







OntoHuman: Ontology-Based Information Extraction Tools with Human-in-the-Loop Interaction

Kobkaew Opasjumruskit^(✉), Sarah Böning, Sirko Schindler,
and Diana Peters

German Aerospace Center (DLR), Institute of Data Science, Jena, Germany
kobkaew.opasjumruskit@dlr.de

Abstract. This paper presents OntoHuman, a toolchain for involving humans in a process of automatic information extraction and ontology enhancement. Document Semantic Annotation Tool (DSAT) [13], a user interface of OntoHuman, offers an automatic function to extract information in the form of key-value-unit tuples from PDF documents based on ontologies. Additionally, it allows users to provide feedback to improve the ontologies used. Although the information extraction can be improved with the ontology, our use cases were previously limited to an area of space engineering. OntoHuman now tackles this shortcoming by allowing users to upload their customized ontologies. This extends usages to various domains and enables this shareable knowledge to be used cooperatively. Then we display the ontologies in a node-link representation so they are easier to understand. Another major improvement in OntoHuman is the graph data points extraction, which is still missing in the existing information extraction tools. The application of OntoHuman can be used for documents related to any engineering domain and makes the work with ontologies intuitive and collaborative for users.

Keywords: Semantic technologies for information-integrated collaboration · Ontology for information sharing · Web based cooperation tools

1 Introduction

In engineering design and development processes, different models are to be integrated and linked to coherent digital system models. These models consist of many components of which information are based on the suppliers' product data sheets and engineers' implicit knowledge. To consolidate scattered sources of information and thus supporting cooperative process, a product data hub is proposed [16]. It enables up-to-date product information to be digitally exchanged between all stakeholders. We further developed a solution to extract information and handle ambiguities with semantic knowledge combining with a human-in-the-loop method as demonstrated in [13]. There, fixed ontologies are used to

maintain the semantic knowledge. They can also link to external entities, e.g. from Wikidata [19]. We use an Ontology-Based Information Extraction (OBIE) to support our automatic extraction.

Most OBIE tools are tailored to extract entities and their relationships [11] but fall short when it comes to extracting literal values in the form of key-value(-unit) pairs. Furthermore, the vocabulary used in technical documents is highly domain-specific and not consistently used [1]. Not detecting correct information in the beginning can have fatal and costly consequences in the later phases of design and production.

This paper is a continuation of our previous contribution by allowing for more flexibility and reduce the barrier in using ontologies collaboratively. DSAT now enables users to upload their ontology for the domain-independent extraction process. To enhance the user experience, the uploaded ontology can be previewed in a node-link representation along with its metadata. We also offer the feedback UI, so multiple users can help correcting the knowledge base. In addition to the text-based information, technical documents often have graphical information, e.g. a plotted graph, which may contain vital information. Therefore, the data points on plotted graphs are considered and extracted from the documents. These improvements have not been fully tackled in existing tools. They are crucial to the automatic extraction process from technical documents, since they will mitigate the human error in misinterpreting data.

Evaluation results for ontology enrichment and information extraction were publicly available¹. Recently we also conducted two workshops² with users from various domains. We reviewed our integrated systems and collected feedback for further improvement. In the following section we review the related work. Then, we explain our system architecture and demonstrate how to use our tools. Finally, we conclude our work and propose the future work.

2 Related Work

The extraction of information from documents, commonly PDF files, has been widely discussed and is publicly available as reviewed in [14]. *Camelot* [5] is an open source software tool to extract tabular data from PDF files. *PDFminer* [15] is an open-source and actively maintained PDF parser library in Python, which offers text, images and tables extraction with customizable parameters. For some document that the textual content can not be extracted directly, Optical Character Recognition (OCR) tools like *OCR Tesseract* [18] can be applied to mitigate the issue. However, most of the existing tools focus on either text or tables. To the extent of our knowledge, there is no unified solution that tackles both of these information sources. To achieve the best result, OntoHuman combined the aforementioned techniques by using *PDFminer* and *OCR Tesseract* to extract text, then *Camelot* to extract tables.

¹ <https://arxiv.org/pdf/1906.06752.pdf>.

² <https://nfdi4ing.de/community-hub/community/>.

In addition to the information extraction from text and tables, images that contain data plots are equally important. *VizExtract* [7] and *ChartOCR* [12] apply OCR, image processing, and Machine Learning (ML) techniques to extract information from different types of charts. While *ChartOCR* focuses on extracting data point values from a chart, *VizExtract* offers more variety of charts and yields better accuracy. Based on these implementation ideas, we extended our OntoHuman’s information extraction with the graph data points extraction.

Entity recognition tools can detect important keywords from the text extracted. *AWS Textract* offers a pay-as-you-use tool to automatically extract key-value pairs from only forms and tables in document images [3]. Many works are also using hybrid approaches using image processing and OCR to extract text and derive key-value pairs using regular expressions such as [10]. DocStruct [20] uses ML techniques based on semantics, layout, and visual clues to detect key-value pairs from documents. Although most of the recent works are tackling the key-value pair extraction problem with ML techniques, these works were evaluated and aiming to extract the information from certain types of documents, especially, forms and receipts. To extract key-value pairs or key-value-unit tuples from documents from wider range of domains, our work combined ML and other recent techniques with existing domain knowledge approaches like OBIE. Unstructured or semi-structured text is processed using ontologies to extract information. The applications of OBIE are found useful in many domains such as medicine [9], engineering [17], and the legal domain [4].

To improve ontologies along the extraction process, OntoHuman engages users to choose, review, and edit ontologies, even if they are not ontology experts. Various ontology visualization tools and methods are reviewed in [2, 8]. The common and simple-to-understand implementations are treemaps, indented lists, and node-link visualizations. Each implementation has its own advantages and drawbacks, depends on the usage. Thus, we will conduct a user study in the future to evaluate which representation fits best.

3 System Overview

The OntoHuman toolchain, as shown in Fig. 1, consists of three main components: an annotation tool, an information extraction pipeline, and an ontology enhancer. The main inputs are technical documents describing products obtained from websites of manufacturers and retailers, and ontologies which describe the concept and properties of such products [6].

DSAT is a standalone tool assisting users to manually or automatically annotate data. Users can trigger an automatic extraction process via (ConTrOn)’s API. Afterwards, the extracted key-value(-unit) tuples are returned and highlighted in the document display on DSAT. Then, users can review the results and correct any mistake made by the system. The corrections will be collected and considered for updating the ontologies later. Up to now, the update must be done manually on the backend, since the ontology is used by several systems

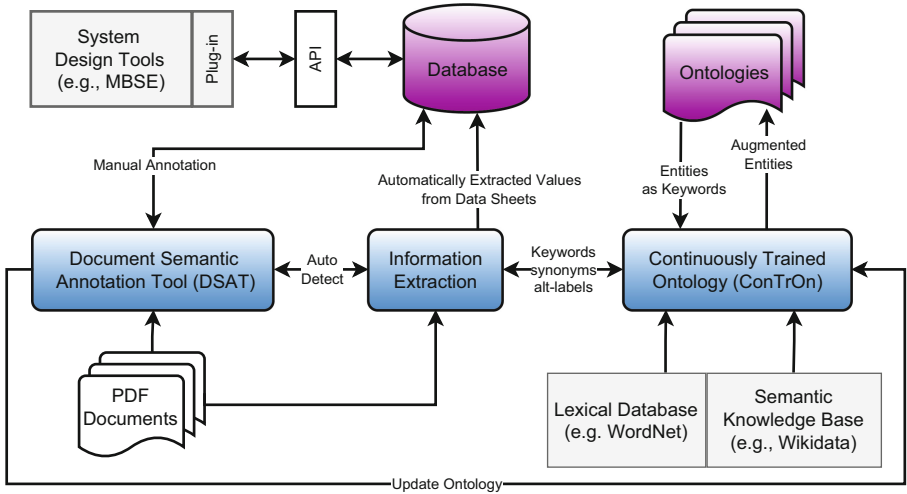


Fig. 1. System architecture of OntoHuman. Its main components are DSAT, the information extraction, and ConTrOn.

and changes need to undergo a curating step before being applied. Finally, the extracted data can be saved to a data hub which further enables the external components to retrieve the data automatically.

The *Information Extraction* part is a standalone package that searches for key-value-unit tuples within given PDF documents. The keywords (i.e. attribute names) are defined alongside allowed units of measure in the domain knowledge. Values can be any floating point number expression including combinations (e.g., $value \times value \times values$), and symbols ($>$, $<$, \leq , \geq , \sim). First, text extracted from the PDF files is distinguished between unstructured (running text) and structured elements (tables). The structure of the tables is preserved and leveraged later for the tuple extraction. Next, all inputs (tables, text, domain knowledge) are processed in a normalization step to remove potential extraction errors and canonicalize them. Lastly, the key-value-unit tuples are extracted from the texts and tables separately and subsequently merged while removing duplicates. The domain knowledge is used to verify found entries and store only valid ones.

ConTrOn, a standalone application with web API, is responsible for parsing ontologies to support the information extraction, also using the extracted information (and user feedback if available) to extend ontologies later. Furthermore, it extends the existing ontology with information from external semantic knowledge bases such as Wikidata. We can extract information such as subclasses, superclasses, related entities, or alternative labels including those from different languages from such knowledge bases.

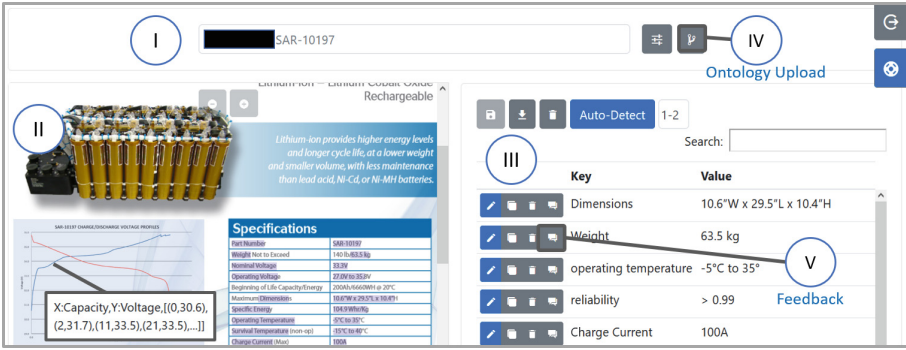


Fig. 2. DSAT interface (I) Document Selection (II) PDF Preview (III) Annotations List (IV) Ontology Preview and Upload (V) Feedback for Auto-Detection

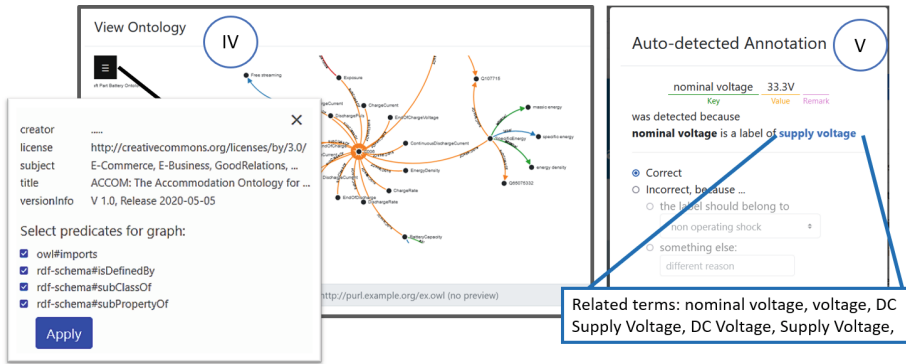


Fig. 3. Left: DSAT’s ontology management interface. The currently selected ontology is displayed in node-links format, and the metadata of the ontology can be shown/hidden. Right: DSAT’s annotation feedback for correcting an ontology.

4 Demo

DSAT’s UI (see Fig. 2) has three main sections: (I) Document and Ontology Selection, (II) Document Preview (PDFView), and (III) Document Annotations. Users can select or upload documents (I) via a modal dialog. The document’s metadata can be updated to define the domain of context. This domain of context is then used for selecting a suitable ontology for the automatic extraction process. Furthermore, users can upload their own ontologies via (IV). The ontology used for the automatic information extraction can be previewed as a node-link map as shown in Fig. 3(left). The metadata of the ontology is also summarized in the side-panel, which is collapsible and expandable. In the current implementation, the ontology displayed is not editable on this user interface, but to upload an externally-edited ontology is possible.

To manually annotate the key-value information, users can select a text in (II) and right-click to create an annotation via the context menu. The document annotations view (III) shows the list of annotations made on the selected document. Each annotation can be edited, cloned, and deleted. When users click the “Auto-Detect” button, the document will be processed by the information extraction and ConTrOn. The results will be appended to the table, as well as be highlighted on the PDFView. Additionally, the automatic detection offers users a graphical information extraction, i.e. data plots can be extracted as arrays of data points with labels as displayed in Fig. 2-bottom-left.

If an attribute (key) is incorrectly identified by the system, users can suggest the correct description via a feedback modal (V) (see Fig. 3-right). Currently, all corrections and suggestions must be reviewed by domain experts before being applied to the ontologies. Consequently, the OBIE process will get improved as well as the quality of information extraction, since the irrelevant keywords will be removed and the unknown keywords will be added to the ontologies.

5 Conclusion and Future Work

Based on previous development, this paper presents a recently improved document annotation tool, which is integrated into a toolchain to serve purposes of the project name OntoHuman. We aim to achieve two goals: to automatically extract technical information from documents, and to involve users in the collaborative improvement of underlying ontologies. Users who are familiar with ontologies can edit and upload their own ontologies. Either uploaded or predefined, ontologies can be viewed in a node-link diagram where users can pan, drag, search, and zoom to explore ontologies. Though the current implementation is only previewing, we plan to enable editing on the node-link diagram directly, so users can suggest the update of ontology more intuitively. Another way to suggest a change to the ontology is to provide feedback on an individual annotation. The respective interface is currently in a prototypical state and will be subject to further improvement. Besides the textual information extraction, we also consider graph data points extraction, which requires both text and image processing. This part is designed to be a standalone module, so that it can be reused and improved independently. Finally, we plan to conduct a formal evaluation in the near future to measure the performance of the information extraction and user experience on the usage of ontology.

References

1. Adnan, K., Akbar, R.: Limitations of information extraction methods and techniques for heterogeneous unstructured big data. *Int. J. Eng. Bus. Manag.* **11** (2019). <https://doi.org/10.1177/1847979019890771>
2. Anikin, A., Litovkin, D., Kultsova, M., Sarkisova, E., Petrova, T.: Ontology visualization: approaches and software tools for visual representation of large ontologies in learning. In: Kravets, A., Shcherbakov, M., Kultsova, M., Groumpos, P. (eds.)

- CIT&DS 2017. CCIS, vol. 754, pp. 133–149. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-65551-2_10
3. Classifying text with AWS Textract. <https://www.bakertilly.com/insights/classifying-text-with-aws-textract>. Accessed 8 Apr 2022
 4. Buey, M.G., Garrido, A.L., Bobed, C., Ilarri, S.: The AIS project: boosting information extraction from legal documents by using ontologies. In: ICAART (2016)
 5. Camelot: PDF Table Extraction for Humans. <https://camelot-py.readthedocs.io/en/master/>. Accessed 8 Apr 2022
 6. ConTrOn. Contron - spacecraft parts ontology 1.2, May 2020
 7. Decatur, D., Krishnan, S.: Vizextract: automatic relation extraction from data visualizations. CoRR abs/2112.03485 (2021)
 8. Dudáš, M., Lohmann, S., Svátek, V., Pavlov, D.: Ontology visualization methods and tools: a survey of the state of the art. Knowl. Eng. Rev. **33**, e10 (2018)
 9. Jusoh, S., Awajan, A., Obeid, N.: The use of ontology in clinical information extraction. J. Phys. Conf. Ser. **1529**(5), 052083 (2020)
 10. Kaló, A.Z., Sipos, M.L.: Key-value pair searching system via tesseract OCR and post processing. In: 2021 IEEE 19th World Symposium on Applied Machine Intelligence and Informatics (SAMII), pp. 000461–000464 (2021)
 11. Konys, A.: Towards knowledge handling in ontology-based information extraction systems. Procedia Comput. Sci. **126**, 2208–2218 (2018). Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 22nd International Conference, KES-2018, Belgrade, Serbia
 12. Luo, J., Li, Z., Wang, J., Lin, C.-Y.: Chartocr: data extraction from charts images via a deep hybrid framework. In: 2021 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1916–1924 (2021)
 13. Opasjumruskit, K., Peters, D., Schindler, S.: DSAT: ontology-based information extraction on technical data sheets. In: SEMWEB (2020)
 14. How to extract data out of a PDF, February 2021. <https://academy.datawrapper.de/article/135-how-to-extract-data-out-of-pdfs>
 15. PDFMiner - a python package for extracting information from PDF documents. <https://pdfminersix.readthedocs.io/en/latest/>. Accessed 8 Apr 2022
 16. Peters, D., Fischer, P.M., Schäfer, P.M., Opasjumruskit, K., Gerndt, A.: Digital availability of product information for collaborative engineering of spacecraft. In: Luo, Y. (ed.) CDVE 2019. LNCS, vol. 11792, pp. 74–83. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30949-7_9
 17. Rizvi, S.T.R., Mercier, D., Agne, S., Erkel, S., Dengel, A., Ahmed, S.: Ontology-based information extraction from technical documents. In: Proceedings of the 10th International Conference on Agents and Artificial Intelligence. SCITEPRESS - Science and Technology Publications (2018)
 18. Tesseract Open Source OCR Engine. <https://tesseract-ocr.github.io/>. Accessed 13 Apr 2022
 19. Vrandečić, D., Krötzsch, M.: Wikidata: a free collaborative knowledgebase. Commun. ACM **57**(10), 78–85 (2014)
 20. Wang, Z., Zhan, M., Liu, X., Liang, D.: Docstruct: a multimodal method to extract hierarchy structure in document for general form understanding. [arXiv:abs/2010.11685](https://arxiv.org/abs/2010.11685) (2020)