

Towards Semantically Enriched Embeddings for Knowledge Graph Completion

Mehwish Alam [a,*], Frank van Harmelen [b] and Maribel Acosta [c]
[a] *Telecom Paris, Institut Polytechnique de Paris, France*
E-mail: mehwish.alam@telecom-paris.fr
[b] *Vrije Universiteit Amsterdam, the Netherlands*
E-mail: frank.van.harmelen@vu.nl
[c] *School of Computation, Information and Technology, Technical University of Munich, Germany*
E-mail: maribel.acosta@rub.de

Abstract. Embedding based Knowledge Graph (KG) Completion has gained much attention over the past few years. Most of the current algorithms consider a KG as a multidirectional labeled graph and lack the ability to capture the semantics underlying the schematic information. In a separate development, a vast amount of information has been captured within the Large Language Models (LLMs) which has revolutionized the field of Artificial Intelligence. KGs could benefit from these LLMs and vice versa. This vision paper discusses the existing algorithms for KG completion based on the variations for generating KG embeddings. It starts with discussing various KG completion algorithms such as transductive and inductive link prediction and entity type prediction algorithms. It then moves on to the algorithms utilizing type information within the KGs, LLMs, and finally to algorithms capturing the semantics represented in different description logic axioms. We conclude the paper with a critical reflection on the current state of work in the community and give recommendations for future directions.

Keywords: Knowledge Graph Embeddings, Semantics in Knowledge Graph Embeddings, Large Language Models

1. Introduction

Knowledge Graphs (KGs) have recently gained attention due to their applicability to diverse fields of research, ranging from knowledge management, representation, and reasoning to learning representations over KGs. KGs represent knowledge in the form of relations between entities, referred to as facts, along with schematic information in the form of ontologies. KGs have been used for various downstream tasks such as web search, recommender systems, and question answering. These tasks can also take advantage of Large Language Models (LLMs) which have recently revolutionalized the landscape of the field of Artificial Intelligence. LLMs are used for various downstream Natural Language Processing (NLP) tasks such as natural language understanding, question answering, reasoning, etc. LLMs include masked language models such as BERT [1], RoBerta [2], etc. and generative language models such as as LLaMa [3], ChatGPT [4], and GPT-4 [5]. LLMs have shown high performance in few-shot or zero-shot learning paradigms via prompting and in-context learning [6].

Despite their remarkable performance in various NLP tasks, LLMs are trained on general-purpose data and have lower performance in domain-specific tasks leading to the release of various domain-specific LLMs such as BioBERT, Galactica, etc. Furthermore, LLMs have shown societal bias leading to discrimination since the data on which the LLMs are trained to contain these kinds of biases. Additionally, LLMs suffer from hallucination problems. Last but not least, LLMs are opaque models that lack interpretability [4]. A potential solution to these problems is to induce the knowledge from KGs to LLMs [7], since KGs explicitly represent factual information in a structured way in the form of triples. KGs are known for their reasoning capabilities and for generating interpretable results.

*Corresponding author. E-mail: mehwish.alam@telecom-paris.fr.

KGs and LLMs are thus complementary and can benefit from the capabilities of each other. This aspect has attracted attention recently where LLMs can be enhanced with external knowledge, KGs can be augmented with LLMs, or both can be brought together to enhance reasoning capabilities [8].

KGs, however, suffer from incompleteness because of manual or automated generation. Manual generation leads to limited knowledge represented by the curator and contains curator bias [9], while automated generation may lead to erroneous or missing information. KG completion in particular includes the tasks of (i) triple classification, i.e., deciding if the triple is true or not, (ii) link prediction (LP) to complete the head, tail, or relation in a triple, and (iii) entity classification, also known as entity type prediction. To perform KG completion, various rule-based as well as embeddings based models have been proposed. These algorithms are computationally expensive and are transductive: they only predict triples involving known entities. This is not readily usable when the inference has to be performed on unseen entities and relations. To this end, inductive KG completion allows for predicting triples that involve unseen entities and relations. These transductive and inductive LP algorithms are mostly based on factual information contained in KGs. Various studies leveraging language models as an external source of knowledge have been proposed for KG completion. These algorithms lag behind in terms of performance w.r.t. KG embedding based methods in terms of ranking-based metrics such as *Hits@k* since KG embedding-based algorithms operate under the Closed World Assumption. This led to the need for human evaluation since LLMs contain more general knowledge and may generate correct answers that are different from what is expected by the ground truth with the highest score. Apart from the factual information (i.e., Assertion Box, ABox), another source of information is the ontological information (i.e., Terminology Box, TBox) contained in the KG. This TBox information is almost completely ignored by the current methods. For rectifying this situation various attempts have been made to include type hierarchies and ontological information with different expressivity levels such as $\mathcal{EL}^{++}$, $\mathcal{ALC}$, etc. In some cases, additional representational capabilities are utilized to capture this information such as box embeddings.

Contributions. This survey and vision paper provides an overview of the evolution of methodologies proposed for KG completion, starting from the embedding-based algorithms and LLM-based approaches to the various categories of algorithms proposed for incorporating schematic information within KG embeddings for performing different kinds of completion tasks. It further discusses each of these categories and concludes with critical reflections and future research directions.

Related Work. Several studies have surveyed the state-of-the-art (SoTA) in KG completion. The work by Paulheim [10] provides a survey of the articles related to KG refinement including various classical and rule-based approaches for KG completion. Wang et al. [11] organizes the algorithms for embedding-based KG completion according to their scoring functions such as translational models, semantic matching models, etc. However, this survey does not discuss the methods proposed for KG completion using multimodal information related to an entity or relation such as images, text, and numeric literals. These aspects are targeted in the survey by Gesese et al. [12], which categorizes these methods based on scoring function (inspired by [11]) as well as based on multiple modalities. The survey shows theoretical as well as experimental comparisons of the existing approaches. As compared to these existing studies, the current article targets the semantic aspects of these methods by summarizing and discussing the approaches that have been proposed so far for leveraging semantics provided in the KG.

2. Preliminaries: Semantics in Knowledge Graphs

Commonly, KGs are defined as a set of head-relation-tail triples (h, r, t), where h and t are nodes in a graph, and r is a directed edge from h to t (e.g., [13–16]). In this way, a KG corresponds to a directed, labeled graph, where triples in the KG are labeled edges between labeled nodes, as captured in the following definition.

Definition 1 (Knowledge Graph: Triple Set Definition)**.** *A knowledge graph is a directed, labeled graph $G = (V, E, l, L_G)$, with V the set of nodes, E the set of edges, L_G the set of labels, and a mapping $l : V \cup E \rightarrow L_G$. A triple $t = (h, r, t)$ is a labelled edge, i.e., $(h, r, t) = (l(h), l(r), l(t))$ where $r \in E$ and $h, t \in V$.*

Definition 1 captures the graph structure of KGs, yet aspects concerning the meaning of nodes and edges in a KG are not explicitly defined. First, there is no distinction about the kind of nodes that can exist in a KG, i.e., whether

Table 1

Axioms in $\mathcal{EL}^{++}$, $\mathcal{ALC}$, and $\mathcal{SH}$. Notation: $\top$ top, $\perp$ bottom, a, b are instances, C, D are concepts, r, s are relations, *trans* denotes transitivity.

	$\top$	$\perp$	$\{a\}$	$r(a,b)$	$C(a)$	$C \sqcap D$	$C \sqsubseteq D$	$C \sqcup D$	$\neg C$	$\exists$	$\forall$	$r \sqsubseteq s$	$trans(r)$
$\mathcal{EL}^{++}$	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✗	✗	✗
$\mathcal{ALC}$	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗
$\mathcal{SH}$	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

nodes correspond to entities or text descriptions. This distinction exists in data models like the Resource Description Framework (RDF), where nodes can represent entities, labeled with IRIs or blank nodes, or literals that are used for values such as strings or different types of numbers like integers, float, etc. Distinguishing between entities and literals impacts both the structure and connectivity of nodes in the graph as well as the meaning of things in the graph. As we will see in Section 3.1, specialised KG embedding models are required to handle both kinds of nodes and different types of literals in a KG. Second, the semantics (formal meaning) of classes or relations in the KG is not provided in Definition 1. This is typically the role of ontologies in KGs, which specify the definitions for classes (or concepts) and relations (or roles) using labels or symbols coming from logic. Different logical languages introduce different levels of expressivity (cf. Table 1). Expressivity here refers to the allowed complexity of statements that go beyond simple assertions about individuals (e.g., class assertions) and their relations to other individuals or literals, that are comprised in the ABox. More expressive statements included in the TBox comprise class and role definitions. The constructs and axioms defined in a logic language $\mathcal{L}$ can be transformed into labels and triples to be encoded in a KG. Based on this, we provide a definition for KGs that takes into account the semantics of statements.

Definition 2 (Knowledge Graph: Semantic Definition)**.** *Let $\mathcal{L}$ be a logic language that defines the semantics of concepts and roles, and $G = (V, E, l, L_G)$ a knowledge graph (KG) as of Definition 1. G is an $\mathcal{L}$-KG if the labels L_G contain all the symbols defined in $\mathcal{L}$, and the triples in G correspond to statements that can be expressed in $\mathcal{L}$.*

Typical logic languages used in KGs are in the family of Description Logics, due to their convenient trade-off between expressivity and scalability [17]. For example, the existential language $\mathcal{EL}$ defines the notions of concept intersection and full existential quantification. The extension $\mathcal{EL}^{++}$ introduces the additional notions of concept intersection and concept subsumption; the latter is necessary to model class hierarchies in KGs. $\mathcal{ALC}$ provides additional expressivity w.r.t. $\mathcal{EL}^{++}$ as it includes concept union, negation, and universal quantification. Higher levels of expressivity are captured in KGs modeled with formalisms defined for the semantic web like the Web Ontology Language (OWL). OWL is based on the $\mathcal{SH}$ language, which includes more complex definitions for roles, including role subsumption and transitivity. Another important aspect of KG semantics is the concept of data types captured in the $(\mathcal{D})$ extension. $(\mathcal{D})$ allows for modeling the meaning of literals in KGs which can be part of statements in the ABox. It is important to note that KGs based on OWL can achieve different levels of expressivity beyond the ones described in this section, e.g., OWL-Lite is based on $\mathcal{SHIF}^{(\mathcal{D})}$, OWL-DL on $\mathcal{SHOIN}^{(\mathcal{D})}$, and OWL2 on $\mathcal{SROIQ}^{(\mathcal{D})}$, the latter providing definitions for role reflexivity, irreflexivity, and disjointness, inverse properties, enumerated classes, and qualified cardinality restrictions.

3. Knowledge Graph Embedding Algorithms

The vast majority of embedding based algorithms for KG completion treat KGs as graph structures as presented in Definition 1. Most of these algorithms perform transductive LP [18] where the inference is performed on the same graph on which the model is trained. On the contrary, inductive LP is performed over the unseen graph, i.e., (parts of) the graph are not seen during training. This section gives an overview of both transductive LP (§ 3.1), entity type prediction (§ 3.2), and inductive LP algorithms (§ 3.3).

3.1. Transductive Link Prediction

Translation based models include TransE, TransH, etc. In TransE [13], the relation in a triple is considered a translation operation between the head and tail entities on a low dimensional space. TransH [19] extends TransE by projecting the entity vectors to relation-specific hyperplanes which helps in capturing different roles of an entity w.r.t. different relations. Both models have obvious limitations, such as the inability to represent symmetric or transitive relations. The scoring function of RotatE [20] models the relation as a rotation in a complex plane to preserve the symmetric/anti-symmetric, inverse, and composition relations in a KG.

Semantic matching models are based on a similarity-based scoring function that measures the plausibility of a triple by matching the semantics of the latent representations of entities and relations. In DistMult [21], each entity is mapped to a $d-$dimensional dense vector, and each relation is mapped to a diagonal matrix. The score of a triple is computed as the matrix multiplication between the entity vectors and the relation matrix. RESCAL [14] models the triple into a three-way tensor. The model explains triples via pairwise interaction of latent features. The score of the triple is calculated using the weighted sum of all the pairwise interactions between the latent features of the head and the tail entities. ComplEx [22] extends DistMult by introducing a Hermitian dot product for better handling asymmetric relations.

Neural network models represent an entity using the average of the word embeddings in the entity name. ConvE [15] uses 2D convolutional layers to learn the embeddings of the entities and relations in which the head entity and the relation embeddings are reshaped and concatenated which serves as an input to the convolutional layer. The resulting feature map tensor is then vectorized and projected into a k-dimensional space and matched with the tail embeddings using the logistic sigmoid function minimizing the cross-entropy loss. In ConvKB [23], each triple is represented as a 3-column matrix which is then fed to a convolution layer. Multiple filters are applied to the matrix in the convolutional layer to generate different feature maps. Next, these feature maps are concatenated into a single feature vector representing the input triple. The feature vector is multiplied with a weight vector via a dot product to return a score which is used to predict whether the triple is valid. Relational Graph Convolutional Networks (R-GCN) [24] extends Graph Convolutional Networks (GCN) to distinguish different relationships between entities in a KG. In R-GCN, different edge types use different projections, and only edges of the same relation type are associated with the same projection weight.

Path-based models such as PTransE [25] extend TransE by introducing a path-based translation model. GAKE [26] considers the contextual information in the graph by considering the path information starting from an entity. RDF2Vec [27] uses random walks to consider the graph structure and then applies the word embedding model on the paths to learn the embeddings of the entities and the relations. However, the prediction of head or tail entities with RDF2Vec is non-trivial because it is based on a language modeling approach.

Multimodal KG embeddings make use of different kinds of literals such as numeric, text, or images (see [12] for detailed analysis). This group of algorithms considers aspects of the $(\mathcal{D})$ extension of logic languages (cf. Section 2). For example, DKRL [28] extends TransE by incorporating the textual entity descriptions encoded using a continuous bag-of-words approach. Jointly (ALSTM) [29] extends the DKRL model with a gating strategy and uses attentive LSTM to encode the textual entity descriptions. MADLINK [30] uses SBERT for representing entity descriptions and the structured representation is learned by performing random walks where at each step the relations to be crawled are ranked using "predicate frequency - inverse triple frequency" *(pf-itf)*.

3.2. Entity Type Prediction

SDType [31] is a statistical heuristic model that exploits links between instances using weighted voting and assumes that certain relations occur only with particular types. It does not perform well if two or more classes share the same sets of properties and also if specific relations are missing for the entities. Many machine learning including neural network based models have been proposed for type prediction. Cat2Type [32] takes into account the semantics of the textual information in Wikipedia categories using language models such as BERT. In order to consider the structural information of Wikipedia categories, a category-to-category network is generated which is then fed to Node2Vec for obtaining the category embeddings. The embeddings of both structural and textual information are combined to classify entities into their types. The approach by Biswas et al. [33] leverages different word embedding models, trained on triples, together with a classification model to predict the entity types. Therefore, contextual information is not captured. Scalable Local Classifier per Node (SLCN) is used by the model in [34] for type prediction based on a set of incoming and outgoing relations. However, entities with only a few relations are likely to be misclassified. FIGMENT [35], uses a global model and a context model. The global model predicts entity types based on the entity mentions from the corpus and the entity names. The context model calculates a score for each context of an entity and assigns it to a type. Therefore, FIGMENT requires a large annotated corpus which is a drawback of the method. In APE [36], a partially labeled attribute entity-to-entity network is constructed containing structural, attribute, and type information for entities, followed by deep neural networks to learn the entity embeddings. MRGCN [37] is a multi-modal message-passing network that learns end-to-end from the structure of KGs as well as from multimodal node features. In HMGCN [38], the authors propose a GCN-based model to predict the entity types considering the relations, textual entity descriptions, and Wikipedia categories. ConnectE [39] and AttET [40] models find a correlation between neighborhood entities to predict the missing types. Ridle [41] learns entity embeddings and latent distribution of the relations using Restricted Boltzmann Machines allowing to capture semantically related entities based on their relations. This model is tailored to KGs where entities from different classes are described with different relations. CUTE [42] performs hierarchical classification for cross-lingual entity typing by exploiting category, property, and property-value pairs. MuLR [43] learns multi-level representations of entities via character, word, and entity embeddings followed by the hierarchical multi-label classification.

3.3. Inductive Link Prediction

Simply adapting most of the existing transductive LP models for inductive settings requires expensive re-training for learning embeddings for unseen entities leading to their inapplicability to perform predictions with unseen entities. To overcome this, inductive LP approaches were introduced which are discussed in the following.

Statistical rule-mining approaches make use of logical formulas to learn patterns from KGs. Systems such as AnyBURL [44, 45] generalise random walks over the graph into Horn Clause rules which are then used for link-prediction: if the body of the rule matches with a path in the graph, the rule predicts that the triple in the conclusion of the rule should also be in the graph. NeuralLP [46] was proposed which learns first-order logical rules in an end-to-end differentiable model. DRUM [47] is another method that applies a differentiable approach for mining first-order logical rules from KGs and provides an improvement over NeuralLP.

Embedding based methods have also been proposed to work in an inductive setting. GraphSAGE [48] performs inductive LP by training entity encoders through feed-forward and graph neural networks. However, in GraphSAGE, the set of attributes (e.g., bag-of-words) are fixed before learning entity representations, restricting their application to downstream tasks [49]. One way to learn entity embeddings is to use graph neural networks for aggregating the neighborhood information [50, 51]. However, these methods require unseen entities to be surrounded by known entities and fail to handle entirely new graphs [52] (i.e. they work only in a semi-inductive setting). KEPLER [53] is a unified model for knowledge embedding and pre-trained language representation by encoding textual entity descriptions with a pre-trained language model (LM) as their embeddings, and then jointly optimizing the KG embeddings and LM objectives. However, KEPLER is computationally expensive due to the additional LM objective and requires more training data. Inspired by DKRL, BLP [54] uses a pre-trained LM for learning representations of entities via an LP objective. QBLP [55] is a model proposed to extend BLP for hyper-relational KGs by exploiting the semantics present in qualifiers. Previously discussed models only consider unseen entities and not unseen relations. RAILD [56] on the other hand generates a relation-to-relation network for efficiently learning the relation features. Therefore it performs inductive LP for unseen relations.

Discussion. The algorithms discussed so far consider only statements in the ABox for generating KG embeddings and performing LP. Some of the algorithms use contextual information of the entity with the help of graph walks such as GAKE and MADLINK. A vast amount of knowledge is captured by LLMs, type hierarchy and the expressivity of the description logic axioms has not been considered. The subsequent sections focus on these aspects of LP.

4. Towards Capturing Semantics in Knowledge Graph Embeddings

4.1. Large Language Models for Knowledge Graph Completion

Large language models (LLMs) can further be categorized into encoder-decoder models, encoder-only, and decoder-only models. The encoder-decoder models and encoder-only models, such as BERT [1], RoBerta [2], etc., are masked language models which are discriminative models and are pretrained for predicting a masked word. These models have achieved SoTA performance on the NLP tasks such as entity recognition, sentiment analysis, etc. On the other hand, decoder-only models, such as LLaMa [3], ChatGPT [4], and GPT-4 [5], are autoregressive models which are generative and are pretrained on the task of predicting the next word. The rest of the subsection focuses on how these two categories of models have been utilized in the context of KG completion.

One of the pioneers of LLM-based approaches for KG completion is KG Bidirectional Encoder Representations from Transformer (KG-BERT) [57] which is fine-tuned on the task of KG completion and represents the triple as textual sequences. It takes the entity-relation-entity sequence and computes the scoring function using KG-BERT. The model represents entities and relations by their names or descriptions and takes the name/description word sequences as the input sentence of the BERT model for fine-tuning. Despite being an LLM-based approach, KG-Bert does not outperform models considering the structural information of a KG in terms of the ranking-based metrics, i.e., *hits@k*. Kim et al. [58] associates the shortcomings of KG-BERT with the fact that KG-BERT does not handle relations properly and it has difficulties in choosing the correct answer in the presence of lexically similar candidates. Leading to a multitask learning based model [58], that combines the task of relation prediction and relevance ranking with LP for leading to better learning of the relational information.

The previously described methods still suffer from high overheads because of costly scoring functions and a lack of structured knowledge of the textual encoders. A Structured Augmented text Representation (StAR) model [59] was proposed where each triple is partitioned into two asymmetric parts, similar to a translation-based graph embedding approach. Both parts were encoded into contextualized representations with the help of a Siamese-style textual encoder. These existing methods still lag in performance w.r.t. structure-based algorithms. PLM-based KGC (PKGC) [60] highlights that the reason underlying this lag is the evaluation setting, which is currently based on a Closed World Assumption. The performance of a link-prediction algorithm is measured by its capability to predict a set of links that were removed from the knowledge graph), while the LLMs introduce external knowledge. This may lead to the prediction of new links which are semantically correct, but which were not in the original KG, hence not in the removed evaluation set, and hence not counted in the success metric. Additionally, the PLMs are utilized in an inappropriate manner, i.e., when triples are used as sentences it leads to incoherence in the generated sentences. PKGC targets the first problem by proposing manual annotation as an alternative. However, a medium-sized dataset containing 10,000 entities and 10,000 triples in the test set will lead to true labels of at most 200 million triples precluding human annotation. This observation leads to a new evaluation metric $CR@1$ where the triples are sampled from the test set and the missing entities are filled with the top-1 predicted entity. Manual annotation is then performed to measure the correct ratio of these triples. The second problem is addressed by converting each triple and its support information (i.e., the definition and the attribute information) into natural prompt sentences. PKGC outperforms structural and LLM-based methods in various modalities, i.e., with the attribute and definition.

GenKGC [61] converts the KG completion task to a sequence-to-sequence (Seq2Seq) generation task. The in-context learning paradigm of GPT-3 learns correct output answers by concatenating the selected samples relevant to the input. GenKGC similarly proposes a relation-guided demonstration by adding triples of the same relation. It introduces an entity-aware hierarchical decoder during generation for better representation learning and reduced time complexity. On traditional datasets such as WN18RR and FB15k-237, GenKGC underperforms as compared to structural SOTA models on *hits@k* (confirming the hypothesis made By Lv et al. [60]) and outperforms masked language model based approaches demonstrating the capabilities of generative models for KG completion. The Seq2Seq Generative Framework KG-Completion algorithm (KG-S2S) [62] takes into account the aspect of emerging KGs. Given a KG query, KG-S2S directly generates the target entity text by fine-tuning the pre-trained language model. KG-S2S learns the input representations of the entities and relations using entity descriptions, soft prompts, and Seq2Seq dropout. The KG elements, i.e., entity, relations, and timestamp are considered flat text, enabling the model to generate novel entities for KGs. The method is further evaluated in static, few-shot, and temporal settings.

4.2. Capturing Type Information for Knowledge Graph Completion

Recent initiatives have been taken for leveraging the schematic information in the form of type information about entities, both with and without considering the type hierarchies. TKRL [63] considers the hierarchical information of the entity types by using hierarchical type encoders. It is based on the assumption that each entity should have multiple representations for its different (super)types. TransT [64] also proposes an approach that considers the entity type and its hierarchical structure. It goes one step further and constructs relation types from entity types and captures the prior distribution of the entity types by computing the type-based semantic similarity of the related entities and relations. Based on the prior distribution, multiple embedding representations of each entity (set of semantic vectors instead of a single vector) are generated in a different context and then the posterior probability of the entity and the relation prediction is estimated. Zhang et al. [65] consider entity types as a constraint on the set of all the entities and let these type constraints induce an isomorphic collection of subsets in the embedding space. The framework introduces additional cost functions to model the fitness between these constraints and the entity and relation embeddings. JOIE [66] employs cross-view and intra-view modeling, such that (i) the cross-view association jointly learns the embeddings of the ontological concepts and the instance entities, and (ii) the intra-view models learn the structured knowledge related to entities as well as the ontological information (the hierarchy-aware encoding) separately. The model is evaluated on the task of triple completion and entity typing. Another proposed method that learns both entity, relation, and entity type embeddings from entity-specific triples and type-specific triples is Automated Entity Type Representation for KG Embedding (AutoETER) for LP tasks [67]. It considers the relation as a translation between the types of the head and the tail entity with the help of the relation-aware projection mechanism. Type-Aware Graph Attention neTworks for reasoning over KGs (TAGAT) [68] is one of the methods which combines the entity type information with the neighborhood information of the types while generating embeddings with the help of Graph ATtention networks (GAT). The relation level attention aims at distinguishing the importance of each associated relation for the entity. The neighborhood information of each type is also considered with the help of type-level attention since each relation may connect different entity groups even if the head entity belongs to the same group. Entity-level attention aims at determining how important each neighboring entity is for an entity under a given relation. TrustE [69] is yet another method that aims at building entity-typed structured embeddings with tuple trustworthiness by taking into account possible noise in entity types which is usually ignored by the current methods. TrustE encodes the entities and entity types into separate spaces with a structural projection matrix. The trustworthiness is ensured by detecting the noisy entity types where the energy function focuses more on the pairs of an entity and its type with high trustworthiness. The model is evaluated by detecting entity-type noise as well as entity-type prediction.

4.3. Semantically Rich Embeddings

Even though the previously discussed approaches use some of the schematic information, such as the types of entities and the type hierarchy, they still ignore much of the concept level knowledge, i.e., TBox information in terms of description logic axioms. Table 2 presents an overview of approaches that do take TBox information into account, and their expressivity in regards to the constructs and axioms of logic languages discussed in Section 2.

A first generation of these systems represented concepts as (high-dimensional) spheres in the embedding space (e.g. [70]). However, while the intersection of concepts is a common operation, the intersection of two n-balls is not an n-ball, leading to challenges when measuring the distance between concepts and inferring equivalence between concepts. A second generation instead represents concepts as high-dimensional boxes, since boxes are closed under intersection. ELEm [71] is one of the first of these approaches and generates low dimensional vector spaces from $\mathcal{EL}^{++}$ by approximating the interpretation function by extending TransE with the semantics of conjunction, existential quantification, and the bottom concept. It is evaluated for LP based on protein-protein interaction. EMEL++ [72] evaluates the algorithm on the task of subsumption reasoning and compares it to the ELEm where these semantics are represented but not properly evaluated. Similar to TransE, EMEL++ interprets relations as the translations operating between the classes. BoxEL [73] and ELBE [74] extend ELEm by representing concepts as axis parallel boxes with two vectors for either the lower and upper corners or the center and offset. In BoxEL [73], the authors show the aforementioned advantage of box embeddings over ball embeddings with the help of an example related to conjunction operator, i.e., the ball embeddings cannot express $Parent \sqcap Male \equiv Father$ as properly as compared to the box embeddings. Moreover, the translations cannot model the relation $isChildOf$ between a $Person$ and a $Parent$ when they have two different volumes. In addition to the box representation, ELBE [74] defines several loss functions for each of the normal forms representing the axioms expressed in $\mathcal{EL}^{++}$ (shown in Table 2) such as conjunction, the bottom concept, etc. Taking a further step, Box^{2}EL [75] learns representations of not only the concepts

Table 2

Comparison of the expressivity of semantically enriched embeddings. Notation: a, b are instances, C, D, E are concepts, r, r_1, r_2, s are relations. The symbol ✓ (resp. (✓)) indicates that it has been demonstrated by construction or empirically that the approach supports (resp. partially) the expression; otherwise, ✗ is indicated.

	Expressions	ELEm	EMEL++	BoxEL	ELBE	Box^{2}EL	CatE	OWL2Vec*	TransOWL
Constructors	$\bot$	✓	✓	✓	✓	✗	✗	✗	✗
	$\top$	✗	✗	✗	✗	✗	✓	✓	✗
	$\{a\}$	(✓)	✗	✓	✗	✓	✗	✗	✓
	$C(a)$	✓	✓	✓	✓	✓	✗	✓	✓
	$r(a, b)$	✓	✓	✓	✓	✓	✗	✓	✓
$\mathcal{EL}^{++}$	$C \sqsubseteq D$	✓	✓	✓	✓	✓	✓	✓	✓
	$C \sqcap D \sqsubseteq E$	(✓)	✓	✓	✓	✓	(✓)	✗	✗
	$\exists r.C \sqsubseteq D$	✓	✓	✓	✓	✓	✓	✓	✓
	$C \sqsubseteq \exists r.D$	✓	✓	✓	✓	✓	✓	✓	✓
	$C \sqcap D \sqsubseteq \bot$	✗	✓	✓	✓	✓	✓	✗	✗
	$\exists r.C \sqsubseteq \bot$	(✓)	(✓)	✓	(✓)	✓	✗	✗	✗
	$C \sqsubseteq \bot$	✗	✓	✓	✓	✓	✗	✗	✗
$\mathcal{ALC}$	$C \sqcup D$	✗	✗	✗	✗	✗	✓	✗	✗
	$\forall r.D$	✗	✗	✗	✗	✗	✓	✗	✗
	$\neg C$	✗	✗	✗	✗	✗	✓	✗	✗
	$C \sqsubseteq \forall r.D$	✗	✗	✗	✗	✗	✗	✓	✓
	$\forall r.C \sqsubseteq D$	✗	✗	✗	✗	✗	✗	✓	✓
$\mathcal{SH}$	$r \sqsubseteq s$	✗	✓	✗	✗	✗	✗	✓	✓
	$r_1 \circ r_2 \sqsubseteq s$	✗	✓	✗	✗	✗	✗	✓	✗

but also the roles as boxes for preserving as much semantics of the ontology as possible. It uses a similar mechanism to BoxEL for the representation of the concepts. The previous methods define roles (binary relations) as translations as in TransE but Box^{2}EL associates every role with a head box and a tail box so that every point in the head box is related to every point in the tail box with the help of bump vectors. Bump vectors model interaction between the concepts and dynamically move the position of the embeddings of related concepts. CatE [76] on the other hand embeds $\mathcal{ALC}$ ontologies with the help of category theoretical semantics, i.e., the semantics of logic languages that formalizes interpretations using categories instead of sets. This is advantageous because categories have a graph-like structure. TransOWL [77] and its extensions allow for the inclusion of OWL axioms into the embedding process by modifying the loss function of TransE so that it gives a higher weight to triples that involve OWL vocabulary such as `owl:inverseOf`, `owl:equivalentClass`, etc. OWL2Vec* [78] uses a word embedding model to create embeddings from entities and words from the generated corpus. This corpus is generated from random walks over the ontology. The method is evaluated on the task of class membership prediction and class subsumption prediction.

Discussion. This section discussed the methods leveraging schematic information contained in the KGs starting from type hierarchy to the semantics in the description logic axioms. The approaches utilizing type hierarchies for LP still lack unified evaluation w.r.t. benchmark datasets leading to unclear performance comparison of the existing algorithms and of the impact of adding such schematic information for training the models. These models in most cases evaluate the approach on the traditional triple and entity-type prediction tasks but completely ignore tasks related to schema completion. The limited expressivity of these models makes them unsuitable for the tasks of deductive reasoning. These shortcomings are covered by the approaches representing the semantics underlying different description logic axioms with the help of box embeddings. Yet, these approaches are limited to certain kinds of axioms and do not cover more expressive statements like role transitivity. This is evidenced in Table 2 where only a few approaches can handle certain aspects of $\mathcal{SH}$, yet the expressivity of OWL ontologies (from $\mathcal{SHIF}^{(\mathcal{D})}$ to $\mathcal{SROIQ}^{(\mathcal{D})}$) is far to be reached.

Table 3
Overview of the tasks and the datasets used for evaluation for each of the categories.

Category	Tasks	Datasets
Transductive Link Prediction	Link Prediction, Triple Classification, Entity Classification	FB15K, FB15K-237, WN18, WN18RR, YAGO3-10
Inductive Link Prediction	Link Prediction, Triple Classification	Wikidata5M, Wikidata68K
LLM-Based Methods	Link Prediction	FB15K237, WN18RR, FB15K-237N, FB15K-237NH, Wiki27K, OpenBG500, Nell-One
Methods using Type-Hierarchy	Link Prediction, Entity Type Prediction	WN18RR, WN18, FB15k, FB15KET, YAGO43KET, FB15K+, FB15K*, NELL-995, JF17K, YAGO26K-906, DB111K-174.
Methods using Description Logic Axioms	Inductive Reasoning (Subsumption, Class Membership), Deductive Reasoning (Unsatisfiability of the named concept, predicting axioms in deductive closure of an ontology)	FoodOn, HeLIS, GO, SNOMED-CT, GALEN, Anatomy, PPI, FOBI, NRO

5. Existing Evaluation Settings

Table 3 shows an overview of the evaluation followed by the algorithms discussed previously. Most of the algorithms are evaluated on the task of LP since they consider only the triple structure in the KG. Exceptions to this are the tasks that are used for evaluating the algorithms incorporating description logic axioms. Most of the algorithms use $hits@k$ where k varies from 1 to 100, Mean Rank (MR) and Mean Reciprocal Rank (MRR). The benchmark datasets vary from task to task, depending on the requirements of the algorithms. Most datasets are based on WordNet, YAGO, and Freebase. ELEm has been evaluated on the Protein-Protein Interaction (PPI) dataset for the LP task. However, the successor algorithms introduced more appropriate datasets such as Gene Ontology (GO) or FoodOn.

In the case of transductive LP, studies have shown that the efficiency of these algorithms greatly depends on the choice of hyperparameters. For example, Ruffinelli et al. [79] show that RESCAL, one of the earliest models, outperforms all the later approaches by training the models with proper hyperparameters. A similar study by Ali et al. [80] performs a large-scale evaluation of 21 KG embedding models and provides the best set of hyper-parameters. This leads to the conclusion that a unified benchmarking is needed which allows for comparing the approaches under the same conditions. This will reveal interesting insights, e.g., some models considered not competitive anymore can still outperform more sophisticated approaches under certain configurations. While the evaluation setting for transductive LP algorithms has been well established, the other three categories still lack a unified evaluation setting.

6. Conclusion and Recommendations

In this section, we take a critical perspective on the SoTA over the past decade, based on the overview given in the preceding sections. We end with recommendations for fruitful future directions of work.

6.1. Critical Reflection

Ignoring semantics. The majority of KG completion algorithms only consider the ABox (definition 1). This effectively reduces a KG to a "data graph": relations between entities, none of which carries any computer-interpretable semantics (Definition 2) except for the recent ones as discussed in Section 4.2 (to a limited extent) and Section 4.3 (to a larger extent).

Limitations of the transductive setting. The majority of the work on KG completion follows a transductive setting, i.e., not allowing for the completion of a KG with new entities or relations. Only a few algorithms set in an inductive setting consider only the prediction of new entities, with almost no algorithms aiming to predict new relations. Limiting algorithms to the transductive setting severely restricts the downstream tasks to which these algorithms can be applied. For example, missing relations between known proteins and drugs can be predicted (protein-drug interaction), but transductive algorithms cannot be used to predict new drugs from the knowledge of known proteins.

Evaluation settings. The community working on KG completion is severely hampered by a lack of standardised evaluation settings. There are multiple concerns in this area:

- *No standard protocols for hyper-parameter sweeping*, leading to incomparable and unreproducible results. This led to the rather embarrassing result that a paper from 2020 [79] threw into doubt the progress of the entire field over a decade since the early work on RESCAL in 2011 [14].
- *Evaluation datasets with serious limitations*. It took years before the widely used FB15K dataset was discovered to suffer from serious data leakage from the training to testing and validation splits because many triples were simply inverses of other triples, allowing trivial inverse-triple prediction algorithms to already gain a high score [81]. Furthermore, the small number of datasets used in the literature (Table 3) carries the risk that methods will optimise for the benchmark, instead of optimise for the task.
- *Dataset size.* Datasets used in evaluations are typically small (e.g. on the order of 10^5 triples for the widely used FB15k-237 and YAGO3-10) in comparison to the KGs that are currently in routine practical use (often on the order of 10^8 triples or beyond).
- *Evaluation metric.* The most widely deployed evaluation metric is *hits@k* on a held-out set of triples. This method favours reconstructing known links (that were already present in the original KG, but held out) over the prediction of genuinely new links that are semantically correct but did not appear in the original KG. A method may produce many genuinely new links which rank higher than a predicted held-out link, and such a high-value method would end up with a low *hits@k* score. Ironically, it is these methods which are likely to be of most value in downstream tasks. The recently proposed metric *sem@k* [82] aims to rectify this to some extent, although it remains unclear at the moment how much of this evaluation can be done without expensive human annotation of gold standards.

Bias against external knowledge. Almost all current KG completion methods aim to predict links in a KG based on the properties of the KG itself, and this even holds true for most of the inductive methods. On the other hand, the new generation of methods that exploit LLMs for KG prediction uses an external knowledge source (i.e., LLMs, see Section 4.1) to predict new links. Not only is this an obviously promising step to take in an inductive setting, but it also holds the promise of taking LP beyond simply extrapolating the structural patterns that are already present in the KG. After all, methods that merely extrapolate the existing patterns in a KG are likely to simply replicate the sources of the incompleteness of the KG and are thus unlikely to actually solve the problem that was motivating the KG completion task in the first place. It is important to observe that LLMs are by no means the only useful source of external knowledge that can be injected into the KG completion process. Other KGs can also provide such background knowledge, leading to an interesting blurring between the tasks of KG linking and KG completion. Recent work on exploiting the temporal evaluation of a KG as the source of information is another example of using information outside the KG for KG completion.

6.2. Recommendations

The above critical reflections lead to a number of recommendations for future directions. First, a number of methodological issues need to be resolved. Evaluation datasets need to grow larger in size and more diverse in nature, and perhaps be more representative of real-world downstream tasks. More methodological hygiene needs to be practiced in both the protocols for evaluations (e.g. hyper-parameter settings), as well as the reporting about these settings. And alternative metrics will need to be developed which better reflect the actual value of the predicted links for downstream tasks. Second, if the work on *knowledge graph* completion is to go beyond simply *data graph* completion, the effort will need to focus on including the semantics of the KG. This can be as simply accounting for known relations between types such as the subsumption hierarchy or known disjointness relations between types to more sophisticated reasoning about the algebraic properties of roles (symmetry, anti-symmetry, transitivity, etc.), or properties such as minimal and maximal cardinality of roles. Finally, we believe that the community will need to leave the purely transductive setting it has focussed on so far and embrace the challenges of an inductive setting. We also believe that this will have to go hand in hand with a willingness to develop KG completion algorithms that take into account knowledge sources beyond the original KG, be that knowledge from LLMs, from external KGs, from temporal evolution, or any other source of information that can help the task of KG completion in a manner that is useful in practical downstream tasks.

References

[1] J. Devlin, M. Chang, K. Lee and K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, in: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019.

[2] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer and V. Stoyanov, RoBERTa: A Robustly Optimized BERT Pretraining Approach, *CoRR* **abs/1907.11692** (2019). http://arxiv.org/abs/1907.11692.

[3] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave and G. Lample, LLaMA: Open and Efficient Foundation Language Models, *CoRR* **abs/2302.13971** (2023). doi:10.48550/arXiv.2302.13971.

[4] Y. Bang, S. Cahyawijaya, N. Lee, W. Dai, D. Su, B. Wilie, H. Lovenia, Z. Ji, T. Yu, W. Chung, Q.V. Do, Y. Xu and P. Fung, A Multitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity, *CoRR* **abs/2302.04023** (2023). doi:10.48550/arXiv.2302.04023.

[5] S. Bubeck, V. Chandrasekaran, R. Eldan, J. Gehrke, E. Horvitz, E. Kamar, P. Lee, Y.T. Lee, Y. Li, S.M. Lundberg, H. Nori, H. Palangi, M.T. Ribeiro and Y. Zhang, Sparks of Artificial General Intelligence: Early experiments with GPT-4, *CoRR* **abs/2303.12712** (2023). doi:10.48550/arXiv.2303.12712.

[6] J. Yang, H. Jin, R. Tang, X. Han, Q. Feng, H. Jiang, B. Yin and X. Hu, Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond, *CoRR* **abs/2304.13712** (2023). doi:10.48550/arXiv.2304.13712.

[7] F.M. Suchanek and G. Varoquaux, On Language Models and Symbolic Representations, in: *The Conversation*, 2022.

[8] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang and X. Wu, Unifying Large Language Models and Knowledge Graphs: A Roadmap, *CoRR* **abs/2306.08302** (2023). doi:10.48550/arXiv.2306.08302.

[9] 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**(3) (2020), e1356.

[10] H. Paulheim, Knowledge graph refinement: A survey of approaches and evaluation methods, *Semantic Web* **8**(3) (2017), 489–508. doi:10.3233/SW-160218.

[11] Q. Wang, Z. Mao, B. Wang and L. Guo, Knowledge Graph Embedding: A Survey of Approaches and Applications, *IEEE Trans. Knowl. Data Eng.* **29**(12) (2017), 2724–2743. doi:10.1109/TKDE.2017.2754499.

[12] G.A. Gesese, R. Biswas, M. Alam and H. Sack, A survey on knowledge graph embeddings with literals: Which model links better literal-ly?, *Semantic Web* **12**(4) (2021), 617–647. doi:10.3233/SW-200404.

[13] A. Bordes, N. Usunier, A. García-Durán, J. Weston and O. Yakhnenko, Translating Embeddings for Modeling Multi-relational Data, in: *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, C.J.C. Burges, L. Bottou, Z. Ghahramani and K.Q. Weinberger, eds, 2013, pp. 2787–2795. https://proceedings.neurips.cc/paper/2013/hash/1cecc7a77928ca8133fa24680a88d2f9-Abstract.html.

[14] M. Nickel, V. Tresp and H. Kriegel, A Three-Way Model for Collective Learning on Multi-Relational Data, in: *Proceedings of the 28th International Conference on Machine Learning, ICML*, L. Getoor and T. Scheffer, eds, Omnipress, 2011, pp. 809–816. https://icml.cc/2011/papers/438_icmlpaper.pdf.

[15] T. Dettmers, P. Minervini, P. Stenetorp and S. Riedel, Convolutional 2D Knowledge Graph Embeddings, in: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18)*, S.A. McIlraith and K.Q. Weinberger, eds, AAAI Press, 2018, pp. 1811–1818. https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17366.

[16] N. Jia, X. Cheng and S. Su, Improving Knowledge Graph Embedding Using Locally and Globally Attentive Relation Paths, in: *Advances in Information Retrieval - 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14-17, 2020, Proceedings, Part I*, J.M. Jose, E. Yilmaz, J. Magalhães, P. Castells, N. Ferro, M.J. Silva and F. Martins, eds, Lecture Notes in Computer Science, Vol. 12035, Springer, 2020, pp. 17–32. doi:10.1007/978-3-030-45439-5_2.

[17] S. Rudolph, Foundations of description logics, in: *Reasoning Web International Summer School*, Springer, 2011, pp. 76–136.

[18] Y. Dai, S. Wang, N.N. Xiong and W. Guo, A Survey on Knowledge Graph Embedding: Approaches, Applications and Benchmarks, *Electronics* (2020).

[19] Z. Wang, J. Zhang, J. Feng and Z. Chen, Knowledge Graph Embedding by Translating on Hyperplanes, in: *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada*, C.E. Brodley and P. Stone, eds, AAAI Press, 2014, pp. 1112–1119. http://www.aaai.org/ocs/index.php/AAAI/AAAI14/paper/view/8531.

[20] Z. Sun, Z. Deng, J. Nie and J. Tang, RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space, in: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, OpenReview.net, 2019. https://openreview.net/forum?id=HkgEQnRqYQ.

[21] B. Yang, W. Yih, X. He, J. Gao and L. Deng, Embedding Entities and Relations for Learning and Inference in Knowledge Bases, in: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, eds, 2015. http://arxiv.org/abs/1412.6575.

[22] T. Trouillon, J. Welbl, S. Riedel, E. Gaussier and G. Bouchard, Complex Embeddings for Simple Link Prediction, in *ICML'16*, JMLR.org, 2016, pp. 2071–2080–.

[23] D.Q. Nguyen, T.D. Nguyen, D.Q. Nguyen and D.Q. Phung, A Novel Embedding Model for Knowledge Base Completion Based on Convolutional Neural Network, in: *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*, M.A. Walker, H. Ji and A. Stent, eds, Association for Computational Linguistics, 2018, pp. 327–333. doi:10.18653/v1/n18-2053.
[24] M.S. Schlichtkrull, T.N. Kipf, P. Bloem, R. van den Berg, I. Titov and M. Welling, Modeling Relational Data with Graph Convolutional Networks, in: *The Semantic Web - 15th International Conference, ESWC, Proceedings*, A. Gangemi, R. Navigli, M. Vidal, P. Hitzler, R. Troncy, L. Hollink, A. Tordai and M. Alam, eds, Lecture Notes in Computer Science, Vol. 10843, Springer, 2018, pp. 593–607. doi:10.1007/978-3-319-93417-4_38.
[25] Y. Lin, Z. Liu, H. Luan, M. Sun, S. Rao and S. Liu, Modeling Relation Paths for Representation Learning of Knowledge Bases, in: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, L. Màrquez, C. Callison-Burch, J. Su, D. Pighin and Y. Marton, eds, The Association for Computational Linguistics, 2015, pp. 705–714. doi:10.18653/v1/d15-1082.
[26] J. Feng, M. Huang, Y. Yang and X. Zhu, GAKE: Graph Aware Knowledge Embedding, in: *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, N. Calzolari, Y. Matsumoto and R. Prasad, eds, ACL, 2016, pp. 641–651. https://aclanthology.org/C16-1062/.
[27] P. Ristoski and H. Paulheim, RDF2Vec: RDF Graph Embeddings for Data Mining, in: *The Semantic Web - ISWC 2016 - 15th International Semantic Web Conference, Kobe, Japan, October 17-21, 2016, Proceedings, Part I*, P. Groth, E. Simperl, A.J.G. Gray, M. Sabou, M. Krötzsch, F. Lécué, F. Flöck and Y. Gil, eds, Lecture Notes in Computer Science, Vol. 9981, 2016, pp. 498–514. doi:10.1007/978-3-319-46523-4_30.
[28] R. Xie, Z. Liu and M. Sun, Representation Learning of Knowledge Graphs with Hierarchical Types, in: *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI*, S. Kambhampati, ed., IJCAI/AAAI Press, 2016, pp. 2965–2971. http://www.ijcai.org/Abstract/16/421.
[29] J. Xu, X. Qiu, K. Chen and X. Huang, Knowledge Graph Representation with Jointly Structural and Textual Encoding, in: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, C. Sierra, ed., ijcai.org, 2017, pp. 1318–1324. doi:10.24963/ijcai.2017/183.
[30] R. Biswas, H. Sack and M. Alam, MADLINK: Attentive Multihop and Entity Descriptions for Link Prediction in Knowledge Graphs, *Semantic Web Journal*.
[31] H. Paulheim and C. Bizer, Type Inference on Noisy RDF Data, in: *ISWC*, 2013.
[32] R. Biswas, R. Sofronova, H. Sack and M. Alam, Cat2Type: Wikipedia Category Embeddings for Entity Typing in Knowledge Graphs, in: *K-CAP: Knowledge Capture Conference*, 2021.
[33] R. Biswas, J. Portisch, H. Paulheim, H. Sack and M. Alam, Entity Type Prediction Leveraging Graph Walks and Entity Descriptions, in: *International Semantic Web Conference (ISWC)*, 2022.
[34] A. Melo, H. Paulheim and J. Völker, Type Prediction in RDF Knowledge Bases Using Hierarchical Multilabel Classification, in: *WIMS*, 2016.
[35] Y. Yaghoobzadeh, H. Adel and H. Schütze, Corpus-Level Fine-Grained Entity Typing, *J. Artif. Intell. Res.* (2018).
[36] H. Jin, L. Hou, J. Li and T. Dong, Attributed and Predictive Entity Embedding for Fine-Grained Entity Typing in Knowledge Bases, in: *International Conference on Computational Linguistics*, 2018.
[37] W.X. Wilcke, P. Bloem, V. de Boer, R.H. van't Veer and F. van Harmelen, End-to-End Entity Classification on Multimodal Knowledge Graphs, *arXiv* (2020).
[38] H. Jin, L. Hou, J. Li and T. Dong, Fine-Grained Entity Typing via Hierarchical Multi Graph Convolutional Networks, in: *Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing*, 2019.
[39] Y. Zhao, A. Zhang, R. Xie, K. Liu and X. Wang, Connecting Embeddings for Knowledge Graph Entity Typing, in: *Annual Meeting of the Association for Computational Linguistics*, 2020.
[40] J. Zhuo, Q. Zhu, Y. Yue, Y. Zhao and W. Han, A Neighborhood-Attention Fine-grained Entity Typing for Knowledge Graph Completion, in: *ACM International Conference on Web Search and Data Mining*, 2022.
[41] T. Weller and M. Acosta, Predicting instance type assertions in knowledge graphs using stochastic neural networks, in: *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 2111–2118.
[42] B. Xu, Y. Zhang, J. Liang, Y. Xiao, S. Hwang and W. Wang, Cross-Lingual Type Inference, in: *International Conference Database Systems for Advanced Applications, DASFAA*, 2016.
[43] Y. Yaghoobzadeh and H. Schütze, Multi-level Representations for Fine-Grained Typing of Knowledge Base Entities, in: *Conference of the European Chapter of the Association for Computational Linguistics*, 2017.
[44] C. Meilicke, M.W. Chekol, D. Ruffinelli and H. Stuckenschmidt, Anytime Bottom-Up Rule Learning for Knowledge Graph Completion, in: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, S. Kraus, ed., ijcai.org, 2019, pp. 3137–3143. doi:10.24963/ijcai.2019/435.
[45] C. Meilicke, M. Fink, Y. Wang, D. Ruffinelli, R. Gemulla and H. Stuckenschmidt, Fine-Grained Evaluation of Rule- and Embedding-Based Systems for Knowledge Graph Completion, in: *SEMWEB*, 2018.
[46] F. Yang, Z. Yang and W.W. Cohen, Differentiable Learning of Logical Rules for Knowledge Base Reasoning, in: *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, 2017, pp. 2316–2325–. ISBN 9781510860964.
[47] A. Sadeghian, M. Armandpour, P. Ding and D.Z. Wang, *DRUM: End-to-End Differentiable Rule Mining on Knowledge Graphs*, 2019.
[48] W.L. Hamilton, R. Ying and J. Leskovec, Inductive Representation Learning on Large Graphs, in: *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, 2017, pp. 1025–1035–. ISBN 9781510860964.

[49] D. Daza, M. Cochez and P.T. Groth, Inductive Entity Representations from Text via Link Prediction, *Proceedings of the Web Conference 2021* (2021).

[50] T. Hamaguchi, H. Oiwa, M. Shimbo and Y. Matsumoto, Knowledge Transfer for Out-of-Knowledge-Base Entities : A Graph Neural Network Approach, in: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 2017, pp. 1802–1808. doi:10.24963/ijcai.2017/250.

[51] P. Wang, J. Han, C. Li and R. Pan, Logic Attention Based Neighborhood Aggregation for Inductive Knowledge Graph Embedding, in: *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19, AAAI Press, 2019. ISBN 978-1-57735-809-1. doi:10.1609/aaai.v33i01.33017152.

[52] K. Teru, E. Denis and W. Hamilton, Inductive Relation Prediction by Subgraph Reasoning, in: *Proceedings of the 37th International Conference on Machine Learning*, H.D. III and A. Singh, eds, Proceedings of Machine Learning Research, Vol. 119, PMLR, 2020, pp. 9448–9457. https://proceedings.mlr.press/v119/teru20a.html.

[53] X. Wang, T. Gao, Z. Zhu, Z. Liu, J.-Z. Li and J. Tang, KEPLER: A Unified Model for Knowledge Embedding and Pre-trained Language Representation, *Transactions of the Association for Computational Linguistics* **9** (2021), 176–194.

[54] D. Daza, M. Cochez and P. Groth, Inductive Entity Representations from Text via Link Prediction, in: *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, J. Leskovec, M. Grobelnik, M. Najork, J. Tang and L. Zia, eds, ACM / IW3C2, 2021, pp. 798–808. doi:10.1145/3442381.3450141.

[55] M. Ali, M. Berrendorf, M. Galkin, V. Thost, T. Ma, V. Tresp and J. Lehmann, Improving Inductive Link Prediction Using Hyper-Relational Facts, in: *SEMWEB*, 2021.

[56] G.A. Gesese, H. Sack and M. Alam, RAILD: Towards Leveraging Relation Features for Inductive Link Prediction In Knowledge Graphs, in: *Proceedings of the 11th International Joint Conference on Knowledge Graphs, IJCKG 2022, Hangzhou, China, October 27-28, 2022*, A. Artale, D. Calvanese, H. Wang and X. Zhang, eds, ACM, 2022, pp. 82–90. https://doi.org/10.1145/3579051.3579066.

[57] L. Yao, C. Mao and Y. Luo, KG-BERT: BERT for Knowledge Graph Completion, *CoRR abs/1909.03193* (2019). http://arxiv.org/abs/1909.03193.

[58] B. Kim, T. Hong, Y. Ko and J. Seo, Multi-Task Learning for Knowledge Graph Completion with Pre-trained Language Models, in: *Proceedings of the 28th International Conference on Computational Linguistics, COLING*, D. Scott, N. Bel and C. Zong, eds, International Committee on Computational Linguistics, 2020, pp. 1737–1743. doi:10.18653/v1/2020.coling-main.153.

[59] B. Wang, T. Shen, G. Long, T. Zhou, Y. Wang and Y. Chang, Structure-Augmented Text Representation Learning for Efficient Knowledge Graph Completion, in: *WWW '21: The Web Conference*, J. Leskovec, M. Grobelnik, M. Najork, J. Tang and L. Zia, eds, ACM / IW3C2, 2021, pp. 1737–1748. doi:10.1145/3442381.3450043.

[60] X. Lv, Y. Lin, Y. Cao, L. Hou, J. Li, Z. Liu, P. Li and J. Zhou, Do Pre-trained Models Benefit Knowledge Graph Completion? A Reliable Evaluation and a Reasonable Approach, in: *Findings of the Association for Computational Linguistics: ACL*, S. Muresan, P. Nakov and A. Villavicencio, eds, Association for Computational Linguistics, 2022, pp. 3570–3581. doi:10.18653/v1/2022.findings-acl.282.

[61] X. Xie, N. Zhang, Z. Li, S. Deng, H. Chen, F. Xiong, M. Chen and H. Chen, From Discrimination to Generation: Knowledge Graph Completion with Generative Transformer, in: *Companion of The Web Conference*, F. Laforest, R. Troncy, E. Simperl, D. Agarwal, A. Gionis, I. Herman and L. Médini, eds, ACM, 2022, pp. 162–165.

[62] C. Chen, Y. Wang, B. Li and K. Lam, Knowledge Is Flat: A Seq2Seq Generative Framework for Various Knowledge Graph Completion, in: *Proceedings of the 29th International Conference on Computational Linguistics, COLING*, N. Calzolari, C. Huang, H. Kim, J. Pustejovsky, L. Wanner, K. Choi, P. Ryu, H. Chen, L. Donatelli, H. Ji, S. Kurohashi, P. Paggio, N. Xue, S. Kim, Y. Hahm, Z. He, T.K. Lee, E. Santus, F. Bond and S. Na, eds, International Committee on Computational Linguistics, 2022, pp. 4005–4017. https://aclanthology.org/2022.coling-1.352.

[63] R. Xie, Z. Liu and M. Sun, Representation Learning of Knowledge Graphs with Hierarchical Types, in: *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, S. Kambhampati, ed., IJCAI/AAAI Press, 2016, pp. 2965–2971. http://www.ijcai.org/Abstract/16/421.

[64] S. Ma, J. Ding, W. Jia, K. Wang and M. Guo, TransT: Type-Based Multiple Embedding Representations for Knowledge Graph Completion, in: *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2017, Skopje, Macedonia, September 18-22, 2017, Proceedings, Part I*, M. Ceci, J. Hollmén, L. Todorovski, C. Vens and S. Dzeroski, eds, Lecture Notes in Computer Science, Vol. 10534, Springer, 2017, pp. 717–733. doi:10.1007/978-3-319-71249-9_43.

[65] R. Zhang, F. Kong, C. Wang and Y. Mao, Embedding of Hierarchically Typed Knowledge Bases, in: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18)*, S.A. McIlraith and K.Q. Weinberger, eds, AAAI Press, 2018, pp. 2046–2053.

[66] J. Hao, M. Chen, W. Yu, Y. Sun and W. Wang, Universal Representation Learning of Knowledge Bases by Jointly Embedding Instances and Ontological Concepts, in: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, A. Teredesai, V. Kumar, Y. Li, R. Rosales, E. Terzi and G. Karypis, eds, ACM, 2019, pp. 1709–1719. doi:10.1145/3292500.3330838.

[67] G. Niu, B. Li, Y. Zhang, S. Pu and J. Li, AutoETER: Automated Entity Type Representation with Relation-Aware Attention for Knowledge Graph Embedding, in: *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, T. Cohn, Y. He and Y. Liu, eds, Findings of ACL, Vol. EMNLP 2020, Association for Computational Linguistics, 2020, pp. 1172–1181. doi:10.18653/v1/2020.findings-emnlp.105.

[68] Y. Wang, H. Wang, J. He, W. Lu and S. Gao, TAGAT: Type-Aware Graph Attention neTworks for reasoning over knowledge graphs, *Knowl. Based Syst.* **233** (2021), 107500. doi:10.1016/j.knosys.2021.107500.
[69] Y. Zhao, Z. Li, W. Deng, R. Xie and Q. Li, Learning entity type structured embeddings with trustworthiness on noisy knowledge graphs, *Knowl. Based Syst.* **215** (2021), 106630. doi:10.1016/j.knosys.2020.106630.
[70] M. Kulmanov, W. Liu-Wei, Y. Yan and R. Hoehndorf, EL Embeddings: Geometric construction of models for the Description Logic EL ++, *CoRR* **abs/1902.10499** (2019). http://arxiv.org/abs/1902.10499.
[71] M. Kulmanov, W. Liu-Wei, Y. Yan and R. Hoehndorf, EL Embeddings: Geometric Construction of Models for the Description Logic EL++, in: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, S. Kraus, ed., ijcai.org, 2019, pp. 6103–6109. doi:10.24963/ijcai.2019/845.
[72] S. Mondal, S. Bhatia and R. Mutharaju, EmEL++: Embeddings for EL++ Description Logic, in: *Proceedings of the AAAI 2021 Spring Symposium on Combining Machine Learning and Knowledge Engineering (AAAI-MAKE 2021), Stanford University, Palo Alto, California, USA, March 22-24, 2021*, A. Martin, K. Hinkelmann, H. Fill, A. Gerber, D. Lenat, R. Stolle and F. van Harmelen, eds, CEUR Workshop Proceedings, Vol. 2846, CEUR-WS.org, 2021. https://ceur-ws.org/Vol-2846/paper19.pdf.
[73] B. Xiong, N. Potyka, T. Tran, M. Nayyeri and S. Staab, Faithful Embeddings for EL^{++} Knowledge Bases, in: *The Semantic Web - ISWC 2022 - 21st International Semantic Web Conference, Virtual Event, October 23-27, 2022, Proceedings*, U. Sattler, A. Hogan, C.M. Keet, V. Presutti, J.P.A. Almeida, H. Takeda, P. Monnin, G. Pirrò and C. d'Amato, eds, Lecture Notes in Computer Science, Vol. 13489, Springer, 2022, pp. 22–38. doi:10.1007/978-3-031-19433-7_2.
[74] X. Peng, Z. Tang, M. Kulmanov, K. Niu and R. Hoehndorf, Description Logic EL++ Embeddings with Intersectional Closure, *CoRR* **abs/2202.14018** (2022). https://arxiv.org/abs/2202.14018.
[75] M. Jackermeier, J. Chen and I. Horrocks, Box^{2}EL: Concept and Role Box Embeddings for the Description Logic EL++, *CoRR* **abs/2301.11118** (2023). doi:10.48550/arXiv.2301.11118.
[76] F. Zhapa-Camacho and R. Hoehndorf, CatE: Embedding ALC ontologies using category-theoretical semantics, *CoRR* **abs/2305.07163** (2023). doi:10.48550/arXiv.2305.07163.
[77] C. d'Amato, N.F. Quatraro and N. Fanizzi, Injecting Background Knowledge into Embedding Models for Predictive Tasks on Knowledge Graphs, in: *The Semantic Web*, R. Verborgh, K. Hose, H. Paulheim, P.-A. Champin, M. Maleshkova, O. Corcho, P. Ristoski and M. Alam, eds, Springer International Publishing, Cham, 2021, pp. 441–457. ISBN 978-3-030-77385-4.
[78] J. Chen, P. Hu, E. Jiménez-Ruiz, O.M. Holter, D. Antonyrajah and I. Horrocks, OWL2Vec*: embedding of OWL ontologies, *Mach. Learn.* **110**(7) (2021), 1813–1845. https://doi.org/10.1007/s10994-021-05997-6.
[79] D. Ruffinelli, S. Broscheit and R. Gemulla, You CAN Teach an Old Dog New Tricks! On Training Knowledge Graph Embeddings, in: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, OpenReview.net, 2020. https://openreview.net/forum?id=BkxSmlBFvr.
[80] M. Ali, M. Berrendorf, C.T. Hoyt, L. Vermue, M. Galkin, S. Sharifzadeh, A. Fischer, V. Tresp and J. Lehmann, Bringing Light Into the Dark: A Large-Scale Evaluation of Knowledge Graph Embedding Models Under a Unified Framework, *IEEE Trans. Pattern Anal. Mach. Intell.* **44**(12) (2022), 8825–8845.
[81] K. Toutanova and D. Chen, Observed versus latent features for knowledge base and text inference, in: *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, 2015.
[82] N. Hubert, P. Monnin, A. Brun and D. Monticolo, Sem@K: Is my knowledge graph embedding model semantic-aware?, *CoRR* **abs/2301.05601** (2023). doi:10.48550/arXiv.2301.05601.