

Knowledge Engineering in the Age of Neurosymbolic Systems

Neurosymbolic Artificial Intelligence
2025, Vol. 1(0) 1–16
© The Author(s) 2025
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: [10.1177/29498732251320078](https://doi.org/10.1177/29498732251320078)
journals.sagepub.com/home/nai



Marta Sabou¹, Majlinda Llugiqi¹, Fajar J Ekaputra^{1,2},
Laura Waltersdorfer² and Stefani Tsaneva¹

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 and Semantic Web 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

Received: 4 August 2023; accepted: 17 September 2024

I 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 Schreiber et al. (2000) 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 focussed primarily on creating ontologies (Noy & McGuinness, 2001) or even networks of ontologies (NeOn) Suárez-Figueroa (2012) using mostly manual approaches. The linked data (LD) movement has highlighted the importance of (instance) data and initiated methods for creating linked datasets (e.g. the various LD life-cycle methods Poveda-Villalón et al., 2022). 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?

¹Institute for Data, Process and Knowledge Management, Vienna University of Economics and Business, Vienna, Austria

²Institute of Information Systems Engineering, TU Wien, Vienna, Austria

Editor: Dagmar Gromann, University of Vienna, Austria

Solicited reviews: Alessandro Oltramari, Bosch Research, USA; Two anonymous reviewers

Corresponding Author:

Marta Sabou, Institute for Data, Process and Knowledge Management, Vienna University of Economics and Business, Vienna, Austria.

Email: marta.sabou@wu.ac.at



Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 4.0 License (<https://creativecommons.org/licenses/by/4.0/>) which permits any use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access page (<https://uk.sagepub.com/en-gb/eur/open-access-at-sage>).

The hypothesis of this article 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 (Kautz, 2022). This general trend impacts 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 (Hitzler et al., 2020). 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 recently published Systematic Mapping Study (SMS) we identified nearly 500 papers reporting such systems in the period 2010–2020, with most papers being published in 2016–2020 (Breit et al., 2023).

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 Breit et al. (2023) which characterised the landscape of SWeML systems (used not only for knowledge engineering purposes). Relying on the results of study (Breit et al., 2023), 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), the Semantic Web resources employed (Section 5) and the maturity of these systems (Section 6).
- 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 7).

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 Breit et al. (2023) 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, Scopus¹) 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 criteria, as discussed in Breit et al. (2023), to identify the most suitable papers. Inclusion criteria were publication date (2010–2020), language (English), publication type (peer re-reviewed), accessibility (accessible to authors), duplicates (latest version), whether the described systems had an interconnection between the SW and ML component, whether the system solved a concrete task and, finally, whether the paper had a sufficiently good level of English and scientific quality to be fully understood. This lead to a corpus of 476 papers. In-depth methodological information about the paper selection process is available in Breit et al. (2023) and its accompanying protocol document.

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).

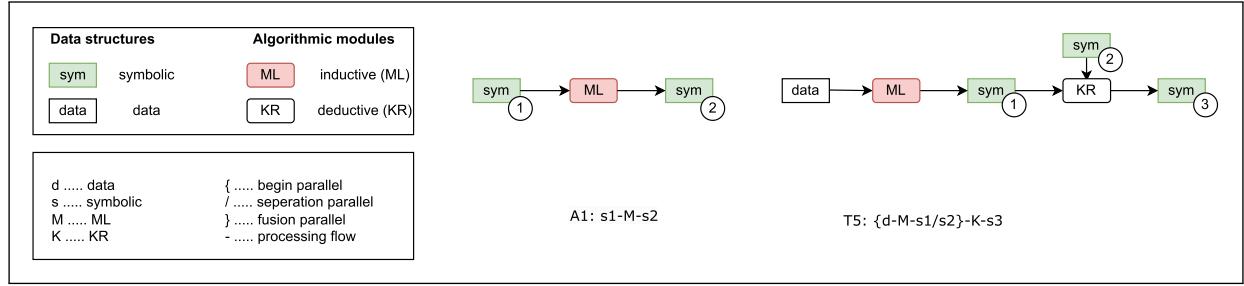


Figure 1. Boxology-based notation of system patterns and three example patterns in graphical/textual notation.

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*.

KE-specific Dataset Selection. The data collected as part of the SMS has been released in the form of a knowledge graph Ekaputra et al. (2023) which can be queried through a SPARQL interface². To answer this paper’s research questions, 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, consisting of 127 papers (out of the 476 papers in the original survey results). Note that the *Graph creation* and *Graph 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 127 KE-related papers) for the analysis described in the next sections.

3 Neurosymbolic Knowledge Engineering Patterns

System architecture was one of the key characteristics extracted during the SMS as explained in Section 2. Through the usage of system patterns to represent these architectures, we were able to present our findings in a comprehensive way and make systems comparable from a workflow and data flow perspective. In this section, we start by providing background information on the *neurosymbolic system patterns* that were identified in Breit et al. (2023), such as the notation used and the various typologies that were introduced (Section 3.1). The rest of the sections analyse the patterns that are most frequent or more specific for KE tasks providing examples of concrete systems employing these patterns.

3.1 Neurosymbolic System Patterns

Pattern Notation. To describe internal processing flows, we used the boxology for neurosymbolic systems introduced by van Harmelen and ten Teije (2019). 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). Note that the distinction between *symbolic* and *non-symbolic* representations is a reflection of “model-based” vs. “model-free” representations as explained in detail in van Harmelen and ten Teije (2019). Figure 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 van Harmelen and ten Teije (2019), we could identify 11 patterns in use in the surveyed systems. Additionally, 33 new systems patterns were discovered, thus resulting in a total of 44 known patterns.

Pattern Typology. The 44 patterns have been classified into a *pattern typology* based on their complexity see two examples, one simple (A-pattern) and a complex (T-pattern) in Figure 3). Simple patterns have a single processing unit. They may accept one input (*atomic* type patterns represented with A-) . The textual notation for A1 is (s-M-s), indicating a machine learning component that takes as input a symbolic component and produces a symbolic output. Patterns can also consist of multiple inputs as shown with (*fusion* type patterns represented with F - see example in Figure 5). More complex patterns can emerge from combining simple patterns, as follows. **I-Patterns** (e.g. Figure 4) are a chain of Atomic

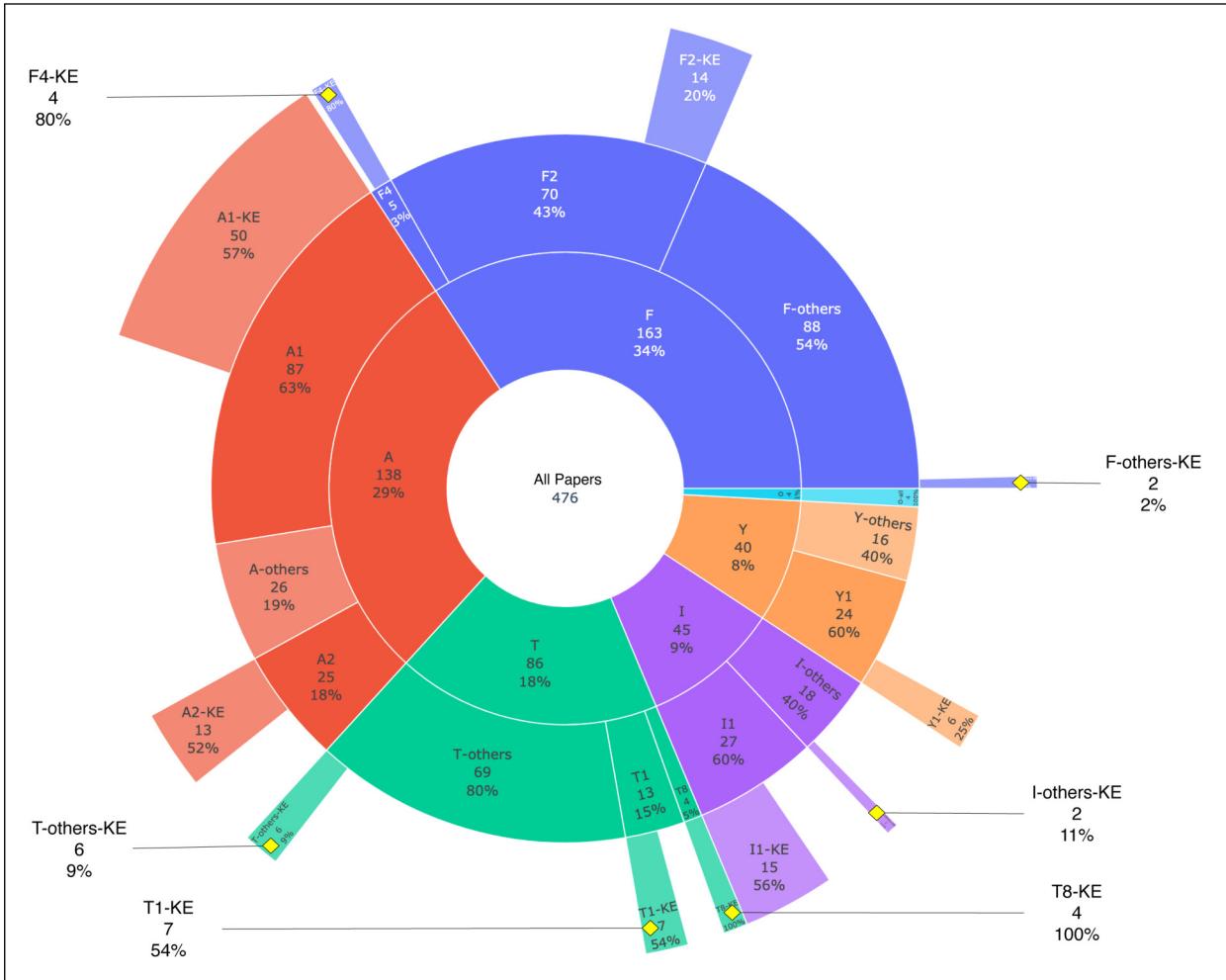


Figure 2. Comparative pattern frequency across the overall SWeML dataset (2 inner layers) and those specific to KE systems (outer layer).

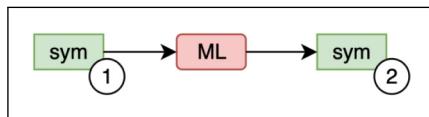
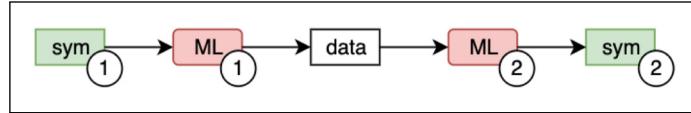
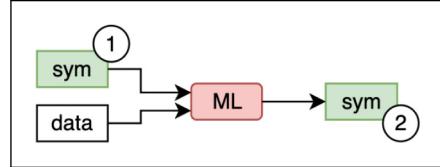
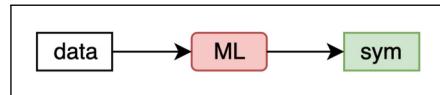
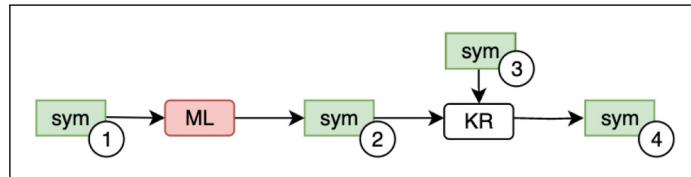


Figure 3. Pattern AI.

Patterns, **T**-patterns (e.g. Figure 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 Waltersdorfer et al. (2023) for a detailed description of all patterns and their classification.

The fact that over 25% (123 from 476) of all SWeML systems supports the completion of a KE task, as discussed in Section 2, 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 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 (of 476 papers) and those for KE (in the subset of 123 papers we selected) as follows:

**Figure 4.** Pattern II.**Figure 5.** Pattern F2.**Figure 6.** Pattern A2.**Figure 7.** Pattern T8.

- **Inner Layer 1:** This layer presents the prevalence of the six pattern types (A, F, I, T, Y and “Other”) within the 476 papers, showing that simple patterns of type A and F are the most frequently used in the overall dataset.
- **Inner Layer 2:** Depicts the frequency of concrete patterns such as A1, F2 etc. In this layer, we only show concrete patterns that are also used for KE and group the rest into a group denoted with “others.” For example, among A patterns, the concrete patterns relevant for KE are patterns A1 and A2. These are explicitly depicted while the other A patterns are shown collectively as A-others.
- **Outer Layer:** Depicts most concrete patterns that are also used in KE (concrete patterns that are only used once are not depicted for the sake of visibility). For example, the A1 pattern occurs in 50 KE papers (i.e. in 57% of all papers reporting on systems based on the A1 pattern) while A2 is used in 13 KE papers (i.e. in 52% of all papers reporting on systems based on the A2 pattern).

Several conclusions can be drawn from this comparative visualisation. First, similarly to the overall dataset, KE systems also predominantly employ simple patterns 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 3.2. 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 3.3. 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 3.4 we investigate which KE tasks are solved with which patterns.

3.2 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 (cf. Table 1 for an overview).

Table 1. Examples of Papers Describing Systems that Employ Patterns that are frequent for Knowledge Engineering.

Ref.	Title	Description	Pattern
Hao et al. (2019)	<i>Universal Representation Learning of Knowledge Bases by Jointly Embedding Instances</i>	Embedding model taking in ontology & instance information from a KG	A1 s-M-s
Zhang et al. (2020)	<i>Representation Learning of Knowledge Graphs With Entity Attributes</i>	Attribute-value pairs (s1) are transformed into word embeddings (d & M1) and then sent to a CNN (M2) to perform relation prediction (s2)	II s1-M1-d-M1-s2
Zhao et al. (2019)	<i>Embedding learning with triple trustiness on noisy knowledge graph</i>	Noisy KG (s1) and textual information (d) to compute trustworthiness of KG triples (s2)	F2 (d/s1-M-s2)
Chandolikar et al. (2019)	<i>Diag2graph: Representing Deep Learning Diagrams In Research Papers As Knowledge Graph</i>	Diagrams from papers are sent to a CNN to extract a KG-based representation of the diagrams	A2 d-M-s

A1 (s-M-s), 50 papers, Figure 3. A1 is a very simple pattern which takes symbolic input and processes it through ML to produce new symbolic output. 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 Hao et al. (2019) 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.

I1 (s1-M1-d-M2-s2), 15 papers, Figure 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 Zhang et al. (2020) focusses 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).

F2 (d/s1-M-s2), 14 papers, Figure 5, is a pattern that is not only frequent in the overall dataset (used in 70 papers) but also in the KE dataset. For example, in Zhao et al. (2019) 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 trust-levels for the KG triples.

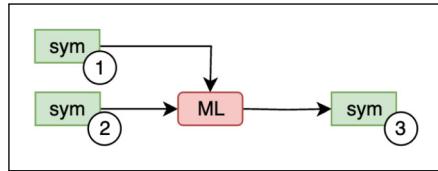
A2 (d-M-s), 13 papers, Figure 6 - The majority of papers employing A2 is focussed on KG creation (7), the rest on ontology learning (4) and taxonomy creation (2). The works focus on extracting information from mostly unstructured and/or domain-specific texts. Domain-specific use cases include the cultural (Buranasing & Phoomvuthisarn, 2019; Chandolikar et al., 2019), cybersecurity (Deng et al., 2019), academia (Buscaldi et al., 2019; Roy et al., 2020) and social media (Xie et al., 2020) domains. Some approaches are used for education purposes due to their contextualisation of implicit knowledge (Chandolikar et al., 2019; Chen et al., 2018; Deng et al., 2019). Other papers describe general approaches for documents (Buscaldi et al., 2019; Jacinto & Antunes, 2012; Rahman & Finin, 2018; Roy et al., 2020), figures (Roy et al., 2020) or relational data (Schlichtkrull et al., 2018). Almost half of the papers (6) exploit word embeddings (w2v) as their ML component.

3.3 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 (cf. Table 2 for an overview of concrete examples):

Table 2. Examples Papers Describing Systems that Rely on Patterns that are specific for Knowledge Engineering.

Ref.	Title	Description	Pattern
Galárraga et al. (2015)	<i>Fast rule mining in ontological knowledge bases with AMIE+</i>	DBpedia, YAGO, Wikidata are inputs (s1) to association rule mining system (M+ AMIE) which automatically extracts rules (s2). Rules are then applied through reasoning (K) to derive new information and complete the KG (s3 before completion, s4 after completion)	T8 s1-M-s2/s3-K-s4
Wang et al. (2018)	<i>Embedding knowledge graphs based on transitivity and asymmetry of rules</i>	KG triples (s1) and logical rules between relation types (s2) are employed to shape the loss function of the graph embedding model, TARE (M, KG embedding) is then used for KG completion, to predict new links in the KG (s3)	F4 s1/s2-M-s3
Li et al. (2020)	<i>Representation Learning of Knowledge Graphs with Embedding Subspace</i>	Word embeddings are learned, then a projection of word + node embeddings are learned to be then applied for link prediction	T1 d1-M1-d2/s1-M2-s2

**Figure 8.** Pattern F4.

T8 (s1-M-s2/s3-K-s4), Figure 7. This pattern occurs entirely in papers focussing on knowledge engineering (specificity 100%). Indeed, all four papers from the overall dataset which utilised pattern T8 were focussed on knowledge completion. This pattern captures approaches where rules are learned from a (large) KG and re-applied to extend, complete, or correct that KG. In paper Mouakher et al. (2019), a winery related ontology is built (WineCloud) and extended in an iterative process. The initial WineCloud ontology (s1) is built based on expert interviews 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 Galárraga et al. (2015) focusses 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 Belth et al. (2020) 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.

F4 (s1/s2-M-s3), Figure 8. Four out of five papers reporting 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 (Guo et al., 2016; Minervini et al., 2017; Wang et al., 2018). Indeed, the authors of Wang et al. (2018) propose the KG embedding approach TARE (Embedding knowledge graphs based on Transitivity and Asymmetry of Rules) where additionally to the KG triples (s2) also logical rules defined between relation types (s1) are employed to shape the loss function of the graph embedding model. TARE then performs KG completion by predicting new links in the KG (s3). The second paper, Minervini et al. (2017) 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 Guo et al. (2016), the authors propose an approach for jointly embedding knowledge graphs and logical rules. The model is evaluated on link prediction and triplet classification tasks.

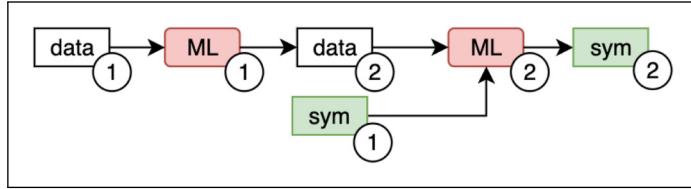


Figure 9. Pattern T1.

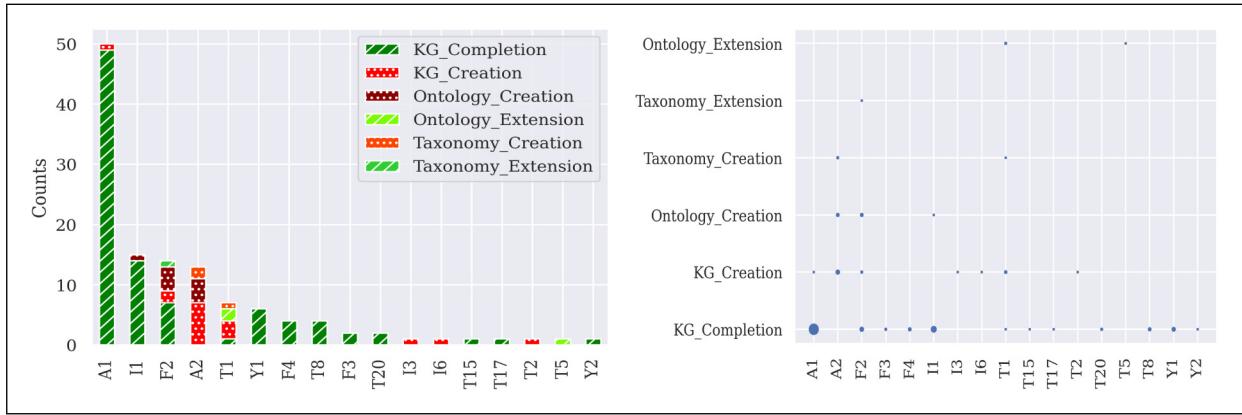


Figure 10. Frequency of the patterns per task. Left side: number of papers reporting a certain pattern, divided by the type of task addressed. Right side: number of papers for a given KE task/pattern combination.

Differently from papers (Guo et al., 2016; Minervini et al., 2017; Wang et al., 2018), paper Gad-Elrab et al. (2016) focusses 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).

T1 (d1-M1-d2/s1-M2-s2), **Figure 9** - is a versatile pattern used for taxonomy creation (Fuceglia et al., 2020), KG completion (Li et al., 2020; Pertsas et al., 2018; Pingle et al., 2019; Xu et al., 2020) and ontology extension (Jayawardana et al., 2017; Shigarov et al., 2019). For example, in Li et al. (2020), word embeddings (d2) are learned first and used in tandem with node emebddings for link prediction (s2).

3.4 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 focussing of tasks related to knowledge graphs are much more frequent than those focussing 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.

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 focussing 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, s1-M1-d1/d2-M2-d3-M3-s2), T8(4, s1-M-s2/s3-K-s4), F4 (4, s1/s2-M-s3), F3 (2, d1/s1-M-d2/s2), T20 (2, s1-M1-d1/s2-M2-s3) T15(1, s1-K-s2/s3-M-s4), T17(1, s1/s2-M1-d-M2-s3), Y2 (1, s1-M1-d1/d2-M2-s2-M3-s3). 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, s1-M1-s2-M2-d), I6(1, d-M-s1-K-s2), T2(1, d1/s1-M1-d2-M2-s2).

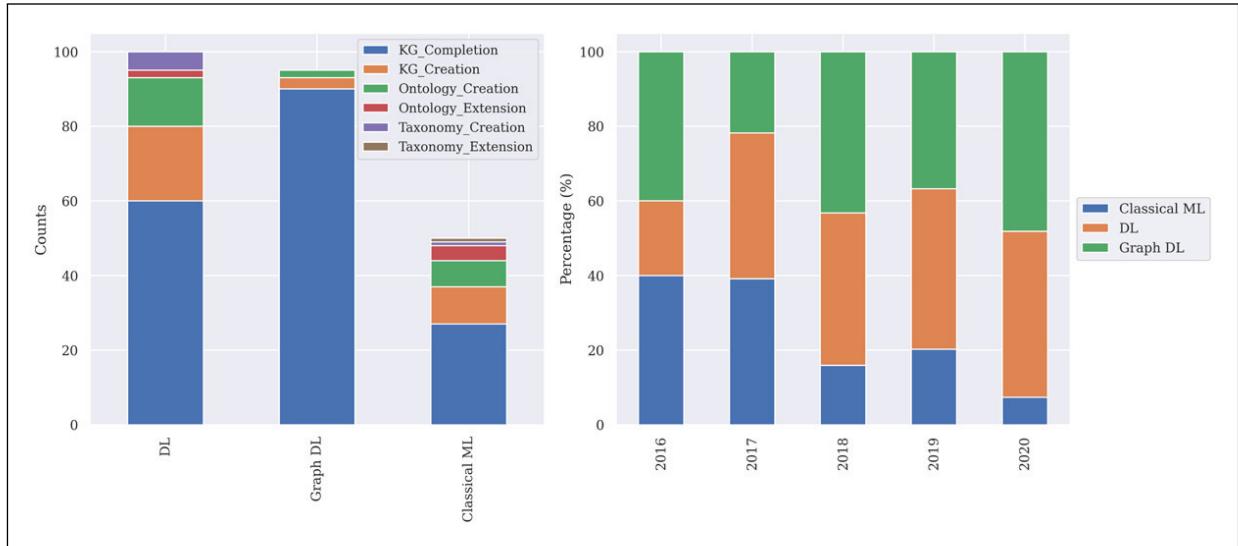


Figure 11. Frequency of machine learning models per task/year. (Left side) Frequency of ML categories related to knowledge engineering tasks. (Right side) Frequency of the ML categories being used by the papers over 5 years.

On the other hand, some patterns seem to find applicability within a range of 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/s1-M-s2), was used in four different task types.
- T1 (d1-M1-d2/s1-M2-s2) 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 to quickly identify patterns that have emerged as adequate for certain tasks from the experience of the 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 that should be part of the toolbox of the future knowledge engineer?

In our analysis of machine learning utilisation in KE tasks, we grouped machine learning models into three categories: Deep Learning (explicitly excluding Graph Deep Learning), Graph Deep Learning and Classical ML. As shown in Figure 11 (left-side), Deep Learning (DL) models are predominant, used 100 times across various tasks, asserting their adaptability and efficacy in KE. Following closely, Graph Deep Learning (Graph DL) models show notable application, especially in KG Completion, with 95 uses. Classical ML models, though less dominant with 50 instances, remain relevant in certain tasks such as KG Completion and KG Creation. This distribution underscores a trend towards more sophisticated models in KE.

The use of machine learning models in KE has seen a significant growth in recent years. Our analysis, presented in Figure 11 (right-side), shows the most trendy machine learning categories for knowledge engineering over the years 2016-2020. In the evolution of machine learning model usage from 2016 to 2020, a significant shift towards advanced models is evident. The year 2016 saw an equal preference for Classical ML and Graph DL models (40% each), with DL at 20%. However, by 2020, DL and Graph DL models had surged to 44.44% and 48.15%, respectively, while Classical ML's usage has receded to merely 7.41%. This shift reflects the increasing complexity of KE tasks that corresponds to more complex models.

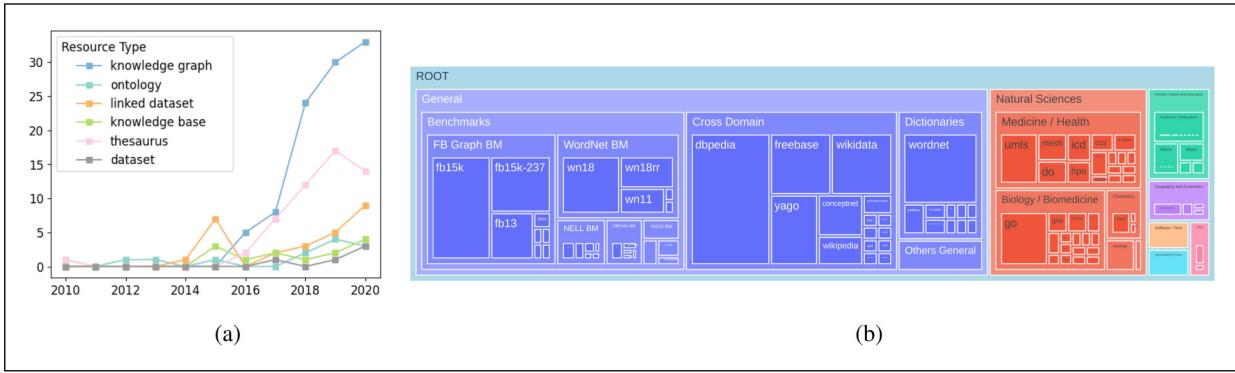


Figure 12. An overview of semantic resources used in KE-focussed SWeML systems. (a) SW resources by types and years; (b) SW resources by domains.

5 Semantic Web Resources for Knowledge Engineering

We perform an analysis of the Semantic Web resources used in SWeML systems for KE tasks based on the categories of resources introduced in our prior survey (Breit et al., 2023). There are six different types of resources found in the systems as described next.

- **Thesaurus** is a controlled vocabulary connected with relations that express linguistic relations (e.g. broader and narrower relations), without a strict logical semantics (e.g. subsumption), e.g. WordNet and ConceptNet.
- **Ontology** is a terminological model richer than a taxonomy containing also additional named relations and axioms (T-Box), e.g. SNOMED CT and UMLS ontology.
- **Dataset** contains semantic instance data (or metadata), corresponding to an A-Box in logics. A collection of triples describing instances can be considered a dataset, e.g. NELL and SUMO dataset.
- **Knowledge base** contains both terminological and instance knowledge (TBox+ABox), e.g. YAGO KB.
- **Linked dataset** contains links (in terms of URI references) to other semantic resources. Such datasets contain large numbers of instance data while may also include (lightweight) terminological knowledge as well, e.g. DBpedia and Wikidata.
- **Knowledge graph** was most recently defined as “*a graph of data intended to accumulate and convey knowledge of the real world, whose nodes represent entities of interest and whose edges represent potentially different relations between these entities*” (Hogan et al., 2021). The definition subsumes all the semantic resource type definitions above and more. Therefore, we use this category for those resources that cannot be assigned to any of the categories above.

Figure 12 provides an overview of SW resource usage on KE-focussed SWeML systems. We found that 92 out of 123 papers use SW resources to various extent. Out of these, 35 employ a single SW resource, while the rest utilise two or more resources for their tasks. We show the SW resource distribution following our categorisation, which can be seen in Figure 12(a). The figure shows rapid growth of KG and thesaurus utilisation since 2016. In contrast, the utilisation of other types of SW resources in KE-focussed SWeML systems is relatively stable. Figure 12(b) shows the classification of the SW resources according to their domain. The resources used for KE tasks typically come from the general domain, shown in dark blue shades. Domain-specific resources, such as natural sciences, are less dominant, which could be an effect of the generic nature of the KE tasks.

6 Maturity, Transparency and Auditability

With increased use of SWeML Systems for knowledge engineering, the 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 deployed 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 focussed 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% and 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 (Garcez & Lamb, 2023). Furthermore, in the area of NLP, neurosymbolic systems use different datasets and benchmarks, which hampers the comparison of results (Hamilton et al., 2022).

7 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 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 are potentially specific for KE tasks. In total, 18 different patterns were reported by the papers in our dataset. These patterns correspond to a variety of 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 as 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 of van Harmelen and ten Teije (2019) played a key role as an 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 for this research area:

New KE Methodologies and Tools. The analysis performed in this paper demonstrates 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 article 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 van Harmelen and Ten Teije (2019), van Bekkum et al. (2021) and Breit et al. (2023) which enable a common understanding of overall data and processing workflows (i.e. the boxology patterns demonstrated in this article). 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 their deployment in high-risk use cases and various incidents (McGregor, 2021), suggestions for documentation templates of different components have emerged: Datasheets (Gebru et al., 2018), ModelCards (Mitchell et al., 2019), FactSheets (Arnold et al., 2018) and MLOPs tools such as MLFlow³ 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 (Janowicz et al., 2018) or lack documentation to enable accountability (Andersen et al., 2023), one of the ultimate goals of auditability. Yet, approaches for making semantic resources (and their life-cycles) auditable 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 auditable 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 van Bekkum et al. (2021) and extended in Meyer-Vitali et al. (2021), focussed on the need to represent actors (agents, robots or humans) that initiate processes in neurosymbolic AI systems. Three patterns were proposed in Meyer-Vitali et al. (2021) 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 Witschel et al. (2020), aimed to extend the original boxology (van Harmelen & ten Teije, 2019) 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 Witschel et al. (2020) have already been successfully applied in describing a particular hybrid AI system involving human participation in Prater and Laurenzi (2022). 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 van Stijna et al. (2021) 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 van Harmelen and ten Teije (2019) is used to describe requirements of the AI elements. These initial works provide a basis for future work focussing on systematically analysing hybrid AI systems involving human participants in order to better understand their components and requirements.

Assessing the Importance of Large Language Models (LLMs) for Knowledge Engineering. When the initial SWeMLS survey took place in October 2020, research on the utilisation of LLMs for Knowledge Engineering was scarce, and present in only 4 out of 123 papers (less than 5%). With the rapid evolution and adaptation of LLMs in the last few years, however, the landscape is changing very rapidly, marked with the emergence of vision papers, e.g. Allen et al. (2023) and special tracks on scientific venues⁴ on this topic. Therefore, an investigation of LLM-based patterns for knowledge engineering is an open topic, which requires the collection of more recent data to be answered reliably.

We conclude with a set of *limitations* of this work. Firstly, the limitations of the broader study whose data we reuse (Breit et al., 2023) also affect this work. In particular, given the broad area addressed by the study in Breit et al. (2023), we needed to perform study scoping to select a reasonable number of papers, with the potential side-effect that some relevant papers were missed. Furthermore, during data analysis, new abstractions were introduced (e.g., sets of domains, types of ML models, types of SW models) which we derived from the extracted data given the lack of such typologies in the literature. Therefore, we cannot claim that these are representative for the entire field. Finally, the version of the boxology notation used was quite restrictive (van Harmelen & ten Teije, 2019) (e.g. no possibility to represent standard processing units, no distinction between training and prediction phases, representation of only input/output relations). As such, several simplifying assumptions had to be taken when representing more complex systems with the boxology thus potentially leading to the loss of some details. Additionally to these limitations, the analysis presented in this paper represents an initial study with many aspects still to be explored (e.g. how the various characteristics of SWeML systems for KE are mirrored by various application domains). Given also that the collected data dates back to 2010–2020, more recent trends, in particular related to the use of LLMs for knowledge engineering, are not reflected in the analysis at this point, but remain a topic of future work requiring the systematic collection of more recent papers, from 2020 onwards.

Despite these limitations, we remain confident that this work captures key influences that neurosymbolic systems have on the knowledge engineering area (whether powered by LLMs or not) and will fuel further development and discussions in the KE field as already apparent from early adopters of our ideas (Allen & Ilievski, 2024).

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.

Statements and Declarations

Funding

The authors received no financial support for the research, authorship and/or publication of this article.

Declaration of Conflicting Interests

The authors declared no potential conflicts of interest with respect to the research, authorship and/or publication of this article.

Notes

1. <http://www.webofknowledge.com/>, <https://dl.acm.org/>, <https://ieeexplore.ieee.org/>, <https://www.scopus.com/>
2. SPARQL interface for querying the knowledge graph based representation of the data extracted by Breit et al. (2023): <https://semantic-systems.net/sparql/>.
3. <https://mlflow.org>
4. ESWC 2024: <https://2024.eswc-conferences.org/call-for-papers-llms/>

References

- Allen, B. P., & Ilievski, F. (2024). Standardizing knowledge engineering practices with a reference architecture. *Transactions on Graph Data and Knowledge*.
- Allen, B. P., Stork, L., & Groth, P. (2023). Knowledge engineering using large language models. *Transactions on Graph Data and Knowledge*, 1(1), 3:1–3:19. <https://doi.org/10.4230/TGDK.1.1.3>
- Andersen, J., Cazalens, S., & Lamarre, P. (2023). Assessing knowledge graphs accountability. In *2023 Extended Semantic Web Conference (ESWC23) (Poster)*.
- Arnold, M., Bellamy, R. K. E., Hind, M., Houde, S., Mehta, S., Mojsilovic, A., Nair, R., Ramamurthy, K. N., Reimer, D., Olteanu, A., Piorkowski, D., Tsay, J., & Varshney, K. R. (2018). FactSheets: increasing trust in AI services through supplier’s declarations of conformity. <http://arxiv.org/abs/1808.07261>
- Belth, C., Zheng, X., Vreeken, J., & Koutra, D. (2020). 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. (pp. 1115–1126). <https://doi.org/10.1145/3366423.3380189>
- Breit, A., Waltersdorfer, L., Ekaputra, J. F., Sabou, M., Ekelhart, A., Iana, A., Paulheim, H., Portisch, J., Revenko, A., Ten Teije, A., & van Harmelen, F. (2023). Combining machine learning and semantic web -a systematic mapping study. *ACM computing surveys*.

- Buranasing, W., & Phoomvuthisarn, S. (2019). Information extraction for cultural heritage knowledge acquisition using word vector representation. *Advances in Intelligent Systems and Computing*, 772, 419–430. https://doi.org/10.1007/978-3-319-93659-8_37
- Buscaldi, D., Dessì, D., Motta, E., Osborne, F., & Recupero, D. R. (2019). 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 (pp. 21–30).
- Chandolikar, N., Shilaskar, S., Peddawad, D., & Bhosale, S. (2019). 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)* (pp. 108–113). <https://doi.org/10.1109/ICAML48257.2019.00029>
- Chen, P., Lu, Y., Zheng, V. W., Chen, X., & Yang, B. (2018). KnowEdu: a system to construct knowledge graph for education. *IEEE Access*, 6, 31553–31563. <https://doi.org/10.1109/ACCESS.2018.2839607>
- Deng, Y., Lu, D., Huang, D., Chung, C.-J., & Lin, F. (2019). 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 (pp. 194–200). <https://doi.org/10.1145/3300115.3309531>
- Ekaputra, F. J., Llugiqi, M., Sabou, M., Ekelhart, A., Paulheim, H., Breit, A., Revenko, A., Waltersdorfer, L., Farfar, K. E., & Auer, S. (2023). Describing and organizing semantic web and machine learning systems in the SWeMLS-KG. <https://arxiv.org/abs/2303.15113>
- Fauceglia, N. R., Gliozzo, A., Dash, S., Chowdhury, M. F. M., & Mihindukulasooriya, N. (2020). 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) (pp. 25–30).
- Gad-Elrab, M. H., Stepanova, D., Urbani, J., & Weikum, G. (2016). 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 (pp. 234–251). https://doi.org/10.1007/978-3-319-46523-4_15
- Galárraga, L., Teflioudi, C., Hose, K., & Suchanek, F. M. (2015). Fast rule mining in ontological knowledge bases with AMIE+. *VLDB Journal*, 24(6), 707–730. <https://doi.org/10.1007/s00778-015-0394-1>
- Garcez, A.D., & Lamb, L. C. (2023). Neurosymbolic AI: the 3 rd wave. *Artificial intelligence review* (pp. 1–20).
- Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J. W., Wallach, H., & K. Crawford, H. D. (2018). Datasheets for datasets.
- Guo, S., Wang, Q., Wang, L., Wang, B., & Guo, L. (2016). Jointly embedding knowledge graphs and logical rules. In *EMNLP 2016 - conference on empirical methods in natural language processing, Proceedings*, Association for Computational Linguistics (ACL) (pp. 192–202). <https://doi.org/10.18653/v1/d16-1019>
- Hamilton, K., Nayak, A., Božić, B., & Longo, L. (2022). Is neuro-symbolic AI meeting its promises in natural language processing? A structured review. *Semantic Web* (pp. 1–42).
- Hao, J., Chen, M., Yu, W., Sun, Y., & Wang, W. (2019). 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 (pp. 1709–1719). <https://doi.org/10.1145/3292500.3330838>
- Hitzler, P., Bianchi, F., Ebrahimi, M., & Sarker, M. (2020). Neural-symbolic integration and the semantic web. *Semantic Web*, 11(1), 3–11.
- Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., Melo, G. D., Gutierrez, C., Kirrane, S., Gayo, J. E. L., Navigli, R., Neumaier, S., et al. (2021). Knowledge graphs. *ACM Computing Surveys (CSUR)*, 54(4), 1–37.
- Jacinto, C., & Antunes, C. (2012). User-driven ontology learning from structured data. In *Proceedings - 2012 IEEE/ACIS 11th international conference on computer and information science, ICIS 2012*, Shanghai (pp. 184–189). <https://doi.org/10.1109/ICIS.2012.115>
- Janowicz, K., Yan, B., Regalia, B., Zhu, R., & Mai, G. (2018). Debiasing knowledge graphs: why female Presidents are not like Female popes. In *International semantic web conference (P&D/Industry/BlueSky)*.
- Jayawardana, V., Lakmal, D., de Silva, N., Perera, A. S., Sugathadasa, K., Ayesha, B., & Perera, M. (2017). Semi-supervised instance population of an ontology using word vector embedding. In *17th International conference on advances in ICT for emerging regions (ICTer)* (pp. 1–7).
- Kautz, H. A. (2022). The third AI summer: AAAI Robert S. Engelmore memorial lecture. *AI Magazine*, 43(1), 105–125. <https://doi.org/10.1002/aaai.12036>
- Li, C., Xian, X., Ai, X., & Cui, Z. (2020). Representation learning of knowledge graphs with embedding subspaces. *Scientific Programming*. <https://doi.org/10.1155/2020/4741963>
- McGregor, S. (2021). Preventing repeated real world AI failures by cataloging incidents: the AI incident database. In *Proceedings of the AAAI conference on artificial intelligence*.
- Meyer-Vitali, A., Mulder, W., & de Boer, M. H. (2021). Modular design patterns for hybrid actors. *arXiv preprint arXiv:2109.09331*.

- Minervini, P., Costabello, L., Muñoz, E., Nováček, V., & Vandenbussche, P.-Y. (2017). 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 (pp. 668–683). https://doi.org/10.1007/978-3-319-71249-9_40
- Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., Spitzer, E., Raji, I. D., & Gebru, T. (2019). Model cards for model reporting. In *Proceedings of the conference on fairness, accountability, and transparency* (pp. 220–229).
- Mouakher, A., Belkaroui, R., Bertaix, A., Labbani, O., Hugol-Gential, C., & Nicolle, C. (2019). 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. (pp. 307–312). <https://doi.org/10.1109/WETICE.2019.00070>
- Noy, N. F., & McGuinness, D. L. (2001). Ontology development 101: a guide to creating your first ontology, Technical Report. <http://www-ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness-abstract.html>
- Pertasas, V., Constantopoulos, P., & Androutsopoulos, I. (2018). 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 (pp. 162–178). https://doi.org/10.1007/978-3-030-00671-6_10
- Pingle, A., Piplai, A., Mittal, S., Joshi, A., Holt, J., & Zak, R. (2019). 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)* (pp. 879–886). <https://doi.org/10.1145/3341161.3343519>
- Poveda-Villalón, M., Fernández-Izquierdo, A., Fernández-López, M., & García-Castro, R. (2022). LOT: an industrial oriented ontology engineering framework. *Engineering Applications of Artificial Intelligence*, 111, 104755. <https://doi.org/10.1016/j.engappai.2022.104755>
- Prater, R., & Laurenzi, E. (2022). A hybrid intelligent approach for the support of higher education students in literature discovery. In *AAAI spring symposium: MAKE*.
- Rahman, M. M., & Finin, T. (2018). 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 (pp. 65–76).
- Roy, A., Akrotirianakis, I., Kannan, A. V., Fradkin, D., Canedo, A., Koneripalli, K., & Kulahcioglu, T. (2020). Diag2graph: representing deep learning diagrams in research papers as knowledge graphs. In *2020 IEEE international conference on image processing (ICIP)* (pp. 2581–2585). <https://doi.org/10.1109/ICIP40778.2020.9191234>
- Schlichtkrull, M., Kipf, T. N., Bloem, P., van den Berg, R., Titov, I., & Welling, M. (2018). 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 (pp. 593–607). https://doi.org/10.1007/978-3-319-93417-4_38
- Schreiber, G., Akkermans, H., Anjewierden, A., de Hoog, R., Shadbolt, N., Van de Velde, W., & Wielinga, B. (2000). *Knowledge engineering and management: the commonKADS methodology*. MIT Press.
- Shigarov, A., Cherepanov, I., Cherkashin, E., Dorodnykh, N., Khristyuk, V., Mikhailov, A., Paramonov, V., Rozhkov, E., & Yurin, A. (2019). 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 (pp. 1–12).
- Suárez-Figueroa, M. C. (2012). *NeOn methodology for building ontology networks: specification, scheduling and reuse*. Dissertations in artificial intelligence, IOS Press. ISBN 9783898383387.
- van Bekkum, M., de Boer, M., van Harmelen, F., Meyer-Vitali, A., & Teije, A. T. (2021). Modular design patterns for hybrid learning and reasoning systems: a taxonomy, patterns and use cases. *Applied Intelligence*, 51(9), 6528–6546.
- van Harmelen, F., & Ten Teije, A. (2019). 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*.
- van Harmelen, F., & ten Teije, A. (2019). A boxology of design patterns for hybrid learning and reasoning systems. *Journal of Web Engineering*, 18(1–3), 97–124.
- van Stijna, J. J., Neerincx, M. A., ten Teije, A., & Vethmanc, S. (2021). 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*.
- Waltersdorfer, L., Breit, A., Ekputra, J. F., Sabou, M., Ekelhart, A., Iana, A., Paulheim, H., Portisch, J., Revenko, A., Ten Teije, A., & van Harmelen, F. (2023). 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 (pp. 266–290), Chapter 10.
- Wang, M., Rong, E., Zhuo, H., & Zhu, H. (2018). 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 (pp. 141–153). https://doi.org/10.1007/978-3-319-93037-4_12
- Witschel, H. F., Pande, C., Martin, A., Laurenzi, E., & Hinkelmann, K. (2020). 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 (pp. 193–204).

- Xie, T., Yang, Y., Li, Q., Liu, X., & Wang, H. (2020). Knowledge graph construction for intelligent analysis of social networking user opinion. *Lecture Notes on Data Engineering and Communications Technologies*, 41, 236–247. https://doi.org/10.1007/978-3-030-34986-8_17
- Xu, Z., Harzallah, M., Guillet, F., & Ichise, R. (2020). Towards a term clustering framework for modular ontology learning. *Communications in computer and information science* 1222 CCIS (pp. 178–201). https://doi.org/10.1007/978-3-030-49559-6_9
- Zhang, Z., Cao, L., Chen, X., Tang, W., Xu, Z., & Meng, Y. (2020). Representation learning of knowledge graphs with entity attributes. *IEEE Access*, 8, 7435–7441. <https://doi.org/10.1109/ACCESS.2020.2963990>
- Zhao, Y., Feng, H., & Gallinari, P. (2019). Embedding learning with triple trustiness on noisy knowledge graph. *Entropy*, 21(11), 1083. <https://doi.org/10.3390/e21111083>