An Approach Based on Semantic Similarity to Explaining Link Predictions on Knowledge Graphs

Claudia d'Amato
claudia.damato@uniba.it
Dipartimento di Informatica & CILA
Università di Bari Aldo Moro
Bari, Italy

Pierpaolo Masella
p.masella1@studenti.uniba.it
Dipartimento di Informatica
Università di Bari Aldo Moro
Bari, Italy

Nicola Fanizzi
nicola.fanizzi@uniba.it
Dipartimento di Informatica & CILA
Università di Bari Aldo Moro
Bari, Italy

ABSTRACT

We propose approxSemanticCrossE, an approach for generating explanations to link prediction problems on Knowledge Graphs. Due to their incompleteness, several models have been proposed to predict missing relationships (*link prediction* task). To date, the most effective methods are based on *embedding models*, representing entities and relationships as a multi-dimensional vectors in a vector space. Explaining the results of this task means finding a meaningful reason for which entities are predicted as linked. This work presents a structural and semantically enriched approach for generating explanations for link predictions, by exploring the data available in the knowledge graph. The solution searches for paths and examples of similar situations that justify the prediction carried out using numerical approaches. Specifically, CrossE is adopted as the underlying embedding model to compute predictions. Then explanations are searched exploiting ad hoc semantic similarity measures. The proposed solution has been experimentally evaluated, showing that the new approach is able to provide meaningful explanations compared to the considered baseline.

CCS CONCEPTS

• **Computer systems organization → Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

KEYWORDS

Knowledge Graphs, link prediction, embedding models, explanation

ACM Reference Format:
Claudia d'Amato, Pierpaolo Masella, and Nicola Fanizzi. 2021. An Approach Based on Semantic Similarity to Explaining Link Predictions on Knowledge Graphs. In *IEEE/WIC/ACM International Conference on Web Intelligence (WI-IAT '21), December 14–17, 2021, ESSENDON, VIC, Australia.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3486622.3493956

1 INTRODUCTION

Real business is founded upon data collection and valuable services that exploit huge amounts of data to produce complex mathematical-statistical models. To allow the extraction of even more value, data can be hosted across the Web infrastructure in the form of the so-called *Knowledge Graphs* (KGs).

A KG is as a multi-relational graph intended to convey knowledge of the real world and composed of entities and relations which are regarded as nodes and different types of edges, respectively [7]. It also integrates information defined in well-established Web vocabularies/ontologies allowing the usage of reasoning to derive further explicit knowledge [8]. Several examples of large KGs exist, spanning from enterprise products, such as those built by Google[1] and Amazon[2] (and many others), to several *Open KGs*, such as the DBpedia, Freebase, Wikidata and YAGO. In this work we consider *Open KGs*, which are published online and freely accessible [7].

Due to their inherent incompleteness, two of the most compelling tasks with KGs are *link prediction* and *triplet classification* that roughly amount, resp., to predicting an unknown component of a triple and whether a new triple is true or not. Mostly numeric-based methods are adopted for the purpose, and specifically those based on vector representations (*embedding models*) due to their ability to scale on very large KGs. However they are also characterized by a very low level of interpretability for the human experts. Hence an elusive aspect regards the *trust* that they can place in predictions made through such models (e.g. a side effect predicted for a compound in the context of a KG regarding the drugs domain): the more complex and accurate the models the more difficult to explain become the reasons for their predictions (even by their designers). As a consequence, providing explanations for the predicted results is becoming increasingly important.

It is possible to distinguish the computed explanations into two categories [10]: those related to the internal mechanisms of a model, and those that can motivate the output predictions. Specifically, two possible approaches can be delineated [5]:

A-posteriori methods aim at constructing explanations after the model has provided its predictions; they do not explain the reasons for which the internal mechanism of the model provided a given output, but try to find a suitable explanation based on the observed output and on the model input.

Pattern-based methods guide the process of creating numerical representations of the data contained in the KG by narrowing the search space so that each dimension corresponds to a pattern;

We will focus on the first approach as it allows to adopt link prediction models based on numerical representations of the data, that are more scalable than pattern-based approaches, hence more suitable for real-world large-scale KGs, and capable of generating

WI-IAT '21, December 14–17, 2021, ESSENDON, VIC, Australia

ACM ISBN 978-1-4503-9115-3/21/12…$15.00
https://doi.org/10.1145/3486622.3493956

[1] https://developers.google.com/knowledge-graph
[2] https://aws.amazon.com/it/neptune/knowledge-graphs-on-aws/

explanations for the predictions computed. Actually, there are only few examples of approaches that are able to explain the results of link prediction problems in KGs.

The objective of this work is to define an a-posteriori method for providing *semantic-based* explanations for link prediction on KGs. Specifically, given the link prediction output: ⟨`NickMason`, `recordLabel`, `CapitolRecords`⟩ we aim to understand why this output has been provided, giving valuable reasons so that the user is able to judge this result, understand motivations, and trust (or not) what has been computed. An explanation model should provide an explanation such as:
⟨`NickMason`, `associatedBand`, `PinkFloyd`⟩,
⟨`PinkFloyd`, `recordLabel`, `CapitolRecords`⟩
supported by analogous situations that can be found in the KG, such as ⟨`RingoStarr`, `recordLabel`, `Parlophone`⟩ for which the explanation computed was:
⟨`RingoStarr`, `associatedBand`, `TheBeatles`⟩,
⟨`Beatles`, `recordLabel`, `Parlophone`⟩.

We delineate an explanation process based on new semantic similarity measures to elicit analogous cases on which more accurate explanations can be built, proposing a theoretical framework capable to exploit more thoroughly the KG underlying semantics.

The rest of this work is organized as follows. §2 introduces the basics that are functional to our method definition. In §3 we present our proposed explanation method. The experimental evaluation is described in §4; the explanation process is evaluated in both quantitative and qualitative way to assess the performance of the explanation algorithm and the conceptual correctness of the explanations provided. In §5, related works in the field of explanation on KGs and other kinds of knowledge bases, are discussed. §6 summarizes the conclusions and delineates future works.

2 BASICS

A *Knowledge Graph* is a graph-based data structure $\mathcal{K}(\mathcal{E}, \mathcal{R})$ where $\mathcal{E}$ is the set of the nodes, also known as *entities*, and $\mathcal{R}$ is the set of arcs, also known as *relationships*, which connect entities with each other.

In the adopted RDF model, a KG can be thought of as a set of triples $\langle s, p, o\rangle$, i.e. *subject*, *predicate*, and *object* where $s, o \in \mathcal{E}$ and $p \in \mathcal{R}$. In RDF, the terms are denoted by the elements of the sets $\mathcal{U}$ (URIs), $\mathcal{B}$ (*blank nodes*) and $\mathcal{L}$ (*literals*). Hence an *RDF Graph* is a set triples [5] with: $s \in \mathcal{U} \cup \mathcal{B}$, $r \in \mathcal{U}$, and $o \in \mathcal{U} \cup \mathcal{B} \cup \mathcal{L}$.

2.1 Embedding Models

Several models have been proposed for embedding KGs in low-dimensional vector spaces [8], by learning a unique *distributed representation* (or *embedding*) for each entity and predicate in the KG and considering different representation spaces (e.g. point-wise, complex, discrete, Gaussian, manifold). Here we focus on vector embedding in the set of real numbers.

Regardless of the learning procedure, these models share a fundamental characteristic: given a KG $\mathcal{K}$, they represent each entity $x \in \mathcal{E}$ by means of a continuous *embedding vector* $\mathbf{e}_x \in \mathbb{R}^k$, where $k \in \mathbb{N}$ is a user-defined hyperparameter. Similarly, each predicate $p \in \mathcal{R}$ is associated to a *scoring function* $f_p : \mathbb{R}^k \times \mathbb{R}^k \to \mathbb{R}$. For each pair of entities $s, o \in \mathcal{E}$, the score $f_p(\mathbf{e}_s, \mathbf{e}_o)$ measures the *confidence* that the statement encoded by $\langle s, p, o\rangle$ holds true.

The embedding of all entities and predicates in $\mathcal{K}$ is learned by minimizing a (margin-based) *loss function*.

2.2 Embedding-based Link Prediction: CrossE

CrossE [14] is a KG embedding model for link prediction tasks. It learns embeddings for relations, entities, and triples (*interaction embeddings*); furthermore, explanations for link predictions are also provided. The formulation of the explanation is based on the search for a path linking the subject and object of a predicted triple: this search is driven by similarities, computed by the use of the Euclidean distance, between relation embeddings and then entity embeddings, making structural comparisons with other paths in the KG to reinforce the reliability of the explanation found on the basis of the presence of similar paths (referred to as *support*).

Example 2.1. Given the predicted triple $\langle X, \texttt{fatherOf}, M\rangle$ a suitable explanation is given by the $\langle X, \texttt{hasWife}, Z\rangle$, $\langle Z, \texttt{hasChild}, M\rangle$ and this is supported by an analogous situation given the presence of a triple $\langle Y, \texttt{fatherOf}, X\rangle$ known as true (so, not a prediction) for which the explanation is $\langle Y, \texttt{hasWife}, S\rangle$, $\langle S, \texttt{hasChild}, X\rangle$.

CrossE is based on the concept of *crossover interaction*, i.e. a notion of interaction from relations to entities and from entities to relations, as illustrated in the following example [14].

Example 2.2. Let a sample KG be represented as follows:

$$\begin{aligned}\mathcal{K} = \{&\langle X, \texttt{fatherOf}, Y\rangle, \langle X, \texttt{hasWife}, Z\rangle, \langle X, \texttt{wonPrize}, Q\rangle,\\ &\langle X, \texttt{starredIn}, T\rangle, \langle Y, \texttt{fatherOf}, X\rangle, \langle Y, \texttt{hasWife}, S\rangle,\\ &\langle S, \texttt{hasChild}, X\rangle, \langle Z, \texttt{hasWife}, M\rangle\}\end{aligned}$$

A query can be written as $\langle X, \texttt{fatherOf}, ?\rangle$. The objectives are: a) predicting the suitable object for the triple (e.g. M); b) generating (a set of) explanations for the produced solutions. The entity X is involved in 5+1 triples as subject or object, but only four of them are relevant, as they deal with family relationships. The relation `fatherOf` concerns the family context, and so influences the choice of entities to be considered to produce a prediction: this represents the concept of "*interaction from relations to entities*". The second concept involves entities having some influence on the path to be chosen as an explanation, which intuitively should guide us to consider $\langle X, \texttt{hasWife}, Z\rangle$, $\langle Z, \texttt{hasWife}, M\rangle$: this represents the concept of "*interaction from entities to relations*".

Hence, for each entity and relationship the model defines:

- a general embedding that preserves structural information about the topology of the KG
- multiple triple-specific embeddings, the interaction embeddings, which maintain properties concerning the crossover interactions, generated by the Hadamard product between an interaction matrix and the general embeddings

Each of these components is represented by a suitable embedding matrix. Specifically, three matrices must be learned (having been initialized according to a uniform distribution $U(-6/\sqrt{d}, 6/\sqrt{d})$):

- $\mathbf{E} \in \mathbb{R}^{n_e \times d}$ *general embeddings* of the entities, one per row
- $\mathbf{R} \in \mathbb{R}^{n_r \times d}$ *general embeddings* of the relations, one per row

- $\mathbf{C} \in \mathbb{R}^{n_r \times d}$ *interaction matrix* in which each row is related to a specific relation based on the contexts they are involved

where n_e and n_r are, resp., the number of entities and relationships in the training KG, while d is the embeddings dimension. These matrices are exploited for generating the *interaction embeddings.* All final interaction matrices are then used by the explanation process.

3 THE PROPOSED EXPLANATION METHOD

We propose approxSemanticCrossE, a method for computing explanations for link predictions on KGs exploiting their semantics. Specifically, moving from CrossE, we extend the solution by:

- adapting the similarity measure that leads the explanation process. Specifically the cosine similarity is adopted rather than the Euclidean distance in order to take into account the angle of the vector embeddings which is disregarded by the Euclidean distance, that focuses rather on the magnitude of the feature values in the embedding space;
- formalizing a theoretical framework to guide the explanation process by exploiting the underlying semantics of the KG in order to create more accurate explanations;
- defining an approximation within this theoretical framework to be used for coping with scalability issues with large KGs.

The base explanation process borrowed from CrossE is now recalled, to introduce the extensions related to the employed similarity measures that are illustrated and motivated in §3.1-§3.3.

Given a predicted triple $\langle h, r, t\rangle$ for the query $\langle h, r, ?\rangle$, the main idea consists in looking for the shortest *paths* from h to t, and provide them as explanations. This search aims at finding analogous situations that can support the explanation: this requires a structural comparison between paths, to draw "support" to the explanation.

The whole process is summarized in Alg. 1, whose steps can be described as follows:

Given the predicted $\langle h, r, t\rangle$:

(1) find the set $\mathcal{S}_r$ of the k_r closest relationships to r
(2) search for the set $\mathcal{P}(h, t)$ of (all) paths between h and t
   - a maximum length is fixed to limit the search space; we will consider length 2, hence six types of paths are possible, length 1: $P_1 = \{\langle h, r_s, t\rangle\}$, $P_2 = \{\langle t, r_s, h\rangle\}$ and length 2: $P_3 = \{\langle e', r_s, h\rangle, \langle e', r', t\rangle\}$, $P_4 = \{\langle e', r_s, h\rangle, \langle t, r', e'\rangle\}$, $P_5 = \{\langle h, r_s, e'\rangle, \langle e', r', t\rangle\}$, $P_6 = \{\langle h, r_s, e'\rangle, \langle t, r', e'\rangle\}$, where r_s is a relationship similar to r, r' stands for any other relationship, and e' is any other entity;
   - a direct search is employed to find similar paths of type 1 and 2, and bidirectional search to find paths of types 3 through 6.
(3) find the set $\mathcal{S}_h$ of the k_e closest entities to h;
   - note that considering $h_s \in \mathcal{S}_h$, entities t_s s.t. $\langle h_s, r, t_s\rangle \in \mathcal{K}$ are also determined
(4) search for similar structures to support the explanation
   - if $\exists P \in \mathcal{P}(h_s, t_s)$ such that $\langle h_s, P, t_s\rangle$ which denotes a path P between h_s and t_s in $\mathcal{K}$ (with $\langle h_s, r, t_s\rangle$ determined at step (3)) then P is an *explanation* for $\langle h, r, t\rangle$
   - the triples satisfying the previous condition describe an analogous situation involving similar entities and relationships: the support is given by P which joins a similar head to a similar tail through a relation that is similar to r, similarly to r in $\langle h_s, r, t_s\rangle$.

Algorithm 1 Explanation of the predictions

```
require K(E, R): knowledge graph
        ⟨h, r, t⟩: predicted triple
ensure  explanations of the predicted triple and and their supports
Expl ← ∅; /* explanation set */
Supp ← ∅ /* support set */
Select S_r ⊆ R, with |S_r| = k_r /* the most similar relations to r */
for each path−type i do
  Find the path−set P_i = {P | ⟨h, P, t⟩ ∈ K ∧ P of type i}
Select S_h ⊆ E, with |S_h| = k_e /* the most similar entities to h */
for P ∈ P = ⋃_i P_i do
  if ∃⟨h_s, p, t_s⟩ ∈ K ∧ h_s ∈ S_h
    Expl ← Expl ∪ P
    Supp ← Supp ∪ {{⟨h_s, p, t_s⟩, ⟨h_s, r, t_s⟩}}
return Expl, Supp
```

Originally, the analogy between pairs of entities or relationships was assessed by means of the Euclidean distance, applied to their embeddings In the next subsections we move towards the cosine similarity and adapt the measures to better exploit the underlying semantics.

3.1 Cosine Similarity

CrossE is grounded on the intuition that having more interaction embeddings for each entity and for each relationship leads to richer and more *generalization-capable* representations, by capturing more of the latent information. The similarity measure in CrossE framework plays a crucial role in this regard. Particularly, the Euclidean distance is adopted to measure the similarity between embeddings (of entities or relations) in the geometric space of the vector embeddings. However it is well known that Euclidean distance, focusing on the magnitude of the vector distances in the geometric spaces, it may present drawbacks when data values are not balanced. In order to avoid incurring in such an issue, we investigated the usage of other measures.

Cosine similarity[3], widely used in NLP contexts, expresses the similarity between vectors in terms of the angle between them (the smaller the angle, the closer they are). Vectors with the same orientation have a cosine similarity of 1, orthogonal vectors (i.e. forming an angle $\theta = \pi/2$) have a null similarity, and diametrically opposed vectors ($\theta = \pi$) have a similarity of -1, independently of their magnitude.

This rationale is more appropriate to our purposes. Indeed, graph embedding methods project the data into an optimal low-dimensional space in which structural information and properties are preserved as much as possible. Two vectors of the embedding space could be far from each other in terms of Euclidean distance, and still exhibit a high cosine similarity. With methods like CrossE, this may be useful since embeddings based on interactions from

[3]Given vectors $\mathbf{x}$ and $\mathbf{y}$, it is expressed as the cosine of the angle θ between them:

$$\mathrm{sim}_{cos}(\mathbf{x}, \mathbf{y}) = \cos(\theta) = \mathbf{x} \cdot \mathbf{y} / \|\mathbf{x}\| \|\mathbf{y}\|.$$

relations to entities and interaction from entities to relations are generated. In §4, we prove the correctness of this intuition by showing corroborating experimental results.

In the following, we motivate the need for exploiting the additional semantics in KG and formalize the theoretical framework to guide the explanation process by means of the added semantics.

3.2 The Semantic Cosine Similarity

Both Euclidean distance and Cosine similarity are not able to take into account the semantics of the KGs, which is rather rich particularly when expressive representation languages such as RDF-S and OWL are adopted. Being able to exploit the KG semantics may lead to generate more accurate explanations for link predictions. To this purpose, in particular domains, ranges, and classes will be considered: given a relationship, domain and range allow to specify the classes whose instances can occur as head or tail in triples. Adopting the notation of *Description Logics* (DLs) [1], that constitute the theoretical foundation of OWL, classes may be defined via (complex) DL expressions based on primitive concepts, such as: $\texttt{Mother} \equiv \texttt{Female} \sqcap \exists \texttt{hasChild.Being}$.

Since many KGs refer to shared OWL ontologies endowed with deductive reasoning capabilities that allow to infer additional knowledge, the idea is to exploit the underlying semantics when searching for similar entities/relations to produce explanations. Together with standard reasoning services such as class subsumption (denoted with $\sqsubseteq$), we will resort to *retrieval* as a function $\mathrm{ret}_{\mathcal{K}}(C)$ (subscript omitted when obvious from the context) returning the known entities that can be proven to belong to a given class C.

Inferences enable the exploration of the data, in terms of domains and ranges (for the relationships) and classes (for entities), to better direct the search for explanations in a semantics-aware fashion, supported also by the similarity measures that can be defined on the embeddings produced by KG embedding methods. Hence, we introduce the *semantic Cosine* measure which is meant to increase,the cosine similarity of two (entities or relationships) vector embeddings on the ground of available additional semantic information. Such information is captured by a *semantic score* function defined for the purpose. Formally:

Definition 3.1 (semantic Cosine). Given the KG $\mathcal{K}(\mathcal{E}, \mathcal{R})$, the *semantic Cosine* measure for two entities $e, e' \in \mathcal{E}$ is defined by:

$$\mathrm{semCos}_{\alpha,\beta}(e, e') = \alpha \cdot \mathrm{sScore}(e, e') + \beta \cdot \mathrm{sim}_{\cos}(\mathbf{e}, \mathbf{e}') \qquad (1)$$

where $\mathbf{e}$ represents the respective entity embedding vector and $\alpha, \beta \in [0, 1]$ are chosen so that $\alpha + \beta = 1$.
In the case of relations $r, r' \in \mathcal{R}$ the measure is defined analogously.

Definition 3.2 (semantic Score). Given the set C of the classes occurring in $\mathcal{K}(\mathcal{E}, \mathcal{R})$, and the functions $Cl\colon \mathcal{E} \to C$, $Do\colon \mathcal{R} \to C$, and $Ra\colon \mathcal{R} \to C$ that return, resp., the conjunction of the classes an entity belongs to, and the domain and range of a relation, the *semantic Score* function for pairs of entities $e, e' \in \mathcal{E}$ is defined by:

$$\mathrm{sScore}(e, e') = \frac{|\mathrm{ret}[Cl(e) \sqcap Cl(e')]|}{|\mathrm{ret}[Cl(e) \sqcup Cl(e')]|}. \qquad (2)$$

Analogously, given any two relationships $r, r' \in \mathcal{R}$, it is defined:

$$\mathrm{sScore}(r, r') = \frac{|\mathrm{ret}[Do(r) \sqcap Do(r')]|}{|\mathrm{ret}[Do(r) \sqcup Do(r')]|} + \frac{|\mathrm{ret}[Ra(r) \sqcap Ra(r')]|}{|\mathrm{ret}[Ra(r) \sqcup Ra(r')]|} \qquad (3)$$

Example 3.3 (Computing the semantic Score). Let us suppose that $Cl(e) = \texttt{Student}$ and $Cl(e') = \texttt{Student} \sqcup \texttt{Worker}$. Then:

$$\mathrm{sScore}(e, e') = \frac{|\mathrm{ret}[\texttt{Student} \sqcap (\texttt{Student} \sqcup \texttt{Worker})]|}{|\mathrm{ret}[\texttt{Student} \sqcup \texttt{Worker}]|}.$$

Similarly, the semantic Score for relations can be computed by considering their domains and/or ranges, that are ultimately class expressions, and summing the degree of similarity between the domains and the degree of similarity between the ranges.

However computing concept retrieval by using a standard reasoner may turn out to be computationally prohibitive, or even infeasible from a practical viewpoint, when very large KGs, consisting of millions of triples, are considered. For this reason, an approximated form of the semantic Cosine measure and more specifically of the semantic Score function is proposed.

Approximate Semantic Score. Starting from the above formulation, we design an approximation of the semantic Score function, also exploiting some recurrent facts of the KGs, summarized below:

- in most cases class names are used, not complex expressions
- domains and ranges are often defined via single class names or their disjunctions
- class Thing is not informative: its extension includes all entities.

The new measure is grounded on the same rationale adopted by the semantic Score function but it moves to a moresyntactic level, working on sets of concept names. Specifically, we do not consider concept retrieval but only the class names themselves, and intersections and unions between class names. To this purpose, the functions Cl, Do, Ra need to be redefined as follows:

$$\widetilde{Cl}\colon \mathcal{E} \to 2^{\mathrm{CNames}}, \quad \widetilde{Do}\colon \mathcal{R} \to 2^{\mathrm{CNames}}, \quad \widetilde{Ra}\colon \mathcal{R} \to 2^{\mathrm{CNames}}$$

where CNames represents the set of class names in the KG, so they return possibly a collection of class names. The *approximate semantic Score function*, denoted $\widetilde{\mathrm{sScore}}$, is defined as in Def. 3.2, replacing these functions to the original ones and class conjunction/disjunction ($\sqcap/\sqcup$) with set intersection/union ($\cap/\cup$). The combination of the cosine similarity with the approximate semantic score, yields the *approximate Cosine similarity* $\widetilde{\mathrm{semCos}}$.

3.3 Explanations with the Semantic Cosine

The (approximate) semantic cosine measure are meant to be used in the explanations process (see Alg. 1) for the construction of sets of relations and entities that are most similar to those involved in the predicted triple.

Before the execution of the explanation search algorithm, the generation of the set $\mathcal{S}_r$ is carried out so that
$\forall r \in \mathcal{R} : \mathcal{S}_r = \emptyset$

(1) $\forall r' \in \mathcal{R} : \mathcal{S}_r \leftarrow \mathcal{S}_r \cup \{(\widetilde{\mathrm{semCos}}(r, r'))\}$
(2) sort $\mathcal{S}_r$ in descending score order

(analogously for $\mathcal{S}_h$, with pairs of entities as arguments of $\widetilde{\mathrm{semCos}}$)

The following considerations need to be made to manage the Thing class case:

- if Thing is the minimal concept both entities belong to, no semantic score can be added as no additional information is available, so just the cosine similarity is used;

- similarly, given two relations
  - if the domain of one of them is `Thing`, the contribution of the domain to the score is null, and only the range is considered in $\widetilde{\text{sScore}}$ as a reward to be added to the cosine similarity
  - if the range of one of them is `Thing`, the contribution of the range in the score is null and only the domain is considered to be added to the cosine similarity in $\widetilde{\text{sScore}}$
  - if both domain and range are `Thing`, nothing is added to the cosine similarity (i.e. $\widetilde{\text{sScore}} \leftarrow 0$).

Some further considerations can be made: the selection of the most similar entities and relations can be optimized through clustering processes based on embeddings: it would be possible to guide the search for the most similar relations/entities by taking the first k relations/entities belonging to the same cluster.

4 EVALUATION

The objective of the experiments is to analyze the explanations generated for link prediction results, and establishing the impact of an added semantic component, as realized by ApproxSemanticCrossE, on the explanations provided by the base-model CrossE. In the following, we present the experimental design and then discuss the outcomes.

The proposed solutions have been implemented in Python, while the experiments have been executed on a remote cluster[4]. Code and datasets are publicly available[5].

4.1 Explanation Evaluation Metrics

A quantitative and qualitative evaluation of the generated explanations for link prediction results has been performed. The former part was aimed at assessing how capable the algorithm is of generating useful explanations for the predicted triples; the latter was aimed at assessing how accurate and meaningful they are. In the literature, there are still no consolidated evaluation methodologies for the explanations offered by this kind of models. In agreement with the evaluation of CrossE in [14], we will adopt the following metrics for the quantitative evaluation:

- *Recall*: the ratio of triples for which the model can generate explanations, determined as follows:
$$\frac{\#\textit{predictions with at least one explanation}}{\#\textit{total predictions}};$$
  - conforming to the evaluation of CrossE, the generation of explanations is limited to paths of max. length 2which affects the number of possible explanations, maintaining a greater focus on their quality and brevity;
  - this metric does not take into account the number of explanations per predicted triple so, regardless of their number, the recall does not change;
- *Average Support*: the number of explanations, on average, for each prediction. It quantifies the reliability of explanations: the larger the support the more reliable and credible the prediction. It is defined as:
$$\textstyle\sum_{x \in P} |\text{Expl}(x)|/|P|$$
where Expl returns the set of explanations generated for the predicted triple x and P is the set of predictions for a query: $\{\langle h, r, t\rangle \mid \langle h, r, t\rangle \text{ is predicted for query } \langle h, r, ?\rangle\}$.

As for the qualitative analysis, explanations have been analyzed to understand if those generated adopting the approximate cosine similarity resulted more meaningful than those generated with the original setting of CrossE, based on the Euclidean distance and its variant based on the cosine similarity.

4.2 Datasets

For the sake of comparison, the very same KGs adopted by CrossE have been considered, except for *FB15k*, due to execution time limits on the remote machines adopted: due to the huge training set, processes were interrupted. Furthermore, since these KGs lack of significant semantic information actually taken into account by ApproxSemanticCrossE, we additionally considered *DBpedia15k* as further test for stressing on the possible utility of the semantic component or not. Details on the adopted KGs are summarized below:

- **FB15k-237** contains 14541 entities and 237 relationships. It is a subset of the original dataset *FB15k* containing relation triples and textual mentions of *Freebase*[6] entity pairs;
- **WN18** contains 40943 entities and 18 relations. It is a dataset extracted from *WordNet*[7], where entities correspond to *synsets* (sets of synonyms) and relations represent lexical connections (e.g., *hypernymy*, etc.);
- **DBpedia15k** contains 12862 entities and 279 relations with 180,000 triples extracted from *DBpedia* [11].

4.3 Parameters Setting

Since CrossE was used for the preliminary link prediction phase, the settings used in [14] were kept unchanged. The authors suggest to consider a fixed initial number k of similar relations and j of most similar entities. Clearly, the larger these values, the greater would be the recall but also the noise entailed. As our objective is to generate good quality explanations, small values have been considered: $k = j = 3$. As concerns the semantic score function, the considered settings for the weights was $\alpha = 0.2$ and $\beta = 0.8$; the motivation is that cosine similarity uses embeddings computed by CrossE, incorporating more latent information learned, and the semantic measure enforces the similarity complementarily.

Finally, also for the link prediction parameters the adopted values were the same used for CrossE in [14]: matrices $\mathbf{E}$, $\mathbf{R}$ and $\mathbf{C}$ were initialized with the uniform distribution; the bias vector $\mathbf{b}$ initialized to $\mathbf{0}$, the positive triples for training were those in $\mathcal{K}_{\text{train}}$ while negative triples for training were sampled among those not occurring in $\mathcal{K}_{\text{train}}$. The TensorFlow implementation of the model was used with Adam optimizer and dropout of 0.5 applied to the similarity operator (max. number of iterations: 500). The parameters that depend on the adopted datasets are in Tab. 1.

[4] A cluster of 20 servers, each with 40 cores equipped with a NVIDIA Tesla K40 GPU.
[5] https://github.com/pierulohacker/SemanticCrossE/tree/master/explanation
[6] https://web.archive.org/web/20100228011242/http://www.freebase.com/
[7] https://wordnet.princeton.edu/

Table 1: Dataset-specific parameter settings

params / datasets	*WN18*	*FB5K-237*	*DBpedia15k*
neg. examples n	50	50	50
learning rate η	0.01	0.01	0.01
embeddings dim. d	100	100	100
regulariz. param. λ	10^{-4}	10^{-5}	10^{-5}
batch size B	2048	4000	4000

Table 2: Results with different measures and percentages: recall and average support per explanation by path type

dataset	measure	%	recall	avg.support					
				1	2	3	4	5	6
FB15k-237	orig.	2%	0.0297	2.94	1.82	39.83	49.47	35.34	29.19
		5%	0.0154	2.88	1.80	38.79	49.16	34.19	26.91
	cos	2%	0.0304	3.20	2.07	39.80	50.82	40.61	33.43
		5%	0.0162	3.15	2.04	37.65	47.28	39.00	30.27
WN18	orig.	2%	0.0026	1.00	4.19	2.94	2.74	2.05	2.04
		5%	0.0010	1.00	4.19	2.86	2.67	1.99	1.99
	cos	2%	0.0029	1.00	4.13	2.16	2.11	2.03	2.06
		5%	0.0010	1.00	4.13	2.12	2.07	1.99	2.01
DBpedia15k	orig.	2%	0.0024	1.34	1.21	2.95	1.46	1.86	2.59
		5%	0.0010	1.34	1.20	2.93	1.45	1.84	2.50
	cos	2%	0.0021	1.35	1.18	2.97	1.51	1.94	2.69
		5%	0.0009	1.34	1.17	3.08	1.50	1.93	2.66
	acos	2%	0.0020	1.35	1.17	2.95	1.51	1.94	2.69
		5%	0.0010	1.35	1.18	2.95	1.50	1.92	2.58

4.4 Results and Discussion

In the following the quantitative and qualitative results of the experiments carried out are summarized and discussed.

4.4.1 Quantitative Evaluation. The *recall* and the *average support* for each of the six types of explanation path (see §3) have been computed: this allows to inspect the types on which the algorithm performs better. The explanation algorithm has been executed on the predicted triples of each dataset. Different ratios of predictions have been considered for building explanations, starting from those ranking higher in the link prediction results. This is because predicted triples that are very low in the returned ranked results may turn out to be incorrect and, as a consequence, the corresponding explanations less reliable. The numbers of link predictions provided for each dataset were, resp.:

- *FB15k-237*: 29,7596,106 (i.e. 20466 test triples $\times$ 14541 entities)
- *WN18*: 20,471,500 (i.e. 500 test triples $\times$ 40943 entities)
- *DBpedia15k*: 471,302,266 (i.e. 36643 test triples $\times$ 12862 entities)

The quantitative results are summarized in Tab. 2. The three settings compared in the evaluation are indicated as orig., cos, acos: the first corresponds to the baseline method adopting the Euclidean distance, i.e. the original approach [14]; in the second setting the cosine similarity is adopted, and the third involves the approximate semantic Cosine measure which was tested only on *DBpedia15k*, since, as argued in §4.2, it was the only dataset with semantic annotations.

The first thing that can be noticed is that for the case of *FB15k-237* (both for 2% and 5% predictions, across all similarity settings), larger support and recall values are registered when compared to the other datasets. This is because *FB15k-237* has almost twice the triples of the other two datasets and, for the explanation process, this means counting on a larger knowledge base in which the algorithm can find explanations for the predictions made. It is also worthwhile to remind that a fixed number of top 3 similar relationships and top 3 similar entities have been considered for computing the explanations (see discussion in §4.3), which limits the computational costs but also the recall. Overall, looking at the outcomes in the table it is possible to notice that the recall values for the three settings are rather close,with a slight exception for DBpedia15k, characterized by richer semantic annotations, where surprisingly recall is higher in the orig. setting, decreases for cos setting, and further decreases for acos setting (for both 2% and 5% options). However, as shown by the qualitative analysis presented in §4.4.2, the use of the Euclidean distance (orig. setting) offers more results, but often introducing noisy (irrelevant) explanations whilst both cos and acos have shown to offer more meaningful results, with acos maintaining the best level of conceptual meaningfulness.

As mentioned before, inspecting the average support by path type helps to understand what kind of explanations are most commonly retrieved. We can observe that, for *FB15k-237*, the explanation paths of type 1 and type 2 are not common at all, in fact they represent respectively 2% and 1% of the paths retrieved as support for the explanations; on the other hand, the support to the explanation in *WN18* and *DBpedia15k* seems to be more stable on average: for *WN18*, the less common support path is type 1 (7%), and for *DBpedia15k* it is the path of type 2 (10%). A common aspect across the datasets is that less type 3 paths are retrieved in the cos setting w.r.t. the orig. setting. The capability of the algorithm to exploit different types of paths is strictly connected to the embedding method adopted: this observation was also noted in [14], given that the authors compared their model (i.e. the orig. setting) with explanations provided using the embeddings generated by TransE [3] and Analogy [12].

4.4.2 Qualitative Results. The qualitative analysis required inspections on the output data to evaluate if and how the explanations change using the different distance/similarity measures. As for *DBpedia15k*, characterized by richer semantic annotations, explanations for link prediction tasks have been computed considering the three settings. The analysis below shows that the use of the Euclidean distance (orig. setting) seems less suitable to produce meaningful explanations; on the other hand, both cos and acos have shown to offer more meaningful results: specifically, acos maintains the best level of conceptual meaningfulness.

In the following, the discussion continues with explanatory examples to describe scenarios where: orig. typically offers more results, but often introducing noisy (irrelevant) explanations; cos ensures a good amount of explanations yet including some that are not very sensible, similarly to the orig. setting; acos is more selective with more correct explanations.

Given the three settings, in some cases the same explanations were produced. As an example, let us consider the predicted triple ⟨`Dio`, `recordLabel`, `WarnerBros`⟩, whose explanations are shown in Fig. 1. The motivation is that the similarities computed for the relationships and the entities involved are quite the same, due to absence of additional information. An important observation can be made on this output case: cosine and semantic settings yielded the same $\mathcal{S}_r$, and the reason is that this relationship had no additional information about domain and range, so these correspond to `Thing`:

from the analysis carried out, in fact, the ratio of relationships with a restricted domain (resp. range) in *DBpedia15k* is 26.16% (36.2%).

There are several cases in which the settings based on the new measures are able to explain predictions that the original setting is not able to explain. This is probably due to the inability to fully exploit latent information. As an example, the prediction ⟨ThomasMitchell, birthPlace, LA⟩ is explained by
{⟨ThomasMitchell, deathPlace, California⟩,
⟨LA, partOf, California⟩}
that is supported by ⟨Glen_ALarson, birthPlace, LongBeachCA⟩ for which the explanation found was:
{⟨Glen_ALarson, deathPlace, California⟩,
⟨LongBeachCA, partOf, California⟩}.

It is interesting to notice that this explanation is also conceptually meaningful because they were both involved in the show business and this latent information was retrieved.

Another meaningful example is shown in Fig. 2: given the prediction ⟨NickMason, recordLabel, Capitol⟩, we can notice that both cos and acos produced two type-5 explanations, but the orig. setting could only offer one type-3 explanation and one type-5 explanation. The orig. setting uses as a type-3 example, namely ⟨RobertPlant, recordLabel, Mercury⟩, for which, the computed explanation was:
{⟨NationalSecurity (2003 film), formerBandMember, RobertPlant⟩,
⟨NationalSecurity (2003 film), recordLabel, Mercury⟩}
that is part of the KG but contains *wrong* triples (*DBpedia* is produced by a partially automated process); the cos and acos settings, thanks to their awareness about latent information, avoided type-3 explanations, being more complete in supporting only type-5 explanations.

The acos setting produced often more specific and accurate results, as shown in the following case: given the prediction ⟨DavidHume, influencedBy, BaruchSpinoza⟩, the acos was able to generate one explanation of type 3 and two explanations of type 5. The type-3 explanation:
{⟨GottfriedWLeibnitz, influenced, DavidHume⟩,
⟨GottfriedWLeibnitz, influencedBy, BaruchSpinoza⟩}
seems meaningful because it makes sense to say that a philosopher is influenced by the one who influenced his/her source of inspiration, however some type-5 explanations seem to be much weaker, for example:
{⟨DavidHume, influenced, ImmanuelKant⟩,
⟨ImmanuelKant, influencedBy, BaruchSpinoza⟩}
it is less compelling to explain that Hume was influenced by Spinoza because Hume influenced Kant who was influenced by Spinoza. On the other hand, both orig. and cos introduced more conceptually wrong explanations. For example, the cos setting offered an analogous type-3 explanation, but also the type-6 explanation:
{⟨DavidHume, influenced, ImmanuelKant⟩,
⟨BaruchSpinoza, influenced, ImmanuelKant⟩} .

The orig. setting introduced also more noisy and weaker explanations. An exception concerns a type-6 explanation provided by orig. and cos:
{⟨DavidHume, mainInterest, Epistemology⟩,
⟨BaruchSpinoza, mainInterest, Epistemology⟩}
which may be also considered as meaningful: if two philosophers share their main topic, one may have influenced the other.

5 RELATED WORK

The mentioned paper [14] is a good starting point for the problem of explaining link predictions in the specific context of the methods for KGs. Useful insights are offered also in [2] which tackles the problem in the case of graphs that evolve over time. The proposed model can: learn representations for new entities not seen during the training phase; infer new links between them and those that already occur in the graph; offer a reasoning path as an explanation downstream of the output obtained by adopting an embedding based method. The model is based on a *Graph Transformer* that learns entity embeddings by iteratively aggregating information from neighbouring nodes, weighted according to the relevance to the query; the problem is cast as a *Partially Observable Markov Decision Process* regarding the graph as a partially observable environment: relationships departing from each node correspond to (deterministic) actions that an agent can explore to reach the response (target entity) from the starting entity and thus receive a reward;a LSTM that preserves the path history is used to guide the search for the explanation path. This *search policy* is optimized to find the goal more efficiently.

In the *Thales XAI Platform* [9], the authors emphasise how crucial it is to contextualize the data by connecting to domain-specific KGs, so to enrich them with additional contextual knowledge. To explain a prediction, highly representative areas of the graph are sought: to detect such areas, parts of the KG are iteratively removed to assess the related loss. The model provides an encoded context, tangible relationships and connections between data, native support for inference and consciousness of cause and effect. Another related problem is the explanation of clustering processes applied to graphs. In [6] the following workflow is proposed: 1) *Embedding learning* with two possible configurations: TransE [3] and Complex [4]; 2) *Clustering* based on the learned embeddings; 3) *Rule learning* to explain the generated clusters: a new learner is proposed for producing *Horn rules*; 4) *Rule-based inference* of the membership of new entities wrt the clusters; 5) *Embedding adaptation*: fine-tuning on target entities by constructing a feedback to guide clustering in successive iterations, thus trying to discover new similarities by adding triples that represent the learned inference rules.

Finally, in [13] a solution is proposed to the link prediction and triple classification tasks that integrates a rule-based with an entity embedding component. Although it does not deal explicitly with explanations, it offers interesting insights for future implementations of a rule-based explanation process. The idea is based on a framework consisting of encoders and decoders: The *encoder* encodes the subjects and objects of the triples into embeddings, exploiting an *aggregator* that guides their generation; the *decoder* evaluates the plausibility of the training triples in relation to the query made. This evaluation borrows the TransE [3] scoring function. A particular attention is paid to the aggregator: it aggregates several vectors producing an output embedding vector for the entity of interest (subject or object) by gathering information from its neighbourhood, while preserving fundamental properties to the purpose, namely: *permutation invariance*: the order of the neighbours (relations) of the target entity does not matter, typical when dealing with graphs; *redundancy awareness*: aggregations must be as informed as possible, and exploit the data redundancy, that is

type	triple **explanation**	**support** examples	**support** explanations
3	{⟨BlackSabbath, associatedMA, Dio⟩, ⟨BlackSabbath, recordLabel, WarnerBros⟩}	⟨SteelyDan, recordLabel, Giant⟩	{⟨WalterBecker, associatedMA, SteelyDan⟩, ⟨WalterBecker, associatedMA, Giant⟩}, {⟨DonaldFagen, associatedMA, SteelyDan⟩, ⟨DonaldFagen, associatedMA, Giant⟩}
		⟨SteelyDan, recordLabel, MCA⟩	{⟨WalterBecker, associatedMA, SteelyDan⟩, ⟨WalterBecker, associatedMA, MCA⟩}, {⟨DonaldFagen, associatedMA, SteelyDan⟩, ⟨DonaldFagen, associatedMA, MCA⟩}
	{⟨BlackSabbath, associatedBand, Dio⟩, ⟨BlackSabbath, recordLabel, WarnerBros⟩}	⟨SteelyDan, recordLabel, Giant⟩	{⟨DonaldFagen, associatedBand, SteelyDan⟩, ⟨DonaldFagen, associatedBand, Giant⟩} {⟨DonaldFagen, associatedBand, SteelyDan⟩, ⟨DonaldFagen, associatedBand, MCA⟩}
5	{⟨Dio, associatedMA, BlackSabbath⟩, ⟨BlackSabbath, recordLabel, WarnerBros⟩}	⟨WalterBecker, recordLabel, Giant⟩	{⟨WalterBecker, associatedMA, SteelyDan⟩, ⟨SteelyDan, recordLabel, Giant⟩}, {⟨WalterBecker, associatedMA, DonaldFagen⟩, ⟨DonaldFagen, recordLabel, Giant⟩}
		⟨WalterBecker, recordLabel, MCA⟩	{⟨WalterBecker, associatedMA, SteelyDan⟩, ⟨SteelyDan, recordLabel, MCA⟩}, {⟨WalterBecker, associatedMA, DonaldFagen⟩, ⟨DonaldFagen, recordLabel, MCA⟩}
	{⟨Dio, associatedBand, BlackSabbath⟩, ⟨BlackSabbath, recordLabel, WarnerBros⟩}	⟨WalterBecker, recordLabel, Giant⟩	{⟨WalterBecker, associatedBand, DonaldFagen⟩, ⟨DonaldFagen, recordLabel, Giant⟩} {⟨WalterBecker, associatedBand, DonaldFagen⟩, ⟨DonaldFagen, recordLabel, MCA⟩}

Figure 1: Explanation for ⟨Dio, recordLabel, WarnerBros⟩

type	triple **explanation**	**support** examples	**support** explanations
3 orig.	{⟨PinkFloyd, formerBandMember, NickMason⟩, ⟨PinkFloyd, recordLabel, Capitol⟩}	⟨RingoStarr, recordLabel, UnitedArtists⟩	{⟨TheBeatles, formerBandMember, RingoStarr⟩, ⟨TheBeatles, recordLabel, UnitedArtists⟩}
		⟨RobertPlant, recordLabel, Mercury⟩	{⟨NationalSecurity (2003 film), formerBandMember, RobertPlant⟩, ⟨NationalSecurity (2003 film), recordLabel, Mercury⟩},
5 orig.	{⟨NickMason, associatedMA, PinkFloyd⟩, ⟨PinkFloyd, recordLabel, Capitol⟩}	⟨RingoStarr, recordLabel, UnitedArtists⟩	{⟨RingoStarr, associatedMA, TheBeatles⟩, ⟨TheBeatles, recordLabel, UnitedArtists⟩}
5 cos acos	{⟨NickMason, associatedMA, PinkFloyd⟩, ⟨PinkFloyd, recordLabel, Capitol⟩}	⟨RickRubin, recordLabel, AmericanRecordings⟩	{⟨RickRubin, associatedMA, Dazing⟩, ⟨Dazing, recordLabel, AmericanRecordings⟩}
		⟨CarlosSantana, recordLabel, Artista⟩	{⟨CarlosSantana, associatedMA, Santana (band)⟩, ⟨Santana (band), recordLabel, Artista⟩}
	{⟨NickMason, associatedBand, PinkFloyd⟩, ⟨PinkFloyd, recordLabel, Capitol⟩}	⟨RickRubin, recordLabel, AmericanRecordings⟩	{⟨RickRubin, associatedBand, ZZTop⟩, ⟨ZZTop, recordLabel, AmericanRecordings⟩}
		⟨CarlosSantana, recordLabel, Artista⟩	{⟨CarlosSantana, associatedBand, Santana (band)⟩, ⟨Santana (band), recordLabel, Artista⟩}

Figure 2: Explanation for ⟨NickMason, recordLabel, Capitol⟩

not a downside but an helpful feature for these tasks; *query relation awareness*: to focus on relevant facts to the query, when aggregating neighbours.

6 CONCLUSIONS AND EXTENSIONS

We have proposed a solution to the problem of generating explanations for the link prediction tasks on KGs aiming at finding meaningful reasons for which two entities are predicted as linked. This work presented an integrated structural and semantic approach. The solution searches for paths and examples of similar situations that justify the prediction carried out using numerical approaches. We adopted CrossE as a base embedding model to compute predictions, and an integrated algorithm based on semantic similarity measures for providing explanations of the computed predictions. The proposed solution has been experimentally evaluated, demonstrating that the semantics-aware approach is able to provide more meaningful explanations, compared to the baseline.

A natural further empowerment of the proposed framework consists in taking into account additional semantic information in KGs that may be exploited, such as transitivity and symmetry properties of the relationships. The base explanation algorithm could also benefit from a preliminary clustering process whose output would be exploited for the recurring selection of the most similar entities and relationships.

REFERENCES

[1] F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi, and P.F. Patel-Schneider (Eds.). 2007. *The Description Logic Handbook: Theory, Implementation and Applications* (2nd ed.). CUP. https://doi.org/10.1017/CBO9780511711787

[2] R. Bhowmik and G. de Melo. 2020. Explainable Link Prediction for Emerging Entities in Knowledge Graphs. In *Proc. of ISWC 2020*, J. Pan et al. (Eds.). Springer, 39–55. https://doi.org/10.1007/978-3-030-62419-4_3

[3] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko. 2013. Translating Embeddings for Modeling Multi-relational Data. In *Proc. of NIPS 2013*, C. J. C. Burges et al. (Eds.). Curran Associates, Inc., 2787–2795.

[4] S. Chari, O. Seneviratne, D.M. Gruen, M.A. Foreman, A.K. Das, and D.L. McGuinness. 2020. Explanation Ontology: A Model of Explanations for User-Centered AI. In *Proc. of ISWC 2020 (LNCS, Vol. 12507)*, J.Z. Pan et al. (Eds.). Springer, 228–243. https://doi.org/10.1007/978-3-030-62466-8_15

[5] M. Färber, F. Bartscherer, C. Menne, and A. Rettinger. 2017. Linked data quality of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO. *Semantic Web* 9 (2017), 1–53. https://doi.org/10.3233/SW-170275

[6] M.H. Gad-Elrab, D. Stepanova, T.-K. Tran, H. Adel, and G. Weikum. 2020. ExCut: Explainable Embedding-Based Clustering over Knowledge Graphs. In *Proc. of ISWC 2020 (LNCS, Vol. 12506)*, J.Z. Pan et al. (Eds.). Springer, 218–237. https://doi.org/10.1007/978-3-030-62419-4_13

[7] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. de Melo, C. Gutiérrez, J.E. Labra Gayo, S. Kirrane, S. Neumaier, A. Polleres, R. Navigli, A.-C. Ngonga Ngomo, S.M. Rashid, A. Rula, L. Schmelzeisen, J.F. Sequeda, S. Staab, and A. Zimmermann. 2020. Knowledge Graphs. *Comput. Surveys* 54 (2020), 1–37. Issue 4. https://doi.org/10.1145/3447772

[8] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P.S. Yu. 2020. A Survey on Knowledge Graphs: Representation, Acquisition and Applications. *CoRR* (2020). arXiv:2002.00388

[9] F. Lécué, B. Abeloos, J. Anctil, M. Bergeron, D. Dalla-Rosa, S. Corbeil-Letourneau, F. Martet, T. Pommellet, L. Salvan, S. Veilleux, and M. Ziaeefard. 2019. Thales XAI Platform: Adaptable Explanation of Machine Learning Systems - A Knowledge Graphs Perspective. In *Proc. of the ISWC 2019 Satellite Tracks (CEUR Workshop Proceedings, Vol. 2456)*, M.C. Suárez-Figueroa et al. (Eds.). CEUR-WS.org, 315–316. http://ceur-ws.org/Vol-2456/paper85.pdf

[10] F. Lécué and J. Wu. 2018. Semantic Explanations of Predictions. *CoRR* abs/1805.10587 (2018). arXiv:1805.10587

[11] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. Mendes, S. Hellmann, M. Morsey, P. Van Kleef, S. Auer, and C. Bizer. 2014. DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. *Semantic Web* 6 (2014), 167–195. https://doi.org/10.3233/SW-140134

[12] H. Liu, Y. Wu, , and Y. Yang. 2017. Analogical inference for multi-relational embeddings. In *Proc. of ICML 2017*. 2168–2178.

[13] P. Wang, J. Han, C. Li, and R. Pan. 2019. Logic Attention Based Neighborhood Aggregation for Inductive Knowledge Graph Embedding. In *Proc. of AAAI 2019*. AAAI Press, 7152–7159. https://doi.org/10.1609/aaai.v33i01.33017152

[14] W. Zhang, B. Paudel, W. Zhang, A. Bernstein, and H. Chen. 2019. Interaction Embeddings for Prediction and Explanation in Knowledge Graphs. In *WSDM'19*. ACM, 96–104. https://doi.org/10.1145/3289600.3291014