Novel Perspectives and Applications of Knowledge Graph Embeddings: From Link Prediction to Risk Assessment and Explainability

Hegler C. Tissot[1[0000−0003−4635−451X]]

University of Pennsylvania
Philadelphia, PA, USA
hegler@seas.upenn.edu
hextrato.com

Abstract. Knowledge graph representation is an important embedding technology that supports a variety of machine learning related applications. By learning the distributed representation of multi-relational data, knowledge embedding models are supposed to efficiently deal with the semantic relatedness of their constituents. However, failing in the fundamental task of creating an appropriate form to represent knowledge harms any attempt of designing subsequent machine learning tasks. Several knowledge embedding methods have been proposed in the last decade. Although there is a consensus on the idea that enhanced approaches are more efficient, more complex projections in the hyperspace that indeed favor link prediction (or knowledge graph completion) can result in a loss of semantic similarity. We propose a new evaluation task that aims at performing risk assessment on domain-specific categorized multi-relational datasets, designed as a classification problem based on the resulting embeddings. We assess the quality of embedding representations based on the synergy of the resulting clusters of target subjects. We show that more sophisticated embedding approaches do not necessarily favor embedding quality, and the traditional link prediction validation protocol is a weak metric to measure the quality of embedding representation. Finally, we present insights about using the synergy analysis to provide risk assessment explainability based on the probability distribution of feature-value pairs within embedded clusters.

Keywords: Knowledge graphs · Link prediction · Risk assessment.

1 Introduction

Decision support system applications based on knowledge graphs (KGs) have been reported in different scenarios, such as entity linking [20], drug-to-drug similarity measurements [22], and recommender systems [32]. Graph-based knowledge representation uses a set of symbolic *(head, relation, tail)* triplets (or facts) to represent the various entities (nodes) and their relationships (edges) from a

multi-relational dataset. Each entity represents one of various types of an abstract concept of the world and each relation is a predicate that represents a fact involving two entities. There is a consensus that the heterogeneous nature of the data sources, where facts are usually extracted from to create a KG, makes the later typically inaccurate. Although containing a huge number of triplets, most open-domain KGs are usually taken as incomplete, covering only a small subset of the true domain knowledge they are supposed to represent.

Knowledge embedding representation (KER) approaches have been proposed as an effective way to map the symbolic entities and relations into a continuous vector space, enforcing the embedding compatibility while preserving semantic information. Embedding vectors are easier to manipulate than the original symbolic entities and relations, and their popularity has led to the development of refined techniques to increase their quality [2]. KER aims to efficiently measure semantic correlations in knowledge bases by projecting entities and relations into a dense low-dimensional space, significantly improving performance on knowledge inference and alleviating sparsity issues, and it is usually presented as an efficient tool to complete knowledge bases (Link Prediction – LP) without requiring extra knowledge [33]. LP aims at predicting new relationships between entities by automatically recovering missing facts based on the observed ones. However, to our knowledge, there has been not much effort on evaluating the embedding representation quality resulting from KG embedding approaches. A preliminary study aims to contrast the effectiveness of hyperparameter choices when using the resulting embedding representation in subsequent machine learning classification tasks, instead of just relying on KG completion [5].

In this work, we propose a new evaluation task that aims at performing risk assessment on domain-specific categorized multi-relational datasets. We redesign the KER evaluation as a risk assessment task to validate the ability of knowledge embedding approaches to retain the semantic relatedness of KG constituents, i.e., quantifying the degree to which two components are associated with each other. We measure the synergy of feature-value pairs within clusters of resulting embeddings in order evaluate the ability of KER approaches on capturing the semantic similarity among target subjects. We provide evidence that simpler approaches perform better than more sophisticated embedding formulations when targeting embedding quality rather than trying to improve knowledge completion. Finally, we present insights on how to use the synergy analysis over the resulting embeddings to provide risk assessment explainability based on the probability distribution of feature-value pairs within the resulting embeddings.

2 Knowledge Embedding Representation

Multi-relational data is usually presented in the form of a KG. Entities (nodes) and relations (edges) provide a structured representation of the knowledge about a specific domain, and a reasoning ability that can be used for inference. In a KG, structured information is encoded in the form of triples *(h, r, t)* (also known as *subject, predicate, object*), where h and t are the *head* and *tail* enti-

ties and r represents the *relation* between h and t. Although containing a huge number of triplets, most open-domain KGs are taken as incomplete, covering only a small subset of the true knowledge that they are supposed to represent, whereas in domain-specific KGs, incompleteness results from missing values and cardinality-related inconsistencies that are usually produced by automatic information extraction processes from unstructured data sources (e.g., clinical notes).

Learning knowledge embedding representation enables a range of tasks including KG completion [4, 26], entity classification [19] and relation extraction [27]. Within this technique, entities and relations are embedded onto a low-dimensional vector space to capture the semantic relatedness behind observed facts and operate on the latent feature representation of the triple constituents. However, embedding quality is an aspect that has not been much explored alongside the KER evaluation process.

Translational embedding approaches use relatively simple assumptions to achieve accurate and scalable results on embedding KGs. Overall, these models try to learn vectors for each constituent *(h, r, t)*, so that every relation r is a translation between h and t in the embedding space, and the pair of embedded entities h and t can be approximately connected by r with low error. Embedding methods operate on the latent feature representation of the constituents and on their semantic relatedness, by defining a distinct relation-based scoring function $f_r(h,t)$ to measure the plausibility of the triplet *(h, r, t)*. $f_r(h,t)$ implies a transformation on the pair of entities which characterizes the relation r. The final embedding representation is learned using an algorithm that optimizes a margin-based objective function or ranking criterion over a training set.

TransE [4] is a baseline translational embedding approach known by its flaws at dealing with *one-to-many*, *many-to-one* and *many-to-many* relations when applied to open-domain data [31]. Other methods extended TransE by varying the way they assign different representations to each entity and each relation to achieve better link prediction performance. For example, TransH [26], TransR [15], and TransD [11] use projection matrices to pre-project each h into a relation-specific vector space. Therefore, they use separate distinct vector spaces to embed entities and relations, each entity can have distinct distributed representations when involved in different relations, which allows entities to play different roles in different relations. However, this makes it hard to compare the similarity of two distinct entities without taking relations into account.

Other KER approaches have been proposed, with a common goal to improve low-dimensional KG representation targeting specific evaluation tasks. However, they differ in the theoretical problem concerned or the solution approach as reflected in their scoring functions, including adapted scoring functions to allow more flexible translations (e.g., TransM [7] and TransA [28]), Gaussian embeddings to model semantic uncertainty (e.g., KG2E [10] and TransG [29]), tensor factorization (RESCAL [18]), compositional vector representation (HolE [17]), complex spaces (ComplEx [25]), transitive relation embeddings (TRE [34]), and neural neighborhood-aware embeddings (LENA [12]). Although these models achieve great results on the benchmark open-domain datasets, their implemen-

tations are scattered and unsystematic, and their codes for model validation and reproducibility are often time-consuming, making them difficulty to be used in further development, and adopting them for real-world applications [9].

In domain-specific KGs, multi-relational data can be categorized, i.e., each entity is presented with its corresponding type and relations are also restricted by domain and range. Type-based constraints can support latent variable models, by integrating prior knowledge about entity and relation types, significantly improving these models in the link prediction tasks, especially when a low model complexity is enforced [13]. In categorized KGs, each entity e is associated with a category (or type) $c \in \mathcal{T}$, and each triple is presented in the form *(c_h:h, r, c_t:t)*, where c_h and c_t represent the types of h and t. For example, in the triple `(Patient:P01, hasGender, Gender:male)`, the relation *hasGender* is constrained by the domain *Patient* and the range *Gender*.

There are multiple suggested ways to apply type-based constraints in training latent variable models: (a) entities belonging to the same semantic type can be placed close together in the embedding space with the use of geometric constraints such as manifold regularization [8]; (b) entities can be projected onto type-specific vector spaces, analogous to the relation-specific projections [30]; (c) type information can also be used to measure semantic similarity, which has been used to calculate prior probabilities in a Bayesian learning process, alongside creating a set of multiple semantic vectors for each entity [16]; and (d) type-independent hyperspaces can be used to accommodate entities that belong to the same type, constraining the selection of negative samples and favoring LP accuracy by restricting the set of entities ranked during evaluation [24].

The LP evaluation task has originally emerged from the idea that KGs are usually incomplete. Several embedding approaches have been proposed for predicting the missing links in the KGs [32]. During the evaluation process, a typical question answering task aims at completing a triple (h, r, t) with h or t missing, by predicting t given $(h, r, ?)$ or predicting h given $(?, r, t)$, where '?' denotes the missing element. Rather than giving the best answer, LP mimics a recommendation system by ranking the plausibility of a set of candidate entities based on a similarity score. Overall results are usually presented by reporting: a) Mean Rank (MR); b) Mean Reciprocal Rank (MRR) of correct entities; and c) the proportion of correct entities in top-N ranked entities (Hits@N, with N usually equals 10). A LP model should achieve lower MR or higher MRR and Hits@N. MRR calculates the average reciprocal rank of all the entities (relations), and it is less sensitive to outliers comparatively with MR.

3 Materials and Methods

In opposite to a general open-domain KG that contains common sense information, a vertical KG is based on more complex domain-specific categorized multi-relational data, mostly suitable for specific industry applications. Whereas open-domain KGs are wider in terms of breadth, deeper and sparser, domain-specific graphs usually have low level of granularity (higher level of detail) and

they are more dense [14]. In addition, the former can be composed of multiple independent sub-graphs, whereas this is usually hard to observe in the latter due to the intra-relational structured data sources they are extracted from.

We aim to use KER learned for categorized multi-relational data in a decision support pipeline. Therefore, instead of trying to complete a KG, we look at the risk assessment task, targeting the probability of a given entity h having (r) a label t that makes a triple in form (h,r,t) true when the resulting probability exceeds a threshold l ($P(h, r, t) > l$), where l is a tuning hyperparameter. Thus, to evaluate the quality of embedding representation, we designed risk prediction as a classification task based on the distribution probability of nearby neighbors in the entity vector space having the target label.

3.1 Datasets

Focused on domain-specific data, we conducted experiments on three publicly available datasets (Mushroom, Epilepsy and CHSI) and on private dataset (Pregnancy) from the clinical domain – data controllers have granted us permission to use and perform analysis on a de-identified version of this dataset. A description for each dataset and the corresponding pre-processing tasks are given below.[1] Overall dataset statistics are shown in Table 1.

Table 1: Benchmark datasets statistics. 'Classes' represents the number of target classification labels (independent target labels are used in *Pregnancy*, whereas target labels are mutually exclusive in *Epilepsy* and *CHSI*. 'Subjects' is the number of entities in the target type (in all datasets consistently represented by the type of *head* entity). 'Triples' in the test set are given by randomly selecting subjects (not triples) from the original KG, except for *Pregnancy*, in which test set was split based on the year each pregnancy started (2010-2014 for training, and 2015 for test); subjects in the test set are never seen in the training set.

		# Subjects				# Triples	
Datasets	**Classes**	**Train**	**Test**	**# Entities**	**# Relations**	**Train**	**Test**
Mushroom	1	7,537	879	8,485	22	163,593	19,079
Epilepsy	5	10,354	1,146	27,473	178	1,843,012	203,988
CHSI	10	2,828	313	7,034	679	1,059,838	117,720
Pregnancy	3	20,200	4,676	31,472	99	1,270,529	288,270

Mushroom[2] is a publicly available dataset deposited on the UCI Machine Learning Repository that classifies hypothetical samples corresponding to distinct species of mushrooms into edible or poisonous based on 22 categorical attributes describing shape, surface, color, odor, gill, stalk, veil, ring, population and habitat characteristics. This dataset was originally used to perform logical

[1] https://github.com/hextrato/KRAL-benchmark
[2] https://archive.ics.uci.edu/ml/datasets/mushroom

rules and further considered as a relatively easy task for machine learning approaches, some reaching accuracy of 100%. Triples are presented in the form of *many-to-one* relations only, and it was used in previous LP evaluation tasks for categorized multi-relational data [24, 5].

Epilepsy[3] (Epileptic Seizure Recognition) is also available on the UCI Machine Learning Repository. It presents 178 continuous variables collected for a recording of brain activity for 23.6 seconds, aiming to assign each of the 11,500 instances (500 individuals $\times$ 23 seconds) to one of five possible classes (1–5), in which subjects in class 1 are taken as having epileptic seizure, and subjects falling in classes 2, 3, 4, and 5 are those who did not have epileptic seizure. Although most authors have done binary classification, namely class 1 (Epileptic seizure) against the others, we kept the risk assessment task focused on all the five independent target classes. All continuous variables have values varying from -1885 to +2047 and they were heuristically normalized into positive and negative ranges of 30 values ($[-30, 0[, [0, 30[, [30, 60[, [60, 90[, ...$) in order to simplify the KG symbolic representation.

CHSI (Community Health Status Indicators)[4] is a dataset designed to support combating obesity, heart disease, and cancer as a component of the Community Health Data Initiative. It provides key health indicators, comprising over 200 measures for 3141 United States counties that enable a more comprehensive understanding on the behavioral factors such as obesity, tobacco use, diet, physical activity, alcohol and drug use, sexual behavior and others substantially contribute to deaths, like the ones due to heart disease and cancer. There is not any specific target label in this dataset. Thereat, for evaluation purposes only, in the context of this work, we designed the evaluation task as a prediction of average life expectancy (ALE) in each county, with target labels varying from 70 to 79+-years-old (10 possible classes).

Pregnancy combines structured and unstructured data extracted from an Electronic Health Record (EHR) system regarding 24,876 pregnancies occurring from 2010 to 2015, comprising demographic and clinical history before (e.g., history of medication, allergies, infections, and other clinical conditions) and during pregnancy (e.g., prescriptions, procedures, and diagnoses). Although the dataset was originally created to perform risk assessment of miscarriage, we added two additional target risk labels: *Hyperemesis gravidarum*, and high risk pregnancy. This is a dataset predominantly composed of *many-to-many* relations (83.7%), expect for the *one-to-many* demographic relations. Data from 2010 to 2014 was used to learn the embedding representation, and risk analysis is performed ever the 2015's patient set (test set).

3.2 Method Outline

Although embedding representation approaches are traditionally evaluated using the LP task, we believe LP does not directly impact quality of entity embed-

[3] https://archive.ics.uci.edu/ml/datasets/Epileptic+Seizure+Recognition

[4] https://healthdata.gov/dataset/community-health-status-indicators-chsi-combat-obesity-heart-disease-and-cancer

dings, and result models are biased, only favoring the LP task accuracy instead. Therefore, our evaluation protocol was designed accordingly to the following phases (further implementation details are given in the subsequent subsection):

1. Firstly, we learn embedding representation for the training set. Triples corresponding to the target classification labels are NOT used during the embedding process to avoid biasing further clustering analysis. MRR score was initially used during training to select the best model.
2. Alternatively, we added a cluster synergy score (KSyn, see 'Implementation Details' section), performed when evaluation embedding representation in conjunction with MRR. KSyn aims to evaluate the ability of each model to capture entity similarities among subjects in each target cluster. We used K-Nearest Neighbor (KNN) algorithm [1] and we tested multiple numbers of clusters (K) to find the best radius to be taken into account when performing synergy analysis.
3. Resulting entity and relation embeddings from the training set are frozen and the test triples are appended to the KG. A second short embedding round is performed to properly accommodate the test subjects in the vector space (only entities from the test set not yet seen during training have their embedding representation learned during this phase).
4. Finally, we extract the vector representation of each subject entity (split into training and test subjects). For each subject in the test set, we calculate the probability distribution of its neighbors (training subjects) regarding each target classification labels. The probability of each test subject belonging to any of the target classes is recorded and subsequently used to perform accuracy analysis, looking for the best threshold to optimize ROC (AUPRC due the unbalanced nature of target labels) and F scores.

3.3 Implementation Details

We used an embedding approach proposed for domain-specific categorized multi-relational datasets [24] that utilizes type-dependent vector spaces as a basis for all our experiments. Additionally, we added a feature to activate a relation-based projection that mimics other enhanced translational approaches, such as TransH and TransR.[5]

Type-dependent vector spaces restrict domain and range for each relation and are effective to optimize the selection of negative samples during training instead of random sampling from the whole set of possible entities, lessening the probability of constructing a poor-quality negative triple, and being more efficient and sped up, with reduced impact from uninformative constituents. In addition, only entities belonging to the same type are scored for comparison in the loss function during the validation step.

Formally, given a training set S of categorized triples $(c_h{:}h,\ r,\ c_t{:}t)$, embedding vectors for entities and relations are learned, so that each categorized

[5] https://github.com/hextrato/KRAL

entity *c:e* is represented by an embedding vector $\mathbf{e_c} \in \mathbb{R}^K$, and each relation r is represented by an embedding vector $\mathbf{r} \in \mathbb{R}^K$. A score function f_r (Equation 1) represents a L2-norm dissimilarity, such that the score $f_r(h_{c_h}, t_{c_t})$ of a plausible typed triple *(c_h:h,r,c_t:t)* is smaller than the score $f_r(h'_{c_h}, t'_{c_t})$ of an implausible typed triple *(c_h:h',r,c_t:t')*. Then, the optimal KER is learned by minimizing a margin-based (γ) loss function $\mathcal{L}$ (Equation 2) adapted from TransE, where γ is the margin parameter, $\mathcal{S}$ is the set of correct triples, $\mathcal{S}'$ is the set of incorrect triples *(c_h:h',r,c_t:t)* $\cup$ *(c_h:h,r,c_t:t')*, and $[x]_+ = max(0, x)$.

$$f_r(h_{c_h}, t_{c_t}) = \|h_{c_h} + r - t_{c_t}\|_{l_2} \tag{1}$$

$$\mathcal{L} = \sum_{\substack{(c_h:h,r,c_t:t)\in\mathcal{S} \\ (c_h:h',r,c_t:t')\in\mathcal{S}'}} [\gamma + f_r(h_{c_h}, t_{c_t}) - f_r(h'_{c_h}, t'_{c_t})]_+ \tag{2}$$

Alternatively, we used a relation-based projection matrix $M_r \in \mathbb{R}^{K\times K}$ to mimic translational approaches that attempt to enhance TransE (Equation 3).

$$f'_r(h_{c_h}, t_{c_t}) = \|M_r \times h_{c_h} + r - t_{c_t}\|_{l_2} \tag{3}$$

A regularization constraint is used during training to restrict the magnitude of embedding vectors and prevent loss-minimization by inappropriately increasing the embedding norms for each entity e, usually given by $|\mathbf{e}| \leq q$, where q is given by Equation 4 [24]. Although there is no proven evidence that q improves embedding performance over a fixed magnitude threshold ($|\mathbf{e}| \leq 1$), we found this adaptive constraint favors the embedding vectors to spread the range of latent values in $[-1, +1]$ for each dimension.

$$q = \max(1, \frac{\sqrt{k}}{2}) \tag{4}$$

The primordial assumption when dealing with any kind of ML model is the ability of such resulting model on generalizing. Embedding models are weak regarding to this aspect. Previous KER approaches usually perform a single learning round, including training, validation, and test sets simultaneously, the latter ones supposedly for testing the generalization ability. However, validation and test sets are required to be designed with entities and relations that appear at least once in the training set.

Instead, we consider the test set should not be seen during the initial training to avoid biasing the resulting model. Therefore, our training protocol introduces substantial changes comparatively to the usual routine for learning KER. We start by using triples from the KG training set only to perform up to 500 training cycles when learning the initial vector representation for entities and relations – during initial experiments we consistently observed models achieved best performance in early training cycles (≈200-300) for categorized datasets. We tested multiple number of dimensions $k \in \{8, 16, 32, 64, 128, 256\}$ and we used an adaptive adjustable learning rate (η) and learning margin (γ) to monitor performance.

η is made smaller (from 0.1 to 0.01) once the performance of the model plateaus, whereas γ is made bigger ($\frac{q}{8}$ to $\frac{2\times q}{3}$). We tuned hyperparameter by selecting the best performance on a 10-folder cross validation performance based on a combination of MRR and KSyn (see below), and we report the results of each model on the corresponding test set.[6] Subsequently, we mimic TransE-like enhanced models. We pick-up the best model during training and we perform additional 500 training steps, now using a relation-based projection matrix (Equation 3).

Cluster Synergy (KSyn) When learning KER, we performed a validation step every 20 training cycles looking at simultaneously improving of two metrics: (a) MRR and cluster synergy (KSyn). From the latter we expect to capture the ability of a given embedding representation to approximate similar entities. The resulting embedding representation for the subject entities are clustered using KNN algorithm with multiple variations of $K \in \{16, 32, 48, 64, 96, 128\}$ – the bigger K is, the smaller the average cluster radius become. For each cluster, we look at each pair *(r, c_t:t)* that correspond to a feature value for a given subject c_h:h. If the probability of *(r, c_t:t)* occurring in a cluster u (i.e. P_u*(r, c_t:t)*) is bigger than the overall probability of the same feature-value pair occurring in the entire training dataset *P(r, c_t:t)*, we consider the difference P_u*(r, c_t:t)* - *P(r, c_t:t)* as the contribution of the feature-value pair to synergy of cluster u. KSyn of a given cluster u is the average of all positive contributions from each possible feature-value pair within that cluster, whereas KSyn of the resulting embedding model is the average KSyn of all clusters. The average radius m from the best cluster setup is saved to be further used in the final classification task.

Differently from previous KER approaches that use training and test triples simultaneously during training, we consider our approach is more realistic when adding the test set only in a subsequent learning step. In addition, subjects from the test set are totally distinct from those used during training. We perform a second embedding training round aiming to accommodate the test subjects in the vector space. Only entities from the test set not yet seen during training have their embedding representation learned during this phase, whereas embedding representation for entities used during the first training phase are kept frozen.

Finally, we use the embedding representation from all training and test subjects to perform risk assessment as a classification task. The best average cluster radius m learned from the KSyn validation is used as a radius threshold when calculating the probabilistic distribution of target labels in each embedding cluster (each one centered by a test subject). For each target label l, we calculate the probability of l for each test subject s. Thus, each test subject is taken as the center of a cluster u_s with radius m. Then we use the resulting embeddings from the training subject neighbors within a maximum L2-norm distance m from s, and we calculate the probability of label l happening in cluster u_s. The probability scores in the range [0,1] of each label l for each test subjects are then analyzed regarding AUPRC and F-score to find the best classification threshold.

[6] We used a Linux x86 64-bits Intel® Xeon® CPU E5-2630 v4 @ 2.20GHz as a computing infrastructure for our experiments.

4 Results and Discussion

We presents our results in four distinct perspectives: (a) we contrast low- *vs.* high-dimensional spaces and we show how the number of dimensions can influence the ability of embedding approaches to capture the semantic relatedness of graph constituents; (b) we present our findings on how LP and cluster quality metrics can be complementary when simultaneously used to both model generalization and embedding quality; (c) we provide evidence that simpler approaches perform better than more sophisticated embedding formulations when targeting embedding quality rather than trying to improve link prediction; and (d) we demonstrate how cluster synergy analysis can be used to provide explainability for a resulting embedding model.

Low- *vs.* high-dimensional spaces In higher dimensional spaces, a density estimator can misbehave when there is no smooth low-dimensional manifold capturing the distribution [3]. Although higher dimensional spaces can provide more space to accommodate entities, this does not necessarily favor the similarity of nearby entities, as evidenced in [5]. We tested the effect of both lower and higher k-dimensional spaces ($8 \leq k \leq 256$), and we present the final classification results (AUPRC) on the test set for each dataset in each k-dimensional space in Table 2. None of the datasets was able to consistently improve classification performance alongside increasing the number of dimensions in the vector space. Oppositely, the number of required k dimensions that best fit the embedding representation seems to be somehow related to the complexity (shape and size) of the dataset and classification tasks. For example, 'Mushroom' and 'Epilepsy' are the datasets devoid of any *many-to-many* relations, thus requiring lower embedding dimensionality.

Table 2: Average AUPRC scores for each dataset on each k-dimensional space on the risk assessment task - average of scores resulting from each classification label - best score in bold for each dataset.

	Datasets			
k-dim	**Mushroom**	**Epilepsy**	**CHSI**	**Pregnancy**
8	0.9991	**0.5475**	0.3592	0.1506
16	**0.9993**	0.5254	0.3799	0.1628
32	0.9979	0.5195	0.3852	0.1492
64	0.9984	0.4911	0.3844	**0.1651**
128	0.9993	0.4499	**0.4019**	0.1586
256	0.9992	0.4100	0.3978	0.1575

Link prediction *vs.* embedding quality MRR and Hits@N are correlated metrics traditionally used as embedding evaluation scores. However, the more

MRR can be improved the better embedding quality it is not necessarily entailed. This becomes more evident when we look at the way embedding approaches try to improve overall model accuracy by adding relation-based projections and how they are affected by hyperparameters (learning rate η and learning margin γ). Figure 1 shows how MRR and the proposed KSyn scores evolve along the training process. Each chart presents the two-phase 500-cycle learning process, each phase following Equations 1 and 3 (in Section 3.3) respectively:

(1) In the first learning phase, η varies from 0.1 to 0.01 in the first 300 cycles, and it is kept fixed at 0.01 so on, whereas γ is fixed at $(q/8)$ along the first 200 learning cycles, when it is then progressively increased up to $(q \times 1.5)$ in the cycle 500 (see Equation 4). Although embedding quality (KSyn) is not necessarily worsen during the last 200 learning cycles, higher values for γ can negatively affect MRR, which seems consistent to results found in [6].

(2) In the second learning phase, a relation-based projection matrix is added to the best model (chosen by selecting the best combination of MRR and KSyn) for additional 500 learning cycles. There is a considerable improvement in the MRR metric in the first cycles (< 50), when no further improvement is shown, and models become stable regarding MRR. However, the MRR improvement implies decay in the KSyn score. We believe relation-based projection matrices do not favor embedding quality and make embedding approaches biased by the traditional LP evaluation protocol. One possible reason for this outcome is that non-similar entities separated by opposite hyper-hemispheres (opposite sides of any dimension within the hyperspace), even if they are very close to each other, can be pre-projected to opposite directions by the relation matrix before having the relation vector added to their latent composition.

(a) Epilepsy (b) CHSI

Fig. 1: MRR *vs* KSyn on two benchmark datasets – although MRR slightly improves when a relation-base projection matrix is added in the second learning phase, there is a decay in the KSyn score indicating loss of embedding quality.

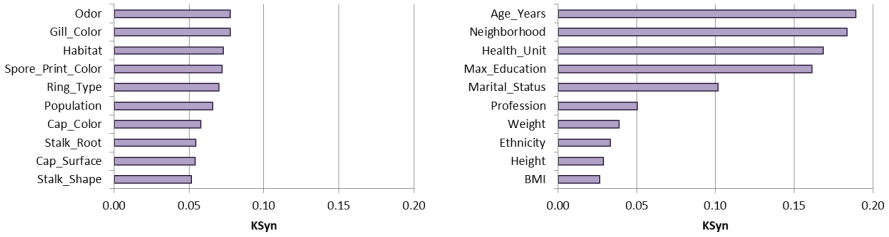
Simple *vs.* complex embedding approaches In Table 3, we present the F1-scores for the risk assessment problem designed as a classification task. We compare three distinct learning approaches: (a) firstly, embedding models are learned based on the MRR metric only; (b) then we used a combination of MRR and KSyn metrics to perform evaluation and select the best model during training; (c) finally, we added a relation-based projection matrix on the top of the best model. Although MRR does not directly reflect the resulting embedding quality and synergy for similar clustered entities, it is still a good evaluation metric to be used alongside the process of learning KER. However, when we pairwise MRR with a way of enforcing embedding synergy (MRR+KSyn Linear), the resulting models are more suitable for a classification tasks that directly relying on the embedding representation and the probabilistic distribution of target labels within the entity neighbors in the vector space. Finally, although previous approaches have been exploring more complex ways of learning KER (MRR+KSyn Matrix), we found strong evidence that the LP diverts attention from the fact the overall embedding representation model is expected to carry on a semantically relatedness among similar entities, favoring the knowledge completion task only, thus badly performing when evaluated on tasks the rely on the embedding quality, such as risk assessment.

Table 3: Resulting F1-scores for the risk assessment task regarding each embedding learning validation strategy.

	Learning Validation Strategy		
Dataset	**MRR (only) (Linear)**	**MRR + KSyn (Linear)**	**MRR + KSyn (Matrix)**
Mushroom	0.9986	**0.9988**	0.8679
Epilepsy	0.5799	**0.5828**	0.4973
CHSI	0.3718	**0.4794**	0.3644
Pregnancy	0.2964	**0.3053**	0.2293

Model explainability Decision trees are known by its capability to efficiently deal with large, complicated datasets without imposing a complicated parametric structure, and break down a complex classification process into a collection of simpler decisions, facilitating feature selection, and thus providing a solution that is easier to interpret [21, 23]. However, they are model-oriented and target specific classification labels. We used the cluster synergy analysis to provide explainability for a resulting embedding model: (a) first, we can provide a feature-relevance analysis that is performed based on the resulting model regarding a specific test set, i.e., the way features are ranked is sensible to the test subjects (Figure 2); and (b) to each test subject we can perform feature-relevance analysis and provide the individual explainability to each test case. Figure 3 compares feature-value relevance from two samples of mushroom (poisonous *vs.* edible) in

the test set. Relevant feature-value pairs differ between each other sample, and also differ comparatively to the resulting feature relevance regarding the overall test set, when comparing Figures 2 and 3.



(a) Mushroom (b) Pregnancy

Fig. 2: Feature relevance analysis for the resulting embedding model regarding the test set over the training set.

5 Conclusions

While deep learning methods have led to many breakthroughs in practical ML applications, there is still a lack on how to develop systems that can 'understand' and 'explain' the decisions they make. A critical step in achieving ML explainability is to design knowledge meaning representations, and KG embeddings are a potential approach towards that direction. We introduce novel perspectives of using KG embeddings techniques to support subsequent ML applications in this sense and we review some hyperparameter tuning effects: (a) higher dimensional spaces do not necessarily improve embedding performance and quality, but they are affected by learning rate and margin; (b) traditional KER evaluation protocol is biased by the LP task, i.e., embedding approaches are expected to provide representation models that express the semantically similarity among similar nearby entities; instead, whereas trying to improve LP accuracy, enhanced models fail on satisfying the intra-cluster semantic similarity of entity vectors; and finally, (c) we introduce a cluster synergy analysis to support model explainability that enables tracking input entities back into the training gold standard sets and understanding the relations between these entities – from the resulting knowledge embedding representation, cluster synergy analysis provides the overall feature-relevance for a test set regarding the training samples, and the ability to individually perform feature-value relevance to each test subject.

We plan to expand current experiments by looking at alternative ways of dealing with test cases, evaluating further embedding constraints (e.g., regularization, disjoint sets, and taxonomies), and using temporal-based datasets to draw the high-level picture on how risk changes and how it is timely affected.

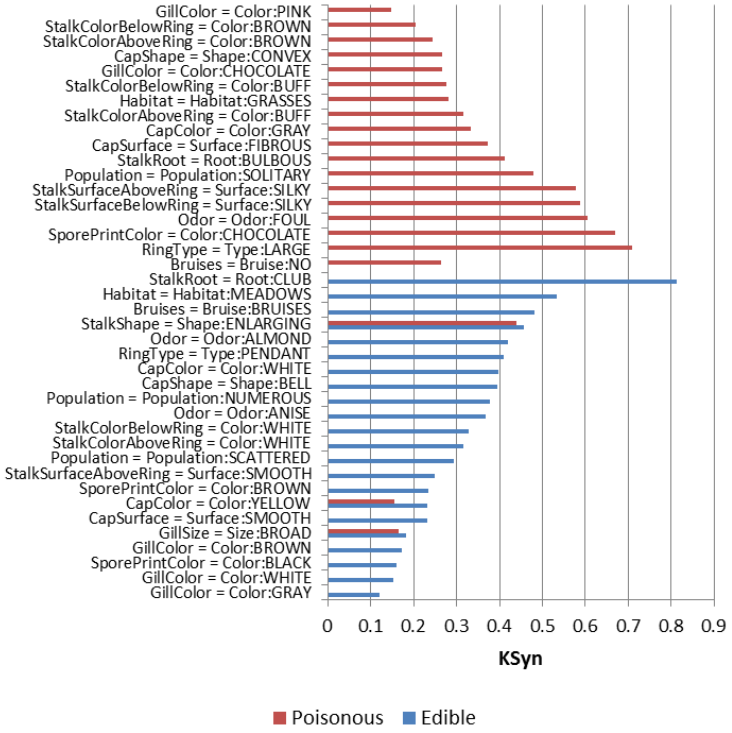


Fig. 3: Risk assessment explainability analysis – KSyn of each pair (feature, value) for two mushroom samples; relevant feature-values pair are different comparing two mushroom samples (poisonous *vs.* edible).

References

1. Altman, N.S.: An introduction to kernel and nearest-neighbor nonparametric regression. The American Statistician **46**(3), 175–185 (1992)
2. Ayala, D., Borrego, A., Hernández, I., Rivero, C.R., Ruiz, D.: Aynec: All you need for evaluating completion techniques in knowledge graphs. In: Hitzler, P., Fernández, M., Janowicz, K., Zaveri, A., Gray, A.J., Lopez, V., Haller, A., Hammar, K. (eds.) The Semantic Web. pp. 397–411. Springer International Publishing, Cham (2019)
3. Bengio, Y., Larochelle, H., Vincent, P.: Non-local manifold parzen windows. In: Advances in Neural Information Processing Systems 18 (NIPS'05). MIT Press, Cambridge, MA (2005)
4. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 26, pp. 2787–2795. Curran Associates, Inc. (2013)
5. Chung, M.W.H., Liu, J., Tissot, H.: Clinical knowledge graph embedding representation bridging the gap between electronic health records and prediction models.

In: Wani, M.A., Khoshgoftaar, T.M., Wang, D., Wang, H., Seliya, N. (eds.) 18th IEEE International Conference On Machine Learning And Applications, ICMLA 2019, Boca Raton, FL, USA, December 16-19, 2019. pp. 1448–1453. IEEE (2019)
6. Chung, M.W.H., Tissot, H.: Evaluating the effectiveness of margin parameter when learning knowledge embedding representation for domain-specific multi-relational categorized data. In: StarAI 2020 - Ninth International Workshop on Statistical Relational AI. AAAI (Feb 2020)
7. Fan, M., Zhou, Q., Chang, E., Zheng, T.F.: Transition-based knowledge graph embedding with relational mapping properties. In: Proceedings of the 28th Pacific Asia Conference on Language, Information and Computing (2014)
8. Guo, S., Wang, Q., Wang, B., Wang, L., Guo, L.: Sse: Semantically smooth embedding for knowledge graphs. IEEE Transactions on Knowledge and Data Engineering **29**(4), 884–897 (4 2017)
9. Han, X., Cao, S., Lv, X., Lin, Y., Liu, Z., Sun, M., Li, J.: OpenKE: An open toolkit for knowledge embedding. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. pp. 139–144. Association for Computational Linguistics (2018)
10. He, S., Liu, K., Ji, G., Zhao, J.: Learning to represent knowledge graphs with gaussian embedding. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. pp. 623–632. CIKM '15, ACM, New York, NY, USA (2015)
11. Ji, G., He, S., Xu, L., Liu, K., Zhao, J.: Knowledge graph embedding via dynamic mapping matrix. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers. pp. 687–696 (2015)
12. Kong, F., Zhang, R., Mao, Y., Deng, T.: Lena: Locality-expanded neural embedding for knowledge base completion. Proceedings of the AAAI Conference on Artificial Intelligence **33**(01), 2895–2902 (Jul 2019)
13. Krompaß, D., Baier, S., Tresp, V.: Type-constrained representation learning in knowledge graphs. In: Proceedings of the 13th International Semantic Web Conference (ISWC) (2015)
14. Lin, J., Zhao, Y., Huang, W., Liu, C., Pu, H.: Domain knowledge graph-based research progress of knowledge representation. Neural Computing and Applications (Jun 2020)
15. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence. pp. 2181–2187. AAAI'15, AAAI Press (2015)
16. Ma, S., Ding, J., Jia, W., Wang, K., Guo, M.: TransT: Type-based multiple embedding representations for knowledge graph completion. In: The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (2017)
17. Nickel, M., Rosasco, L., Poggio, T.: Holographic embeddings of knowledge graphs. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence. pp. 1955–1961. AAAI'16, AAAI Press (2016)
18. Nickel, M., Tresp, V., Kriegel, H.P.: A three-way model for collective learning on multi-relational data. In: Proceedings of the 28th International Conference on International Conference on Machine Learning. pp. 809–816. ICML'11, Omnipress, USA (2011)

19. Nickel, M., Tresp, V., Kriegel, H.P.: Factorizing yago: Scalable machine learning for linked data. In: Proceedings of the 21st International Conference on World Wide Web. pp. 271–280. WWW '12, ACM, New York, NY, USA (2012)
20. Rotmensch, M., Halpern, Y., Tlimat, A., Horng, S., Sontag, D.: Learning a health knowledge graph from electronic medical records. Scientific Reports **7**(1), 5994 (Jul 2017)
21. Safavian, S.R., Landgrebe, D.: A survey of decision tree classifier methodology. IEEE Transactions on Systems, Man, and Cybernetics **21**(3), 660–674 (1991)
22. Shen, Y., Yuan, K., Dai, J., Tang, B., Yang, M., Lei, K.: Kgdds: A system for drug-drug similarity measure in therapeutic substitution based on knowledge graph curation. Journal of Medical Systems **43**(4), 92 (Mar 2019)
23. Song, Y.Y., Lu, Y.: Decision tree methods: applications for classification and prediction. Shanghai archives of psychiatry **27**(2), 130–135 (Apr 2015)
24. Tissot, H.: Using ontology-based constraints to improve accuracy on learning domain-specific entity and relationship embedding representation for knowledge resolution. In: Proceedings of the 10th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, IC3K 2018, Volume 1: KDIR, Seville, Spain, September 18-20, 2018. pp. 70–79 (2018)
25. Trouillon, T., Welbl, J., Riedel, S., Gaussier, E., Bouchard, G.: Complex embeddings for simple link prediction. In: Proceedings of the 34 Annual International Conference on Machine Learning (ICML) (2016)
26. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Brodley, C.E., Stone, P. (eds.) Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence. pp. 1112–1119. AAAI Press (2014)
27. Weston, J., Bordes, A., Yakhnenko, O., Usunier, N.: Connecting language and knowledge bases with embedding models for relation extraction. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. pp. 1366–1371. Association for Computational Linguistics, Seattle, Washington, USA (Oct 2013)
28. Xiao, H., Huang, M., Hao, Y., Zhu, X.: TransA: An adaptive approach for knowledge graph embedding. CoRR (2015)
29. Xiao, H., Huang, M., Zhu, X.: TransG: A generative model for knowledge graph embedding. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 2316–2325. Association for Computational Linguistics (2016)
30. Xie, R., Liu, Z., Sun, M.: Representation learning of knowledge graphs with hierarchical types. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence. pp. 2965–2971. IJCAI'16, AAAI Press (2016)
31. Xiong, S., Huang, W., Duan, P.: Knowledge graph embedding via relation paths and dynamic mapping matrix. In: Woo, C., Lu, J., Li, Z., Ling, T.W., Li, G., Lee, M.L. (eds.) Advances in Conceptual Modeling. pp. 106–118. Springer International Publishing, Cham (2018)
32. Zhang, Y., Wang, J., Luo, J.: Knowledge graph embedding based collaborative filtering. IEEE Access **8**, 134553–134562 (2020)
33. Zhiyuan, L., Maosong, S., Yankai, L., Ruobing, X.: Knowledge representation learning: A review. Journal of Computer Research and Development **53**(2), 247 (2016)
34. Zhou, Z., Liu, S., Xu, G., Zhang, W.: On completing sparse knowledge base with transitive relation embedding. Proceedings of the AAAI Conference on Artificial Intelligence **33**(01), 3125–3132 (Jul 2019)