

# Bridging the gap between an ontology and deep neural models by pattern mining<sup>1</sup>

Tomas MARTIN <sup>a</sup>, Abdoulaye Baniré DIALLO <sup>a,b</sup>, Petko VALTCHEV <sup>a</sup> and René LACROIX <sup>c</sup>

<sup>a</sup> *CRIA, UQAM, Montreal, Canada*

<sup>b</sup> *LACIM, UQAM, Montreal, Canada*

<sup>c</sup> *LactaNet, Sainte-Anne-de-Bellevue, Canada*

## Abstract.

A domain ontology (DO) is a machine-readable knowledge repository compatible with the popular knowledge graph (KG) format. An intriguing question is how to leverage a DO plus a KG in a neural learning process. We propose to use ontology-rooted graph patterns mined from a DO-compatible graph translation of the raw data as a vector for injecting some domain knowledge into the neural network. Such patterns represent a frequently occurring regularities in the data yet they are expressed in terms of the ontological entities (classes, properties, etc.) and reflect additional knowledge from the KG. Using them as an additional input to the learning process seems a promising way to guide it towards improved explainability, accuracy and convergence, as well as, in a more general vein, increase the generalization power of the neural models.

**Keywords.** Knowledge graphs, Ontologies, Neural models, Graph pattern mining

## 1. Introduction

Decision support systems (DSS) aim at helping practitioners in complex activities by providing suggestions as to the best action to perform. Many of them use machine learning (ML) to predict the outcome of a specific problem and select concrete actions correspondingly. Deep learning (DL) has risen as a promise to expand the reach of successful automation, hence the expectation for effective decision support to profuse.

However, predicting or learning representations on complex domains requires large amounts of data of sufficiently high quality. In practice, though, such data are not always readily available, especially when dealing with biological entities, for which data acquisition can be pricey. Conversely, expert knowledge about the domain can be available in a machine-readable form. Since it reflects the expertise underlying decision making, it is natural to look for ways to inject that knowledge into the learning process, e.g. to mitigate data scarceness.

Ontologies –structured representations of concepts and their relationships [1]– provide the means to express descriptive domain knowledge [2]. As such, they have gained

---

<sup>1</sup>Copyright © 2020 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

significant popularity in life sciences [3], in particular, in precision agriculture [4]. Moreover, a domain ontology (DO), as being targeted at generic domain knowledge, is often complemented by a compatible expression of the factual knowledge from that domain, typically formatted as a knowledge graph (KG).

Here, we tackle the problem of injecting the domain knowledge encoded in a pair DO + KG into a modern neural architecture that accounts for heterogeneous data. We face it within a collaborative project on dairy production optimization whereby the encompassing goal is to correctly predict the milk yield of an individual cow. To that end, a historical dataset about the provincial livestock is fed into a dedicated deep learning (DL) architecture. To improve its predictive and generalization power, as well as to increase the explanation opportunities, we decided to infuse a good amount of domain knowledge as encoded into the DO and its KG. Thus, we turned to graph patterns as a vector for the infusion: We first transformed the raw historical data into RDF<sup>2</sup> graphs compatible with the DO and then mined all relevant patterns from the resulting graph set. As our patterns are expressed in a language that only refers to entities from the DO and its KG (OWL<sup>3</sup>) we used them in a specific sort of data augmentation, i.e. as complementary domain knowledge-rooted descriptive features to be fed into a dedicated ANN learning architecture that combines them with the remaining features from the original data.

While the expected benefits range from increased accuracy of the resulting model w.r.t. the baseline data-only learner, to greater generalization power to enhanced explainability of the overall neural learning process, there are significant challenges we face at various steps of the process. We expand on those below.

The remainder of the paper is organized as follows: Section 2 motivates our study by outlaying the context of our ongoing project on dairy production. Section 3 summarizes related work on DOs and KGs as support for ANN learning as well as on pattern mining from RDF and KG. Section 4 expands on our vision as to the way ontologically-generalized patterns could bring domain knowledge into the neural learning context while in section 5 we describe the overall process and zoom on the graph mining task. Finally, we conclude in section 6.

## 2. Motivation

Precision farming is a new trend in agriculture that emphasizes the use of the available data about a farm, e.g. a dairy one, in optimizing the production process in terms of revenues, ecological impact, animal welfare, etc. Indeed, daily activities in agri-food sector generate large amounts of data through sensors (animal body mass), manually (visual assessment of body parameters like corn shape) or as the product of complex analyses (milk fat). Data is typically split into datasets w.r.t. the separate aspects of the management process such as farm yield, environment, animal health, genetics, etc. Precision farming looks at analyzing it to support the decision making of domain stakeholders [5,6]: farmers, agronomists, veterinarians, dairy processors, government agencies, etc.

To deal with such a heterogeneity, a two-fold approach seems suitable: On one hand, a significant data engineering effort is needed to achieve a unique global dataset as com-

---

<sup>2</sup>Resource Description Framework

<sup>3</sup>Web Ontological Language

prehensive dairy data standard is not yet available (due to diverging formats and/or granularity, alternative definitions, different ways of calculating indicators, privacy, etc.). This requires the design of an expressive data schema federating all sources, i.e. a lightweight DO. On the other hand, existing statistical models only partially cover the heterogeneous set of domain variables pertaining to, say, cow profitability, which underlies the decision to keep an animal in the herd. Therefore, dedicated deep artificial neural network (ANN) architectures are currently experimented for the prediction of such fundamental quantities as milk yield, total cost, etc. However, our experiments have shown that when applied on dairy production data, most popular deep models fail to fully catch the dynamics in a cow life-cycle [7] and thus suffer on lower accuracy, e.g. in predicting the yield of sub-categories of cows like dry ones.

We see here a need to support the learning process by injecting some domain knowledge into the ANN, and a rich DO seems the right candidate for that. As a first-class component of the DSS, the DO could be the source of machine-readable domain knowledge to leverage in the data analysis process. In a different vein, explainability and intelligibility [8] are highly desirable for a DSS which interacts with domain experts and practitioners if its recommendations are to be heeded and results trusted.

In summary, to be effective, the DSS outcomes have to reflect existing practices and relevant technical knowledge (e.g. at lactations's end cows get dry) while they also need be explainable to its broad range of users [9,10]. A reasonably-developed DO might be the answer to both requirements. Yet, while definitely a research track to be followed, combining ANNs with DOs is a challenging task due to the divergence in the respective levels of knowledge expression (*sub-symbolic* and *symbolic*, respectively).

### 3. Current State Of The Art

Symbolic representations such as ontologies have been exploited for many decades due to their ability to intuitively and logically model abstractions for knowledge management and problem solving [11]. With well-defined concepts, rules, and hierarchies, they are built around basic blocks forming a complex conceptual structure. ANN-based representations constitute an alternative knowledge capture tool [12]. By trading modelling entities, i.e. discrete and man-made, for machine-made and loosely defined “patterns”, the later breaks free of prior knowledge in order to benefit from, arguably, a more powerful yet difficult to interpret representation language. At its core, information is distilled throughout a network as a set of waves (or pulses) representing captured knowledge.

Only recently, the collaboration of DOs/KGs with modern analytical architectures such as deep ANN started to attract the attention of the scientific community in artificial intelligence [9]. A variety of topics have been covered by the ensued research trend, among them being domain knowledge infusion [13], reasoning [14], explainability [15], etc. Below, we provide an overview of ANN methods exploiting ontologies and then briefly mention alternative approaches exploiting/encoding (parts of) DOs or KGs.

*Combining ANNs with DOs and/or KGs* In a nutshell, a majority of existing work exploits the symbolic representations as a source of external knowledge for domain specific tasks or as a pre-processing step prior to the learning process. For example, [16] exploits a DO to learn better text embeddings by injecting external terms and domain entities. Alternatively, [17] investigates the improvement of gene annotation prediction accuracy

of a deep auto-encoder ANNs whenever supported by the Gene Ontology [18]. While being practical applications of the knowledge contained in DOs, these methods do not fully embrace its conceptual structure.

To the best of our knowledge, few methods aim for a generic DO integration into the neural learning process: [19] aims to exploit additional neural layers, dedicated to learning weights for each abstraction level in the DO. Such layers are responsible for learning the relationship between classes, sub-classes and super-classes. [15] targets explainability as a representation learning problem and proposes to learn concepts by identifying key characteristics of individuals (i.e. sets of properties) expressed using a DO, prior to a prediction step.

In a different vein, reasoning with neural networks [20,14] amounts to a link prediction problem where new links represent inferences (e.g. transitivity, subsumption, etc.) or alternatively could be viewed as predicting individual membership to pre-defined categories (e.g. ontology class). Surprisingly, approaching it as a translation problem with auto-encoders achieves good results on noisy-data.

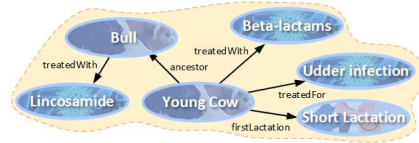
On a broader scope, the embedding of concepts and relationships from a DO has been extensively studied since at least [21] and that of individual resources and triples from a KG even more so (see [22] for a survey).

Yet the corresponding methods seem to better fit the processing of graph data alone rather than exploiting the DO in the overall analytical process.

*Mining patterns from a DO-compliant graph dataset* Pattern mining [23], aims at extracting recurrent patterns capturing the most relevant regularities in the data. Relevance is typically measured in term of frequency. Beyond existing vanilla graph pattern mining [24], few approaches tackle mining such patterns using a DO. The trend was initiated by [25] which introduced *generalized* graph patterns and proposed an extension of an existing graph miner that efficiently mines non-redundant such patterns. Here, the generalized items are drawn from a taxonomy, i.e. a light-weight DO. In [26], the authors introduced ontologically-generalized patterns, in the sense that: (1) the generalized items stem from a DO, and (2) unlike previous studies, generalization was performed on graph edges. On the downside, while the data records are oriented graphs, the backbone thereof represents a sequence which substantially reduces the processing effort [27] but limits the scope of the method. Recently, in [28] the use of graph patterns for KG summarization has been proposed. While technically different from graph pattern mining, the method still explores the space of possible summaries, i.e. patterns.

As a simplified scenario, the mining of flat generalized patterns from RDF data has been studied at least since [29]. The method works on triple sets representing the initial graphs and outputs what is called *generalized relation sets* which do not necessarily represent a connected graph. In a slightly different vein, mining association rules (AR) from RDF dataset with a DO has also been investigated. In [30], a method is proposed for extracting logical association rules made of ontological components implementing an inductive logic programming (ILP)-based strategy for the traversal of the rule space. A special flavor of flat AR custom-tailored for RDF data has been introduced in [31]: Unlike standard AR, they explicitly incorporate the supporting set of RDF resources, i.e. domain individuals while items represent shared combinations (predicate, object) in the corresponding triples.

In summary, despite a significant amount of work on a wide range of combination scenarios for DOs/KGs and ANNs, the question of how to efficiently and effectively in-



**Figure 1.** An example of ontological graph pattern.

ject domain knowledge, generic and or factual, into an ANN is still wide open. Moreover, to the best of our knowledge, no prior work has set the use of generalized graph patterns as a vector for this task. More intriguing even, the notion of ontologically-generalized pattern has not been studied in its most general settings, i.e. with RDF data graph(s) as input and fully-blown DO and KG as additional parameters.

#### 4. Our Vision

As a vector for bringing domain knowledge into the ANN, we propose to use frequent patterns that: (1) are mined from the data graphs and (2) use the vocabulary provided by the DO and the KG. By bringing in ontological entities we aim at making explicit the shared conceptual structure otherwise invisible in raw data. Indeed, while exact values/labels may mismatch, higher-order abstractions describing them may well coincide. For instance, assume two lactating cows from a herd have been treated for *mastitis* (udder infection) with two different antibiotics, say *amoxicillin* and *penicillin*, respectively. Now, considering what both cases have in common and how to express this as a unique shared graph structure, we easily see the graph should comprise nodes for cow and mastitis connected by an edge expressing the illness relationship. Moreover, the cure with antibiotics could also be represented, and, if higher precision is desired, even the fact that both drugs used are of the  $\beta$ -lactams category might be reflected. Obviously, the latter increase in the common structure would require a taxonomy on drugs or, more ambitiously, a drug DO (e.g. linking drugs to symptoms, health disorders, adverse effects, etc.)

In a more general vein, inserting typing information and property generalizations as available within a DO helps reveal hidden commonalities that wouldn't be easily spotted by a neural learner. Here, our goal is to discover such relevant shared fragments in data so that they could support ANN learning. As an illustration, Figure 1 presents a (purely fictional) pattern which summarizes the above example. The pattern hints at possible causes for a shorter *firstLactation* for *Young Cows*: In summary, such a cow and a male *ancestor* of it, have been both treated with antibiotics of specific types. Observe that the pattern is a graph whose components are classes (vertices) or properties (edges) from a hypothetical DO.

The context of our pattern extraction is illustrated in Figure 2: On top, the relevant excerpts of a hypothetical DO for dairy production are drawn whereas beneath lay two data graphs, #1 and #2 (named RDF graphs), both matched by the pattern in Figure 1.

Observe now that, while vertices in data graphs are instances, patterns are made of classes, or rather exemplars thereof, that match the ontological types of respective data nodes. For instance, vertex-wise, in graph #1, Duke is a *Bull* while Clindamycin is a *Lincosamide*, another subclass of *Antibiotic*. Edge-wise, in pattern they correspond to

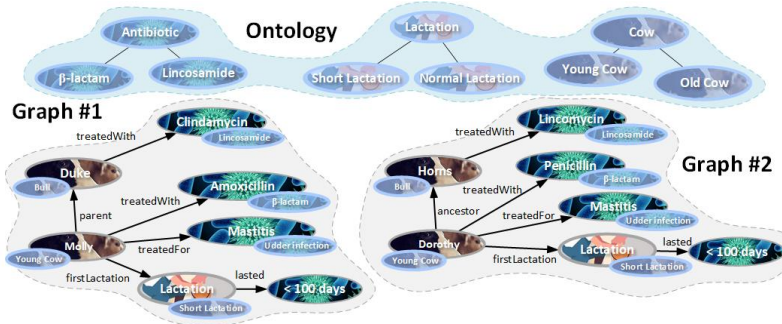


Figure 2. Context of the example in Figure 1.

ontological properties that are identical to or more general than their counterparts in data graphs, e.g. *ancestor* in the pattern generalizes *parent* from graph #1 and is identical to the corresponding link in graph #2. The above specifications make for a very specific graph structure, a doubly-labelled –on both vertices and edges– multi-graph.

Now, back to ANN, the grounding idea is that we discover the relevant patterns from a graph dataset and then we feed them to the network as higher-level descriptors of the matching data graphs. This is not unlike data augmentation, a preprocessing step aimed at improving the learning outcome. For instance, on images, additional expert knowledge in the form of binary masks, heatmaps or bounding boxes are expert-fed into the original data to help the network discriminate objects [32]. Similarly, modern NLP methods typically enhance text data with manual annotations to heed finer-grained type labelling [33].

In comparison, our approach offers quasi-full automation: Even if patterns could be manually crafted and then attached to data, a more effective approach is to both automatically mine and assign them.

Yet there are more palpable advantages of using the DO-based patterns. On one hand, unlike isolated bounding boxes in images, they offer an integrated view of the shared structure: Edges standing for properties connect class vertices, thus providing context to each of them. Moreover, individual patterns pertain to potentially varying abstraction levels, thus leaving the determining of the right level in a particular case to the learning component.

Interestingly enough, our patterns intertwine complementary aspects of entity description: Part of it reflects a definitional view (intrinsic features), e.g. cow ID, birth date, race. The remainder translates dynamic aspects of the domain, i.e. here: calving, lactation, milk tests, health events, etc. While the former represents a sort of invariant mirroring the DO structure, the latter admits substantial variation, e.g. a healthy cow with no health issues vs a poorly bred one which gets ill fairly often. Obviously, both categories of descriptors would contribute differently to the shared structure and hence appear in the patterns with unequal frequency. Since the underlying DO components are known beforehand the pattern discovery process could be biased to favour one or the other.

Expected **immediate benefits** of the ontological knowledge injection into the neural learning process include higher accuracy in predictive architecture and faster convergence. Additionally, the explainability of the results should be increased.

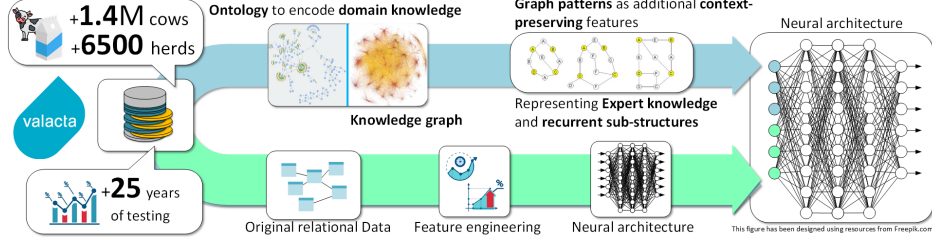


Figure 3. High-level view of our hybrid learning process.

## 5. Technical aspects of our approach

Our approach can be summarized as follows: First, we designed a DO for the dairy production field as a rich unified data schema for the available datasets that enables integration and interrogation thereof. It is currently complemented with a knowledge graph that reflects the current practices in the Quebec province. Next, starting from the raw data in the Valacta warehouse, a dataset of named RDF graphs representing animals and compatible with our DO was produced to populate a dedicated triple store. At the following step, these graphs are mined using the DO and its KG as a domain knowledge source in order to extract the most relevant graph patterns of a very specific flavor. Indeed, rather than referring to actual graph components, our patterns are expressed in terms of ontological entities, i.e. OWL classes and properties, which qualifies them as (*ontologically*) *generalized graph patterns*. The resulting patterns are then fed into the predictive ANN models for dairy production. In fact, they are used as supplementary features for cow records that are submitted to the target neural architecture. Depending on the kind of features selected in the initial cow-centered datasets, this may or may not require a preliminary embedding of the initial records into a lower-dimension space.

Figure 3 shows a detailed view of the entire learning pipeline. First, based on original data and domain expert feedback, we model and populate a dairy DO. Then, we mine graph patterns representing recurring sub-graphs made of ontological abstractions. These patterns refine the initial data with extra features reflecting the content of the DO which are to help improve the subsequent neural learning step.

Technically speaking, a pattern mining task [23] is specified by defining two languages, one for data records one and one for patterns, and a pattern interestingness criterion. Our languages are both based on the DO populated by, on the one hand, the KG and on the other hand the input dataset translated into RDF triples. Let this extended version of the DO be denoted  $\Omega = \langle O, C, \mathcal{R}, \leq_C, \leq_{\mathcal{R}}, \in_C, \rho \rangle$  be an ontology where  $O, C$  and  $\mathcal{R}$  are its sets of objects (RDF instances), classes, and object properties, respectively. Observe that we do not consider property graphs, hence we ignore the data property part of our dataset. Both classes and properties are organized into well-defined hierarchies where  $H_C = \langle C, \leq_C \rangle$  and  $H_{\mathcal{R}} = \langle \mathcal{R}, \leq_{\mathcal{R}} \rangle$  with  $\leq_C$  denoting the `rdfs:subClassOf` relationship and  $\leq_{\mathcal{R}}$  the `rdfs:subPropertyOf` one. The instantiation relationship  $\in_C \subseteq O \times C$  is the translation of `rdf:type`. The incidence relation  $\rho \subseteq C \times \mathcal{R} \times C$  is made of triples  $c_1 \times r \times c_2$  denoting a relation  $r$  between classes  $c_1$  and  $c_2$  as its domain and range, respectively.

In the above notations, a graph data record  $g_d$  (see Figure 2) represents a doubly-labelled directed multi-graph. Such graph record  $g_d = \langle V_d, E_d \rangle$  where  $V_d$  is a set of ver-

tices and  $E_d$  is a multi-set of pairs of vertices. A labelling function  $\lambda$  maps each vertex in  $V_d$  to a label in  $O$  and edge in  $E_d$  to  $\mathcal{R}$ . Moreover, intuitively, a pair of adjacent vertices in  $g_d$  exists iff the corresponding RDF triple exists in the populated ontology. Obviously, our data language fully coincides with RDF. Now, a pattern  $g_p$ , expressed by our pattern language, is also a doubly-labelled directed multi-graph  $g_p = \langle V_p, E_p \rangle$ , yet its vertices are labelled with classes from  $C$  while edges are mapped to the same set  $\mathcal{R}$ . Two examples of such pattern graphs have been discussed in section 4 (see Figure 2).

Alternatively,  $p$  can be represented as a set of triples. Finally, interestingness of pattern is typically measured by their frequency in the dataset, yet other generic measures, e.g. utility, or domain-dependent ones can be adopted.

Next, an effective mining method requires a strategy for pattern space traversal and a technique to perform a pattern-to-data matching. Matching with graphs is akin to sub-graph isomorphism which, in the presence of a DO, must be enhanced by generality and instantiation relationships. Efficient traversals, in turn, require defining a spanning tree of the pattern generality relationship (the transitive reduction thereof) seen as a graph itself. This spanning tree can then be itself traversed with a particular discipline that yields a total order on patterns. Furthermore, a canonical representation of graph patterns is another mandatory construct to avoid multiple generations of the same pattern [24]. While DOs and patterns have been combined before, to the best of our knowledge, no mining method has targeted data of such complexity as our *ontologically-generalized graphs patterns*. Known downsides of pattern mining include sensibility on frequency threshold and the potential combinatorial explosion, yet these admit mitigation strategies such as using condensed representations, e.g. closed patterns [34] or maximal ones [35]. This is the subject of our current research.

## 6. Conclusion

We describe an approach for infusing some domain knowledge encoded into a DO and a compatible KG into a neural learning process that boils down to augmenting the learning dataset with additional descriptive features. The features correspond to graph patterns mined from the raw data formatted as graphs under the DO, which are themselves expressed in terms of DO/KG entities. Key advantages of our approach include higher abstraction level of the new features, contextualized expression of commonalities in data, potential automation, etc. Expected benefits thereof range from increased prediction accuracy to faster convergence of the ANN to higher explainability of the results.

At the current stage, we are fine-tuning the design of our graph mining tool which faces a huge and highly combinatorial pattern space that is induced over graphs by the ontological relationships of generality and instantiation. It required the development of dedicated mining methods as classical graph pattern miners could not go beyond label equality on vertices and edges.

## References

- [1] Gruber TR, et al. A translation approach to portable ontology specifications. Knowledge acquisition. 1993;5(2):199–220.
- [2] Kramer F, Beißbarth T. Working with ontologies. In: Bioinformatics. Springer; 2017. p. 123–135.



- [3] Cannataro M, et al. Biomedical and bioinformatics challenges to computer science. *Procedia Computer Science*. 2010;1(1):931–933.
- [4] Drury B, et al. A survey of semantic web technology for agriculture. *Information Processing in Agriculture*. 2019 Mar. Available from: <https://linkinghub.elsevier.com/retrieve/pii/S2214317318302580>.
- [5] Kuwata K, Shibasaki R. Estimating crop yields with deep learning and remotely sensed data. In: 2015 IEEE IGARSS. IEEE; 2015. p. 858–861.
- [6] Barbosa A, et al. Modeling yield response to crop management using convolutional neural networks. *Computers and Electronics in Agriculture*. 2020;170:105197. Available from: <http://www.sciencedirect.com/science/article/pii/S0168169919308543>.
- [7] Frasco C, et al. Towards an Effective Decision-making System based on Cow Profitability using Deep Learning. In: *Proc. of the 12th ICAART*. Valletta, Malta; 2020. p. 949–958. Available from: <http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0009174809490958>.
- [8] Fuji M, et al. “Trustworthy and Explainable AI” Achieved Through Knowledge Graphs and Social Implementation. *Fujitsu Scientific & Technical Journal*. 2020;56(1):39–45.
- [9] Bordes A, et al. A semantic matching energy function for learning with multi-relational data: Application to word-sense disambiguation. *Machine Learning*. 2014 Feb;94(2):233–259. Available from: <http://link.springer.com/10.1007/s10994-013-5363-6>.
- [10] Dumančić S, et al. Learning relational representations with auto-encoding logic programs. In: *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. AAAI Press; 2019. p. 6081–6087.
- [11] Guarino N. The ontological level. *Philosophy and the cognitive sciences*. 1994.
- [12] Bengio Y, et al. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*. 2013;35(8):1798–1828.
- [13] Sheth A, et al. Shades of Knowledge-Infused Learning for Enhancing Deep Learning. *IEEE Internet Computing*. 2019 Nov;23(6):54–63.
- [14] Makni B, Hendler J. Deep learning for noise-tolerant RDFS reasoning. *Semantic Web*. 2019;10(5):823–862.
- [15] Phan N, et al. Ontology-based deep learning for human behavior prediction with explanations in health social networks. *Information sciences*. 2017;384:298–313.
- [16] Arguello Casteleiro M, et al. Deep learning meets ontologies: experiments to anchor the cardiovascular disease ontology in the biomedical literature. *Journal of Biomedical Semantics*. 2018 Dec;9(1):13.
- [17] Chicco D, et al. Deep autoencoder neural networks for gene ontology annotation predictions. In: *Proceedings of the 5th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics - BCB '14*. ACM Press; 2014. p. 533–540. Available from: <http://dl.acm.org/citation.cfm?doid=2649387.2649442>.
- [18] Day-Richter J, et al. OBO-Edit—an ontology editor for biologists. *Bioinformatics*. 2007;23(16):2198–2200.
- [19] Jiménez A, et al. Sound event classification using ontology-based neural networks. In: *Proceedings of the Annual Conference on Neural Information Processing Systems*; 2018. .
- [20] Hohenecker P, Lukasiewicz T. *Deep Learning for Ontology Reasoning*. CoRR. 2017.
- [21] Bordes A, et al. Learning Structured Embeddings of Knowledge Bases. In: *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*; 2011. p. 301–306.
- [22] Wang Q, et al. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*. 2017;29(12):2724–2743.
- [23] Aggarwal CC, Han J, editors. *Frequent Pattern Mining*. Springer; 2014.
- [24] Yan X, Han J. gspan: Graph-based substructure pattern mining. In: *Proc. of the IEEE ICDM 2002*. IEEE; 2002. p. 721–724.
- [25] Inokuchi A. Mining Generalized Substructures from a Set of Labeled Graphs. In: *Fourth IEEE International Conference on Data Mining (ICDM'04)*. IEEE; . p. 415–418.
- [26] Adda M, et al. Toward Recommendation Based on Ontology-Powered Web-Usage Mining. *IEEE Internet Computing*. Jul.;(4):45–52.
- [27] Adda M, et al. A framework for mining meaningful usage patterns within a semantically enhanced web portal. In: *Proc. of the 3rd C\* CCSE*. ACM; 2010. p. 138–147.
- [28] Song Q, Wu Y, Lin P, Dong LX, Sun H. Mining Summaries for Knowledge Graph Search. *IEEE Transactions on Knowledge and Data Engineering*. 2018 Oct;30(10):1887–1900.

- [29] Jiang T, et al. Mining generalized associations of semantic relations from textual web content. *IEEE transactions on knowledge and data engineering*. 2007;19(2):164–179.
- [30] Lavrač N, et al. Using ontologies in semantic data mining with segs and g-segs. In: *International Conference on Discovery Science*. Springer; 2011. p. 165–178.
- [31] Barati M, et al. SWARM: An Approach for Mining Semantic Association Rules from Semantic Web Data. In: *PRICAI 2016: Trends in Artificial Intelligence*. vol. 9810. Springer; 2016. p. 30–43.
- [32] Shorten C, Khoshgoftaar TM. A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*. 2019;6(1):60. Available from: <https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0197-0>.
- [33] Giridhara P, et al. A Study of Various Text Augmentation Techniques for Relation Classification in Free Text. In: *Proc. of the 8th ICPR*; 2019. p. 360–367. Available from: <http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0007311003600367>.
- [34] Yan X, Han J. CloseGraph: mining closed frequent graph patterns. In: *Proc. of the 9th ACM SIGKDD*. ACM; 2003. p. 286–295.
- [35] Huan J, et al. Spin: mining maximal frequent subgraphs from graph databases. In: *ACM SIGKDD 2004*. ACM; 2004. p. 581–586.