OntoHuman: Ontology-based Information Extraction tools with Human-in-the-loop interaction

Authors name are hidden.

Institutes are hidden.

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), a user interface of OntoHuman, offers on the one hand an automatic function to extract information in the form of key-value-unit tuples from PDF documents based on ontologies. On the other hand, 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 tackles this previous shortcoming by allowing users to upload customized ontologies and display them in a node-link representation so they are easier to understand. Another major improvement in OntoHuman is the graph data points extraction. This expands the scope of data extraction beyond the textual data to include plotted graphs, which usually requires human interpretation. The application of OntoHuman can be used for documents related to any engineering domain and makes the work with ontologies intuitive 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, Model-Based Systems Engineering (MBSE) tools are used to integrate different models and link them with a coherent digital system model. These models can consist of many components provided by different suppliers. Their information and attributes are based on the suppliers' product data sheets and, oftentimes, engineers' implicit knowledge. To consolidate scattered sources of information, a product data hub is proposed [18]. It enables up-to-date product information to be digitally exchanged between all stakeholders. We further developed a solution with Natural Language Processing (NLP) to help extracting information and handle ambiguous issues with semantic knowledge combining with a human-in-the-loop method as demonstrated in [9]. An ontology is used to maintain the semantic knowledge, which can also link to external entities, e.g. from Wikidata [21]. Therefore, we use an Ontology-Based Information Extraction (OBIE) to support our automatic extraction implementation.

Most OBIE tools are tailored to extract entities and their relationships [14] but fall short when it comes to extracting literal values. These values are the crucial information in the documents, since they often appear 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 aforementioned contribution by allowing for more flexibility and reduce the barrier in using ontologies. Document Semantic Annotation Tool (DSAT) enables users to upload their own ontology for the automatic extraction process making the complete process domain-independent. To enhance the user experience, the uploaded ontology can be previewed in a node-link representation along with its metadata. Apart from 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 as well. These improvements are crucial to the automatic extraction process from technical documents, since they will mitigate the human error in misinterpreting data.

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 from PDF files, has been widely discussed and is publicly available as reviewed in [16]. *Camelot* [6] is an open source software tool to extract tabular data from PDF files which can be executed locally. *PDFminer* [17] is an open-source and actively maintained PDF parser library in Python, which offers text, images and tables extraction with customizable parameters. For some documents, the textual content can not be extracted directly. In such cases, Optical Character Recognition (OCR) tools like *OCR Tesseract* [20] can be applied to mitigate this issue. However, most of the existing tools focus on either text or tables, and, to the extent of our knowledge, there is no unified solution that tackles both of these information sources. To achieve the best result, we use a combination of the aforementioned techniques by using *PDFminer* and *OCR Tesseract* to extract text, then *Camelot* to extract tables.

In addition to the information extraction from text and tables, images that contain data plots are equally important. *VizExtract* [8] and *ChartOCR* [15] 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. Nevertheless, *VizExtract* focuses on the conclusion of the information, i.e. if a data line is increasing, decreasing or neutral, not data point values.

Entity Recognition To detect important keywords from the text extracted, we use entity recognition tools. *AWS Textract* offers a pay-as-you-use tool to automatically extract key-value pairs from forms and tables in document images as exemplified in [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 [13]. DocStruct [22] 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, we can combine ML and other recent techniques with existing domain knowledge approaches like Ontology-Based Information Extraction (OBIE).

Ontology-Based Information Extraction (OBIE) Baclawski et al. [4] summarize the current trends that combine ML, information extraction, and ontology techniques to solve complex problems. Here, unstructured or semi-structured text is processed using ontologies to extract information. The applications of OBIE are found useful in many specific domains such as medicine [12], engineering [19], and the legal domain [5].

Ontology Visualization The ontology can also be improved along the extraction process by engaging users to choose, review, and edit ontologies, even if they are not ontology experts. Various ontology visualization tools and methods are reviewed in [2] and [10]. The common and simple-to-understand implementations are treemaps, indented lists, and node-link visualizations. Node-link visualizations can be found in most of the reviewed methods and have several styles, such as UML-blocks, trees, force-directed (spring embedded), radial, circle layouts and Euler diagrams.

3 System Overview

The OntoHuman toolchain, as shown in Figure 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 an ontology which describes the concept and properties of such products [7].

DSAT is a standalone tool assisting users to manually or automatically annotate data. Users can then trigger an automatic extraction process via an API, Continuously Trained Ontologies (ConTrOn). 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 and and changes need to undergo

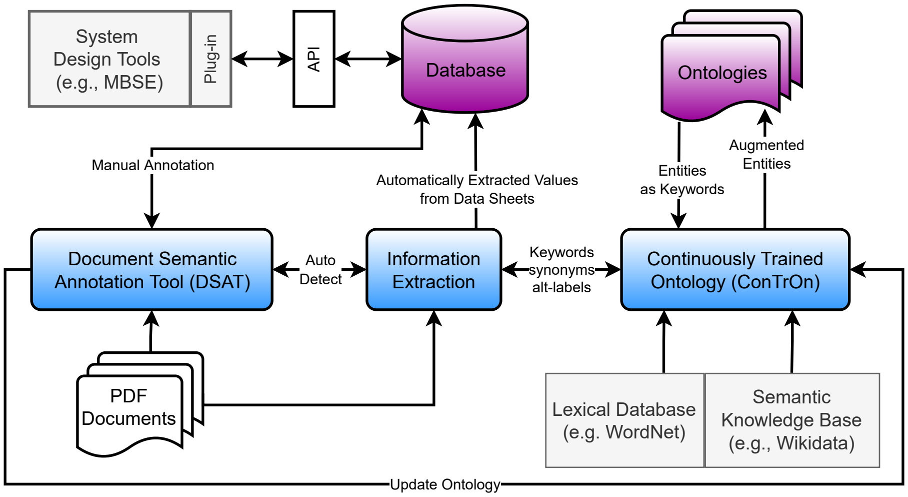
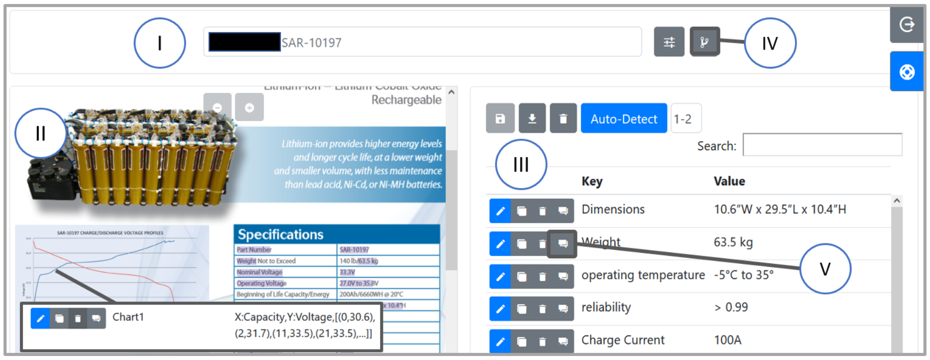


Fig. 1. System architecture of OntoHuman. Its main components are DSAT, the information extraction, and ConTrOn.

another curation step before being applied. Finally, the extracted data can be saved to a data hub which further enables the external components, such as MBSE tools 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 ($\{>, <, \leq, \geq, \sim\}$ $value$). In a first step of the information extraction workflow, text is extracted from the PDF files. Here, it 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.


I
II
III
IV
V
Lithium-Ion — Lithium Cobalt Oxide
Rechargeable
Lithium-ion provides higher energy levels and longer cycle life, at a lower weight and smaller volume, with less maintenance than lead acid, Ni-Cd, or Ni-MH batteries.
SAR-10197 CHARGE/DISCHARGE VOLTAGE PROFILES
Auto-Detect
1-2

[redacted] SAR-10197

Specifications

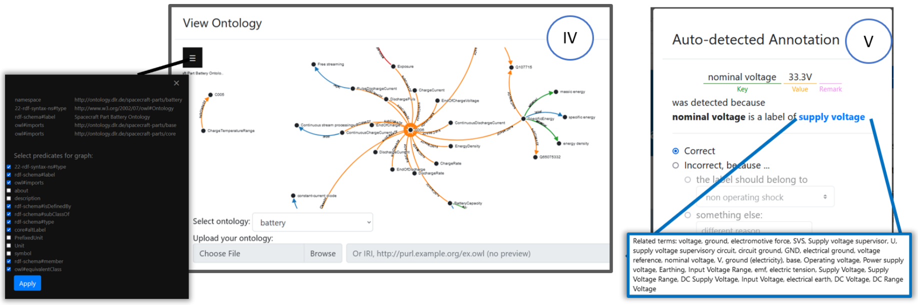
Part Number	SAR-10197
Weight Not to Exceed	140 lb/63.5 kg
Nominal Voltage	33.3V
Operating Voltage	27.0V to 35.8V
Beginning of Life Capacity/Energy	200Ah/6660WH @ 20°C

Chart1	X:Capacity,Y:Voltage,[(0,30.6),(2,31.7),(11,33.5),(21,33.5),...]]

Search:

Key	Value
Dimensions	10.6"W x 29.5"L x 10.4"H
Weight	63.5 kg
operating temperature	-5°C to 35°
reliability	> 0.99
Charge Current	100A

Fig. 2. DSAT interface allows either manual or automatic annotation (ontology-based) of a document.



namespace	http://ontology.dlr.de/spacecraft-parts/battery
22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Ontology
rdf-schema#label	Spacecraft Part Battery Ontology
owl#imports	http://ontology.dlr.de/spacecraft-parts/base
owl#imports	http://ontology.dlr.de/spacecraft-parts/core

Select predicates for graph:

☑ 22-rdf-syntax-ns#type
☑ rdf-schema#label
☑ owl#imports
□ about
□ description
☑ rdf-schema#isDefinedBy
☑ rdf-schema#subClassOf
☑ rdf-schema#type
☑ core#altLabel
□ PrefixedUnit
□ Unit
□ symbol
☑ rdf-schema#member
☑ owl#equivalentClass

Apply

Select ontology: battery

Upload your ontology: Choose File Browse Or IRI, http://purl.example.org/ex.owl (no preview)

nominal voltage 33.3V

was detected because
nominal voltage is a label of **supply voltage**

● Correct
○ Incorrect, because ...
○ the label should belong to
non operating shock
○ something else:
different reason

Related terms: voltage, ground, electromotive force, SVS, Supply voltage supervisor, U, supply voltage supervisory circuit, circuit ground, GND, electrical ground, voltage reference, nominal voltage, V, ground (electricity), base, Operating voltage, Power supply voltage, Earthing, Input Voltage Range, emf, electric tension, Supply Voltage, Supply Voltage Range, DC Supply Voltage, Input Voltage, electrical earth, DC Voltage, DC Range Voltage

Fig. 3. Left: DSAT's ontology management interface. Right: DSAT's annotation feedback for correcting an ontology.

4 Demo

DSAT's User Interface (UI) (see Figure 2) has three main sections: (I) Document and Ontology Selection View, (II) Document Preview (PDFView), and (III) Document Annotations View. Users can select or upload documents to annotate (I) via a modal dialog. The document's metadata can be updated to define the domain of the context. This domain of context is then used for selecting a suitable ontology for the automatic extraction process (if needed). 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 in Figure 3 (left). The metadata of the ontology is also summarized in the side-panel, which is initially hidden and can be expanded from the left side of the screen.

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 as displayed in Figure 2-bottom-left can be extracted as arrays of data points with x- and y-axis labels.

If an attribute (key) is incorrectly identified by the system, users can suggest the correct description, or even suggest a new description via a feedback modal (V) (see Figure 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 [11], since the irrelevant keywords will be removed and the unknown keywords will be added to the ontologies.

5 Conclusion & 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 improvement of underlying ontologies. Users who are familiar with ontologies can upload their own ontologies. Both, uploaded or predefined ontologies, can be viewed in a node-link diagram where users can pan, drag, search, and zoom to explore ontologies in more detail. Though the current implementation is only previewing, we plan to enable editing on the node-link diagram directly, so users can provide feedback to update 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 only in a prototypical state and will be subject to further development. 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 tool, so that it can be reused and improved independently.

References

1. Adnan, K., and Akbar, R. Limitations of information extraction methods and techniques for heterogeneous unstructured big data. *International Journal of Engineering Business Management 11* (2019), 1847979019890771.
2. Anikin, A., Litovkin, D., Kultsova, M., Sarkisova, E., and Petrova, T. Ontology visualization: Approaches and software tools for visual representation of large ontologies in learning. In *Creativity in Intelligent Technologies and Data Science* (08 2017), pp. 133–149.
3. Classifying text with AWS Textract. `https://www.bakertilly.com/insights/classifying-text-with-aws-textract`, accessed April 8, 2022.
4. Baclawski, K., Bennett, M., Berg-Cross, G., Fritzsche, D. M., Schneider, T., Sharma, R., Sriram, R. D., and Westerinen, A. Ontology summit 2017 communiqué - ai, learning, reasoning and ontologies. *Applied Ontology 13* (2017), 3–18.

5. Buey, M. G., Garrido, A. L., Bobed, C., and Ilarri, S. The ais project: Boosting information extraction from legal documents by using ontologies. In *ICAART* (2016).
6. Camelot: PDF Table Extraction for Humans. `https://camelot-py.readthedocs.io/en/master/`, accessed April 8, 2022.
7. ConTrOn. Contron - spacecraft parts ontology 1.2, May 2020.
8. Decatur, D., and Krishnan, S. Vizextract: Automatic relation extraction from data visualizations. *CoRR abs/2112.03485* (2021).
9. Demo paper (citation hidden due to a blind review process).
10. Dudáš, M., Lohmann, S., Svátek, V., and Pavlov, D. Ontology visualization methods and tools: a survey of the state of the art. *The Knowledge Engineering Review 33* (2018), e10.
11. Evaluation paper (citation hidden due to a blind review process).
12. Jusoh, S., Awajan, A., and Obeid, N. The use of ontology in clinical information extraction. *Journal of Physics: Conference Series 1529*, 5 (may 2020), 052083.
13. Kaló, A. Z., and Sipos, M. L. Key-value pair searhing system via tesseract ocr and post processing. In *2021 IEEE 19th World Symposium on Applied Machine Intelligence and Informatics (SAMI)* (2021), pp. 000461–000464.
14. Konys, A. Towards knowledge handling in ontology-based information extraction systems. *Procedia Computer Science 126* (2018), 2208–2218. Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 22nd International Conference, KES-2018, Belgrade, Serbia.
15. Luo, J., Li, Z., Wang, J., and 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)* (01 2021), pp. 1916–1924.
16. How to extract data out of a pdf. `https://academy.datawrapper.de/article/135-how-to-extract-data-out-of-pdfs`, Feb. 2021.
17. PDFMiner - a python package for extracting information from PDF documents. `https://pdfminersix.readthedocs.io/en/latest/`, accessed April 8, 2022.
18. Peters, D., Fischer, P. M., Schäfer, P. M., Opasjumruskit, K., and Gerndt, A. Digital availability of product information for collaborative engineering of spacecraft. In *Cooperative Design, Visualization, and Engineering* (Cham, 2019), Y. Luo, Ed., Springer International Publishing, pp. 74–83.
19. Rizvi, S. T. R., Mercier, D., Agne, S., Erkel, S., Dengel, A., and Ahmed, S. Ontology-based information extraction from technical documents. In *Proceedings of the 10th International Conference on Agents and Artificial Intelligence* (2018), SCITEPRESS - Science and Technology Publications.
20. Tesseract Open Source OCR Engine. `https://tesseract-ocr.github.io/`, accessed April 13, 2022.
21. Vrandečić, D., and Krötzsch, M. Wikidata: A free collaborative knowledgebase. *Commun. ACM 57*, 10 (Sept. 2014), 78–85.
22. Wang, Z., Zhan, M., Liu, X., and Liang, D. Docstruct: A multimodal method to extract hierarchy structure in document for general form understanding. *ArXiv abs/2010.11685* (2020).