BILINGUAL PUBLISHING CO.
Pioneer of Global Academics Since 1984

Artificial Intelligence Advances
https://ojs.bilpublishing.com/index.php/aia/index

ARTICLE

Machine Learning Meets the Semantic Web

Konstantinos Ilias Kotis* **Konstantina Zachila** **Evaggelos Paparidis**

University of the Aegean, 81100, Greece

ARTICLE INFO

Article history
Received: 29 April 2021
Accepted: 31 May 2021
Published Online: 10 June 2021

Keywords:
Knowledge graph
Semantic web
Ontology
Machine learning
Deep learning
Graph neural networks

ABSTRACT

Remarkable progress in research has shown the efficiency of Knowledge Graphs (KGs) in extracting valuable external knowledge in various domains. A Knowledge Graph (KG) can illustrate high-order relations that connect two objects with one or multiple related attributes. The emerging Graph Neural Networks (GNN) can extract both object characteristics and relations from KGs. This paper presents how Machine Learning (ML) meets the Semantic Web and how KGs are related to Neural Networks and Deep Learning. The paper also highlights important aspects of this area of research, discussing open issues such as the bias hidden in KGs at different levels of graph representation.

1. Introduction

Due to the large volume of data on the Web, there is a growing interest in KGs, playing an important role in applications such as Search Engines. KGs were first introduced by Google in 2012 and used as an Internet search strategy. Using these graphs, simple word processing has become a symbolic representation of knowledge. KGs are also used by social networking and e-commerce Web applications and are of particular interest to the Semantic Web (SW) community [1]. Although there is no single definition of a KG, it can be defined as a way of representing a database of interconnected descriptions of real-world entities and events or abstract concepts.

Although KGs are easily understood by humans and contain high-level information about the world, it is difficult to exploit them for ML, one of the more significant research fields of Artificial Intelligence. Its goal is to create systems that can be trained from empirical/sample data that they have observed in the past, to perform the work for which they are intended more effectively by analyzing a vast amount of operational data [2].

ML and KGs combination is fast-moving. On the one hand, ML techniques improve the performance of various data-driven tasks with great accuracy. On the other hand, KGs provide the competence to represent knowledge about entities and their relationships with high reliability, explanation, and reuse. Consequently, a combination of KGs and ML will systematically improve systems accuracy, explainability, and reuse, expanding the limits of ML capabilities [3].

This paper is structured as follows: Section 2 presents KGs, while Section 3 discusses KGs and Ontologies' relation. The connection between KGs and ML is presented

**Corresponding Author:*
Konstantinos Ilias Kotis,
University of the Aegean, 81100, Greece;
Email: kotis@aegean.gr

 DOI: https://doi.org/10.30564/aia.v3i1.3178

in Section 4. Section 5 discusses the association of KGs to Deep Learning and Section 6 discusses open issues and challenges in this domain. Finally, Section 7 concludes the paper.

2. Knowledge Graphs

There have been several attempts to definite what a Knowledge Graph is. Due to the different definitions already present in the literature [1], some inconsistencies have been inevitably merged. In addition to the definition in Wikipedia, other newer and important definitions have been proposed by various researchers [4,5,6]. A knowledge graph is the organization and representation of a knowledge base (KB) as a multidomain graph, whose nodes represent entities of interest, combining different sources of controlled vocabularies and data. An example is illustrated in Figure 1.

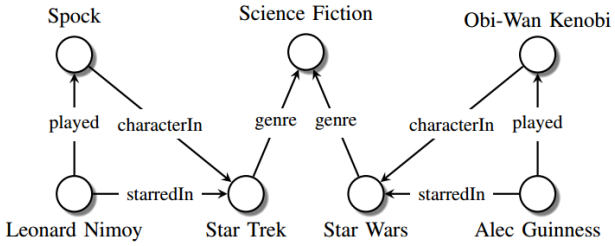

Figure 1. Knowledge graph example. Nodes represent entities, edge labels represent types of relations, edges represent existing relationships.

According to Ehrlinger [7], a KG obtains and incorporates information of an ontology and utilizes a reasoner to derive new knowledge. This definition assumes that a KG is considered more advanced than a KB because a) it uses a reasoning engine in order to generate new knowledge and b) supports more than one information sources. This definition does not consider the size factor, as it is not clear what a ”large” graph is.According to Farber [8] and Huang [9], a KG is defined as being a graph represented with Resource Description Framework (RDF).According to Paulheim [10], KGs are considered to cover a notable part of the domains that exist in the world and are not defined to be limited to only one.According to Bizer [11], KGs provide an opportunity to explore in more depth the understanding of how knowledge can be managed on the Web and how the knowledge gained from it is broken down into more accustomed Web-based data publishing schemes like Linked Data.

Some of the most well-known non-public KGs are Google KG, Microsoft KG, and Facebook KG. There are also other well-known and widely used KGs available to the public, such as DBpedia and Wikidata KGs. There are not existing explicit references for most KGs, regarding the methods of extracting knowledge that KGs use, nor about their general shape, visualization, and storage of all this knowledge [12]. There are mainly two approaches to creating a KG: a) top-down (schema-driven) and b) bottom-up (data-driven). In the first approach, the ontology and the schema of the graph are first defined and then the data are entered. In the second, knowledge is extracted from various sources (e.g., text documents, databases, as well as from linked open data) and after being merged, the schema of the KG is constructed.

Facebook KG is an essential tool enabling internal search within the Facebook platform to produce more and more accurate results when used by connected users. Google KG is a valuable tool of Google, which contains information from various sources like Wikipedia in order to produce better and more complete results in the search engine. DBpedia KG is a huge knowledge base created by processing information from Wikipedia for the purpose of making it available on the Web of Data. However, in Wikipedia, because of the plethora of pages it contains, and especially those in different languages, there are contradictions, creating inaccuracies in the information. The problem of managing this data was solved by Wikidata KG, where all languages were integrated into one version of Wikidata so that information could be linked to multiple languages at the same time. It also allows the existence of conflicting information by providing a system that organizes everything properly [8].

3. Knowledge Graphs and Ontology

A domain ontology is a formal and explicit specification of shared and agreed conceptualizations that are related to specific domain e.g., an ontology for a museum, an ontology for security, an ontology for surveillance. An ontology may define only the schema of the represented knowledge (classes, relations between them, and class restriction axioms) or the schema and the actual data that are semantically described by the defined schema. In the second case, the ontology is a populated one i.e., an ontology with populated classes. In other words, ontology is a knowledge base that stores knowledge about domain-specific entities, and those entities are classified as instances/individuals of its ontological classes.

A KG and a populated ontology are similar in a way. They are both related to the Resource Description Framework (RDF) for representing their data. They may both represent domain knowledge using semantic relations (links/edges) between entities (nodes). Knowledge about entities is represented by a statement in the form of a triple i.e., subject, predicate, object (SPO), where predicate

DOI: https://doi.org/10.30564/aia.v3i1.3178

is a relationship between the other two entities, as presented in Figure 2.

However, KGs and populated ontology (ontology-based knowledge bases) have some differences in respect to their aim. KGs often contain large volumes of factual information (facts about represented entities) with less formal semantics (class restriction axioms, definitions). On the other hand, an ontology defines the terminology of the domain and the semantic relations between terms, making knowledge available for machine processing, whereas data is not the main concern at its design time. In addition, KGs can also represent knowledge about multiple domains and therefore may contain more than one ontology [13].

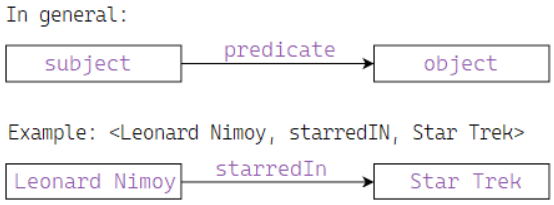

Figure 2. A triple example representing the statement that Leonard Nimoy starred in Star Trek movie.

4. Knowledge Graphs and Machine Learning

ML is a way for the machine to become intelligent by learning from the data that is provided as input to a process of data analysis, thus evolving a machine that performs tasks into an intelligent machine. An ML method generally means a set of specific types of algorithms that are suitable for solving a particular type of computational problem. Such a method addresses any constraints that the problem brings along with it.

The most popular ML methods are supervised learning, semi-supervised learning, and reinforcement learning. The learning process consists of three stages: the acquisition of data, the processing/analysis of data so as to find possible generalizations or specializations, and the use of processing results to perform the objective work.

ML in KGs and ontology are used to provide solutions to the type and link prediction, ontology enrichment, and integration [2]. In particular, because abstract reasoning was not applicable, and at the same time, while the ontology were consistent, the information in it was incorrect in relation to a reference domain, ML methods were developed for the Semantic Web (SW) to resolve this topic.

Large-scale KGs, i.e., knowledge-based graphs, store real information as relationships between entities. In an automated knowledge base construction method, triples are automatically extracted from unstructured text through ML and Natural Language Processing (NLP) techniques. In recent years, automated methods have been gaining more attention, because all other methods either have several limitations or do not scale well due to their dependence on human experts [3].

The Semantic Web technology targets to make the Web readable from the machines [14], by enriching resources with metadata. To manage these metadata, the OWL format is used while the reasoning capabilities —provided by the ontology— are also taken into account. However, metadata management process meets significant limitations —especially in cases of linked data— that include but are not limited to the time consumption for ontology construction, inconsistent and noisy data.

Hence, problems of query answering, instance retrieval, link prediction, and ontology refinement and enrichment have emerged with the three firsts considered by ML methods as classification problems, while the last as a concept learning problem. ML methods' introduction to solving classification and concept learning problems on the Semantic Web domain considers the advantages of the numeric-based (sub-symbolic) approaches such as Deep Learning [15] and embeddings. These ML methods are generally categorized into two categories, symbol and numeric-based.

The symbol-based category consists of methods that aim to tackle the Semantic Web problem from the viewpoint of reasoning. This includes methods that aim to deal with the problems related to (i) instance retrieval, (ii) concept learning for ontology enrichment, (iii) knowledge completion, and (iv) learning disjointness axioms. The instance retrieval problem defined as the assessment if an individual is an instance of a given concept and has been solved as a classification problem. Similarity-based methods such as K-Nearest Neighbor (KNN) and Support Vector Machines (SVM) have been proposed in early years [16,17,18], while later, more intelligible methods, based on Terminological Decision Tree (TDT), have been also proposed [19,20].

Concept learning for ontology enrichment problem is focused on the enrichment of ontology and of learning concepts descriptors. This problem is managed as a supervised concept learning problem at approximating an intentional Descriptions Logics representation. There are a number of methods related to this category [21,22,23,24,25]. The problem of knowledge completion aims to find information missing from the knowledge base. An indicative method that tackles this problem is AMIE [26,27]. AMIE target to mine logic rules from RDF knowledge bases in order to further predict new assertions. In a recent work [27], the system targets to mine SWRL rules from OWL ontolo-

 DOI: https://doi.org/10.30564/aia.v3i1.3178

gy. Learning disjointness axioms methods aim to discover axioms from the data that during the modelling process are overlooked and lead to misunderstanding the negative knowledge of the target domain. Indicative methods that tackle this problem are proposed in other works [28,29]. These methods study the correlation between the classes, the negative and association rules, and the correlation coefficient.

The numeric-based category consists of methods target to link prediction problem. This problem is solved as a classification problem and refers to the existence or not of triplets, usually in RDF format. Two types of models have been proposed to address this problem so far, the probabilistic latent models and the embedding models. A probabilistic latent model's indicative method is the Infinite Hidden Semantic Model (IHSM) [30]. This model formalizes a probabilistic latent variable that associates a latent class variable with each resource/node and makes use of constraints expressed in First-Order Logic during the learning process. Similarly, to the probabilistic models, embedding models represent each resource/node with a continuous embedding vector encoding its intrinsic latent features within the data collection. An indicative method embedding model has been proposed [31], named RESCAL, which implements graph embedding by computing a three-way factorization of an adjacency tensor representing the multi-graph structure of the data collection.

5. Knowledge Graphs And Deep Learning

Over the last few years Deep Learning has introduced a great number of machine-learning tasks, ranging from image scene classification and video analysis to natural speech and language understanding and recognition. The data used within these tasks are characterized by various data types such as images, voice signals, feature vectors, and are typically represented in the Euclidean space. Deep Learning approaches successfully manage the aforementioned data types build upon variant deep network architectures include but are not limited to Convolutional Neural Networks (CNNs) [32], Recurrent Neural Networks (RNNs) [33], Long Short-Term Memory (LSTM) [34] and auto-encoders [35]. Opposite to the architectures mentioned, there are a great number of deep learning-based architectures that their data used forms are projected to a non-Euclidean space, especially in the form of KGs [36]. This data structure overcomes the restrictions of interaction assumption following a linking approach. Thus, during the linking between the items and their attributes, each node is linked with various nodes and variant types.

Connecting nodes in the KG may have distinct neighborhood size, and the relationships between them could vary as well. The need to handle the above complexity of KGs stimulates new neural network architectures mentioned as Graph Neural Networks (GNNs) [37].

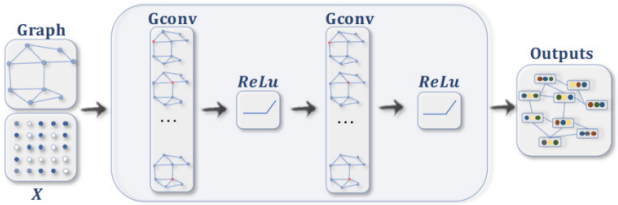

Figure 3. A ConvGNN with two graph convolutional layers. Each convolutional layer encapsulates each node's hidden representation by aggregating feature information from its neighbors. ReLu [48] activation function is applied to the resulted outputs. The final hidden representation of each node receives messages from a further neighborhood[37].

Graph Neural Networks were initially mentioned in 2005 in the work of Gori et al. [38] and later extended by the work of Scarselli et al. (2009) [39], and Gallicchio et al. (2010) [40]. According to the above works, Graph Neural Networks learn a target node's representation based on propagating the information to one or many neighbors in a recurrent way until a stable fixed point is reached. This process lacks computational efficiency; thus, recently, there have been increasing efforts to overcome this limitation [41,42]. These studies belong to the category of Recurrent Graph Neural Networks (RecGNNs). Considering the success of Convolutional Neural Networks in the computer vision domain, many methods that adapt the notion of convolution to the graph data have been developed. These methods belong to the research field of Convolutional Graph Neural Networks (ConvGNNs). An example of a ConvGNNs architecture is illustrated in Figure 3. Methods based on Convolutional Graph Neural Networks are generally separated into two distinct categories, the spectral-based and the spatial-based. The indicative method that belongs to the spectral-based category is the one proposed by Bruna et al. (2013) [43]. Later related works [44,45,46,47] proposed methods in order to further improve and extend spectral-based Convolutional Graph Neural Networks.

The research of spatial-based Convolutional Graph Neural Networks started much earlier than spectral based Convolutional Graph Neural Networks. In 2009, Micheli et al. [49] first addressed graph mutual dependency by architecturally composite non recursive layers, while inheriting ideas of message passing from Recurrent Graph Neural Networks. Although, the significance of this work was overlooked, until recently, many spatial-based Convolutional Graph Neural Networks [50,51,52] have been

DOI: https://doi.org/10.30564/aia.v3i1.3178

emerged. Apart from Recurrent Graph Neural Networks and Convolutional Graph Neural Networks, many alternative Graph Neural Networks have been proposed in the past few years, including but are not limited to Graph Autoencoders (GAEs), and Spatial-Temporal Graph Neural Networks (STGNNs). These learning frameworks can be built on Recurrent Graph Neural Networks, Convolutional Graph Neural Networks, or other neural architectures for graph modeling.

Wu et al. [37] presented a taxonomy of graph neural networks in the context of a comprehensive study of Graph Neural Networks. The Graph Neural Network architectures are categorized into four categories. The first category consists of the Recurrent Graph Neural Networks (RecGNNs) that mostly are pioneer works of Graph Neural Networks. Recurrent Graph Neural Networks aim to learn node representations with recurrent neural architectures. They assume that a node in a graph constantly exchanges information with its neighbors until a stable equilibrium is reached. Recurrent Graph Neural Networks are conceptually important and inspired later research on Convolutional Graph Neural Networks. In particular, the idea of message passing is inherited by Spatial-Based Convolutional Graph Neural Networks.

The second category is related to Convolutional Graph Neural Networks (ConvGNNs) that aim to generalize the operation of convolution from grid data to graph data. The main idea is to generate a node representation by aggregating its own features and neighbors' features. Different from Recurrent Graph Neural Networks, Convolutional Graph Neural Networks stack multiple graph convolutional layers to extract high-level node representations. Convolutional Graph Neural Networks proved beneficial to be the base for developing more complex Graph Neural Network models.

The third category consists of the Graph Auto-Encoders (GAEs) that benefits from the lack of the need for labelled data and subsequently are following unsupervised learning training. Besides, these frameworks encode nodes/graphs into a latent vector space and reconstruct graph data from the encoded information. Graph Auto-Encoders are used to learn network embeddings and graph generative distributions. For network embedding, Graph Auto-Encoders learn latent node representations through reconstructing graph structural information such as the graph adjacency matrix. For graph generation, some methods generate nodes and edges of a graph step by step, while other methods output a graph all at once.

The last category embodies the Spatial-Temporal Graph Neural Networks (STGNNs). The goal of the methods belongs to this category is to learn hidden patterns from spatiotemporal graphs and subsequently to be applied to variant applications related to traffic speed forecasting [53], driver maneuver anticipation [54], and human action recognition [55]. The main idea of Spatial-Temporal Graph Neural Networks is to consider spatial dependency and temporal dependency at the same time. Many current approaches integrate graph convolutions to capture spatial dependency with Recurrent Neural Networks or Convolutional Neural Networks to model the temporal dependency.

6. Open Issues and Challenges

One of the major recent issues and challenges in KG and ML research is bias. In general, research bias concerns the interference in the results of research (mainly in AI research) by predetermined ideas. Data in ML algorithms used in AI systems can be biased, but so can the algorithms that analyze it. Both data and algorithms are created by people, and people are usually subjective. When data is subjective/biased, data samples are not perfect representatives of their relative datasets involved in algorithmic analysis. A recent case of representational bias is Google's image search for "CEO", depicting mostly males that can be used to "teach" an intelligent system to recommend doctors as a career choice for men and nurse for women.

ML/DL community work about how to address representational biases have not yet reached the KG and Semantic Web communities. The current status of Linked Open Data (LOD) cloud may be free of sampling bias. However, the data available to both open and commercial KGs today is, in fact, highly biased. Debiasing KGs (data and schema) will soon become a major issue as these are now rapidly used in several ML-based algorithms of AI systems and applications.

Debiasing KGs must be examined at the level of data (data bias) as well as the level of schema (schema/ontology bias). Entities represented in DBpedia's KG, for instance, either spatial or non-spatial, do not cover the global range of available data (all the world as we know it). Instead, the coverage of data related to Europe- and US-based entities is clearly larger than Asia's. On the other hand, bias at the schema/ontology level, is also highly possible, since most of the ontologies are engineered following a top-down methodological approach, often with application needs in mind. In such cases, knowledge engineers/workers and domain experts collaborate, propagating their subjective engineering choices in the ontological design patterns (human-centered approach). Last but not least, if a bottom-up (data-driven) ontological engineering approach is followed, for instance, based on ML algorithms that extract/learn on-

 DOI: https://doi.org/10.30564/aia.v3i1.3178

tological axioms/rules from available sample data, the bias problem remains as it is probably propagated from biased data, as discussed earlier.

Therefore, a new methodology for the engineering of bias-free KGs is required, supported by suitable tools for managing KGs both at the level of data and schema, aligned with modern policies/rules for AI bias elimination. Based on such a methodology (specifying distinct phases, processes and tasks), it is expected that unbiased AI applications will prevail. From a mis-rejected job application to the false arrest of an innocent fellow and the misidentifying of the actual criminal or threat, debiasing AI applications must be a priority and a continuous concern of actions to be taken in the era of AI applications that are highly based on ML/DL and KGs.

7. Conclusions

In this paper, the interconnection of KGs and ML/DL has been presented, and important applications in several fields are discussed. In addition, the interrelation between ontology and KGs has been pointed out. A detailed representation of both symbol and numeric-based ML methods has been provided, in order to overview the aforementioned connections. Deep learning and neural networks are related to KGs via Graph Neural Networks, which have become powerful and practical tools for ML tasks in the graph domain. In particular, the paper presents a categorization of Graph Neural Networks into Recurrent Graph Neural Networks, Convolutional Graph Neural Networks, Graph auto-encoders, and Spatial-temporal Graph Neural Networks. Last but not least, the paper discusses open issues and challenges in this research domain, highlighting the importance of KGs bias at both the schema (ontology) and the data level. It has been argued that, only if proper attention is given in the debiasing of KGs, ML/DL-based AI applications will really prevail.

References

[1] P. A. Bonatti, S. Decker, A. Polleres, and V. Presutti, "Knowledge graphs: New directions for knowledge representation on the semantic web (dagstuhl seminar 18371)," in Dagstuhl Reports, vol. 8, no. 9. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

[2] C. d'Amato, "Machine learning for the semantic web: Lessons learnt and next research directions," Semantic Web, no. Preprint, pp. 1-9, 2020.

[3] M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich, "A review of relational machine learning for knowledge graphs," Proceedings of the IEEE, vol. 104, no. 1, pp. 11-33, 2015.

[4] E. Marchi and O. Miguel, "On the structure of the teaching-learning interactive process," International Journal of Game Theory, vol. 3, no. 2, pp. 83-99, 1974.

[5] H. van den Berg, "First-order logic in knowledge graphs," Current Issues in Mathematical Linguistics, vol. 56, pp. 319-328, 1993.

[6] R. R. Bakker, Knowledge Graphs: representation and structuring of scientific knowledge, 1987.

[7] L. Ehrlinger and W. Woß, "Towards a definition of knowledge graphs."¨ SEMANTiCS (Posters, Demos, SuCCESS), vol. 48, pp. 1-4, 2016.

[8] M. Farber, F. Bartscherer, C. Menne, and A. Rettinger, "Linked data¨ quality of dbpedia, freebase, opencyc, wikidata, and yago," Semantic Web, vol. 9, no. 1, pp. 77-129, 2018.

[9] Z. Huang, J. Yang, F. van Harmelen, and Q. Hu, "Constructing diseasecentric knowledge graphs: a case study for depression (short version)," in Conference on Artificial Intelligence in Medicine in Europe. Springer, 2017, pp. 48-52.

[10] H. Paulheim, "Knowledge graph refinement: A survey of approaches and evaluation methods," Semantic web, vol. 8, no. 3, pp. 489-508, 2017.

[11] C. Bizer, T. Heath, and T. Berners-Lee, "Linked data: The story so far," in Semantic services, interoperability and web applications: emerging concepts. IGI global, 2011, pp. 205-227.

[12] Z. Zhao, S.-K. Han, and I.-M. So, "Architecture of knowledge graph construction techniques," International Journal of Pure and Applied Mathematics, vol. 118, no. 19, pp. 1869-1883, 2018.

[13] J. P. McCusker, J. Erickson, K. Chastain, S. Rashid, R. Weerawarana, and D. McGuinness, "What is a knowledge graph," Semantic Web Journal, 2018.

[14] C. d'Amato, N. Fanizzi, and F. Esposito, "Inductive learning for the semantic web: what does it buy?" Semantic Web, vol. 1, no. 1, 2, pp. 53-59, 2010.

[15] L. Deng and D. Yu, "Deep learning: methods and applications," Foundations and trends in signal processing, vol. 7, no. 3-4, pp. 197-387, 2014.

[16] C. d'Amato, N. Fanizzi, and F. Esposito, "Query answering and ontology population: An inductive approach," in European Semantic Web Conference. Springer, 2008, pp. 288-302.

[17] A. Rettinger, U. Losch, V. Tresp, C. d'Amato, and N. Fanizzi, "Mining¨ the semantic web," Data Mining and Knowledge Discovery, vol. 24, no. 3, pp. 613-662, 2012.

[18] S. Bloehdorn and Y. Sure, "Kernel methods for mining instance data in ontologies," in The Semantic Web. Springer, 2007, pp. 58-71.

[19] N. Fanizzi, C. d'Amato, and F. Esposito, "Induction

DOI: https://doi.org/10.30564/aia.v3i1.3178

of concepts in web ontologies through terminological decision trees," in Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, 2010, pp. 442-457.

[20] G. Rizzo, N. Fanizzi, C. d'Amato, and F. Esposito, "Approximate classification with web ontologies through evidential terminological trees and forests," International Journal of Approximate Reasoning, vol. 92, pp. 340-362, 2018.

[21] N. Fanizzi, C. d'Amato, and F. Esposito, "Dl-foil concept learning in description logics," in International Conference on Inductive Logic Programming. Springer, 2008, pp. 107-121.

[22] A. C. Tran, J. Dietrich, H. W. Guesgen, and S. Marsland, "An approach to parallel class expression learning," in International Workshop on Rules and Rule Markup Languages for the Semantic Web. Springer, 2012, pp. 302-316.

[23] J. Lehmann, S. Auer, L. Buhmann, and S. Tramp, "Class expression¨ learning for ontology engineering," Journal of Web Semantics, vol. 9, no. 1, pp. 71-81, 2011.

[24] G. Rizzo, N. Fanizzi, C. d'Amato, and F. Esposito, "A framework for tackling myopia in concept learning on the web of data," in European Knowledge Acquisition Workshop. Springer, 2018, pp. 338-354.

[25] A. C. Tran, J. Dietrich, H. W. Guesgen, and S. Marsland, "Parallel symmetric class expression learning," The Journal of Machine Learning Research, vol. 18, no. 1, pp. 2145-2178, 2017.

[26] F. Baader, D. Calvanese, D. McGuinness, P. Patel-Schneider, D. Nardi et al., The description logic handbook: Theory, implementation and applications. Cambridge university press, 2003.

[27] C. d'Amato, A. G. Tettamanzi, and T. D. Minh, "Evolutionary discovery of multi-relational association rules from ontological knowledge bases," in European knowledge acquisition workshop. Springer, 2016, pp. 113-128.

[28] J. Volker, D. Fleischhacker, and H. Stuckenschmidt, "Automatic acqui-¨ sition of class disjointness," Journal of Web Semantics, vol. 35, pp. 124-139, 2015.

[29] J. Volker and M. Niepert, "Statistical schema induction," in¨ Extended Semantic Web Conference. Springer, 2011, pp. 124-138.

[30] A. Rettinger, M. Nickles, and V. Tresp, "Statistical relational learning with formal ontologies," in Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, 2009, pp. 286-301.

[31] M. Nickel, V. Tresp, and H.-P. Kriegel, "A three-way model for collective learning on multi-relational data," in Icml, 2011.

[32] Y. LeCun, Y. Bengio et al., "Convolutional networks for images, speech, and time series," The handbook of brain theory and neural networks, vol. 3361, no. 10, p. 1995, 1995.

[33] J. Schmidhuber and S. Hochreiter, "Long short-term memory," Neural Comput, vol. 9, no. 8, pp. 1735-1780, 1997.

[34] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8, pp. 1735-1780, 1997.

[35] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, and L. Bottou, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion." Journal of machine learning research, vol. 11, no. 12, 2010.

[36] Y. Gao, Y.-F. Li, Y. Lin, H. Gao, and L. Khan, "Deep learning on knowledge graph for recommender system: A survey," arXiv preprint arXiv:2004.00387, 2020.

[37] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," IEEE transactions on neural networks and learning systems, 2020.

[38] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005., vol. 2. IEEE, 2005, pp. 729-734.

[39] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," IEEE transactions on neural networks, vol. 20, no. 1, pp. 61-80, 2008.

[40] C. Gallicchio and A. Micheli, "Graph echo state networks," in The 2010 International Joint Conference on Neural Networks (IJCNN). IEEE, 2010, pp. 1-8.

[41] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, "Gated graph sequence neural networks," arXiv preprint arXiv:1511.05493, 2015.

[42] H. Dai, Z. Kozareva, B. Dai, A. Smola, and L. Song, "Learning steadystates of iterative algorithms over graphs," in International conference on machine learning. PMLR, 2018, pp. 1106-1114.

[43] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," arXiv preprint arXiv:1312.6203, 2013.

[44] M. Henaff, J. Bruna, and Y. LeCun, "Deep convolutional networks on graph-structured data," arXiv preprint arXiv:1506.05163, 2015.

[45] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," arXiv preprint arX-

 DOI: https://doi.org/10.30564/aia.v3i1.3178

iv:1606.09375, 2016.

[46] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," arXiv preprint arXiv:1609.02907, 2016.

[47] R. Levie, F. Monti, X. Bresson, and M. M. Bronstein, "Cayleynets: Graph convolutional neural networks with complex rational spectral filters," IEEE Transactions on Signal Processing, vol. 67, no. 1, pp. 97-109, 2018.

[48] A. F. Agarap, "Deep learning using rectified linear units (relu)," arXiv preprint arXiv:1803.08375, 2018.

[49] A. Micheli, "Neural network for graphs: A contextual constructive approach," IEEE Transactions on Neural Networks, vol. 20, no. 3, pp. 498-511, 2009.

[50] J. Atwood and D. Towsley, "Diffusion-convolutional neural networks," arXiv preprint arXiv:1511.02136, 2015.

[51] M. Niepert, M. Ahmed, and K. Kutzkov, "Learning convolutional neural networks for graphs," in International conference on machine learning. PMLR, 2016, pp. 2014-2023.

[52] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in International Conference on Machine Learning. PMLR, 2017, pp. 1263-1272.

[53] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," arXiv preprint arXiv:1707.01926, 2017.

[54] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, "Structural-rnn: Deep learning on spatio-temporal graphs," in Proceedings of the ieee conference on computer vision and pattern recognition, 2016, pp. 5308-5317.

[55] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in Proceedings of the AAAI conference on artificial intelligence, vol. 32, no. 1, 2018.

DOI: https://doi.org/10.30564/aia.v3i1.3178