

# Efficient Explanation of Predictions on DL Knowledge Graphs through Enhanced Similarity Search

Claudia d'Amato, Francesco Benedetti and Nicola Fanizzi

Dipartimento di Informatica, Università degli Studi di Bari Aldo Moro, Campus Via Orabona 4, 70215 Bari, Italy

## Abstract

Knowledge Graphs are inherently incomplete, so the relationships that hold between their entities have to be discovered on the go. Generating explanations for such predictions has become a fundamental task in the perspective of *eXplainable AI*. This task boils down to finding meaningful (knowledge-level) reasons for predicting a certain relationship to hold between entities. To date, effective link prediction methods are based on *embedding models*, that represent entities and relationships in a vector space to be learned. Our goal is to extend a semantically enriched approach to generating explanations by exploring graph searches for patterns in similar situations. These patterns justify predictions made through an underlying embedding model, leading to the production of explanations. Since a bottleneck of the method is the search for similar entities and relations in the embedding space, we propose a solution based on the integration of clustering to make this search more efficient. This solution has been empirically evaluated with new experiments, proving the improvement in efficiency while preserving the effectiveness.

## Keywords


embedding models, knowledge graph, explanation, similarity, clustering

## 1. Introduction


*Knowledge Graphs* (KGs) are multi-relational graphs designed to organize and share real-world knowledge where nodes represent entities of interest and edges represent different types of relationships between such entities [1]. We will focus on the latter category of KGs embodied as shared ontologies, ultimately expressed in *Description Logics* (DLs) [2], and rely on reasoning to better exploit the wealth of available underlying knowledge.

Due to the inherent incompleteness and heterogeneity of the sources for large KGs, two of the most compelling basic tasks on them are *link prediction* and *triple classification* that amount, respectively, predicting an unknown component of a triple and assessing the truth of a new or existing triple. For these purposes, lots of numeric-statistical models have been proposed, in particular methods that learn vector representations (*embedding models*) that have been shown to scale even to very large KGs. The downside is that these models are difficult for human experts to interpret and verify. Thus an elusive aspect concerns the *trust* in predictions made

---


 DL 2023: 36th International Workshop on Description Logics, September 2–4, 2023, Rhodes, Greece


 claudia.damato@uniba.it (C. d'Amato); f.benedetti1@studenti.uniba.it (F. Benedetti); nicola.fanizzi@uniba.it (N. Fanizzi)

 <https://www.di.uniba.it/~cdamato> (C. d'Amato); <https://github.com/itsfrank98/> (F. Benedetti);

<https://www.di.uniba.it/~fanizzi> (N. Fanizzi)

 0000-0002-3385-987X (C. d'Amato); 0000-0001-5319-7933 (N. Fanizzi)

 © 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

through such models (e.g., in the context of a KG about drugs, the prediction of a side effect for a given compound): the more complex and accurate the models get, the less explainable become the reasons supporting their predictions. As a consequence, providing explanations for the predicted results has become increasingly important.

Current solutions to the problem of computing explanations (e.g. see [3, 4, 5]) can be distinguished into two main categories [6]: those related to the internal mechanisms of a model, and those that can motivate the output predictions. Specifically, two possible approaches can be identified [7]: *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. *A-posteriori methods* aim at constructing explanations after the model has delivered its predictions; they do not explain the reasons for which the internal mechanism of the model produced a given output, but try to find a suitable explanation based on the observed output and on the model input, i.e., the KG evidence.

We will focus on the latter approach as it allows to adopt link prediction models based on numerical representations of the data that are more scalable with respect to the former, and thereby more suitable for real large-scale KGs and more capable of generating explanations for the predictions made. In fact, there are only a few examples of approaches that are able to explain link predictions with KGs. We will focus on an improved *a-posteriori* method that can provide *semantic-based* explanations for link prediction on KGs. In particular, given a prediction, the goal is to understand why it was made, giving valuable reasons to enable the user to judge the output, understand the motivations, and thus increase confidence in the prediction.

Specifically, moving from the SEMANTICCROSS E explanation approach [8], that is based on eliciting analogous justification patterns (exploiting a semantic similarity measure) to build effective explanations, we propose a more efficient solution that exploits a clustering structure of the embeddings to speed up the search and, consequently, the entire explanation process. This is motivated by the fact that whilst the adoption of the semantic measures employed yielded a greater ability to capture the underlying KG semantics which improved the effectiveness of the process, it also showed noticeable limitations in efficiency which restrains the scalability of the approach to the large dimensions of current KGs. The new solution preserves the handling of semantics that characterizes the approach while making the explanation process more applicable in practice. Indeed, in our comparative study, we show experimentally that while the adoption of clustering results in a significantly more efficient explanation process, the quality of the explanations generated using either variant of the method is substantially comparable.

The rest of this paper is organized as follows. The functional foundations of our solution are recalled in §2. The proposed explanation process is presented in §3, while in §4 we illustrate the comparative experimental study in terms of efficiency and effectiveness of the variants. §5 summarizes the conclusions and delineates a possible further extension.

## 2. Basics: Embedding Models for KGs and Explanations

Embedding models, a popular solution to link prediction problems on KGs are briefly recalled, then we focus on the task of *explaining* predictions and specifically on SEMANTICCROSS E, a method that can exploit schema-level semantics which was targeted for further enhancements.

## 2.1. Knowledge Graphs and Embedding Models

A *Knowledge Graph* [1] can be defined as a data structure denoted with  $\mathcal{K}(\mathcal{E}, \mathcal{R})$  involving a set of the nodes  $\mathcal{E}$ , or *entities*, and a set of arcs  $\mathcal{R}$ , i.e. the *relationships* which connect entities. Adopting the RDF data model, a KG can be regarded 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  $\langle s, p, o \rangle$  with:  $s \in \mathcal{U} \cup \mathcal{B}$ ,  $p \in \mathcal{U}$ , and  $o \in \mathcal{U} \cup \mathcal{B} \cup \mathcal{L}$  [7].

Several models have been proposed for embedding KGs in low-dimensional vector spaces [9], that learn a unique *distributed representation* (or *embedding*) for each entity and predicate therein considering different types of components (e.g. point-wise, complex, discrete, Gaussian, manifold). Here we adopt an embedding vector-space based on  $\mathbb{R}$ .

Regardless of the learning procedure, these models share a fundamental characteristic: given  $\mathcal{K}(\mathcal{E}, \mathcal{R})$ , they represent each entity  $x \in \mathcal{E}$  as a continuous *embedding vector*  $\mathbf{e}_x \in \mathbb{R}^k$ , where the dimension  $k \in \mathbb{N}$  is a user-defined hyperparameter. Similarly, each predicate  $p \in \mathcal{R}$  is associated to a *scoring function*  $f_p : \mathbb{R}^k \times \mathbb{R}^k \rightarrow \mathbb{R}$ . For each pair of entities  $s, o \in \mathcal{E}$ , the score  $f_p(\mathbf{e}_s, \mathbf{e}_o)$  measures the *confidence* in the fact 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* (i.e. one that takes into account differences with their sign).

## 2.2. Computing Explanations of Link Predictions

SEMANTICCROSS<sub>E</sub>, a method for computing explanations for link predictions on KGs, was devised to better exploit the underlying semantics [8]. Given a predicted triple, the formulation of an explanation is based on searching for paths of relations that link its subject to the object. We are not interested in the orientation of the edges. This search is driven by the similarities among both relation embeddings and entity embeddings, and makes structural comparisons with other paths in the KG: the reliability of an explanation is reinforced by the presence of similar paths (referred to as *support*).

**Example 1** ([8]). Given the predicted triple  $\langle X, \text{fatherOf}, M \rangle$  a suitable explanation may be given by the chain pattern (path)  $\langle X, \text{hasWife}, Z \rangle, \langle Z, \text{hasChild}, M \rangle$  that is supported by an analogous situation, the occurrence of a similar triple  $\langle Y, \text{fatherOf}, X \rangle$  that is known to be true (not just a hypothesis) for which the explanation is  $\langle Y, \text{hasWife}, S \rangle, \langle S, \text{hasChild}, X \rangle$ .

Given a predicted triple  $\langle h, r, t \rangle$  for the query  $\langle h, r, ? \rangle$ , the main idea consists in looking for (short) *paths* from  $h$  to  $t$ , and provide them as explanations (see [8] for a longer example on the application of this method). This search aims at finding analogous situations that can support the explanation (similarity will be discussed in the next §3.1): this requires a structural comparison between paths (patterns) that *support* the explanation. The process is summarized in Algo. 1, whose steps are 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  $r$  (line 6);
2. Search for the set  $\mathcal{P}(h, t)$  of (all) paths between  $h$  and  $t$  (lines 7-8):

---

**Algorithm 1** Explanation and support of a prediction
 

---

```

1 input  $\mathcal{K}(\mathcal{E}, \mathcal{R})$ : knowledge graph;  $\langle h, r, t \rangle$ : predicted triple;
2    $k_e, k_r$ : max numbers of similar entities, relations;
3 output explanations of the predicted triple and and their supports
4    $Expl \leftarrow \emptyset$ ; /* working set of explanations */
5    $Supp \leftarrow \emptyset$  /* support set */
6   /* Find the most similar relations to  $r$  */
7   Select  $\mathcal{S}_r \subseteq \mathcal{R}$ , with  $|\mathcal{S}_r| = k_r$ 
8   for each path-type  $i$  do
9     Compute the path-set  $\mathcal{P}_i = \{P \text{ of type } i \mid \forall \text{triple} \in P: \mathcal{K} \vdash \text{triple}\}$ 
10    /* Find the most similar entities to  $h$  */
11    Select  $\mathcal{S}_h \subseteq \mathcal{E}$ , with  $|\mathcal{S}_h| = k_e$ 
12    for each path  $P \in \bigcup_i \mathcal{P}_i$  do
13      if ( $P \in \mathcal{P}(h_s, t_s)$  s.t.  $\forall T \in P: \mathcal{K} \vdash T \wedge (h_s \in \mathcal{S}_h)$ )
14         $Expl \leftarrow Expl \cup P$ 
15         $Supp \leftarrow Supp \cup \{\langle h_s, P, t_s \rangle, \langle h_s, r, t_s \rangle\}$ 
16    return  $Expl, Supp$ 

```

---

- a maximum length is fixed to limit the search space; considering only lengths 1 and 2, then six types of patterns are possible:  $P_1 = \{\langle h, r_s, t \rangle\}$ ,  $P_2 = \{\langle t, r_s, h \rangle\}$ ,  $P_3 = \{\langle e', r_s, h \rangle, \langle e', r', t \rangle\}$ ,  $P_4 = \{\langle e', r_s, h \rangle, \langle t, r', e' \rangle\}$ ,  $P_5 = \{\langle h, r_s, e' \rangle, \langle e', r', t \rangle\}$ ,  $P_6 = \{\langle h, r_s, e' \rangle, \langle t, r', e' \rangle\}$ , where  $r_s$  is a relationship similar to  $r$ ,  $r'$  stands for any other relationship, and  $e'$  for 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  $h$  (line 9);
    - 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.  $\mathcal{P}(h_s, t_s)$  denotes the set of paths involving the entities found in this step.
  4. Search for similar structures to support the explanation (lines 10-13):
    - if there exists a path  $P$  from  $h_s$  to  $t_s$  (i.e. similar to  $\langle h_s, r, t_s \rangle$  determined at the previous step) whose triples  $T$  can be derived from  $\mathcal{K}$ , then  $P$  is an *explanation* for  $\langle h, r, t \rangle$  (this is denoted with a special triple:  $\langle h_s, P, t_s \rangle$ );
    - triples in  $P$  describe an analogous situation w.r.t.  $\langle h, r, t \rangle$  involving similar entities and relationships: then the support is extended with  $P$  which joins a similar head to a similar tail through a relation that is similar to  $r$ , i.e. analogously to  $\langle h_s, r, t_s \rangle$ .

In the original formulation of CROSS-E [3], the analogy between pairs of entities or relationships was assessed using the Euclidean distance, applied to their embeddings. However it is well known that this metric may be inadequate when the assumption of *isotropy* for the underlying vector-space does not hold. Considering the crucial role of similarity in the explanation process, other measures have been considered in SEMANTIC-CROSS-E.

### 3. Extending the Explanation Method

Moving from the initial ideas behind CROSSER, various extensions were foreseen and finally incorporated in the ultimate implementation of the explanation method, namely:

- further measures that are capable of capturing the underlying semantics of the KG to better direct the process towards more accurate explanations;
- the usage of such measures within clustering algorithms is intended to produce groupings of embeddings that can be exploited to accelerate the key task of similarity search.

#### 3.1. Extended Measure

The *Semantic Cosine* similarity measure was motivated by the purpose of enhancing the explanation process by better exploiting the available knowledge. Compared to the Euclidean norm, it can capture additional and/or complementary information in the resulting embedding space. The semantics of the KGs, particularly when represented by rich representation languages, such as RDF-S and OWL, is often disregarded. Being able to exploit the KG semantics may lead to generate more accurate explanations for link predictions.

Hence, the *semantic Cosine* measure was introduced [8] to better assess the similarity of two vector embeddings on the ground of additional semantic information. Such information is captured by a *score function* defined to this purpose.

We consider the set  $\mathcal{C}$  of the classes occurring in  $\mathcal{K}(\mathcal{E}, \mathcal{R})$ , and the functions  $Cl: \mathcal{E} \rightarrow \mathcal{C}$ ,  $Do: \mathcal{R} \rightarrow \mathcal{C}$ ,  $Ra: \mathcal{R} \rightarrow \mathcal{C}$  that return, resp., the conjunction of the classes an entity belongs to, the domain and range of a relation. We also resort to the *retrieval* service [2], denoted in the following as function  $ret: \mathcal{C} \rightarrow 2^{\mathcal{E}}$ , a reasoning service that returns the entities in  $\mathcal{K}(\mathcal{E}, \mathcal{R})$  that can be proven to belong to a given class.

The *semantic Score* function for pairs of entities  $e, e' \in \mathcal{E}$  is defined by:

$$sScore(e, e') = \frac{|ret[Cl(e) \sqcap Cl(e')]|}{|ret[Cl(e) \sqcup Cl(e')]|}. \quad (1)$$

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

$$sScore(r, r') = \frac{|ret[Do(r) \sqcap Do(r')]|}{|ret[Do(r) \sqcup Do(r')]|} + \frac{|ret[Ra(r) \sqcap Ra(r')]|}{|ret[Ra(r) \sqcup Ra(r')]|}. \quad (2)$$

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

$$semCos_{\alpha, \beta}(e, e') = \alpha \cdot sScore(e, e') + \beta \cdot sim_{cos}(\mathbf{e}, \mathbf{e}') \quad (3)$$

where  $\mathbf{e}$  represents the respective embedding vector and  $\alpha, \beta \in [0, 1]$  s.t.  $\alpha + \beta = 1$ . In the case of relations  $r, r' \in \mathcal{R}$  the measure is defined analogously.

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 was proposed [8].

### 3.2. Learning Clustering Structures to Enhance Similarity Search

Clustering has been shown to provide an added value to the explanation process [10]. Moving from the base approach delineated in the previous section, the selection of the most similar entities and relations can be optimized by grouping their embeddings in clusters.

Thus, by prepending a preliminary phase to find good clustering structures over the embedding vectors it is possible to guide the search for the most similar relations/entities by considering only relation/entity embeddings within a single cluster. To this purpose, vector similarity measures can be employed for an efficient computation. Of course with large numbers of clusters, they will tend to be less crowded hence some semantically similar embeddings may be missing from the targeted cluster. This may cause finding sub-optimal neighbors hence limiting the effectiveness of the explanation method. Hence a trade-off has to be made between number of clusters and quality of the resulting explanations.

Two simple unsupervised algorithms, namely *k*-MEANS and AGGLOMERATIVE clustering, were considered to find a given number of groupings exploiting the similarity measures employed also in the explanation process. More complex tree-structures borrowed from NEAREST NEIGHBORS methods may represent a natural extension to the approach we adopt in this work.

## 4. Experiments

The objective of the experiments was twofold: first, to analyze the quality of the explanations generated for link predictions, and second, to assess how the choice of the similarity measure affected the quality of the explanations produced by the prediction method. The study also aimed at exploring how clustering structures could help speed up similarity search.

Clustering techniques were exploited in order to speed up the similarity search in the presented explanation methods while preserving their effectiveness. The adopted similarity measure varied in accordance to the metric employed in the explanation phase. The clustering phase preceded the explanation process. This phase consisted in finding, for each embedding, the (three) closest neighbors. Then the explanations were produced for the predicted triples, and the outcomes of the process were evaluated via the same metrics adopted in the former experiment. Code and datasets employed are publicly available<sup>1</sup>.

**Evaluation Metrics** To assess the quality of the predictions, we adopted the metrics employed in the original evaluation of CROSS E [3] also in combination with other measures [8], namely:

- *Elapsed Time* per experiment to assess the gain in efficiency resulting from employing pre-computed clustering structures in the key-task of similarity search;

---

<sup>1</sup>[https://github.com/itsfrank98/SemanticCrossE/tree/clustering\\_2](https://github.com/itsfrank98/SemanticCrossE/tree/clustering_2)



- *Recall*: proportion of predicted triples for which the model can generate explanations:  $\#EPs/\#Ps$ , where  $\#EPs$  counts the predictions with *at least one explanation* and  $\#Ps$  stands for the total number of predictions;
  - conforming to the mentioned previous experiments, only short explanation paths (maximal length 2) were considered, which limits the number of possible explanations, maintaining a greater focus on their quality and brevity;
  - note that the number of explanations generated per predicted triple is not taken into account: the recall is not affected by this number;
- *Average Support*: number of explanations generated, on average, for each prediction:  $\frac{1}{|\mathcal{P}|} \sum_{T \in \mathcal{P}} |Expl(T)|$ , counting the number of explanations  $|Expl(T)|$  generated for each predicted triple  $T$ , where  $\mathcal{P}$  is the set of predictions for a query:  $\{T \mid T = \langle h, r, t \rangle \text{ predicted for } \langle h, r, ? \rangle\}$ 
  - essentially the measure quantifies the reliability of the explanations: the larger the support the more reliable and credible the prediction;
  - each of the six types of explanation path was evaluated in terms of the adopted metrics: this allows a quantitative comparison of the different settings.

**Knowledge Graphs** For the sake of comparison, the same KGs adopted in the mentioned original evaluations were considered [3]. We recall that, since they lack of significant semantic information actually taken into account by SEMANTICROSSE, we considered *DBpedia15k* as additional dataset for performing further tests in order to stress on the possible utility of the semantic component or not. Details on the adopted KGs are summarized below:

- **WN18** contains 40,943 entities and 18 relations. It was extracted from *WordNet*<sup>2</sup>, where linguistic relations (e.g., *hypernymy*, etc.) between *synsets/entities* are represented;
- **FB15k-237** contains 14,541 entities and 237 relationships. It is a subset of the original dataset *FB15k* containing relation triples and textual mentions of *Freebase*<sup>3</sup> entity pairs.
- **DBpedia15k** contains 12,862 entities and 279 relations with 180,000 triples extracted from *DBpedia* (see [11]).

**Parameters Setup** For the comparison, as the same algorithm was used in the preliminary link prediction phase, the settings used for the original evaluation in [3] (and also in [8]) were maintained in the new experiments.

Specifically, in [3] it was suggested 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 resulting noise. In the aim of generating explanations of good quality, small values have been considered:  $k = j = 3$ . As regards the semantic score function, the considered settings for the weights was  $\alpha = 0.2$  and  $\beta = 0.8$ ; the motivation is that cosine similarity applies to the embeddings computed by CROSS-E, incorporating more latent information learned, while the semantic measure enforces the similarity complementarily.

<sup>2</sup><https://wordnet.princeton.edu/>

<sup>3</sup><https://web.archive.org/web/20100228011242/http://www.freebase.com/>

**Table 1**

Parameter settings per KG (see [8] for their exact role in the algorithm)

parameters	KGs		
	<i>WN18</i>	<i>FB5K-237</i>	<i>DBpedia15k</i>
negative examples $n$	50	50	50
learning rate $\eta$	0.01	0.01	0.01
embeddings dim. $d$	100	100	100
regularization $\lambda$	$10^{-4}$	$10^{-5}$	$10^{-5}$
batch size $B$	2048	4000	4000

Finally, also for the settings of the link prediction parameters were the same used in [8]. The TENSORFLOW implementation of the model exploited an ADAM optimizer and a *dropout* of 0.5 was applied to the similarity operator (max. number of iterations: 500). Further settings with parameter values were selected differently on a per-dataset basis and are reported in Table 1. The parameter values are the same used for the experimental evaluations in [8].

As regards the clustering methods considered in the preliminary phase, finding an optimal number of clusters  $K$  may be done beforehand via cross-validation or during their execution. We preliminarily tested both techniques on various values of  $K$ . In the following we present results for  $K = 8, 10, 15$  as larger numbers have been shown to worsen the performance of the explanation methods, as expected. More complete results are made available in the repository.

Finally, as regards the choice of the similarity measure for the explanation algorithm (and also by the clustering procedure), the settings involved in the evaluation will be indicated as *orig.*, *cos*, *acos*: the first corresponds to adopting the Euclidean distance, as in the original approach [3]; in the second setting the cosine similarity is adopted, and the third involves the approximate semantic Cosine measure [8]. This was tested only on *DBpedia15k* since it was the only KG with semantic annotations. In this case the *Manhattan distance* has been considered for clustering as a faster replacement for the semantic cosine similarity.

In the following the results of the experiments carried out are summarized and discussed. We first recall those collected by testing the models with embeddings not grouped in clusters by similarity [8]. Then we illustrate and discuss the results of new experiments where clustering techniques have been exploited for grouping the embeddings into different numbers of clusters. In the experiments described in [8] the explanation algorithm was executed on the predicted triples of each dataset. For each measure, the explanations were produced considering limited portions (2% and 5%) of the total amount of predictions. This is because very low ranked predicted triples might turn out to be incorrect and, as a consequence, the corresponding explanations would turn out to be useless.

**Efficiency Gain** The results reported in Table 2 show the advantage, in terms of elapsed time, achieved by prepending the construction of a clustering structure to speed up the retrieval of neighbouring embeddings, a crucial task for the explanation process.

Various values for the number of clusters were experimented, varying also the similarity measure which was also adopted by the two clustering methods. Specifically they were run using Euclidean distance, cosine similarity and the Manhattan distance as a replacement for the approximated Cosine similarity additionally considered only in the experiments with *DBpedia15k*. As expected, with the usage of clustering, the total elapsed time reduced along



**Table 2**

Elapsed time (hh:mm:ss): comparing the original setting without clustering to the ones exploiting the clusters produced by  $K$ -MEANS and AGGLOMERATIVE for various values of  $K$

dataset	measure	Orig.	$K$ -MEANS CLUSTERING			AGGLOMERATIVE		
			8	10	15	8	10	15
WN18	orig.	2:19:46	0:24:45	0:18:53	0:21:37	1:09:47	0:43:44	0:43:01
	cos	3:40:47	1:17:22	0:44:50	0:33:38	0:30:19	0:25:16	0:17:51
DBpedia15k	orig.	0:13:42	0:02:23	0:01:49	0:00:57	0:01:55	0:01:40	0:01:20
	cos	0:36:37	0:04:26	0:03:36	0:01:43	0:10:29	0:10:39	0:09:18
	alt.	1:33:13	0:24:44	0:14:15	0:09:36	0:32:36	0:27:39	0:14:03
FB15k	orig.	0:18:54	0:03:11	0:02:15	0:01:27	0:02:22	0:02:07	0:01:30
	cos	0:33:22	0:06:16	0:03:53	0:02:22	0:16:47	0:13:57	0:11:06

**Table 3**

Results of the experiments with the various settings and measures (no clustering exploited for similarity search): recall and average support per explanation by path type [8]

dataset	measure	%	recall	avg.support					
				1	2	3	4	5	6
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
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

with the number of clusters that were considered as the number of required comparisons decays. This happens because each embedding is compared only against those in the same cluster.

Of course there is a sort of trade-off to be made with effectiveness of the explanation process. However, considering the effectiveness of the explanation process discussed in the following, it is possible to conclude that clusterings can yield an appreciable gain in efficiency to the overall explanation process with no significant loss in effectiveness.

**Effectiveness** For comparative purposes, the outcomes of the experiments in terms of the metrics aiming at assessing the effectiveness of the plain explanation process, with no clustering involved, are recalled in Table 3. Tables 4 and 5 report the outcomes the explanation process evaluation when preceded by the preliminary clustering phase, involving, respectively, the two mentioned algorithms and the different similarity measures. It is worthwhile to notice that the fixed number of most similar relationships and most similar entities considered in the generation of the explanations (see discussion in §4) limits the computational costs but also the recall. Differently from the discussion on time, for brevity, the outcomes reported to these tables are related to experiments with a fixed minimal number of clusters, as larger numbers tended to worsen the performance of the overall explanation process, as expected.

**Table 4**Results (explanation) for the extended method with clusters computed by  $K$ -MEANS

dataset	measure	%	recall	avg.support						
				1	2	3	4	5	6	
WN18	orig.	2%	0.0010	0.0	2.39	3.24	2.91	1.93	1.91	
		5%	0.0004	0.0	2.39	3.17	2.85	1.89	1.86	
	cos	2%	0.0028	1.00	4.10	2.12	2.00	1.84	1.81	
		5%	0.0011	1.00	4.10	2.10	1.98	1.83	1.79	
	DBpedia15k	orig.	2%	0.0021	1.32	1.22	3.27	1.38	1.60	2.54
			5%	0.0021	1.32	1.22	3.27	1.38	1.60	2.54
cos		2%	0.0019	1.24	1.17	3.13	1.38	1.62	2.62	
		5%	0.0009	1.24	1.17	3.08	1.38	1.61	2.52	
acos		2%	0.0019	1.39	1.21	3.61	1.53	1.93	2.69	
		5%	0.0009	1.38	1.21	3.61	1.53	1.91	2.57	
FB15k	orig.	2%	0.0594	16.69	5.98	206.59	377.97	185.39	210.45	
		5%	0.0340	15.14	5.81	193.41	366.44	172.41	187.08	
	cos	2%	0.0593	16.69	5.98	206.59	377.97	185.38	210.45	
		5%	0.0339	15.14	5.82	193.41	366.44	172.42	187.08	

$K$ -MEANS. Table 4 reports the results with a clustering structure produced by  $K$ -MEANS with  $K = 8$ . Considering the outcomes of the experiments with *WN18*, it can be noticed that in the case where the Euclidean distance was used the recall is much inferior with respect the original method with no clustering (see Table 3). Conversely, the results in terms of average support are more comparable, with some cases in favor of the extended method. In the experiments where the cosine similarity was used the outcomes in terms of both recall and support are only slightly inferior w.r.t. those of the original method. In the experiments with *DBpedia15k*, the outcomes observed when the Euclidean distance was adopted show no noticeable difference. Hence the gap previously observed in the experiments with *WN18* may be an exception, probably due to the specific embeddings produced in that case. Indeed no noticeable difference can be appreciated also in the cases involving  $K$ -MEANS clustering in combination with the other two similarity measures. Similar considerations can be made for the outcomes in terms of avg. support which presented even some cases (path types) where the extended method performed slightly better. Examining the outcomes of experiments with *FB15k*, there is a noticeable improvement observed brought by the employment of the extended method: in almost all cases the recall doubled while the gain is even larger in terms of average support in all of the cases considered. A possible motivation is that, especially with this KG, querying for more explanations is possible as more relevant embeddings are considered after exploiting the clustering structure.

*Agglomerative Clustering*. Table 5 presents the results obtained with the same KGs, employing AGGLOMERATIVE in its extended version targeting  $K = 10$  clusters. Considering the outcomes of the experiments with *WN18*, we observe that the recall measures are quite comparable (with even a tiny improvement in the 5% sub-case where the cosine measure is adopted). Similar considerations can be made for the outcomes in terms of average support. In the case of *DBpedia15k*, one can observe that recall for the various sub-cases was almost similar. Analogous considerations can be made for the outcomes in terms of average support where, again, small improvements were observed for some specific path-types. Finally, in the experiments with *FB15k*, we observed more significant differences of the performance of the method when the clustering is adopted. Namely the outcomes show a higher recall for all sub-cases. This is even

**Table 5**

Results (explanation) for the extended method with clusters computed by AGGLOMERATIVE

dataset	measure	%	recall	avg.support						
				1	2	3	4	5	6	
WN18	orig.	2%	0.0025	1.00	4.29	2.31	2.30	2.08	2.05	
		5%	0.0010	1.00	4.29	2.26	2.23	2.02	2.05	
	cos	2%	0.0027	1.00	4.22	2.20	2.13	1.97	2.00	
		5%	0.0011	1.00	4.22	2.16	2.10	1.93	1.96	
	DBpedia15k	orig.	2%	0.0023	1.33	1.21	2.89	1.45	1.86	2.61
			5%	0.0011	1.33	1.21	2.86	1.44	1.84	2.52
cos		2%	0.0019	1.19	1.25	3.02	1.29	1.56	2.44	
		5%	0.0009	1.19	1.12	2.97	1.28	1.55	2.35	
acos		2%	0.0019	1.38	1.21	3.61	1.53	1.92	2.69	
		5%	0.0009	1.37	1.21	3.61	1.53	1.89	2.57	
FB15k	orig.	2%	0.0599	16.79	6.01	211.22	379.76	188.33	210.39	
		5%	0.0341	15.49	5.84	198.01	370.54	175.18	187.41	
	cos	2%	0.0674	19.04	7.72	208.39	285.69	209.59	195.57	
		5%	0.0396	17.06	7.45	184.81	253.52	186.21	166.07	

more apparent in terms of the average support outcomes where, again, major improvements were recorded for almost all path-types. Regarding these improvements, the considerations made in the analogous experiments with the different clustering method still apply.

## 5. Conclusion and Further Extensions

We have proposed a solution to the problem of generating explanations for link predictions on KGs. This work presented an integrated structural and semantic approach based on searching for paths and examples of similar structures that justify the predictions made exploiting an embedding model. CROSS-E was adopted as a base embedding model to compute predictions, and an integrated algorithm based on semantic similarity measures was employed for producing explanations of the predictions. This procedure was further extended with a preliminary clustering phase aimed at grouping similar embeddings to improve the efficiency of the recurring key-task of retrieving neighbors. The solution enhanced with this extension have been experimentally evaluated, demonstrating that the semantics-aware approach is able to provide more meaningful explanations, compared to the baseline and that the preliminary clustering phase can speed up the overall process without degrading its effectiveness.

A natural further enhancement of the proposed framework will consist in taking into account additional semantic information in KGs that can be exploited, such as transitivity and symmetry properties of the relationships. Another extension regarding the clustering phase can come from adopting classical methods from the related literature that can autonomously decide the number of clusters, such as nonparametric methods, e.g. those that produce hierarchical structures, such as *ball-trees* or *kD-trees* based on the similarity of the entities.

## Acknowledgments

This work was partially supported by the project FAIR - Future AI Research (PE00000013), spoke 6 - Symbiotic AI, under the NRRP MUR program funded by the NextGenerationEU.

## References

- [1] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. de Melo, C. Gutiérrez, J. Labra Gayo, S. Kirrane, S. Neumaier, A. Polleres, R. Navigli, A. Ngonga Ngomo, S. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, A. Zimmermann, Knowledge graphs, *ACM Computing Surveys* 54 (2021) 1–37. doi:10.1145/3447772.
- [2] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, P. Patel-Schneider (Eds.), *The Description Logic Handbook: Theory, Implementation and Applications*, 2nd ed., CUP, 2007. doi:10.1017/CBO9780511711787.
- [3] W. Zhang, B. Paudel, W. Zhang, A. Bernstein, H. Chen, Interaction embeddings for prediction and explanation in knowledge graphs, in: *Proceedings of WSDM 2019*, ACM, 2019, pp. 96–104. doi:10.1145/3289600.3291014.
- [4] P. Wang, J. Han, C. Li, R. Pan, Logic attention based neighborhood aggregation for inductive knowledge graph embedding, in: *Proceedings of AAAI 2019*, AAAI Press, 2019, pp. 7152–7159. doi:10.1609/aaai.v33i01.33017152.
- [5] R. Bhowmik, G. de Melo, Explainable link prediction for emerging entities in knowledge graphs, in: J. Pan, et al. (Eds.), *Proceedings of ISWC 2020*, volume 12506 of *LNCS*, Springer, 2020, pp. 39–55. doi:10.1007/978-3-030-62419-4\_3.
- [6] F. Lécué, J. Wu, Semantic explanations of predictions, 2018. arXiv:1805.10587.
- [7] M. Färber, F. Bartscherer, C. Menne, A. Rettinger, Linked data quality of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO, *Semantic Web 9 (2017)* 1–53. doi:10.3233/SW-170275.
- [8] C. d'Amato, P. Masella, N. Fanizzi, An approach based on semantic similarity to explaining link predictions on knowledge graphs, in: J. He, et al. (Eds.), *Proceedings of WI-IAT 2021*, ACM, 2021, pp. 170–177. doi:10.1145/3486622.3493956.
- [9] S. Ji, S. Pan, E. Cambria, P. Marttinen, P. S. Yu, A survey on knowledge graphs: Representation, acquisition, and applications, *IEEE Transactions on Neural Networks and Learning Systems* 33 (2022) 494–514. doi:10.1109/TNNLS.2021.3070843.
- [10] M. Gad-Elrab, D. Stepanova, T. Tran, H. Adel, G. Weikum, Excut: Explainable embedding-based clustering over knowledge graphs, in: J. Pan, et al. (Eds.), *Proceedings of ISWC 2020*, volume 12506 of *LNCS*, Springer, 2020, pp. 218–237. doi:10.1007/978-3-030-62419-4\_13.
- [11] Z. Sun, W. Hu, C. Li, Cross-lingual entity alignment via joint attribute-preserving embedding, in: C. d'Amato, et al. (Eds.), *Proceedings of ISWC 2017, Part I*, volume 10587 of *LNCS*, Springer, 2017, pp. 628–644. doi:10.1007/978-3-319-68288-4\_37.