



HAL
open science

Link Prediction in Knowledge Graphs with Concepts of Nearest Neighbours

Sébastien Ferré

► **To cite this version:**

Sébastien Ferré. Link Prediction in Knowledge Graphs with Concepts of Nearest Neighbours. The Semantic Web (ESWC), Jun 2019, Portoroz, Slovenia. pp.84-100. hal-02281789

HAL Id: hal-02281789

<https://inria.hal.science/hal-02281789v1>

Submitted on 9 Sep 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Link Prediction in Knowledge Graphs with Concepts of Nearest Neighbours

Sébastien Ferré*

Univ Rennes, CNRS, IRISA
Campus de Beaulieu, 35042 Rennes, France
Email: ferre@irisa.fr

Abstract. The open nature of Knowledge Graphs (KG) often implies that they are incomplete. Link prediction consists in inferring new links between the entities of a KG based on existing links. Most existing approaches rely on the learning of latent feature vectors for the encoding of entities and relations. In general however, latent features cannot be easily interpreted. Rule-based approaches offer interpretability but a distinct ruleset must be learned for each relation, and computation time is difficult to control. We propose a new approach that does not need a training phase, and that can provide interpretable explanations for each inference. It relies on the computation of Concepts of Nearest Neighbours (CNN) to identify similar entities based on common graph patterns. Dempster-Shafer theory is then used to draw inferences from CNNs. We evaluate our approach on FB15k-237, a challenging benchmark for link prediction, where it gets competitive performance compared to existing approaches.

1 Introduction

There is a growing interest for knowledge graphs (KG) as a way to represent and share data on the Web. The Semantic Web [1] defines standards for representation (RDF), querying (SPARQL), and reasoning (RDFS, OWL), and thousands of open KGs are available: e.g., DBpedia, Wikidata (formerly Freebase), YAGO, WordNet. The open nature of KGs often implies that they are incomplete, and a lot of work have studied the use of machine learning techniques to complete them.

The task of *link prediction* [16] consists in predicting missing edges or missing parts of edges. Suppose that film *Avatar* is missing a director in the KG, one wants to predict it, i.e. identify it among all KG nodes. The idea is to find regularities in the existing knowledge, and to exploit them in order to rank the KG nodes. The higher the correct node is in the ranking, the better the prediction is. Link prediction was originally introduced for social networks with a single edge type (a single *relation*) [12], and was later extended to multi-relational data and applied to KGs [16]. Compared to supervised classification, link prediction faces several challenges. First, there are as many classification problems as there are relations, which count in the hundreds or thousands in KGs. Second, for each relation, the

* This research is supported by ANR project PEGASE (ANR-16-CE23-0011-08).

number of “classes” is the number of different *entities* in the range of the relation, which typically counts in the thousands for relations like *spouse* or *birthPlace*. Third, some relations can be multi-valued, like the relation from films to actors.

In this paper, we report on first experimental results on a novel approach to link prediction based on *Concepts of Nearest Neighbours (CNN)*, which were introduced in [5], and applied to *query relaxation* in [6]. This approach is a symbolic form of the k-nearest neighbours where numerical distances are replaced by graph patterns that provide an intelligible representation of how similar two nodes are. Our hypothesis is that the partitioning of the KG nodes into CNNs (see Section 4) provides a valuable basis for different kinds of inference. We here focus on link prediction, i.e. the inference of the missing node of an incomplete edge. The contribution of this work is a novel approach to link prediction that has the following properties:

1. it is a form of *instance-based learning*, i.e. it has no training phase;
2. it is a *symbolic approach*, i.e. it can provide explanations for each inference;
3. it shows *competitive performance* on a challenging link prediction benchmark.

The rest of the paper is organized as follows. Section 2 discusses related work on link prediction. Section 3 contains preliminaries about knowledge graphs and queries. Section 4 recalls the definition of CNNs, and their efficient computation. Section 5 presents our method to perform link prediction, using CNNs and Dempster-Shafer theory. Section 6 reports positive experimental results on benchmark FB15k-237 and two other datasets. Finally, Section 7 concludes and sketches future work.

2 Related Work

Nickel *et al* [16] have recently written a “review of relational machine learning for knowledge graphs”, where link prediction is the main inference task. They identify two kinds of approaches that differ by the kind of model they use: *latent feature models*, and *graph feature models*. The former is by far the most studied one. Before going into the details, it is useful to set the vocabulary as it is used in the domain. Nodes are called *entities*, edge labels are called *relations*, and edges are *triples* (e_i, r_k, e_j) , where e_i is the *head* entity, e_j is the *tail* entity, and r_k is the relation that links the head to the tail.

Latent feature models learn *embeddings* of entities and relations into low-dimensional vector spaces, and then make inferences about a triple (e_i, r_k, e_j) by combining the embeddings of the two entities and the embedding of the relation. The existing methods vary by how they learn the embeddings, and how they combine them. Those methods are based on a range of techniques including: matrix factorization, tensor factorization, neural networks, and gradient descent. For example, one of the first method for KGs, TransE [2], models a relation as a translation in the embedding space of entities, and scores a candidate triple according to the distance between the translated head and the tail. Bordes *et al* also introduced two datasets, FB15k and WN18, respectively derived from Freebase and Wordnet, which became references in the evaluation of link prediction methods. Toutanova and Chen [19] however showed that a very simple method was able to outperform previous methods because of a flaw in the datasets: many test triples have their inverse among the training triples. They

introduced a challenging subset of FB15k, called FB15k-237, where all inverse triples are removed. Lately, performance was significantly improved on FB15k-237 by using convolutional architectures to learn embeddings [18] or to combine them in scoring functions [4]. The task of link prediction has also been extended with the embedding model RAE to *multi-fold relations* (aka. n-ary relations) and to *instance reconstruction* where only one entity of an n-ary relation is known, and all other entities have to be inferred together [20]. In this work, we limit ourselves to binary relations.

Graph feature models, also called *observed feature models*, make inferences directly from the observed edges in the KG. Random walk inference [11] takes relation paths as features, and sets the feature values through random walks in the KG. The feature weights are learned by logistic regression for each target relation, and then used to score the candidate triples. The method has shown improvement over Horn clause generation with ILP (Inductive Logic Programming) [15]. AMIE+ [8] manages to generate such Horn clauses in a much more efficient way by designing new ILP algorithms tailored to KGs. They also introduce a novel rule measure that improves the inference precision under the Open World Assumption (OWA) that holds in KGs. Both methods offer the advantage to produce intelligible explanations for inferences, unlike the latent feature models. However, they require a distinct training phase for each of the hundreds to thousands of target relations, whereas the latent feature models are generally learned in one phase. A fine-grained evaluation [14] has shown that rule-based approaches are competitive with latent-based approaches, both in performance and in running time.

A key difference of our approach is that there is no training phase, and all the learning is done at inference time. It is therefore an instance-based approach rather than a model-based approach. Given an incomplete triple $(e_i, r_k, ?)$ we compute Concepts of Nearest Neighbours (CNN) from the observed features of head entity e_i , where CNNs have a representation equivalent to the bodies of AMIE+'s rules. From there, we infer a ranking of candidate entities for the tail of relation r_k . In fact, as r_k is not involved in the computation of CNNs, many target relations can be inferred at nearly the same cost as a single relation. Indeed the main cost is in the computation of CNNs, which is easily controlled because the computation algorithm is any-time.

3 Preliminaries

A *knowledge graph* (KG) is defined by a structure $K = \langle E, R, T \rangle$, where E is the set of nodes, also called *entities*, R is the set of edge labels, also called *relations*, and $T \subseteq E \times R \times E$ is the set of directed and labelled edges, also called *triples*. Each triple (e_i, r_k, e_j) represents the fact that relation r_k relates entity e_i to entity e_j . This definition is very close to RDF graphs, where entities can be either URIs or literals (or blank nodes) and relations are URIs called properties. It is also equivalent to sets of logical facts, where entities are constants, and relations are binary predicates. As a running example, Figure 1 defines a small KG describing (part of) the British royal family (where the notation $(\{a,b\}, r, \{c,d\})$ is an abbreviation for $(a,r,c), (a,r,d), (b,r,c), (b,r,d)$).

Queries based on *graph patterns* play a central role in our approach as they are used to characterize the CNNs, and can be used as explanations for inferences. There are two kinds of *query elements*: triple patterns and filters. A *triple pattern* $(x,r,y) \in V \times R \times V$

$$\begin{aligned}
E &= \{Charles, Diana, William, Harry, Kate, George, Charlotte, Louis, male, female\} \\
R &= \{parent, spouse, sex\} \\
T &= \{(\{William, Harry\}, parent, \{Charles, Diana\}), \\
&\quad (\{George, Charlotte, Louis\}, parent, \{William, Kate\}), \\
&\quad (Charles, spouse, Diana), (Diana, spouse, Charles), \\
&\quad (William, spouse, Kate), (Kate, spouse, William), \\
&\quad (\{Charles, William, Harry, George, Louis\}, sex, male), \\
&\quad (\{Diana, Kate, Charlotte\}, sex, female)\}
\end{aligned}$$

Fig. 1. Example knowledge graph describing part of the British royal family.

is similar to a triple but with variables (taken from V) in place of entities. A *filter* is a Boolean expression on variables and entities. We here only consider equalities between a variable and an entity: $x = e$. A *graph pattern* P is a set of query elements. Equality filters are equivalent to allowing entities in triple patterns. There are two advantages in their use: (1) simplifying the handling of triple patterns that have a single form (var-var) instead of four (var-var, entity-var, var-entity, entity-entity); (2) opening perspectives for richer filters (e.g., intervals of values: $x \in [a, b]$). A *query* $Q = (x_1, \dots, x_n) \leftarrow P$ is the projection of a graph pattern on a subset of its variables. Such queries find a concrete form in SPARQL with syntax `SELECT ?x1...?xn WHERE { graph pattern }`. Queries can be seen as anonymous rules, i.e. rules like those in AMIE+ [8] but missing the relation in the head. For example, the query $Q_{ex} = (x, y) \leftarrow (x, parent, u), (u, parent, v), (y, parent, v), (y, sex, s), s = male$ retrieves all (person, uncle) pairs, i.e. all pairs (x, y) where y is a sibling of a parent of x , and is male.

We now define the *answer set* that is retrieved by a query. A *matching* of a pattern P on a KG $K = \langle E, R, T \rangle$ is a mapping μ from variables in P to entities in E such that $\mu(t) \in T$ for every triple pattern $t \in P$, and $\mu(f)$ evaluates to true for every filter $f \in P$, where $\mu(t)$ and $\mu(f)$ are obtained from t and f by replacing every variable x by $\mu(x)$. In the example KG, a possible matching for the pattern of the above query is $\mu_{ex} = \{x \mapsto Charlotte, y \mapsto Harry, u \mapsto William, v \mapsto Diana, s \mapsto male\}$. A matching is therefore a homomorphism from the pattern to the graph. Term “matching” is taken from the evaluation of SPARQL queries. In logics, terms “grounding” and “instantiation” are used instead. The *answer set* $ans(Q, K)$ of a query $Q = (x_1, \dots, x_n) \leftarrow P$ is the set of tuples $(\mu(x_1), \dots, \mu(x_n))$ for every matching μ of P on K . In the running example, the pair $(Charlotte, Harry)$ is therefore an answer of query Q_{ex} . Note that several matchings can lead to a same answer, and that duplicate answers are ignored. In the following, we only consider queries with a single projected variable, whose sets of answers are assimilated to sets of entities.

4 Concepts of Nearest Neighbours (CNN)

In this section, we shortly recall the theoretical definitions underlying Concepts of Nearest Neighbours (CNN), as well as the algorithmic and practical aspects of

computing their approximation under a given timeout. Further details are available in [5,6]. In the following definitions, we assume a knowledge graph $K = \langle E, R, T \rangle$.

4.1 Theoretical Definitions

Definition 1. A graph concept is defined as a pair $C = (A, Q)$, where A is a set of entities and Q is a query such that $A = \text{ans}(Q)$ is the set of answers of Q , and $Q = \text{msq}(A)$ is the most specific query that verifies $A = \text{ans}(Q)$. A is called the extension $\text{ext}(C)$ of the concept, and Q is called the intension $\text{int}(C)$ of the concept.

That most specific query $Q = \text{msq}(A)$ represents what the neighborhood of entities in A have in common. It is well-defined under graph homomorphism (unlike under subgraph isomorphism). It can be computed from A by using the categorical product of graphs (see PGP intersection \cap_q in [7]), or equivalently Plotkin’s anti-unification of sets of facts [17]. In the example KG, William and Charlotte have in common the following query that says that they have married parents: $Q_{WC} = x \leftarrow (x, \text{sex}, s), (x, \text{parent}, y), (y, \text{sex}, t), t = \text{male}, (x, \text{parent}, z), (z, \text{sex}, u), u = \text{female}, (y, \text{spouse}, z), (z, \text{spouse}, y)$. We have $A_{WC} = \text{ans}(Q_{WC}) = \{\text{William}, \text{Harry}, \text{George}, \text{Charlotte}, \text{Louis}\}$ so that $C_{WC} = (A_{WC}, Q_{WC})$ is a graph concept. A concept $C_1 = (A_1, Q_1)$ is more specific than a concept $C_2 = (A_2, Q_2)$, in notation $C_1 \leq C_2$, if $A_1 \subseteq A_2$. For example, by adding filter $y = \text{William}$ to the previous example, we get a more specific concept whose extension is $\{\text{George}, \text{Charlotte}, \text{Louis}\}$.

Definition 2. Let $e_1, e_2 \in E$ be two entities. The conceptual distance $\delta(e_1, e_2)$ between e_1 and e_2 is the most specific graph concept whose extension contains both entities, i.e. $\delta(e_1, e_2) = (A, Q)$ with $Q = \text{msq}(\{e_1, e_2\})$, $A = \text{ans}(Q)$.

For example, the above concept C_{WC} is the conceptual distance between William and Charlotte. The “distance values” have therefore a symbolic representation through the concept intension Q that represents what the two entities have in common. The concept extension A contains in addition to the two entities all entities e_3 that match the common query ($e_3 \in \text{ans}(Q)$). Such an entity e_3 can be seen as “between” e_1 and e_2 : in formulas, for all $e_3 \in \text{ext}(\delta(e_1, e_2))$, $\delta(e_1, e_3) \leq \delta(e_1, e_2)$ and $\delta(e_3, e_2) \leq \delta(e_1, e_2)$. Note that order \leq on conceptual distances is a partial ordering, unlike classical distance measures. A numerical distance $\text{dist}(e_1, e_2) = |\text{ext}(\delta(e_1, e_2))|$ can be derived from the size of the concept extension, because the closer e_1 and e_2 are, the more specific their conceptual distance and the smaller the extension. Dually, a numerical similarity $\text{sim}(e_1, e_2) = |\text{int}(\delta(e_1, e_2))|$ can be derived from the size of the concept intension (number of query elements), because the more similar e_1 and e_2 are, the more specific their conceptual distance and the larger the intension. For example, between William and Charlotte, the numerical distance is 5 and the numerical similarity is 9.

Definition 3. Let $e \in E$ be an entity. A Concept of Nearest Neighbours (CNN) of e is a pair (S_l, δ_l) where S_l is the non-empty set of entities that are at the same conceptual distance δ_l from e . Therefore, a CNN verifies $S_l = \{e' \in E \mid \delta(e, e') = \delta_l\} \neq \emptyset$. It also verifies $S_l \subseteq \text{ext}(\delta_l)$, and S_l is called the proper extension of the CNN. We note $\text{CNN}(e, K)$ the collection of all CNNs of e in knowledge graph K .

Here are the 6 CNNs of Charlotte in the running example.

l	S_l	$ ext(\delta_l) $	$int(\delta_l)$	$\{l' \mid \delta_{l'} \preceq \delta_l\}$
1	{Charlotte}	1	$x \leftarrow x = Charlotte$	-
2	{Diana, Kate}	3	$x \leftarrow (x, sex, s), s = female$	1
3	{George, Louis}	3	$x \leftarrow (x, sex, s), (x, parent, y), y = William, \dots$	1
4	{William, Harry}	5	$x \leftarrow (x, sex, s), (x, parent, y), \dots$	1, 3
5	{Charles}	8	$x \leftarrow (x, sex, s)$	1, 2, 3, 4
6	{male, female}	10	$x \leftarrow \emptyset$	1, 2, 3, 4, 5

The proper extensions of $CNN(e, K)$ define a partition over the set of entities E , where two entities are in the same cluster S_l if they are at the same conceptual distance to entity e . The intension of the associated graph concept δ_l provides a symbolic representation of the distance/similarity between every $e' \in S_l$ and e . The partial ordering over CNNs means that some CNNs are closer to e than others. As such, each CNN can be seen as a cluster of nearest neighbours of e , where the size of the extension of δ_l can be used as a numerical distance.

Discussion. Given that $CNN(e, K)$ partitions the set of entities, the number of CNNs can only be smaller or equal to the number of entities, and in practice it is generally much smaller. This is interesting because, in comparison, the number of graph concepts is exponential in the number of entities in the worst case. Note that the search space of ILP approaches like AMIE+ is the set of queries, which is even larger than the set of all graph concepts. Computing the CNNs for a given entity is therefore a much more tractable task than mining frequent patterns or learning rules, although the space of representations is the same. The pending questions that we start studying in this paper is whether those CNNs are useful for inference, and how they compare to other approaches.

Compared to the use of numerical measures, like commonly done in k-nearest neighbours approaches, CNNs define a more subtle ordering of entities. First, because conceptual distances are only partially ordered, it can be that among two entities none is more similar than the other to the chosen entity e . This reflects the fact that there can be several ways to be similar to something, without necessarily a preferred one. For instance, which is most similar to Charlotte? Diana because she is also a female (CNN S_2) or George because he is also a son of William (CNN S_3)? Second, because conceptual distances partition the set of entities, it can be that two entities are at the exact same distance, and are therefore undistinguishable in terms of similarity (ex. George and Louis in CNN S_3). Third, the concept intension provides an intelligible explanation of the similarity to the chosen entity.

4.2 Algorithmic and Practical Aspects

We here sketch the algorithmic and practical aspects of computing the set $CNN(e, K)$ of concepts of nearest neighbours of query entity e in a knowledge graph K . More details are available in [6]. The core principle of the algorithm is to iteratively refine a partition $\{S_l\}_l$ of the set of entities, in order to get an increasingly accurate partition converging to the partition induced by the proper extensions of CNNs.

Each cluster S_l is associated to a query $Q_l = x \leftarrow P_l$, and a set of candidate query elements H_l . The relationship to CNNs is that when H_l is empty, (S_l, δ_l) with $\delta_l = (ans(Q_l), Q_l)$ is a CNN, i.e. S_l is the proper extension of a CNN whose conceptual distance has intension Q_l . When H_l is not empty, S_l may be an union of several proper extensions (lack of discrimination), and Q_l is not necessarily the most specific query that matches all entities in S_l (lack of precision in the conceptual similarity). In that case, one gets overestimates of conceptual distances for some entities in S_l .

Initially, there is a single cluster $S_0 = E$ with P_0 being the empty pattern, and H_0 being the *description* of e . The description of an entity e is a graph pattern that is obtained by extracting a subgraph around e and, for each entity e_i in the subgraph, by replacing e_i by a variable y_i , and by adding filter $y_i = e_i$. Here, we choose to extract the subgraph that contains all edges starting from e up to some depth.

Then at each iteration, any cluster S – with pattern P and set of candidate query elements H – is split in two clusters S_1, S_0 by using an element $h \in H$ as a discriminating feature. Element h must be chosen so that $P \cup \{h\}$ defines a connected pattern including variable x . In this work, this element is chosen so as to have a trade-off between depth-first and breadth-first exploration of the description of e but many other strategies are possible. The new clusters are defined as follows:

$$\begin{array}{lll} P_1 = P \cup \{h\} & S_1 = S \cap ans(Q_1 = x \leftarrow P_1, K) & H_1 = H \setminus \{h\} \\ P_0 = P & S_0 = S \setminus S_1 & H_0 = H \setminus \{h\} \end{array}$$

The equations for S_1, S_0 ensure that after each split there is still a partition, possibly a more accurate one. The empty clusters ($S_i = \emptyset$) are removed from the partition. As a consequence, although the search space is the set of subgraphs of the description of e , which has a size exponential in the size of the description, the number of clusters remains below the number of entities at all time. In the running example, the initial cluster S_{1-6} (the union of clusters S_1 to S_6) is split with element $x = Charlotte$ into S_1 and S_{2-6} . Then cluster S_{2-6} is split with element (x, sex, s) into S_{2-5} and S_6 . Then cluster S_{2-5} is split with element $s = female$ into S_2 and S_{3-5} . Next splits involve elements $(x, parent, y)$ and $y = William$ on S_{3-5} .

Discussion. The above algorithm terminates because the set H decreases at each split. However, in the case of large descriptions or large knowledge graphs, it can still take a long time. Runtime is easily controlled with a timeout because the algorithm is anytime. Indeed it can output a partition of entities at any time, along with an overestimate of conceptual distance for each cluster. Previous experiments indicate that the algorithm has the good property to output more than half of the concepts in a small proportion of the total runtime.

Actually, the above algorithm converges to an approximation of the CNNs, in the sense that the conceptual distance may be still be an overestimate at full runtime for some entities. This is because graph patterns are constrained to be subsets of the description of e . In order to get exact results, the duplication of variables and their adjacent edges should be allowed, which would considerably increase the search space.

Experiments on KGs with up to a million triples have shown that the algorithm can compute all CNNs for descriptions of hundreds of edges in a matter of seconds or minutes. In contrast, query relaxation does not scale beyond 3 relaxation steps,

which is insufficient to identify similar entities in most cases; and computing pairwise symbolic similarities does not scale to large numbers of entities. A key ingredient of the efficiency of the algorithm also lies in a notion of *lazy join* for the computation of answer sets of queries. In short, the principle is to avoid the enumeration of all matchings of a query pattern by computing joins only as much as necessary to compute the set of query answers (see details in [6]).

5 Link Prediction

The problem of *link prediction* is to infer a missing entity in a triple (e_i, r_k, e_j) , i.e. either infer the tail from the head and the relation, or infer the head from the tail and the relation. Because of the symmetry of the two problems, we only describe here the inference of the tail entity. In the following, we therefore consider e_i and r_k as fixed (we avoid them in indices), and e_j as variable. Our approach to link prediction is inspired by the work of Denœux [3], and adapted to our Concepts of Nearest Neighbours (CNNs). Denœux defines a k -NN classification rule based on Dempster-Shafer (D-S) theory. Each k -nearest neighbour x_l of an instance to be classified x is used as a piece of evidence that supports the fact that x belongs to the class c_l of x_l . The degree of support is defined as a function of the distance between x and x_l , in such a way that the choice of k is less sensitive so that large values of k can be chosen. D-S theory enables to combine the k pieces of evidence into a global evidence, and to define a measure of *belief* for each class.

We adapt Denœux’s work to the inference of e_j in triple (e_i, r_k, e_j) in the following way. Given a computed partition of entities $\{(S_l, Q_l) \text{ with } Q_l = x \leftarrow P_l\}_l$, as an approximation of $CNN(e_i, K)$, each cluster (S_l, Q_l) is used as a piece of evidence for the inference of the tail entity e_j relative to relation r_k . The degree of support depends on the *extensional distance* d_l between e_i and entities in S_l ,

$$d_l = |\text{ans}(Q_l)|,$$

i.e. the number of answers of query Q_l , and on the *confidence* $\phi_{l,j}$ of the association rule $(x, r_k, e_j) \leftarrow P_l$, which is defined as

$$\phi_{l,j} = \frac{|\text{ans}(x \leftarrow P_l \cup \{(x, r_k, e_j)\})|}{|\text{ans}(Q_l)|},$$

i.e. the proportion of entities among the answers of Q_l that have an r_k -link to entity e_j . Because in KGs a head entity can be linked to several tail entities through the same relation, we consider a distinct classification problem for each candidate tail entity $e_j \in E$ with two classes c_j^1 (e_j is a tail entity) and c_j^0 (e_j is not a tail entity). For each cluster S_l and each candidate tail entity $e_j \in E$, the degree of support can therefore be formalized by defining a mass distribution $m_{l,j}$ over sets of classes as follows.

$$m_{l,j}(\{c_j^1\}) = \alpha_0 \phi_{l,j} e^{-d_l} \quad m_{l,j}(\{c_j^0\}) = 0 \quad m_{l,j}(\{c_j^1, c_j^0\}) = 1 - \alpha_0 \phi_{l,j} e^{-d_l}$$

$m_{l,j}(\{c_j^1\})$ represents the degree of belief for e_j being the tail entity, while $m_{l,j}(\{c_j^1, c_j^0\})$ represents the degree of uncertainty. $m_{l,j}(\{c_j^0\})$ is set to 0 to reflect the OWA (Open

World Assumption) of KGs according to which a missing fact is not considered as false. Constant α_0 determines the maximum degree of belief, which can be lower than 1 to reflect uncertainty about known triples (e.g. 0.95). The degree of belief decreases exponentially with distance. Finally, we make the degree of belief proportional to the confidence of inferring entity e_j from Q_l . In [3], that confidence factor does not exist because it would be 1 for the class of the nearest neighbour, and 0 for every other class.

The Dempster’s rule is then used to combine the evidence from all clusters of our partition $\{(S_1, Q_1), \dots, (S_L, Q_L)\}$. It states that the joint mass distribution is defined for every non-empty set of classes $\emptyset \neq C \subseteq \{c_j^1, c_j^0\}$ by

$$m_j(C) = \frac{\sum_{C_1 \cap \dots \cap C_L = C} m_{1,j}(C_1) \dots m_{L,j}(C_L)}{1 - \sum_{C_1 \cap \dots \cap C_L = \emptyset} m_{1,j}(C_1) \dots m_{L,j}(C_L)}$$

Because $m_{l,j}(\{c_j^0\}) = 0$ for all l, j , it follows that the denominator equals 1, and $m_j(\{c_j^0\}) = 0$, and hence $m_j(\{c_j^1\}) = 1 - m_j(\{c_j^1, c_j^0\})$. Then, for $C = \{c_j^1, c_j^0\}$, $C = C_1 \cap \dots \cap C_L$ implies that $C_1 = \dots = C_L = C$, and hence $m_j(\{c_j^1, c_j^0\}) = \prod_{l \in 1..L} m_{l,j}(\{c_j^1, c_j^0\})$. Finally, we arrive at the following equation for the belief of each candidate tail entity e_j .

$$Bel_j = m_j(\{c_j^1\}) = 1 - \prod_{l \in 1..L} (1 - \alpha_0 \phi_{l,j} e^{-d_l})$$

From the belief of each entity e_j , we can rank the entities by decreasing belief. Then, rankings of entities can be evaluated with measures such as Hits@ N (the proportion of inference tasks where the correct tail entity appears in the first N entities) and MRR (Mean Reciprocal Rank, the average of the inverse of the rank of the correct entity).

Note that the above method can easily be generalized to the joint inference of the relation r_k and the tail entity e_j . It suffices to use indices k, j everywhere index j is used: $\phi_{l,k,j}$ would be the confidence of inferring relation r_k and tail entity e_j from Q_l , $c_{k,j}^1$ would be the class of entities linked to e_j through r_k , and $Bel_{k,j}$ would be the belief of inferring such a link.

6 Experiments

We here report on experiments comparing our approach to other approaches on several datasets. We first present the methodology, then we report the main performance results before an in-depth analysis, and examples of inferences and explanations. The companion page¹ provides links to the source code, the datasets, and the output logs.

6.1 Methodology

Datasets. We use three datasets to evaluate our approach. Table 1 provides statistics about them (numbers of entities, relations, train edges, validation edges (if any), and test edges). The main dataset is FB15k-237, introduced by Toutanova and Chen [19] as a challenging subset of dataset FB15k, which was formerly introduced

¹ Companion page: http://www.irisa.fr/LIS/ferre/pub/link_prediction/

Table 1. Statistics of datasets.















































Dataset	entities	relations	train edges	valid. edges	test edges
FB15k-237	15,541	237	272,115	17,535	20,466
JF17k	28,645	322	171,559	-	66,615
Mondial	2,473	20	7,979	778	970

by Bordes *et al* [2] for link prediction evaluation. It is a set of triples derived from the Freebase KG. FB15k-237 is more challenging than FB15k because relations that are almost equivalent to another relation or to the inverse of another relation have been removed, and because (head, tail) entity pairs that exist in the train dataset have been removed from the validation and test datasets to avoid potential trivial inferences. The dataset also comes with textual mentions but we ignore them as we focus on knowledge graphs. The two other datasets are used to complement and confirm results. JF17k is another dataset extracted from Freebase, introduced by [10] and available at <http://github.com/lijp12/SIR>. Although it was designed to go beyond binary relations, we here only consider binary relations, letting n-ary relations to future work. We introduce Mondial as a subset of the Mondial database [13], which contains facts about world geography. We simplified it to the task of link prediction by removing labelling edges and edges containing dates and numbers, and by unreifying n-ary relations. It is available from the companion page.

Task. We follow the same protocol as introduced in [2], and followed by subsequent work. The task is to infer, for each test triple, the tail entity from the head and the relation, and also the head entity from the tail and the relation. We call *test entity* the known entity, and *missing entity* the entity to be inferred. We evaluate the performance of our approach by using the same four measures as in [18]: MRR and Hits@{1,3,10}. Like in previous work, we use filtered versions of those measures to reflect the fact that, for instance, there may be several correct tail entities for a 1-N relation (e.g., the relation from awards to nominees). For example, if the correct entity is at rank 7 but 2 out of the first 6 entities form triples that belong to the dataset (and are therefore considered as valid), then it is considered to be at rank 5.

Method. Because our approach has no training phase we can use both train and validation datasets as examples for our instance-based inference. Our approach has only two (hyper-)parameters (and no parameter to learn) for the computation of CNNs: the *depth* of the description of the test entity, and the *timeout* (i.e. the allocated computation time). We study the sensitivity to those parameters. For the inference of a ranking of entities, we set $\alpha_0=0.95$ and use all computed CNNs (no selection of the k-nearest CNNs). The implementation of our approach has been integrated to SEWELIS as an improvement of previous work on the guided edition of RDF graphs [9]. A standalone program for link prediction is available from the companion page. We ran our experiments on Fedora 25, with CPU Intel(R) Core(TM) i7-6600U @ 2.60GHz, and 16GB DDR4 memory. So far, our implementation is simple and uses a single core, although our algorithm lends itself to parallelization. We have observed that in all our experiments the memory footprint remains under 1.5%, i.e. about 240Mb.

Table 2. Results on FB15k-237 for *Freq*, latent-based approaches (TransE, DistMult, HolE, ComplEx, R-GCN, ConvE), a rule-based approach (AMIE+), and our approach (CNN) with three timeouts (0.01s, 0.1s, 1s): *-results are from [18], **-results are from [4].

Approach	MRR	Hits@1	Hits@3	Hits@10
<i>Freq</i>	.236 	.175 	.253 	.356 
AMIE+	.143 	.096 	.155 	.241 
(from [14])	-	.174 	-	.409 
DistMult*	.191 	.106 	.207 	.376 
ComplEx*	.201 	.112 	.213 	.388 
HolE*	.222 	.133 	.253 	.391 
TransE*	.233 	.147 	.263 	.398 
R-GCN*	.248 	.153 	.258 	.414 
ConvE**	.325 	.237 	.356 	.501 
CNN 0.01s (ours)	.250 	.186 	.268 	.377 
CNN 0.1s (ours)	.264 	.198 	.284 	.395 
CNN 1s (ours)	.286 	.215 	.311 	.428 

Baselines. We compare our approach to latent-based approaches by choosing the same tasks and measures as in previous work because it was not possible for us to run them ourselves (no access to a GPU), and also because it allows for a fairer comparison (e.g., choice of hyper-parameters by authors). On FB15k-237, we use results from [18,4] to compare with TransE, DistMult, HolE, ComplEx, R-GCN, and ConvE. On JF17k, we use results from [20] to compare with mTransH and RAE. We also compare our approach to a rule-based approach, AMIE+, which we ran with its default parameters². As suggested by AMIE+’s authors (equation 8, [8]), we ranked entities e_j by aggregating their *PCA confidence* $\phi_{l,j}$ of each rule R_l that enables to infer triple (e_i, r_k, e_j) : $\phi_j = 1 - \prod_l (1 - \phi_{l,j})$. We also report better results on FB15k-237 from [14], although we were not able to reproduce them. We add yet another baseline *Freq* that simply consists in ranking entities e_j according to their decreasing frequency of usage in r_k over the train+valid dataset, as defined by $freq_j = |\text{ans}(x \leftarrow (x, r_k, y), y = e_j)|$. It is independent of the test entity, and therefore acts as a default ranking.

6.2 Results

Table 2 compares the results of our approach (CNN) to other approaches presented above as baselines on dataset FB15k-237. CNN was run with timeouts that are compatible with user interaction (0.01s, 0.1s, 1s), and description depth 10, which ensures that most if not all relevant graph features of the test entity are captured. The output logs of CNN predictions and explanations is available from the companion page. Except for ConvE that outperforms other approaches on FB15k-237, CNN outperforms all other approaches as soon as 0.01s for the fine-grain measures (MRR,

² We also ran it with advanced parameters on a 8-core server under AMIE+’s authors guidance. That led to many more rules but did not improve the results.

Table 3. Results on JF17k and Mondial for baseline *Freq*, two latent-based approaches (mTransH, RAE), a rule-based approach (AMIE+), and our approach (CNN, timeout=0.1s). Results marked with * are from [20].

Approach	JF17k				Mondial			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
<i>Freq</i>	.234	.170	.252	.355	.142	.069	.159	.309
mTransH*	-	-	-	.497	-	-	-	-
RAE*	-	-	-	.504	-	-	-	-
AMIE+	.211	.139	.252	.360	.179	.127	.208	.281
CNN 0.1s (ours)	.409	.334	.440	.556	.327	.271	.355	.433

Table 4. Results on FB15k-237 of CNN (depth=10, timeout=1s) for all prediction tasks, for predicting tails only, and for predicting heads only.

predicting	MRR	Hits@1	Hits@3	Hits@10
all	.286	.215	.311	.428
tails	.391	.307	.428	.553
heads	.182	.123	.194	.303

Hits@1, Hits@3), and as soon as 1s for measure Hits@10. CNN-1s reaches MRR=0.286, halfway between the two other best approaches, R-GCN (-3.8%) and ConvE (+3.9%).

It can also be observed that CNN outperforms the simple *Freq* baseline on all measures and for all timeouts, which implies that it learns something useful beyond global statistics. Note that this is not the case of other approaches except ConvE, especially for fine-grain measures, MRR and Hits@1. Apart from CNN and ConvE, none improves Hits@1 over *Freq*.

Those positive results are confirmed on the two other datasets, as far as results are available (see Table 3). CNN results are significantly higher than *Freq* and AMIE+ in both datasets and for all measures, by an MRR margin of 17.5% on JF17k and 14.8% on Mondial. On JF17k, CNN also outperforms the latent-based approach RAE by a margin of 5.2% with measure Hits@10=0.556. Given that our approach tends to be better at fine-grain measures, as shown on FB15k-237, the latter result is very encouraging. Indeed, CNN’s first ranked entity is correct 33% of the time on JF17k, which makes its predictions really effective.

6.3 In-depth Analysis

Predicting heads vs tails. Table 4 details the results of CNN on FB15k-237, distinguishing between predicting tails and heads. It shows clearly that it is much easier to predict tails than heads. This is not surprising given that, in KGs, relations are generally oriented in the more deterministic direction. For example, the relation between films and genres is oriented from films to genres because each film has only one or a few genres, while for each genre there are many films. This behaviour is even stronger in Mondial (MRR-all=0.327, MRR-tails=0.584, MRR-heads=0.057) but very small in JF17k (± 0.004). The latter can be explained by the fact that binary relationships are derived from n-ary relations, and therefore do not have a privileged orientation.

Table 5. Results of tail prediction for some of the most frequent relations in FB15k-237. #heads (resp. #tails) is the number of unique heads (resp. tails) for that relation. The MRR of baseline Freq is also included for comparison.

relation	#heads	#tails	MRR _{Freq}	MRR	Hits@1	Hits@3	Hits@10
profession	4245	150	.434	.601	.455	.694	.874
gender	4094	2	.882	.899	.798	1	1
nationality	4068	100	.720	.772	.662	.866	.941
award	3386	406	.080	.270	.154	.296	.511
type_of_union	3033	4	.971	.971	.942	1	1
place_of_birth	2613	704	.155	.183	.100	.235	.359
place_lived	2519	804	.172	.194	.108	.239	.344
film/genre	1875	123	.315	.380	.226	.429	.711
film/language	1735	59	.744	.759	.688	.790	.911
film/country	1708	61	.685	.701	.573	.809	.931

Table 6. Evolution of the number of concepts, the maximum belief, and MRR as a function of timeout on FB15k-237 (depth=1).

timeout	#concepts	max. belief	MRR
0.01	11.8	.467	.235
0.1	49.1	.795	.264
1	219.6	.943	.286

Results per relation. Table 5 details the results further for a selection of 10 of the most frequent relations in dataset FB15k-237, only considering tail prediction. In order to give an idea of the difficulty to predict tails for each relation, we give the number of unique heads and tails in the train+valid dataset, as well as the MRR for baseline *Freq*. The results show that MRR is significantly increased with our approach for all relations, except *type_of_union* whose baseline MRR is already very high at .971. For half of the relations, Hits@1 is greater than .5, which means that predicting the first entity would be more than 50% correct. This includes properties with large numbers of tails, e.g. relation *nationality* has 100 unique tails and Hits@1=.668. In Mondial, the best predicted relations are the continent of a country, the category of a volcano or island, the neighbour sea/ocean of a place, the archipelago of an island or the range of a mountain, with MRR all above 0.500.

Influence of timeout. Table 6 shows the influence of timeout on the resulting MRR, and also on the number of computed concepts and on the maximal belief achieved for predicted entities. It can be observed that with only 1% of the largest timeout, the MRR is already at 82% of the largest MRR, despite the fact that only 5% of the concepts have been computed. This indicates that early approximations of concepts of nearest neighbours are already informative. Furthermore, improving the approximation with more concepts does not only improve MRR but also increases confidence in predictions as indicated by the steadily increasing maximal beliefs.

Influence of description depth. The description depth has a huge impact on the size of descriptions from which edges are chosen to discriminate concepts of nearest neighbours. In FB15k-237, descriptions have on average 750 edges at depth 1, already 20,000 edges at depth 2, and 110,000 edges at depth 10, i.e. 38% of the train+valid edges. This has an impact on computation times but in a reasonable proportion: the runtime overhead relative to the computation of CNNs for timeout=0.1s (i.e., loading triples, computing descriptions, triple inference), ranges from 0.03s at depth 1 to 0.3s at depth 10.

We now look at the impact of description depth on results. We can expect *a priori* that a greater depth provides more information to discriminate candidate entities but is computationally more demanding. However, by varying depth between 1 and 20 for timeout=1s, we observed very small standard deviations for the four measures, all around ± 0.005 . This shows that most of the useful information is already present at depth 1. Nonetheless, it is a good property that increasing depth does not deteriorate performance because it means that depth does not need to be learned and can be set to a large value safely. The explanation for that property is that the iterative partitioning algorithm starts with shallow triples, proceeds with triples of increasing depth, and is usually stopped by timeout rather than by maximum depth.

6.4 Example Inferences and Explanations

Finally, we illustrate our inference method by looking at a few examples in detail (all inferences are available as logs from the companion page). In FB15k-237, the language of film “Dragon Ball Z: Bojack Unbound” is correctly predicted to be “Japanese” with $MRR=1$, compared to $MRR=0.2$ for baseline Freq. CNN-1s generates 26 concepts, from which the best explanation (in terms of belief) for “Japanese” is that the film is from Japan and has Toshiyuki Morikawa as an actor. The living place of “Tabu” is correctly predicted to be “Mumbai” with $MRR=1$, compared to $MRR=0.059$ for baseline Freq. CNN-1s generates 32 concepts, from which the best explanation is that “Tabu” has been awarded with “Filmfare Award for Best Actress”, indicating that many people who earned this award live in Mumbai. The next predicted places are other cities in India. In a more systematic way, we looked at all successful and non-trivial inferences of the nationality of people, i.e. inferences where the correct entity is ranked first by our method, and is not in the three most frequent nationalities ($MRR_{Freq} < 0.333$). Over the 10 such inferences, the number of concepts generated by CNN-1s is remarkably consistent and small, between 22 and 37. The concept intents have 2 or 3 elements (the explanation query), and the concept extents contain between 2 and 29 entities (the similar entities that serve as examples). The best explanations tell that nationality can be inferred by either the living place, the death place, the spoken language, a film in which the person played, or a won award.

In Mondial, the continent of Sweden is correctly predicted as Europe because it is a constitutional monarchy (3-elements query), similarly to Denmark. Matterhorn is correctly predicted to be located in Switzerland because it is a mountain in the Alps that is also located in Italy (5-elements query), similarly to Monte Rossa. Lagen is correctly predicted to be located in Norway because it is the estuary of a river located in Norway (4-elements query).

Our instance-based approach is able to find very specific explanations, as shown by the above illustrations, which a rule-based approach would be unlikely to produce given their huge number. However, a limitation of our inference method is that it cannot yet provide generalized explanations such as “if a person X lives in any city of country Y, then X has nationality Y”, which are the main kind of explanations rule-based and path-based approaches rely on.

7 Conclusion

We have shown that a symbolic approach to the problem of link prediction in knowledge graphs can be competitive with state-of-the-art latent-based approaches. This comes with the major advantage that our approach can provide detailed explanations for each inference, in terms of the graph features. Compared to rule-based approaches, which can provide similar explanations, we avoid the need for a training phase that can be costly in runtime and memory (rule mining), while achieving superior performance. Our approach is analogous to classification with k-nearest neighbours but our distances are defined as partially-ordered graph concepts instead of metrics.

There are many tracks for future work. Extending graph patterns with n-ary relations or richer filters over numbers, dates, etc. Optimizing the computation of CNNs by finding good strategies to drive the partitioning process, or by parallelizing it. Extending the CNN-based inference procedure to mimick non-instantiated AMIE+’s rules (e.g., *the nationality is the country of the living place*). Evaluate our approach on other datasets, and other inference tasks.

Acknowledgement. I warmly thank Luis Galárraga for his support about AMIE+.

References

1. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. *Scientific American* 284(5), 34–43 (2001)
2. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: *Advances in neural information processing systems*. pp. 2787–2795 (2013)
3. Denœux, T.: A k-nearest neighbor classification rule based on Dempster-Shafer theory. *IEEE Trans. Systems, Man, and Cybernetics* 25(5), 804–813 (1995)
4. Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S.: Convolutional 2D knowledge graph embeddings. In: McIlraith, S.A., Weinberger, K.Q. (eds.) *Conf. Artificial Intelligence (AAAI)*. pp. 1811–1818. AAAI Press (2018)
5. Ferré, S.: Concepts de plus proches voisins dans des graphes de connaissances. In: *Ingénierie des Connaissances (IC)*. pp. 163–174 (2017)
6. Ferré, S.: Answers partitioning and lazy joins for efficient query relaxation and application to similarity search. In: Gangemi, A., et al. (eds.) *Int. Conf. The Semantic Web (ESWC)*. pp. 209–224. LNCS 10843, Springer (2018)
7. Ferré, S., Cellier, P.: Graph-FCA in practice. In: Haemmerlé, O., et al. (eds.) *Int. Conf. Conceptual Structures (ICCS)*. pp. 107–121. LNCS 9717, Springer (2016)
8. Galárraga, L., Teflioudi, C., Hose, K., Suchanek, F.: Fast rule mining in ontological knowledge bases with AMIE+. *Int. J. Very Large Data Bases* 24(6), 707–730 (2015)

9. Hermann, A., Ferré, S., Ducassé, M.: An interactive guidance process supporting consistent updates of RDFS graphs. In: ten Teije, A., et al. (eds.) *Int. Conf. Knowledge Engineering and Knowledge Management (EKAW)*. pp. 185–199. LNAI 7603, Springer (2012)
10. Jianfeng, W., Jianxin, L., Yongyi, M., Shini, C., Richong, Z.: On the representation and embedding of knowledge bases beyond binary relations. In: *Int. Joint Conf. Artificial Intelligence (IJCAI)*. pp. 1300–1307 (2016)
11. Lao, N., Mitchell, T., Cohen, W.W.: Random walk inference and learning in a large scale knowledge base. In: *Conf. Empirical Methods in Natural Language Processing*. pp. 529–539. Association for Computational Linguistics (2011)
12. Liben-Nowell, D., Kleinberg, J.: The link-prediction problem for social networks. *Journal of the American society for information science and technology* 58(7), 1019–1031 (2007)
13. May, W.: Information extraction and integration with FLORID: The MONDIAL case study. Tech. Rep. 131, Universität Freiburg, Institut für Informatik (1999), available from <http://dbis.informatik.uni-goettingen.de/Mondial>
14. Meilicke, C., Fink, M., Wang, Y., Ruffinelli, D., Gemulla, R., Stuckenschmidt, H.: Fine-grained evaluation of rule- and embedding-based systems for knowledge graph completion. In: Vrandečić, D., et al. (eds.) *The Semantic Web (ISWC)*. pp. 3–20. LNCS 11136, Springer (2018)
15. Muggleton, S.: Inverse entailment and Progol. *New Generation Computation* 13, 245–286 (1995)
16. Nickel, M., Murphy, K., Tresp, V., Gabrilovich, E.: A review of relational machine learning for knowledge graphs. *Proc. IEEE* 104(1), 11–33 (2016)
17. Plotkin, G.: *Automatic Methods of Inductive Inference*. Ph.D. thesis, Edinburgh University (august 1971)
18. Schlichtkrull, M., Kipf, T.N., Bloem, P., van den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: *The Semantic Web Conf. (ESWC)*. pp. 593–607. Springer (2018)
19. Toutanova, K., Chen, D.: Observed versus latent features for knowledge base and text inference. In: *Work. Continuous Vector Space Models and their Compositionality*. pp. 57–66 (2015)
20. Zhang, R., Li, J., Mei, J., Mao, Y.: Scalable instance reconstruction in knowledge bases via relatedness affiliated embedding. In: *Conf. World Wide Web (WWW)*. pp. 1185–1194 (2018)