Knowledge Engineering in the Age of Neurosymbolic Systems

Marta Sabou [a,*], Majlinda Llugiqi [a], Fajar J. Ekaputra [a,b], Laura Waltersdorfer [b] and Stefani Tsaneva [a]
[a] *Institute for Data, Process and Knowledge Management, Vienna University of Economics and Business, Austria*
E-mails: marta.sabou@wu.ac.at, majlinda.llugiqi@wu.ac.at, fajar.ekaputra@wu.ac.at, stefani.tsaneva@wu.ac.at
[b] *Institute of Information Systems Engineering, TU Wien, Vienna, Austria*
E-mail: laura.waltersdorfer@tuwien.ac.at

Abstract. The field of knowledge engineering is experiencing a substantial impact from the rapid growth and widespread adoption of Neurosymbolic Systems (NeSys). In this paper, we investigate how NeSys are already used in knowledge engineering practices leading to the emergence of the new area of *neurosymbolic knowledge engineering*. To that end, we apply a data-driven analysis based on data collected in a large scale Systematic Mapping Study about systems used to create knowledge resource by employing NeSy approaches that combine Machine Learning and Semantic Web components. We characterise several aspects of this novel field, including specific approaches to knowledge engineering with NeSys identified from the data, the maturity of these systems as well as the main machine learning modules used. Additionally, we also provide concrete examples of neurosymbolic knowledge engineering systems. We conclude with an overview of research challenges such as the need for new methodologies, increased auditability, and considering the impact of human users in neurosymbolic knowledge engineering.

Keywords: Semantic Web, Knowledge Engineering, Neurosymbolic Systems

1. Introduction

Knowledge engineering (KE), broadly defined as the collection of activities for eliciting, capturing, conceptualising and formalising knowledge for the purpose of being used in information systems looks back to a long history. At the turn of the century, CommonKADS [1] proposed a methodology for knowledge engineering defined as “the development of information systems in which knowledge and reasoning play pivotal roles". Emerging research on the topic of the Semantic Web has lead to knowledge engineering methods focused primarily on creating ontologies [2] or even networks of ontologies (NeOn) [3] using mostly manual approaches. The linked data movement has highlighted the importance of (instance) data and initiated methods for creating linked datasets (e.g., the various LD life-cycle methods [4]). The focus on and availability of large-scale data continued ever since. Especially coupled with the increased popularity of machine learning models, knowledge engineering has evolved far beyond what was foreseen in the first decade of the century. So what is the next major stage in KE?

The hypothesis of this paper is that, *the advent of and recent intensified interest in neurosymbolic (NeSy) systems will represent the next major turning point in the field of KE*. Indeed, the development and application of neurosymbolic approaches is seen as one of the key trends in Artificial Intelligence (AI) research [5]. This general trend impacts

[*] Corresponding author. E-mail: marta.sabou@wu.ac.at.

on several sub-fields of AI leading to a variety of interpretations of this vision. For example, in the Semantic Web area, the community proposed techniques such as knowledge graph embeddings (KGE) and deductive reasoning [6]. Furthermore, there is a pronounced trend of building systems that combine Semantic Web and Machine Learning components (which we refer to as SWeML systems). Indeed, in a recent Systematic Mapping Study we identified nearly 500 papers reporting such systems in the period 2010-2020, with most papers being published in 2016-2020 [7].

Such intense developments, trigger the emergence of new ways of performing knowledge engineering activities by making use of these new types of neurosymbolic systems. We see this trend as the emergence of a new phase in KE namely that of *Neurosymbolic Knowledge Engineering*. For this introductory special issue of the journal on *Neurosymbolic Artificial Intelligence*, we aim to answer two main research questions:

– Is there a new field of *Neurosymbolic Knowledge Engineering* emerging? And if yes, what are its key characteristics? Our goal in answering this research question is both to provide data-driven evidence of the emergence of this field and its characteristics as well as to provide a flavour and concrete examples of neurosymbolic systems that perform KE. To that end, we analysed NeSy systems that were used in a knowledge engineering setting to produce a knowledge resource such as a taxonomy, an ontology or a knowledge graph. Given the considerable breadth of the NeSy research area, we focus our analysis on a sub-family of NeSyS, namely SWeML systems. The papers describing such systems were collected and analysed as part of the broader Systematic Mapping Study mentioned above [7] which characterised the landscape of SWeML systems (used not only for knowledge engineering purposes). Relying on the results of this study, allows deriving data-driven conclusions about this field. After a description of our methodology for collecting the data for analysis (Section 2), we present our initial, data driven findings on the characteristics of the emerging area of *Neurosymbolic Knowledge Engineering* such as typical system patterns (Section 3), the main machine learning models most often used (Section 4) and the maturity of these systems (Section 5).
– What are open challenges for the field of *Neurosymbolic Knowledge Engineering*? Based on the conclusions from the analysis of existing neurosymbolic KE systems, as well as additional considerations, we derive a number of open challenges for the *Neurosymbolic Knowledge Engineering* field (Section 6).

2. Methodology and Collected papers

Paper collection through an SMS. We base our analysis on data collected as part of a large Systematic Mapping Study (SMS) [7] which aimed to characterise SWeML systems that have been published during the 2010-2020 period. During the SMS, the main digital libraries (WebOfScience, ACM Digital Library, IEEE Xplore, Scopus[1]) were queried for those papers that, in their abstract and keywords, mention terms related to the Semantic Web (e.g., knowledge graph, linked data, semantic web, ontolog* etc,) and to Machine Learning (e.g., deep learning, neural network, embedding, representation learning, feature learning, language model etc). Additionally, as the aim was to collect papers describing concrete systems that fulfil a given task, paper abstracts also needed to mention typical application areas (e.g., Natural Language Processing, Computer Vision, Information Retrieval, Data Mining, Information integration, Knowledge management, Pattern recognition, Speech recognition). The collected 1986 papers underwent two cycles of selection in which authors systematically applied a number of selection and exclusion criteria to identify the most suitable papers. This lead to a corpus of 476 papers.

Data extraction from papers during the SMS. After reading the 476 papers, key data was extracted, related to:

1. *Bibliographic information* such as authors, their institutions, publication year and venue.
2. The *domain of application* (e.g., life sciences) and the *task solved* by the system (e.g., text analysis).
3. *System architecture* in terms of their inputs/outputs and the order of their processing units.
4. Characteristics of the *Machine Learning* modules such as the *type* (e.g., attention) and *training* (e.g., supervised).

[1] http://www.webofknowledge.com/, https://dl.acm.org/, https://ieeexplore.ieee.org/, https://www.scopus.com/

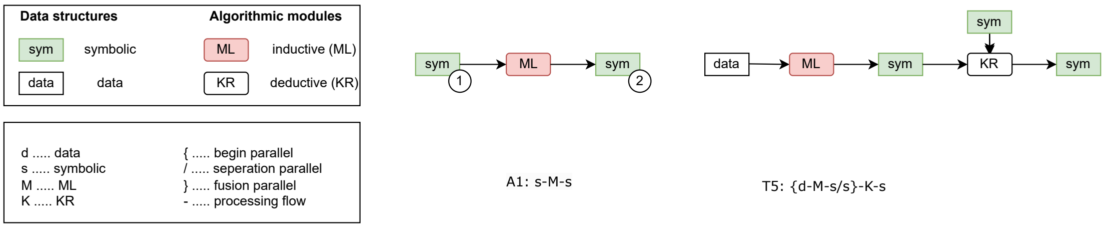


Fig. 1. Boxology-based notation of system patterns and three example patterns in graphical/textual notation.

5. Characteristics of the *Semantic Web modules* used as input to the system, such as their *type* (e.g., taxonomy, ontology, knowledge graphs), *size*, *formalisation language* etc.
6. The level of *maturity of the systems* (e.g., prototype, industrial strength application), *system transparency* in terms of sharing source code, details of infrastructure and evaluation setup as well as the existence of *provenance capturing mechanisms*.

Related to point 3 above, for describing the system architectures in terms of their internal processing flows, we used the boxology for neurosymbolic systems introduced by [8]. This boxology proposes two base elements: algorithmic modules (i.e., objects that perform some computation) that can be of type *inductive (ML)* or *deductive (KR)*, and data structures, which are the input and output to such modules that can be of *symbolic (sym)* (such as semantic entities or relations) or *non-symbolic (data)* nature (such as text, images, or embeddings). Fig. 1 depicts both the components of the boxology (left) as well as two concrete system patterns based on this boxology (right). The boxology has both a graphical notation and a corresponding textual notation which we use interchangeably in this paper. From the 15 system patterns introduced in [8], we could identify 11 patterns in use in the surveyed systems. Additionally, 33 new systems patterns were discovered.

The 44 patterns have been classified into a *pattern typology* based on their complexity. For example, simple patterns have a single processing unit. They may accept one input (*atomic* type patterns represented with **A** - see example in Fig. 3) or multiple inputs (*fusion* type patterns represented with **F** - see example in Fig. 5). More complex patterns can emerge from combining simple patterns, as follows. ***I****-Patterns* (e.g., Fig. 4) are a chain of Atomic Patterns, ***T****-patterns* (e.g., Fig. 7) are a chain of Atomic and Fusion Patterns, and ***Y****-patterns* are a combination of two (or more) Atomic Patterns via a Fusion Pattern. See [9] for a detailed description of all patterns and their classification.

KE-specific dataset selection. The data collected as part of the SMS has been released in the form of a knowledge graph [10] which can be queried through a SPARQL interface[2]. To answer this paper's research question, we use the SPARQL interface to select a subset of papers relevant to KE. Concretely, we select those papers that reported systems performing the tasks of *Graph creation* and *Graph extension* while producing a *Symbol* as the final output. Note that Graph creation and extension tasks are high-level tasks which cover more detailed general tasks such as *Ontology Creation*, *Taxonomy Creation*, as well as domain-specific tasks, e.g., *Drug Target Prediction* and *Drug Repurposing*. For this paper, we filter out domain-specific tasks and identify 123 KE-related papers (out of 476 papers in total) for the analysis described in the next sections.

3. Neurosymbolic Knowledge Engineering Patterns

The fact that over 25% (123 from 476) of all SWeML systems supports the completion of a KE task, is a strong evidence for the emergence of a new field for *Neurosymbolic Knowledge Engineering*. We start characterising this field from the perspective of the system patterns employed and by giving examples of concrete KE systems. We perform our analysis in comparison with the overall dataset to answer questions such as: Which of the general

[2]https://semantic-systems.net/sparql/

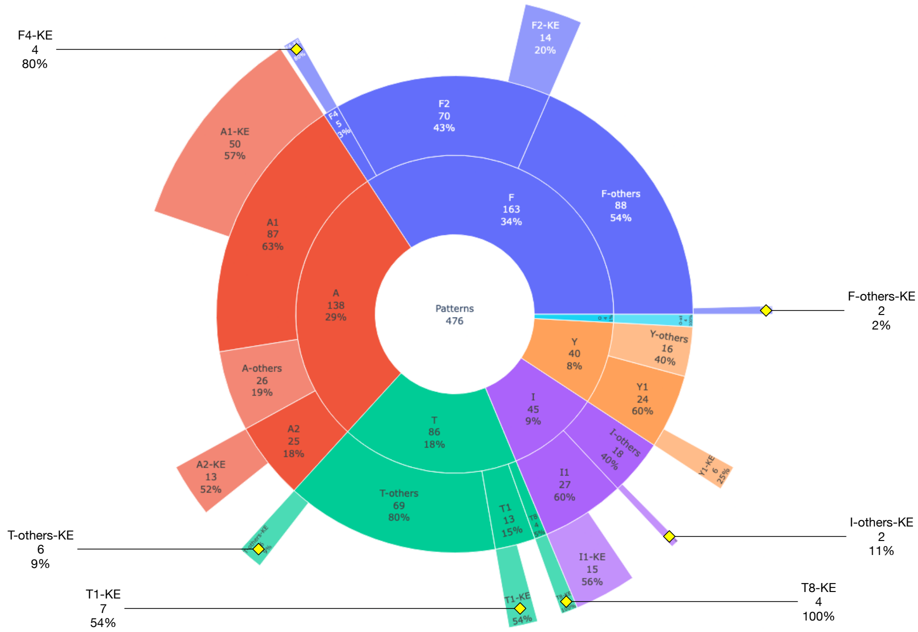


Fig. 2. Comparative pattern frequency across the overall SWeML dataset (3 inner layers) and the KE systems (outer layer).

SWeML system patterns are used for KE? What is the distribution and frequency of these KE patterns? We found that the 123 KE systems employed 18 distinct patterns from the total of 44 patterns, thus, in this dataset, less than half of the possible patterns were used for KE. Figure 2 depicts the relation between patterns used in the overall dataset and those for KE as follows:

– Layer 1: the most inner white circle, represents the 476 patterns reported by the investigated SWeML systems;
– Layer 2: presents the classification of these patterns in the 6 pattern types (A, F, I, T, Y and "Other"), showing that simple patterns of type A and F are the most frequently used in the overall dataset.
– Layer 3: depicts the frequency of concrete patterns such as A1, F2 etc. We only show the frequency of patterns that are also used for KE and group the rest of the patterns in that pattern type into a group denoted with "others". For example, among A patterns, the patterns relevant for KE are patterns A1 and A2. These are explicitly depicted while the other A patterns are shown collectively as *A-others*.
– Layer 4: depicts most patters that are also used in KE (patterns that are only used once are not depicted for the sake of visibility). Continuing the case of the A patterns, the A1 pattern is used in 50 of the KE papers while A2 is used in 13 KE papers.

Several conclusions can be drawn from this comparative visualisation. First, similarly to the overall dataset, KE systems also predominantly employ simple patters of type A and F. Second, patterns that are frequent in the overall dataset, also tend to be *frequent in the KE dataset*, in particular A1, A2, F2, and I1, which we further discuss in Section 3.1. Third, some patterns are more often used in KE systems than in other systems, thus representing patterns that are likely *specific for KE* as detailed in Section 3.2. These frequent and specific patterns are of particular interest to knowledge engineers as potential blue-prints for their work. The next sections provide more in-depth details about the various frequent/specific KE patterns as well as examples of (typical) systems that employ them. Finally, in Section 3.3 we investigate which KE tasks are solved with which patterns.

3.1. Frequent Patterns for Knowledge Engineering

In the case of the papers related to knowledge engineering tasks the most frequent patterns (each used in more than 10 papers) are A1, A2, F2, and I1. Next, we describe and exemplify the use of these patterns.

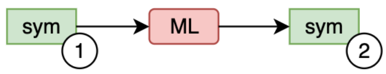


Fig. 3. Pattern A1

A1 (s-M-s), 50 papers, Fig. 3. A1 is a very simple pattern which takes symbolic input and processes it through ML to produce new symbolic data. Although the most frequent, this pattern is only used for two KE tasks: (mainly) *KG Completion* and *KG Creation*. The typical papers make use of KG embedding on a semantic structure which is then used on a downstream task such as entity typing, link prediction or ontology population. For example, in [11] authors propose an embedding model that considers both ontology and instance information from a KG. The created embedding is used for triple prediction and ontology population.

Fig. 4. Pattern I1

I1 (s1-M1-d-M2-s2), 15 papers, Fig. 4 corresponds to graph embedding approaches (M1) which embed a KG (s1) into a vector space (d) which is then further processed by a second ML component (M2) to create a symbolic structure. This pattern is almost exclusively used for KG completion tasks (e.g., for link prediction). For example, paper [12] focuses on representation learning which incorporates also attribute values as follows: attribute-value pairs (s1) are transformed into word embeddings (d) through word2vec (M1), which are then input for CNN (M2) to perform relation prediction (s2).

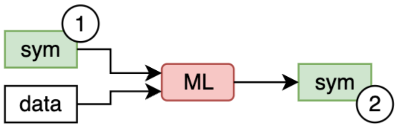


Fig. 5. Pattern F2

F2 (d/s1-M-s2), 14 papers, Fig. 5, is a pattern that is not only frequent in the overal dataset (used in 70 papers) but also in the KE dataset. For example, in [13] authors focus on classical knowledge graph embedding for supporting KG completion tasks. However, in this case authors focus on a noisy KG (s1, i.e., a KG with incorrect information) and additionally to the KG they also embed supporting textual descriptions of the KG entities (d) as a basis for computing trustiness of the KG triples.

Fig. 6. Pattern A2

A2 (d-M-s), 13 papers, Fig. 6 - The majority of papers employing A2 is focused on KG creation (7), the rest on ontology learning (4) and taxonomy creation (2). The works focus on extracting information from mostly unstructured and/or domain-specific texts. Domain-specific use cases include the cultural [14, 15], cybersecurity [16], academia [17, 18] and social media [19] domain. Some approaches are used for education purposes due to their contextualisation of implicit knowledge [14, 16, 20]. Other papers describe general approaches for documents [17, 18, 21, 22], figures [17] or relational data [23]. Almost half of the papers (6) exploit word embeddings (w2v) as their ML component.

3.2. Patterns specific for knowledge engineering

Are there patterns that are specifically used for knowledge engineering tasks? To identify such patterns, we compute the specificity of patterns as a ratio between their use in the KE dataset and the number of times they are used in the overall dataset. We identify that three of the frequent patterns are also often used in the KE dataset and can be considered specific to KE. These are: **I1** (Figure 4) for which out of 27 systems that employ this pattern 15 systems address knowledge engineering (specificity 56%), **A1** (Figure 3), with 50 systems out of a total of 92 are used for KE (specificity 54%) and **A2** (Figure 6) with 13 out of 26 uses of this pattern being for KE (specificity 54%).

Additionally to these three patterns which are both frequent and specific for KE, there are other three patterns with high specificity scores, as follows:

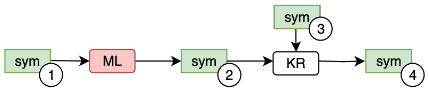


Fig. 7. Pattern T8

T8 (s1-M-s2/s3-K-s4), Fig. 7. This pattern occurs entirely in papers focusing on knowledge engineering (specificity 100%). Indeed, all four papers from the overall dataset which utilised pattern T8 were focused on knowledge completion. This pattern captures approaches where rules are learned from a (large) KG and re-applied to extend, complete, correct that KG. In paper [24], a winery related ontology is built (WineCloud) and extended in an iterative process. The initial WineCloud ontology is built based on expert interviews (s1) and is taken as input by an Association Rule Mining (M) module to deduce a set of SWRL rules (s2). The Pellet reasoner (K) is used on the initial version of the ontology to apply the derived rules and provide an extended version of the WineCloud ontology. Paper [25] focuses on knowledge graph completion through rules. Large KGs such as DBpedia, YAGO, Wikidata are inputs to an association rule mining system (M/the paper introduced AMIE+) which automatically extracts rules (s2). Rules are then applied through reasoning to derive new information and complete the KG. In paper [26] a rule induction technique is presented to mine graph patterns from large KGs and find abnormalities and missing links. The notion of rule is not a first-order logic rule, but a graph pattern that captures the expected neighbourhood around a KG.

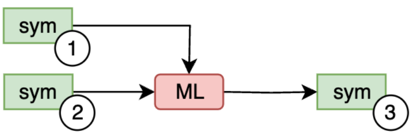


Fig. 8. Pattern F4

F4 (s1/s2-M-s3), Fig. 8. Four out of five papers using pattern F4 were centred on knowledge graph completion (specificity 80%). Three of these four papers are similar: they propose modifications to KGE methods by infusing background knowledge, in particular, in the form of logical rules [27–29]. Indeed, the authors of [27] propose the KG embedding approach TARE (Embedding knowledge graphs based on Transitivity and Asymmetry of Rules) where additionally to the KG triples (s1) also logical rules defined between relation types (s2) are employed to shape the loss function of the graph embedding model. TARE is then used for KG completion, to predict new links in the KG (s3). The second paper, [28] proposes a principled and scalable method for leveraging equivalent and inversion axioms during the learning process, by imposing a set of model-dependent soft constraints on the predicate embeddings. The approach is tested on three different KGE methods (TransE, DistMult, ComplEx) and leads to increased link prediction performance on WordNet, DBpedia and YAGO3 datasets. Finally, in paper [29], the authors propose an approach for jointly embedding knowledge graphs and logical rules. The model is evaluated on link prediction and triplet classification tasks.

Differently from papers [27–29], paper [30] focuses on extracting non-monotonic rules from a KG and associated rules. Given a KG (s1) and a set of associated Horn rules (s2), these are input to an Association Rule Mining module (M) that produces *non-monotonic rules* (s3, i.e., exception aware rules).

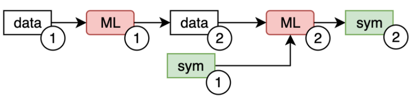


Fig. 9. Pattern T1

T1 (d-M-d/s-M-s), Fig. 9 - is a versatile pattern used for taxonomy creation [31], KG completion [32–35] and ontology extension [36, 37].

3.3. *Patterns specific for KE task types*

We continue our analysis by investigating the relation between KE patterns and the supported KE tasks. In Figure 10 we depict six KE tasks related to the creation or completion of taxonomies, ontologies and knowledge graphs as well as the pattern types employed to address these tasks. From the right side of the figure it is clear that papers focusing of tasks related to knowledge graphs are much more frequent than those focusing on tasks related to taxonomy/ontology engineering. This suggests a shift in research focus towards graph engineering, which has been less-covered by current KE methodologies.

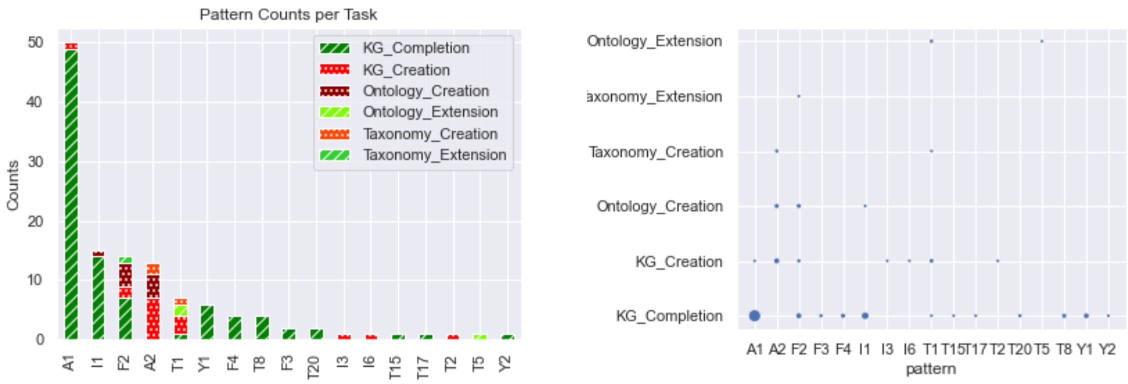


Fig. 10. Frequency of the patterns per task. Left side: number of papers reporting a certain pattern, divided by the type of task addressed. Right side: number of papers for a given KE task/pattern combination.

Specific vs. versatile patterns. We observe that some patterns are specific for certain tasks, as follows:

- Although it appears very frequently, pattern A1 is used almost exclusively for KG completion, within papers focusing on knowledge engineering. There are a number of other patterns used exclusively, in our dataset, for knowledge graph completion. These are, in the decreasing order of their frequency in the KE dataset: Y1(6, s-m-d/d-M-d-M-s), T8(4, s-m-s/s-K-s), F4 (4, s/s-M-s), F3 (2, d/s-M-d/s), T20 (2, s-M-d/s-M-s) T15(1, s-kr-s/s-M-s), T17(1, s/s-M-d-M-s), Y2 (1, s-m-d/d-M-s-M-s). As *KG Completion* encompasses several sub-tasks such as link prediction, type completion etc., future work could analyse whether some of these patters are specifically used for one of those sub-tasks.
- Patterns used exclusively for the task of knowledge graph creation are I3(1, s-M-s-M-d), I6(1, d-M-s-K-s), T2(1, d/s-M-d-M-s).

On the other hand, some patterns seam to find applicability within a range to tasks, thus being more versatile:

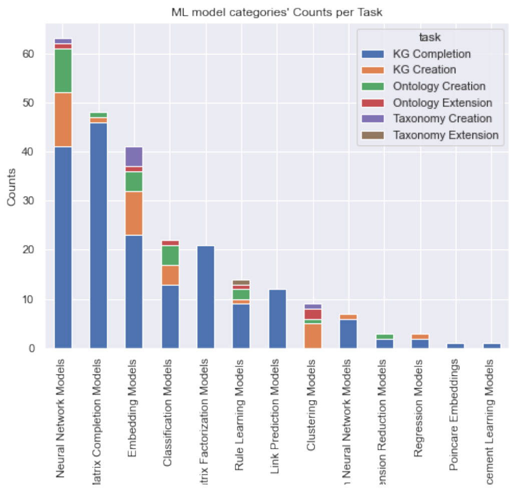
- A2 (d-M-s), appears 13 times in the dataset, and supports tasks for creating different types of knowledge structures (taxonomies, ontologies, knowledge graphs) by applying ML to a data type input.
- F2 (d/s-M-s), was used in four different task types.
- T1 (d-M-d/s-M-s) was also used in four tasks spanning all three types of knowledge structures and various activities such as completion, creation and extension.

Understanding which patterns can be used for which tasks could play an important role in guiding knowledge engineers in choosing promising system architectures for a task at hand. In particular, this would enable novice knowledge engineers of quickly identifying patterns that have emerged as adequate for certain task from teh experience of teh broader KE community.

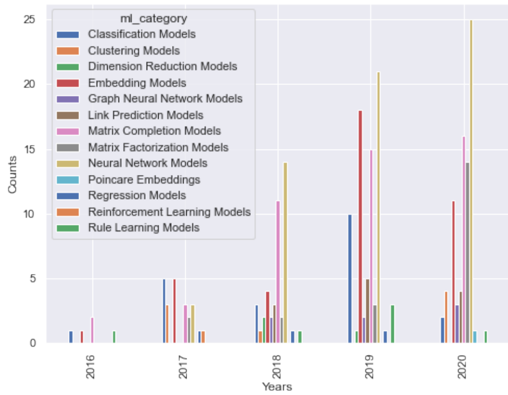
4. Machine Learning for Knowledge Engineering

How about the use of machine learning components for knowledge engineering tasks? What are the most popular ML categories (and concrete models) that should be part of the toolbox of the future knowledge engineer?

In a first analysis, we organized machine learning models into thirteen distinct categories and investigated their use for the six main KE tasks, as depicted in Figure 11a. We observe that the most frequently employed category is that of neural networks, followed by matrix completion models and embedding models, among others. Notably, certain machine learning categories such as matrix factorization, link prediction, poincare embeddings and reinforcement learning have only been utilised for knowledge graph completion purposes.



(a) Frequency of machine learning categories related to knowledge engineering tasks



(b) Frequency of the machine learning categories being used by the papers over years

The use of machine learning models in knowledge engineering has seen a significant growth in recent years. Our analysis, presented in Figure 11b, shows the most trendy machine learning categories for knowledge engineering over the years. Notably, neural networks gained significant popularity with zero papers in 2016, to 25 in 2020. Similarly, matrix completion demonstrated an upward trend. Another growth is observed in the use of embedding methods for knowledge engineering tasks. The number of papers utilizing this method increased from only one in 2016 to eleven in 2020. These findings suggest that ML models, particularly neural networks, matrix completion and embedding methods have become increasingly popular in knowledge engineering research in recent years and should become part of the toolbox of modern knowledge engineers.

5. Maturity, Transparency and Auditability

With increased use of SWeML Systems for knowledge engineering, transparency and auditability of these systems become increasingly important to better understand the quality and context of the created knowledge resources.

Maturity. In the original dataset, three levels were established to assess the maturity of the overall application: *low/probably low*, describing scripts or prototypes, *medium* systems with simple user interface or error handling or *high*, describing stable systems that are used in industrial environments. The entire subset of KE systems was assigned to be of *low/probably low* maturity, which is in line with the overall trend in the entire set of analysed SWeML Systems (over 90% being of low/probably low maturity).

Transparency. The evaluation of transparency parameters was focused on evaluation parameters and their distribution is also similar to the overall superset of SWeML Systems: Metrics were the best documented parameters (92%), followed by data (87%), parameters (76%), data-split (71%), software (29%) and infrastructure (15%). All of the transparency parameters are almost equal or lower (between 1-3%), only parameters and data-split are slightly better documented in this subset.

Auditability. There is no KE system with any additional provenance capturing, which is not surprising considering the overall low number of SWeML Systems (three systems) in the entire data set containing any provenance mechanism. However, in critical domains and with increasing amounts of heterogeneous data sources, metadata and provenance information across the entire lifecycle should/must be included (cf. EU AI Act or similar regulation).

To conclude, SWeML in general (including KE systems) are still of low maturity. Yet, we expect that SWeML Systems will continue to mature over the next years in terms of their functionalities and stability. However, despite this (expected) increase in maturity, there are still open questions in terms of the transparency and auditability of these systems which has already been identified by others. Indeed, there is a lack of mature evaluation techniques and standard benchmarks for neurosymbolic systems [38]. Furthermore, in the area of NLP, neurosymbolic systems use different datasets and benchmarks, which hampers the comparison of results [39].

6. Open Challenges for (Neurosymbolic) Knowledge Engineering

From the previous sections, we draw several data-driven conclusions about *neurosymbolic knowledge engineering*:

- *Emerging field.* The fact that over 25% of all systems collected by the Systematic Mapping Study focus on solving a task relevant for knowledge engineering demonstrates that neurosymbolic knowledge engineering is indeed an emerging field.
- *Focus on new tasks.* The more established ontology/taxonomy creation/extension tasks are less frequent in comparison with KG extension/creation tasks which are currently the key focus (Figure 10). Therefore, not only the type of systems used for KE is changing, but also the KE tasks to be achieved.
- *High variety of system patterns*. The analysis of the KE systems revealed that there are groups of systems that follow the same high-level approach, or pattern. We identified both frequent patterns and patterns that seezm to be specific for KE tasks. In total, 18 different patterns were used by the papers in our dataset. These patterns correspond to a variety or KE processes, e.g., from systems that learn a semantic structure by applying ML methods to unstructured data (A2 pattern), to systems that learn and apply rules to extend a semantic resource (T8) or systems that “infuse" background knowledge (such a logical rules) in KG embedding components (F4 pattern). Similarly to SWeML systems in general, simple patterns (A/I type) prevail. We note that the boxology notation played a key role as and instrument for organising the reviewed systems and finding similarities.
- *KE task specific patterns.* Some of the KE patterns seem to be specifically used for certain KE tasks, at least in the scope of the analysed systems. This opens the possibility for (novice) knowledge engineers to identify (and use) community-tested patterns for the task at hand.
- *Low maturity, transparency and auditability* characterises current neurosymbolic systems used for knowledge engineering (and also other tasks).

Starting from these conclusions, we see the following open challenges:

New KE methodologies and tools. The analyses performed in this paper demonstrate that we are at turning point in the KE community: not only do KE systems focus increasingly on tasks related to knowledge graphs as opposed to taxonomies/ontologies, but they also employ a variety of diverse neurosymbolic approaches (patterns). This status-quo is insufficiently covered by current KE methodologies and tools. Therefore, this area will require the development of new methodologies and tools to cater for the variety of the neurosymbolic KE approaches. The boxology-based patterns used in this paper could offer a valuable mechanism for dealing with the broad diversity of the systems. In particular, extensions to the boxology notation (e.g., in terms of representing other system module types, a richer set of relation types between system components) would be a line of work in itself and could foster an even richer analysis and methodological support for such systems. Finally, better understanding what KE tasks can be achieved with which patterns (and what are the benefits/challenges of each pattern) could provide further methodological support for knowledge engineers.

Towards auditable knowledge engineering. Semantic structures developed through the KE process underpin a variety of (often mission critical) modern intelligent systems. As such, the transparency of the process of their creation is increasingly important for several stakeholders (e.g., from a technical, managerial or legal perspective). Such transparency can be ensured by making knowledge engineering processes *auditable*. Yet, while our analysis in this paper was rather narrow due to the exploratory nature of the original data set, it suggests that there are still many gaps regarding transparency and auditability guidelines for SWeML Systems.

While auditability of AI systems in general is an active research area, current solutions fall short of the needs of neurosymbolic (including SWeML) systems that underpin neurosymbolic knowledge engineering. First, at the level of neurosymbolic systems, initial steps towards auditability have been made with design patterns and templates [7, 40, 41] which enable a common understanding of overall data and processing workflows (i.e., the boxology patterns demonstrated in this paper). These approaches are however very preliminary and still need to be adopted at scale by system engineers and practitioners to reach their full potential. Second, in the area of purely machine learning based systems, due to the deployment of systems in high-risk use cases and various incidents [42], suggestions for documentation templates of different components have emerged such as Datasheets [43], ModelCards [44], FactSheets [45] and MLOPs tools such as MLFlow[3] are supporting low-level record keeping and tracing. However, the majority of these documentation templates is still artefact-based with low semantics and the integration of provenance traces from different components is still an open question. Finally, semantic web technologies are associated with increased explainability and context, but might also include negative biases [46] or lack documentation to enable accountability [47], one of the ultimate goals of auditability. Yet, approaches for making semantic resources (and their life-cycles) audible were only weakly addressed in particular in comparison to ML systems. Therefore, exciting research opportunities lie in extending auditability notions to neurosymbolic systems by potentially extending existing work in the area of audible machine learning systems.

Clarifying the role of human agents. Knowledge engineering inherently involves human participants such as the knowledge engineer that captures and formalises knowledge or (domain) experts whose knowledge is represented. Therefore, in the changing landscape of knowledge engineering, there is a need to understand and represent the interactions between machine learning models, knowledge engineering methods and human participants in complex AI systems. However, there is still a lack of common understanding regarding the roles of humans, their necessary expertise, and their authority in such systems.

There are, nevertheless, important initial works in this direction. Concretely, in the last years, the role of human agents in neurosymbolic systems has gained attention, resulting in the introduction of two strategies to extend the collection of proposed patterns of these systems. The first approach, introduced in [41] and extended in [48], focused on the need to represent actors (agents, robots or humans) that initiate processes in neurosymbolic AI systems. Three patterns were proposed in [48] that include an actor element, visualising the roles of different actors (i.e., initiating or supporting a process) and their interactions. Additionally, a concrete use case was described exemplifying the applicability of these patterns. The second strategy, proposed in [49], aimed to extend the original boxology [8] with patterns of systems with human-in-the-loop (HiL). Two abstract HiL patterns were formalised, where the human element acts as a feedback-provider or a feedback-receiver and contributes toward the enhancement of a KR/ML component. The extended HiL patterns from [49] have already been successfully applied in describing a particular hybrid AI system involving human participation in [50]. More broadly, the need for design patterns describing the interactions between humans and AI has also been identified by the hybrid (human-AI) intelligence research community. For instance, in [51] the authors proposed design patterns for representing the collaboration between human agents and AI systems for a moral decision making domain. While the patterns focus on the interactions between the agents, the original boxology of hybrid-AI systems [8] is used to describe requirements of the AI elements. These initial works provide a basis for future work focusing on systematically analysing hybrid AI systems involving human participants in order to better understand their components and requirements.

Acknowledgements

This work was supported by the FWF HOnEst project (V 754-N). We thank the following collaborators that participated in the collection of the data used as basis for the analysis presented in this paper: Anna Breit, Andreas Ekelhart, Andreea Iana, Heiko Paulheim, Jan Portisch, Artem Revenko, Anette Ten Teije, Frank van Harmelen.

[3]https://mlflow.org

References

[1] G. Schreiber, H. Akkermans, A. Anjewierden, R. de Hoog, N. Shadbolt, W. Van de Velde and B. Wielinga, *Knowledge Engineering and Management: The CommonKADS Methodology*, MIT Press, Cambridge, MA, 2000.

[2] N.F. Noy and D.L. McGuinness, Ontology Development 101: A Guide to Creating Your First Ontology, Technical Report, 2001. http://www-ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness-abstract.html.

[3] M.C. Suárez-Figueroa, *NeOn Methodology for Building Ontology Networks: Specification, Scheduling and Reuse*, Dissertations in artificial intelligence, IOS Press, 2012. ISBN 9783898383387.

[4] M. Poveda-Villalón, A. Fernández-Izquierdo, M. Fernández-López and R. García-Castro, LOT: An industrial oriented ontology engineering framework, *Eng. Appl. Artif. Intell.* **111** (2022), 104755. doi:10.1016/j.engappai.2022.104755.

[5] H.A. Kautz, The third AI summer: AAAI Robert S. Engelmore Memorial Lecture, *AI Magazine* **43**(1) (2022), 105–125. doi:https://doi.org/10.1002/aaai.12036.

[6] P. Hitzler, F. Bianchi, M. Ebrahimi and M. Sarker, Neural-symbolic integration and the Semantic Web, *Semantic Web* **11**(1) (2020).

[7] A. Breit, L. Waltersdorfer, J.F. Ekaputra, M. Sabou, A. Ekelhart, A. Iana, H. Paulheim, J. Portisch, A. Revenko, A. Ten Teije and F. van Harmelen, Combining Machine Learning and Semantic Web -A Systematic Mapping Study, *ACM Computing Surveys* (2023).

[8] F. van Harmelen and A. ten Teije, A boxology of design patterns for hybrid learning and reasoning systems, *J. of Web Engineering* **18**(1–3) (2019), 97–124.

[9] L. Waltersdorfer, A. Breit, J.F. Ekaputra, M. Sabou, A. Ekelhart, A. Iana, H. Paulheim, J. Portisch, A. Revenko, A. Ten Teije and F. van Harmelen, Semantic Web Machine Learning Systems: An Analysis of System Patterns, in: *Compendium of Neuro-Symbolic Artificial Intelligence*, P. Hitzler, M.K. Sarker and A. Eberhart, eds, IOS Press, 2023, pp. 266–290, Chapter 10.

[10] F.J. Ekaputra, M. Llugiqi, M. Sabou, A. Ekelhart, H. Paulheim, A. Breit, A. Revenko, L. Waltersdorfer, K.E. Farfar and S. Auer, Describing and Organizing Semantic Web and Machine Learning Systems in the SWeMLS-KG (2023). https://arxiv.org/abs/2303.15113.

[11] J. Hao, M. Chen, W. Yu, Y. Sun and W. Wang, Universal Representation Learning of Knowledge Bases by Jointly Embedding Instances and Ontological Concepts, in: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 1709–1719. doi:10.1145/3292500.3330838.

[12] Z. Zhang, L. Cao, X. Chen, W. Tang, Z. Xu and Y. Meng, Representation Learning of Knowledge Graphs With Entity Attributes, *IEEE Access* **8** (2020), 7435–7441. doi:10.1109/ACCESS.2020.2963990.

[13] Y. Zhao, H. Feng and P. Gallinari, Embedding learning with triple trustiness on noisy knowledge graph, *Entropy* **21**(11) (2019). doi:10.3390/e21111083.

[14] N. Chandolikar, S. Shilaskar, D. Peddawad and S. Bhosale, Semi-Automated Ontology Building Using Deep Learning to Provide Domain-Specific Knowledge Search in the Marathi Language, in: *2019 International Conference on Applied Machine Learning (ICAML)*, 2019, pp. 108–113. doi:10.1109/ICAML48257.2019.00029.

[15] W. Buranasing and S. Phoomvuthisarn, Information extraction for cultural heritage knowledge acquisition using word vector representation, *Advances in Intelligent Systems and Computing* **772** (2019), 419–430. doi:10.1007/978-3-319-93659-8_37.

[16] Y. Deng, D. Lu, D. Huang, C.-J. Chung and F. Lin, Knowledge Graph Based Learning Guidance for Cybersecurity Hands-on Labs, in: *Proceedings of the ACM Conference on Global Computing Education*, CompEd '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 194–200. doi:10.1145/3300115.3309531.

[17] A. Roy, I. Akrotirianakis, A.V. Kannan, D. Fradkin, A. Canedo, K. Koneripalli and T. Kulahcioglu, Diag2graph: Representing Deep Learning Diagrams In Research Papers As Knowledge Graphs, in: *2020 IEEE International Conference on Image Processing (ICIP)*, 2020, pp. 2581–2585. doi:10.1109/ICIP40778.2020.9191234.

[18] D. Buscaldi, D. Dessì, E. Motta, F. Osborne and D.R. Recupero, Mining scholarly data for fine-grained knowledge graph construction, in: *CEUR Workshop Proceedings*, Vol. 2377, C.M.O.F.R.D.R.S.H. Alam M. Buscaldi D., ed., CEUR-WS, 2019, pp. 21–30.

[19] T. Xie, Y. Yang, Q. Li, X. Liu and H. Wang, Knowledge Graph Construction for Intelligent Analysis of Social Networking User Opinion, *Lecture Notes on Data Engineering and Communications Technologies* **41** (2020), 236–247. doi:10.1007/978-3-030-34986-8_17.

[20] P. Chen, Y. Lu, V.W. Zheng, X. Chen and B. Yang, KnowEdu: A System to Construct Knowledge Graph for Education, *IEEE Access* **6** (2018), 31553–31563. doi:10.1109/ACCESS.2018.2839607.

[21] M.M. Rahman and T. Finin, Understanding and representing the semantics of large structured documents, in: *CEUR Workshop Proceedings*, Vol. 2241, N.A.-C.N.S.M.A.L.E.K.J.-D.D.T.G.D. Usbeck R. Choi K.-S., ed., CEUR-WS, 2018, pp. 65–76.

[22] C. Jacinto and C. Antunes, User-driven ontology learning from structured data, in: *Proceedings - 2012 IEEE/ACIS 11th International Conference on Computer and Information Science, ICIS 2012*, Shanghai, 2012, pp. 184–189. doi:10.1109/ICIS.2012.115.

[23] M. Schlichtkrull, T.N. Kipf, P. Bloem, R. van den Berg, I. Titov and M. Welling, Modeling Relational Data with Graph Convolutional Networks, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **10843 LNCS** (2018), 593–607. doi:10.1007/978-3-319-93417-4_38.

[24] A. Mouakher, R. Belkaroui, A. Bertaux, O. Labbani, C. Hugol-Gential and C. Nicolle, An Ontology-Based Monitoring System in Vineyards of the Burgundy Region, in: *Proceedings - 2019 IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises, WETICE 2019*, Institute of Electrical and Electronics Engineers Inc., 2019, pp. 307–312. doi:10.1109/WETICE.2019.00070.

[25] L. Galárraga, C. Teflioudi, K. Hose and F.M. Suchanek, Fast rule mining in ontological knowledge bases with AMIE+, *VLDB Journal* **24**(6) (2015), 707–730. doi:10.1007/s00778-015-0394-1.

[26] C. Belth, X. Zheng, J. Vreeken and D. Koutra, What is Normal, What is Strange, and What is Missing in a Knowledge Graph: Unified Characterization via Inductive Summarization, in: *The Web Conference 2020 - Proceedings of the World Wide Web Conference, WWW 2020*, Association for Computing Machinery, Inc, 2020, pp. 1115–1126. doi:10.1145/3366423.3380189.

[27] M. Wang, E. Rong, H. Zhuo and H. Zhu, Embedding knowledge graphs based on transitivity and asymmetry of rules, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **10938 LNAI** (2018), 141–153. doi:10.1007/978-3-319-93037-4_12.

[28] P. Minervini, L. Costabello, E. Muñoz, V. Nováček and P.-Y. Vandenbussche, Regularizing Knowledge Graph Embeddings via Equivalence and Inversion Axioms, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **10534 LNAI** (2017), 668–683. doi:10.1007/978-3-319-71249-9_40.

[29] S. Guo, Q. Wang, L. Wang, B. Wang and L. Guo, Jointly embedding knowledge graphs and logical rules, in: *EMNLP 2016 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, Association for Computational Linguistics (ACL), 2016, pp. 192–202. doi:10.18653/v1/d16-1019.

[30] M.H. Gad-Elrab, D. Stepanova, J. Urbani and G. Weikum, Exception-enriched rule learning from knowledge graphs, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **9981 LNCS** (2016), 234–251. doi:10.1007/978-3-319-46523-4_15.

[31] N.R. Fauceglia, A. Gliozzo, S. Dash, M.F.M. Chowdhury and N. Mihindukulasooriya, Automatic taxonomy induction and expansion, in: *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Proceedings of System Demonstrations*, Association for Computational Linguistics (ACL), 2020, pp. 25–30.

[32] A. Pingle, A. Piplai, S. Mittal, A. Joshi, J. Holt and R. Zak, RelExt: Relation Extraction using Deep Learning approaches for Cybersecurity Knowledge Graph Improvement, in: *2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 2019, pp. 879–886. doi:10.1145/3341161.3343519.

[33] V. Pertsas, P. Constantopoulos and I. Androutsopoulos, Ontology driven extraction of research processes, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **11136 LNCS** (2018), 162–178. doi:10.1007/978-3-030-00671-6_10.

[34] Z. Xu, M. Harzallah, F. Guillet and R. Ichise, Towards a Term Clustering Framework for Modular Ontology Learning, *Communications in Computer and Information Science* **1222 CCIS** (2020), 178–201. doi:10.1007/978-3-030-49559-6_9.

[35] C. Li, X. Xian, X. Ai and Z. Cui, Representation Learning of Knowledge Graphs with Embedding Subspaces, *Scientific Programming* **2020** (2020). doi:10.1155/2020/4741963.

[36] V. Jayawardana, D. Lakmal, N. de Silva, A.S. Perera, K. Sugathadasa, B. Ayesha and M. Perera, Semi-supervised instance population of an ontology using word vector embedding, in: *17th Int. Conf. on Advances in ICT for Emerging Regions (ICTer)*, 2017, pp. 1–7.

[37] A. Shigarov, I. Cherepanov, E. Cherkashin, N. Dorodnykh, V. Khristyuk, A. Mikhailov, V. Paramonov, E. Rozhkow and A. Yurin, Towards end-to-end transformation of arbitrary tables from untagged portable documents (PDF) to linked data, in: *CEUR Workshop Proceedings*, Vol. 2463, K.D. Bychkov I.V., ed., CEUR-WS, 2019, pp. 1–12.

[38] A.d. Garcez and L.C. Lamb, Neurosymbolic AI: The 3 rd wave, *Artificial Intelligence Review* (2023), 1–20.

[39] K. Hamilton, A. Nayak, B. Božić and L. Longo, Is neuro-symbolic AI meeting its promises in natural language processing? A structured review, *Semantic Web* (2022), 1–42.

[40] F. van Harmelen and A. Ten Teije, A boxology of design patterns for hybrid learning and reasoning systems, in: *31st Benelux Conference on Artificial Intelligence and the 28th Belgian Dutch Conference on Machine Learning, BNAIC/BENELEARN*, 2019.

[41] M. van Bekkum, M. de Boer, F. van Harmelen, A. Meyer-Vitali and A.t. Teije, Modular design patterns for hybrid learning and reasoning systems: a taxonomy, patterns and use cases, *Applied Intelligence* **51**(9) (2021), 6528–6546.

[42] S. McGregor, Preventing repeated real world AI failures by cataloging incidents: The AI incident database, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

[43] T. Gebru, J. Morgenstern, B. Vecchione, J.W. Vaughan, H. Wallach, H.D. III and K. Crawford, Datasheets for Datasets, 2018.

[44] M. Mitchell, S. Wu, A. Zaldivar, P. Barnes, L. Vasserman, B. Hutchinson, E. Spitzer, I.D. Raji and T. Gebru, Model cards for model reporting, in: *Proceedings of the conference on fairness, accountability, and transparency*, 2019, pp. 220–229.

[45] M. Arnold, R.K.E. Bellamy, M. Hind, S. Houde, S. Mehta, A. Mojsilovic, R. Nair, K.N. Ramamurthy, D. Reimer, A. Olteanu, D. Piorkowski, J. Tsay and K.R. Varshney, FactSheets: Increasing Trust in AI Services through Supplier's Declarations of Conformity (2018). http://arxiv.org/abs/1808.07261.

[46] K. Janowicz, B. Yan, B. Regalia, R. Zhu and G. Mai, Debiasing Knowledge Graphs: Why Female Presidents are not like Female Popes., in: *International Semantic Web Conference (P&D/Industry/BlueSky)*, 2018.

[47] J. Andersen, S. Cazalens and P. Lamarre, Assessing Knowledge Graphs Accountability, in: *2023 Extended Semantic Web Conference (ESWC23) (Poster)*, 2023.

[48] A. Meyer-Vitali, W. Mulder and M.H. de Boer, Modular Design Patterns for Hybrid Actors, *arXiv preprint arXiv:2109.09331* (2021).

[49] H.F. Witschel, C. Pande, A. Martin, E. Laurenzi and K. Hinkelmann, Visualization of patterns for hybrid learning and reasoning with human involvement, in: *New Trends in Business Information Systems and Technology: Digital Innovation and Digital Business Transformation*, Springer, 2020, pp. 193–204.

[50] R. Prater and E. Laurenzi, A Hybrid Intelligent Approach for the Support of Higher Education Students in Literature Discovery., in: *AAAI Spring Symposium: MAKE*, 2022.

[51] J.J. van Stijna, M.A. Neerincx, A. ten Teije and S. Vethmanc, Team Design Patterns for Moral Decisions in Hybrid Intelligent Systems: A Case Study of Bias Mitigation, in: *AAAI Spring Symposium: Combining Machine Learning with Knowledge Engineering*, 2021.