

# A Review of Machine Learning Applications in Semantic Web

Nabila El-mukhtar Mohammad Albannai  
Institute of Science, Kastamonu University, 37150, Kastamonu, Turkey  
E-mail: rehamranemrahf2009@gmail.com

Yasemin Gultepe (Corresponding author)  
Computer Engineering Department, Engineering and Architecture Faculty,  
Kastamonu University, 37150, Kastamonu, Turkey  
E-mail: yasemingulpepe@kastamonu.edu.tr

Abdulmawla Najih  
Computer and Communication Systems Engineering Department, Engineering Faculty,  
Putra Malaysia University, 43400 Selangor Darul Ehsan, Malaysia  
E-mail: nabdulmawla@gmail.com

## Abstract

In recent years, the rapid growth of the semantic web has encouraged the use of knowledge graphs to store and exchange data among computers. The change in data storage and retrieval has imposed the need to integrate other techniques that interact with these data. Machine Learning (ML) is the most important fields of study that interacts with data, to extract the patterns and relations that can be used in future applications. However, these techniques are implemented to be used with structured data, where each data instance is characterized using a set of predefined features. Thus, to apply machine learning techniques to knowledge graphs, it is important to convert the formation of the information in these graphs. The knowledge graph represents a collection of interconnected descriptions of entities (objects, events, situations, or abstract concepts in the real world). This work reviews the knowledge graphs and how they are structured and stored. Then, the techniques being used to employ machine learning technique to improve the knowledge graphs are illustrated. Additionally, the integration of the information in the knowledge graphs, so that, they can be used in machine learning techniques are also reviewed. This review can provide a rigid overview for future researchers to provide more attention on the main drawbacks in the existing techniques. Moreover, the review shows that the existing integration approaches limit the performance of the machine learning techniques, as the knowledge extracted from the knowledge graph is trimmed to match the requirements of the ML technique. In conclusion, new machine learning techniques that can extract knowledge directly from the knowledge graph can provide significantly better performance.

**Keywords:** Machine learning, Knowledge graph, Semantic web, Classification, Clustering

DOI: 10.7176/JSTR/5-8-01

## 1. Introduction

Machine Learning techniques attempt to enable computers to use examples collected from a certain domain to interact with that domain (Freitag, 2000). These techniques allow the extraction of knowledge that is too complex to be recognized by humans or hidden in an enormous number of examples. Such knowledge extraction increases the efficiency of the interaction between the computer and the domain it is interacting with, as the interaction rules are extracted based on the environment rather than static rules that reflect the expert's point of view (Burrell, 2016; LeCun, Bengio and Hinton, 2015).

Earlier, objects in an environment are described using a set of attributes, i.e. a vector that defines the value of each attribute in the environment (Buczak and Guven, 2016). However, with the rapid development in information technology, the characteristics of an object are being defined by defining the attributes of each object and the relations among these objects, semantically, in a certain domain. This representation has proposed the semantic web, where objects and relations among them are being defined

using graphs, known as knowledge graphs, instead of structured tables (Ristoski, Bizer and Paulheim, 2015; Ristoski and Paulheim, 2016).

Despite the more flexible and comprehensive representation in knowledge graphs, which allow analyzing data semantically, objects in the environment can no longer be represented using a predefined set of attributes. The definition of an object based on its relations to other others in the environment imposes the difficulty of defining that object to the ML techniques, which normally accept vectors of predefined size as their inputs (Chakkarwar and Joshi, 2016; Lausch, Schmidt and Tischendorf, 2015). However, according to the importance of machine learning, different approaches have been proposed to allow ML techniques to interact with knowledge graphs (Bühmann, Lehmann and Westphal, 2016).

In this study, knowledge graphs created for the semantic web, machine learning techniques and their applications, as well as the methods being used to integrate knowledge graphs with machine learning, are reviewed. The remainder of this paper is organized as follows: Section 2 describes knowledge graphs, their construction approaches and schemas. Section 3 reviews machine learning and how its techniques are categorized. Section 4 illustrates the applications that ML is employed to improve knowledge graphs. Section 5 describes the use of ML based on the knowledge extracted from knowledge graphs and how these graphs are converted into a suitable form for the existing ML techniques. Section 5 summarizes the conclusions of the paper.

## 2. Knowledge Graphs

In order to illustrate the challenges that the application of ML faces with knowledge graphs, it is important to describe how characteristics and relations among the entities in these graphs are defined and stores. Semantic networks, which are based on the knowledge representation and knowledge retrieval processes that computers acquire in the process of machine learning, are simply the relationships between the inferences shown by computers.

### 2.1. Knowledge Representation

Information in knowledge graphs is modeled based on the entities in the environment that the graph is representing and the relations among them. Although such representation has been employed, for decades, in semantic frames and network, the recent emphases on semantic web has brought significant attention to these graphs (Davis, Shrobe and Szolovits, 1993; Sowa, 2006). The main aim of the semantic web is to create the “web of data”, which allows computers to read each other’s data. Despite the lack of a complete vision of the semantic web, significant attention has been brought to the concept of linked data. This concept allows the data to be facilitated and linked on the web using Resource Description Framework (RDF), defined by the World Wide Web Consortium (W3C) (Carroll and Klyne, 2004; Cyganiak, Wood and Lanthaler, 2014).

RDF is a data model developed to express information about resources available on the World Wide Web. This model is a conceptual modeling approach as in the entity-relationship model and it is expressed in source-property-value trilogy called triples in RDF terminology. A resource is any entity being discussed. Each resource has a URI (Uniform Resource Identifier). The URI is the unique value for the resource. This value can be an internet address or an identification number. Properties are special types of resources, again defining the relationship between resources. Value is the value of the properties of resources. It can be a simple data type, or it can be used as a value in other URIs. The RDF / XML syntax standard is used for writing, displaying, and moving RDF expressions in XML (Sun and Han, 2012; Zeng et al., 2013).

### 2.2. Open and Closed World Assumptions

As illustrated earlier, the existence of a triple indicates the definite existence of the relation between the subject and object entities. However, interpreting the absence of a relation between two items depends on the type of the assumed world. In a Closed World Assumption (CWA) the absence of the triple that defines a relation between two entities indicates the definite absence of that relation. In such assumption, handling missing data can impose additional challenges to creating an accurate knowledge graph (Patel-Schneider, 2015).

In contrast, the absence of a triple that defines a certain relation between two entities in Open World Assumption (OWA) indicates that the existence of such relation is unknown, i.e. such relation may or may not exist. Such assumption allows the generation of more flexible knowledge graphs. For instance, keep in mind that even the birthplace attribute you typically think is known to be missing for 71% of all

people involved in Freebase. (Salam et al., 2018; Shi and Weninger, 2018). The ontology languages developed to realize the semantic web vision adopt an OWA. Considering that these languages are designed to be used in an environment where distributed and conflicting data can be found, such as the web, which contains dense and constantly changing data, the OWA will be more appropriate to these conditions.

### 2.3. Constructing Knowledge Bases

Different approaches are being adopted to construct knowledge bases. The usefulness of the produced knowledge base, which is reflected by the quality of the data in the knowledge base, their importance and accuracy, can vary depending on the approach adopted for constructing the knowledge base. These approaches can be classified into the following four categories (Nickel et al., 2016):

- **Curated:** A group of experts manually create the triples in the knowledge base. Despite the high accuracy of the curated knowledge bases, as they are manually created by experts, these knowledge bases suffer from limited scalability imposed by its dependency on human experts.
- **Collaborative:** A group of volunteers manually create the triples of the knowledge base. Although knowledge bases extracted using collaborative approaches scale better than the curated ones, their integrity and accuracy are questionable.
- **Semi-Structured:** Triples are extracted automatically from semi-structured text, by manually defining the extraction rules or allow the approach to learn these rules automatically. Studies, such as (Biega, Kuzey and Suchanek, 2013), have shown that knowledge bases extracted using this approach can have high accuracies but the use of structured text limits the completeness and coverage of the extracted knowledge base.
- **Unstructured:** Machine learning and natural language processing are used to automatically extract the triples from unstructured text. As more information can be extracted from unstructured text, this approach produces knowledge bases with higher coverage, compared to the use of structured text. Thus, more emphasis on using this approach, to generate knowledge bases, has been applied in recent years.

### 2.4. Open and Fixed Knowledge Base Schemas

Regardless of information presented in knowledge bases, there are two main types of schemas that are used to define the entities and relations among them, schema-based and schema-free. In schema-based approaches, a global dictionary is implemented to hold the names of all the entities and relation in the knowledge base. Then, when a triple is created, the unique identifiers of the components in the triple are collected from the dictionary. In contrast, the names of the entities and relation in a schema-free knowledge base are stated directly in the triple and no dictionary is required. Despite the ease and speed of interacting with triples in schema-free knowledge bases, synonyms cannot be recognized as the same entity or relation, as they have different names in their corresponding triples (Canim et al., 2017; Etzioni et al., 2011).

## 3. Machine Learning

To allow computers to interact with an external environment, the traditional approach is to provide a set of rules, defined by an expert, that can be applied to any input to select the appropriate output. Recently, more emphasis is being applied to allowing computers extracting these rules automatically, from examples collected from the environment. Such approach is known as Machine Learning and allows more flexible knowledge extraction, as this knowledge is directly extracted from the environment instead of being limited by the expert's point of view (Luo, 2016; Wuest et al., 2016). This knowledge is defined by the patterns and relations among the attributes of the inputs collected from the environment, where each input is defined by a set of attributes, i.e. features. Such patterns and relations can be too difficult to be noticed by an expert, according to the complexity of these patterns or rarely occur in the collected examples (Anjos et al., 2017). Mainly, ML techniques can be divided into two main categories, unsupervised and supervised:

- **Unsupervised:** In this type of ML, the examples collected from the environment, which is used for knowledge extraction, is inputted to the ML technique without any additional information. Unsupervised ML techniques are usually used to detect patterns and relation amongst the inputs (Gentleman and Carey, 2008). One of the popular unsupervised applications is clustering, where the inputted data is grouped based on the similarity among them, so that, an input in a certain group, i.e.

cluster, is less similar to any other input in another group than those in its group (Berkhin, 2006). Unsupervised creation of knowledge graph embeddings, e.g. the tasks from clustering and node comparison.

- **Supervised:** This type of ML requires embedding additional information to each input. Then, the supervised technique attempts to recognize the relation between the added information and the patterns in the collected inputs. The rules that can be used to detect these patterns are summarized in models that can be used to recognize the additional information that can be assigned to future inputs, based on the patterns detected in them (Caruana and Niculescu-Mizil, 2004). Classification is the most widely used type of supervised ML. Each of the example inputs collected from the environment is assigned with a label that represents the category that the example belongs to. Then, the classifier attempts to recognize the patterns in these inputs that define the characteristics of each category. The behavior of future unlabeled inputs can be predicted by passing their features values through the model created by the classifier, in order to predict their behavior or the appropriate output to interact with each input (Kotsiantis, Zaharakis and Pintelas, 2007). Supervised learning from a knowledge graph for prediction, e.g. multiple classification, regression, link estimation.

#### 4. Machine Learning for Knowledge Graphs

Machine learning techniques have been widely employed for creating and improving the knowledge graphs, as these graphs cannot present a complete model of the real-life relations among its entities. This section illustrates the main tasks that ML techniques are employed to create or refine knowledge graphs.

##### 4.1. Predicting Types and Relations

According to the impossibility of creating a complete knowledge graph that includes all the knowledge of the world, each of these graphs misses some pieces of that knowledge. As the completeness of the knowledge in the graph is one of the usefulness measures of that graph, completing any interpretable missing information can be of significant importance to improve the graph's quality. For instance, the Statistical Distribution (SDType) (Paulheim and Bizer, 2013) approach investigates the relations of each entity in the graph in order to predict its type. Thus, the existence of any triple "*Subject-Predicate-Object*" is investigated by collecting the type of all the Subject and Object entities from the graph. Then, the dominant type per each entity is examined as a suitable type for that entity. Some of the predicates can have very strict types as their subjects or object, such as the predicate "*hasCapital*", which normally has a subject of type "*Country*" and an object of type "*City*". Thus, the type of any entity that has no defined type but appears in a triple with "*hasCapital*" predicate can be predicted. However, some other predicates may not have such strict types for its entities, so that, not all predicates are suitable for such prediction.

By analyzing such a problem, from ML's point of view, an entity can have only one of multiple possible types, which are all the types existing in the graph. Thus, classification techniques can be used to predict that type, based on the relations the entity has with other entities. As the number of relations is not constant for all the entities, hierarchical classification is the most suitable for such task, according to its ability to use a subset of features to predict a label for the input. Thus, such classifier investigates the existence of relations that join this entity to another and can be used to predict a certain label. If such relations do not exist, another subset of relations is investigated to predict another label, i.e. type (Paulheim, 2018).

Another type of information that machine learning is being used, for knowledge graphs integration, is relations among the entities in the graph. The machine learning technique attempts to extract heuristic rules from the input text, based on the facts found in the knowledge graph, using local Closed World Assumption (Dong et al., 2014) or partial completeness assumption (Galárraga et al., 2013). These assumptions are used to generate negative examples from the graph using the relations that do not exist, as the existing relation produce positive example, to train the classifier. The method proposed in (Heist and Paulheim, 2017), for example, has learned that the first entity of type place that appears in the text after an entity of type person, is that person's place of birth.

##### 4.2. Error Detection

As illustrated earlier, besides curated knowledge graphs, the accuracy of the knowledge graphs cannot be guaranteed, according to noise imposed by inaccurate triples. However, as curated knowledge graphs are of limited scalability, a tradeoff between the scalability of the graph and its accuracy must be

addressed, by improving the accuracy of the other knowledge graphs creation approaches. Thus, machine learning techniques are being used to detect any errors in the created knowledge graphs.

Similar to SDType, an error detection method, based on machine learning, denoted as SDValidate (Paulheim and Bizer, 2014) is proposed to detect errors in the knowledge graph. With a similar methodology, the SDValidate relies on the types of the entities in the subject and object position per each predicate. Errors are detected by investigating anomaly in these subjects and object, depending on the predicate. However, as this method only relies on the type of the entity, it has limited error detection performance (Ringler and Paulheim, 2017). Another error detection approach, PaTyBRED (Paths and Types with Binary Relevance for Error Detection) (Melo and Paulheim, 2017), based on machine learning includes the connections between the entities in the subject and object positions of a predicate to detect the error. The inclusion of extra knowledge in error detection has been able to improve the predictions produced by this method, compared to SDValidate (Paulheim and Pan, 2012).

#### 4.3. Approximate Local Reasoning

One of the main features of knowledge graphs is the ability to conclude new relations based on the properties of the domain, defined in the ontology. For example, a bidirectional relation from a certain subject to another object can be used to reason the existence of the same relation from the object of the first relation to its subject. Additionally, the ontology can also define restrictions that must not be violated by the triples, such as disjoint classes. An entity in the knowledge graph of a type *A* that is disjoint from another type *B* is not allowed to be in a position of a triple that is defined to be of type *B*, by the predicate. Despite the ability of using standard reasoners to detect such error, the enormous number of triples in a knowledge graph imposes the need for extremely long processing time. Thus, machine learning is being used to detect such violations in shorter time. The method proposed in (Paulheim and Stuckenschmidt, 2016) processed the validation problem, based on the defined ontology, as a binary classification problem. As the output of the binary classifier is one of only two possible states, these states are assigned as consistent or inconsistent. Then, the classifier is trained using small batches collected from the knowledge graph and validated through the reasoner, so that, the classifier can learn to distinguish consistent from inconsistent triples, with the ontology. The results of the study show that a validation accuracy greater than 95% has been achieved by this method, while reducing the execution time to less than two hours, compared to several weeks required by the Hermit (Glimm et al., 2014) state-of-the-art reasoner.

### 5. Applying Machine Learning to Knowledge Graphs

With the rapid growth of machine learning and knowledge graphs usage, it has become essential to allow machine learning techniques to extract knowledge from these graphs. To allow such interaction, the information presented in the knowledge graph is processed and converted to a more suitable form for machine learning techniques. Then, machine learning techniques can be used to extract different types of knowledge from the graph.

#### 5.1. Knowledge Graph Conversion

As machine learning techniques normally accept vectorized data, i.e. a set of features describe each instance, and according to the dynamic topology of knowledge graphs, a pipeline to process the knowledge graphs for machine learning techniques is presented in Figure 1 (Bloem and De Vries, 2014).

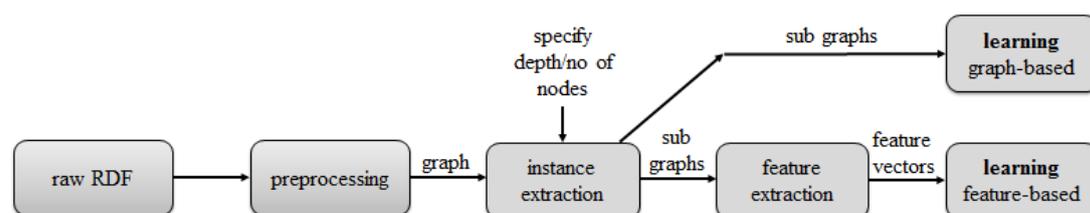


Figure 1. Overview of machine learning pipeline for RDF data.

As shown in Figure 1, this pipeline consists of the following main steps:

- **Preprocessing:** According to the verbosity of the information in a knowledge graph, the efficiency of the ML technique can be significantly improved by eliminating some of less-relevant relations. However, such elimination can impose some changes in the output knowledge, compared to the input, such as blank nodes. Thus, it is important to handle these changes and produce accurate output.
- **Instance Extraction:** Although the instances in a knowledge graph are described according to its resource, these resources do not provide the overall characteristics of that instance. Mainly, the relations between the instance, from one side, and other instances in the graph, from another, provide better description of that instance. Thus, different approaches are being used to characterize the instance, such as extracting a subgraph, up to a certain depth, around that instance (de Vries, 2013; Lösch, Bloehdorn and Rettinger, 2012).
- **Feature Extraction:** After extracting the instances from the knowledge graph, these instances are vectorized by extracting the features that characterize each instance. This step is the key step of this pipeline, as the machine learning techniques rely mainly on these features to extract the required knowledge and create the model that is used to interact with future inputs. Tensor decomposition (Nickel, Tresp and Kriegel, 2011) and Weisfeiler-Lehman algorithm (de Vries, 2013) are two of the most popular approaches being used for this purpose.

### 5.2. Performance Evaluation

To evaluate the performance of a machine learning technique, with structured datasets, a set of instances is extracted from the dataset for the evaluation, which is not included in the training phase. However, according to the dense interconnections among the instances in knowledge graphs, such extraction cannot be a straightforward process. Thus, the following two options are widely used to extract the evaluation data from the entire knowledge graph:

- **Instances Removal:** According to the difficulty of removing data from a knowledge graph, this method suggests the removal of certain instances from the dataset produced from the conversion process illustrated earlier. This approach also enables the evaluation using k-fold cross-validation, for unbiased performance measures (Loesch, Bloehdorn and Rettinger, 2012).
- **Using Real-World Graphs:** The removal of information added by experts to the graphs cannot be guaranteed, as such information can still persist in more complex relations in the graph. Thus, this approach suggests the exclusion of a certain part of the graph before adding any additional information, especially the one required from the machine learning technique.

### 5.3. Machine Learning From Knowledge Graph Applications

Machine learning techniques are being used for many applications, based on the knowledge extracted from the knowledge graph. Some of the applications that widely employ machine learning from knowledge graphs are:

- **Search Optimization:** According to the strict definition of the entities and relations in the knowledge graphs, search queries are required to match these strict rules to retrieve the required information. Machine learning is then used to provide more flexibility to these searches, so that, most relevant results can be retrieved even if they do not identically match the query. The method proposed in (Hadi et al., 2013) provides a probabilistic model to search knowledge graphs, using a classifier. First, the knowledge graph is converted into a three-dimensional array, where each of these dimensions represents the predicate, object and subject. Then, each predicate in the knowledge graph is extracted alongside with the corresponding information from the array, where predicates with less important information are eliminated to reduce the complexity of the model. The results of the study show that the use of nonlinear classifiers has achieved better results, compared to linear ones, which implies that the classes extracted from the graph, for the search purpose, are not of linear representation. This nonlinearity reflects the importance of using the machine learning techniques to achieve such probabilistic search on knowledge graphs.
- **Classification:** As illustrated earlier, the dynamic dimensionality of the entities in knowledge graphs is one of the most important challenges against applying machine learning to information in these graphs. Additionally, machine learning techniques, normally, require fixed-sized inputs, in order to achieve the intended task. Thus, to classify the individuals in a knowledge graph, it is important to describe each of these individuals using a fixed size vector that holds the most valuable features that represent the individual, from the classifier's point of view. In (Parundekar, 2018), to classify the

individuals in the knowledge graphs, the features are selected from the relations between the individual and other entities in the graphs, using random walk approach (Tsuda and Saigo, 2010). Using this approach, the features are extracted per an individual in the graph, by hopping over the four possible types of hops any individual can have, which are a relation to an attribute, incoming relation, outgoing relation or a step over the relation, which unlike the other three hops lands on a different individual. A feature that represents the individual can be represented by the extracted attributes the random walk hops through, which can be of the same or a different individual that shares a relation with the correspondent individual. Depending on the size of the required vector, the previous step can be repeated for the required number of features in that vector. The extracted features are then inputted to different classifiers to predict their types, where the results indicate that this approach has produced better results than the SDType approach. Thus, the random walk approach can be efficiently used to extract features, from knowledge graphs, for classification tasks.

- **Recommendations:** Based on the characteristics of a certain individual, a recommendation system selects the most relevant ones from the graph and recommend them to the user. Measuring the similarity between individuals in the graphs is challenged by the different number of relations the individual has, which define its characteristics. The method proposed in (Morshed, Dutta and Aryal, 2013) uses three clustering techniques to find these similarities. The output of two of these clustering techniques, Principle Component Analysis (PCA) and Fuzzy-C-Means (FCM), are used to evaluate the correlation of the relations in the graph, which is then used to guide the third technique, guided Self-Organizing Map (g-SOM) to find individuals from the knowledge graph that fall in the same cluster of the inputted individual. These individuals are considered the most relevant to the input individual and outputted as the recommendation of the system. Such approach can be used to increase the flexibility of the similarity measurement among the individuals in the knowledge graph, which can significantly improve the performance of many ML-based tasks.

## 6. Conclusion

With the rapid growth of the semantic web, the use of knowledge graphs to represent the information stored in the data and the relations among them is becoming of more significance. Additionally, many of the recent tasks are being implemented using machine learning techniques, according to their good performance. However, machine learning techniques normally require fixed-size inputs, where each input is characterized by a vector of features. As the individuals in a knowledge graph are defined based on their relations with other individuals, the extraction of a fixed-size vector per an individual is a challenging process.

In this study, the structure of knowledge graphs and the approaches being used to collect the data and create the corresponding graphs are reviewed. Then, an overview of machine learning techniques and how they can be employed for tasks that involve knowledge graphs, rather than the structured dataset, is provided. Finally, the methods currently being used to integrate the knowledge graphs and machine learning techniques with each other are described. The main contribution of this work is to provide future researchers with the required guidelines to integrate machine learning with knowledge graphs, as well as improving the integrations techniques. One of the most important fields that require more attention in this integration is the extraction of features of the individuals in the knowledge graph, so that, machine learning can be applied to them. Moreover, according to the review provided in this work, the implementation of machine learning techniques that can directly interact with the knowledge graphs, instead of the features extraction, is recommended, as the extraction of fixed-size vectors can eliminate some important features or include less important ones, according to the ML's point of view.

## References

- Anjos, A., Günther, M., de Freitas Pereira, T., Korshunov, P., Mohammadi, A., and Marcel, S. (2017). Continuously reproducing toolchains in pattern recognition and machine learning experiments, ICML 2017.
- Berkhin, P. (2006). A survey of clustering data mining techniques. Indian Journal of Science & Technology, 25-71 Springer.

- Biega, J., Kuzey, E., and Suchanek, F. M. (2013). *Inside YAGO2s: a transparent information extraction architecture*. Paper presented at the Proceedings of the 22nd International Conference on World Wide Web.
- Bloem, P., and De Vries, G. K. (2014). Machine learning on linked data, a position paper. *Linked Data for Knowledge Discovery*, 15-19.
- Buczak, A. L., and Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys & Tutorials*, 18(2), 1153-1176.
- Bühmann, L., Lehmann, J., and Westphal, P. (2016). DL-Learner—A framework for inductive learning on the Semantic Web. *Journal of Web Semantics*, 39, 15-24.
- Burrell, J. (2016). How the machine ‘thinks’: Understanding opacity in machine learning algorithms. *Big Data & Society*, 3(1).
- Canim, M., et al. (2017). A knowledge and reasoning toolkit for cognitive applications. Proceedings of the fifth ACM/IEEE Workshop on Hot Topics in Web Systems and Technologies.
- Carroll, J. J., and Klyne, G. (2004). Resource description framework ({RDF}): Concepts and abstract syntax.
- Caruana, R., and Niculescu-Mizil, A. (2004). Data mining in metric space: an empirical analysis of supervised learning performance criteria. Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining.
- Chakkarwar, V., and Joshi, A. A. (2016). Semantic Web Mining using RDF Data. *International Journal of Computer Applications*, 975, 8887.
- Cygniak, R., Wood, D., and Lanthaler, M. (2014). RDF 1.1 Concepts and Abstract Syntax , web sites: <https://www.w3.org/TR/rdf11-concepts/>.
- Davis, R., Shrobe, H., and Szolovits, P. (1993). What is a knowledge representation? *AI magazine*, 14(1), 17-17.
- de Vries, G. K. (2013). *A fast approximation of the Weisfeiler-Lehman graph kernel for RDF data*. Paper presented at the Joint European Conference on Machine Learning and Knowledge Discovery in Databases.
- Dong, X., et al. (2014). *Knowledge vault: A web-scale approach to probabilistic knowledge fusion*. Paper presented at the Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining.
- Etzioni, O., Fader, A., Christensen, J., and Soderland, S. (2011). *Open information extraction: The second generation*. Paper presented at the Twenty-Second International Joint Conference on Artificial Intelligence.
- Freitag, D. (2000). Machine learning for information extraction in informal domains. *Machine learning*, 39(2-3), 169-202.
- Galárraga, L. A., Teflioudi, C., Hose, K., and Suchanek, F. (2013). *AMIE: association rule mining under incomplete evidence in ontological knowledge bases*. Paper presented at the Proceedings of the 22nd international conference on World Wide Web.
- Gentleman, R., and Carey, V. (2008). Unsupervised machine learning. In *Bioconductor Case Studies* (pp. 137-157): Springer.

- Glimm, B., Horrocks, I., Motik, B., Stoilos, G., and Wang, Z. (2014). Hermit: an OWL 2 reasoner. *Journal of Automated Reasoning*, 53(3), 245-269.
- Hadi, A. S., Fergus, P., Dobbins, C., and Al-Bakry, A. M. (2013). *A machine learning algorithm for searching vectorised RDF data*. Paper presented at the 2013 27th International Conference on Advanced Information Networking and Applications Workshops.
- Heist, N., and Paulheim, H. (2017). *Language-agnostic relation extraction from wikipedia abstracts*. Paper presented at the International Semantic Web Conference.
- Kotsiantis, S. B., Zaharakis, I., and Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160, 3-24.
- Lausch, A., Schmidt, A., and Tischendorf, L. (2015). Data mining and linked open data—New perspectives for data analysis in environmental research. *Ecological Modelling*, 295, 5-17.
- LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553), 436.
- Lösch, U., Bloehdorn, S., and Rettinger, A. (2012). *Graph kernels for RDF data*. Paper presented at the Extended Semantic Web Conference.
- Luo, G. (2016). A review of automatic selection methods for machine learning algorithms and hyperparameter values. *Network Modeling Analysis in Health Informatics and Bioinformatics*, 5(1), 18.
- Melo, A., and Paulheim, H. (2017). *Detection of relation assertion errors in knowledge graphs*. Paper presented at the Proceedings of the Knowledge Capture Conference.
- Morshed, A., Dutta, R., and Aryal, J. (2013). *Recommending environmental knowledge as linked open data cloud using semantic machine learning*. Paper presented at the 2013 IEEE 29th International Conference on Data Engineering Workshops (ICDEW).
- Nickel, M., Murphy, K., Tresp, V., and Gabrilovich, E. (2016). A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1), 11-33.
- Nickel, M., Tresp, V., and Kriegel, H.-P. (2011). *A Three-Way Model for Collective Learning on Multi-Relational Data*. Paper presented at the ICML.
- Parundekar, R. (2018). Classification of Things in DBpedia using Deep Neural Networks. *arXiv preprint arXiv:1802.02528*.
- Patel-Schneider, P. F. (2015). *Using description logics for RDF constraint checking and closed-world recognition*. Paper presented at the Twenty-Ninth AAAI Conference on Artificial Intelligence.
- Paulheim, H. (2018). *Machine learning with and for semantic web knowledge graphs*. Paper presented at the Reasoning Web International Summer School.
- Paulheim, H., and Bizer, C. (2013). *Type inference on noisy RDF data*. Paper presented at the International semantic web conference.
- Paulheim, H., and Bizer, C. (2014). Improving the quality of linked data using statistical distributions. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 10(2), 63-86.
- Paulheim, H., and Pan, J. Z. (2012). Why the semantic web should become more imprecise. *What will the Semantic Web look like*, 10.

- Paulheim, H., and Stuckenschmidt, H. (2016). *Fast approximate a-box consistency checking using machine learning*. Paper presented at the European Semantic Web Conference.
- Ringler, D., and Paulheim, H. (2017). *One knowledge graph to rule them all? Analyzing the differences between DBpedia, YAGO, Wikidata & co.* Paper presented at the Joint German/Austrian Conference on Artificial Intelligence (Künstliche Intelligenz).
- Ristoski, P., Bizer, C., and Paulheim, H. (2015). Mining the web of linked data with rapidminer. *Web semantics: science, services and agents on the World Wide Web*, 35, 142-151.
- Ristoski, P., and Paulheim, H. (2016). Semantic Web in data mining and knowledge discovery: A comprehensive survey. *Web semantics: science, services and agents on the World Wide Web*, 36, 1-22.
- Salam, S., Voit, E. A., Clemm, L. A., and Warnicke, E. A. (2018). Network control and management using semantic reasoners in a network environment. In: US Patent App. 14/339,312.
- Shi, B., and Wenginger, T. (2018). *Open-world knowledge graph completion*. Paper presented at the Thirty-Second AAAI Conference on Artificial Intelligence.
- Sowa, J. F. (2006). Semantic Networks. In *Encyclopedia of Cognitive Science*.
- Sun, Y., and Han, J. (2012). Mining heterogeneous information networks: principles and methodologies. *Synthesis Lectures on Data Mining and Knowledge Discovery*, 3(2), 1-159.
- Tsuda, K., and Saigo, H. (2010). Graph classification. In *Managing and mining graph data* (pp. 337-363): Springer.
- Wuest, T., Weimer, D., Irgens, C., and Thoben, K.-D. (2016). Machine learning in manufacturing: advantages, challenges, and applications. *Production & Manufacturing Research*, 4(1), 23-45.
- Zeng, K., Yang, J., Wang, H., Shao, B., and Wang, Z. (2013). *A distributed graph engine for web scale RDF data*. Paper presented at the Proceedings of the VLDB Endowment.