Journal of Machine Learning Research 26 (2025) 1-6 Submitted 7/24; Revised 1/25; Published 2/25

Ontolearn—A Framework for Large-scale OWL Class Expression Learning in Python

Caglar Demir CAGLAR.DEMIR@UPB.DE
Alkid Baci ALKID@CAMPUS.UNI-PADERBORN.DE
N'Dah Jean Kouagou NDAH.JEAN.KOUAGOU@UPB.DE
Leonie Nora Sieger LEONIE.SIEGER@UPB.DE
Stefan Heindorf HEINDORF@UPB.DE
Simon Bin SBIN@INFORMATIK.UNI-LEIPZIG.DE
Lukas Blübaum LUKASBL@CAMPUS.UNI-PADERBORN.DE
Alexander Bigerl ALEXANDER.BIGERL@UPB.DE
Axel-Cyrille Ngonga Ngomo AXEL.NGONGA@UPB.DE
Department of Computer Science
Paderborn University
Warburger Str. 100, 33098 Paderborn, Germany

Editor: Sebastian Schelter

Abstract

In this paper, we present Ontolearn—a framework for learning OWL class expressions over large knowledge graphs. Ontolearn contains efficient implementations of recent state-of-the-art symbolic and neuro-symbolic class expression learners including EvoLearner and DRILL. A learned OWL class expression can be used to classify instances in the knowledge graph. Furthermore, Ontolearn integrates a verbalization module based on an LLM to translate complex OWL class expressions into natural language sentences. By mapping OWL class expressions into respective SPARQL queries, Ontolearn can be easily used to operate over a remote triplestore. The source code of Ontolearn is available at `https://github.com/dice-group/Ontolearn`.

Keywords: machine learning, machine reasoning, knowledge graphs, description logics

1 Introduction

Explainability is quintessential to establishing trust in AI decisions (Rudin, 2019). It becomes particularly important when an AI algorithm relies on data from the Web—the largest and arguably most used information infrastructure available to humanity with over 5 billion users (Demir and Ngomo, 2023a). A key development over the last decade has been the increasing availability of Web data in the form of large-scale knowledge bases in RDF (Hogan et al., 2022). Although devising explainable ML approaches for Web-scale RDF knowledge graphs (KGs) is an indisputable building block of a trustworthy Web, most symbolic learners cannot operate well on large KGs having millions of triples.

©2025 Caglar Demir, Alkid Baci, N'Dah Jean Kouagou, Leonie Nora Sieger, Stefan Heindorf, Simon Bin, Lukas Blübaum, Alexander Bigerl, and Axel-Cyrille Ngonga Ngomo.

License: CC-BY 4.0, see `https://creativecommons.org/licenses/by/4.0/`. Attribution requirements are provided at `http://jmlr.org/papers/v26/24-1113.html`.

Given a knowledge base and sets of positive and negative examples, the goal of class expression learning is to learn a class expression in description logics such that the positive examples are instances of this expression and the negative examples are not (see Figure 2 for an example and Lehmann and Hitzler (2010) for a formal definition). DL-Learner (Lehmann, 2009) was regarded as the most mature system for OWL class expression learning (Bühmann et al., 2018; Sarker and Hitzler, 2019), featuring the symbolic learners ELTL, OCEL, and CELOE. However, because DL-Learner has not been actively maintained since its last release in 2021, it does not integrate the latest neuro-symbolic models.

In this paper, we present Ontolearn, an open-source Python library that facilitates OWL class expression learning over large RDF knowledge graphs. Figure 1 shows the software architecture of Ontolearn. Ontolearn

1. provides nine recent state-of-the-art symbolic, neuro-symbolic and deep learning algorithms to learn OWL class expressions along with efficient Python implementations of CELOE and OCEL from DL-Learner,
2. uses various OWL reasoners for class expression learning via Owlapy,[1]
3. verbalizes complex OWL class expressions through large-language models (LLMs).

At the time of writing, Ontolearn provides more OWL class expression learning algorithms than any other publicly available framework. Moreover, Ontolearn is a well-tested framework that comes with 156 unit and regression tests along with 95% test coverage. Ontolearn has already been downloaded more than 26,000 times. Importantly, we provide 26 example scripts along with a documentation to guide new users.[2] Ontolearn can easily be used via PyPI under the MIT license.[3]

2 Ontolearn

The **Knowledge Base** class is designed for efficient and easy access to a given OWL ontology. We support reading OWL ontologies into memory in RDF/XML, OWL/XML, and N-Triples formats. Given the challenges of loading large-scale ontologies with more than 10^8 triples into memory, data can alternatively be loaded into a triplestore like Tentris (Bigerl et al., 2020) and OWL class expressions can be mapped to SPARQL queries (Karalis et al., 2024), enabling efficient instance retrieval from the triplestore. Users can either provide the file path of an ontology or the endpoint of a triplestore when using Ontolearn.

Reasoner is an external component used to infer new knowledge from asserted axioms and to determine all instances of a given class expression. During concept learning in Ontolearn, we use OWL reasoners, e.g. Hermit (Glimm et al., 2014) and Pellet (Sirin et al., 2007)), which are implemented in Owlapy, to retrieve instances of class expressions.

The **Learning Problem** class encapsulates all relevant information related to OWL class expression learning problems. Figure 2 visualizes a supervised OWL class expression learning problem based on positive E^+ and negative E^- examples. Ontolearn provides scalable implementations of recent state-of-the-art OWL class expression learners, including

1. `https://pypi.org/project/owlapy`
2. `https://ontolearn-docs-dice-group.netlify.app/`
3. `https://pypi.org/project/ontolearn`

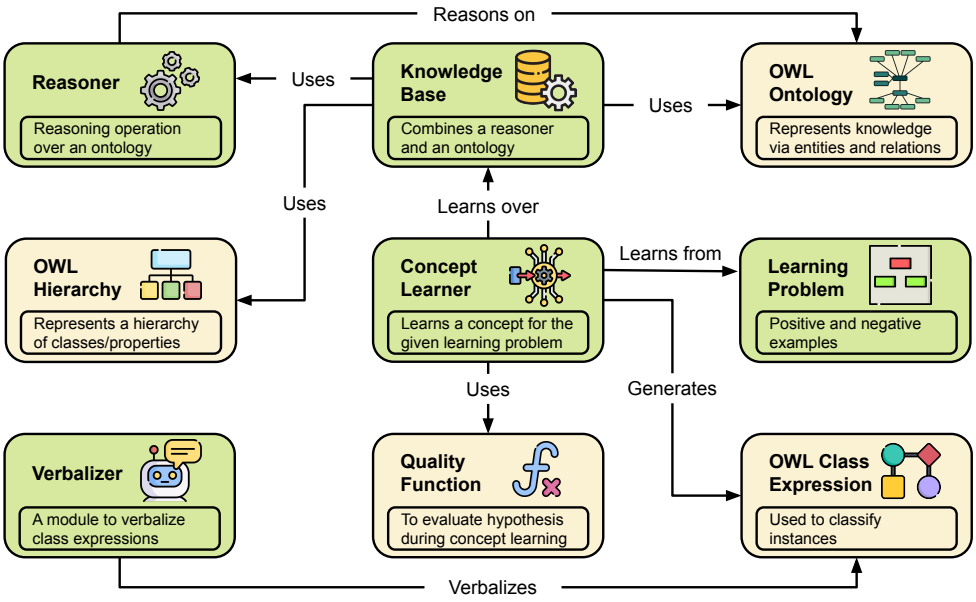

Figure 1: Ontolearn architecture. Green rectangles show the top-level components, whereas beige rectangles show the sub-components.

EvoLearner (Heindorf et al., 2022), CLIP (Kouagou et al., 2022), NCES (Kouagou et al., 2023a), NCES2 (Kouagou et al., 2023b), DRILL (Demir and Ngomo, 2023a), Nero (Demir and Ngomo, 2023b), ROCES (Kouagou et al., 2024) and efficient Python implementations of CELOE and OCEL (Lehmann, 2009). Moreover, Ontolearn includes wrappers for the DL-Learner framework, allowing access to the original implementations of the previous state-of-the-art symbolic learners CELOE and OCEL. Ontolearn also supports sampling techniques to accelerate the learning process (Baci and Heindorf, 2023), and triplestore implementations for loading and querying large-scale knowledge bases in distributed and cloud-based environments. Finally, Ontolearn offers the functionality of verbalizing class expressions into natural language using the power of LLMs like Llama (Touvron et al., 2023) or Mistral (Jiang et al., 2023).

3 Implementation

The documentation[4] gives an overview of Ontolearn's key functions and how to use them. The Ontolearn project consists of approximately 20,000 lines of code. At the time of writing, it contains 26 example scripts showcasing its main functionalities. The code's correctness is ensured by 156 test cases in Python's *unittest* framework.

Ontolearn provides all the functionalities and helper functions to easily extend it with novel concept learners. It also includes the code required to launch it as a web service.

4. `https://ontolearn-docs-dice-group.netlify.app/`

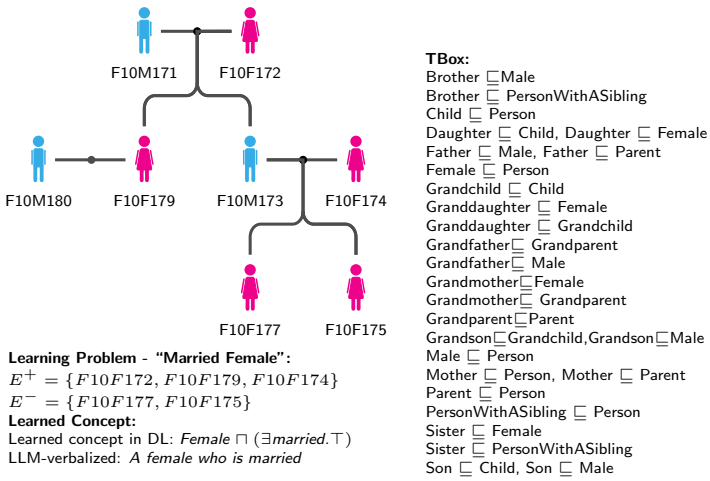

Figure 2: A partial visualization of the Family knowledge base along with a learning problem defined by E^+ and E^-. An example of a learned concept is given in DL syntax which is verbalized into natural language using an LLM.

4 Use Cases & Complementary Libraries

Ontolearn has already been applied in industrial projects, where ante-hoc explainability is required. For example, Ontolearn has been used within Industry 4.0 settings (Demir et al., 2022) to automatically learn human-interpretable descriptions of the skills of machines, which is crucial for tasks like skill matching in production processes.

EDGE (Sapkota et al., 2024) employs Ontolearn to explain graph neural networks in terms of class expressions. AutoCL (Li et al., 2024) facilitates feature selection and hyperparameter tuning for Onotlearn's concept learners. OntoSample (Baci and Heindorf, 2023) applies advanced graph sampling techniques prior to inputting the ontology into Ontolearn, enabling substantial speedups while maintaining high predictive performance. Tab2Onto (Zahera et al., 2022) allows to automatically convert tabular data to an OWL ontology, simplifying the application of Ontolearn to a wide range of industrial use cases.

Acknowledgment

This work has received funding from the European Union's Horizon 2020 research and innovation programme within the project KnowGraphs under the Marie Skłodowska-Curie grant No 860801, the European Union's Horizon Europe research and innovation programme within the project ENEXA under the grant No 101070305, and the European Union's Horizon Europe research and innovation programme within the project LEMUR under the Marie Skłodowska-Curie grant agreement No 101073307. We acknowledge the contributions of the following creators for their icons used in Figure 1: freepik, becris, gravisio, and alla-afanasenko, available at `https://www.flaticon.com`.

References

Alkid Baci and Stefan Heindorf. Accelerating concept learning via sampling. In *CIKM*, pages 3733–3737. ACM, 2023.

Alexander Bigerl, Felix Conrads, Charlotte Behning, Mohamed Ahmed Sherif, Muhammad Saleem, and Axel-Cyrille Ngonga Ngomo. Tentris - A tensor-based triple store. In *ISWC*, pages 56–73. Springer, 2020.

Lorenz Bühmann, Jens Lehmann, Patrick Westphal, and Simon Bin. Dl-learner structured machine learning on semantic web data. In *WWW (Companion Volume)*, pages 467–471. ACM, 2018.

Caglar Demir and Axel-Cyrille Ngonga Ngomo. Neuro-symbolic class expression learning. In *IJCAI*, pages 3624–3632. ijcai.org, 2023a.

Caglar Demir and Axel-Cyrille Ngonga Ngomo. Learning permutation-invariant embeddings for description logic concepts. In *IDA*, pages 103–115. Springer, 2023b.

Caglar Demir, Anna Himmelhuber, Yushan Liu, Alexander Bigerl, Diego Moussallem, and Axel-Cyrille Ngonga Ngomo. Rapid explainability for skill description learning. In *ISWC (Posters/Demos/Industry)*, volume 3254 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2022.

Birte Glimm, Ian Horrocks, Boris Motik, Giorgos Stoilos, and Zhe Wang. Hermit: An OWL 2 reasoner. *J. Autom. Reason.*, 53(3):245–269, 2014.

Stefan Heindorf, Lukas Blübaum, Nick Düsterhus, Till Werner, Varun Nandkumar Golani, Caglar Demir, and Axel-Cyrille Ngonga Ngomo. Evolearner: Learning description logics with evolutionary algorithms. In *WWW*, 2022.

Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan F. Sequeda, Steffen Staab, and Antoine Zimmermann. Knowledge graphs. *ACM Comput. Surv.*, 54(4):71:1–71:37, 2022.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. *CoRR*, abs/2310.06825, 2023.

Nikolaos Karalis, Alexander Bigerl, Caglar Demir, Liss Heidrich, and Axel-Cyrille Ngonga Ngomo. Evaluating negation with multi-way joins accelerates class expression learning. In *ECML/PKDD*, volume 14946 of *Lecture Notes in Computer Science*, pages 199–216. Springer, 2024.

N'Dah Jean Kouagou, Stefan Heindorf, Caglar Demir, and Axel-Cyrille Ngonga Ngomo. Learning concept lengths accelerates concept learning in ALC. In *ESWC*, pages 236–252. Springer, 2022.

N'Dah Jean Kouagou, Stefan Heindorf, Caglar Demir, and Axel-Cyrille Ngonga Ngomo. Neural class expression synthesis. In *ESWC*, pages 209–226. Springer, 2023a.

N'Dah Jean Kouagou, Stefan Heindorf, Caglar Demir, and Axel-Cyrille Ngonga Ngomo. Neural class expression synthesis in *ALCHIQ(D)*. In *ECML/PKDD (4)*, pages 196–212. Springer, 2023b.

N'Dah Jean Kouagou, Stefan Heindorf, Caglar Demir, and Axel-Cyrille Ngonga Ngomo. Roces: Robust class expression synthesis in description logics via iterative sampling. In *IJCAI*. ijcai.org, 2024.

Jens Lehmann. Dl-learner: Learning concepts in description logics. *J. Mach. Learn. Res.*, 10:2639–2642, 2009.

Jens Lehmann and Pascal Hitzler. Concept learning in description logics using refinement operators. *Mach. Learn.*, 78(1-2):203–250, 2010.

Jiayi Li, Sheetal Satheesh, Stefan Heindorf, Diego Moussallem, René Speck, and Axel-Cyrille Ngonga Ngomo. Autocl: Automl for concept learning. In *xAI (1)*, volume 2153 of *Communications in Computer and Information Science*, pages 117–136. Springer, 2024.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nat. Mach. Intell.*, 1(5):206–215, 2019.

Rupesh Sapkota, Dominik Köhler, and Stefan Heindorf. EDGE: evaluation framework for logical vs. subgraph explanations for node classifiers on knowledge graphs. In *CIKM*, pages 4026–4030. ACM, 2024.

Md. Kamruzzaman Sarker and Pascal Hitzler. Efficient concept induction for description logics. In *AAAI*, pages 3036–3043. AAAI Press, 2019.

Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. Pellet: A practical OWL-DL reasoner. *J. Web Semant.*, 5(2):51–53, 2007.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288, 2023.

Hamada M. Zahera, Stefan Heindorf, Stefan Balke, Jonas Haupt, Martin Voigt, Carolin Walter, Fabian Witter, and Axel-Cyrille Ngonga Ngomo. Tab2onto: Unsupervised semantification with knowledge graph embeddings. In *ESWC (Satellite Events)*, pages 47–51. Springer, 2022.