Towards a Standardized Description of Semantic Web Machine Learning Systems

Fajar J. Ekaputra[1,2], Laura Waltersdorfer[1], Anna Breit[3] and Marta Sabou[1,2]

[1] TU Wien, Vienna, Austria
[2] WU Wien, Vienna, Austria
[3] Semantic Web Company, Vienna, Austria

Abstract

In this paper, we report on our proposed approach towards a standardized description for systems combining machine learning (ML) components with techniques developed by the Semantic Web (SW) community (SWeMLS), which is one of lessons learned from our large-scale survey (476 papers) on the topic. We elaborate the key information that should be described of SWeMLS and selected methods to support its documentation.

Keywords

neuro-symbolic systems, semantic web, machine learning

1. Introduction

Neuro-symbolic AI [1], which combines Machine Learning (ML) and Knowledge Representation (KR) techniques is a strongly emerging trend in AI. At the same time, the Semantic Web (SW) research community has popularised knowledge representation techniques and resources in the last two decades [2] leading to a great interest in and uptake of SW resources outside of the Semantic Web research community [3]. These two trends have led to the development of systems that rely on both Semantic Web resources and Machine Learning components (*SWeML*).

This research area of SWeML has gained a lot of traction in the last few years, as shown in a rapidly growing number of publications in different outlets. At the same time, this growth poses challenges that threaten to hamper the further development of the field. One major challenge is the *lack of a standardized way to report SWeMLS* which leads to heterogeneous ways of reporting such systems depending on the background of the authors. As a result, the described systems lack crucial information for readers coming from other communities, which not only hinders the understandability of these systems, but also the comparability of different systems.

In this paper, we aim to discuss some lessons learned from a *Systematic Mapping Study (SMS)* [4] on SWeMLS. We first focus on necessary system information to be described in SWeMLS (cf. Section 2) and later discuss the methods and tools for improving reporting and documentation (cf. Section 3).

SemAI 2022: First Workshop on Semantic AI, co-located with SEMANTiCS Conference 2022, September 13–15, 2022, Vienna, Austria

fajar.ekaputra@tuwien.ac.at (F. J. Ekaputra); laura.waltersdorfer@tuwien.ac.at (L. Waltersdorfer); anna.breit@semantic-web.com (A. Breit); marta.sabou@wu.ac.at (M. Sabou)

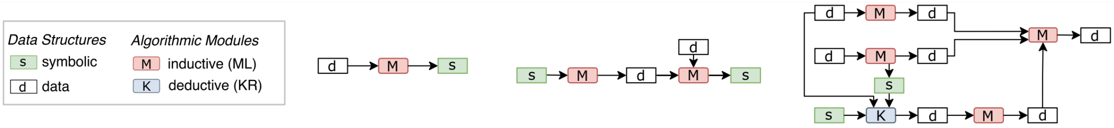

Figure 1: SWeMLS boxology, depicting the processing flow of information of three example systems.

2. Relevant SWeMLS information

In order to adequately represent SWeML Systems, we identify four categories of relevant system information: (i) System Settings, (ii) System Overview, (iii) System Details, and (iv) System Evaluation, which will be briefly described in the following.

System Settings To estimate the applicability of a presented approach, the description of the **domain** in which a SWeMLS was evaluated is essential. The targeted **task** should both be described from the use-case side –if applicable– i.e., which specific problem is being solved (e.g., drug-drug-interaction prediction) as well as from the framing of the problem in the system setting (e.g., link prediction task). Finally, explicitly stating the development **maturity** of the presented system helps the reader to estimate its state of adaptation as well as its reliability.

System Overview Depicting the general **processing flow** information through the presented system facilitates common understanding of the main processes, without diving into too much detail. Special focus should be laid to distinct and describe the **main components** in this processing flow, being *processing units*, i.e., Machine Learning components and Reasoning modules, as well as the *data structures* (e.g., symbolic data such as KGs, or non-symbolic data such as embeddings) on which the processing units operate. Finally, it is essential to highlight possible differences in these processing flows in **different phases** of the system, e.g., during training and deployed solution.

System Details To further describe the aforementioned **ML components** in more detail, the authors should provide information about the model architecture including the base models used (e.g., BERT-base), additional modules (e.g.cross-attention layer), as well as development and training details such as the training procedure (e.g., distantly supervised), loss function and utilized optimizers. For the **SW components** characteristics such as size and formalism of the SW resource are interesting. Furthermore, the type (e.g., taxonomy / ontology) as well as their semantic exploitation (e.g., only labels / one type of relation are used) provide highly useful information. Finally, any used semantic processors (e.g., reasoners) should be well documented.

System Evaluation To increase the reproducibility, evaluation details need to be captured and reported such as **pre-processing steps** (e.g., hyperparameter tuning), **final model parameters**, **hardware specifications** and **auditability**, such as relevant context information on the system lifecycle, starting from the design phase to the operating system.

3. Methods for SWeMLS Documentation

This section describes selected methodologies to facilitate describing and documenting SWeMLS based on the identified relevant information (cf. Section 2).

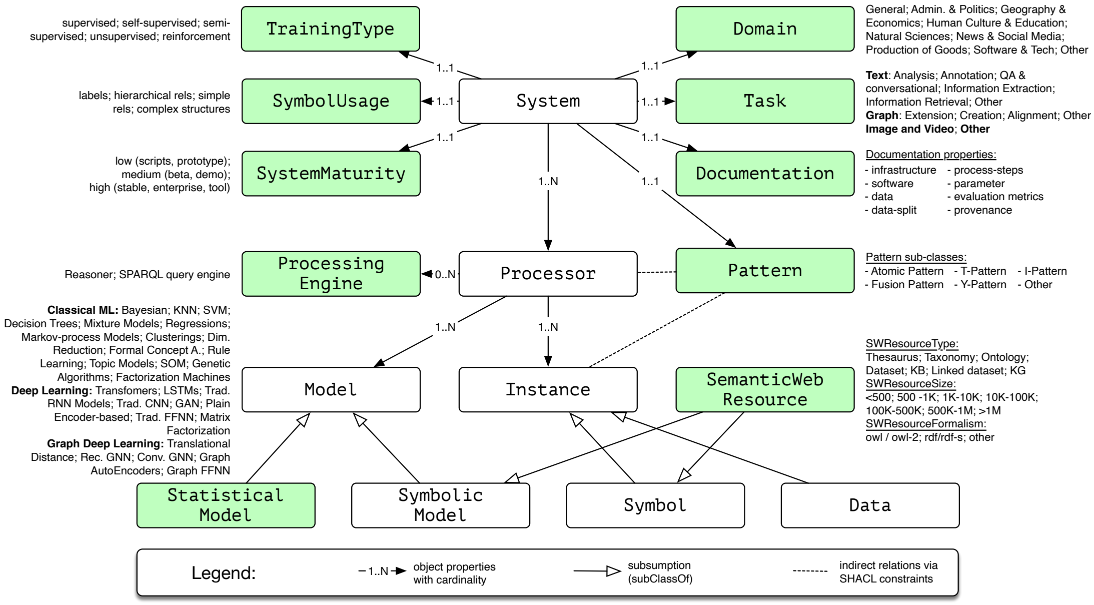
supervised; self-supervised; semi-supervised; unsupervised; reinforcement
TrainingType
labels; hierarchical rels; simple rels; complex structures
SymbolUsage
low (scripts, prototype); medium (beta, demo); high (stable, enterprise, tool)
SystemMaturity
Reasoner; SPARQL query engine
Processing Engine
Classical ML: Bayesian; KNN; SVM; Decision Trees; Mixture Models; Regressions; Markov-process Models; Clusterings; Dim. Reduction; Formal Concept A.; Rule Learning; Topic Models; SOM; Genetic Algorithms; Factorization Machines
Deep Learning: Transfomers; LSTMs; Trad. RNN Models; Trad. CNN; GAN; Plain Encoder-based; Trad. FFNN; Matrix Factorization
Graph Deep Learning: Translational Distance; Rec. GNN; Conv. GNN; Graph AutoEncoders; Graph FFNN
System
1..1
1..1
1..1
1..1
1..1
1..1
1..N
1..1
Processor
0..N
1..N
1..N
Model
Instance
Statistical Model
Symbolic Model
Symbol
Data
Domain
General; Admin. & Politics; Geography & Economics; Human Culture & Education; Natural Sciences; News & Social Media; Production of Goods; Software & Tech; Other
Task
Text: Analysis; Annotation; QA & conversational; Information Extraction; Information Retrieval; Other
Graph: Extension; Creation; Alignment; Other
Image and Video; Other
Documentation
Documentation properties:
- infrastructure
- software
- data
- data-split
- process-steps
- parameter
- evaluation metrics
- provenance
Pattern
Pattern sub-classes:
- Atomic Pattern
- Fusion Pattern
- T-Pattern
- Y-Pattern
- I-Pattern
- Other
SemanticWeb Resource
SWResourceType:
Thesaurus; Taxonomy; Ontology; Dataset; KB; Linked dataset; KG
SWResourceSize:
<500; 500 -1K; 1K-10K; 10K-100K; 100K-500K; 500K-1M; >1M
SWResourceFormalism:
owl / owl-2; rdf/rdf-s; other
Legend:
1..N
object properties with cardinality
subsumption (subClassOf)
indirect relations via SHACL constraints

Figure 2: SWeMLS classification concepts, relations/properties and identified instances. *Green-colored boxes represent main concepts, while white boxes represent supplementary classes*

Describing SWeMLS For describing the overall SWeMLS, we propose the usage of the ***SWeMLS classification system*** introduced in [4]. For this framework, we reuse a number of concepts introduced by van Bekkum et al. [5], including `Instance`, `Model`, and their sub-concepts, and add a number of classes and properties related to SWeMLS to align the classification system with the main characteristics described in Section 2. Therefore, the classification system provides guidance on what information should be provided when describing such systems, as well as a unified and machine-readable way to document SWeMLS. For the most common processing flows, we further provide a set of re-usable patterns. We formalized the classification of SWeMLS as an ontology and provide instances identified during the SMS (cf. Figure 2). The complete documentation of the ontology is available online[1].

Describing SWeMLS Processing Flows As the processing flow forms one of the most essential parts in understanding SWeMLS, great effort should be put in its documentation. To facilitate the description, we propose a visual representation based on existing ***boxologies***, as they provide an intuitive way of abstraction and documentation. The boxology used in our conducted SMS is built on the framework introduced in [6] which proposes algorithmic modules, i.e. inductive (ML) or deductive (KR), and data structures, i.e., symbolic (such as semantic entities or relations) or non-symbolic (such as text, images, or embeddings) (cf. Figure 1). The modular design patterns of [5] additionally provides the possibility to describe actors and processes.

Describing SWeMLS Auditability The aim of documenting the auditability characteristics of SWeMLS, is to increase the transparency of design decisions and operational details. For the ML design phase, Naja et al. [7] propose a semantic framework to capture and manage traces

[1] http://semantics.id/semsys/ns/swemls/index-en.html

for accountability and audit purposes. However, there are neither considerations for the entire lifecycle, nor for SWeMLS. To overcome this gap, we have introduced a **SWeMLS lifecycle** [8] to achieve a common view and make system interactions explicit. The model is divided into three perspectives: ML resource, SW resource and Application. Both types of resources have a Design and Operation Phase with various steps. The framework can support the identification of design and operation traces to increase the auditability of SWeMLS.

4. Conclusion and Future Work

SWeMLS are used to solve problems in diverse research fields proving their broad applicability. Domain experts and AI researchers, however, are hampered by the lack of standardized system description. This paper identify *1) essential system information* to foster common understanding and comparability of approaches and 2) *diverse methods to support documentation* as a basis towards a standardized approach to describe and document SWeMLS.

Future Work. We plan to extend our classification system and to further enhance the machine-readability of the descriptions (e.g., via Open Research Knowledge Graph initiative) and propose an evaluation framework to assess the level of auditability of such systems.

Acknowledgments This work is supported by the FFG Project OBARIS (Grant Agreement No 877389).

References

[1] G. Booch, F. Fabiano, L. Horesh, K. Kate, J. Lenchner, N. Linck, A. Loreggia, K. Murugesan, N. Mattei, F. Rossi, B. Srivastava, Thinking fast and slow in ai, in: AAAI, 2021.

[2] P. Hitzler, A review of the semantic web field, Communications of the ACM 64 (2021).

[3] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. D. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, et al., Knowledge graphs, ACM CSUR 54 (2021) 1–37.

[4] A. Breit, L. Waltersdorfer, J. F. Ekaputra, M. Sabou, A. Ekelhart, A. Iana, H. Paulheim, J. Portisch, A. Revenko, A. Ten Teije, F. van Harmelen, Combining Machine Learning and Semantic Web -A Systematic Mapping Study (under review), ACM CSUR (2022).

[5] M. van Bekkum, M. de Boer, F. van Harmelen, A. Meyer-Vitali, A. ten Teije, Modular Design Patterns for Hybrid Learning and Reasoning Systems, Applied Intelligence 51 (2021) 6528–6546.

[6] F. van Harmelen, A. ten Teije, A boxology of design patterns for hybrid learning and reasoning systems, J. of Web Engineering 18 (2019) 97–124. arXiv:1905.12389.

[7] I. Naja, M. Markovic, P. Edwards, C. Cottrill, A semantic framework to support ai system accountability and audit, in: ESWC, Springer, 2021, pp. 160–176.

[8] A. Breit, L. Waltersdorfer, F. J. Ekaputra, M. Sabou, T. Miksa, A lifecycle framework for semantic web machine learning systems (accepted), in: DEXA, 2022.