Towards Learning from User Feedback for Ontology-based Information Extraction

Kobkaew Opasjumruskit[1] [0000−0002−9206−6896]
, Sirko Schindler[1] [0000−0002−0964−4457], Laura Thiele[1]
, and Philipp Matthias Schäfer[1] [0000−0003−3931−6670]

German Aerospace Center, Institute of Data Science,
Mälzerstraße 3, 07745 Jena, Germany firstname.lastname@dlr.de

Abstract. Many engineering projects involve the integration of various hardware parts from different suppliers. In preparation, parts that are best suited for the project requirements have to be selected. Information on these parts' characteristics is published in so called data sheets usually only available in textual form, e.g. as PDF files. To realize the automated processing, these characteristics have to be extracted into a machine-interpretable format. Such a process requires a lot of manual intervention and is prone to errors. Domain ontologies, among other approaches, can be used to implement the automated information extraction from the data sheets. However, ontologies rely solely on the experiences and perspectives of their creators at the time of creation.

To automate the evolution of ontologies, we developed ConTrOn - Continuously Trained Ontology - that automatically extracts information from data sheets to augment an ontology created by domain experts. The evaluation results of ConTrOn show that the enriched ontology can help improve the information extraction from technical documents. Nonetheless, the extracted information should be reviewed by experts before using it in the integration process. We want to provide an intuitive way of reviewing, in which the extracted information will be highlighted on the data sheets. The experts will be able to accept, reject, or correct the extracted data via a graphical interface. This process of revision and correction can be leveraged by the system to improve itself: learning from its own mistakes and identifying common patterns to adapt in the next extraction iteration. This paper presents ideas how to use machine learning based on user feedback to improve the information extraction process.

Keywords: Ontology-based information extraction · Machine learning · Knowledge representation · Pattern recognition

1 Introduction

The emerging of Industry 4.0[1] triggered an automation process in engineering projects from development to production, sometimes including customer feedback. Such digitalized processes demand the automatic exchange of data that is machine-interpretable data. Meanwhile, component parts are described in data sheets provided in textual format, such as PDF files. This enforces engineers to manually extract the data required by engineering applications, which is not only time and energy consuming, but also error-prone. Here, automated extraction of this information can mitigate such tedious tasks and enable engineers to focus on the actual product design.

To realize a machine-interpretable description of parts' model, we represent the description as ontologies. An ontology, as defined by Noy and McGuinness [18], is a machine-interpretable definition of basic concepts in a specific domain and relations between them. Its prime use case is information sharing/exchange. Since ontologies provide formal specifications of concepts, they can be used to guide the information extraction process. However, most ontologies were created based on a human's personal experience and perspective at some point in time and thus can be biased or become outdated. Moreover, during the ontology-based information extraction from domain specific data sheets, new concepts and relations that the ontologies do not cover might appear. Hence, to represent a more complete view of the domain, ontologies constantly need to be augmented with new concepts, relations, or labels for existing concepts. These enriched ontologies now in turn improve the information extraction process and allow discovery of more information from the unstructured text.

We developed ConTrOn (Continuously Trained Ontology), a system that automatically extends ontologies with information extracted from data sheets and knowledge bases [19]. Based on classes defined in an initial ontology, ConTrOn extracts textual information from data sheets. Meanwhile, guided by ontology classes, ConTrOn retrieves semantic knowledge from external data sources, i.e. WordNet [7] and Wikidata [22], to enrich the incomplete classes. The initial ontology is then augmented with the concepts retrieved from those external knowledge bases. The process can be executed as soon as new data sheets are available to automatically enrich the ontology over time.

According to the evaluation results from our first prototype, when compared to keyword-based information extraction, ontologies provide more relevant concepts, including subclasses and superclasses, and thus increase the amount of discovered information. Nevertheless, the automatically extracted information from our approach still requires human revision before archiving into a database. During the review process, a human can identify mistakes and correct them. Patterns of mistakes and corrections can then be analyzed using Natural Langauage Processing (NLP) and Machine Learning (ML) techniques. The previous functions can form a model to improve the information extraction process further.

[1] `https://www.plattform-i40.de/I40/Navigation/EN`

In this paper, we present our vision to improve ConTrOn with ML techniques based on user feedback processes. The related work and techniques will be reviewed in the next section. In Section 3, we elaborate on ConTrOn's workflow and present an approach to improve it. Finally, the conclusion of this paper and ideas for future work are described in Section 5.

2 Related Work

In this paper, we focus on the improvement of ConTrOn using ML and NLP techniques. First, we review the existing work on Ontology-Based Information Extraction (OBIE), which is one of ConTrOn's applications. Then, we elaborate on promising approaches for learning key-value patterns from unstructured text.

2.1 Ontology-Based Information Extraction

Baclawski et al. [1] summarized the current research tracks that combine ML, information extraction, and ontologies techniques to solve complex problems, such as OBIE. OBIE, as described by Wimalasuriya and Dou [24], is a system that processes unstructured or semi-structured text to extract certain types of information guided by ontologies and present the output as instances of those ontologies. The extracted information from an OBIE system is used not only to populate and enrich ontologies, but also to improve NLP workflows.

Maynard et al. [14] described NLP techniques for ontology population using an OBIE. XONTO [20] proposed an OBIE system for semantic extraction of data from PDF documents with the guide of ontologies. In contrast, Dal and Maria [5] suggested an ontology creation method using ML and external knowledge. They extract concepts from documents using latent semantic analysis and clustering techniques. Meanwhile, properties, axioms, and restrictions are retrieved from WordNet.

Barkschat [2] proposed an OBIE workflow that exploit technical data sheets to populate ontologies using a classifier model and regular expressions. Likewise, Smart-dog [16] extracts data from data sheets of spacecraft parts to populate an ontology. It features an ontology enrichment, but relies on domain experts. Meanwhile, Rizvi et al. [21] included irrelevant terms and probably-relevant terms in their ontology so that they can calculate the confidence score of the extracted information.

2.2 Key-Value Patterns Extraction

The dominant technique for extracting key-value pairs from unstructured text is to use regular expressions. ReLIE [11] presented automatic approach of regular expressions learning based on text from web pages and emails. However, it requires a man-made regular expression to start the learning process. The full automatic regular expressions generation is addressed by Brauer et al. [3]. They

used different features, which are word level and character level features, to form regular expressions that are easily understandable and configurable by experts.

DeepDive [17] presented a knowledge-base construction system by performing deep NLP to extract entities and relationships from web pages and ontology. The extraction of entities is done using the external knowledge base, Freebase (later Wikidata). To extract relationships between two entities, an SQL script is needed. However, the extraction of entities and corresponding numeric literals is not addressed.

Chakraborty et al. [4] proposed unsupervised (graph based) and supervised (conditional random field based) algorithms for extracting key-value pairs data from advertisements. The unstructured advertising text is similar to data sheets in the way that they both lack inherent grammar or a well-defined dictionary.

Machine learning techniques have been used by many studies on text processing such as XSYSTEM [8] and a study by Wang et al. [23]. XSYSTEM extracts text pattern from structured text, i.e. text from databases. It is an automated technique for extracting text pattern by incrementally learning on different text features. Wang et al. focuses on a text classification task by using Deep Convolutional Neural Networks combining with NLP techniques.

Recently, the combination of regular expressions and machine learning approaches are studied, e.g. by Locascio et al. [12] and Luo et al. [13]. Locascio et al. use a Recurrent Neural Network to generate regular expressions from text. They also generate synthetic descriptions for the generated regular expressions. However, the descriptions still requires human effort to rephase them into more natural descriptions. Luo et al. cope with the question-answering task by using regular expressions combined with neural networks. They did not specify the source of regular expressions, but their application is used to extract key-value pairs from unstructured text.

Another method to extract key-value patterns is to use Entity Matching (EM). EM takes two collections of text as inputs, then matches the entities that refer to a similar concept, e.g. “Big Apple” and “New York”. Mudgal et al. [15] presented Deep Learning (DL) solutions for EM. Their results show that DL solutions outperform state-of-the-art learning-based EM solutions like Magellan [9] on textual data at the cost of training time. Although DL solutions became popular recently, they still depend on human supervision, at least in the training phase, as Doan et al. [6] pointed out in their report.

3 ConTrOn Overview

ConTrOn offers a solution to extract information from data sheets guided by ontologies. In the process, the used ontologies are continuously enriched with information from external semantic knowledge bases, thus adapting the foundation of the extraction process to unforeseen terminologies. Figure 1 gives an overview of ConTrOn’s architecture. The remainder of this section will give an overview of its modules and their relations.

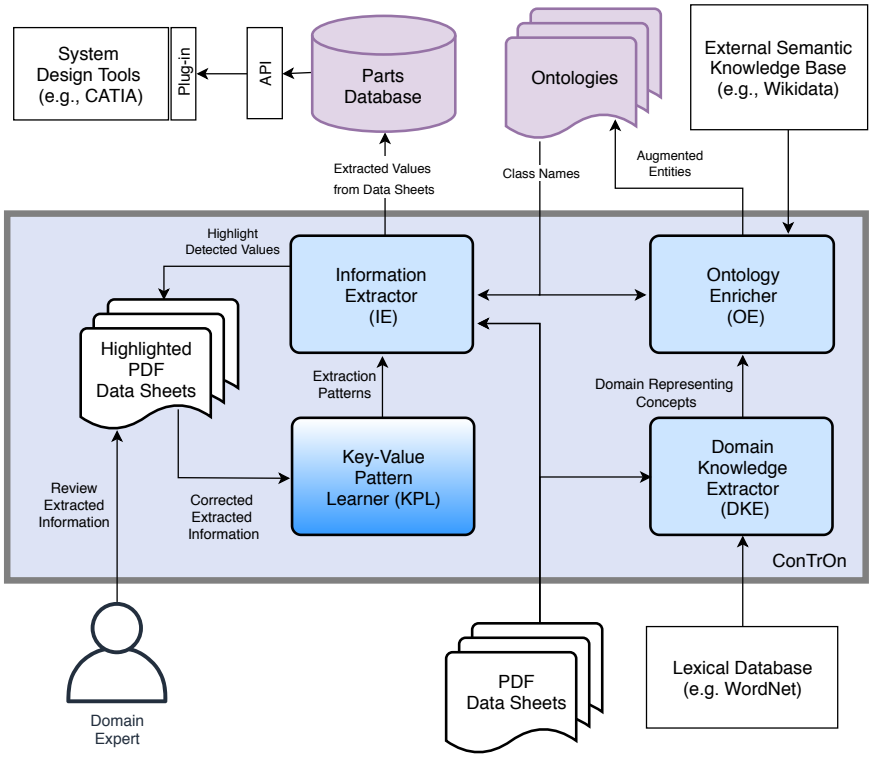

Fig. 1. ConTrOn Architecture.

Domain Knowledge Extractor (DKE). The DKE extracts all terms from all data sheets that might represent concepts and ranks them according to their TF-IDF[2] score. Subsequently, the terms are mapped to concepts whenever possible employing WordNet [7] for disambiguation. Finally, high-ranked concepts are considered *domain representing concepts* and are returned alongside their WordNet definitions.

Ontology Enricher (OE). Classes in the ontologies may lack a description, relations to other concepts, and alternative names. The OE retrieves the missing information from external, semantic knowledge bases like Wikidata. For this, it will match entities from the local ontologies to their counterparts in those knowledge bases.

If multiple candidate entities are found, their descriptions, including the terms extracted by DKE, are represented using Doc2Vec [10] algorithm. Using a Vector Space Model (VSM) and cosine similarity, the OE will now pick the most similar candidate to a vector that represents the terms extracted by DKE as a match.

[2] Term Frequency-Inverse Document Frequency

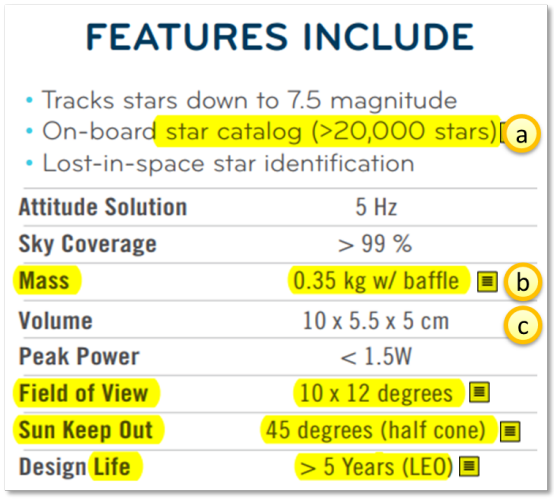

Fig. 2. Extract of a processed data sheet. Fragment includes(a) incorrectly detected data, (b) correctly identified and (c) undetected information.

If no matching entity is found, OE retrieves synonyms and relevant terms of the original terms from WordNet. These new terms are then used to retrieve a new set of candidates from Wikidata and repeat the entity selection process.

Information Extractor (IE). Using labels, alternative labels and synonyms obtained from the DKE and OE as keys, the IE scans the data sheets for associated values. Here, the assumption is that a value is most likely preceded by the respective term such as "temperature 40°C" or "Output data: MIL1553B". If no value can be found for a term this way, sentence or list patterns are applied to widen the search scope.

After the scan, all discovered terms and values are highlighted within the data sheet and are annotated with a reason for the highlighting like "The highlighted text (Life span: 5 Years) is corresponding to the *Lifetime* property".

This base system consisting of DKE, OE, and IE was previously implemented, integrated, and evaluated in [19]. The proposed addition of a Key-Value Pattern Learner (KPL) will be described in the following section.

4 Key-Value Pattern Learner (KPL)

Based on the evaluation result of the aforementioned modules, the IE process can be improved further if we involve domain experts in providing feedback on the extracted concepts and their values. These experts are presented with data sheets including the highlighted pieces of extracted information as shown in Figure 2. They are then able to accept, reject, or edit each occurrence individually.

Consider the example of an annotated data sheet in Figure 2(a). Here, ConTrOn identified the phrase "$(> 20,000)$" as the value for the term "star catalog".

As this is incorrect, reviewers can intervene in one of several ways: The annotation can be removed, or the annotation can be replaced by a fixed value like the string "available" or a boolean "true". Furthermore, there is the option to preserve the original value phrase as a remark to this entry.

Some manufacturers also use different terms for an entity, such as a property "Mass" in Figure 2(b) is sometimes mentioned as "Weight". In a domain of space system, these two terms differ due to the gravitational field. However, we can use ML techniques to solve the entity linking problem as suggested by Mudgal et al. [15].

In Figure 2(c) ConTrOn missed highlighting a fact. Reviewers can now manually add this entry by highlighting the respective phrases and annotate them with the corresponding concepts.

If reviewers adjust the extracted information in any way, then the Key-Value Pattern Learner (KPL) will analyze the change. Rejected or edited entries are passed through a part-of-speech (POS) tagger to identify a syntactic pattern. For the example of the rejected phrase "$(> 20,000)$" this would return a pattern of *bracket+symbol+number+ noun+bracket*. This can be interpreted as: 1) text that is surrounded by brackets should be considered as a remark rather than a value, and 2) a value for a keyword "star catalog" should not have a tag that contains *number+noun*. Both interpretations will be translated into two regular expressions, which will be fed into IE. The results obtained from IE will then be used to decide on which regular expression, or the combination of both, yields the most accurate result.

Similarly to the example of missed information in Figure 2(c), the KPL would learn the new term "Volume" and the pattern of its value. For the value the POS tagger identifies a pattern of *number+"x"+number+"x"+number+noun*. An entity recognizer will then extract the unit of measurement ("cm"), while a regular expression generator translates the POS pattern into a regular expression like `[\d]+.?[\d]*\sx[\d]+.?[\d]*\sx\s[\d]+.?[\d]*\s cm`. The learned patterns will be used by IE to search for terms and their values.

However, such patterns cannot be generated based on only one data sheet. We aim to train a model that takes similar key-value pairs over multiple data sheets as input and is able to generate similar regular expressions that the system did not encounter so far. These generated expressions are then applied to the existing corpus to validate them and extract further knowledge. Again, the extracted key-value pairs resulting from these automatically generated patterns have to be validated by a human expert following the general workflow as presented in Section 3.

5 Conclusion

In this paper, we presented our vision to automatically improve the information extraction from data sheets by learning from user feedback. We discussed the additions needed for ConTrOn, a system to semi-automatically build a knowledge base for engineering parts from parsing data sheets with the help of domain on-

tologies. In an ever changing field ConTrOn continuously adapts these ontologies based on user feedback by using external knowledge bases. Until a completely automated yet sufficiently robust workflow is reached, we have to rely on expert users to review the extraction results. Using both NLP and ML techniques these reviews themselves can be used to learn from past mistakes and over time improve the extraction process.

Our next step is to implement and evaluate the Key-Value Pattern Learner module within the ConTrOn workflow. We expect this self-improving process to decrease the number of extraction errors and thus lower the reviewing efforts needed. Although our approach is created as a part of ConTrOn, the basic ideas are domain-independent and can therefore be re-used in other applications that require automatic information extraction from unstructured text.

References

1. 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.
2. Barkschat, K. Semantic information extraction on domain specific data sheets. In *ESWC* (2014).
3. Brauer, F., Rieger, R., Mocan, A., and Barczynski, W. M. Enabling information extraction by inference of regular expressions from sample entities. In *CIKM* (2011).
4. Chakraborty, S., Subramanian, L., and Nyarko, Y. Extraction of (key, value) pairs from unstructured ads. In *AAAI Fall Symposia* (2014).
5. Dal, A., and Maria, J. Simple method for ontology automatic extraction from documents. *International Journal of Advanced Computer Science and Applications 3*, 12 (2012).
6. Doan, A., Ardalan, A., Ballard, J. R., Das, S., Govind, Y., Konda, P., Li, H., Mudgal, S., Paulson, E., PaulSuganthanG., C. S. G., and Zhang, H. Human-in-the-loop challenges for entity matching: A midterm report. In *HILDA@SIGMOD* (2017).
7. Fellbaum, C. Wordnet : an electronic lexical database. vol. 76, JSTOR, p. 706. Available at `https://doi.org/10.2307/417141`.
8. Ilyas, A., da Trindade, J. M. F., Fernandez, R. C., and Madden, S. Extracting syntactic patterns from databases. *CoRR abs/1710.11528* (2017).
9. Konda, P., Das, S., PaulSuganthanG., C. S. G., Doan, A., Ardalan, A., Ballard, J. R., Li, H., Panahi, F., Zhang, H., Naughton, J. F., Prasad, S., Krishnan, G., Deep, R., and Raghavendra, V. Magellan: Toward building entity matching management systems. *PVLDB 9* (2016), 1197–1208.
10. Le, Q. V., and Mikolov, T. Distributed representations of sentences and documents. *CoRR abs/1405.4053* (2014).
11. Li, Y., Krishnamurthy, R., Raghavan, S., Vaithyanathan, S., and Jagadish, H. V. Regular expression learning for information extraction. In *EMNLP* (2008).
12. Locascio, N., Narasimhan, K., DeLeon, E., Kushman, N., and Barzilay, R. Neural generation of regular expressions from natural language with minimal domain knowledge. In *EMNLP* (2016).

13. Luo, B., Feng, Y., Wang, Z., Huang, S., Yan, R., and Zhao, D. Marrying up regular expressions with neural networks: A case study for spoken language understanding. In *ACL* (2018).
14. Maynard, D., Li, Y., and Peters, W. Nlp techniques for term extraction and ontology population. In *Ontology Learning and Population* (2008).
15. Mudgal, S., Li, H., Rekatsinas, T. I., Doan, A., Park, Y., Krishnan, G., Deep, R., Arcaute, E., and Raghavendra, V. Deep learning for entity matching: A design space exploration. In *SIGMOD Conference* (2018).
16. Murdaca, F., Berquand, A., Kumar, K., Riccardi, A., Soares, T., Gerené, S., and Brauer, N. Knowledge-based information extraction from datasheets of space parts. In *8th International Systems & Concurrent Engineering for Space Applications Conference* (September 2018).
17. Niu, F., Zhang, C., Ré, C., and Shavlik, J. W. Deepdive: Web-scale knowledge-base construction using statistical learning and inference. In *VLDS* (2012).
18. Noy, N. F., and McGuinness, D. L. Ontology development 101: A guide to creating your first ontology. Tech. rep., March 2001.
19. Opasjumruskit, K., Peters, D., and Schindler, S. Contron: Continuously trained ontology based on technical data sheets and wikidata. Available at `http://arxiv.org/pdf/1906.06752`.
20. Oro, E., and Ruffolo, M. XONTO: An ontology-based system for semantic information extraction from PDF documents. *2008 20th IEEE International Conference on Tools with Artificial Intelligence 1* (nov 2008), 118–125.
21. 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.
22. Vrandečić, D., and Krötzsch, M. Wikidata: A free collaborative knowledge-base. *Commun. ACM 57*, 10 (Sept. 2014), 78–85.
23. Wang, S., Wang, Z., Zhang, D., and Yan, J. Combining knowledge with deep convolutional neural networks for short text classification. In *IJCAI* (2017).
24. Wimalasuriya, D. C., and Dou, D. Ontology-based information extraction: An introduction and a survey of current approaches. *Journal of Information Science 36* (2010), 306–323.