Relevant Entity Selection: Knowledge Graph Bootstrapping via Zero-Shot Analogical Pruning

Lucas Jarnac, Miguel Couceiro, Pierre Monnin

Relevant Entity Selection: Knowledge Graph Bootstrapping via Zero-Shot Analogical Pruning

Lucas Jarnac
Orange
France
Université de Lorraine, CNRS, LORIA
Nancy, France
lucas.jarnac@orange.com

Miguel Couceiro
Université de Lorraine, CNRS, LORIA
Nancy, France
miguel.couceiro@loria.fr

Pierre Monnin
Orange
France
pierre.monnin@orange.com

ABSTRACT

Knowledge Graph Construction (KGC) can be seen as an iterative process starting from a high quality nucleus that is refined by knowledge extraction approaches in a virtuous loop. Such a nucleus can be obtained from knowledge existing in an open KG like Wikidata. However, due to the size of such generic KGs, integrating them as a whole may entail irrelevant content and scalability issues. We propose an analogy-based approach that starts from seed entities of interest in a generic KG, and keeps or prunes their neighboring entities. We evaluate our approach on Wikidata through two manually labeled datasets that contain either domain-homogeneous or -heterogeneous seed entities. We empirically show that our analogy-based approach outperforms LSTM, Random Forest, SVM, and MLP, with a drastically lower number of parameters. We also evaluate its generalization potential in a transfer learning setting. These results advocate for the further integration of analogy-based inference in tasks related to the KG lifecycle.

CCS CONCEPTS

• **Computing methodologies** → **Knowledge representation and reasoning**; *Artificial intelligence*; • **Information systems** → **Clustering and classification**; *World Wide Web.*

KEYWORDS

knowledge graph, construction, analogical inference, zero-shot learning, graph embedding

1 INTRODUCTION

Knowledge graphs (KGs) are "graphs of data intended to accumulate and convey knowledge of the real world, whose nodes represent entities of interest and whose edges represent relations between these entities" [16]. More formally, KGs are directed and labeled multigraphs $(\mathcal{E}, \mathcal{R}, \mathcal{T})$, where $\mathcal{E}$ is the set of entities, $\mathcal{R}$ is the set of relations, and $\mathcal{T}$ is the set of triples $\langle h, r, t\rangle \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}$, where r qualifies the relation holding between h and t. An example of such a triple could be $\langle$`BarackObama`, `instanceOf`, `Person`$\rangle$. KGs have proven useful in many academic and industrial applications, including search enhancement, question-answering, recommender systems, and eXplainable Artificial Intelligence [16, 35, 45].

Building and completing a KG can be achieved with knowledge extraction approaches from structured or unstructured data (*e.g.*, tables, texts) [40, 48]. This forms a virtuous loop in which the KG is both a supporting structure that provides entities and relations of interest to detect in data and the target structure to refine and complete. However, the cold start problem appears when the initial KG is empty, which motivates the need to build first a high quality nucleus [48]. Such a nucleus could be manually bootstrapped by experts, but this process is time-consuming. Some authors thus propose to focus on premium sources of entities and categories to automatically constitute the nucleus [48]. In this view, several works consider Wikidata [47], a large and generic KG collaboratively built to support Wikipedia, as a premium source [19, 41]. However, the sheer size of Wikidata entails a need to restrict Wikidata knowledge to be integrated into the KG nucleus to avoid irrelevant knowledge and scalability issues. As Wikidata contains more than 100 million entities[1], authors adopt a distillation [41] or a pruning [19] process, in which seed entities[2] of interest are identified in Wikidata and only parts of their neighborhood are included in the KG nucleus (see Figure 1). The selection of the neighboring entities leverages the ontology hierarchy, either only upward [41], or both upward and downward [19]. The latter brings much more entities, which makes it more prone to gathering irrelevant knowledge in the KG nucleus. For instance, the downward neighbors of `Microsoft SharePoint` include `Content Management System`, a relevant entity to keep, and `Dating App`, an irrelevant one to prune. Jarnac and Monnin [19] thus use several pruning thresholds based on node degrees and distances in the embedding space but highlight the difficulty to set global thresholds when applied to heterogeneous entities with different distributions of degrees and distances. Additionally, to the best of our knowledge, there is no publicly available benchmark dataset to evaluate such approaches.

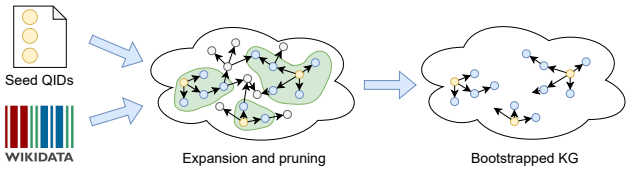


Figure 1: Outline of the bootstrapping of a knowledge graph from Wikidata.

In our work, we propose to tackle the limitations of fixed thresholds by training classifiers to select (or keep) relevant neighboring entities and prune irrelevant ones in a KG bootstrapping process. Specifically, we propose an analogy-based zero-shot approach. Analogies are quadruples of the form `Paris : France :: Berlin :`

[1] https://www.wikidata.org/wiki/Wikidata:Statistics
[2] Note that entities in Wikidata are identified by QIDs.

Germany, which can be read "Paris is to France as Berlin is to Germany", that simultaneously capture similarities and dissimilarities between objects [28, 31]. Analogical reasoning is a remarkable capability of the human mind, that has recently obtained impressive results on NLP tasks when applied on character and word embeddings [23, 27, 43]. Such a reasoning has also been proposed for KGs [18, 25, 32, 38, 49], using KG embeddings, *i.e.*, vector representations of KG entities and relations that preserve as much as possible the properties of the graph [5].

In our approach, we combine analogical reasoning and graph embedding to keep or prune neighboring entities. Our intuition is that the analogy-based model will be able to capture relative similarities and dissimilarities between seed entities and their neighbors to keep or to prune, and thus avoid the caveats of fixed thresholds that have difficulty generalizing to heterogeneous entities. Furthermore, our approach is zero-shot: the model learns to detect relative similarities and dissimilarities on a set of seed entities and their neighbors and can extrapolate to unseen seed entities and their neighbors. We experiment our approach on the Wikidata KG and two annotated datasets of seed entities and their neighbors to keep or prune, and that we make available for the benefit of the community. We empirically compare the behavior of our approach with several classifiers (*e.g.*, Random Forest, LSTM) and symbolic approaches (*e.g.*, depth pruning, threshold pruning [19]). We also assess the performance of the different approaches based on evaluation metrics and number of parameters. Moreover, we test the generalization of this analogy-based approach on a transfer learning setting.

The main contributions of the paper are:

- We propose an analogy-based zero-shot approach to select relevant entities in the neighborhood of seed entities. This approach only needs training examples of entities to keep or prune for some seed entities and can extrapolate to new seed entities without selection / pruning examples for them.
- We present a comparative study of our analogy-based approach to other methodologies with respect to different performance metrics, the number of parameters to be trained and the generality of the models considered.
- We provide two annotated datasets of seed entities and relevant or irrelevant neighbors in Wikidata to start constituting publicly available benchmarks for the community.

The remainder of this article is structured as follows. We briefly survey related work about KG bootstrapping and analogy-based inference in Section 2, and we detail our analogy-based zero-shot approach to select relevant entities to bootstrap a KG in Section 3. We experiment with Wikidata and two datasets in Section 4 and we discuss our results in Section 5. Finally, Section 6 summarizes our work and outlines future research work.

2 RELATED WORK

Our work positions within approaches focusing on bootstrapping KGs, especially by pruning to limit the scope of the built KG. We review some prominent works of this line of research in Subsection 2.1. Additionally, we rely on analogical reasoning which has recently achieved significant performance on NLP-related tasks, and has been identified as a promising research direction for KG-related tasks as outlined in Subsection 2.2.

2.1 KG Bootstrapping and Pruning

The construction of ontologies and KGs usually entails the possibility of their reuse for other purposes. That is why, it is common to leverage existing ontologies and KGs to bootstrap others [15, 44], since they can be seen as premium sources [48]. To illustrate, YAGO3 combines the taxonomy of WordNet and the categories of Wikipedia pages [26], Knowledge Vault integrates the FreeBase KG [11], and PGxLOD first integrates several biomedical KGs to then interconnect and enrich them [33].

Due to the size and generic aspect of some KGs and ontologies, some authors resort to pruning to construct domain-specific KGs from them. One of the early examples is the work of Swartout *et al.* in 1996 [44] where they propose to build a domain-specific ontology starting from a large and generic ontology, SENSUS, of 50,000+ concepts. To do so, they start with seed concepts from the domain of interest that are manually linked to SENSUS. They then include all super-concepts up to the root. They also discuss that some subtrees bring additional concepts of interest. They manually identify them with the rationale that if some nodes of a subtree have been identified relevant, then the other nodes of the subtree may be relevant too. However, such a manual process is time-consuming. Furthermore, incorporating subtrees of large ontologies may come at the expense of incorporating irrelevant knowledge, which is difficult to manually assess. To face such issues, automatic distillation or pruning approaches can be considered. The distillation process can be guided by documents from the domain of interest. For instance, Babayeva *et al.* develop a domain ontology for Cyber Defence exercises by collecting concepts from an existing ontology and documents on this topic [4]. Shbita *et al.* [41] build a KG about customer requirements starting from client verbatim and Wikidata. Specifically, they detect entities in text, link them to Wikidata, and integrate their direct classes and all their super-classes.

Regarding pruning approaches, they can be classified into two categories: *aggressive pruning* based on topology of the graph and *soft pruning* that requires human input to define the relevant taxonomic concepts. In [14], Faralli *et al.* introduce the CrumbTrail algorithm that prunes a directed noisy knowledge graph with the aim of obtaining an acyclic subgraph that contains all previously selected seed nodes. Using this CrumbTrail algorithm, Bordea *et al.* propose to build domain-specific taxonomies from the KG of Wikipedia categories. After a user has selected leaf nodes and a root node, the algorithm is applied on the KG to build the directed and acyclic graph that will form the taxonomy. They provide three datasets but they are not directly applicable to our task. Indeed, they have specific concerns w.r.t. to taxonomy building (*e.g.*, upward extension from leaves to root, acyclic aspect) while we mainly focus on gathering terms of interest w.r.t. a domain without such concerns. Additionally, we consider that seed nodes may not be leaves and propose to perform a downward expansion (see Subsection 3.1). This is a more difficult pruning task since not all subclasses of a class of interest may be relevant. In this view, Jarnac and Monnin [19] bootstrap an enterprise KG by expanding a set of business terms semi-manually aligned to Wikidata entities along their ontology

hierarchy. According to the authors, the distance in the embedding space appears to be a good indicator of topic similarity or drift. Thus, to limit the expansion, they propose an automatic approach relying on node degree and distance thresholds. However, such thresholds are globally fixed for all seed entities, which may lead to varying performance when these entities belong to heterogeneous domains.

2.2 Analogy-Based Inference in KGs

Analogy-based inference is a basic process in human cognition [6, 31] that is tightly related to abstraction, adaptation and creativity. Analogy-based inference can be viewed as transferring knowledge from a source domain to a different, but somewhat similar, target domain by leveraging simultaneously on similarities and dissimilarities. Most of the literature in analogy-based inference is built on the notion of *analogical proportions, i.e.,* statements of the form "A is to B as C is to D" represented as $A : B :: C : D$ [28], and relies on two main tasks, namely, *analogy detection* that involves deciding whether a quadruple (A, B, C, D) constitutes a valid analogy $A : B :: C : D$, and *analogy solving* that consists in finding the possible elements X that make $A : B :: C : X$ a valid analogy.

When the underlying objects A, B, C and D are represented as vectors e_A, e_B, e_C and e_D, respectively, in some vector space $\mathbb{R}^n$, analogical proportions can be thought of in geometric terms as the parallelogram rule $e_D - e_C = e_B - e_A$. For instance, the underlying elements A, B, C and D of the analogical proportion could be words [46], and e_A, e_B, e_C and e_D their vectorial representations [29, 30] or larger chunks of text such as sentences [1, 2, 51]. Analogy-based inference has been used to solve hard reasoning tasks and has shown its potential with competitive results in several machine learning tasks such as classification, decision making and recommendation [9, 12, 13, 17], in data augmentation through analogical extrapolation for model learning, especially in environments with few labeled examples [7, 8]. Moreover, it has been successfully applied in classical natural language processing (NLP) tasks such as machine translation [21], several semantic [23, 24] and morphological tasks [3, 27, 34], as well as in (visual) question answering [39], solving puzzles and scholastic aptitude tests [37], and target sense verification [50].

Analogy-based inference can also be used to address and tackle tasks related to the KG lifecycle such as semantic table interpretation or knowledge matching [32]. A few works already exist in this line of research and rely on graph embedding, similarly to NLP approaches relying on character or word embeddings. For example, Liu *et al.* [25] tackles the task of link prediction, *i.e.*, completing triples $(h, r, ?)$, and study whether KG embedding models respect the parallelogram rule of analogical inference. They show that modeling analogical structures in KG embedding models brings additional performance. Similarly, Yao *et al.* [49] propose a model based on analogy functions to enhance a KG embedding model for link prediction. Alternatively, Portisch *et al.* [38] evaluate whether KG embedding models for link prediction or data mining can be used for analogy detection. In a similar fashion, we propose to leverage KG embeddings in an analogy detection task, where analogies serve to detect relevant or irrelevant entities w.r.t. seed entities of interest.

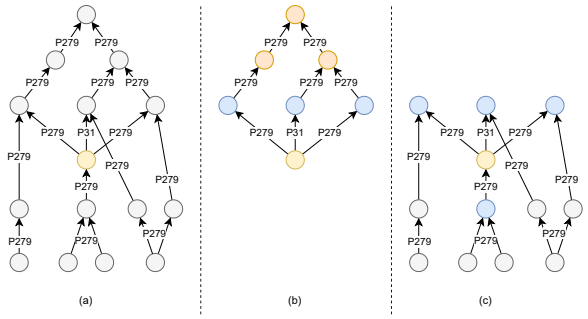


Figure 2: Expansion from a seed entity along the ontology hierarchy. P31 stands for "instance of" and P279 stands for "subclass of". (a) depicts the full ontology hierarchy, (b) depicts classes reached in the upward expansion, and (c) depicts classes reached in the downward expansion.

3 ANALOGY-BASED ZERO-SHOT SELECTION OF RELEVANT ENTITIES

We consider that we have at our disposal a set of seed entities of interest. Such entities can be identified (semi-)manually by experts and/or via an automatic extraction of entities from texts in the domains of interest [19, 41, 44]. These seed entities are aligned with a generic KG, *e.g.*, Wikidata, and we retrieve their neighboring entities of interest along the ontology hierarchy. We first describe how we traverse the ontology hierarchy (Subsection 3.1), and then how we keep relevant entities and prune irrelevant ones during this traversal using an analogy-based model (Subsection 3.2).

3.1 Expansion Along the Ontology Hierarchy

We retrieve the neighboring entities of seed entities of interest along the ontology hierarchy as illustrated in Figure 2, which distinguishes two directions for the expansion.

In the upward expansion (Figure 2b), we retrieve from a seed entity, its first-level classes following P31 ("instance of") and P279 ("subclass of") edges, *i.e.*, we retrieve the classes that the entity directly instantiates, or those that directly subsume it. Then, we retrieve all their superclasses by following P279 edges up to the root of the hierarchy in a breadth-first search expansion.

In the downward expansion (Figure 2c), we retrieve from a seed entity, its first-level classes by following P31, P279, and reversed P279 edges, *i.e.*, we retrieve the classes that the entity directly instantiates, those that directly subsume it, or those that it directly subsumes. We then retrieve all their subclasses following reversed P279 edges up to the leaves, in a breadth-first search expansion.

3.2 Selection of Relevant Entities

In this subsection, we focus on the problem of keeping relevant and pruning irrelevant entities when traversed during the expansion described in Subsection 3.1.

3.2.1 Formalization. We formalize the problem as follows: given a seed entity e_s and an entity e_r reached during the graph expansion from e_s, our goal is to decide whether to keep or to prune e_r. If e_r is

kept, then its neighbors are explored as described in Subsection 3.1. Otherwise, its neighbors are not explored.

We propose an analogy-based zero-shot classifier model $\mathcal{A}$ such that:

$$\mathcal{A}(e_s, e_r) = \begin{cases} 1 \text{ if } e_r \text{ should be kept} \\ 0 \text{ if } e_r \text{ should be pruned} \end{cases} \quad (1)$$

Recall that analogies are statements of the form "A is to B as C is to D" represented as A : B :: C : D [27]. In our case, we use analogies of the form

$$e_s^1 : e_r^1 :: e_s^2 : e_r^2 \quad (2)$$

where e_s^1 and e_s^2 are two seed entities, e_r^1 is reached during the expansion from e_s^1, and e_r^2 is reached during the expansion from e_s^2. Using analogical inference, if Equation (2) is a valid analogy and we know the decision for the pair (e_s^1, e_r^1), then we can extrapolate the decision for the pair (e_s^2, e_r^2). To illustrate, if the analogy `Hadoop : Big Data :: SharePoint : Content Management System` is valid and we know that Big Data should be kept, then Content Management System should also be kept.

We propose three different configurations for our analogy-based classifier, that consider different valid and invalid analogies. To ease notation, we note k a keeping decision for a pair, and p a pruning decision for a pair. The three configurations are as follows:

Configuration C_1 Valid analogies are of the form $k :: k$. Invalid analogies are of the form $k :: p$.

Configuration C_2 Valid analogies are of the form $k :: k$. Invalid analogies are of the form $k :: p$ and $p :: p$.

Configuration C_3 Valid analogies are of the form $k :: k$ and $p :: p$. Invalid analogies are of the form $p :: k$ and $k :: p$

It is worth noting that depending on the chosen configuration, the aforementioned analogical inference is adapted. For example, with C_1 and C_2, valid analogies only allow to extrapolate keeping decisions. On the contrary, with C_3, valid analogies can conclude on keeping or pruning the pair (e_s^2, e_r^2), depending on the known decision for the pair (e_s^1, e_r^1). The same rationale can be applied on invalid analogies. For example, with C_1, invalid analogies lead to a pruning decision for the pair (e_s^2, e_r^2). On the contrary, with C_3, invalid analogies lead to decide for the pair (e_s^2, e_r^2) the opposite decision of the pair (e_s^1, e_r^1).

In addition to configurations, we propose to consider or not paths in the graph within the analogy-based model. To illustrate, consider the analogy in Equation (2) and the two following expansion paths that generated it:

$$e_s^1 \rightarrow e_r^3 \rightarrow e_r^1$$
$$e_s^2 \rightarrow e_r^4 \rightarrow e_r^5 \rightarrow e_r^2$$

Our first formalization in Equation (2) only considers seed entities and reached entities, on which a decision is known or a decision is to be made by the model. We also propose to consider the paths leading to the reached entities. In this view, the analogy in Equation (2) becomes as follows:

$$e_s^1 : (e_r^3, e_r^1) :: e_s^2 : (e_r^4, e_r^5, e_r^2). \quad (3)$$

3.2.2 Model. We adopt the supervised machine learning model proposed by Lim *et al.* [23]. This model is presented in Figure 3 and relies on convolutional neural networks (CNNs). It takes as input the vector embeddings of each constituent of a quadruple.

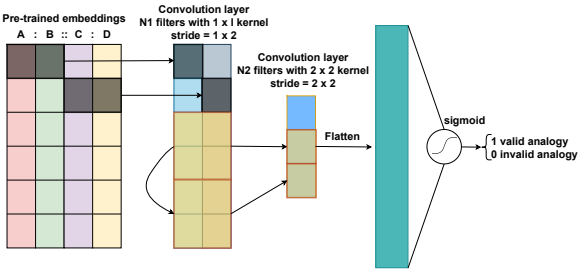


Figure 3: The analogy-based classifier model with two convolution layers and one fully connected layer from [23].

In our case, for a quadruple without paths (*i.e.*, Equation (2)), we simply concatenate the embeddings of the entities (*i.e.*, e_s^1, e_r^1, e_s^2, e_r^2). For a quadruple with paths (*i.e.*, Equation (3)), we concatenate the embeddings of the entities (*i.e.*, e_s^1, e_r^3, e_r^1, e_s^2, e_r^4, e_r^5, e_r^2) and use zero-padding in order to respect the model's fixed input dimension. We experimented with three zero-padding methods:

before zeros are added before the sequence (*e.g.*, before e_s^1 and before e_s^2)

between zeros are added between the embedding of the seed entity and the embeddings of the entities in the path (*e.g.*, between e_s^1 and e_r^3, and between e_s^2 and e_r^4)

after zeros are added after the embeddings of the entities in the path (*e.g.*, after e_r^1 and after e_r^2)

The model has two convolution layers, followed by a flattening operation and one fully connected layer. The first convolution layer is composed of n_1 filters. Each filter has a kernel size of $1 \times$ sequence length and a ReLU activation function. Filters are initialized with a He normal initializer and applied with a stride of $(1,$ sequence length$)$. The second convolution layer is composed of n_2 filters. Each filter has a kernel size of 2×2 and a ReLU activation function. Filters are initialized with a He normal initializer and applied with a stride of $(2, 2)$. The last layer is a fully connected layer with one output and a sigmoid activation function to obtain a binary classification score in $[0, 1]$. We also add dropout after each convolution layer.

This model is well-suited to the task at hand. Indeed, the first convolution layer allows to compute dissimilarities for each pair of entities, while the second convolution layer compares these dissimilarities between the first and the second pairs forming the quadruple to classify as a valid or invalid analogy.

3.2.3 Training. We consider that we have at our disposal pairs (e_s^1, e_r^1) whose keeping or pruning decision is known (*e.g.*, annotations by experts).

To train our model, for each annotated pair (e_s^1, e_r^1), and for each form of valid and invalid analogies of the considered configuration, we build M analogies by sampling M other adequate labeled pairs. To illustrate, in configuration C_1, invalid analogies are of the form $k :: p$. Thus, for a pair (e_s^1, e_r^1) whose decision is keep, we build M invalid analogies by selecting M other pairs whose decision is prune. These M pairs are selected by ascending order of proximity of seed entities in the embedding space. We then train our model by

minimizing the binary cross-entropy loss and taking into account possible unbalancing between valid and invalid analogies.

3.2.4 Inference. At inference, on an unknown pair (e_s^2, e_r^2):

(1) We select N pairs (e_s^1, e_r^1) whose decision is known to be keeping and N pairs (e_s^1, e_r^1) whose decision is known to be pruning. Specifically, for each type of decision, we order known pairs by ascending proximity of e_s^1 and e_s^2 in the embedding space and select the N first.
(2) We generate $2N$ quadruples with selected known pairs on the left and the unknown pair (e_s^2, e_r^2) on the right[3].
(3) For each of these quadruples, our model predicts whether it is a valid or invalid analogy, which constitutes a keeping or pruning prediction, depending on the chosen configuration (see Subsubsection 3.2.1).
(4) We compute the average of the scores output by the model (in $[0, 1]$) on each of the $2N$ quadruples as follows:
    - For C_1: we interpret the score as a vote for keeping
    - For C_2: we interpret the score as a vote for keeping
    - For C_3: *(i)* if the known pair (e_s^1, e_r^1) has a keeping decision, the score is considered as a vote for keeping; *(ii)* if the known pair (e_s^1, e_r^1) has a pruning decision, the score is considered as a vote for pruning. Indeed, in this case, a score close to 1 corresponds to a valid analogy of the form $p :: p$. A score close to 0 corresponds to an invalid analogy of the form $p :: k$. Thus, we use $1 - score$ as a vote for keeping.
(5) If the averaged keeping score is above a fixed threshold, we keep e_r^2. Otherwise we prune it.

It should be noted that, at inference, our model extrapolates on pairs in which e_s^2, and potentially e_r^2 were not seen in training. This makes our approach fundamentally zero-shot.

4 EXPERIMENTS

We evaluate our analogy-based model on the Wikidata knowledge graph [47] and two datasets containing seed entities and labeled keeping and pruning decisions for their neighboring entities. In particular, we compare the latter with baseline models such as Multi-Layer Perceptron (MLP), Long Short Term Memory (LSTM), Support Vector Machine (SVM), Random Forest, depth pruning, and threshold pruning [19]. We also evaluate our model in a transfer learning setting.

Since our approach requires KG embeddings, we experiment with the pre-trained embeddings of Wikidata available in PyTorch-BigGraph [22][4]. These embeddings were learned for more than 78,000,000 entities of the 2019-03-06 version of Wikidata. For building, training, and evaluating our models we used TensorFlow's Keras API and scikit-learn [36]. Datasets[5] and code[6] of our experiments are publicly available.

[3]It is noteworthy that for configuration C_1, the N pairs (e_s^1, e_r^1) whose decision is pruning are not used, leading to only N quadruples being generated.
[4]https://torchbiggraph.readthedocs.io/en/latest/pretrained_embeddings.html
[5]https://doi.org/10.5281/zenodo.8091584
[6]https://github.com/Orange-OpenSource/analogical-pruning

4.1 Datasets

To the best of our knowledge, there is no publicly available benchmark dataset for the present task. This motivated us to build and publicly release the two datasets whose characteristics are detailed in Table 1. Specifically, we gathered two sets of seed entities: 455 seed entities from the Computer Science / Information Technology domain for Dataset 1 (*e.g.,* entities related to telecommunications, network, or programming languages), and 105 seed entities from more heterogeneous domains for Dataset 2 (*e.g.,* entities related to food, music, sport, or science). Table 1 shows the number of nodes reached with an unconstrained (*i.e.,* without pruning) upward and downward expansion as described in Section 3.1 for both datasets. It can be noticed that the number of reached nodes upward is drastically lower than the number of reached nodes downward. This motivated us to solely focus on pruning during the downward expansion.

To obtain labeled keeping and pruning decisions for downward nodes for both datasets without having to label the whole neighborhood, we adopted the following process. We performed a downward expansion with the pruning approach proposed by Jarnac and Monnin [19] with thresholds based on node degrees and distance in the embedding space. To configure these thresholds, we set $\alpha = 1.5$, $\gamma = 20$, $\beta = 1.3$ following [19]. Accordingly to their proposal, we also consider two different embeddings for entities: *(i)* Embedding $\mathcal{E}_1$ in which the embedding of an entity is its vector in the considered pre-trained embeddings, and *(ii)* Embedding $\mathcal{E}_2$ in which the embedding of an entity is the centroid of the embeddings of its instances. If an entity does not have instances, its pre-trained embedding vector is used instead. Then, we manually labeled keeping and pruning decisions output by this approach on the two sets of seed entities. Note that, since we use pre-trained embeddings from 2019 and a Wikidata dump from 2022, some entities do not have embeddings. To deal with this problem, we filtered both datasets to ensure that all seed and reached entities have an embedding vector.

4.2 Experimental Setup

We now describe our experimental protocol.

4.2.1 Cross validation. We applied a 5-fold cross validation. We split the seed entities of each dataset into 5 sets $\mathcal{S}_i$, $i \in \{1, 2, 3, 4, 5\}$, where each set contains the same number of seed entities. Each set $\mathcal{S}_i$ is successively used for testing, while $\mathcal{S}_{(i-1)}$ is used for validation, and the remaining sets are used for training. To prevent over-fitting, we implement an early-stopping method based on the validation loss. We set the patience to 5 for models trained with 50 epochs, and to 20 for models trained with 200 epochs.

Such a splitting on seed entities at testing, guarantees that we evaluate the ability of the model to generalize on unseen seed entities. Additionally, some entities reached from these unseen seed entities are not seen during training either, as highlighted in Table 2. Such an experimental setup thus assesses the model's capability to learn a relative similarity or dissimilarity between seed entities and reached entities, and to extrapolate it on unseen seed and reached entities. This extrapolation roots our zero-shot approach.

It can be noticed in Table 2 that test seed entities in Dataset 1 lead to more entities that were seen in training than in Dataset 2 (51-61% instead of 10-21%). This is a direct consequence of the

Dataset		# Seed entities	# Nodes up	# Nodes down	# P decisions	Depths P	# K decisions	Depths K
Dataset 1	w/o filtering	455	1,507	2,593,609	3,464	$[\![1, 4]\!]$	1,769	$[\![1, 4]\!]$
	w/ filtering	439	1,469	2,593,575	2,910	$[\![1, 4]\!]$	1,619	$[\![1, 4]\!]$
Dataset 2	w/o filtering	105	1,159	1,247,385	388	$[\![1, 2]\!]$	594	$[\![1, 3]\!]$
	w/ filtering	104	1,152	1,247,383	314	$[\![1, 2]\!]$	577	$[\![1, 3]\!]$

Table 1: Statistics of Dataset 1 and Dataset 2 before and after filering to only retain entities with embeddings. K stands for Keeping and P stands for Pruning.

Dataset	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Dataset 1	56.51	59.61	51.72	55.76	61.19
Dataset 2	20.50	21.49	12.36	17.04	10.05

Table 2: Percentage of entities reached when testing and already seen when training, for each fold and dataset.

homogeneity of seed entities in Dataset 1. Since all seed entities are from the Computer Science / Information Technology domain, the entities traversed during the expansion from each seed entity may overlap. On the contrary, Dataset 2 involves heterogeneous seed entities from different domains, leading to different entities being traversed when expanding from each seed entity.

4.2.2 Transfer learning. We also tested our model in a transfer learning setting. We trained our model on labeled decisions of Dataset 1 and tested it by traversing the neighborhood of seed entities in Dataset 2.

4.3 Models

We compare our proposed analogy-based model to the following baseline models: MLP, LSTM, Random Forest, SVM, depth pruning, and threshold pruning [19]. We call *analogy* the model that does not considers paths (Equation (2)), and *path analogy* the model that consider paths (Equation (3)).

Note that the dimension of the pre-trained embeddings of Wikidata is 200. Some parameters are used by several models and are detailed below:

Batch size We set the batch size do 32.
Optimizer We use the Adam optimizer.
Embedding We consider the embeddings $\mathcal{E}_1$ and $\mathcal{E}_2$, as proposed in [19] and explained in Subsection 4.1.
Concatenation Consider a pair (e_s, e_r) formed by a seed entity and an entity reached. To decide whether to keep or prune e_r, some models can take as input the horizontal concatenation of the embeddings of e_s and e_r (called *horizontal*) or their difference (called *translation*).
Zero padding We consider three zero-padding methods: *before*, *between*, and *after*, as detailed in Subsubsection 3.2.2.
Learning rate We test with learning rates $\in \{0.0001, 0.001, 0.01\}$.
Dropout rate We test with dropout rates $\in \{0, 0.3, 0.5\}$.
Path length We consider paths of length $\in \{3, 4, 5\}$.
Analogy configuration We consider the three configurations for valid and invalid analogies C_1, C_2, C_3 presented in Subsubsection 3.2.1.
Number of filters We test with $(n_1, n_2) \in \{(2, 1), (4, 2), (8, 4), (16, 8), (32, 16), (64, 32), (128, 64), (256, 128)\}$.

The parameters used by the different considered models are given below, where we also describe specific parameters that are only applicable to one model.

Analogy (A) Batch size, optimizer, embedding, learning rate, dropout rate, analogy configuration, number of filters, $M \in \{5, 10, 15, 20, 50, 100\}$, $N = 20$.
Path analogy (PA) Batch size, optimizer, embedding, learning rate, dropout rate, zero padding, path length, analogy configuration, number of filters, $M \in \{5, 10, 15, 20, 50, 100\}$, and $N = 20$.
SVM Embedding, concatenation, and unlimited number of iterations.
Random Forest (RF) Embedding, concatenation, and number of estimators $\in \{10, 50, 100, 150, 200, 250, 300, 400, 500\}$.
MLP Batch size, optimizer, embedding, concatenation, learning rate, dropout rate, and hidden layers $\in \{(100), (100, 50), (100, 50, 25), (200), (200, 100), (200, 100, 50), (200, 100, 50, 25)\}$.
LSTM Batch size, optimizer, embedding, learning rate, zero padding, path length, and number of units $\in \{50, 100, 150\}$.
Depth pruning (D) Depth threshold $\in [\![1, 20]\!]$.
Threshold pruning (T) Embedding, $\alpha \in \{1.0, 1.1, \ldots, 2.0\}$, $\gamma = 20$, $\beta \in \{1.0, 1.1 \ldots, 2.0\}$, and absolute degree = 200.

Note that for models with a non-zero dropout rate, we use Monte Carlo Dropout.

For all models except analogy-based models, we explored all combinations of different parameter values. Given the important parameter space, for analogy-based models we first fixed the embedding to $\mathcal{E}_1$, and the configuration to C_1 on Dataset 2 in order to find the three best numbers of filters, the two best path lengths, the best zero padding method, and the best dropout rate. We then experimented with this reduced parameter space on Dataset 1 and Dataset 2.

4.4 Evaluation Metrics

Figure 4 illustrates the expansion along the ontology hierarchy from a seed entity (◯) with a pruning model. In such a setting, we only evaluate the model on nodes that are associated with a gold decision, *i.e.*, nodes depicted by ◯ are not considered. In this view, it should be noted that it is possible for the model to leave nodes with gold decisions unexplored due to erroneous pruning decisions

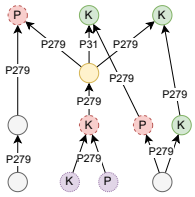


Figure 4: Example of an evaluation setting for a pruning model. Gold decisions are represented by the letters K (for keep) and P (for prune). ○ represents the seed entity; ○ are entities kept by the model; ○ are entities pruned by the model; ○ are entities unexplored by the model due to pruning decisions but that have a gold decision; ○ represent unlabeled entities that are not considered in the evaluation.

higher in the hierarchy (*i.e.*, nodes depicted by ○). To take into account these various cases in our evaluation, we use the following metrics:

$$\text{Precision} = \frac{K}{K + P} \qquad \text{Recall} = \frac{K}{K + K + K}$$

$$\text{Accuracy} = \frac{K + P}{K + P + K + P + K + P}$$

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

This corresponds to a binary classification in which the keeping decision is the positive class and the pruning decision is the negative class.

4.5 Results

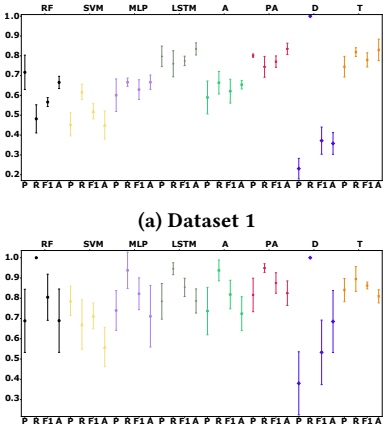
We introduce our results in this section following the two setups described in Section 4. We further discuss them in Section 5.

4.5.1 Cross validation. We present in Table 3 the performance of the different models on the task of keeping relevant entities and pruning irrelevant ones on Dataset 1 and Dataset 2. Figure 5 depicts these results with error plots to better assess the variability or stability of each model. Note that we present the results of the reference Threshold whose decisions were labeled to build the datasets. However, we do not use them to draw comparisons and conclusions because of the bias that would constitute using such results in both the dataset building and evaluation processes.

For each model, Table 3 and Figure 5 only present the best results in terms of F1-score (primary criterion) and accuracy (secondary criterion) that were obtained when exploring the parameter space. The best parameters were the following:

Analogy Embedding $\mathcal{E}_1$, $(n_1, n_2) = (16, 8)$, dropout = 0.5, configuration C_2 and learning rate = 0.001 (for Dataset 1), configuration C_1 and learning rate = 0.01 (for Dataset 2).

Path analogy Embedding $\mathcal{E}_1$, configuration C_1, learning rate = 0.001, zero-padding = between, path length = 4, $(n_1, n_2) = (16, 8)$, and dropout = 0 (for Dataset 1), path length = 3, $(n_1, n_2) = (4, 2)$, and dropout = 0.3 (for Dataset 2).



(a) Dataset 1

(b) Dataset 2

Figure 5: Error plot depicting the precision (P), recall (R), F1-score (F1), and accuracy (A) of each model on Dataset 1 and Dataset 2.

SVM Embedding $\mathcal{E}_1$, concatenation = horizontal (for Dataset 1), and concatenation = translation (for Dataset 2).

Random Forest concatenation = horizontal, embedding $\mathcal{E}_2$ and 200 estimators (for Dataset 1), embedding $\mathcal{E}_1$ and 300 estimators (for Dataset 2).

MLP hidden layers = (200, 100, 50), embedding $\mathcal{E}_1$, concatenation = horizontal, learning rate = 0.001, dropout = 0 (for Dataset 1), and Embedding $\mathcal{E}_2$, concatenation = translation, learning rate = 0.01, dropout = 0.3 (for Dataset 2).

LSTM Embedding $\mathcal{E}_1$, zero-padding = before, number of units = 150, path length = 5 and learning rate = 0.01 (for Dataset 1), path length = 3 and learning rate = 0.001 (for Dataset 2).

Depth pruning Depth threshold = 4 (for Dataset 1), and depth threshold = 3 (for Dataset 2).

Threshold pruning $\alpha = 1.0$, embedding $\mathcal{E}_1$, $\beta = 2.0$ (for Dataset 1), $\beta = 1.8$ (for Dataset 2).

4.5.2 Transfer learning. For our transfer learning setting, we used the best parameters found during the cross-validation on Dataset 1. We trained each model on 80% of Dataset 1, using 20% as validation for early-stopping. We then tested the trained models on Dataset 2. Results are presented in Table 4.

5 DISCUSSION

Table 3 highlights that depth in the ontology hierarchy cannot be used to keep relevant entities and prune irrelevant ones. Here, with a depth of 3-4, we obtain a perfect recall but a low precision. Using greater thresholds does not change results while lower ones would reduce the recall in favor of the precision. This was expected, especially in a collaborative and generic knowledge graph such as Wikidata, since different communities may have different granular representations of knowledge. To tackle this issue, one would need to specify different depth thresholds depending on the seed entity domains. Additionally, not all subclasses of an interesting

Model	Dataset 1				Dataset 2			
	P	R	F1	ACC	P	R	F1	ACC
Random Forest	71.68 ± 8.66	48.18 ± 7.10	56.66 ± 2.25	66.57 ± 3.09	68.85 ± 15.59	**100.00 ± 0.00**	80.50 ± 11.39	68.93 ± 15.62
SVM	45.43 ± 5.84	61.79 ± 3.90	52.02 ± 3.86	45.08 ± 7.09	78.73 ± 7.29	67.01 ± 12.18	71.29 ± 6.34	55.88 ± 9.63
MLP	60.12 ± 8.20	66.68 ± 2.09	62.94 ± 4.98	66.68 ± 3.62	73.99 ± 9.80	93.80 ± 8.96	82.22 ± 7.86	71.10 ± 15.18
LSTM	79.72 ± 5.17	**76.00 ± 6.59**	**77.43 ± 2.38**	83.48 ± 3.05	78.49 ± 8.80	94.58 ± 2.96	85.36 ± 4.53	78.66 ± 5.95
Analogy	58.99 ± 8.31	66.41 ± 5.67	62.15 ± 6.01	65.43 ± 2.16	73.67 ± 11.71	93.75 ± 5.12	81.85 ± 7.03	72.39 ± 8.38
Path analogy	**80.10 ± 0.84**	74.44 ± 5.28	77.06 ± 2.89	**83.51 ± 2.87**	**81.63 ± 8.27**	94.90 ± 2.16	**87.54 ± 5.05**	**82.50 ± 6.07**
Depth	23.06 ± 5.19	100.00 ± 0.00	37.19 ± 6.85	35.78 ± 5.48	38.01 ± 15.57	100.00 ± 0.00	53.30 ± 15.92	68.51 ± 15.30
Threshold	74.50 ± 5.22	81.86 ± 2.25	77.94 ± 3.58	83.02 ± 5.38	84.02 ± 5.72	89.51 ± 6.12	86.30 ± 1.66	80.96 ± 3.24

Table 3: Pruning evaluation results on Dataset 1 and Dataset 2, with the parameters leading to the best results for each model. P stands for average precision, R stands for average recall, F1 stands for average F1-score, and ACC stands for average accuracy. The best results are in bold and we underline the second best result. Also, we decided to present the reference threshold results used in the construction of both datasets.

Model	Dataset 1 → 2			
	P	R	F1	ACC
Random Forest	83.09	20.40	32.75	33.06
SVM	64.62	57.04	60.59	43.97
MLP	69.90	51.99	59.63	48.26
LSTM	**92.83**	74.73	82.80	80.04
Analogy	74.95	64.26	69.19	59.86
Path analogy	91.49	**83.39**	**87.25**	**84.33**

Table 4: Transfer results obtained by training on Dataset 1 and testing on Dataset 2. P stands for average precision, R stands for average recall, F1 stands for average F1-score, and ACC stands for average accuracy. The best results are in bold and the second best are underlined.

Model	Dataset 1	Dataset 2
LSTM	210,751	210,751
MLP	105,401	65,401
Analogy	1,369	1,369
Path analogy	1,401	251

Table 5: Number of trainable parameters for each model.

class may be of interest w.r.t. a seed entity, especially in case of errors in the ontology hierarchy (*e.g.*, erroneous subclass edges, misinterpretation of the subclass semantics by users, introduction of cycles). Such results motivate the use of classifiers to learn a relative similarity and dissimilarity between reached entities and seed entities, based on labeled examples given by a user (*e.g.*, an expert, the KG owner).

Regarding classifier performance, the LSTM and path analogy models are the best performing models, which outlines the importance of paths leading to entities to decide on their relevance. Figure 5 indicates that the LSTM and path analogy models are the more stable, with the path analogy model being particularly stable on Dataset 1 for the precision metric. On this dataset, the path analogy model obtains the best results in precision and accuracy whereas the LSTM has the best F1-score and recall. However, it should be noted that scores are close. On Dataset 2, the path analogy model outperforms the LSTM by about 3 points in precision, 2 points in F1-score, and 4 points in accuracy, while obtaining a similar recall. Recall that Dataset 1 is based on a set of homogeneous seed entities from the Computer Science / Information Technology domain whereas Dataset 2 mixes heterogenous domains such as food, sport, and science. Dataset 1 may thus provide less diversity for models to correctly learn similarity and dissimilarity between reached entities and seed entities. Additionally, such an homogeneity may also entail a fuzzy keeping/pruning boundary. To illustrate, starting from a network protocol of a specific layer of the OSI model, a protocol from another OSI layer was manually labeled with a pruning decision. Such a very fine-grained decision may be difficult to capture by models. To better reflect the performance of each model, we also provide their number of trainable parameters in Table 5. It is then striking that the path analogy model obtains close or better performance than the LSTM with 150 to 800 times fewer parameters. This global evaluation, taking into account the performance measured with traditional metrics as well as the number of trainable parameters, shows the superiority of our proposed analogy-based model.

Table 4 shows the performance of the compared models on the transfer learning setting. Again, it appears that the LSTM and path-analogy models are the two best performing models. While the LSTM obtains a better precision, the path-analogy model outperforms on recall, F1-score, and accuracy by 4 to 9 points. To better assess the generalization capability of these two models, we provide in Table 6 the breakdown of results from Table 3 depending on whether the reached entities when testing were seen or unseen during training. As can be expected, both models perform less on unseen entities. We notice that their scores are similar on Dataset 1 whereas the path analogy model outperforms the LSTM on both unseen and seen entities in Dataset 2. Recall that Dataset 2 contains much more unseen entities in testing that Dataset 1 (Table 2). These results thus demonstrate the higher generalization capability of our proposed analogy-based model. We posit that the formalization of

		Dataset 1				Dataset 2			
		P	R	F1	ACC	P	R	F1	ACC
LSTM	Unseen entities	**73.90 ± 5.97**	**69.69 ± 7.90**	71.33 ± 4.73	78.06 ± 6.15	75.44 ± 8.99	94.21 ± 2.99	83.33 ± 4.72	76.35 ± 6.37
	Seen entities	84.63 ± 6.80	**80.95 ± 5.51**	**82.32 ± 1.29**	**87.55 ± 1.83**	94.62 ± 1.71	**95.51 ± 3.99**	95.02 ± 2.14	91.71 ± 3.39
Path analogy	Unseen entities	73.82 ± 3.62	69.44 ± 6.09	**71.36 ± 3.64**	**78.20 ± 4.78**	**78.84 ± 8.34**	**94.46 ± 2.83**	**85.67 ± 5.01**	**80.49 ± 5.97**
	Seen entities	**85.20 ± 2.10**	78.25 ± 5.20	81.47 ± 3.01	87.41 ± 2.05	**96.07 ± 2.47**	94.90 ± 5.45	**95.44 ± 3.82**	**93.15 ± 4.98**

Table 6: Performance of the LSTM and path analogy models depending on whether entities reached when testing where seen or unseen in training. P stands for average precision, R stands for average recall, F1 stands for average F1-score, and ACC stands for average accuracy. Best results for each category (*i.e.*, seen or unseen) are in bold.

analogical quadruples and the use of a CNN lead the model to learn to compute and compare *relative* similarities and dissimilarities between the two pairs in a quadruple. In turn, this leads to better extrapolation capabilities to decide on an unseen pair when compared to a seen pair within an analogical quadruple. Consequently, we think analogy-based model are well-suited for such zero-shot settings.

To extend our approach, we could envision to test our model in a few-shot setting by having some labeled decisions to train on for seed entities considered in testing. In a real-world use-case scenario, this would correspond to asking experts to label a few neighbors of each seed entity before performing the expansion along the ontology hierarchy. This could be of interest to further test the extrapolation capability of our analogy-based model. However, we believe that in a real-world scenario, experts would rather label as many neighbors as possible of some seed entities and expect the model to extrapolate on new seed entities, hence our focus on the zero-shot setting. As aforementioned, some pruning decisions are motivated by errors in the ontology hierarchy of Wikidata, which is known to contain to be potentially noisy [42][7]. Another extension of our approach thus consists in applying it to ontology maintenance.

Regarding our model, we leverage KG embeddings pre-trained with a translational model. However, there exist several types of KG embedding models, such as translational, complex, Gaussian or Graph Neural Network-based ones [20]. It would thus be interesting to evaluate which types of embedding models are better suited to serve an analogy-based model. Additionally, instead of using frozen KG embeddings previously learned on a specific task, we could envision learning simultaneously the graph embeddings and the CNN layers, similarly to what was done in [27] with character and word embeddings. Finally, it is noteworthy that our work does not rely on Large Language Models (LLMs) such as BERT [10]. This purposely allows us to assess if the structure of the graph provides enough information to learn useful embeddings for selecting relevant entities. Future research directions could involve enriching our approach with LLMs, while raising additional issues to face such as noisy labels or homonyms. To illustrate, the entity "role" can be a part played by a performer[8] or an identity of an item in relation to another specified item[9].

[7]See also https://commons.wikimedia.org/wiki/File:WikidataCon_2021_-_Overview_of_ontology_issues.pdf.
[8]https://www.wikidata.org/wiki/Q1707847
[9]https://www.wikidata.org/wiki/Q4897819

6 CONCLUSION

In this paper, we considered the task of bootstrapping a knowledge graph (KG) by selecting relevant entities in the neighborhood of seed entities of interest in a generic KG. We proposed an analogy-based model to keep or prune neighors of seed entities in a zero-shot setting and two labeled datasets to evaluate models on this task. Compared with standard classifiers, our model outperformed while presenting a drastically lower number of parameters. Additionally, it showed better extrapolation capabilities in zero-shot and transfer learning settings. Such results advocate for the further study of analogy-based models in tasks related to the KG lifecycle or requiring extrapolation capabilities, which we will address in future work.

ACKNOWLEDGMENTS

This work is supported by the AT2TA project (https://at2ta.loria.fr/) funded by the French National Research Agency ("Agence Nationale de la Recherche" – ANR) under grant ANR-22-CE23-0023.

REFERENCES

[1] Stergos D. Afantenos, Tarek Kunze, Suryani Lim, Henri Prade, and Gilles Richard. 2021. Analogies Between Sentences: Theoretical Aspects - Preliminary Experiments. In *Symbolic and Quantitative Approaches to Reasoning with Uncertainty - 16th European Conference, ECSQARU 2021, Prague, Czech Republic, September 21-24, 2021, Proceedings (Lecture Notes in Computer Science, Vol. 12897)*. Springer, 3–18. https://doi.org/10.1007/978-3-030-86772-0_1

[2] Stergos D. Afantenos, Suryani Lim, Henri Prade, and Gilles Richard. 2022. Theoretical Study and Empirical Investigation of Sentence Analogies. In *Proceedings of the Workshop on the Interactions between Analogical Reasoning and Machine Learning (International Joint Conference on Artificial Intelligence - European Conference on Artificial Intelligence (IJAI-ECAI 2022)), Vienna, Austria, July 23, 2022 (CEUR Workshop Proceedings, Vol. 3174)*. CEUR-WS.org, 15–28. https://ceur-ws.org/Vol-3174/paper2.pdf

[3] Safa Alsaidi, Amandine Decker, Puthineath Lay, Esteban Marquer, Pierre-Alexandre Murena, and Miguel Couceiro. 2021. A Neural Approach for Detecting Morphological Analogies. In *8th IEEE International Conference on Data Science and Advanced Analytics, DSAA 2021, Porto, Portugal, October 6-9, 2021*. IEEE, 1–10. https://doi.org/10.1109/DSAA53316.2021.9564186

[4] Gulkhara Babayeva, Kaie Maennel, and Olaf Manuel Maennel. 2022. Building an Ontology for Cyber Defence Exercises. In *IEEE European Symposium on Security and Privacy, EuroS&P 2022 - Workshops, Genoa, Italy, June 6-10, 2022*. IEEE, 423–432. https://doi.org/10.1109/EuroSPW55150.2022.00050

[5] Hongyun Cai, Vincent W. Zheng, and Kevin Chen-Chuan Chang. 2018. A Comprehensive Survey of Graph Embedding: Problems, Techniques, and Applications. *IEEE Transactions on Knowledge and Data Engineering* 30, 9 (2018), 1616–1637. https://doi.org/10.1109/TKDE.2018.2807452

[6] François Chollet. 2019. On the Measure of Intelligence. *CoRR* abs/1911.01547 (2019).

[7] Miguel Couceiro, Nicolas Hug, Henri Prade, and Gilles Richard. 2017. Analogy-preserving functions: A way to extend Boolean samples. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*. ijcai.org, 1575–1581. https://doi.org/10.24963/ijcai.2017/218

[8] Miguel Couceiro, Nicolas Hug, Henri Prade, and Gilles Richard. 2018. Behavior of Analogical Inference w.r.t. Boolean Functions. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*. ijcai.org, 2057–2063. https://doi.org/10.24963/ijcai.2018/284

[9] Miguel Couceiro and Erkko Lehtonen. 2023. Galois theory for analogical classifiers. *AMAI* (2023). https://doi.org/10.1007/s10472-023-09833-6

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, 4171–4186. https://doi.org/10.18653/v1/n19-1423

[11] Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 2014. Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*. ACM, 601–610. https://doi.org/10.1145/2623330.2623623

[12] Mohsen Ahmadi Fahandar and Eyke Hüllermeier. 2018. Learning to Rank Based on Analogical Reasoning. 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), New Orleans, Louisiana, USA, February 2-7, 2018*. AAAI Press, 2951–2958.

[13] Mohsen Ahmadi Fahandar and Eyke Hüllermeier. 2021. Analogical Embedding for Analogy-Based Learning to Rank. In *Advances in Intelligent Data Analysis XIX - 19th International Symposium on Intelligent Data Analysis, IDA 2021, Porto, Portugal, April 26-28, 2021, Proceedings (Lecture Notes in Computer Science, Vol. 12695)*. Springer, 76–88. https://doi.org/10.1007/978-3-030-74251-5_7

[14] Stefano Faralli, Irene Finocchi, Simone Paolo Ponzetto, and Paola Velardi. 2018. Efficient Pruning of Large Knowledge Graphs. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*. ijcai.org, 4055–4063.

[15] Mariano Fernández-López, Asuncion Gomez-Perez, and Natalia Juristo. 1997. METHONTOLOGY: from ontological art towards ontological engineering. *Engineering Workshop on Ontological Engineering (AAAI97)* (03 1997).

[16] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann. 2021. *Knowledge Graphs*. Morgan & Claypool Publishers. https://doi.org/10.2200/S01125ED1V01Y202109DSK022

[17] Nicolas Hug, Henri Prade, Gilles Richard, and Mathieu Serrurier. 2019. Analogical proportion-based methods for recommendation - First investigations. *Fuzzy Sets Systems* 366 (2019), 110–132. https://doi.org/10.1016/j.fss.2018.11.007

[18] Filip Ilievski, Jay Pujara, and Kartik Shenoy. 2022. Does Wikidata Support Analogical Reasoning?. In *Knowledge Graphs and Semantic Web - 4th Iberoamerican Conference and third Indo-American Conference, KGSWC 2022, Madrid, Spain, November 21-23, 2022, Proceedings (Communications in Computer and Information Science, Vol. 1686)*. Springer, 178–191. https://doi.org/10.1007/978-3-031-21422-6_13

[19] Lucas Jarnac and Pierre Monnin. 2022. Wikidata to Bootstrap an Enterprise Knowledge Graph: How to Stay on Topic?. In *Proceedings of the 3rd Wikidata Workshop 2022 co-located with the 21st International Semantic Web Conference (ISWC2022), Virtual Event, Hanghzou, China, October 2022 (CEUR Workshop Proceedings, Vol. 3262)*. CEUR-WS.org. https://ceur-ws.org/Vol-3262/paper16.pdf

[20] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. 2022. A Survey on Knowledge Graphs: Representation, Acquisition, and Applications. *IEEE Transactions on Neural Networks and Learning Systems* 33, 2 (2022), 494–514. https://doi.org/10.1109/TNNLS.2021.3070843

[21] Philippe Langlais, François Yvon, and Pierre Zweigenbaum. 2009. Improvements in Analogical Learning: Application to Translating Multi-Terms of the Medical Domain. In *EACL 2009, 12th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, Athens, Greece, March 30 - April 3, 2009*. The Association for Computer Linguistics, 487–495. https://aclanthology.org/E09-1056/

[22] Adam Lerer, Ledell Wu, Jiajun Shen, Timothée Lacroix, Luca Wehrstedt, Abhijit Bose, and Alex Peysakhovich. 2019. Pytorch-BigGraph: A Large Scale Graph Embedding System. In *Proceedings of Machine Learning and Systems 2019, MLSys 2019, Stanford, CA, USA, March 31 - April 2, 2019*. mlsys.org.

[23] Suryani Lim, Henri Prade, and Gilles Richard. 2019. Solving Word Analogies: A Machine Learning Perspective. In *Symbolic and Quantitative Approaches to Reasoning with Uncertainty, 15th European Conference, ECSQARU 2019, Belgrade, Serbia, September 18-20, 2019, Proceedings (Lecture Notes in Computer Science, Vol. 11726)*. Springer, 238–250. https://doi.org/10.1007/978-3-030-29765-7_20

[24] Suryani Lim, Henri Prade, and Gilles Richard. 2021. Classifying and completing word analogies by machine learning. *International Journal of Approximate Reasoning* 132 (2021), 1–25. https://doi.org/10.1016/j.ijar.2021.02.002

[25] Hanxiao Liu, Yuexin Wu, and Yiming Yang. 2017. Analogical Inference for Multi-relational Embeddings. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017 (Proceedings of Machine Learning Research, Vol. 70)*. PMLR, 2168–2178. http://proceedings.mlr.press/v70/liu17d.html

[26] Farzaneh Mahdisoltani, Joanna Biega, and Fabian M. Suchanek. 2015. YAGO3: A Knowledge Base from Multilingual Wikipedias. In *Seventh Biennial Conference on Innovative Data Systems Research, CIDR 2015, Asilomar, CA, USA, January 4-7, 2015, Online Proceedings*. www.cidrdb.org. http://cidrdb.org/cidr2015/Papers/CIDR15_Paper1.pdf

[27] Esteban Marquer, Safa Alsaidi, Amandine Decker, Pierre-Alexandre Murena, and Miguel Couceiro. 2022. A Deep Learning Approach to Solving Morphological Analogies. In *Case-Based Reasoning Research and Development - 30th International Conference, ICCBR 2022, Nancy, France, September 12-15, 2022, Proceedings (Lecture Notes in Computer Science, Vol. 13405)*. Springer, 159–174. https://doi.org/10.1007/978-3-031-14923-8_11

[28] Laurent Miclet, Sabri Bayoudh, and Arnaud Delhay. 2008. Analogical Dissimilarity: Definition, Algorithms and Two Experiments in Machine Learning. *Journal of Artificial Intelligence Research* 32 (2008), 793–824. https://doi.org/10.1613/jair.2519

[29] Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*. http://arxiv.org/abs/1301.3781

[30] Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. 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*. 3111–3119. https://proceedings.neurips.cc/paper/2013/hash/9aa42b31882ec039965f3c4923ce901b-Abstract.html

[31] Melanie Mitchell. 2021. Abstraction and analogy-making in artificial intelligence. *Annals of the New York Academy of Sciences* 1505, 1 (2021), 79–101.

[32] Pierre Monnin and Miguel Couceiro. 2022. Interactions Between Knowledge Graph-Related Tasks and Analogical Reasoning: A Discussion. In *Workshop Proceedings of the 30th International Conferece on Case-Based Reasoning co-located with the 30th International Conference on Case-Based Reasoning (ICCBR 2022), Nancy (France), September 12-15th, 2022 (CEUR Workshop Proceedings, Vol. 3389)*. CEUR-WS.org, 57–67. https://ceur-ws.org/Vol-3389/ICCBR_2022_Workshop_paper_75.pdf

[33] Pierre Monnin, Joël Legrand, Graziella Husson, Patrice Ringot, Andon Tchechmedjiev, Clément Jonquet, Amedeo Napoli, and Adrien Coulet. 2019. PGxO and PGxLOD: a reconciliation of pharmacogenomic knowledge of various provenances, enabling further comparison. *BMC Bioinformatics* 20-S, 4 (2019), 139:1–139:16. https://doi.org/10.1186/s12859-019-2693-9

[34] Pierre-Alexandre Murena, Marie Al-Ghossein, Jean-Louis Dessalles, and Antoine Cornuéjols. 2020. Solving Analogies on Words based on Minimal Complexity Transformation. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*. ijcai.org, 1848–1854. https://doi.org/10.24963/ijcai.2020/256

[35] Natalya Fridman Noy, Yuqing Gao, Anshu Jain, Anant Narayanan, Alan Patterson, and Jamie Taylor. 2019. Industry-scale knowledge graphs: lessons and challenges. *Commun. ACM* 62, 8 (2019), 36–43. https://doi.org/10.1145/3331166

[36] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake VanderPlas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[37] Julia Peyre, Josef Sivic, Ivan Laptev, and Cordelia Schmid. 2019. Detecting Unseen Visual Relations Using Analogies. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, 1981–1990. https://doi.org/10.1109/ICCV.2019.00207

[38] Jan Portisch, Nicolas Heist, and Heiko Paulheim. 2022. Knowledge graph embedding for data mining vs. knowledge graph embedding for link prediction - two sides of the same coin? *Semantic Web* 13, 3 (2022), 399–422. https://doi.org/10.3233/SW-212892

[39] Fereshteh Sadeghi, C. Lawrence Zitnick, and Ali Farhadi. 2015. Visalogy: Answering Visual Analogy Questions. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*. 1882–1890.

[40] Juan Sequeda and Ora Lassila. 2021. *Designing and Building Enterprise Knowledge Graphs*. Morgan & Claypool Publishers. https://doi.org/10.2200/S01105ED1V01Y202105DSK020

[41] Basel Shbita, Anna Lisa Gentile, Pengyuan Li, Chad DeLuca, and Guang-Jie Ren. 2023. Understanding Customer Requirements - An Enterprise Knowledge Graph Approach. In *The Semantic Web - 20th International Conference, ESWC*

2023, Hersonissos, Crete, Greece, May 28 - June 1, 2023, Proceedings (Lecture Notes in Computer Science, Vol. 13870). Springer, 625–643. https://doi.org/10.1007/978-3-031-33455-9_37

[42] Kartik Shenoy, Filip Ilievski, Daniel Garijo, Daniel Schwabe, and Pedro A. Szekely. 2022. A study of the quality of Wikidata. *Journal of Web Semantics* 72 (2022), 100679.

[43] Oren Sultan and Dafna Shahaf. 2022. Life is a Circus and We are the Clowns: Automatically Finding Analogies between Situations and Processes. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*. Association for Computational Linguistics, 3547–3562. https://aclanthology.org/2022.emnlp-main.232

[44] Bill Swartout, Ramesh Patil, Kevin Knight, and Tom Russ. 1996. Toward distributed use of large-scale ontologies. In *Proceedings of the Tenth Workshop on Knowledge Acquisition for Knowledge-Based Systems*, Vol. 138. 25.

[45] Ilaria Tiddi and Stefan Schlobach. 2022. Knowledge graphs as tools for explainable machine learning: A survey. *Artificial Intelligence* 302 (2022), 103627. https://doi.org/10.1016/j.artint.2021.103627

[46] Peter D. Turney. 2008. The Latent Relation Mapping Engine: Algorithm and Experiments. 33 (2008), 615–655. https://doi.org/10.1613/jair.2693

[47] Denny Vrandecic and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM* 57, 10 (2014), 78–85.

[48] Gerhard Weikum, Xin Luna Dong, Simon Razniewski, and Fabian M. Suchanek. 2021. Machine Knowledge: Creation and Curation of Comprehensive Knowledge Bases. *Foundations and Trends Databases* 10, 2-4 (2021), 108–490.

[49] Zhen Yao, Wen Zhang, Mingyang Chen, Yufeng Huang, Yi Yang, and Huajun Chen. 2023. Analogical Inference Enhanced Knowledge Graph Embedding. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*. AAAI Press, 4801–4808. https://ojs.aaai.org/index.php/AAAI/article/view/25605

[50] Georgios Zervakis, Emmanuel Vincent, Miguel Couceiro, Marc Schoenauer, and Esteban Marquer. 2022. An Analogy based Approach for Solving Target Sense Verification. In *Proceedings of the 2022 6th International Conference on Natural Language Processing and Information Retrieval, NLPIR 2022, Bangkok, Thailand, December 16-18, 2022*. ACM, 144–151. https://doi.org/10.1145/3582768.3582794

[51] Xunjie Zhu and Gerard de Melo. 2020. Sentence Analogies: Linguistic Regularities in Sentence Embeddings. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*. International Committee on Computational Linguistics, 3389–3400. https://doi.org/10.18653/v1/2020.coling-main.300