Symbolic Vs Sub-symbolic AI Methods: Friends or Enemies?

Eleni Ilkou[a,b], Maria Koutraki[a,b]

[a]L3S Research Center, Appelstrasse 9a, 30167 Hannover, Germany
[b]Leibniz University of Hannover, Welfengarten 1, 30167 Hannover, Germany

Abstract
There is a long and unresolved debate between the symbolic and sub-symbolic methods. However, in recent years, there is a push towards in-between methods. In this work, we provide a comprehensive overview of the symbolic, sub-symbolic and in-between approaches focused in the domain of knowledge graphs, namely, *schema representation*, *schema matching*, *knowledge graph completion*, *link prediction*, *entity resolution*, *entity classification* and *triple classification*. We critically present key characteristics, advantages and disadvantages of the main algorithms in each domain, and review the use of these methods in knowledge graph related applications.

Keywords
Symbolic methods, sub-symbolic methods, in-between methods, knowledge graph tasks, knowledge graph completion, schema

1. Introduction

Symbolic and sub-symbolic represent the two main branches of Artificial Intelligence (AI). The AI field saw huge progress and established itself in the 1950s, after some of the most notable and inaugural works of McCulloch and Pittes, who in 1943 established the foundations of neural networks (NN), and Turing's work, who introduced in 1950s the test of intelligence for machines, known as the *Turing test*.

Since its invention, the field has seen ups and downs in its development, which are colloquially known as the AI seasons, and are characterised as "summers" and "winters". The exact periods of these ups and downs are unclear, however, we adopt an intermediate convention based on Wikipedia and Henry Kautz's talk[1] "The Third AI Summer" in AAAI 2020. We display a timeline of these developments in Figure 1.

The first AI summer, also called the golden years, begins a few years after the birth of AI, and it was based on the optimism in *problem solving* and *reasoning*. The dominant paradigm was *symbolic AI* until the 1980s. This is when the *sub-symbolic AI* starts taking the lead and gains attention until the recent years. There is a long and unresolved debate between the two different approaches. However, this grapple between the different AI domains is approaching to its end, as we are currently experiencing the third AI summer, where the presiding wave is the combination of *symbolic* and *sub-symbolic* AI approaches, which we refer to as *in-between methods*.

Table 1 shows an overview of some of the basic different characteristics of the symbolic and sub-symbolic AI methods. It presents an easy visual comparison between the two AI fields; as it was discussed in [1, 2] and according to our thorough analysis of the fields. Apart from the core symbolic or sub-symbolic methods, nowadays there are symbolic applications with sub-symbolic characteristics and vice versa [3]. We choose to adopt an annotation where a method belongs to symbolic or sub-symbolic if it uses only symbolic or sub-symbolic parts respectively; otherwise we categorise it in the in-between methods. The main differences between these two AI fields are the following: (1) symbolic approaches produce logical conclusions, whereas sub-symbolic approaches provide associative results. (2) The human intervention is common in the symbolic methods, while the sub-symbolic learn and adapt to the given data. (3) The symbolic methods perform best when dealing with relatively small and precise data, while the sub-symbolic ones are able to handle large and noisy datasets.

In this paper, we discuss in detail some of the well-known approaches in each AI domain, and their application use-cases in some of the most prominent downstream tasks in the domain of knowledge graphs. We focus on their applicability in the schema representation, schema matching, knowledge graph completion and more specifically in entity resolution, link prediction, entity and triple classification. In this work, we make the following contributions:

- A overview of the characteristics, advantages and disadvantages of the symbolic and sub-symbolic AI methods (Sections 2 and 3).

Proceedings of the CIKM 2020 Workshops, October 19-20, Galway, Ireland.
ilkou@l3s.de (E. Ilkou); koutraki@l3s.de (M. Koutraki)
0000-0002-4847-6177 (E. Ilkou)

[1]https://roc-hci.com/announcements/the-third-ai-summer/

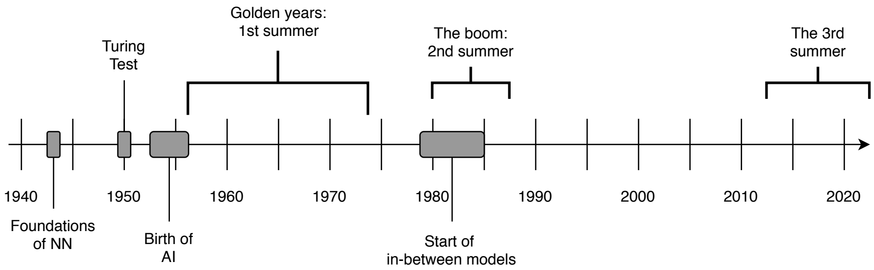

Figure 1: The timeline of Artificial Intelligence methods

- An analysis of the in-between methods and their different categories as they are presented in the literature, and their general characteristics (Section 4).
- An overview of the most common applications of the symbolic, sub-symbolic and in-between methods in knowledge graphs (Section 5).

The rest of this paper is structured as following: Section 2 presents an overview of the main characteristics of the symbolic AI methods. Similarly, in Section 3 we discuss the main characteristics of the sub-symbolic methods. In Section 4, we present an overview of the approaches that combine both symbolic and sub-symbolic methods, namely, the in-between methods. Then, in Section 5 we present some of the most important downstream tasks in the field of knowledge graphs and we analyse the different approaches (symbolic, sub-symbolic and in-between methods) that have been followed in the literature to tackle these tasks.

2. Symbolic Methods

Symbolic methods, also known as Good Old Fashioned Artificial Intelligence (GOFAI), refer to human-readable and explainable processes. The symbolic techniques are defined by explicit symbolic methods, such as formal methods and programming languages, and are usually used for deductive knowledge [4]. They consist of first-order logic rules, while other methods include rules, ontologies, decision trees, planning and reasoning. According to Benderskaya et al [5] the symbolic AI is usually associated with knowledge bases and expert systems, and it is a continuation of the von Neumann and Turing machines.

A key characteristic of symbolic methods is their ability to *explain* and *reason* about the reached conclusion. Furthermore, even their intermediate steps are often explainable. The symbolic systems provide a human-understandable computation flow which makes them easier to debug, explain and control. In particular, the rule based systems have the advantage of *rule modularity*, as the rules are discrete and autonomous knowledge units that can easily be inserted or removed from a knowledge base [6]. Moreover, they provide knowledge *interoperability*; meaning that in closely related applications, knowledge transfer is possible. Also, they are better for abstract problems as they are not highly dependent on the input data.

On the other hand, the symbolic methods are typically not well-suitable for cases where datasets have data-quality issues and might be prone to noise. Under such circumstances, they are often yielding to sub-optimal results [5], and they are not possible to conclude ("brittleness") [7]. Further, the rules and the knowledge usually are hard and hand-coded, creating the Knowledge Acquisition Bottleneck [8], which refers to the high cost of human involvement in converting real-world problems into inputs for symbolic AI systems. Finally, the maintenance of rule bases is difficult as it requires complex verification and validation.

In terms of applications, the symbolic methods work best on well-defined and static problems, and on manipulating and modelling abstractions. However, traditionally, they do not have good performance in real-time dynamic assessments and massive empirical data streams.

3. Sub-symbolic Methods

Contrary to symbolic methods, where the learning happens through the human supervision and intervention, sub-symbolic methods establish correlations between input and output variables. Such relations have high complexity, and are often formalized by functions that map the input to the output data or the target variables.

Sub-symbolic methods represent the Connectionism movement that is trying to mimic a human brain and its complex network of interconnected neurons with the Artificial Neural Networks (ANN). The sub-symbolic AI includes statistical learning methods, such as Bayesian learning, deep learning, backpropagation, and genetic algorithms.

Table 1
Symbolic vs Sub-symbolic methods characteristics

Symbolic	Sub-symbolic
Symbols	Numbers
Logical	Associative
Serial	Parallel
Reasoning	Learning
von Neumann machines	Dynamic Systems
Localised	Distributed
Rigid and static	Flexible and adaptive
Concept composition and expansion	Concept creation, and generalization
Model abstraction	Fitting to data
Human intervention	Learning from data
Small data	Big data
Literal/precise input	Noisy/incomplete input

The sub-symbolic methods are more robust against noisy and missing data, and generally have high computing performance. They are easier to scale up, therefore, they are well suitable for big datasets and large knowledge graphs. Moreover, they are better for perceptual problems, and they require less knowledge upfront.

However, connectionist methods have some disadvantages. The most important one is the lack of interpretability in these methods. This presents a big obstacle to their applicability in domains where explanations and interpretations are key points. Further, based on the General Data Protection Regulation of European Union [9], sub-symbolic techniques are proving to be usually restricted in critical or high-risk decision applications such as the medical, legal or military decision applications and the autonomous cars. Furthermore, they are highly dependant on the training data they process. At first glance, it might not seem like a problem, however, this results in an inability to extrapolate results to unseen instances or data which do not follow a similar distribution as the training data. Additionally, due to the typically large amount of parameters that need to be estimated in sub-symbolic models, they require huge computation power and huge amounts of data. Another issue arising is the availability of high quality data for training the algorithms, which often are difficult to find. Data need to be correctly labelled and to have decent representatives of the normal not to lead to biased outcomes [10].

Most common applications of sub-symbolic methods include prediction, clustering, pattern classification and recognition of objects, and Natural Language Processing (NLP) tasks. Further, we find in sub-symbolic applications the text classification and categorization, as well as recognition of speech and text.

4. In-between Methods

Despite the fundamental differences between symbolic and sub-symbolic the last years there is a link between them with the in-between methods. Since late 1980s, there is a discussion about the need of cognitive sub-symbolic level [11]. The in-between methods consist of the efforts to bridge the gap between the symbolic and sub-symbolic paradigms. The idea is to create a system which can combine the advantages of both methods: the ability to learn from the environment and the ability to reason the results.

Most of the recent applications use a combination of symbolic and connectionist parts to create their algorithms. The used terminology for the range between the symbolic and sub-symbolic varies, as can be seen in this Section many methods are found with different names. Therefore, we refer to them as *in-between methods.*

4.1. General characteristics

The advantages of the in-between computations are evident and measurable to specific applications, with higher accuracy, efficiency and knowledge readability [12]. They have an explanation capacity with no need for a-priori assumptions, and they are comprehensive cognitive models which integrate statistical learning with logical reasoning. They also perform well with noisy data [13]. Another advantage is that these systems during learning can combine logical rules with data, while fine-tuning the knowledge based on the input data. Overall, they seem suitable for applications which have large amounts of heterogeneous data and need knowledge descriptions [14].

In the in-between algorithms we find the Knowledge-based Neural Networks (KBNN or KBANN) [16], Hybrid Expert System (HES) [17], Connectionist Inductive Learning and Logic Programming (CILP) and Connectionist Temporal Logics of Knowledge (CTLK) [14], Graph Neural Networks (GNN) [18], Tensor Product Representation [19], in which the core is a neural network that is loosely-coupled with a symbolic problem-solver. Also, we find the Logic Tensor Networks [20], Neural Tensor Networks [21] for representing complex logical structures, and the latter's extension are the knowledge graph translating embedding models [22].

The applications of these methods can be found in many domains which combine learning and reasoning parts according to a specific problem. However, the existing hybrid models are non-generalizable, they cannot be applied in multiple domains; each time the model is developed to answer a specific question. Also, there is no guide deciding the combinations of symbolic and sub-symbolic parts for computation and representation [2]. Recent downstream applications tend to combine symbolic and sub-symbolic methods for their computation

<table>
<tr><td rowspan="3">CONNECTIONISM</td><td colspan="4">NEUROSYMBOLIC INTEGRATION</td><td rowspan="3">SYMBOLICISM</td></tr>
<tr><td colspan="2">Unified approaches</td><td colspan="2">Hybrid approaches</td></tr>
<tr><td>Neuronal Symbol Proc.</td><td>Connectionist Symbol Proc.</td><td>Functional hybrids</td><td>Translational hybrids</td></tr>
<tr><td>Segregation</td><td>Neuronal eliminativism</td><td>Connectionist eliminativism Limitivism Revisionism</td><td colspan="2">Hybridization or cohabitation</td><td>Segregation Implementation–alism</td></tr>
</table>

Figure 2: The range from symbolic to sub-symbolic as proposed by Hilario [15]

model, more often than using a strictly only one of the two as can be seen in Section 5.

4.2. Existing Categorisations in Literature

Some of the in-between methods are found in literature as connectionist expert systems (or neural network based expert systems) [23], multi-agent systems [5], hybrid representations [24], neural-fuzzy [25, 26] and neural-symbolic (or neurosymbolic [15]) computing, learning and reasoning[2], and its sub-type neurules [27].

To the best of our knowledge, there is no report containing all the in-between methods. Moreover, there is no standard categorisation or common taxonomy for the methods which belong in the range between the symbolic and sub-symbolic techniques. The used terminology varies, therefore, we refer to them as *in-between methods*, and not as neural-symbolic, hybrid or unified. In the last years, there is an increased interest about in-between methods [28], and there are some review works in the domain each one presenting a taxonomy. Most of them refer to the in-between methods as neural-symbolic approaches.

Garcez et al [14] present a neural-symbolic computing table that separates the methods to applications of knowledge representation, learning, reasoning and explainability. Bader and Hitzler [13] study the dimensions of neural-symbolic integration and propose the dimensions of usage, language and interrelation. In the neural-symbolic techniques, they identify two models, the hybrid and integrated (also called unified or translational) [29]. The difference between the two is that hybrid models combine two or more symbolic and sub-symbolic techniques which run in parallel, while the integrated neural symbolic systems consist of a main sub-symbolic component which uses symbolic knowledge in the processing.

Hilario [15] separates neurosymbolic integration to unified and hybrid approaches, each consisted of two subcategories. Both categories, unified and hybrid, have a similar description to Bader and Hitzler [13]. In the unified approaches, Hilario identifies two main categories the neuronal and the connectionist symbol processing, and on the hybrid approaches the translational and functional hybrids respectively. The translational models translate representations between NNs and symbolic structures. Furthermore, she creates a visual continuous representation from connectionism to symbolism, in which includes the categories she is proposing in the range between sub-symbolic and symbolic techniques, as it is illustrated in Figure 2.

In continuation of Hilario's model, McGarry et al [30] focuses on hybrid rule-based systems. They are proposing the categorisation of the symbolic rules and neural networks integrations into unified, transformational and adds the modular subcategory. The latter covers the hybrid models that they consist of several ANNs and rule-based modules, which are coupled and integrated with many degrees. They support that most of hybrid models are modular.

[2] http://www.neural-symbolic.org/

5. Knowledge Graph Tasks

There are plenty of symbolic, sub-symbolic and in-between applications in different domains. Our main focus in this study will be knowledge graph related applications.

A knowledge graph (KG) consists of a set of triples $K \subseteq E \times R \times (E \cup L)$, where E is a set of resources that we refer as entities, L a set of literals, and R a set of relations. Given a triple (h, r, t) (aka a statement), h is known as subject, r as relation, and t as object. A KG can represent any kind of information for the world such as `(Anna_Karenina, writtenBy, Leo_Tolstoy)` and `(Leo_Tolstoy, bornIn, Russia)`, which means that Anna Karenina is written by Leo Tolstoy, who was born in Russia. The above notation will help us explain and analyse the following tasks.

5.1. Schema Representation

Schemata are present from the beginning of databases and data management systems, and stand in for the structure of the data and knowledge. In the last years, there is attention towards linking and structuring data in the web. Connecting information on the web can also be achieved by schemata; an example is schema.org[3] which focuses on the schema creation, representation and maintenance [31]. When modelling knowledge graphs, schemata can be used to prescribe high-level rules that the graph should follow [4]. Knowledge schema or otherwise schema representation contains a conceptual model of the KG. A schema defines the types of entities and relations which can exist in a KG, and an abstract way of combining these entities and relations in (h,r,t) triples. In our example, a schema representation could exist in the form of triples which will state that `(Book, writtenBy, Author)` and `(Author, bornIn, Country)` [32].

Schema representation is traditionally a symbolic task. First-order logic, ontologies, and formal knowledge representation languages, such as RDF(S), OWL [33], XML [34] as well as rules have been used for schema formulations. Some of the most representative examples of schema representation in terms of knowledge graphs construction are YAGO [35] and DBpedia [36]. The two of the most frequently used KGs are following a symbolic approach as they are mostly rely on rule mining techniques used to extract knowledge and represent it in RDF(S) terms.

5.2. Schema Matching

Different KGs use different schemata to represent the same information which create the need for schema matching. Schema matching or mapping, also can be found as schema alignment, is happening between two or more KGs, when we want to perform data integration or mapping, and it refers to the process of identifying semantically related objects. It is similar to the entity resolution, in Section 5.3.1, with the difference that the latter cares about mapping object references, such as "L. Tolstoy" and "Leo Tolstoy", while the schema matching works on the schema definitions, such as *Author* and *Person*.

Over the last decades, many models and prototypes have been introduced on schema matching. Based on a survey in schema matching [37], each model uses an input schema with most common an OWL data model, then a RDF, and finally a document type definition. They process the symbolic input by using different models, which can be linguistic or language based, constrain-based, and structured-based. The linguistic matchers combine the symbolic input with sub-symbolic NLP algorithms [38]. The constrain-based matchers are exploiting the constrains in data features, such as the data types and ranges. The structured-based matchers focus on the database/graph structure. Both constrain and structured based models use mostly symbolic techniques, while there is also an interesting raise in combinations of matchers (hybrid models). The application domain for schema mapping evolves from specific to generic applications. The majority of these methods focus on class alignment, however, there also are works focus on relation alignment [39, 40, 41].

[3]https://schema.org/

5.3. Knowledge Graph Completion

Once a knowledge graph is created, it contains a lot of noisy and incomplete data [42]. In order to fill the missing information for a constructed knowledge graph, we use the task of Knowledge Graph Completion (KGC). KGC, similar to knowledge graph identification [43], is an intelligent way of performing data cleaning. This is usually addressed with filling the missing edges (link prediction), deduplicating entity nodes (entity resolution) and dealing with missing values.

Mostly in-between methods are used for KGC, with the Knowledge Graph Embeddings (KGEs) to be one of the most powerful and commonly used techniques. KGEs aim to create a low dimensional vector representation of the KG and model relation patterns, hence reduce the complexity of KG related tasks while achieving high accuracy. We further analyse the KGC task into the specific sub-tasks of entity resolution and link prediction.

5.3.1. Entity Resolution

Entity resolution (ER) is also known as record linkage, reference matching or duplication. It is the process of finding duplicated references or records in a dataset. It is related to data integration as it is one of its foundational problems [44]. Based on our example, we will have to perform entity resolution between the entities "L. Tolstoy" and "Leo Tolstoy" which can exist in the triples `(Anna_Karenina, writtenBy, L._Tolstoy)` and `(Leo_Tolstoy, bornIn, Russia)`, and refer to the same person.

Record linkage was introduced by Halbert L. Dunn in 1946. In 1960s, there are statistical sub-symbolic models describing the process of entity resolution, which formulate the mathematical basis for many of the current models [45]. In 1990s, machine learning techniques are applied for this assignment and the techniques used are mostly based on the in-between methods [46]. Commonly, ER techniques rely on attribute similarity between the entities [47]. The algorithms deployed for ER are inspired by information retrieval and relational duplicate elimination [48].

5.3.2. Link Prediction

Link prediction techniques, also known as edge prediction, have been applied in many and different fields. Edge prediction refers to the task of adding new links to an existing graph. In practice, this can be used as a recommendation system for future connections, or as completion-correction tool that foresees the missing links between entities.

The link prediction is a well studied field consisted of many and different approaches. A survey of link prediction in complex networks [49] separates the approaches based on the method they are using. It identifies the edge prediction problem to a few techniques that belong to sub-symbolic, such as AI based ANN, probabilistic and Monte Carlo algorithms. However, most of the solutions it proposes, belong to the in-between range. The current state-of-the-art for link prediction tasks is focused on in-between methods. KGEs play a big role in this task and they can be found in different forms of translational models [22], neural based KGE with logical rules by [50], and hierarchy-aware KGEs [51]. In link prediction tasks, we find the triple classification, entity classification, and the head `(?, writtenBy, Leo_Tolstoy)`, relation `(Anna_Karenina, ?, Leo_Tolstoy)`, and tail `(Anna_Karenina, writtenBy, ?)` prediction respectively. We additionally focus on the analysis of entity classification, and triple classification in the next paragraphs.

Entity Classification. It can also be found as node classification or type prediction in the literature. Entity classification tries to predict the type or the class of an entity given some characteristics. In our case, the input triple would be `(Leo_Tolsto, isA, ?)`, which could give the results `(1, Person, 99%)` and `(2, Author, 98%)`.

While most of link prediction tasks are sub-symbolic based with combination of some symbolic parts, the entity classification is more related to the schema and ontology of the KG, hence, the techniques are symbolic based [52, 53].

Triple Classification. The triple classification is a binary problem which answers whether a triple (h,r,t) is true or not, for example the input `(Anna_Karenina, writtenBy, Leo_Tolstoy)?` leads to result `(yes, 92%)`. Systems like the Trans* KGEs [22] use a density function to make predictions about the triple classification based on a probability function. Mayank Kejriwal [54] claims that the correct metric for this task with the usage of KGEs is accuracy, if the test data are balanced.

Triple classification algorithms usually belong to in-between methods, with some examples using neural tensor networks [21] and time-aware [55], latent factor and semantic matching models.

6. Conclusions

We represented the symbolic, sub-symbolic and in-between methods in AI and analysed the key characteristics, main approaches, advantages and disadvantages of each technique respectively. Further, we argued that the current, and possibly future, area of processes is the application of the in-between methods. We justified this belief by discussing principal downstream tasks related to knowledge graph.

Acknowledgments

This work was partially funded by the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No 860801.

References

[1] L. R. Medsker, Hybrid neural network and expert systems, Springer Science & Business Media, 2012.
[2] T. R. Besold, K.-U. Kühnberger, Towards integrated neural–symbolic systems for human-level ai: Two research programs helping to bridge the gaps, Biologically Inspired Cognitive Architectures 14 (2015) 97–110.
[3] G. Marcus, Deep learning: A critical appraisal, arXiv preprint arXiv:1801.00631 (2018).
[4] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. de Melo, C. Gutierrez, J. E. L. Gayo, S. Kirrane, S. Neumaier, A. Polleres, et al., Knowledge graphs, arXiv preprint arXiv:2003.02320 (2020).
[5] E. N. Benderskaya, S. V. Zhukova, Multidisciplinary trends in modern artificial intelligence: Turing's way, in: Artificial Intelligence, Evolutionary Computing and Metaheuristics, Springer, 2013, pp. 319–343.
[6] I. Hatzilygeroudis, J. Prentzas, Neuro-symbolic approaches for knowledge representation in expert systems, International Journal of Hybrid Intelligent Systems 1 (2004) 111–126.
[7] D. B. Lenat, M. Prakash, M. Shepherd, Cyc: Using common sense knowledge to overcome brittleness and knowledge acquisition bottlenecks, AI magazine 6 (1985) 65–65.
[8] J. Cullen, A. Bryman, The knowledge acquisition bottleneck: time for reassessment?, Expert Systems 5 (1988) 216–225.
[9] E. Salami, An analysis of the general data protection regulation (eu) 2016/679, Available at SSRN 2966210 (2017).
[10] E. Ntoutsi, P. Fafalios, U. Gadiraju, V. Iosifidis, W. Nejdl, M.-E. Vidal, S. Ruggieri, F. Turini, S. Papadopoulos, E. Krasanakis, et al., Bias in data-driven

artificial intelligence systems—an introductory survey, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 10 (2020) e1356.

[11] M. Frixione, G. Spinelli, S. Gaglio, Symbols and subsymbols for representing knowledge: a catalogue raisonne, in: Proceedings of the 11th international joint conference on Artificial intelligence-Volume 1, 1989, pp. 3–7.

[12] A. d. Garcez, T. R. Besold, L. De Raedt, P. Földiak, P. Hitzler, T. Icard, K.-U. Kühnberger, L. C. Lamb, R. Miikkulainen, D. L. Silver, Neural-symbolic learning and reasoning: contributions and challenges, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2015.

[13] S. Bader, P. Hitzler, Dimensions of neural-symbolic integration-a structured survey, arXiv preprint cs/0511042 (2005).

[14] A. d. Garcez, M. Gori, L. C. Lamb, L. Serafini, M. Spranger, S. N. Tran, Neural-symbolic computing: An effective methodology for principled integration of machine learning and reasoning, arXiv preprint arXiv:1905.06088 (2019).

[15] M. Hilario, An overview of strategies for neurosymbolic integration, Connectionist-Symbolic Integration: From Unified to Hybrid Approaches (1997) 13–36.

[16] G. Agre, I. Koprinska, Case-based refinement of knowledge-based neural networks, in: Proceedings of the International Conference" Intelligent Systems: A Semiotic Perspective, volume 2, 1996, pp. 20–23.

[17] S. Sahin, M. R. Tolun, R. Hassanpour, Hybrid expert systems: A survey of current approaches and applications, Expert systems with applications 39 (2012) 4609–4617.

[18] L. Lamb, A. Garcez, M. Gori, M. Prates, P. Avelar, M. Vardi, Graph neural networks meet neural-symbolic computing: A survey and perspective, arXiv preprint arXiv:2003.00330 (2020).

[19] P. Smolensky, Tensor product variable binding and the representation of symbolic structures in connectionist systems, Artificial intelligence 46 (1990) 159–216.

[20] I. Donadello, L. Serafini, A. D. Garcez, Logic tensor networks for semantic image interpretation, arXiv preprint arXiv:1705.08968 (2017).

[21] R. Socher, D. Chen, C. D. Manning, A. Ng, Reasoning with neural tensor networks for knowledge base completion, in: Advances in neural information processing systems, 2013, pp. 926–934.

[22] H. Cai, V. W. Zheng, K. C.-C. Chang, A comprehensive survey of graph embedding: Problems, techniques, and applications, IEEE Transactions on Knowledge and Data Engineering 30 (2018) 1616–1637.

[23] S. I. Gallant, S. I. Gallant, Neural network learning and expert systems, MIT press, 1993.

[24] M. Moreno, D. Civitarese, R. Brandao, R. Cerqueira, Effective integration of symbolic and connectionist approaches through a hybrid representation, arXiv preprint arXiv:1912.08740 (2019).

[25] R. Fullér, Neural fuzzy systems (1995).

[26] L. Magdalena, A first approach to a taxonomy of fuzzy-neural systems, Connectionist symbolic integration (1997).

[27] J. Prentzas, I. Hatzilygeroudis, Neurules-a type of neuro-symbolic rules: An overview, in: Combinations of Intelligent Methods and Applications, Springer, 2011, pp. 145–165.

[28] A. d'Avila Garcez, Proceedings of ijcai international workshop on neural-symbolic learning and reasoning nesy 2005 (2005).

[29] R. Sun, F. Alexandre, Connectionist-symbolic integration: From unified to hybrid approaches, Psychology Press, 2013.

[30] K. McGarry, S. Wermter, J. MacIntyre, Hybrid neural systems: from simple coupling to fully integrated neural networks, Neural Computing Surveys 2 (1999) 62–93.

[31] J. Ronallo, Html5 microdata and schema. org, Code4Lib Journal (2012).

[32] C. Belth, X. Zheng, J. Vreeken, D. Koutra, What is normal, what is strange, and what is missing in a knowledge graph: Unified characterization via inductive summarization, in: Proceedings of The Web Conference 2020, 2020, pp. 1115–1126.

[33] K. Sengupta, P. Hitzler, Web ontology language (owl), Encyclopedia of Social Network Analysis and Mining (2014).

[34] F. Zhang, L. Yan, Z. M. Ma, J. Cheng, Knowledge representation and reasoning of xml with ontology, in: Proceedings of the 2011 ACM symposium on applied computing, 2011, pp. 1705–1710.

[35] F. M. Suchanek, G. Kasneci, G. Weikum, Yago: A large ontology from wikipedia and wordnet, Journal of Web Semantics 6 (2008) 203–217.

[36] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, C. Bizer, Dbpedia - A large-scale, multilingual knowledge base extracted from wikipedia, Semantic Web 6 (2015) 167–195.

[37] E. Sutanta, R. Wardoyo, K. Mustofa, E. Winarko, Survey: Models and prototypes of schema matching., International Journal of Electrical & Computer Engineering (2088-8708) 6 (2016).

[38] O. Unal, H. Afsarmanesh, et al., Using linguistic techniques for schema matching., in: International Conference on Software Technologies ICSOFT (2), 2006, pp. 115–120.

[39] M. Koutraki, N. Preda, D. Vodislav, Online relation

alignment for linked datasets, in: European Semantic Web Conference, Springer, 2017, pp. 152–168.
[40] M. Koutraki, N. Preda, D. Vodislav, SOFYA: semantic on-the-fly relation alignment, in: International Conference on Extending Database Technology (EDBT), 2016.
[41] R. Biswas, M. Koutraki, H. Sack, Exploiting equivalence to infer type subsumption in linked graphs, in: European Semantic Web Conference, Springer, 2018, pp. 72–76.
[42] R. West, E. Gabrilovich, K. Murphy, S. Sun, R. Gupta, D. Lin, Knowledge base completion via search-based question answering, in: Proceedings of the 23rd international conference on World wide web, 2014, pp. 515–526.
[43] J. Pujara, H. Miao, L. Getoor, W. Cohen, Knowledge graph identification, in: International Semantic Web Conference, Springer, 2013, pp. 542–557.
[44] S. Thirumuruganathan, M. Ouzzani, N. Tang, Explaining entity resolution predictions: Where are we and what needs to be done?, in: Proceedings of the Workshop on Human-In-the-Loop Data Analytics, 2019, pp. 1–6.
[45] I. P. Fellegi, A. B. Sunter, A theory for record linkage, Journal of the American Statistical Association 64 (1969) 1183–1210.
[46] T. Ebisu, R. Ichise, Graph pattern entity ranking model for knowledge graph completion, arXiv preprint arXiv:1904.02856 (2019).
[47] V. Christophides, V. Efthymiou, T. Palpanas, G. Papadakis, K. Stefanidis, End-to-end entity resolution for big data: A survey, arXiv preprint arXiv:1905.06397 (2019).
[48] N. Koudas, S. Sarawagi, D. Srivastava, Record linkage: similarity measures and algorithms, in: Proceedings of the 2006 ACM SIGMOD international conference on Management of data, 2006, pp. 802–803.
[49] B. Pandey, P. K. Bhanodia, A. Khamparia, D. K. Pandey, A comprehensive survey of edge prediction in social networks: Techniques, parameters and challenges, Expert Systems with Applications 124 (2019) 164–181.
[50] M. Nayyeri, C. Xu, J. Lehmann, H. S. Yazdi, Logicenn: A neural based knowledge graphs embedding model with logical rules, arXiv preprint arXiv:1908.07141 (2019).
[51] Z. Zhang, J. Cai, Y. Zhang, J. Wang, Learning hierarchy-aware knowledge graph embeddings for link prediction., in: Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence, 2020, pp. 3065–3072.
[52] H. Paulheim, C. Bizer, Type inference on noisy rdf data, in: International semantic web conference, Springer, 2013, pp. 510–525.
[53] J. Sleeman, T. Finin, Type prediction for efficient coreference resolution in heterogeneous semantic graphs, in: 2013 IEEE Seventh International Conference on Semantic Computing, IEEE, 2013, pp. 78–85.
[54] M. Kejriwal, Advanced topic: Knowledge graph completion, in: Domain-Specific Knowledge Graph Construction, Springer, 2019, pp. 59–74.
[55] T. Jiang, T. Liu, T. Ge, L. Sha, S. Li, B. Chang, Z. Sui, Encoding temporal information for time-aware link prediction, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2016, pp. 2350–2354.