An Integrated System for Interacting with Multi-Page Scholarly Documents

Lorenzo Massai[1,*], Simone Marinai[1]

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

Abstract

In this work we present a preliminary version of a comprehensive interface for supporting users to interact with scholarly documents, enabling multi-layered exploration and offering deeper insights by integrating diverse features and contextual information. By bridging diverse information our work pursues the identification, characterization, and linking of visual elements to semantic and context data, leveraging large language models for interoperability. Recent advances in retrieval augmented generation are also exploited to address some language models limitations, allowing them to access latent information from document representations such as graph and vector embeddings.

The system under development performs an analysis of input documents and enables the extraction of visual and semantic features, making them accessible in a comprehensive framework. The association of structural information to visual data allows formal analysis of documents and is exploited in our model to enhance visual extraction, performing a novel ontology-based constraint violation detection. The information extracted through this framework is semantically explorable, providing access to the document structure, which can be exploited in many applications like question answering and document understanding.

Keywords

Natural language processing, document layout analysis, conversational agents, retrieval augmented generation, large language models, question answering, document understanding, linked data, scholarly document processing, multi-modal feature extraction, text mining

1. Introduction

In recent years digital analysis of documents has gained attention due to the massive process of online media publishing and to the large availability of shared knowledge. Narrowing the field to the scientific literature context, readers are able to understand 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 PDF or images and getting automatic access to different kinds of knowledge requires to make several disjoint 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 exploration. The main challenges which are addressed in this work can be found in the following areas.

Semantic segmentation. The association of semantics to visual data is a key research problem in computer vision, including tasks like object recognition, image captioning, and image segmentation. In document analysis the goal is to understand contents by extracting geometric properties of visual elements such as *figures*, *tables*, *text* and layout elements such as *columns*, *footnotes*, *titles*, classifying them into semantic categories. Most document understanding systems are limited to text blocks and figure/text classification, lacking contextual information and domain-specific recognition (i.e. *listings*, *formulas*, and *chemical structures*). The scientific literature, with its variety of visual and text data,

IRCDL 2025: 21st Conference on Information and Research Science Connecting to Digital and Library Science, February 20-21, 2025, Udine, Italy

*Corresponding author.

lorenzo.massai@unifi.it (L. Massai); simone.marinai@unifi.it (S. Marinai)

0000-0002-8252-0549 (L. Massai); 0000-0002-6702-2277 (S. Marinai)

is particularly suitable for semantic segmentation and can be used to estimate associations between representations. However, these representations are limited by their lack of interoperability and moreover relying on images restricts analysis to one 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.

Semantic integration. The integration of different document attributes can be pursued merging extracted information in shared structures, either for retrieving information about visual and layout elements in the document, or to get context information about the publication such as the author and the research field. The presence of a formal structure helps maintain and extend context; relations can also be exploited to identify structural constraint violations like overlapping layout regions 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.

Layer interoperability. Recent trends for navigating different layers of information go towards intelligent agents which are aware of the subject being asked of, its context, and the context of who asks. 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. Visual Question Answering systems' dependency on document images reflects in limited awareness of the whole document and lack of any contextual information or specific domain recognition. 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.

This paper presents original contributions that advance the fields of analysis and interaction with scholarly documents. By integrating different media representations, this work pursues the enhancement of document understanding and interaction, addressing specific challenges in document processing. In particular, our main achievements can be identified in:

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

These contributions enhance the accessibility, explainability, and interoperability of scholarly document analysis, enabling semantic processing and navigation of academic papers.

2. Related work

The review of the state-of-the-art focuses on three distinct and related research areas: document segmentation, semantic linking, and visual question answering.

2.1. Document segmentation

Various approaches exist for identifying layout elements in visually structured documents, typically targeting specific types like *tables* [1], *formulas* [2], *bibliographic references* [3] and many others. Most approaches rely on OCR and TEI-XML conversion; for multi-page documents, current methods like HRDOC [4] use Mask R-CNN and language models to extract semantic regions and their relations.

Regarding the conversion from PDF to a more structured format such as TEI-XML, Grobid [3] is considered one of the ten best tool for extracting bibliography data from document images [5], allowing multi-page analysis and being also capable of extracting other layout elements. Whole document

analysis increases the problem complexity; to this end some efforts have been made to extract relations between pages in the form of triples [6].

Several multi-purpose datasets exist in the area of scholarly document understanding, focusing on layout analysis, text and visual elements extraction, and document structure identification. The largest datasets have to deal with the multiplicity of different layouts which are present in scholarly articles, addressing the complications of storing different data types into suitable structures. For this reason the most extended sources of information rely on flexible data containers like XML and JSON formats, which allow enough versatility for managing such a variety of descriptive data. Among recent datasets for scholarly documents layout analysis Publaynet [7] and DocBank [8] are considered the most relevant, although they exhibit limited variability in contents and layout.

2.2. Semantic linking

There are technical and pragmatic reasons to pursue abstract representations of knowledge with Linked (Open) Data and ontologies. Using natural language processing and computer vision strategies to obtain searchable content does not ensure the maintenance of the visual or logic structure of the original data, which is essential for data context analysis and is necessary to perform structured queries and inference. The definition of a structure capable of hosting data extracted from raw sources allows to keep the context and easily extend it, exploiting relations which exist among data and that are not explicitly declared.

The most encouraging effort in the direction of a unified structure for modeling scholarly documents can be found in the Semantic Publishing And Referencing (SPAR) ontologies project [9], [10], which includes several ontologies that are depicted in Figure 1. SPAR ontologies integrate models such as the Document Components Ontology (DoCO) [11] which, in turn, includes pattern ontologies, discourse elements ontologies, bibliographic resources ontologies, citation ontologies [12], and many others describing different aspects of scholarly documents.

The Document Components Ontology is composed of a rhetorical and a structural layer: rhetorical classes describe logical entities such as *references*, *bibliographic references*, *captions*, *introduction*, *materials*, *methods*, *results*, *related work* and *future work*. The structural layer links rhetorical elements with structural components like *title*, *section titles*, *paragraphs*, *footnotes*, *tables*, *figures*, *captioned boxes*, *figure boxes*, *lists*, *bibliographic reference* list, *front matter*, *body matter*, *back matter*, *chapters*, *sections*, *bibliography*, and *abstract*. Each class defines semantic relations with other classes, e.g. the class `Sentence` includes `DiscourseElement` when it is found with the attribute `inline`.

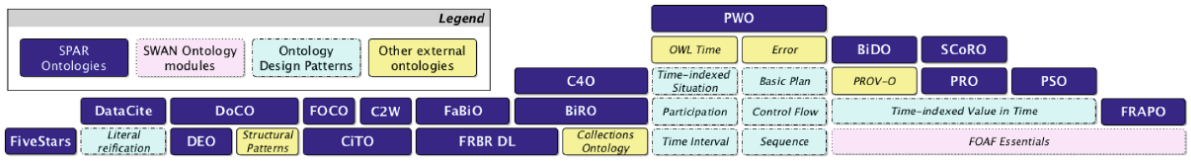

Figure 1: The SPAR ontologies modules [9].

2.3. Visual question answering

Visual Question Answering (VQA) represents the main point of contact between the communities of natural language processing and computer vision. Technologies such as conversational agents and chatbots [13] are suitable for this purpose. These technologies can interface with neural networks and ontologies [14], exploiting their functionalities like graph reasoning [15] for extending context. Question answering systems can be integrated and trained to respond to questions on both visual and contextual information. Retrieval Augmented Generation (RAG) [16] can further enhance these capabilities by combining retrieval to access custom knowledge bases and provide more accurate answers.

Toolformer [17] and KnowledGPT [18] integrate knowledge bases to Large Language Models (LLMs) with program-of-thought prompting, allowing questions requiring broader context knowledge. An effective application of RAG to scholarly articles can be found in ChatDOC[1] and in PaperQA [19], describing RAG agents that can answer scientific questions. Document images pose distinct challenges due to their spatially organized elements and the combination of visual and textual information. To this end LayoutLM [20] introduces 2D position embeddings, merging visual and text embeddings.

The main limitation of current research in scholarly documents VQA can be found in its reliance on page images, restricting the analysis to single-pages and disregarding semantic context. Some efforts have been made in this direction [21]. VQA datasets supporting multi-page documents are hard to find; among the most recent, comprehensive resources can be found in the MP-DocVQA dataset [22], the GRAM dataset [23] and the DUDE dataset [24]. The DUDE dataset includes a wide range of document types and sources, covering diverse topics and layouts, and allows full support for multi-page analysis, however having limited layout semantics. The lack of valuable multi-page datasets can also be addressed through document generation [25]. The most comprehensive resource for scholarly document analysis, in the best of our knowledge, is the Semantic Scholar Open Research Corpus (S2ORC) dataset [26]. S2ORC is composed by 8.1M open-access PDF-parsed papers across different academic disciplines and offers full reproducibility.

3. System architecture

The proposed architecture (Figure 2) is aimed at extracting different layers of information from multi-page scholarly articles exploiting state-of-the-art tools; future work is represented as dashed elements and bracketed labels. To achieve a comprehensive characterization of different kinds of layout elements, document data is extracted with vision, natural language, and semantic technologies. Information is made accessible altogether through conversational agents based on language models.

3.1. Document segmentation module

To extract geometric information a segmentation strategy aimed at identifying layout categories and their properties is presented. The PDF articles are converted to TEI-XML format through the Grobid API[2] in order to estimate the PDF structure as an XML tree. The resulting output contains the recognized structures which are *title*, *doi*, *keywords*, *abstract*, *authors*, *authors data*, *emails*, *tables*, *figures*, *captions*, *formulas*, *dates*, *sections/subsections*, *acknowledgments*, *bibliographic entries* and *raw text blocks*. Positional information includes page number and is present for most classes. Some structures have deeper characterization, for instance author consolidation is made through the integration with CrossRef APIs.

The output of Grobid processing is parsed through Beautiful Soup to extract the TEI tags and serialize the information into key-value pairs (Figure 3). The coordinates of semantic elements are then used to draw bounding boxes on the original document and to associate a label to each semantic region. The hierarchy of the document is also extracted. The recognized layout elements that are provided with geometric information are highlighted in the user interface through bounding boxes (Figure 4) and the information that is not provided with spatial data is associated to them as linked pop-ups.

The features in development that are related to the document segmentation module are represented in Figure 2 as dashed elements and bracketed labels.

3.2. Semantic linking module

The semantic characterization assigned to the extracted information is derived from the Document Components Ontology [11], a specialized ontology designed for modeling the layout elements of scholarly and research documents. The semantic network provided by DoCO is imported into the

[1] https://chatdoc.com/
[2] https://kermitt2-grobid.hf.space/

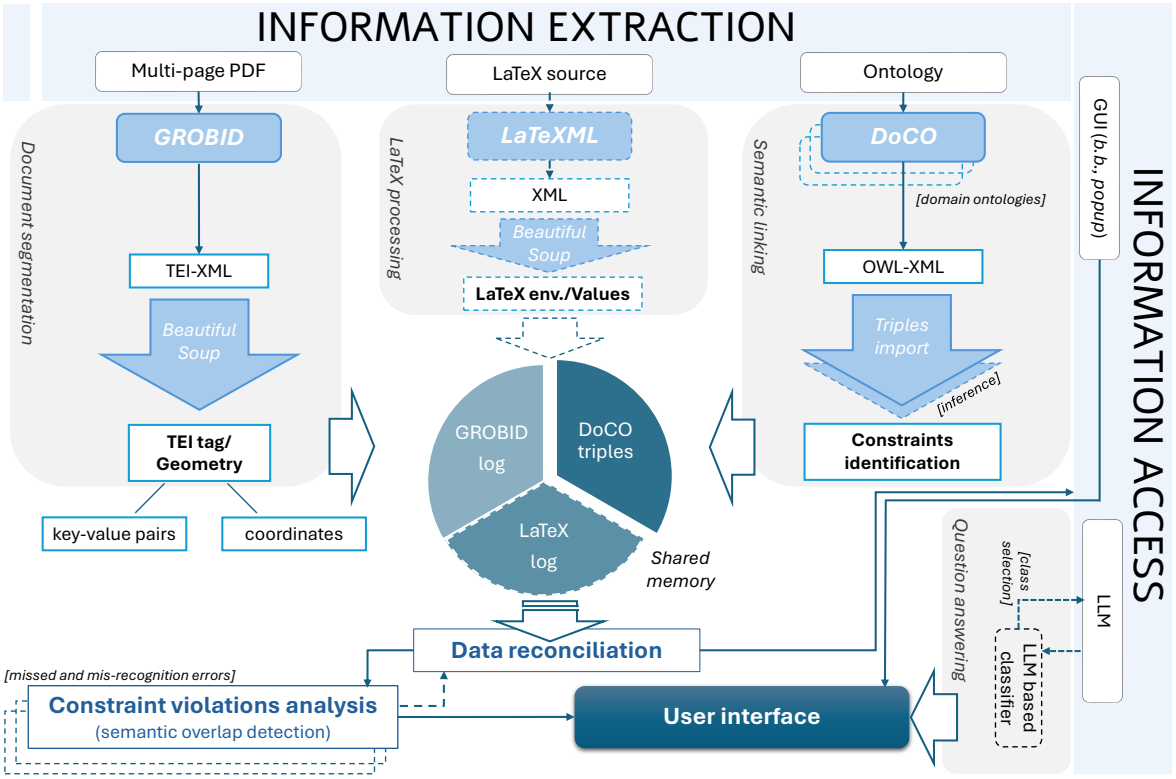

Figure 2: The architecture of the system. The features in development are represented as dashed elements and bracketed labels.

system, enabling a structured and standardized representation of document components. To ensure compatibility, a detailed mapping process is performed between the Grobid XML tags used for document parsing and the corresponding DoCO classes. This mapping is carried out by aligning the typical organization and content structure of a research article, ensuring semantic coherence and consistency across the extracted data. As detailed in Section 5 the semantic characterization of layout elements is leveraged to enhance visual recognition, exploiting the relations defined among the ontology classes to detect unfounded overlaps.

The features in development that are related to the semantic linking module are represented in Figure 2 as dashed elements and bracketed labels.

3.3. Question answering module

The question answering module is managed exploiting LLMs, specifically the Llama3 model through the Ollama Python API. This model has been chosen because of its ease of installation and integration with custom scripts and external resources. The question answering module is designed to enable the LLM to access the serialized output of Grobid, which is stored in a shared memory (Figure 3). The results obtained with Llama3 are excellent, ensuring an adequate understanding of the questions given the resources provided and their eventual lack. The response is fairly fast, even though the local installation does not have access to significant computational resources.
The user question is augmented and proposed to the LLM in the form:

"*Given that:* `log_data`, *question*"

where `log_data` represents the output of the modules described in Sections 3.1 and 3.2 and *question* is the query input by the user through the user interface. The system context is given to the LLM as:

"The questions will be about a scholarly article from which some data has been extracted in structured form and given as context."

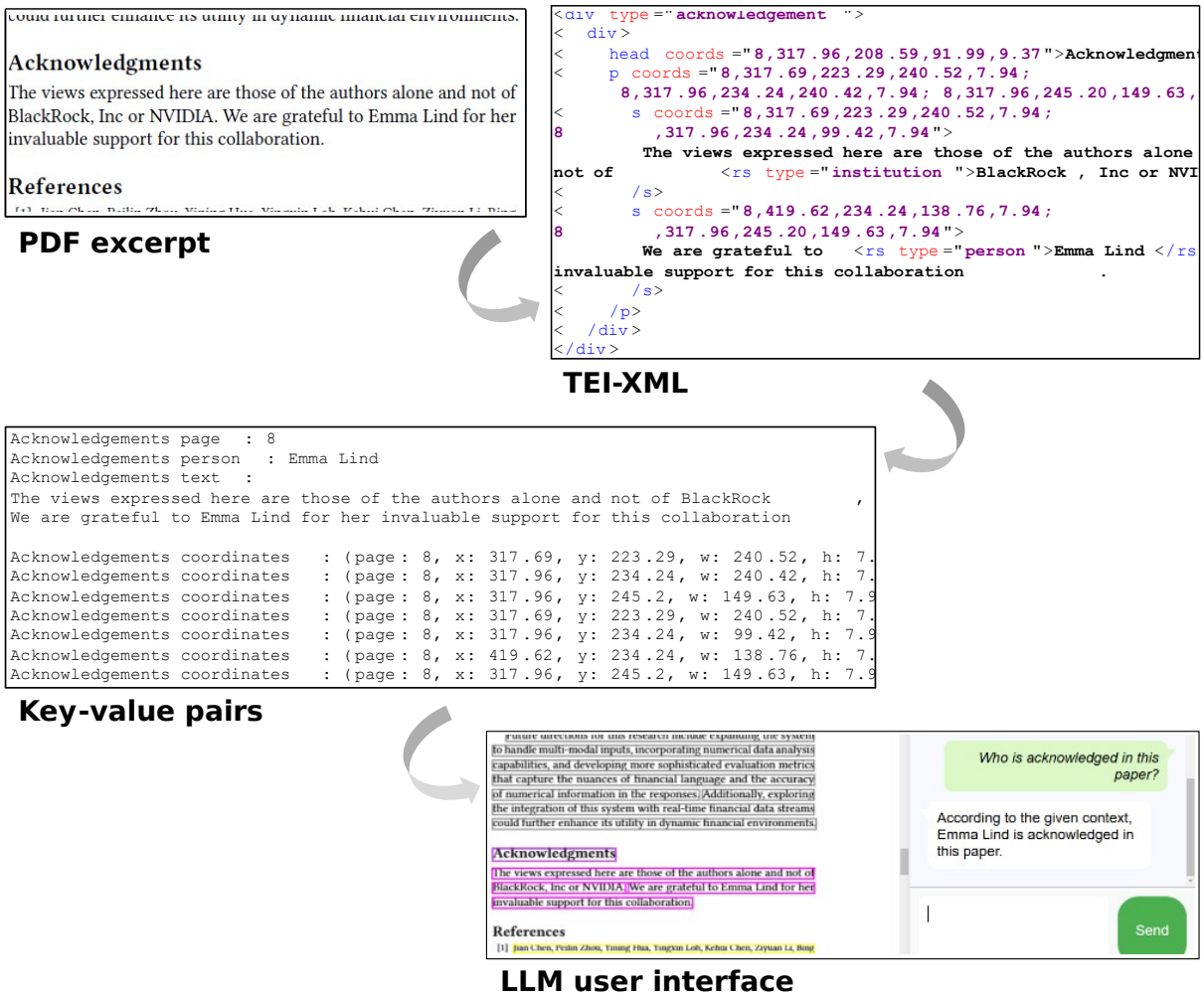

Figure 3: An example of the flow from PDF data to the structured information which is given as context to the LLM.

The context length for an off-the shelf language model as Llama3 is set to 2048 characters, which is restrictive both for the output of the Grobid and DoCO modules and for the extracted key-value pairs. To address the LLM context length issues and limiting the context to the part that is most pertinent to the question, the user interface described in Section 4 allows the user to provide a classification of the questions choosing any number of labels among: *Article_title*, *Author*, *Abstract*, *Caption*, *Caption_Figure*, *Figure*, *Table*, *Formula*, *Section*, *Link*, *Note*, *Acknowledgments*, and *Reference*. These classes correspond to Grobid-extracted TEI-XML tags, which are mapped to the DoCO ontology entities. The labels provided by the user are exploited to split the context to be given to the LLM, retaining only the portions that constitute the object of the classification.

The features in development that are related to the LLM module are represented in Figure 2 as dashed elements and bracketed labels.

4. User interface

The user interface (Figure 4) is composed of five web pages interacting with the Python server that routes the user choices. Through this interface the user is able to upload a PDF document (*Upload page*) and to process it (*PDF processor*), extracting information which is exploited by the LLM for accessing information. Upon PDF processing, its output is shown to the user, which is the document augmented with bounding boxes that span the classes described in Section 3.1. The process of drawing bounding boxes is managed by generating a separate PDF layer for each class, each layer being assigned

distinct colors, and then overlapping these layers onto the original input PDF to visually represent the annotations. The whole process is completed in a variable amount of time, mainly depending on the input PDF length and network capabilities, since Grobid is used as a network service. Processing a 10/20-page PDF 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 facilitate user interaction and exploration of the system's functionalities. The LLM computation time for each question varies based on local GPU capabilities, generally taking a few seconds.

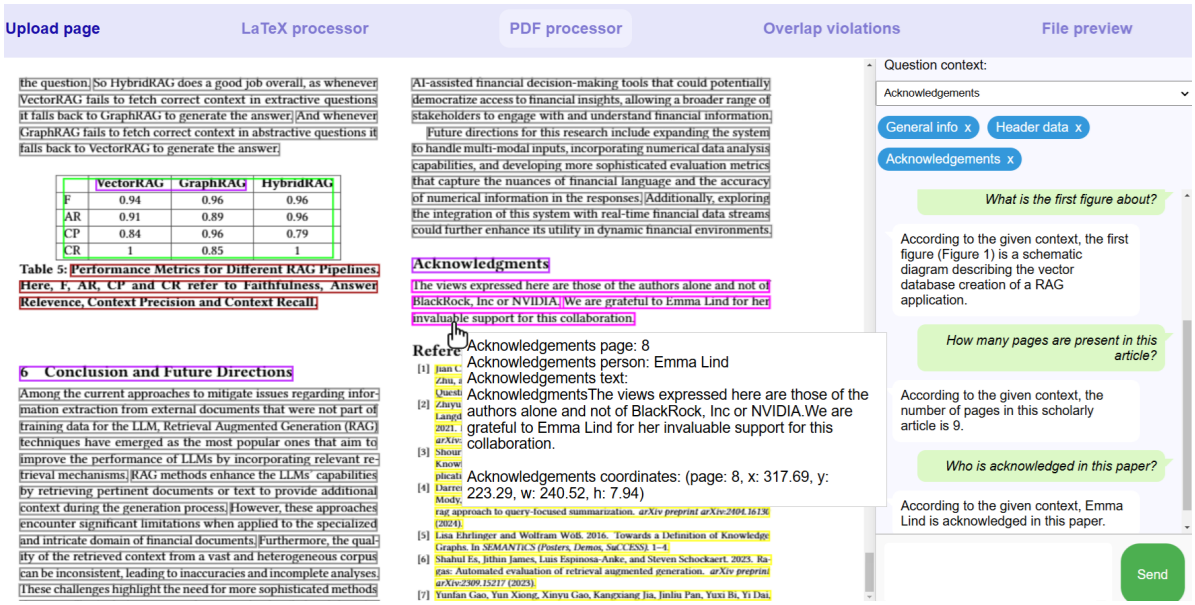

Figure 4: User interface with bounding boxes, element data popups (displaying `Acknowledgements` data) and LLM dialogue (best viewed in color).

Since LLMs have context length issues, the context of the question is chosen by the user filtering by question topic, which can be any number of layout classes, as detailed in Section 3.3. The interface includes a graphical preview of data, which consists of PDF images with bounding boxes overlaying the layout elements associated with coordinates in the Grobid output, each provided with a layout element label. In addition, informative pop-ups containing all data retrieved by data processing are present.
The user interface includes also a specific perspective (*Overlap violations*) 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 violating the imported ontology constraints.

5. Constraint violations analysis

To understand how the relations defined in an ontology can be applied to visual extracted classes and how they can improve the classification of layout elements, we analyze the interactions among the DoCO ontology classes[3]. The idea is to exploit the ontology relations that occur between layout elements to check the admissibility of geometric overlaps.
Since the assertions defined in the ontology can involve objects lacking spatial characterization in the Grobid-extracted counterpart, it is essential to identify the relations among objects with coordinates. Then, a geometric notion for each relation between class instances in the format `(page, x, y, width, height)` has to be defined. Afterwards, ontology constraints can be applied to detect visual recognition errors like overlap errors (Figure 5).

We distinguish overlaps as *geometric overlaps* and *semantic overlaps*, since the former can be admissible, while the latter are most likely recognition errors. We need to determine whether an overlap is admissible

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

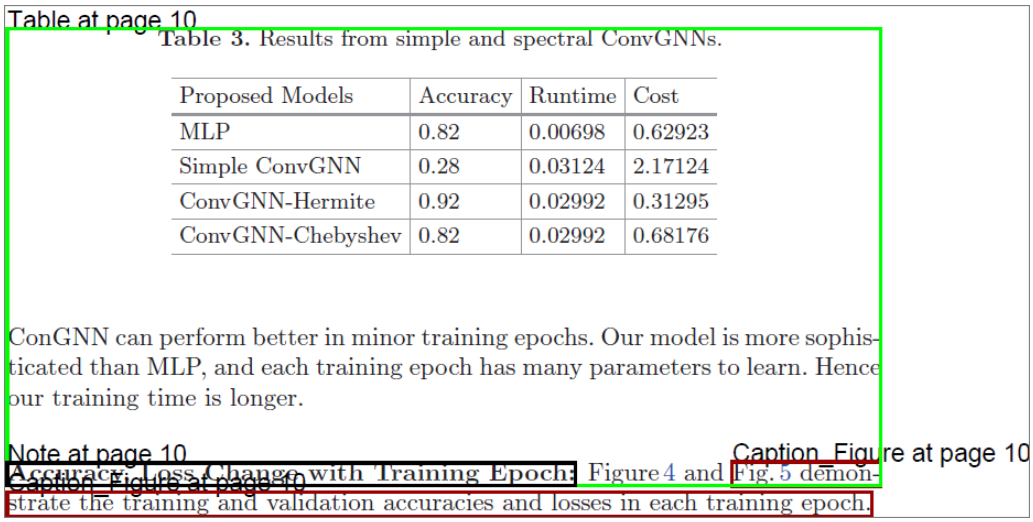

Figure 5: Example of overlapping bounding boxes corresponding to disjoint classes (best viewed in color)

(e.g., a *Name* object overlapping with an *Author* object is admissible, whereas a *Title* object overlapping with a *Figure* object is not) based on constraints defined in a formal model. To allow constraint verification the DoCO ontology relations are exploited; to express inadmissible layout element overlaps the `owl:disjointWith` relation is considered.

To formally determine which two-dimensional elements overlap in a context where we have coordinates defining their position on a page, we can treat the elements as rectangles defined by the following properties:

- **Page**: the page number (if two elements are on different pages, they 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 in a two-dimensional space. Formally, given two rectangles defined by:

1. Rectangle A:
   a) x_A, y_A (coordinates of the top-left corner)
   b) width_A, height_A
2. Rectangle B:
   a) x_B, y_B (coordinates of the top-left corner)
   b) width_B, height_B

To check for overlap, we need to verify whether 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:

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

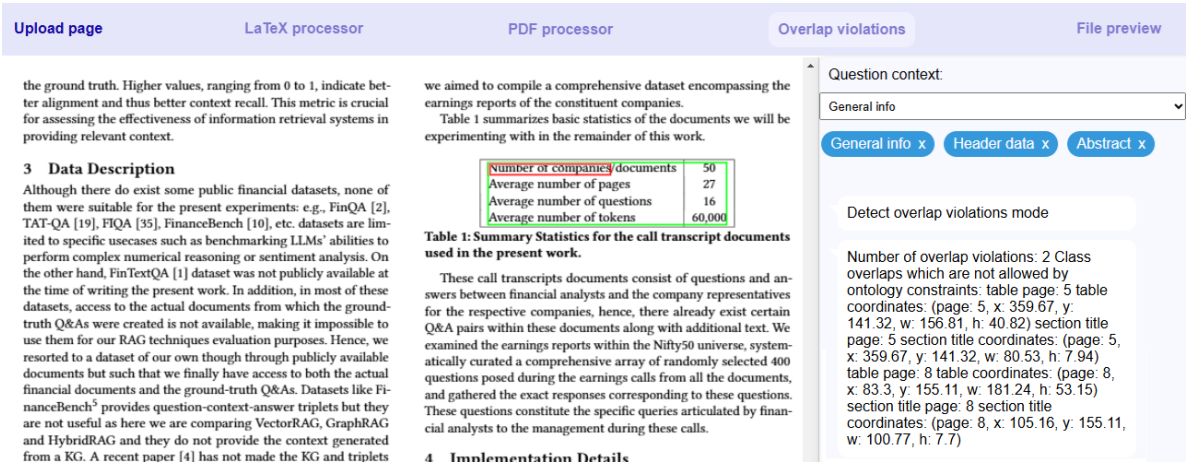

Figure 6: Detection of not-allowed overlaps based on semantic constraint analysis (best viewed in color).

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

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

To identify the overlapping errors we focus on the rectangles which overlap and for each we classify it as *admissible* or *not admissible*, checking as *not admissible* an overlap generated by classes which are disjoint in the ontology general axioms[4] and that are reported in Table 1. The Overlap violations section of the user interface described in Section 5 helps to highlight the bounding boxes which are associated to classes that are not allowed to overlap (Figure 6).

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 Figure 5 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

6. Future work

Future directions involve leveraging LaTeX source attributes, the analysis of more relations and ontologies, and a broader employment of the LLM for context length optimization.

Currently, the system allows the exploration of visual data which is extracted through the segmentation module. More modules can be linked for extending the knowledge associated with explorable elements, such as the LaTeX representation of the document [27]. Associating different representations of the document elements would also enable automatic construction of class-specific datasets, i.e. formulas and chemical structures datasets. Moreover, the custom-made user interface described in

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

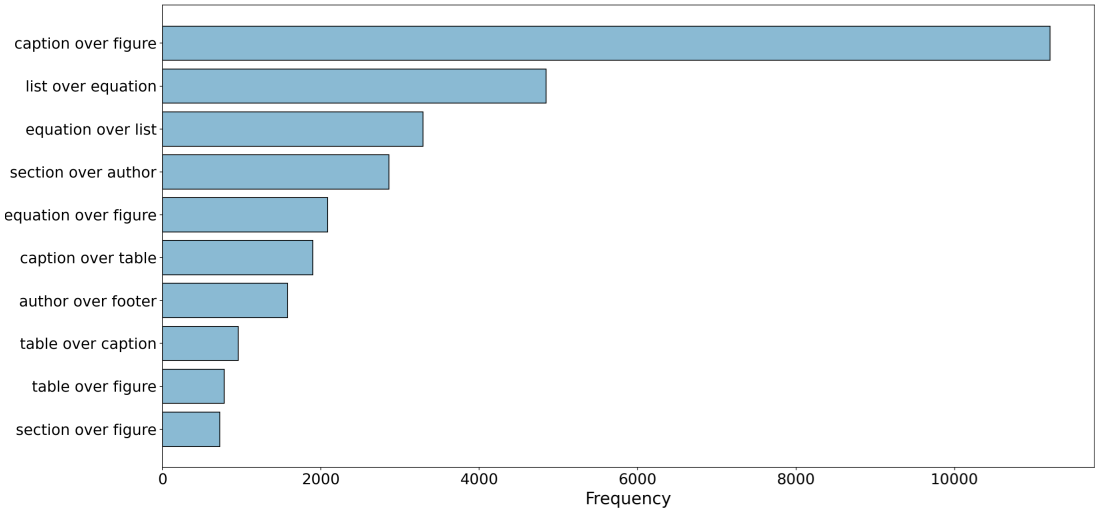

Figure 7: Top 10 overlap kinds on the Docbank dataset.

Section 5 allows code instrumentation, enabling comparison with state-of-the-art systems with similar purposes such as ChatDOC and Amazon Textract.

The variety of ontology relations employments can be extended for identifying more than overlap errors. To this end, parent relations can be exploited to detect misclassifications and elements missing paired classes (i.e. *figures* and *captions*). In addition, the presence of the ontology layer enables the possibility of extending the present structure with broader context ontologies [28] and exploiting reasoning capabilities to expand actual relations with the inferable ones.

The main limitation of the LLM module lies in its reliance on user classification of the query, aimed at reducing context length. The same result is achievable through unsupervised classification of the user query, which can be demanded to a dedicated LLM module. It is also to be noticed that the LLM performances would increase by employing language models with more parameters.

Current objectives include an assessment over the Docbank dataset, which contains numerous layout classes and overlaps such as *caption over figure*, *list over equation* and *section over author*, among others, an excerpt of which is presented in Figure 7. It is to be noticed that the present analysis on the Docbank dataset does not take into account semantic characterization, thus including some overlapping layout elements that are admissible (i.e. *equation over list*).

7. Conclusions

This paper extends scholarly document understanding and document question answering research fields pursuing the association of semantics and context to visual features, integrating them in a comprehensive interface which allows multi-layer exploration via LLM and interactive visualization. Linking semantic information to documents is challenging from a research perspective: most of the solutions reviewed in the state-of-the-art exhibit limited awareness of the described domain, considering only basic relations between text chunks. We leverage the Document Components Ontology focusing on semantic relations among layout elements to detect a specific kind of visual recognition errors, which are overlap errors, paving the way for more sophisticated layout elements interaction analysis. The proposed approach is based on the use of disjointness relations that may exist between overlapping layout elements. This relation is analyzed and interpreted as an indicator of potential recognition errors, providing a systematic way to identify and address inconsistencies in the detected layout structure. By exploiting this property, our method improves the accuracy and reliability of the recognition process. In addition, we exploit LLMs in our framework to enhance the accessibility of diverse information which is not directly available from data, enabling navigation of different kinds of information from an integrated interface.

Acknowledgments

This research has been partially funded by CAI4DSA[5] actions (Collaborative Explainable neuro-symbolic AI for Decision Support Assistant), of the FAIR national project on artificial intelligence, PE 1 PNRR (https://fondazione-fair.it/).

References

[1] A. Gemelli, E. Vivoli, S. Marinai, Graph neural networks and representation embedding for table extraction in pdf documents, in: 2022 26th International Conference on Pattern Recognition (ICPR), IEEE, 2022. URL: http://dx.doi.org/10.1109/ICPR56361.2022.9956590. doi:10.1109/icpr56361.2022.9956590.

[2] M. Tran, T. Pham, T. Nguyen, T. Do, T. D. Ngo, A robust framework for mathematical formula detection, in: International Conference on Multimedia Analysis and Pattern Recognition, MAPR 2021, Hanoi, Vietnam, October 15-16, 2021, IEEE, 2021, pp. 1–6. URL: https://doi.org/10.1109/MAPR53640.2021.9585197. doi:10.1109/MAPR53640.2021.9585197.

[3] P. Lopez, Grobid, https://github.com/kermitt2/grobid, 2008–2024. arXiv:1:dir:dab86b296e3c3216e2241968f0d63b68e8209d3c.

[4] J. Ma, J. Du, P. Hu, Z. Zhang, J. Zhang, H. Zhu, C. Liu, 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, AAAI Press, 2023, pp. 1870–1877. doi:10.1609/AAAI.V37I2.25277.

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

[6] M. Arif Demirtaş, B. Oral, M. Yasin Akpınar, O. Deniz, Semantic parsing of interpage relations, in: 2022 26th International Conference on Pattern Recognition (ICPR), 2022, pp. 1579–1585. doi:10.1109/ICPR56361.2022.9956546.

[7] X. Zhong, J. Tang, A. Jimeno-Yepes, 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, IEEE, 2019, pp. 1015–1022. URL: https://doi.org/10.1109/ICDAR.2019.00166. doi:10.1109/ICDAR.2019.00166.

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

[9] S. Peroni, D. Shotton, 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, Springer, 2018, pp. 119–136.

[10] S. Persiani, M. Daquino, S. Peroni, A programming interface for creating data according to the spar ontologies and the opencitations data model, in: European Semantic Web Conference, Springer, 2022, pp. 305–322.

[11] A. Constantin, S. Peroni, S. Pettifer, D. M. Shotton, F. Vitali, The document components ontology (doco), Semantic Web 7 (2016) 167–181. URL: https://doi.org/10.3233/SW-150177. doi:10.3233/SW-150177.

[12] S. Peroni, D. Shotton, Fabio and cito: ontologies for describing bibliographic resources and citations, Journal of Web Semantics 17 (2012) 33–43.

[13] D. S. Mishra, A. Agarwal, B. Swathi, K. C. Akshay, Natural language query formalization to sparql for querying knowledge bases using rasa, Progress in Artificial Intelligence 11 (2022) 193–206.

[14] L. Massai, P. Nesi, G. Pantaleo, Paval: A location-aware virtual personal assistant for retrieving geolocated points of interest and location-based services, Engineering Applications of Artificial Intelli-

[5] CAI4DSA ID:EP_FAIR_001 CUP:B13C23005640006

gence 77 (2019) 70–85. URL: https://www.sciencedirect.com/science/article/pii/S0952197618301994. doi:https://doi.org/10.1016/j.engappai.2018.09.013.

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

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

[17] T. Schick, J. Dwivedi-Yu, R. Dessì, R. Raileanu, M. Lomeli, E. Hambro, L. Zettlemoyer, N. Cancedda, T. Scialom, Toolformer: Language models can teach themselves to use tools, Advances in Neural Information Processing Systems 36 (2024).

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

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

[20] Y. Huang, T. Lv, L. Cui, Y. Lu, F. Wei, 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, ACM, 2022, pp. 4083–4091. URL: https://doi.org/10.1145/3503161.3548112. doi:10.1145/3503161.3548112.

[21] D. Napolitano, L. Vaiani, L. Cagliero, On leveraging multi-page element relations in visually-rich documents, in: 2024 IEEE 48th Annual Computers, Software, and Applications Conference (COMPSAC), IEEE, 2024, pp. 360–365.

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

[23] T. Blau, S. Fogel, R. Ronen, A. Golts, R. Ganz, E. Ben Avraham, A. Aberdam, S. Tsiper, R. Litman, Gram: Global reasoning for multi-page vqa, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 15598–15607.

[24] J. Van Landeghem, R. Tito, Ł. Borchmann, M. Pietruszka, P. Joziak, R. Powalski, D. Jurkiewicz, M. Coustaty, B. Anckaert, E. Valveny, et al., Document understanding dataset and evaluation (dude), in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 19528–19540.

[25] L. Pisaneschi, A. Gemelli, S. Marinai, Automatic generation of scientific papers for data augmentation in document layout analysis, Pattern Recognition Letters 167 (2023) 38–44. URL: https://www.sciencedirect.com/science/article/pii/S0167865523000247. doi:https://doi.org/10.1016/j.patrec.2023.01.018.

[26] K. Lo, L. L. Wang, M. Neumann, R. Kinney, D. Weld, S2ORC: The semantic scholar open research corpus, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 4969–4983. URL: https://www.aclweb.org/anthology/2020.acl-main.447. doi:10.18653/v1/2020.acl-main.447.

[27] D. Müller, An html/css schema for tex primitives–generating high-quality responsive, TUGboat 44 (2023) 275–286. URL: https://kwarc.info/people/dmueller/pubs/tug23.pdf.

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