ABSTRACT

Computational understanding of documents is focused on visual and text analysis, building upon computer vision and natural language processing. With the advent of transformers document understanding is even more shifted from the actual comprehension of documents, which relies on concurrent perception of text, layout elements and document structure, to convoluted feature representations. Recent trends for providing unified access to such representations go towards Large Language Models (LLMs). However, these models have limitations: they lack explainability, demand significant resources for training and inference, and are not well suited for processing extensive inputs nor for direct application in specialized domains.

This paper aims at creating a comprehensive interface for document analysis, enabling multi-layered exploration and integrating diverse features and contextual information. By bridging diverse information, our work pursues the identification, characterization, and linking of visual elements to semantic and contextual data, leveraging LLMs for interoperability. This enables a unified access to textual, visual, and structural layers, embedding levels of structured knowledge directly in the LLM context. Recent advances in Retrieval Augmented Generation (RAG) are also exploited to address some LLM limitations related to context length, allowing access to latent information from document representations such as graph and vector embeddings. The association of structural information to visual data allows formal analysis of documents and is exploited in our model to enhance visual recognition, improved through multi-modal LLM correction supported by ontology-based constraint violation detection. The framework enables semantic retrieval over extracted information, providing direct access to the document structure which can be exploited in many applications such as Question Answering (QA) and document understanding.

As a result of this work, the DocLap (Document Layout Parser) system for document analysis and retrieval is proposed, which enables the extraction of visual and semantic features from documents and makes them accessible through natural language in an integrated framework providing conversational reasoning. The system's segmentation, error detection, and information retrieval capabilities are extensively evaluated through experiments.

1. Introduction

In recent years, document analysis has attracted significant attention, driven by the rapid growth of online media publishing and by commercial interest in analyzing documents of specialized sectors such as medical, legal, and financial. Narrowing the field to scientific literature, readers are able to understand the document meaning exploiting different kinds of contextual information like layout information and geometric properties of the elements which come along with text. The production of scientific literature is typically shared with unstructured media like PDFs and accessing different types of knowledge requires users to make several separated queries, linking data from different sources and keeping the original context at the same time.

This paper aims at extending the research field of Visual Document Understanding (VDU) in the scientific literature domain through the association of text semantics to visual features, merging them in a shared structure which allows multi-modal document exploration. The main challenges in this work can be found in the following three areas.

Semantic segmentation. The association of semantics to visual data is a prominent research problem in computer vision, including tasks like object recognition, image captioning, and image segmentation. In document analysis, the goal is to understand content by extracting the geometric properties of visual elements, such as *figures*, *tables*, and *text*, as well as layout components like *columns*, *footnotes*, and *titles*, and assigning them to semantic categories. Many document understanding

* Corresponding author.
E-mail address: lorenzo.massai@unifi.it (L. Massai).

https://doi.org/10.1016/j.eswa.2026.133839
Received 19 October 2025; Received in revised form 4 July 2026; Accepted 29 July 2026
Available online 31 July 2026

systems are limited to text blocks and figure/text classification, lacking domain specific recognition (e.g. *keywords*, *formulas*, and *chemical structures*). The scientific literature, with its variety of visual and textual content, is particularly suitable for semantic segmentation and offers an excellent environment for exploiting associations between representations. However, these representations are limited by their lack of interoperability and relying on images restricts document analysis to a single page. When considering the whole document, the problem becomes much more complex since inter-page relations and semantic regions spanning through multiple pages must be considered. Building on recent document segmentation architectures, our framework provides multi-page annotations and, by combining different layers of information (visual information and semantic relations), is capable of improving their performance. This also plays a significant role in reducing the impact of automatic annotation errors and labeling errors in large-scale datasets, which are commonly used to finetune models for downstream tasks.

Semantic integration. The integration of different document attributes can be pursued by merging extracted information into shared structures. This is useful to retrieve information about visual and layout elements in the document, or to get context information about the publication such as the author or the DOI. The presence of a formal structure helps maintain and extend context. Relations can also be exploited to identify structural constraint violations, such as overlapping layout elements, and to allow category-based searches. To achieve such awareness multiple layers of the same data have to be considered and an exhaustive semantic characterization of the entities is necessary. By combining visual information with ontology-based relations, we propose a framework that is able to detect and correct recognition issues such as overlap errors and inconsistencies between layout classes. This is achieved by introducing a post-processing strategy based on formal semantics that improves segmentation quality across different engines.

Information access. Recent trends for navigating different layers of information go towards intelligent agents capable of understanding both the subject of the query and its broader context, as well as the background of the user. Such agents are able to understand questions and to provide coherent answers spanning through different layers of information, adapting solutions and recommendations as the conversation evolves and learning what to say also from the dialogue. The reliance of visual question answering systems on document images results in limited awareness of the overall document and in the absence of contextual or domain-specific information. However, in real scenarios documents are mostly composed of multiple pages that should be processed altogether.

One of the goals of this work is to link different layers of information to visual media across the whole document and make them interoperable through conversational agents. We therefore propose a system that allows an interactive and conversational interface with documents from which different information layers (document annotations, structure and content) can be accessed.

This paper presents original contributions that advance the fields of analysis and interaction with scholarly documents by integrating different document representations and making them easily accessible to the user.

In particular, our main achievements can be identified in:

- building a comprehensive interface to allow multi-page navigation and a conversational interaction with scholarly documents;
- performing explainable association of layout information to visual and text data;
- enhancing detection of visual recognition anomalies exploiting semantic constraints and LLMs for their correction;
- allowing multi-layer interoperability through large language models.

To this end, we propose a framework that improves document segmentation and corrects recognition anomalies using visual, textual, and semantic information, enhanced by semantic constraints analysis and LLMs. Our framework is provided with a comprehensive interface for multi-page interaction, both navigational and conversational. It performs associations between layout and content and supports multi-layer interoperability to enhance access to scholarly documents.

Although the results of our research are primarily realized and evaluated within the context of existing pipelines, we aim to demonstrate their generality beyond the specific technologies used in our prototype. To this end, we quantitatively evaluate these improvements across multiple systems. This supports the view that our findings can be generalized across different pipelines and instantiated within a broad range of architectures. Consequently, the proposed system should not be interpreted as a novel end-to-end pipeline for segmentation and question answering, but rather as a reference implementation illustrating how targeted improvements can enhance existing pipelines performing the same tasks, independently of the specific components employed.

In particular, the main contributions of this work are to show how segmentation outputs can be improved through the application of explicit formal constraints, and how question answering performance can be improved by optimizing the context provided to an orchestrating LLM through automatic user query classification. The segmentation optimization mechanism detects inconsistencies arising from unintended spatial overlaps between semantically disjoint elements (e.g., a table over its caption). In such cases, the system detects the violation and enforces its correction through LLMs, that are also used for query classification.

2. Related work

The review of the state of the art focuses on three distinct and related research areas: computer vision, knowledge representation, and Natural Language Processing (NLP). Comprehensive processing and interaction with scholarly documents requires addressing current limitations in document layout segmentation systems (Section 2.2), semantic and context processing (Section 2.3), and question answering systems (Section 2.4). General purpose and domain specific datasets provide essential complementary resources in this context and are reviewed in Section 2.2.

Examining these areas allows us to place our work within the current literature and to contextualize our contributions, which improve document segmentation through semantic coherence checks and formal constraints analysis, while enabling multilayer document exploration through conversational agents that mitigate LLM long context limitations through question classification.

2.1. Text and layout segmentation

Text extraction from scholarly documents can be approached through multiple strategies, some of which aim to preserve the logical and structural organization of a document. Relations among layout elements are commonly inferred by associating visual cues such as spatial position and typographic features with document components (Xu et al., 2020), Park et al. (2019). In other cases, when suitable source files are available, the neded information can be obtained by converting documents into structured representations such as XML (Lopez, 2009). While effective for downstream processing, these approaches inevitably introduce conversion errors, limiting the reliability of the extracted structures.

To mitigate this issue, several works leverage the original document sources (when available), where structural information is explicitly encoded and can serve as a faithful reference. Many of these solutions rely on text and LaTeX parsing, which are able to preserve logical structures more accurately than visual extraction methods. In this direction, existing datasets and tools for scholarly document analysis primarily focus on layout analysis, text and visual element extraction and structure iden-

tification, often addressing challenges related to layout variability and interoperability.

Representative tools for converting LaTeX into XML include Pandoc (MacFarlane, 2012–2025) and NIST's LaTeXML (Ginev et al., 2011), Ginev and Miller (2013), Ginev et al. (2014), the latter serving as the backbone of the recent Ar5iv project (Müller, 2023). LaTeXML converts LaTeX documents into XML while retaining a substantial portion of the original semantic markup, enabling further transformation into formats such as HTML, EPUB, JATS, and TEI. This semantic fidelity distinguishes it from other LaTeX-to-XML processors and motivates its use in maintaining continuously updated corpora of ArXiv documents. Alternative approaches prioritize human-readable unicode output from LaTeX sources, but typically do so at the cost of reduced structural expressiveness (Grimm, 2008).

Overall, existing approaches expose a trade off between visual flexibility and semantic coherence, while lacking explicit mechanisms for validating structural accuracy. This gap motivates the exploration of methods that integrate source-aware representations with semantic consistency checks.

2.2. Document segmentation

Various approaches exist for identifying layout elements in visually structured documents, typically targeting specific layout classes like tables, formulas, or bibliographic references. Logical structure identification is often driven by visual features, though some methods use text and font features with transformers and transfer learning. Most approaches rely on OCR and TEI-XML conversion (Tkaczyk et al., 2015); for multi-page documents, current methods process pages independently with Mask R-CNN and language models, aggregating structure afterward.

The complexity of document structure induction from PDFs has led some efforts in the NLP community towards manual annotation of documents (Neumann et al., 2021), LabelImg (2022), which is inconvenient given the amount of data needed to train recent machine learning models. To this end, data augmentation can also be performed (Pisaneschi et al., 2023) on these datasets.

With respect to automatic segmentation systems, there are several different approaches for identifying and classifying layout elements. A large number of them is specific for a single layout class like tables (Shafait & Smith, 2010), charts (Tanasă & Oprea, 2025), formulas (Tran et al., 2021), bibliographic references (Lopez, 2009) and many others.

According to Tkaczyk et al. (2018), Grobid is a well established tool for extracting bibliographic data from document images; it allows multi-page analysis and it is able to extract also other data related to *title*, *authors*, *abstract*, *paragraphs*, *section headings*, etc. Logical structure recovery is usually performed exploiting visual features, with few exceptions that rely only on text and font features (Huang et al., 2022a), Kashyap and Kan (2020), Shinyama (2015) making use of transformers and transfer learning. Some approaches try to extract the logical structure of visually structured documents by using OCR tools and R-CNN-based approaches. The problem can be seen as the prediction of transition arc labels among text blocks that are mapped to a tree (Koreeda & Manning, 2021).

Whole document analysis increases the problem complexity, requiring to take into account relations between elements in different pages or semantic regions that span through multiple pages. To this end, some efforts (Arif Demirtaş et al., 2022) attempted to extract inter-page dependencies in the form of triples like [`head page`, `next`, `continuation` and [`head page`, `back`, `back page` using a neural-based dependency parser built upon SpaCy. These strategies are multi-modal, as they rely on visual feature extraction for both structure and text features. In fact, approaches like DocParser (Rausch et al., 2021) and DocStruct (Wang et al., 2020) succeed in extracting several semantic regions and their relations from documents, although being based on document renderings limits the extraction to a single page. DocParser is also exploited in HRDOC (Ma et al., 2023) to evaluate the hierarchical multi-page document reconstruction.

More recent segmentation approaches make use of LLMs and include multi-modal reasoning, that is particularly relevant for complex layout classes such as tables and figures. Approaches like M-LongDoc (Chia et al., 2025) demonstrate that long documents require specific strategies to maintain robustness when reasoning over multi-modal documents. Architectures like MDocAgent (Han et al., 2025) leverage specialized agents for text and images to integrate visual and textual cues, improving the detection of figure/caption and table/caption associations and enhancing overall segmentation accuracy. DocKylin (Zhang et al., 2025) addresses the computational challenges of high resolution visual layouts by performing slimming at pixel and token level, enabling more efficient and focused processing of dense tables and charts. Liu et al. (2025) proposes an approach that preserves document structure by encoding document elements using the LaTeX paradigm. This method maintains hierarchical and spatial relationships, guiding attention toward semantically meaningful regions and reducing errors in segmentation and question answering. Finally, Docopilot (Duan et al., 2025) provides a native multi-modal model capable of handling multi-page dependencies without external retrieval, improving coherence and accuracy for complex document structures.

Effective solutions for layout extraction can be found in commercial and open source systems. MinerU (Wang et al., 2024b), He et al. (2024) is able to detect *titles*, *body paragraphs*, *images*, *image captions*, *tables*, *table captions*, *image table notes*, *inline formulas* and *formula labels* (Zhao et al., 2024), Wang et al. (2024), and discard classes (such as *headers*, *footers*, *page numbers*, and *page notes*). While MinerU is designed primarily to convert PDFs to structured content, alternative approaches offer end-to-end pipelines: the recent PP-StructureV3 (Cui et al., 2025a) is a leading example that integrates PaddleOCR (Cui et al., 2025b), layout analysis, table and formula recognition, and document reconstruction, supporting a wide range of document intelligence applications.

Among commercial solutions, Amazon Textract (Amazon Web Services, 2025) delivers excellent results, being able to extract layout classes such as *title*, *header*, *footer*, *section title*, *page number*, *list*, *figure*, *table*, *key value* (formula) and *text*, also offering question answering functionalities. However, these systems primarily focus on accurate content extraction and readability, with limited emphasis on validating the semantic coherence of extracted layout relations or supporting structured, document-level interaction.

Even if HRDoc and DocStruct address hierarchical reconstruction at different granularities, they remain limited to structural recovery and do not provide mechanisms for validating the semantic coherence of layout relations across pages. Furthermore, although several systems such as DocStruct can generate multi-page annotated datasets or extract parent-child relations between visual elements, they often lack interactive interfaces that enable multi-page navigation, inspection, or conversational access to document content. Solutions like MinerU offer strong capabilities for layout annotation and content extraction but focus post-processing on improving accuracy, ordering, and readability rather than structural and semantic validation.

2.3. Semantic linking

Since the semantic web era, the most effective technologies for linking data to concepts have been Linked (Open) Data and ontologies, which provide abstract knowledge representations and allow inference through relations among entities. Such formal structures represent relations among document elements independently of the underlying data, enabling scalability and supporting reasoning that derives additional relations from those explicitly defined in the model. Ontologies can also be extended to link concepts across domains, enabling retrieval of data conceptually related to an entity even if not explicitly part of it. Populating ontologies with data can be automated using NLP and computer vision techniques to extract information from raw text and images.

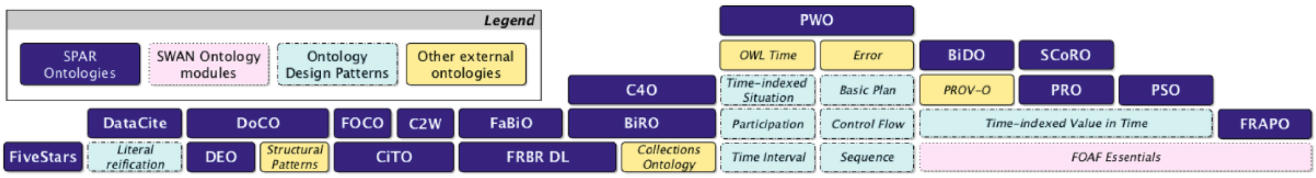


Fig. 1. The SPAR ontologies (Constantin et al., 2016).

There are both technical and pragmatic reasons (Ciotti, 2018) to represent document structures (e.g., TEI-XML) with ontologies (Bedini et al., 2011), Eide (2014). Extracting searchable content via NLP and computer vision alone does not guarantee preservation of the original visual or logical structure, which is essential for performing contextual analysis and structured queries. Without a formal structure, all data attributes must be explicitly declared, leading to redundancy in storage and computational cost. Modern systems (Formal et al., 2024) use indexes and pre-trained models to retrieve large volumes of data efficiently, but they do not provide scalable semantic consistency at low computational cost. By defining a structure capable of hosting extracted data it is possible to maintain the context and enable its expansion using relations that are defined in the structure.

Formal structures can be exploited to detect violations of structural constraints, such as overlapping layout elements linked by a disjoint relation, and can support query expansion with semantically related terms (Massai, 2024). In scholarly documents, distinguishing a person's role as the author of the paper versus the author of a cited work often relies on visual cues such as page location or section placement. Metadata from sources like CrossRef (Hendricks et al., 2020) can further enrich missing information, linking entities to an extended context.

Significant efforts toward unified ontological modeling include the SPAR ontologies project (Peroni & Shotton, 2018), Persiani et al. (2022), which integrates several ontologies (Fig. 1), including the Document Components Ontology (DoCo) (Constantin et al., 2016), the Pattern and Discourse Elements Ontologies (DEO), bibliographic resources and citation ontologies (Peroni & Shotton, 2012), among others. DoCo combines a *rhetorical* layer describing logical entities such as *references*, *results*, *related work*, etc with a *structural* layer linking them to more granular entities such as *title*, *figure*, *table*, *figures*, etc. Entities define relations one to each other; for example, `Sentence` includes `DiscourseElement` when it occurs with the attribute `inline` (`Sentence` $\subseteq$ `deo:DiscourseElement` $\cap$ `po:inline`). The ontology is populated using PDFX (Constantin et al., 2013), which extracts 19 layout classes with geometric coordinates from PDFs, and Di Iorio et al. (2013) to convert them into Utopia Documents XML (Attwood et al., 2010).

Ontology-based approaches demonstrate the value of formal semantics for checking consistency, yet they are typically not designed to enhance segmentation robustness or to detect anomalies such as not admissible overlaps or class inconsistencies

2.4. Visual question answering

Visual Question Answering (VQA) is one of the main points of contact between the communities of natural language processing and computer vision. VQA is often addressed with vision language pre-training approaches, which include image to text, computer vision, and video to text tasks (Gan et al., 2022). VQA can be exploited to address problems like question reasoning, object recognition, and design of conversational agents (Achiam et al., 2023).

Most VQA approaches rely on deep learning and multi-modal reasoning architectures. Systems such as MDETR (Kamath et al., 2021) and CLIP (Radford et al., 2021) use Convolutional Neural Networks (CNNs) for object detection paired with BERT models (Devlin et al., 2019) to extract features from captions and learning visual concepts from natural language supervision. Although these architectures effectively learn visual concepts from natural language supervision, they primarily operate at the image level and, with few exceptions (Deng et al., 2025), offering limited support for document-scale structure and context extraction. Also, user-generated data can be used as a large data source for training, exploiting repositories like ArXiv.; however, relying on completely arbitrary layouts can lead to annotation errors. The SPIQA dataset (Pramanick et al., 2025), built from 25,859 ArXiv documents and focused on images and tables, exemplifies this trend, providing scale at the cost of reduced control over semantic consistency and cross-page dependencies.

Natural language provides the most intuitive interface for navigating and interacting with linked data, motivating the widespread interest in conversational agents and chatbots (Mishra et al., 2022). Although these systems can integrate neural models with ontologies (Massai et al., 2019) and exploit graph reasoning mechanisms (Zhang, 2023) to extend the effective context of large language models, their reasoning capabilities remain strongly dependent on the quality and structure of the underlying representations. The aims of natural language processing systems include intent understanding, dialogue management, and the generation of context-aware responses based on conversational cues (e.g., sensor data and user preferences); however, these mechanisms alone are not sufficient to compensate the errors introduced by large-scale, automatically annotated training data.

Question answering systems can be integrated and trained to respond to questions of type *Who*, *What*, *Where*, *When*, *Why* on both visual and contextual information and can also operate as recommender systems by suggesting related topics, authors, or journal information upon data ingestion.

Retrieval-augmented generation (RAG) (Yu et al., 2024), Li et al. (2022), Gao et al. (2023) can enhance these capabilities by combining retrieval mechanisms with generative models and enabling access to external or domain-specific knowledge bases. However, the effectiveness of these strategies depends on the quality and coherence of the underlying structure, which is hard to construct without supervision and difficult to keep up to date. Recent RAG techniques, such as Toolformer (Schick et al., 2024), exploit APIs to access external tools that supplement answers and partially compensate for inherent LLM limitations like long-context issues. Similarly, KnowledGPT (Wang et al., 2023) integrates structured knowledge bases using program-of-thought prompting, enabling responses to questions requiring broader contextual knowledge. Applications of RAG to scholarly articles include ChatDOC[1] and PaperQA (Lála et al., 2023), which demonstrate the potential of these agents for scientific question answering. Despite leveraging external tools and knowledge bases, these approaches do not necessarily improve the quality or reliability of the answers produced; furthermore, integrating these systems into larger pipelines remains challenging, particularly when LLM outputs vary unpredictably, potentially compromising stability and reproducibility.

Unlike standard NLP tasks, which operate on unstructured text, vision-language tasks process visual elements jointly with textual content. Document images pose additional challenges due to their spatial organization and heterogeneous visual-textual composition. Models such as LayoutLMv3 (Huang et al., 2022b) address these challenges by build-

[1] https://chatdoc.com/

ing on the transformer architecture, incorporating 2D positional embeddings and unified visual-text representations and leveraging pre-training objectives such as image-text alignment.

Recent surveys on understanding visually rich documents emphasize not only deep learning frameworks that integrate text, layout, and visual features, but also the design of interfaces enabling interactive exploration and structured query over documents (Ding et al., 2024). In the context of document visual question answering, work reviewing methods that combine layout and language understanding highlights persistent challenges in supporting multi-page reasoning and contextualized interaction, which is directly related to interface design (Barboule et al., 2025). Additionally, studies on natural language and dialogue-driven interfaces for data visualization demonstrate mechanisms for intuitive query and interaction, providing insights that can be transferred to interfaces for document VQA (Shen et al., 2022).

The commitment of major leading companies in developing commercial systems reflects these trends. Google NotebookLM[2] supports question answering over user-provided PDFs, offering summarization and concept mapping functionalities. Amazon Textract[3] offers solid segmentation and question answering capabilities for structured documents such as pay stubs, financial records, vaccination cards, forms, etc; however, it supports a limited set of layout classes and operates primarily at single-page level, without enabling multi-page reasoning. ChatPDF,[4] currently based on GPT-3.5, offers effective QA over multi-page PDFs with strong comprehension of natural language, also featuring summarization and detection of AI-generated content.

While achieving good performance on page-level and standard tasks, existing VQA and conversational systems remain largely constrained by their reliance on single-page image representations, which limits multi-page reasoning and the preservation of structural and semantic coherence. Initial efforts have been proposed to address these limitations (Napolitano et al., 2024), yet comprehensive document-level reasoning and robust long-context support remain open challenges, highlighting key areas for further investigation.

2.5. Datasets

Several multi-purpose datasets exist in the area of scholarly document analysis, focusing on layout analysis, text and visual elements extraction and document structure identification. The broader datasets must deal with the variability of layouts of scholarly articles, addressing the difficulties of storing different data types into suitable structures. For this reason the most extended datasets rely on flexible data containers like XML and JSON formats, which allow versatility for managing such a variety of descriptive data.

Among recent datasets for scholarly documents layout analysis DocBank (Li et al., 2020) and Publaynet (Zhong et al., 2019) are considered the most relevant, although they exhibit limited variability in content and layout, respectively. Moreover, while providing an XML counterpart for the analyzed PDFs, DocBank and PubLayNet cover a few layout categories compared to other smaller datasets and are both based on single-page conversion, which excludes applications in whole-document tasks like document question answering and document generation. Still limited to single-page documents, while having more layout variability with respect to Publaynet and Docbank, Doclaynet (Pfitzmann et al., 2022) presents a document layout annotation dataset in COCO format, containing 80k pages manually annotated with 11 layout classes, while Markewich et al. (2022) distinguish 43 different layout classes.

Recent datasets like HRDOC (Ma et al., 2023) support document structure reconstruction by converting multi-page documents into hierarchical representations, using OCR to generate XML and covering 14 classes. VQA datasets supporting multi-page documents are hard to find; among the most recent, comprehensive resources can be found in the MP-DocVQA (Tito et al., 2023), the GRAM (Blau et al., 2024) and the DUDE datasets (Van Landeghem et al., 2023). These datasets are focused on document VQA tasks and aim at fulfilling general purpose objectives such as domain generalization and task-agnosticism of the understanding process. The DUDE dataset includes a wide range of document types and sources. It covers diverse topics and layouts, giving full support for multi-page analysis unlike other multi-purpose datasets like (Minesh et al., 2020), however having limited layout semantics. The lack of valuable multi-page datasets can also be addressed through document generation (Pisaneschi et al., 2023).

To the best of our knowledge, the most comprehensive dataset is the Semantic Scholar Open Research Corpus (S2ORC) dataset (Lo et al., 2020), which is composed by 8.1M open-access PDF-parsed papers across different academic disciplines. The S2ORC corpus is triple-size with respect to the open-access subset of Pubmed Central in 2020 (Roberts, 2001) and the whole dataset is composed by 81.1M academic papers obtained from the Semantic Scholar literature corpus (Ammar et al., 2018) using ScienceParse and Grobid. Similarly to other relevant datasets like (Kershaw & Koeling, 2020) and Bozsolik et al. (2024), information is described in JSON format. The lexicon is based on the S2ORC data categorization, which has ground in the Microsoft Academic Graph and more recently in the Semantic Scholar Academic Graph (S2AG), Wade (2022). 1.5M of documents in the corpus are also LaTeX-parsed providing near-perfect *citation*, *reference*, *caption*, *section header*, and *equation* extraction. In the S2ORC repository[5] can be found the python scripts to parse TEI-XML and LaTeX source into the S2ORC JSON. S2ORC can also be exploited for language model pre-training, using citation contexts to address tasks like bibliometric analysis and similar papers identification.

Other resources take into account LaTeX data as source for integrating the extracted information. Saier and Färber (2019) use the LaTeX source to extract text, citation spans, and bibliography entries, linking papers to references extracted with NeuralParscit (Prasad et al., 2018), using the Microsoft Academic Graph for metadata resolution. The result is the UnarXive dataset (Saier et al., 2023), composed by 1M JSON files generated from all ArXiv submissions, which are approximately 1.4M. UnarXive exploits LaTeXML for LaTeX parsing and metadata extraction, making also use of LaTeXpand[6] for aggregating different sources in a single XML file. In the most recent version of this dataset full-text, tables, figures and equations are also available and the number of documents rises to 1.9M. Tralics[7] is also preferred over LaTeXML for generating the XML data and Grobid replaces NeuralParscit for reference parsing, consistently with Tkaczyk et al. (2018) evaluation. Since the Microsoft Academic Graph is no longer maintained, OpenAlex (Priem et al., 2022) is chosen for reference resolution. An interesting application for this information source is that it seems well suited for extracting high quality plain text and accurate citation information from documents. In fact, it has been used to generate ground truths for the evaluation of PDF-to-text conversion tools (Bast & Korzen, 2017).

The S2ORC and the UnarXive datasets are the most comprehensive and heterogeneous resources, even if both lack geometric information about layout elements. A similar purpose resource exhibiting also geometric coordinates can be found in the Article Regions Dataset (ARD) (Soto & Yoo, 2019) which has few (~100) single-page documents while having a discrete number (9) of different classes.

Although these datasets provide a valuable basis for layout, content, and multi-page reasoning, their scale precludes exhaustive manual validation, and automatic annotation inevitably introduces errors that persist in the data. Existing resources rarely incorporate effective mech-

[2] https://notebooklm.google.com/
[3] https://aws.amazon.com/en/textract/
[4] https://www.chatpdf.com/en
[5] https://github.com/allenai/s2orc
[6] https://ctan.org/pkg/latexpand
[7] https://www-sop.inria.fr/marelle/tralics/

anisms for systematic post-processing or quality control, which limits their reliability for model training and evaluation. Approaches leveraging formal semantic models for automated validation and refinement could substantially enhance dataset consistency and support more robust document-level analysis.

In conclusion, the analysis of current methods reveals four main gaps:

1. the lack of interoperability among different document representations and associated information
2. limited support for improving document segmentation with formal validation of the resulting layout
3. limited availability of interactive interfaces supporting multi-page analysis, navigation, and question answering
4. long-context issues in LLM-based interaction.

These gaps define the areas that we investigate in this work and that we address in developing our framework.

3. System architecture

All the shortcomings identified in Section 2 are addressed in an application scenario by designing an architecture (Fig. 2) capable of performing document layout analysis and interaction. The proposed system aims at extracting different layers of information from multi-page scholarly articles, making them accessible through state of the art computer vision and natural language technologies and performing formal validation of the results.

Multi-page segmentation capability and how its limitations are dealt with is described in Sections 3.1 and 5. The architecture relies on a semantic structure for accounting relations among the extracted layout elements; the way it is integrated into the architecture and exploited to improve segmentation is described in Sections 3.2 and 5. The interoperability among document representations and the access to the associated information are addressed through a specifically designed user interface that exploits conversational agents based on language models. Its components are described in Sections 4 and 3.3.

3.1. Document segmentation module

To extract geometric information, a segmentation strategy is presented to identify layout elements and their properties. The PDF articles are converted to the TEI-XML format through the Grobid API[8] in order to extract the PDF structure as an XML tree. The resulting output contains the recognized regions that belong to 17 categories: *title*, *doi*, *keywords*, *abstract*, *authors*, *authors data*, *emails*, *tables*, *figures*, *captions*, *formulas*, *dates*, *sections/subsections*, *acknowledgements*, *bibliographic entries* and *raw text blocks*.

Positional information includes page number and is present for most classes. Some structures have deeper characterization in the Grobid processing, for instance, author consolidation is performed through integration with the CrossRef APIs (CrossRef, 2024).

The system also supports LaTeX input: to this end, the LateXML tool is exploited to generate the XML representation of the TeX source. The output of the LateXML conversion is then serialized through Beautiful Soup (Richardson, 2007) and the information that is extracted from the LaTeX source (mostly tags and text) is represented as key-value pairs and stored to support the rest of the processing steps. In the case of LaTeX input, the sources are then compiled and the pipeline for processing PDF files is applied.

The output of Grobid processing is parsed through Beautiful Soup to extract TEI tags and serialize the information into key-value pairs that are enriched by semantic processing (Section 3.2) and improved through LLM-based correction (Section 5). The recognized semantic regions are then associated to coordinates, extracted through Grobid as well, that are later used to draw bounding boxes on the original document. The hierarchy of the document is also extracted to better characterize its structure.

The layout elements associated with geometric information are used in the interface to draw different colored layers on the input PDF. The information that is not provided with spatial data is visually associated to them through linked pop-ups (Fig. 3).

Within the proposed architecture, document segmentation is improved through a post-processing phase that enforces structural and semantic constraints among layout elements (Section 3.2). Unlike approaches such as HRDOC (Ma et al., 2023) and DocStruct (Wang et al., 2020), which primarily reconstruct hierarchical relations, DocLap segmentation is refined by validating the coherence of spatial and class-level relations, enabling the detection of invalid overlaps and incompatible layout configurations. As in HRDOC, segmentation operates at multi-page level, while differing from page-centric systems such as DocStruct. MinerU (Wang et al., 2024b) provides comparable segmentation outputs to DocLap (Table 3), also offering a post-processing phase to improve segmentation. In systems like MinerU the post-processing phase focuses on improving accuracy, ordering, and readability of extracted content; by contrast, our architecture targets structural and semantic validation to explicitly address segmentation errors.

3.2. Semantic linking module

The semantic characterization assigned to the extracted information is derived from the Document Components Ontology (DoCO) (Constantin et al., 2016). The DoCO ontology, which is described in Section 2.3, is a specialized ontology designed for modeling layout elements of scholarly documents. As described in Section 5, the properties of layout elements included in the ontology are leveraged to improve segmentation capabilities. This is achieved by using the logical relationships among ontology classes to identify *not admissible* overlaps.

The semantic network provided by DoCO is imported into the system, enabling a structured and standardized representation of document components. To ensure compatibility, a detailed mapping process is performed between the Grobid tags and the corresponding DoCO classes. This mapping is carried out by aligning the intended meaning of tags, ensuring semantic coherence and consistency across the extracted data.

3.3. Question answering module

The question answering module is managed by exploiting LLMs, specifically the open source LLaVA 1.6 model (Liu et al., 2024) through the Ollama[9] API. This model has been chosen for its lightweight multimodal capabilities and its seamless integration in production frameworks. The question answering module is designed to allow the LLM to access both the page images extracted from the input and the serialized output of Grobid.

The conversation context is given to the LLM as key-vaue pairs extracted by the system, corresponding to the relevant sections of the Grobid output and the images on which they occur. The results obtained with LLaVA ensure understanding of the questions in relation to state of the art systems, given the resources provided and their eventual lack.

The user question is augmented and proposed to the LLM with the prompt:

"You are given a question about a research article. Consider the following context extracted from it: `key_value_pairs`. *Now answer this question:* `input`. *(Please answer in one sentence only)"*.

In this statement, `key_value_pairs` represents the output of the modules described in Sections 3.1–3.3 and `input` is the query that the user can write in the interface.

[8] https://kermitt2-grobid.hf.space/

[9] https://ollama.com/

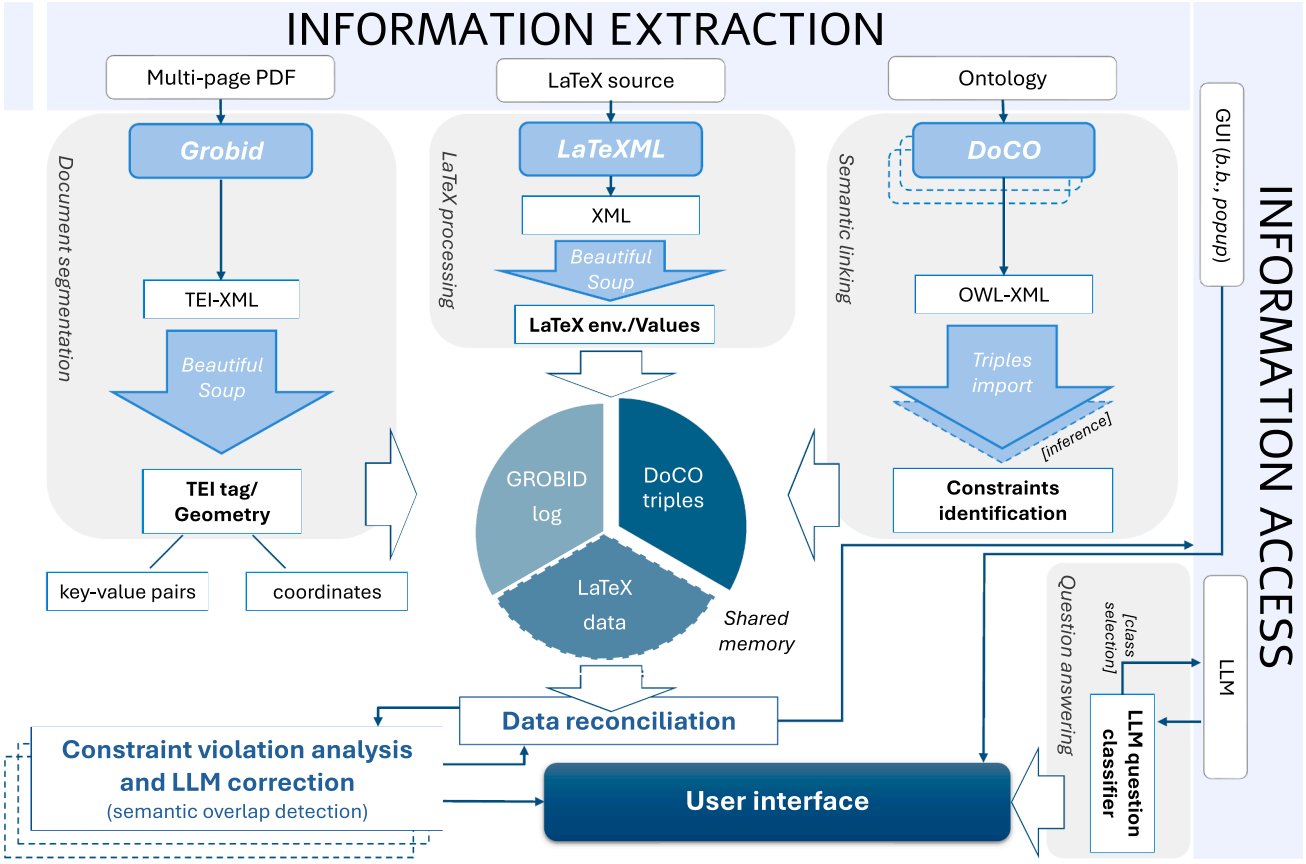


Fig. 2. Overview of the DocLap architecture. The left side shows document segmentation and LaTeX processing modules, which process both PDF and LaTeX inputs to extract key-value pairs. The semantic linking module shows how the DoCO ontology is exploited to extract the semantic constraints that are used to improve segmentation. The central assembly phase reconciles these outputs and prepares the data for error detection and LLM correction. The right side depicts the information access component, showing the user interface supported by the LLM question answering module with context classification.

The context length for LLaVA is 4096 tokens, which can be restrictive considering the images and the key-value pairs resulting from the output of the segmentation and semantic linking modules. To address the LLM context length issues and limit the context to the part that is most pertinent to the question, the user interface described in Section 4 features an LLM question classifier module that exploits the `gpt-oss:120b-cloud` engine[10] to provide context for the question answering functionality. The LLM context is classified into 12 classes: *General info*, *Header data*, *Abstract*, *Figures*, *Tables*, *Formulas*, *Sections*, *Links*, *Notes*, *Acknowledgements*, *References*, and *Phrase*, that are referenced in the prompt as `class_list`.

The aim of this component is to provide information for filtering the serialized output of Grobid based on the LLM question classification, that is performed with the prompt:

"You are given a phrase from a research article. Classify it using one of these structure labels (comma-separated): `class_list`. *The label Header data includes the article title and author names.*

Note: The correct class label will most likely appear in the phrase. Output format (no extra text): [your selected label]. Now classify this phrase: `question`".

As can be seen in Fig. 3, the system also includes the ability for the user to disable this functionality and autonomously provide a classification of the question, choosing any combination of the labels listed above. These classes correspond to TEI-XML tags, which are afterwards associated with the DoCO ontology entities. The labels classified by the system or provided by the user are exploited to split the context that is given to the LLaVA engine, retaining only the portions that are considered relevant for answering the user question. A comprehensive evaluation of the question answering capabilities of DocLap, including the details of the MP-DocVQA dataset used for evaluation, can be found in Section 6.3.

4. User interface

The DocLap system is freely available at: https://github.com/lmassai/DocLap and consists of five web pages interacting with a Python server that routes the user choices, exploiting custom utilities and routines aimed at managing the system's functionalities. Through the interface (Fig. 3) the user is able to upload a PDF document (*Upload page*) and process it (*PDF processor*), extracting information that the LLM can exploit to provide answers.

The system also supports LaTeX input (*LaTeX processor*): the TeX source is compiled and the flow remains the same of the PDF input; also, the key-value pairs extracted from the XML representation of the TeX source are available for the rest of the processing steps. Upon PDF processing, the document augmented with layers of colored bounding boxes spanning the classes described in Section 3.1 is shown to the user. The whole process is completed in variable time, mainly depending on the length of the PDF paper and network capabilities, since Grobid is used as a cloud service. In our experiments, processing a 10/20-pages paper takes a few seconds while longer papers may require more than 10 seconds. The serialized information is used as context for the LLM, which is included in the interface to ease user interaction and exploration of the system's functionalities. The LLM computation time for each ques-

[10] https://openai.com/index/introducing-gpt-oss/

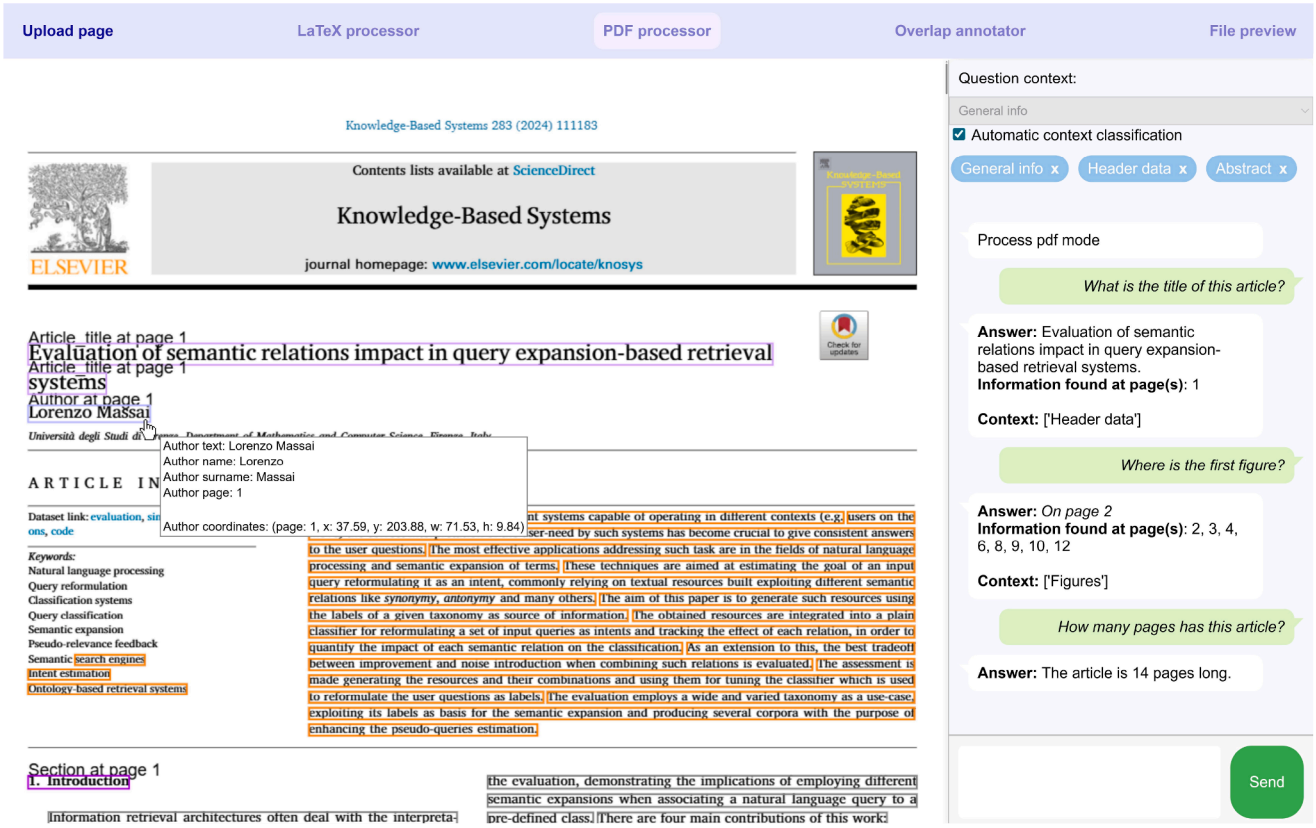


Fig. 3. User interface of the *PDF processor*. The interface is divided into two main sections: on the left, the input PDF is overlaid with different colored bounding boxes corresponding to layout classes; contextual pop-ups displaying element metadata (e.g., *Author*) appear on mouse hover. On the right, the conversational endpoint shows user queries and system responses, along with the automatically inferred question classification, which can be manually overridden to control the LLM context (best viewed in color).

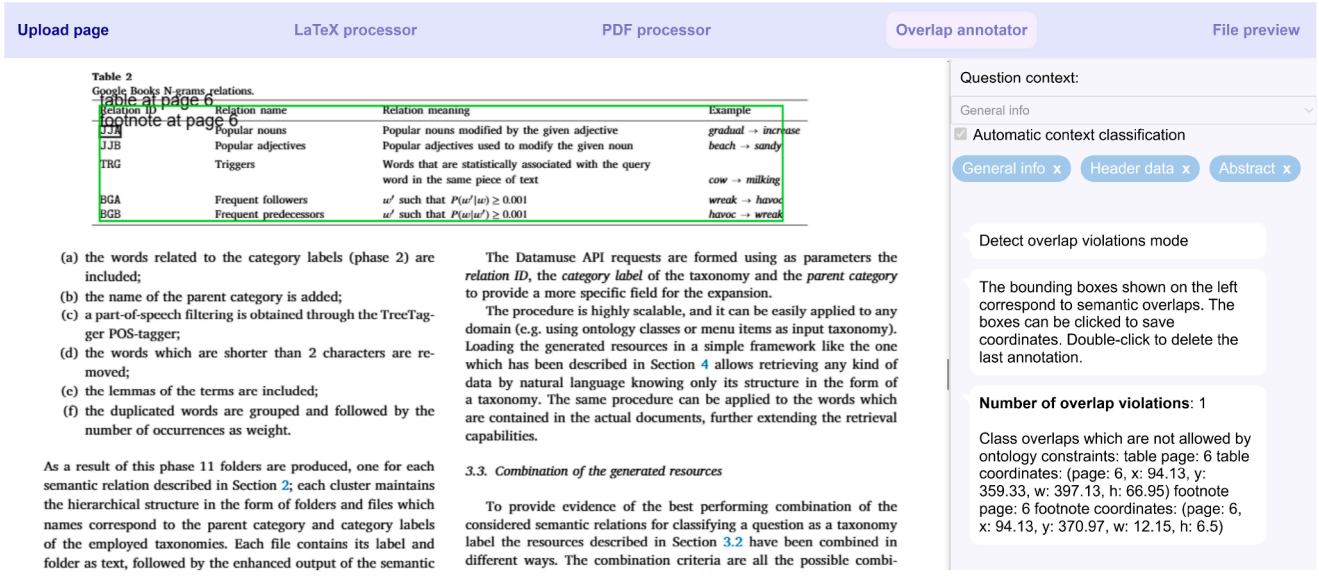


Fig. 4. Detection of non admissible overlaps based on semantic constraint analysis. The figure shows a semantic overlap highlighted in the *Overlap annotator* view, caused by a geometric intersection between *table* and *footnote* elements, which are defined as disjoint classes in the DoCO ontology (best viewed in color).

tion varies according to local GPU capabilities, generally taking a few seconds.

Since the LLM-based question classifier (Section 3.3) can significantly influence the final answer by filtering the extracted question context, the user is allowed to disable the automatic classification from the interface and select it manually. In this case, the user can provide an autonomous classification of the question selecting any number of labels, as described in Section 3.3. The interface provides a graphical preview of the input data, displaying PDF pages with overlaid bounding boxes that correspond to the layout elements identified by Grobid, each associated

Table 1
General axioms of the DoCO ontology; the layout elements in each row are disjoint from each other. The classes generating the *not admissible* overlaps in Fig. 4 are underlined.

All Disjoint Classes
back matter, body matter, captioned box, chapter, complex run-in quotation, footnote, formula, formula box, front matter, list, part, section, table;
abstract, afterword, appendix, colophon, foreword, glossary, index, list of figures, list of tables, preface, table of contents;
label, paragraph, subtitle, title;
list of authors, list of contributors, list of organizations;
sentence, simple run-in quotation, text chunk;

with its spatial coordinates and semantic label. In addition, informative pop-ups containing all data retrieved by data processing are shown.

The user interface also includes a specific view (*Overlap annotator*) that is designed to outline the semantic integration described in Section 5. The purpose of this view is to highlight the layout elements that overlap and violate the ontology constraints. This view also allows the user to keep track of the highlighted elements. From this view, the overlapping bounding boxes that are drawn on the PDF can also be selected to store their coordinates in a separate file for further processing.

5. Constraint violations analysis

To understand how the relations defined in an ontology can be applied to visual elements and how they can improve their classification, we consider the DoCO ontology.[11] The idea is to exploit the ontology relations to check the admissibility of some specific geometric relations that may occur among layout elements.

Since ontology axioms may involve entities that lack geometric information in the Grobid XML, while being associated with other metadata, relations are evaluated exclusively among entities for which spatial coordinates are available. Then, semantic properties can then be applied to such objects for detecting visual recognition errors such as overlap errors (Fig. 4).

We distinguish between *geometric* and *semantic overlaps*. Geometric overlaps are two bounding boxes that intersect. Semantic overlaps are geometric overlaps whose associated labels violate a constraint defined in the ontology. Building on the intuition that some geometric overlaps are admissible (e.g., *Name* and *Author*) and semantic overlaps are most likely recognition errors (e.g., *Title* and *Figure*), we try to establish a correlation between semantic overlaps and recognition errors. To perform such correlation, we exploit the DoCO ontology relations; in particular, to express the fact that an overlap is not admissible, the `owl:disjointWith` relation is considered.

To identify overlapping elements, they are described as rectangles with the following properties:

- **Page**: the page number (two elements on different pages cannot overlap)
- **x, y**: the coordinates of the top-left corner of the rectangle
- **width, height**: the width and height of the rectangle.

Overlap Criterion

Two elements overlap if and only if their rectangles intersect. Given two rectangles defined by:

1. Rectangle A: x_A, y_A, width_A, height_A
2. Rectangle B: x_B, y_B, width_B, height_B

To check for overlap, we verify that there is no separation between the two rectangles in both the horizontal and vertical dimensions.

Conditions for Non-Overlap

1. The rectangles do not overlap if one is entirely to the right of the other:

$$x_A + \text{width}_A \leq x_B \quad \text{or} \quad x_B + \text{width}_B \leq x_A$$

2. The rectangles do not overlap if one is entirely below the other:

$$y_A + \text{height}_A \leq y_B \quad \text{or} \quad y_B + \text{height}_B \leq y_A$$

Then, two rectangles A and B overlap if none of the above conditions are true. Formally, they overlap if all the following conditions are fulfilled:

$$x_A < x_B + \text{width}_B, \qquad x_B < x_A + \text{width}_A$$
$$y_A < y_B + \text{height}_B, \qquad y_B < y_A + \text{height}_A.$$

To identify the errors we focus on the rectangles that overlap. For each overlapping pair we classify it as *admissible* or *not admissible*. An overlap is *not admissible* when it involves classes that are disjoint in the ontology general axioms[12] as reported in Table 1.

The *Overlap annotator* section of the user interface described in Section 4 is intended to highlight the bounding boxes associated to classes that are not allowed to overlap (Fig. 4).

The overlapping bounding boxes are tuples in the format [*Layout_class*, *Page*, *x*, *y*, *w*, *h*], and are exploited also in the document segmentation module (Section 3.1) to improve the overall recognition capabilities of the system. This is performed by giving their coordinates to the constraint violation analysis and LLM correction module, which implements unsupervised interaction with the open source multi-modal LLaVA 1.6 model (Liu et al., 2023).

The LLaVA model is included in the system through the Ollama API. The model temperature is set to 0 for maximum determinism. Its contribution occurs at the end of the segmentation phase: the system generates annotations that are specific for semantic overlaps, containing the bounding boxes and the corresponding labels of the elements that are detected as semantic overlaps. These annotations are also used in the interface to draw on the PDF paper only the semantic overlaps (Figs. 4, 6). The annotation files and the corresponding pages are then converted to images that are used to feed LLaVA with the prompt:

"Given the attached image, which represents a page from a research article, it contains some overlapping bounding boxes, each associated with a label and coordinates in the format: label,page,x,y,w,h. The label and coordinates of two of the overlaps are given below: `lines`".

where `lines` is the set of (label, bounding box) pairs that overlap and violate a semantic constraint. The LLM is asked to classify as correct or incorrect the bounding boxes that participate in each overlap on a page,

[11] https://sparontologies.github.io/doco/current/doco.html#d4e145

[12] https://sparontologies.github.io/doco/current/doco.html#generalaxioms

Table 2
Class mapping among different segmentation engines wrt ground truth.

DocLap	Grobid	MinerU	Textract	PP-Structure-V3	DAD (GT)	DoCO ontology
Article_title	title	title	LAYOUT_TITLE	doc_title	title	title
Figure	figure	image	LAYOUT_FIGURE	image,chart	figure	figure
Caption_{Figure,Table}	Caption_{Figure,Table}	{image,table}_caption	*LAYOUT_TEXT*	{table,figure}_title	caption	{table,figure} label
Table	table	table	LAYOUT_TABLE	table	table	table
Abstract	abstract	*text*	*LAYOUT_TEXT*	abstract	abstract	abstract
Acknowledgements	Acknowledgements	*text*	*LAYOUT_TEXT*	text	Acknowledgements	Acknowledgements
Author	author	*text*	*LAYOUT_TEXT*	text	list of authors	list of authors
Reference	reference	*text*	*LAYOUT_TEXT*	reference, reference_content	reference	Reference
Formula	formula	interline_equation	LAYOUT_KEY_VALUE	formula, formula_number	formula	formula
Section	section_title	title	LAYOUT_SECTION_HEADER	paragraph_title	section_heading, subheading	section title
Link	Link, ref_to_elem	*text*	-	text	citation	Link
phrase	phrase	text	LAYOUT_TEXT	text	core_text	sentence
Note	Footnote	*text*	LAYOUT_FOOTER	footer, footnote	Footnote	footnote

specifying the meaning and expected content of each layout class in the prompt:

"For each block, one of the two is correct, the other is wrong. The goal is to verify which block is wrong. Correct means that it encloses only the expected content defined by the label (neither too much nor too little)".

To ensure correct integration with the workflow and avoid hallucinations, the prompt is completed with the statement:

"Return only the wrong bounding box in the format label,page,x,y,w,h. In the output please double-check that the block that you give is one of the two given as input, without any change neither in the label nor in the numbers. No explanation, no extra text, no comments.".

The overlapping bounding boxes that are classified by the LLM as incorrect are then excluded from the system's output. As shown in Section 6.1 this operation, that is a light classification task, enhances the overall classification capabilities of the system.

In the landscape of document layout analysis systems several approaches that employ post-processing strategies for similar purposes are present, such as Non-Maximum Suppression (NMS) and heuristics to detect overlapping regions. Some of these strategies are employed in recent frameworks such as MinerU 2.5 (Niu et al., 2025), PaddleOCR, and PP-Structure-V3 (Cui et al., 2025a), Cui et al. (2025b). However, our semantic validation goes beyond.

NMS and geometric heuristics operate exclusively on spatial information, whereas the proposed method exploits semantic knowledge encoded in an ontology. Consequently, the two approaches address different aspects of the problem and should be regarded as complementary. The ontology-based validation is intended as a semantic post-processing layer that complements existing detection pipelines.

While NMS and layout-specific heuristics address geometric ambiguities, our method targets semantic inconsistencies that remain undetectable when only spatial information is considered. In these cases, ontology constraints provide an additional source of information that is unavailable to standard NMS procedures.

Furthermore, explicit heuristic rules are usually handcrafted for specific spatial configurations (e.g., captions below figures, authors below titles). Our approach derives admissibility conditions directly from ontology relations, making the validation process independent of specific page layouts and at the same time explainable and extensible by updating the semantic model only.

6. Evaluation

Evaluating the performance of DocLap requires a comparison with state of the art systems in the main topics addressed by the system, which are *document segmentation* (Section 3.1), *semantic overlap detection* (Section 3.2), and *question answering* (Section 3.3).

This chapter presents a comprehensive assessment of DocLap capabilities in these three main topics using publicly available datasets and benchmark tools. The evaluation criteria focus on accuracy, efficiency, and robustness in handling multi-page scholarly documents.

6.1. Document segmentation

In this section are evaluated the segmentation capabilities of DocLap. For segmentation capability is intended the ability of the system to extract the coordinates of bounding boxes corresponding to layout elements from document pages and associate them to semantic classes (i.e. *title*, *author*, *table*, *caption*, etc).

Several challenges related to the availability of annotated multi-page datasets, to the output format comparability, and to layout class heterogeneity are discussed. Constraint violation analysis is a core component of the segmentation module; its assessment is shown in Section 6.2.

6.1.1. Methodology

The ability of the system to extract bounding boxes and their class must be evaluated against datasets of scholarly articles that maximize layout variability, original source availability, annotation format versatility, page number traceability, and the number of covered layout classes.

The dataset that best fits these requirements is the Dense Article Dataset (DAD) (Markewich et al., 2022) which is annotated by hand and contains various document layouts from several publishers.

The DAD dataset is also rich in layout classes, making feasible the evaluation of DocLap capabilities in recognizing a wide range of layout classes across different domains. As can be seen in Table 5 it spans different journals and publishers, maximizing the variety of different layouts that can be found across multiple disciplines and publishers.

Ground truth construction Since the DAD dataset is annotated by hand, it is suitable to be used as ground truth (GT). The dataset is composed of images that represent the document pages and associated JSON files that contain the annotations, which include page dimensions, region co-

Table 3
Segmentation performance of different engines (match: $IoU \geq 0.9$).

Model	Precision	Recall	Accuracy	F-measure	mAP@IoU
DocLap	0.82	**0.93**	**0.77**	**0.87**	0.74
Grobid	0.81	0.91	0.75	0.85	0.59
Textract	0.64	0.56	0.42	0.60	0.42
PP-Structure-V3	0.66	0.70	0.51	0.68	0.59
MinerU	**0.89**	0.59	0.55	0.71	**0.80**

ordinates, and semantic labels. Citation annotations maintain the same format of main annotations.

To fit the data format of DocLap and allow a reliable comparison, the DAD annotation files have been merged and represented in the format: [*File_name*, *Layout_class*, *Page*, *x*, *y*, *w*, *h*, *Publisher*, *Journal*, *Year*, *Area*, *Article_number_in_volume*], where coordinates refer to the bounding boxes of layout elements.

All the original documents of the DAD dataset are traced performing OCR on the first page of each document; this allows us to leverage the PDFs that have been used to build the dataset without relying on the images that are included into the dataset.

The labels and coordinates of layout elements are extracted from the annotation files and further integrated with annotation data from the dataset to fit the above format.

It is to be noticed that, like most datasets available online, DAD relies on single-page images. However, for our purposes the multi-page nature of the original documents needs to be recovered. This is viable in the DAD dataset since each annotation file refers to a single (image) page and page numbers are encoded into the image filenames.

This allows us to retrieve the original documents and to associate the annotation to the page number. Additional relevant information, such as the *DOI*, can be extracted from the filenames, while details like *journal* and *publisher* can be obtained from the annotation folder names.

Assessment. To ensure consistency when comparing the DocLap output with the DAD annotation format, it is necessary to perform a normalization step on both formats.

The coordinates of the bounding boxes are normalized in the range [0, 1] with respect to the original page dimensions, which are extracted from PDFs in the case of DocLap and from images in the case of DAD annotations. With respect to all class labels covered by the DAD dataset and by the DoCO ontology, and also considering the distribution of class labels among the adopted datasets, the name space is reduced to 12 classes.

The semantic units are aligned by their label (Table 2) and classes without associated coordinates are pruned. The elements emphasized in Table 2 refer to semantic units that are recognized by the engine with a more general label with respect to the ground truth (i.e. the *abstract* class in DAD corresponds to *text* in MinerU).

The elements that are not recognized by the engine are not mapped (marked with – in the table), that is the case of the *Link* class for the Textract engine. Since DAD does not separate figure caption and table caption labels, for the engines that are able to recognize them, these classes are mapped to the more general DAD class *caption*.

The same is done, in reverse, for the *section heading* and *subheading* classes, which are commonly referred as a single class. In this context, it is to be noticed that although MinerU is able to detect a discrete layout element, it does not define a section heading class and instead maps it to *title*.

Positional matching is performed exploiting Intersection over Union (IoU) between DocLap and DAD bounding boxes, with threshold set to 0.9%; also, since the annotation level of DocLap is performed at row level, for larger areas the total inclusion of a bounding box with corresponding label is considered a match.

Table 4
Class precision values for DocLap layout classes.

System class	Precision	Correct / Total
phrase	0.89	354,307 / 397,179
reference	0.93	58,493 / 62,585
link	0.25	11,653 / 45,928
caption_figure	0.46	13,345 / 28,858
abstract	0.84	7775 / 9286
section	0.89	7256 / 8118
figure	1.00	6708 / 6714
formula	0.75	3168 / 4247
acknowledgements	0.85	3159 / 3706
caption_table	0.60	1484 / 2457
author	0.77	1656 / 2148
article_title	0.94	1130 / 1208
table	0.93	1095 / 1181
note	0.25	222 / 898

With these assumptions, we define:

- a *true positive* is a row of the DocLap output which, with respect to the DAD ground truth, has the same *File_name*, the same *Page*, the same *Layout_class*,[13] and one of the following:
  - bounding box $IoU \geq 0.9$;
  - bounding box entirely included in a larger one with the same *Layout_class*;
- a *false positive* is a bounding box in the DocLap output that does not match any valid ground truth box according to the above-mentioned criteria;
- *true negatives* are not considered in this evaluation due to the nature of the task. Cases where neither the system nor the ground truth have a bounding box are not accounted (i.e. background, margin, spacing, etc);
- a *false negative* is any ground truth box that remains unmatched after processing all bounding boxes extracted by DocLap, meaning that the system was unable to detect it.

These counts are used to measure the performance metrics of precision, recall, accuracy, F-measure, and mean Average Precision at fixed threshold (mAP@IoU).

$$\textbf{Precision} = \frac{TP}{TP+FP} \qquad \textbf{Recall} = \frac{TP}{TP+FN}$$

$$\textbf{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

$$\textbf{F-measure} = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

$$\textbf{mAP@IoU} = \frac{1}{C}\sum_{c=1}^{C} AP_c$$

where AP_c is the average precision for class c, and C is the total number of classes.

In this case, precision represents the number of correctly detected bounding boxes among all bounding boxes predicted by the system. Recall represents the number of correctly detected bounding boxes among all bounding boxes. The F-measure is the harmonic mean of precision and recall, whereas accuracy quantifies the ratio of correctly detected bounding boxes with respect to the total number of relevant bounding boxes considered.

6.1.2. Results

In Table 3 are reported the DocLap capabilities in the segmentation task. The results are consistent with the state of the art performance of similar purpose multi-page segmentation engines. It is to be noticed

[13] with relaxed constraints for *caption_figure/caption_table* and *section_heading/sub_heading*

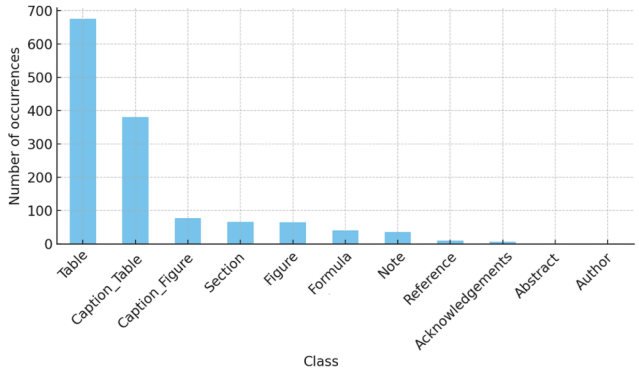


Fig. 5. Class distribution of labels participating to semantic overlaps (removed).

that some evaluated engines define additional classes with respect to the ground truth. Classes that cannot be mapped to the ground truth taxonomy (e.g., *header*, *aside_text* in PP-Structure-V3, *LAYOUT_LIST*, *PAGE* in Textract) are treated as false positives in the evaluation. The mean Average Precision (mAP) at IoU = 0.9 is computed independently for each layout class, then averaged across the number of classes of each dataset. For each class, average precision is defined as the ratio between correctly matched predictions and the total number of predictions. This macro-averaging highlights how segmentation performance is distributed across the layout categories, showing how well the system recognizes different classes. Table 4 reports DocLap precision values at class level.

As shown in Table 3, DocLap improves the segmentation output of the underlying engine, which in this case is Grobid, being able to remove overlap errors that occur during recognition. Since Grobid is used as the base segmentation engine, DocLap performance is comparable to that of Grobid, while the application of semantic constraint analysis enables the resolution of semantic overlap errors.

In Fig. 5 is reported the distribution of the labels associated with bounding boxes that participate in an overlap classified as *not admissible* and removed by the system. It is to be noticed that most of the corrections are performed in presence of tables and table captions, which are considered the hardest layout classes to detect and classify (Gemelli et al., 2022).

The evaluation script and the annotation files to assess each engine's performance are available at this link.

6.2. Semantic overlap detection

The evaluation of semantic overlap detection capabilities aims at demonstrating the correlation between semantic overlaps and recognition errors that is exploited to perform visual recognition enhancement. The assessment shows the ability of DocLap to find errors that occur in overlapping bounding boxes, which are not directly related to recognition errors (i.e. *link* over *abstract*). The identification of this class of errors aims at improving the recognition capabilities of segmentation-based systems in general.

6.2.1. Methodology

For the same reasons discussed in Section 6.1.1, the DAD dataset is found to be adequate and adopted for this evaluation as well.

The purpose of this assessment is to express the correlation between semantic overlaps, and recognition errors. The assessment shows the capability of the system to find annotation errors independently of the recognition engine.

Ground truth construction To evaluate the capabilities of DocLap to detect and correct overlap recognition errors, it is necessary to build a ground truth of potential errors. To build such a resource, all PDF files are processed performing OCR on the first image of each document in the DAD dataset.

To align the annotation format of the DAD dataset with our system's, its lexicon is standardized to match both our terminology and the DoCO taxonomy. Specific scripts have been implemented for each annotation set, with the purpose of extracting the information needed to fit the format used in Section 6.1.1. Labels, coordinates, page numbers and other metadata are extracted likewise.

In this scenario, semantic overlaps are computed using a custom overlap detection view in the interface (Fig. 4), specifically designed to accept PDF input and show semantic overlaps.

Through this interface an annotator is able to visualize the extracted bounding boxes and store them as actual errors for the ground truth. In summary, the evaluation data for the semantic linking module consist of three data sources, which are the cumulative annotations, the cumulative semantic overlaps and the ground truth data, which are made available online[14] for research purposes.

Assessment By comparing with the ground truth it is possible to compute precision, recall, F-measure, and accuracy.

In this case, precision represents the capability of the semantic linking module to detect actual errors which are annotated in the ground truth. This allows us to demonstrate the correlation between semantic overlaps and recognition errors.

Recall is used to measure the system's capability to retrieve all the overlap errors that are present in the annotated documents.

F-measure is defined according to the standard definition and accuracy quantifies the ability of the classifier to match the expected results.

The metrics defined in Section 6.1.1 are calculated considering the number of True Positives (TP), False Positives (FP), False Negatives (FN) and True Negatives (TN) as follows:

- a *true positive* is considered a bounding box detected by DocLap as part of a semantic overlap that is also present in the ground truth;
- a *false positive* is considered a bounding box detected by DocLap as part of a semantic overlap that is not contained in the ground truth;
- a *true negative* is considered a bounding box that is not detected by DocLap as part of a semantic overlap and is not contained in the ground truth as well;
- a *false negative* is considered a bounding box that is not detected by DocLap as part of a semantic overlap and the ground truth contains its coordinates.

  However, false negatives never occur in this analysis due to the ground truth construction method, which involves the annotation of detected overlaps.

In Fig. 6 an example of some of the above-mentioned cases is reported: the *table* (green) bounding box overlaps with several *table_label* bounding boxes (dark red), thus DocLap highlights all of them. The same happens with the *formula* (blue) and the *section_title* (red) bounding boxes. The list of overlaps detected by DocLap contains all these bounding boxes, a subset of which is also included in the ground truth, containing the errors. In this case, all the *table_label* and the *formula* bounding boxes are identified as TP, having been identified by the system as semantic overlaps that are also contained in the ground truth as errors. The *section_title* and *table* bounding boxes are, in contrast, identified as FP since they are identified by the system as part of an overlap, while not being errors.

6.2.2. Results

Table 6 shows the relation between semantic overlaps and recognition errors, expressing DocLap performance in detecting errors among overlapping bounding boxes. These values refer to the subset of errors involving semantic overlaps, which is different from the system's overall

[14] https://drive.google.com/file/d/1nFzwtiXEkcXj7V4dpLO-239O2lNhR3lf/view?usp=sharing

Table 5
Frequencies (%) of different layout classes among journals.

Journal	Ack.	Link	Ref.	Abstract	Fig.	Fig.label	Foot.	Math	Authors	Section	Tab.	Tab.label	Title
Advanced Science	4.32	23.92	25.51	6.71	9.53	22.18	0.9	0.49	3.23	2.22	0.1	0.23	0.65
Advances in Mech. Eng.	0.65	15.11	25.29	4.25	10.75	33.15	0.26	3.84	1.12	3.07	0.65	1.03	0.83
Ampersand	1.86	33.52	37.98	3.13	1.20	12.81	0.93	0.24	0.54	4.42	1.03	1.80	0.55
Asia Pac. Policy Stud.	2.19	26.99	45.79	8.45	0.66	5.50	1.50	0.50	0.94	4.05	0.72	1.75	0.96
Brain and Behavior	0.14	13.27	41.56	4.22	0.01	32.08	0.26	0.24	1.11	3.73	0.26	2.68	0.44
Int. J. Eng. Sci.	2.09	33.67	19.10	4.16	4.16	14.24	0.16	16.67	0.83	3.68	0.27	0.49	0.50
Lang. Testing in Asia	1.69	36.21	36.40	9.73	0.26	1.37	0.34	0.18	0.88	8.18	1.56	2.15	1.04
Prot. Control Mod. Power Syst.	1.29	27.49	35.04	5.86	1.95	9.48	0.21	6.72	1.16	6.63	1.06	2.16	0.96
Research in Eng. Design	3.91	30.93	42.26	5.43	1.00	5.67	0.58	0.69	0.75	5.85	0.81	1.73	0.39
Results in Eng.	2.48	25.39	29.02	6.15	7.22	18.04	0.10	1.84	1.36	5.52	1.00	1.02	0.86
Sage Open	0.48	25.38	51.48	4.46	0.96	7.13	0.36	0.30	0.85	4.89	0.87	2.10	0.75
Trans. Affect. Comput.	2.55	32.98	46.96	2.83	1.61	3.79	0.56	1.55	0.78	4.07	0.75	1.21	0.37
Trans. Cloud Comput.	2.78	26.75	42.96	4.36	0.75	8.28	0.60	1.81	0.98	6.73	0.94	2.48	0.56
Transactions on Computers	3.46	24.30	28.67	4.24	2.99	20.63	0.61	4.40	1.02	5.58	0.81	2.36	0.92
Trans. Mobile Comput.	2.72	24.43	28.54	3.35	3.97	22.53	0.42	4.20	0.85	5.58	0.71	2.24	0.45

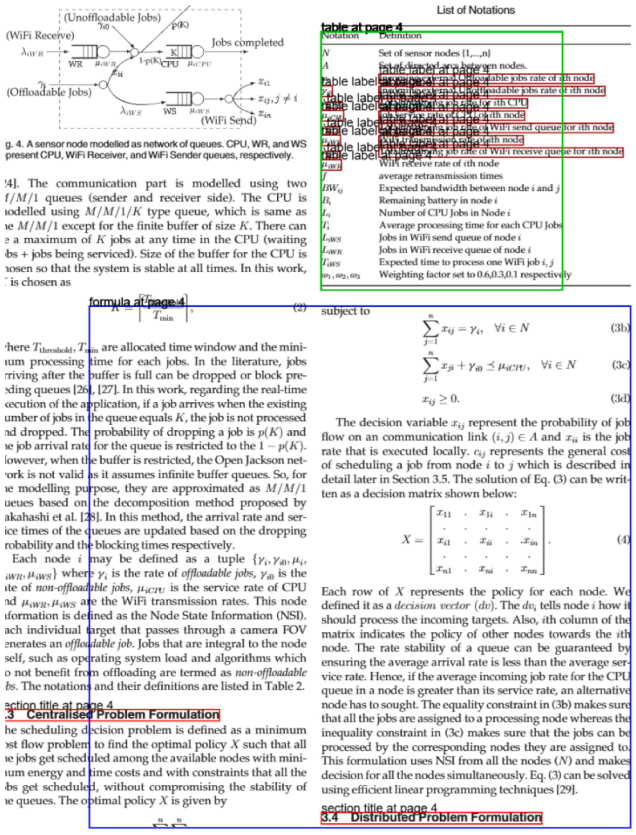


Fig. 6. Interpretation of the evaluation measures (best viewed in color).

Table 6
DocLap overlap error detection performance.

Measure	Value
Precision	0.78
Recall	0.57
Accuracy	0.49
F-measure	0.66

error detection capability; therefore, they must be interpreted in relation to the number of semantic overlaps detected by the system.

Tables 7 and 9 report the number of overlapping layout errors by journal and publisher. The results in these tables show the number of semantic overlaps compared to the number of actual recognition errors (i.e. 180 semantic overlaps are detected in the BRAIN AND BEHAVIOR journal documents involving the *Table label* class and 146 of these bounding boxes are errors).

The comparison of the values highlights that there is a high correlation between semantic overlaps and actual errors.

Furthermore, the distribution of layout classes among journals must be taken into account (i.e. overlap error values for class label *Math* have meaning only in a scientific context); these values are expressed as layout class frequencies among journals in Table 5.

As expected, the most error-prone layout classes are *Table* and *Table label*, due to their textual similarity and spatial proximity. For example, IEEE and WILEY show high error rates in *Table label* (129 and 180, respectively), in particular in some of their journals that are BRAIN AND BEHAVIOR and TRANSACTIONS ON COMPUTERS, which report 146 and 73 *Table label* errors, respectively. These overlaps are expected in dense layouts or biomedical content, where the disposition of captions over tables is not standard and more difficult to predict.

Conversely, regions such as *Acknowledgements*, *Figure*, and *Figure label* are more stable, with low error rates across publishers and journals. This is likely due to their isolated placement, often at section boundaries or end-of-document positions. Journals with a more structured layout, such as TRANSACTIONS ON AFFECTIVE COMPUTING and TRANSACTIONS ON CLOUD COMPUTING, are less affected by recognition errors, while visually dense sources such as WILEY journals show frequent layout conflicts, particularly in caption-heavy regions. Fig. 7 shows the most frequent layout classes in the detected semantic overlaps.

The evaluation script and the annotation files to assess the system performance are available at this link.

6.3. Question answering

In this section, we evaluate the question answering capabilities of DocLap. The goal is to assess how effectively DocLap extracts relevant information from documents to answer specific questions.

6.3.1. Methodology

The DocLap capability in answering questions over multi-page scholarly documents is evaluated using the multi-page MP-DocVQA dataset (Tito et al., 2023). MP-DocVQA is selected since it is one of the few publicly available benchmarks supporting question answering over multi-page documents, whereas most existing document VQA datasets are restricted to single-page settings (Cho et al., 2025). Although MP-DocVQA is not specifically designed for scholarly documents, it includes a substantial subset of scholarly articles and contains questions that explicitly target document layout elements that are directly addressed by our system, making it a suitable benchmark for our evaluation.

A well recognized limitation in this area is the scarcity of publicly available datasets providing question-answer annotations specifically

Table 7
Number of recognition errors wrt detected semantic overlaps by journal.

Journal	Ack.	Link	Ref.	Abst.	Fig.	Fig.label	Foot.	Math	Auth.	Section	Tab.	Tab.label	Title	Total
Advanced Science	0/0	0/0	0/0	0/0	1/2	0/4	2/2	3/3	0/0	4/4	4/7	7/10	0/0	21/32
Advances in Mech. Eng.	4/4	0/0	2/2	0/0	0/1	0/0	2/2	3/3	0/0	1/4	2/2	19/19	0/0	33/37
Ampersand	0/0	0/0	4/4	0/0	0/1	21/21	4/5	0/0	0/0	5/7	19/21	59/66	0/0	112/125
Asia Pac. Policy Stud.	0/0	0/0	1/1	0/0	1/1	0/0	2/4	0/0	0/0	0/1	5/8	27/30	0/0	35/45
Brain and Behavior	0/0	0/0	0/0	0/2	0/0	18/26	7/8	2/2	0/0	5/13	15/22	146/180	0/0	193/253
Int. J. Eng. Sci.	0/0	0/0	0/0	0/0	0/0	0/0	3/3	4/7	0/0	6/6	2/4	2/2	0/0	17/22
Lang. Testing in Asia	0/0	0/0	0/0	0/0	0/0	0/0	2/2	0/0	0/0	0/0	3/3	5/5	0/0	10/10
Prot. Control Mod. Power Syst.	2/2	0/0	4/5	0/0	0/0	0/0	1/1	5/5	0/2	0/16	7/8	23/30	0/0	42/69
Research in Eng. Design	0/0	0/0	5/5	0/0	0/0	0/0	5/5	2/2	0/0	14/14	13/15	8/14	0/0	47/55
Results in Eng.	0/0	0/0	3/3	0/0	1/1	0/0	2/2	1/1	0/0	1/1	3/4	3/5	0/0	14/17
Sage Open	0/0	0/0	2/2	0/0	0/0	0/0	3/3	0/0	0/0	0/2	3/7	19/20	0/0	27/34
Trans. Affect. Comput.	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	1/2	1/2	0/0	2/4
Trans. Cloud Comput.	0/0	0/0	0/0	0/0	0/0	0/0	1/1	1/1	0/0	0/0	3/3	3/8	0/0	8/13
Trans. Computers	0/0	0/0	0/0	0/0	0/0	0/5	5/5	3/4	0/0	6/7	6/7	73/74	0/0	93/102
Trans. Mobile Comput.	0/0	0/0	0/0	0/0	0/0	0/1	6/6	3/3	0/0	1/5	10/13	52/54	0/0	72/82

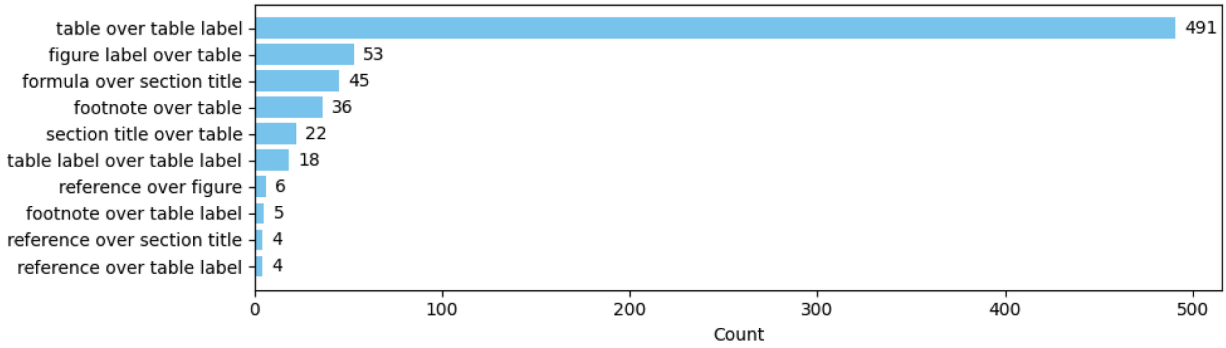


Fig. 7. Top 10 frequent semantic overlap errors in the DAD dataset documents.

Table 8
Occurrence of semantic overlaps ($IoU \geq 0.1$) over leading document datasets (excluding generic text boxes).

Dataset	Docs	Classes	# boxes	Overlaps (%)	Semantic overlaps (%)
DocBank	156k	13	6,72M	0.17	0.08
Publaynet	118k	5	3.26M	0.03	0.01
Doclaynet	3k	11	2.21M	53.92	2.03
HRDOC	1k	18	653k	26.31	4.44

for multi-page scholarly documents (Xia et al., 2024), which makes rigorous domain-specific evaluation challenging for any system operating in this setting. Using the scholarly subset of MP-DocVQA therefore represents a pragmatic compromise aligned with current practice in the field. An additional caveat introduced by this choice is that MP-DocVQA consists of scanned documents rather than digital-born PDFs. Since DocLap relies on document segmentation to extract the context provided to the LLM (Fig. 2), segmentation errors on scanned documents inevitably propagate to the downstream QA task, negatively impacting the performance of our system.

Given these premises, the main steps performed in this evaluation are as follows:

- Dataset filtering and integration
- Ground truth construction (segmentation, context extraction and QA)
- Assessment (segmentation, context extraction and QA).

Dataset filtering and integration Original MP-DocVQA dataset consists in 46K questions defined on 48K document images, which are limited for this evaluation to those containing one of the layout labels covered by the DocLap system, which are *title*, *author*, *abstract*, *caption*, *figure*, *table*, *formula*, *section*, *link*, *note*, *acknowledgements* and *reference*. It is to be noticed that these layout categories can be found only in the scholarly article domain. Original PDFs are extracted by hand and filtered by their compliance with the scholarly article domain. The resulting set is evaluated by means of the system's segmentation, context extraction and question answering capabilities; the evaluations are then correlated.

Ground truth construction

- The segmentation ground truth is obtained exploiting a custom-made labeling tool (available at this link) based on PyMuPDF[15] that allows annotating MP-DocVQA PDFs by drawing the bounding boxes directly on the PDF and storing the annotation in the usual evaluation format (Section 6.1.1).
- The ground truth for the question classification is built using the layout labels that determined the question filtered from the MP-DocVQA dataset. Each question is associated to a context class that is therefore already contained in the question.
- The ground truth for question answering is also obtained from the MP-DocVQA dataset. The filtered question-answer pairs were verified by human assessors and used as ground truth. The resulting data is then converted to the format: [*File_name*, *question*, *context*, *answer*]. The same is done for the answers provided by DocLap.

Assessment

- The DocLap annotations and the assessment of the segmentation capabilities for the PDFs of the QA evaluation are obtained in the same way as described in Section 6.1.
- The system's question classifications, which identify the context provided to the LLM, are also evaluated. The question classifications are compared with the layout labels that determined the question filtered from the MP-DocVQA dataset.

[15] https://github.com/pymupdf/pymupdf/

Table 9
Number of recognition errors wrt detected semantic overlaps by publisher.

Publisher	Ack.	Link	Ref.	Abst.	Fig.	Fig.label	Foot.	Math	Auth.	Section	Tab.	Tab.label	Title	Total
Elsevier	0/0	0/0	7/7	0/0	1/2	21/21	9/10	5/8	0/0	12/14	24/29	64/73	0/0	143/164
IEEE	0/0	0/0	0/0	0/0	0/0	0/6	12/12	7/8	0/0	7/12	20/25	129/138	0/0	175/201
SAGE	4/4	0/0	4/4	0/0	0/1	0/0	5/5	3/3	0/0	1/6	5/9	38/39	0/0	60/71
Springer	2/2	0/0	9/10	0/0	0/0	0/0	8/8	7/7	0/2	14/30	23/26	36/49	0/0	99/134
Wiley	0/0	0/0	1/1	0/2	2/3	18/30	11/14	5/5	0/0	9/18	24/37	180/220	0/0	249/330

Table 10
Answer quality evaluation with different engines.

Model	Precision	Recall	Accuracy	F-measure	ANLS	ANLS*n*
DocLap	0.54	0.94	0.51	0.68	0.14	0.25
NotebookLM	0.69	0.86	0.62	0.77	0.21	0.59
ChatPDF	0.48	**1.00**	0.48	0.65	0.18	0.41
Textract	**0.93**	**1.00**	**0.93**	**0.96**	**0.86**	**0.92**

- The question answering capabilities are evaluated comparing the QA pairs of DocLap with those of the ground truth, assessing the quality of the answer by considering the inclusion of the DocLap response terms within the MP-DocVQA answer.

  To provide direct comparison with established benchmarks, we include the evaluation of the standard Average Normalized Levenshtein Similarity (ANLS) (Biten et al., 2019), which is computed as:

$$\text{ANLS} = \frac{1}{N}\sum_{i=0}^{N}\left(\max_{j} s(a_{ij}, o_{q_i})\right)$$

$$s(a_{ij}, o_{q_i}) = \begin{cases} (1 - NL(a_{ij}, o_{q_i})) & \text{if } NL(a_{ij}, o_{q_i}) < \tau \\ 0 & \text{if } NL(a_{ij}, o_{q_i}) \geqslant \tau \end{cases}$$

  where N is the total number of questions in the dataset, M is the total number of GT answers per question, a_{ij} are the ground truth answers where $i = \{0, \ldots, N\}$ and $j = \{0, \ldots, M\}$, and o_{q_i} is the system's answer for the i^{th} question q_i. $NL(a_{ij}, o_{q_i})$ is the normalized Levenshtein distance between the strings a_{ij} and o_{q_i} with threshold $\tau = 0.8$ to mitigate the penalization of verbose answers that actually contain the answer. Since ANLS is optimized for short extractive responses, it may downweight long generative responses even if containing the correct answer. In our experiments, LLMs frequently produce natural language answers where the expected information is wrapped inside longer sentences. To better reflect answer correctness in these cases, as done for other measures, we also considered a containment-based matching criterion (ANLS*n*), assigning a true positive when the ground truth answer appears as a substring of the generated response. The results are included in Table 10.

6.3.2. Results

The DocLap document question answering capabilities with respect to state of the art technologies are reported in Table 10. This table presents a comparative analysis of PDF chatbots, including commercial solutions such as Amazon Textract, Google NotebookLM, and ChatPDF. All considered platforms provide document segmentation, navigation, and question answering capabilities.

Results are heavily dependent on the question classification and segmentation performance over the documents under evaluation, which are lower than those observed for digital-born documents (Table 3). It is to be noticed that Amazon Textract requires the specification of the page to analyze for processing the question, which significantly impacts the quality of the answer. Also, Textract OCR capabilities mitigate the shortcomings of the question answering module. Moreover, DocLap is the only open source system among the considered solutions.

In Table 11 are reported the task-specific performances of DocLap contributing to the question answering results.

Table 11
DocLap task-specific performance on MP-DocVQA (match: $IoU \geq 0.5$).

	Segmentation	Classification	QA
Precision	0.43	0.76	0.54
Recall	0.60	1.00	0.94
Accuracy	0.34	0.76	0.51
F-measure	0.50	0.86	0.68

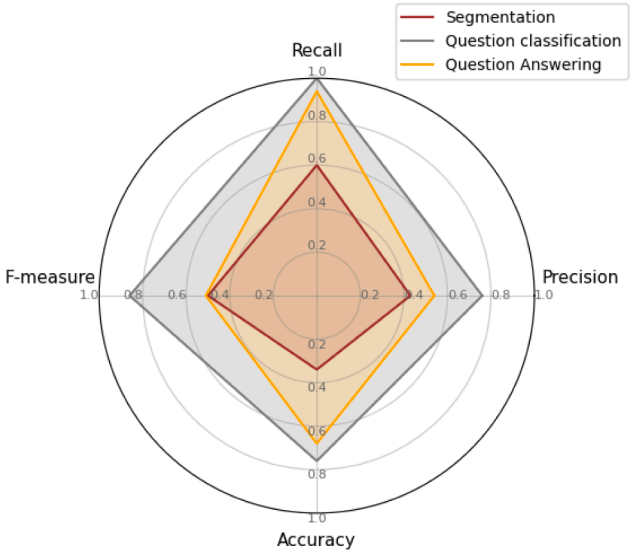


Fig. 8. Correlation between segmentation, classification and QA performance on the MP-DocVQA dataset.

Fig. 8 shows the correlation between segmentation, question classification and QA capabilities for the selected MP-DocVQA documents, highlighting the impact of segmentation and classification on the measures registered for the question answering task.

In this setting, DocLap underperforms on the question answering task primarily due to the impact of document segmentation on the overall pipeline. Furthermore, MP-DocVQA consists of scanned documents rather than digital-born PDFs, and many of its documents are old, exhibiting non-standard layouts. Since segmentation performance is known to degrade under these conditions, errors introduced at this stage propagate to the downstream QA component, ultimately lowering the reported performance. However, MP-DocVQA is one of the few publicly available benchmarks supporting multi-page document reasoning and also includes a scholarly subset with layout-oriented questions, making it a suitable evaluation setting for our system. These factors should be taken into account when interpreting the results.

To support the reliability of our QA evaluation we identified the LongDocURL dataset (Deng et al., 2025) as another likely suitable resource. LongDocURL includes long documents that partially resemble scholarly articles, such as theses among other document types, and also provides the original PDFs which can be directly processed by our system. Performing experiments on LongDocURL, we followed the same protocol adopted for MP-DocVQA, and conducted a QA evaluation on a

Table 12
Document segmentation and error correction performance across different document lengths and configurations.

Length (pages)	Segm. time (s) *@test*	Segm. time (s) *@prod*	Segm. time (s) *@cloud*	Sem. overl. (avg)	Error fix time (s) (per-error) *@test*	Error fix time (s) (per-error) *@prod*
5–10	52.5 ± 0.5	16.56 ± 0.5	22.03 ± 0.5	0.5	16.12	5.28 ± 0.5
10–15	102.6 ± 0.5	41.70 ± 0.5	33.34 ± 0.5	0.5	19.75	5.78 ± 0.5
15–20	84 ± 0.5	37.51 ± 0.5	37.30 ± 0.5	0.3	20.54	15.24 ± 0.5

Table 13
Question answering performance across different document lengths and configurations.

Length (pages)	QA latency (s) *@test*	QA latency (s) *@prod*	QA latency (s) *@cloud*
5–10	90 ± 0.5	20.64 ± 0.5	19.77 ± 0.5
10–15	187 ± 0.5	30.43 ± 0.5	38.65 ± 0.5
15–20	92 ± 0.5	22.84 ± 0.5	24.40 ± 0.5

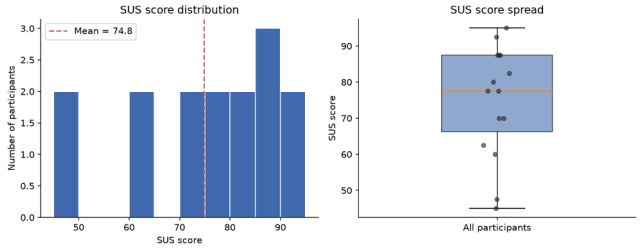


Fig. 9. Distribution of SUS scores across participants (histogram and boxplot).

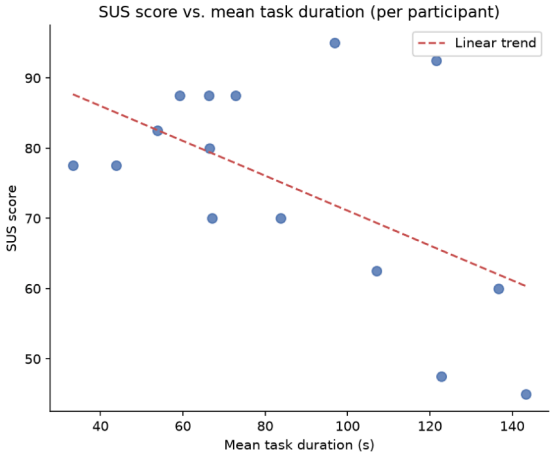


Fig. 10. SUS score vs. mean task duration per participant, with a linear trend ($r = -0.51$).

representative sample to assess the system's behavior on this dataset as well. The results showed no noticeable differences in performance with respect to MP-DocVQA, demonstrating the robustness of the approach beyond a single benchmark. Additional experiments carried out with different LLMs like Llama3 and DeepSeek-R1 show performance below 20% and are not reported for this evaluation, while being available as supplementary data.

7. Experimental setup, time efficiency and usability analysis

The response time of the system mostly depends on the number of pages (images) on which the relevant information is presumed to be and on the number of semantic overlaps which are detected and corrected. The time required may vary between seconds and minutes depending on local installation resources. All our experiments are conducted on a workstation equipped with Intel Core i7-10750H CPU@2.60GHz, 16 GB of RAM, and NVIDIA GeForce GTX 1650 with Max-Q Design (laptop) GPU with 4 GB of VRAM. Additional evaluations to assess the system's performance on a production environment are performed on Intel Core Ultra 9 275HX CPU, 32 GB of RAM, and NVIDIA GeForce GTX 5080 (laptop) GPU with 16 GB of VRAM.

Evaluations of time efficiency for segmentation and question answering across different models and document lengths are reported in Tables 12 and 13. The local model for question answering is `llava:latest` (Liu et al., 2024) that is currently v1.6; the cloud model is `qwen3-vl:235b-instruct-cloud` (Bai et al., 2023), both leveraged through the Ollama API v0.13.4[16,17]. All the experiments are conducted under cold start conditions.

These experiments use the same dataset employed in the QA evaluation (Section 6.3.1), which consists of documents uniformly sampled across different document length ranges. The reported values correspond to average timings computed over the entire dataset. As expected, segmentation latency exhibits an approximately linear growth with respect to the average number of pages and to the number of semantic overlaps to correct. These two factors are correlated, as longer documents are more likely to contain a higher number of errors. The variable presence of semantic overlaps is the factor that impacts the most on overall efficiency, since DocLap performs automatic correction of these errors and each correction is carried out by an LLM call. It is to be noticed that in the considered documents the short and medium length categories contain, on average, more errors per document than long documents (Table 12). This explains why, for the local model, medium length documents can take longer to process than longer documents. In contrast, the cloud model remains efficient across all document lengths as each LLM request for corrections is completed in a few seconds compared to the several minutes required by the local configuration. These measurements include the time required by the system to render annotation layers on the original PDFs in the interface.

Time complexity analysis for question answering shows that latency generally increases from short (5–10 pages) to medium length documents (10–15 pages), not scaling linearly for long documents (15–20 pages). This behavior occurs because the time required for answer generation is driven more by the number of content elements relevant to the query than by the total number of pages and because both local and cloud models exploit internal optimizations such as caching, that reduce the impact of less relevant pages.

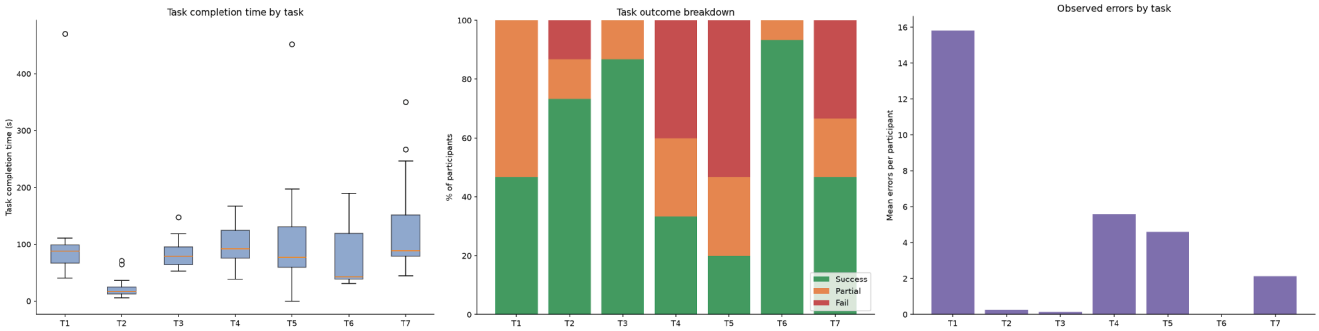
For this evaluation, the computational complexity of question classification must also be taken into account. Question classification is handled by the `gpt-oss:120b-cloud` model which processes only the user questions, that are typically short textual inputs; as a result, its response time remains consistently low (1–2 seconds) and independent of document length, relatively contributing to overall latency. In contrast, the latency of the response generation is strongly influenced by both the document length and the system configuration.

The local models show an evident increase in time complexity, that becomes high as document length and number of overlaps grows, indicating limited scalability when reasoning over extended multi-page contexts. Conversely, the cloud models maintain lower response times, particularly benefiting from scenarios presenting more semantic overlaps, which are resolved more efficiently.

[16] https://ollama.com/library/llava
[17] https://ollama.com/library/qwen3-vl:235b-instruct-cloud

Table 14
Per-task completion rate, timing *@test*, and error statistics ($N = 15$ participants per task).

Task	Description	Success	Partial	Fail	Mean dur. (s)	Median dur. (s)	Mean errors
T1	Process the document	47%	53%	0%	106.1	88.1	15.80
T2	Locate a specific element	73%	13%	13%	24.0	17.1	0.27
T3	Ask a contextual question	87%	13%	0%	83.4	79.1	0.13
T4	Ask a question on layout elements	33%	27%	40%	97.3	91.7	5.60
T5	Ask a question on layout elements (filtered context)	20%	27%	53%	111.2	77.1	4.60
T6	Find an overlap violation	93%	7%	0%	78.8	43.0	0.00
T7	Process a LaTeX document	47%	20%	33%	131.7	88.3	2.13



(a) Completion time by task. (b) Outcome breakdown by task. (c) Mean errors by task.

Fig. 11. Task-level results: completion time, outcome distribution, and observed errors for each of the seven tasks T1-T7.

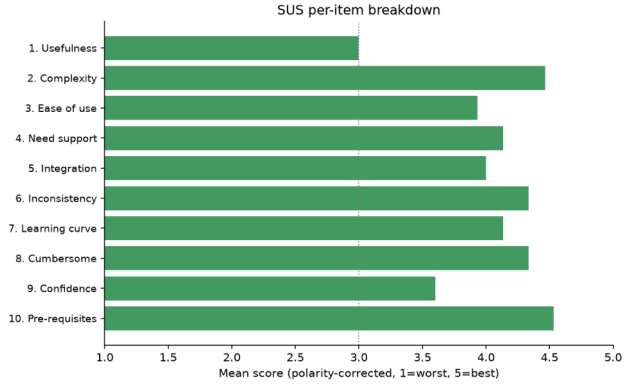


Fig. 12. Per-item SUS means with polarity correction (higher values = better usability).

The latency observed in the test deployment is to be interpreted in the context of an experimental setup. To assess the impact of hardware resources on the system's efficiency, the experiments that are reported in Tables 12 and 13 include a production configuration capable of supporting high-concurrency workloads. In this setting, results show a significant reduction in execution times, with performance comparable to that obtained using cloud-based models. These experiments show that worst-case latencies are largely influenced by the experimental hardware configuration and may benefit from both production-level hardware resources and code-level optimization.

The evaluation script and the annotation files to assess each engine's performance are available at this link.

Usability analysis. We conducted a usability evaluation on the local test installation with $N = 15$ participants drawn from different backgrounds and not previously aware of the system's functionalities.

During the session, participants self-reported their familiarity with PDF manipulation tools and with LLM-based tools on a 5 point scale, allowing us to relate previous experience to observed outcomes. Each participant completed the same set of seven tasks T1-T7 (Table 14) on PDFs randomly chosen from the DAD dataset, covering document processing, element location, contextual question answering, layout-related queries (with and without filtered context), detection of overlap violations, and processing of LaTeX documents.

For each task, we logged the completion time (local processing), the number of errors observed, and a categorical outcome (*success*, *partial*, or *fail*). After task completion, participants completed the System Usability Scale (SUS) (Lewis, 2018), a standard 10-item, 5 point Likert questionnaire, from which a 0–100 score per participant is derived using a polarity-corrected scoring procedure (odd items scored as response−1, even items as 5−response, summed and multiplied by 2.5). We report descriptive statistics for the SUS score (mean, standard deviation, median, range), map the sample mean onto the Sauro-Lewis curved-grading scale norms (Sauro & Lewis, 2016) to obtain a qualitative grade, and assess the internal consistency of the 10 items via Cronbach's α. At the task level we report, for each task, the proportion of success/partial/fail outcomes, mean and median completion time, and mean error count (Fig. 11). Finally, we compute pairwise Pearson correlations between background-familiarity variables and the outcome measures (SUS score, mean task duration, total errors).

The sample obtained a mean SUS score of 74.8 ($SD = 15.5$, median = 77.5, range $[45.0, 95.0]$), corresponding to a letter grade of B on the Sauro-Lewis curved grading scale, placing the system at the ∼70th percentile (good usability) (Fig. 9). Internal consistency is slightly above the conventional acceptability threshold ($\alpha = 0.85 > 0.70$), supporting the reliability of the per-item breakdown (Fig. 12); the lowest rated item concerns perceived usefulness ($M = 3.00$), while complexity, cumbersomeness, and pre-requisites are rated most favorably ($M = 4.47$, $M = 4.33$, $M = 4.53$ respectively). At the task level (Table 14), success rates ranged from 20% (T5, layout-element questions with filtered context) to 93% (T6, overlap violation detection), with T4 and T5 showing the highest fail rates (40% and 53% respectively) and, together with T1, the most frequent errors (in the case of T1, segmentation errors). In the local test installation setting, mean completion times vary substantially across tasks (24 − 132 s), with T7 and T1 the slowest and T2 the fastest on average. In a production setting, we expect the system's efficiency to scale in line with the results reported in Tables 12

and 13. Across participants, SUS score correlates negatively with mean task duration ($r = -0.55$) and positively, though weakly, with total errors ($r = 0.21$), while duration and error count are negatively associated ($r = -0.39$) (Fig. 10), suggesting that participants who completed the tasks faster tend to perceive the system as more usable. Self-reported LLM-tool familiarity shows a moderate negative association with total errors ($r = -0.36$), while PDF-tool familiarity shows only weak associations with all outcomes (all $|r| \leq 0.16$), indicating that prior experience with LLM-based tools, more than with PDF tools, may partially reduce error frequency in this sample.

The usability test scripts and the resulting data are available at this link.

8. Conclusions and future work

Conclusions

This paper extends scholarly document understanding and document question answering by integrating visual, semantic, and contextual information into a unified, modular, and open source framework that supports multi-layer exploration through interactive visualization and LLM-based interaction. Linking semantic information to documents is challenging from a research perspective: most recent solutions exhibit limited domain awareness, considering only basic relations between text chunks. Our approach addresses several limitations identified in the current literature, including interoperability across document representations, support for layout validation, multi-page capabilities, and long context issues when performing multi-page question answering. Interoperability is achieved by integrating heterogeneous representations such as PDF, LaTeX, XML, and ontologies, enabling complementary navigation and analysis strategies. This integration enables mutual improvement among components, aiding both system performance and user experience. Segmentation remains a pivotal component, providing the geometric information required for document representation and answer generation. In particular we leverage the Document Components ontology, that is rich in layout variability, focusing on semantic relations to detect visual overlap errors and paving the way for more sophisticated layout coherence analysis. The proposed approach is based on the use of disjointness relations that may exist between overlapping regions. This relation is analyzed and interpreted as an indicator of potential recognition errors, providing a systematic way to identify and address inconsistencies in detected layout structures.

By incorporating semantic validation, our pipeline improves segmentation quality, providing more reliable input to the question answering module, and supporting the generation of high-quality datasets. Unlike purely generative conversational systems, whose quality is largely dependent on training data, our approach bases its validation mechanism on a formal knowledge model. Being independent of data, this model provides an additional and less biased layer of verification.

While general purpose systems such as ChatGPT, NotebookLM, and ChatPDF can perform effective question answering over multi-page PDFs, they do not explicitly validate the consistency of the document representation that they build internally. By introducing an ontology-based layer that enables the system to reason about the structural coherence of document components, our work enables explainable semantic validation and interoperability of document representations. This is particularly relevant in scholarly document processing pipelines, where downstream tasks such as question answering rely on document representations that are learned from highly variable layout structures. In such settings, a formal semantic model provides explicit constraints on valid document interpretations, reducing the propagation of recognition errors and making downstream applications more robust. Moreover, the ontology provides a representation that can be extended with inference mechanisms, supporting more refined reasoning.

We exploit LLMs both for context length optimization and to enhance the accessibility of diverse information which is not directly available from data. This enables navigation of different representations from an integrated interface and allows us to study the role of structured context in LLM-based QA. Moreover, in our setting, LLMs are employed only for the narrowest possible tasks (classification and correction), while separable computations are delegated to other components using RAG, reducing ambiguity in the associated tasks.

Future work

Currently, DocLap allows exploration of visual data that is extracted through the document segmentation module and the LaTeX processing module. As future work, having a tool capable of effectively associating different representations of document layout elements with precise coordinates enables the automatic construction of class-specific datasets (e.g. formula and chemical structure datasets). Moreover, the custom-made user interface described in Section 4 allows flexible adjustment of the underlying engines to enhance both recognition capabilities and question answering performance.

An important direction for future research emerging from this work is to address the lack of multi-page scholarly document datasets annotated with question-answer pairs. Such resources are currently scarce, and their absence limits the systematic evaluation of document-level QA systems that rely on semantic segmentation and structural reasoning.

With regard to the proposed semantic-based enhancements, the variety of ontology relation employments can be extended to identify more than just overlap errors. To this end, parent relations can be exploited to detect misclassifications and elements missing paired classes (i.e. *figures* and *captions*, *bibliography elements* and *section*). In addition, the presence of an ontology layer enables the possibility of extending the present structure with broader context ontologies (Hendricks et al., 2020) and also exploiting reasoning capabilities to expand actual relations with inference.

The DocLap overlap detection capabilities can also be easily exploited to compute and enhance the most widely used datasets for layout analysis like DocBank and PubLayNet (Massai & Marinai, 2026) which, as shown in Table 8, are not immune to such issues. Moreover, question answering performance would increase both by improving the segmentation capabilities as described above and by employing language models with more parameters. Since the proposed approach supports and optimizes the processes handled by LLMs for question answering, overall system performance is expected to enhance significantly as models advance and computational capabilities improve.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Software, evaluation data and code are made available online and links are provided in the article.

References

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S. et al. (2023). GPT-4 technical report. arXiv preprint arXiv:2303.08774.

Amazon Web Services, I. (2025). Amazon textract. https://aws.amazon.com/textract/.

Ammar, W., Groeneveld, D., Bhagavatula, C., Beltagy, I., Crawford, M., Downey, D., Dunkelberger, J., Elgohary, A., Feldman, S., Ha, V. et al. (2018). Construction of the literature graph in semantic scholar. arXiv preprint arXiv:1805.02262.

Arif Demirtaş, M., Oral, B., Yasin Akpınar, M., & Deniz, O. (2022). Semantic parsing of interpage relations. In *2022 26th International conference on pattern recognition (ICPR)* (pp. 1579–1585). https://doi.org/10.1109/ICPR56361.2022.9956546

Attwood, T. K., Kell, D. B., McDermott, P., Marsh, J., Pettifer, S. R., & Thorne, D. (2010). Utopia documents: Linking scholarly literature with research data. *Bioinformatics, 26* (18), i568–i574.

Bai, J., Bai, S., Yang, S., Wang, S., Tan, S., Wang, P., Lin, J., Zhou, C., & Zhou, J. (2023). Qwen-VL: A versatile vision-language model for understanding, localization, text reading, and beyond. arXiv preprint arXiv:2308.12966.

Barboule, C., Piwowarski, B., & Chabot, Y. (2025). Survey on question answering over visually rich documents: Methods, challenges, and trends. arXiv preprint arXiv:2501.02235.

Bast, H., & Korzen, C. (2017). A benchmark and evaluation for text extraction from PDF. In *2017 ACM/IEEE Joint conference on digital libraries (JCDL)* (pp. 1–10). IEEE.

Bedini, I., Matheus, C., Patel-Schneider, P. F., Boran, A., & Nguyen, B. (2011). Transforming XML schema to OWL using patterns. In *2011 IEEE Fifth international conference on semantic computing* (pp. 102–109). IEEE.

Biten, A. F., Tito, R., Mafla, A., Gomez, L., Rusiñol, M., Valveny, E., Jawahar, C. V., & Karatzas, D. (2019). Scene text visual question answering. https://arxiv.org/abs/1905.13648.

Blau, T., Fogel, S., Ronen, R., Golts, A., Ganz, R., Ben Avraham, E., Aberdam, A., Tsiper, S., & Litman, R. (2024). Gram: Global reasoning for multi-page vqa. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition* (pp. 15598–15607).

Bozsolik, T., Tricot, J., Rishi, D., Maltzan, B., & Brinn, S. (2024). arxiv dataset. https://doi.org/10.34740/KAGGLE/DSV/7548853

Chia, Y. K., Cheng, L., Chan, H. P., Song, M., Liu, C., Aljunied, M., Poria, S., & Bing, L. (2025). M-LongDoc: A benchmark for multimodal super-long document understanding and a retrieval-aware tuning framework. In C. Christodoulopoulos, T. Chakraborty, C. Rose, & V. Peng (Eds.), *Proceedings of the 2025 Conference on empirical methods in natural language processing* (pp. 9244–9261). Suzhou, China: Association for Computational Linguistics. https://doi.org/10.18653/v1/2025.emnlp-main.469

Cho, J., Mahata, D., Irsoy, O., He, Y., & Bansal, M. (2025). M3docVQA: Multi-modal multi-page multi-document understanding. In *Proceedings of the IEEE/CVF International conference on computer vision (ICCV) workshops* (pp. 6237–6247).

Ciotti, F. (2018). A formal ontology for the text encoding initiative. *Umanistica Digitale*, *2* (3). https://doi.org/10.6092/issn.2532-8816/8174

Constantin, A., Peroni, S., Pettifer, S., Shotton, D. M., & Vitali, F. (2016). The document components ontology (doCO). *Semantic Web*, *7* (2), 167–181. https://doi.org/10.3233/SW-150177

Constantin, A., Pettifer, S., & Voronkov, A. (2013). PDFX: Fully-automated pdf-to-xml conversion of scientific literature. In S. Marinai, & K. Marriott (Eds.), *ACM Symposium on document engineering 2013, Doceng '13, Florence, Italy, September 10–13, 2013* (pp. 177–180). ACM. https://doi.org/10.1145/2494266.2494271

CrossRef (2024). Resource available online. https://www.crossref.org/documentation/retrieve-metadata/rest-api/.

Cui, C., Sun, T., Liang, S., Gao, T., Zhang, Z., Liu, J., Wang, X., Zhou, C., Liu, H., Lin, M., Zhang, Y., Zhang, Y., Zheng, H., Zhang, J., Zhang, J., Liu, Y., Yu, D., & Ma, Y. (2025a). PaddleOCR-VL: Boosting multilingual document parsing via a 0.9b ultra-compact vision-language model. https://arxiv.org/abs/2510.14528.

Cui, C., Sun, T., Lin, M., Gao, T., Zhang, Y., Liu, J., Wang, X., Zhang, Z., Zhou, C., Liu, H., Zhang, Y., Lv, W., Huang, K., Zhang, Y., Zhang, J., Zhang, J., Liu, Y., Yu, D., & Ma, Y. (2025b). PaddleOCR 3.0 technical report. https://arxiv.org/abs/2507.05595.

Deng, C., Yuan, J., Bu, P., Wang, P., Li, Z.-Z., Xu, J., Li, X.-H., Gao, Y., Song, J., Zheng, B., & Liu, C.-L. (2025). LongDocURL: A comprehensive multimodal long document benchmark integrating understanding, reasoning, and locating. In W. Che, J. Nabende, E. Shutova, & M. T. Pilehvar (Eds.), *Proceedings of the 63rd Annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 1135–1159). Vienna, Austria: Association for Computational Linguistics. https://doi.org/10.18653/v1/2025.acl-long.57

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the north american chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)* (pp. 4171–4186).

Di Iorio, A., Peroni, S., Poggi, F., Vitali, F., & Shotton, D. M. (2013). Recognising document components in XML-based academic articles. In S. Marinai, & K. Marriott (Eds.), *ACM Symposium on document engineering 2013, Doceng '13, Florence, Italy, September 10–13, 2013* (pp. 181–184). ACM. https://doi.org/10.1145/2494266.2494319

Ding, Y., Han, S. C., Lee, J., & Hovy, E. (2024). Deep learning based visually rich document content understanding: A survey. arXiv preprint arXiv:2408.01287.

Duan, Y., Chen, Z., Hu, Y., Wang, W., Ye, S., Shi, B., Lu, L., Hou, Q., Lu, T., Li, H., Dai, J., & Wang, W. (2025). Docopilot: Improving multimodal models for document-level understanding. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition (CVPR)* (pp. 4026–4037).

Eide, Ø. (2014). Ontologies, data modeling, and TEI. *Journal of the Text Encoding Initiative*, (8).

Formal, T., Lassance, C., Piwowarski, B., & Clinchant, S. (2024). Towards effective and efficient sparse neural information retrieval. *ACM Transactions on Information Systems*, *42* (5). https://doi.org/10.1145/3634912

Gan, Z., Li, L., Li, C., Wang, L., Liu, Z., Gao, J. et al. (2022). Vision-language pre-training: Basics, recent advances, and future trends. *Foundations and Trends® in Computer Graphics and Vision*, *14* (3–4), 163–352.

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., & Wang, H. (2023). Retrieval-augmented generation for large language models: A survey. arXiv preprint arXiv:2312.10997.

Gemelli, A., Vivoli, E., & Marinai, S. (2022). Graph neural networks and representation embedding for table extraction in PDF documents. In *2022 26th International conference on pattern recognition (ICPR)* (pp. 1719–1726). https://doi.org/10.1109/ICPR56361.2022.9956590

Ginev, D., & Miller, B. R. (2013). LateXML 2012-a year of lateXML. In *Intelligent computer mathematics: MKM, calculemus, DML, and systems and projects 2013, held as part of CICM 2013, bath, UK, July 8–12, 2013. proceedings 6* (pp. 335–338). Springer.

Ginev, D., Miller, B. R., & Oprea, S. (2014). E-books and graphics with LATExml. In *Intelligent computer mathematics: CICM 2014 Joint Events: Calculemus, DML, MKM, and systems and projects 2014, Coimbra, Portugal, July 7–11, 2014. Proceedings, vol. 8543*, pp. 427.

Ginev, D., Stamerjohanns, H., Miller, B. R., & Kohlhase, M. (2011). The LATEXML daemon: Editable math on the collaborative web. In *Intelligent computer mathematics: 18th Symposium, calculemus 2011, and 10th international conference, MKM 2011, Bertinoro, Italy, July 18–23, 2011. Proceedings 4* (pp. 292–294). Springer.

Grimm, J. (2008). Tralics, a LaTeX to XML translator; Part I. Ph.D. thesis. Inria.

Han, S., Xia, P., Zhang, R., Sun, T., Li, Y., Zhu, H., & Yao, H. (2025). Mdocagent: A multi-modal multi-agent framework for document understanding. https://arxiv.org/abs/2503.13964.

He, C., Li, W., Jin, Z., Xu, C., Wang, B., & Lin, D. (2024). Opendatalab: Empowering general artificial intelligence with open datasets. arXiv preprint arXiv:2407.13773.

Hendricks, G., Tkaczyk, D., Lin, J., & Feeney, P. (2020). CrossRef: The sustainable source of community-owned scholarly metadata. *Quantitative Science Studies*, *1* (1), 414–427.

Huang, P., Kashyap, A. R., Qin, Y., Yang, Y., & Kan, M. (2022a). Lightweight contextual logical structure recovery. In A. Cohan, G. Feigenblat, D. Freitag, T. Ghosal, D. Hermannova, P. Knoth, K. Lo, P. Mayr, M. Shmueli-Scheuer, A. de Waard, & L. L. Wang (Eds.), *Proceedings of the third workshop on scholarly document processing, SDP@COLING 2022, Gyeongju, Republic of Korea, October 12, - 17, 2022* (pp. 37–48). Association for Computational Linguistics. https://aclanthology.org/2022.sdp-1.5.

Huang, Y., Lv, T., Cui, L., Lu, Y., & Wei, F. (2022b). LayoutLMV3: Pre-training for document AI with unified text and image masking. In J. Magalhães, A. D. Bimbo, S. Satoh, N. Sebe, X. Alameda-Pineda, Q. Jin, V. Oria, & L. Toni (Eds.), *MM '22: The 30th ACM International conference on multimedia, Lisboa, Portugal, October 10, - 14, 2022* (pp. 4083–4091). ACM. https://doi.org/10.1145/3503161.3548112

Kamath, A., Singh, M., LeCun, Y., Synnaeve, G., Misra, I., & Carion, N. (2021). Mdetr-modulated detection for end-to-end multi-modal understanding. In *Proceedings of the IEEE/CVF International conference on computer vision* (pp. 1780–1790).

Kashyap, A. R., & Kan, M. (2020). SciWING- a software toolkit for scientific document processing. In M. K. Chandrasekaran, A. de Waard, G. Feigenblat, D. Freitag, T. Ghosal, E. H. Hovy, P. Knoth, D. Konopnicki, P. Mayr, R. M. Patton, & M. Shmueli-Scheuer (Eds.), *Proceedings of the first workshop on scholarly document processing, SDP@EMNLP 2020, online, November 19, 2020* (pp. 113–120). Association for Computational Linguistics. https://doi.org/10.18653/V1/2020.SDP-1.13

Kershaw, D., & Koeling, R. (2020). Elsevier oa cc-by corpus. arXiv preprint arXiv:2008.00774.

Koreeda, Y., & Manning, C. D. (2021). Capturing logical structure of visually structured documents with multimodal transition parser. In N. Aletras, I. Androutsopoulos, L. Barrett, C. Goanta, & D. Preotiuc-Pietro (Eds.), *Proceedings of the natural legal language processing workshop 2021, NLLP@EMNLP 2021, Punta Cana, Dominican Republic, November 10, 2021* (pp. 144–154). Association for Computational Linguistics. https://aclanthology.org/2021.nllp-1.15.

LabelImg (2022). Resource available online. https://www.github.com/tzutalin/labelImg.

Lála, J., O'Donoghue, O., Shtedritski, A., Cox, S., Rodriques, S. G., & White, A. D. (2023). PaperQA: Retrieval-augmented generative agent for scientific research. arXiv preprint arXiv:2312.07559.

Lewis, J. R. (2018). The system usability scale: Past, present, and future. *International Journal of Human–Computer Interaction*, *34* (7), 577–590.

Li, H., Su, Y., Cai, D., Wang, Y., & Liu, L. (2022). A survey on retrieval-augmented text generation. arXiv preprint arXiv:2202.01110.

Li, M., Xu, Y., Cui, L., Huang, S., Wei, F., Li, Z., & Zhou, M. (2020). DocBank: A benchmark dataset for document layout analysis. arXiv preprint arXiv:2006.01038.

Liu, C., Chen, H., Cai, Y., Wu, H., Ye, Q., Yang, M.-H., & Wang, Y. (2025). Structured attention matters to multimodal LLMs in document understanding. https://arxiv.org/abs/2506.21600.

Liu, H., Li, C., Li, Y., & Lee, Y. J. (2024). Improved baselines with visual instruction tuning. https://arxiv.org/abs/2310.03744.

Liu, H., Li, C., Wu, Q., & Lee, Y. J. (2023). Visual instruction tuning. https://arxiv.org/abs/2304.08485.

Lo, K., Wang, L. L., Neumann, M., Kinney, R., & Weld, D. (2020). S2ORC: The semantic scholar open research corpus. In *Proceedings of the 58th Annual meeting of the association for computational linguistics* (pp. 4969–4983). Online: Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.447

Lopez, P. (2009). Grobid: Combining automatic bibliographic data recognition and term extraction for scholarship publications. In M. Agosti, J. Borbinha, S. Kapidakis, C. Papatheodorou, & G. Tsakonas (Eds.), *Research and advanced technology for digital libraries* (pp. 473–474). Berlin, Heidelberg: Springer Berlin Heidelberg.

Ma, J., Du, J., Hu, P., Zhang, Z., Zhang, J., Zhu, H., & Liu, C. (2023). Hrdoc: Dataset and baseline method toward hierarchical reconstruction of document structures. In B. Williams, Y. Chen, & J. Neville (Eds.), *Thirty-seventh AAAI conference on artificial intelligence, AAAI 2023, thirty-fifth conference on innovative applications of artificial intelligence, IAAI 2023, thirteenth symposium on educational advances in artificial intelligence, EAAI 2023, Washington, DC, USA, February 7–14, 2023* (pp. 1870–1877). AAAI Press. https://doi.org/10.1609/AAAI.V37I2.25277

MacFarlane, J. (2012–2025). _pandoc user's guide_.

Markewich, L., Zhang, H., Xing, Y., Lambert-Shirzad, N., Jiang, Z., Lee, R. K.-W., Li, Z., & Ko, S.-B. (2022). Segmentation for document layout analysis: Not dead yet. *International Journal on Document Analysis and Recognition (IJDAR)*. https://doi.org/10.1007/s10032-021-00391-3

Massai, L. (2024). Evaluation of semantic relations impact in query expansion-based retrieval systems. *Knowledge Based Systems*, *283*, 111183. https://doi.org/10.1016/J.KNOSYS.2023.111183

Massai, L., & Marinai, S. (2026). Improving large-scale DLA datasets through semantic validation and relation-aware multimodal LLMs. In *Proceedings of the 22nd Conference*

on information and research science connecting to digital and library science (IRCDL 2026) CEUR Workshop Proceedings. Modena, Italy. [in press] https://ircdl2026.unimore.it/.

Massai, L., Nesi, P., & Pantaleo, G. (2019). PAVAL: A location-aware virtual personal assistant for retrieving geolocated points of interest and location-based services. *Engineering Applications of Artificial Intelligence*, *77*, 70–85. https://doi.org/10.1016/j.engappai.2018.09.013

Minesh, M., Rubèn, T., Dimosthenis, K., Manmatha, R., & Jawahar, C. V. (2020). Document visual question answering challenge 2020. *CoRR, abs/2008.08899*. https://arxiv.org/abs/2008.08899.

Mishra, D. S., Agarwal, A., Swathi, B. P., & Akshay, K. C. (2022). Natural language query formalization to SPARQL for querying knowledge bases using rasa. *Progress in Artificial Intelligence*, *11* (3), 193–206.

Müller, D. (2023). An HTML/CSS schema for TEX primitives–generating high-quality responsive. *TUGboat*, *44* (2), 275–286. https://kwarc.info/people/dmueller/pubs/tug23.pdf.

Napolitano, D., Vaiani, L., & Cagliero, L. (2024). On leveraging multi-page element relations in visually-rich documents. In *2024 IEEE 48th Annual computers, software, and applications conference (COMPSAC)* (pp. 360–365). IEEE.

Neumann, M., Shen, Z., & Skjonsberg, S. (2021). Pawls: Pdf annotation with labels and structure. arXiv preprint arXiv:2101.10281.

Niu, J., Liu, Z., Gu, Z., Wang, B., Ouyang, L., Zhao, Z., Chu, T., He, T., Wu, F., Zhang, Q., Jin, Z., Liang, G., Zhang, R., Zhang, W., Qu, Y., Ren, Z., Sun, Y., Zheng, Y., Ma, D., Tang, Z., Niu, B., Miao, Z., Dong, H., Qian, S., Zhang, J., Chen, J., Wang, F., Zhao, X., Wei, L., Li, W., Wang, S., Xu, R., Cao, Y., Chen, L., Wu, Q., Gu, H., Lu, L., Wang, K., Lin, D., Shen, G., Zhou, X., Zhang, L., Zang, Y., Dong, X., Wang, J., Zhang, B., Bai, L., Chu, P., Li, W., Wu, J., Wu, L., Li, Z., Wang, G., Tu, Z., Xu, C., Chen, K., Qiao, Y., Zhou, B., Lin, D., Zhang, W., & He, C. (2025). Mineru2.5: A decoupled vision-language model for efficient high-resolution document parsing. https://arxiv.org/abs/2509.22186.

Park, S., Shin, S., Lee, B., Lee, J., Surh, J., Seo, M., & Lee, H. (2019). Cord: A consolidated receipt dataset for post-ocr parsing. In *Workshop on document intelligence at neurIPS 2019* (pp. 1–4). https://openreview.net/forum?id=SJl3z659UH.

Peroni, S., & Shotton, D. (2012). Fabio and ciTO: Ontologies for describing bibliographic resources and citations. *Journal of Web Semantics*, *17*, 33–43.

Peroni, S., & Shotton, D. (2018). The SPAR ontologies. In *The semantic web–ISWC 2018: 17th International semantic web conference, Monterey, CA, USA, October 8–12, 2018, proceedings, part II 17* (pp. 119–136). Springer.

Persiani, S., Daquino, M., & Peroni, S. (2022). A programming interface for creating data according to the SPAR ontologies and the opencitations data model. In *European semantic web conference* (pp. 305–322). Springer.

Pfitzmann, B., Auer, C., Dolfi, M., Nassar, A. S., & Staar, P. (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* (pp. 3743–3751).

Pisaneschi, L., Gemelli, A., & Marinai, S. (2023). Automatic generation of scientific papers for data augmentation in document layout analysis. *Pattern Recognition Letters*, *167*, 38–44. https://doi.org/10.1016/j.patrec.2023.01.018

Pramanick, S., Chellappa, R., & Venugopalan, S. (2025). Spiqa: A dataset for multimodal question answering on scientific papers. https://arxiv.org/abs/2407.09413.

Prasad, A., Kaur, M., & Kan, M.-Y. (2018). Neural parscit: A deep learning-based reference string parser. *International Journal on Digital Libraries*, *19*, 323–337.

Priem, J., Piwowar, H., & Orr, R. (2022). Openalex: A fully-open index of scholarly works, authors, venues, institutions, and concepts. arXiv preprint arXiv:2205.01833.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J. et al. (2021). Learning transferable visual models from natural language supervision. In *International conference on machine learning* (pp. 8748–8763). PMLR.

Rausch, J., Martinez, O., Bissig, F., Zhang, C., & Feuerriegel, S. (2021). Docparser: Hierarchical document structure parsing from renderings. In *Thirty-fifth AAAI conference on artificial intelligence, AAAI 2021, thirty-third conference on innovative applications of artificial intelligence, IAAI 2021, the eleventh symposium on educational advances in artificial intelligence, EAAI 2021, virtual event, February 2–9, 2021* (pp. 4328–4338). AAAI Press. https://doi.org/10.1609/AAAI.V35I5.16558

Richardson, L. (2007). Beautiful soup documentation. Available at: https://www.crummy.com/software/BeautifulSoup/bs4/doc/.

Roberts, R. J. (2001). Pubmed central: The genbank of the published literature. *Proceedings of the National Academy of Sciences 98*(2), 381–382. https://doi.org/10.1073/pnas.98.2.381.

Saier, T., & Färber, M. (2019). Bibliometric-enhanced arxiv: A data set for paper-based and citation-based tasks. In *Bir@ ecir* (pp. 14–26).

Saier, T., Krause, J., & Färber, M. (2023). Unarxive 2022: All arxiv publications pre-processed for nlp, including structured full-text and citation network. In *2023 ACM/IEEE Joint conference on digital libraries (JCDL)* (pp. 66–70). IEEE.

Sauro, J., & Lewis, J. R. (2016). Quantifying the user experience: Practical statistics for user research. Morgan Kaufmann.

Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Hambro, E., Zettlemoyer, L., Cancedda, N., & Scialom, T. (2024). Toolformer: Language models can teach themselves to use tools. In *Advances in neural information processing systems*, *vol. 36*.

Shafait, F., & Smith, R. (2010). Table detection in heterogeneous documents. In D. S. Doermann, V. Govindaraju, D. P. Lopresti, & P. Natarajan (Eds.), *The ninth IAPR international workshop on document analysis systems, DAS 2010, June 9–11, 2010, Boston, Massachusetts, USA* ACM International Conference Proceeding Series (pp. 65–72). ACM. https://doi.org/10.1145/1815330.1815339

Shen, L., Shen, E., Luo, Y., Yang, X., Hu, X., Zhang, X., Tai, Z., & Wang, J. (2022). Towards natural language interfaces for data visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics*, *29* (6), 3121–3144.

Shinyama, Y. (2015). Pdfminer: Python pdf parser and analyzer. Available at: https://github.com/euske/pdfminer.

Soto, C., & Yoo, S. (2019). Visual detection with context for document layout analysis. In *Proceedings of the 2019 Conference on empirical methods in natural language processing and the 9th International joint conference on natural language processing (EMNLP-IJCNLP)* (pp. 3464–3470).

Tanasă, A.-M., & Oprea, S.-V. (2025). Rethinking chart understanding using multimodal large language models. *Computers, Materials and Continua*, *84* (2), 2905–2933. https://doi.org/10.32604/cmc.2025.065421

Tito, R., Karatzas, D., & Valveny, E. (2023). Hierarchical multimodal transformers for multipage docvqa. *Pattern Recognition*, *144*, 109834.

Tkaczyk, D., Collins, A., Sheridan, P., & Beel, J. (2018). Evaluation and comparison of open source bibliographic reference parsers: A business use case. arXiv preprint arXiv:1802.01168.

Tkaczyk, D., Szostek, P., Fedoryszak, M., Dendek, P. J., & Bolikowski, Ł. (2015). Cermine: Automatic extraction of structured metadata from scientific literature. *International Journal on Document Analysis and Recognition (IJDAR)*, *18*, 317–335.

Tran, M., Pham, T., Nguyen, T., Do, T., & Ngo, T. D. (2021). A robust framework for mathematical formula detection. In *International conference on multimedia analysis and pattern recognition, MAPR 2021, Hanoi, Vietnam, October 15–16, 2021* (pp. 1–6). IEEE. https://doi.org/10.1109/MAPR53640.2021.9585197

Van Landeghem, J., Tito, R., Borchmann, Ł., Pietruszka, M., Joziak, P., Powalski, R., Jurkiewicz, D., Coustaty, M., Anckaert, B., Valveny, E. et al. (2023). Document understanding dataset and evaluation (DUDE). In *Proceedings of the IEEE/CVF International conference on computer vision* (pp. 19528–19540).

Wade, A. D. (2022). The semantic scholar academic graph (s2ag). In *Companion proceedings of the web conference 2022* (pp. 739).

Wang, B., Gu, Z., Xu, C., Zhang, B., Shi, B., and He, C. (2024). UniMERNet: A Universal Network for Real-World Mathematical Expression Recognition. https://arxiv.org/abs/2404.15254.

Wang, B., Xu, C., Zhao, X., Ouyang, L., Wu, F., Zhao, Z., Xu, R., Liu, K., Qu, Y., Shang, F. et al. (2024b). Mineru: An open-source solution for precise document content extraction. arXiv preprint arXiv:2409.18839.

Wang, X., Yang, Q., Qiu, Y., Liang, J., He, Q., Gu, Z., Xiao, Y., & Wang, W. (2023). KnowledGPT: Enhancing large language models with retrieval and storage access on knowledge bases. arXiv preprint arXiv:2308.11761.

Wang, Z., Zhan, M., Liu, X., & Liang, D. (2020). Docstruct: A multimodal method to extract hierarchy structure in document for general form understanding. In T. Cohn, Y. He, & Y. Liu (Eds.), *Findings of the association for computational linguistics: EMNLP 2020, online event, 16–20 november 2020* (pp. 898–908). Association for Computational Linguistics (*vol. EMNLP 2020*). Findings of ACL. https://doi.org/10.18653/V1/2020.FINDINGS-EMNLP.80

Xia, R., Mao, S., Yan, X., Zhou, H., Zhang, B., Peng, H., Pi, J., Fu, D., Wu, W., Ye, H., Feng, S., Wang, B., Xu, C., He, C., Cai, P., Dou, M., Shi, B., Zhou, S., Wang, Y., Wang, B., Yan, J., Wu, F., & Qiao, Y. (2024). Docgenome: An open large-scale scientific document benchmark for training and testing multi-modal large language models. https://arxiv.org/abs/2406.11633.

Xu, Y., Li, M., Cui, L., Huang, S., Wei, F., & Zhou, M. (2020). LayoutLM: Pre-training of text and layout for document image understanding. In R. Gupta, Y. Liu, J. Tang, & B. A. Prakash (Eds.), *KDD '20: The 26th ACM SIGKDD conference on knowledge discovery and data mining, virtual event, CA, USA, August 23–27, 2020* (pp. 1192–1200). ACM. https://doi.org/10.1145/3394486.3403172

Yu, H., Gan, A., Zhang, K., Tong, S., Liu, Q., & Liu, Z. (2024). Evaluation of retrieval-augmented generation: A survey. arXiv preprint arXiv:2405.07437.

Zhang, J. (2023). Graph-toolformer: To empower LLMs with graph reasoning ability via prompt augmented by chatgpt. arXiv preprint arXiv:2304.11116.

Zhang, J., Yang, W., Lai, S., Xie, Z., & Jin, L. (2025). Dockylin: A large multimodal model for visual document understanding with efficient visual slimming. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 9923–9932). (*vol. 39*).

Zhao, Z., Kang, H., Wang, B., & He, C. (2024). Doclayout-YOLO: Enhancing document layout analysis through diverse synthetic data and global-to-local adaptive perception. https://arxiv.org/abs/2410.12628.

Zhong, X., Tang, J., & Jimeno-Yepes, A. (2019). Publaynet: Largest dataset ever for document layout analysis. In *2019 International conference on document analysis and recognition, ICDAR 2019, Sydney, Australia, September 20–25, 2019* (pp. 1015–1022). IEEE. https://doi.org/10.1109/ICDAR.2019.00166