Faithfully Explaining Predictions of Knowledge Embeddings

Gustavo Padilha Polleti[1], **Fabio Gagliardi Cozman**[1]

[1]Escola Politécnica da Universidade de São Paulo (USP)
São, Paulo – SP – Brazil

{gustavo.polleti,fgcozman}@usp.br

Abstract. *Knowledge embeddings are key ingredients of advanced question-answering and recommender systems. Even though their predictions are accurate, they are rather hard to interpret by human users; interpretability techniques are needed so as to provide meaningful human-friendly explanations for prediction generated by embeddings. We propose a novel model-agnostic method inspired by local surrogate approaches that generates faithful explanations for knowledge embedding predictions.*

1. Introduction

Large-scale Knowledge Graphs (KG), such as Freebase [Bollacker et al. 2008], are often built by automatic knowledge base construction [Nickel et al. 2016]. Automatic KB construction aims at extracting factual information from unstructured or semi-structured textual data typically using natural language processing (NLP) techniques. A KG built automatically, although less dependent on human experts than curated KGs, has usually limited applicability due to missing or incorrect facts. The collaborative construction of KGs may also lead to missing or incorrect data that may be unknown by the volunteers filling the KG. For instance, the place of birth attribute is missing for 71% of all people in Freebase [Murphy et al. 2012], a KG that is constructed collaboratively.

Knowledge base completion has received significant attention [Nickel et al. 2016]; most techniques are based on the assumption that statistical regularities pervade the collection of known facts and, once identified, they can be used to infer new information. While some methods are based on graph feature models, e.g. Subgraph Feature Extraction (SFE) [Gardner and Mitchell 2015] and Path Ranking Algorithm (PRA) [Lao et al. 2011], others are based on knowledge embeddings, e.g. TransE [Bordes et al. 2013]. Knowledge embeddings (KEs) map entities and relations to a real-valued vector space where simple arithmetic operations are used to complete the KG.

Knowledge Embeddings also represent a promising approach for question-answering [Huang et al. 2019] and recommender systems [He et al. 2017]. Even though knowledge embeddings can lead to accurate recommendations or responses, a high degree of interpretability is also needed either for the user to effectively accept a suggestion or to agree with an answer. However, KE results are hard to interpret as they are represented by operations in a real-valued space.

On the other hand, graph feature models, e.g. SFE, even though not as accurate as knowledge embeddings, usually produce highly interpretable explanations for their link predictions [Nickel et al. 2016]. Graph feature models have recently been used generate explanations for KEs, but their explanations have low fidelity [Gusmão et al. 2018]. Other

approaches offer transparency on the underlying embedding but, despite understandable by experts, their explanations are not meaningful for most users [Xie et al. 2017] [Kazemi and Poole 2018].

This paper aims at developing and evaluating a novel method for human-friendly explanations for knowledge embeddings in a model-agnostic and faithful manner. To achieve this goal, we explore a local surrogate approach that has produced promising results in non-relational classifiers, e.g. LIME [Ribeiro et al. 2016]. Our proposal provides insights into the development of local scoped explanation techniques for explainable knowledge embeddings, in contrast to previous efforts in the literature that have focused on global explanations [Gusmão et al. 2018].

The paper is organized as follows. Section 2 and 3 presents some notation and terminology followed by some theoretical foundations about knowledge graphs and interpretability, respectively. Section 4 discusses related works and highlights the research gap. We then propose in Section 5 our explanation method. Finally, we discuss empirical results in Section 6, and offer some concluding remarks in Section 7.

2. Knowledge Graphs

In this paper, we follow loosely the *RDF* notation [W3], focusing on relational databases where a triple representing a fact is written as $\langle h, r, t\rangle$ where h, r and t are, respectively, the *subject (head)*, *predicate (relation)* and *object (tail)*. A knowledge graph $\mathcal{KG}$ consists of the set of all entities $\mathcal{E} = \{e_1, \ldots, e_{N_e}\}$, and the set of relations $\mathcal{R} = \{r_1, \ldots, r_{N_r}\}$, where N_e and N_r represent the number of entities and relations in the KG, respectively. The existence of a triple $x_{h,r,t} = \langle h, r, t\rangle$ is indicated by a random variable $y_{h,r,t} \in \{0, 1\}$.

Several approaches have been developed to address the task of *Knowledge Base Completion (KBC)*. The main assumption behind those methods is that it is possible to predict new facts from a statistical model based on existing facts (triples) [Nickel et al. 2016]. There are two major approaches for KBC: the first one focuses on on observable graph features, while the second one works by converting semantically rich factual information into low-dimensional vector spaces. We now examine both approaches.

2.1. Graph Feature Models

These models aim at predicting new triples by extracting features from the observed graph [Nickel et al. 2016]. For instance, a number of popular techniques based on graph feature models are derived from the *Path Ranking Algorithm (PRA)* [Lao et al. 2011]. In particular the *Subgraph Feature Extraction (SFE)* [Gardner and Mitchell 2015] method has displayed promising performance and computational efficiency. SFE works by performing random walks to extract path patterns connecting two entities, to latter construct a feature matrix to be used as input of a classifier to predict new triples.

We define π_l as a path type composed of a number L of arbitrary relations (edges of the graph) in the form $r_1 - r_2 - \ldots - r_l$. To extract the features for a given triple $\langle e_h, r, e_t\rangle$, SFE operates by searching values for paths π_l. One traditional approach is to search for PRA-styled features or paths π_l connecting head e_h and tail e_t entities, where the existence of each path is the feature value. Another more expressive approach is to search for the tail entity e_π where the head is e_h and π_l is the path connecting them. Therefore, the

feature vector for a given entity e_h is represented by $\phi_h = [e_\pi : \pi \in \Pi_L]$, where Π_L is the set of all possible paths of length L; this approach is known as one-sided path features. The features extracted by SFE can be considered as Horn clauses and assume the form of bodies of weighted rules, that is considered easily interpretable [Nickel et al. 2016].

2.2. Knowledge Embeddings (KEs)

Approaches based on Knowledge Embeddings (KE) now display state-of-the-art performance in KBC. KEs offer expressive representations for multi-relational graphs [Wang et al. 2017]. The main intuition behind KEs is that interactions of latent features can represent actual relationships [Nickel et al. 2016]. Because these interactions can be both diverse and complex, many models have been proposed, each one with distinct characteristics [Bordes et al. 2013] [Wang et al. 2014].

Each model defines a particular scoring function $f_r(h, t \mid \Theta)$ to measure the plausibility of fact $\langle h, r, t \rangle$, where Θ contains the parameters. Embedding models build the latent features through an optimization process that maximizes the total plausibility of all known facts. The training is carried out either under the *open world assumption* (OWA) or *closed world assumption* (CWA) [Nickel et al. 2016]. The OWA considers a missing triple as simply unknown, whilst CWA considers missing facts as negative. In this paper we consider OWA KGs.

3. Interpretability

Several techniques have been developed to help in understanding how complex models, such as Deep Neural Networks and Random Forests, make their predictions. These techniques can be divided in two groups: model-specific and model-agnostic. The first one is bound to specific class of models; these techniques usually have access to internal or structural information about the explainee. On the other hand, the model-agnostic techniques can be applied to any machine learning model, as they consider the explainee as a black-box, i.e. do not make any assumptions about its internal behavior.

A popular model-agnostic approach is the construction of interpretable surrogate models. These surrogate techniques vary in scope: some aim to explain the model of interest as a whole (a *global* or *holistic* approach) while others focus on a single or set of predictions.

3.1. Global Surrogate

The global surrogate technique consists of training an intrinsically interpretable model using the black-box predictions as ground truth, so that the global surrogate mimics the black-box model. If the global surrogate is intrinsically interpretable, explanations about the black-box model can be drawn from it. However, because the black-box model is presumably complex, it is not to be expected that another model (the global surrogate) can mimic its behavior in a faithfully manner while remaining simple and interpretable. Also, if a faithful interpretable model is achieved, one could ask why keep the black-box model itself.

3.2. Local Surrogate

One popular model-agnostic strategy consists of reducing the scope of the surrogate model [Ribeiro et al. 2016]. The interpretable model is then expected to mimic the black-box behavior only partially.

The main intuition behind local surrogates is that an interpretable and simple model should be faithful to a complex model at least locally. For instance, suppose that one intends to use a local surrogate to explain a single prediction of a black-box. First, the input data of interest is perturbed, generating a set of variations of the original data or *"neighborhood"*. This set of data points around the original input are then fed to the black-box model that provides labels. Finally, an interpretable surrogate model is trained considering the data points around the original input and their respective labels given by the black-box.

Formally, local surrogate models with interpretability constraints are defined as follows:

$$explanation(x) = \arg \min_{g \in G} L(f, g, d_x) + \Omega(g). \quad (1)$$

The explanation given for the instance x is the interpretable model $g* \in G$, where G is the set of all possible models that minimizes the loss function L and the complexity constraint Ω. The loss function measures the unfaithfulness of the surrogate model g to the black-box function f considering the neighborhood around instance x limited by the distance parameter d_x. The complexity function Ω balances the trade-off between interpretability and fidelity; it may be for instance a measure of model sparsity.

It is worth noting that if Expression (1) covers the entire training set instead of the neighborhood d_x, the resulting model $g*$ would correspond to a global surrogate.

One of the most popular local surrogate techniques is LIME [Ribeiro et al. 2016]. Roughly speaking, LIME runs a sensitivity test around the instance of interest, then it presents as explanations the most significant features for each label. Even though it is effective in producing explanations virtually for all kinds of data, e.g. tabular, textual or visual data, LIME is limited to non-relational classifiers. LIME explains based on the assumption that features themselves are interpretable, a weak assumption in the context of embeddings. For instance, LIME, when applied to a binary text classifier, produces explanations such as "Word XYZ is significant for the prediction"; similarly, when applied to embeddings, LIME explanations would be such as "Dimension 123 is significant for the prediction" — a sentence that is hard to interpret because the dimensions are latent features and, thus, convey no meaning for users.

The dimensions of an embedding are part of the underlying structure of the model, so explanations should rely only on features from semantic field instead. Multi-relational classifiers, such as knowledge embeddings, require techniques for mapping real-valued latent features to the semantic field in order to produce human-friendly explanations; this issue is addressed in Section 5.

4. Related Work

Many proposals related to the interpretability of knowledge embeddings follow the model-specific approach, e.g. ITransF [Xie et al. 2017], SimplE [Kazemi and Poole 2018] and CrossE [Zhang et al. 2019].

SimplE model claims that each dimension of an entity embedding can be considered a feature and the correspondent element of a relation representation is a measure of how important that feature is to the relation. Even though this characteristic provide a certain degree of transparency, it does not seem to be really interpretable. The

Table 1. Comparative summary of related works

Method	Human-friendly	Model-Agnostic	Faithful
SimplE	No	No	-
ITransF	No	No	-
CrossE	Yes	No	Yes
XKE	Yes	Yes	No

model allows one to include background knowledge into the embeddings, however since its interpretability focuses embedding dimension, it is not possible to drawn meaningful explanations for its predictions.

Similarly to SimplE, ITransF model also deals with interpretability on latent features level. ITransF proposes a sparse attention mechanism to represent shared concepts among relations. For instance, both relations "nominated for" and "honored for" represent a concept of high quality work even as they are distinct. The attention mechanism of ITransF allows the identification of latent features, or concepts, however as these features are given in the embedding level, they are not really interpretable. Back to our example, even though we identify a strong link between the relations "nominated for" and "honored for", it is hard to infer what is the actual concept shared. Despite SimplE and ITransF techniques provide a certain degree of interpretability, as their insights are mostly related to the embedding internal structure and are given in terms of real-valued vector or heat maps, their explanations are only understandable by data scientists or by experts.

On the other hand, the CrossE model exploits a particular type of interaction between relations and entities called *crossover interactions* (CI) to explain embedding predictions in the semantic field instead of the real-valued one. For instance, suppose one has to explain the triple $\langle person\, X, isFatherOf, person\, Z\rangle$. An explanation that supports this triple could be the path $\xrightarrow{\text{hasWife}}\xrightarrow{\text{hasChild}}$ connecting $head$ and $tail$ entities. Despite being highly interpretable [Gardner and Mitchell 2015] and faithful, once these support paths are reconstructed considering the embedding, CrossE method cannot explain all predicted triples; it is also not affected by negative instances.

In contrast to the model-specific approaches described previously in this section, the XKE method [Gusmão et al. 2018] is a model-agnostic. XKE consists in training a global surrogate logistic regression on SFE graph features while using the embedding labels, so that explanations can be drawn from the interpretable classifier. Even though XKE is easy to interpret, it displays relatively low fidelity [Gusmão et al. 2018].

As presented in Table 1, it is possible to identify a research gap regarding explanation methods for knowledge embeddings that are both model-agnostic and faithful.

5. A Proposal for Faithful Model-Agnostic Explanations

Our proposal is a novel model-agnostic explanation method for knowledge embeddings, inspired by the local-surrogate approach as adopted by LIME [Ribeiro et al. 2016]. We address a series of challenges due to the complex nature of embedding techniques and provide a method to effectively produce faithful explanations for link predictions.

Explanations drawn from local surrogates usually are given in form of weighted features, what implies that the feature itself must be meaningful for the user. Even though this is true for most traditional classifiers, in some cases, e.g. knowledge embeddings, the features considered by the model to realize their predictions are too complex to be comprehensible or bear no explicit meaning for the target audience. For instance, while a machine learning practitioners may feel comfortable when analyzing vector dimensions, a layman shall prefer a small list of reasons instead.

Knowledge embeddings map semantic rich input (entities and relations) into numeric representations that carry no meaning for humans, thus, latent features or embedding dimensions are inappropriate for human-friendly explanations. Furthermore, it implies that the features in the explanations need to be different than the features (real-valued vectors) used by the knowledge embedding. To address this issue, we argue that the knowledge embedding itself can be used to extract interpretable representations for entity embeddings, so that we can generate meaningful explanations while remaining faithful to the model. This feature extraction procedure is described below, but first consider some important definitions.

The knowledge embedding, for top-1 tail prediction task, can be defined as a set of black-box classifiers $g_r \in \mathcal{G}$, one for each relation $r \in \mathcal{R}$, where $g_r(h|\Theta)$ returns the tail entity $e_t \in \mathcal{E}$ that gives the greatest plausibility score f_r for the triple $\langle e_h, r, e_t \rangle$. That is,

$$g_r(h|\Theta) = \arg\max_{e_i \in \mathcal{E}} f_r(e_h, e_i \mid \Theta). \tag{2}$$

As there exists a real-valued vector representation for each entity $e_i \in \mathcal{E}$ in the knowledge embedding parameters Θ, i.e. $\exists \hat{e}_i \in \Theta$, $\forall e_i \in \mathcal{E}$, we can define the classifier function g_r so that it takes as input the head entity embedding. That is,

$$g_r(\hat{e}_h) = \arg\max_{\hat{e}_i \in \Theta} f_r(\hat{e}_h, \hat{e}_i). \tag{3}$$

We have defined $g_r(\hat{e}_h)$ as a classifier that takes the head entity embedding $\hat{e}_h$ (or tabular data) and outputs the most plausible tail entity (or label), thus g_r and a tabular data traditional classifier are alike.

Example 1 *To illustrate our definitions, consider the toy example where we realize the tail prediction for the triple* $\langle dom\_pedro\_ii, religion, ? \rangle$. *Let us define:*

$$\mathbb{T} = [f_{religion}(e_{pedro}, e_i \mid \Theta), e_i \in \mathcal{E}].$$

$$sort\_desc(\mathbb{T}) = \begin{bmatrix} catholic \\ protestant \\ \vdots \\ e_m \end{bmatrix}, \quad g_{religion}(pedro|\Theta) = catholic. \tag{4}$$

The list $\mathbb{T}$ *represents plausibility score calculated for all entities. Thus, to discover the most plausible candidates for Dom Pedro II's religion, we sort* $\mathbb{T}$ *in descending order and identify that he is most presumably catholic, then protestant and so on. Our function* g_r *returns only the top 1 ranked entity, in this case catholic.*

At this point, we should be able to train a local surrogate to g_r. However, as its input is given in term of latent features, i.e. embedding dimensions, we still cannot extract meaningful explanations. Thus, to proceed we need to answer the following questions:

1. **Q1**: What should be considered an interpretable representation for entities embeddings?
2. **Q2**: How to extract these interpretable representations from the real-valued vector space?

To answer the first question we take graph feature models as alternatives [Gardner and Mitchell 2015]. Even though we consider that other feature types could be used, e.g. PRA-style features, in this work we opted for one-sided path features due to its simplicity, and leave such exploration for future works. As described in Section 2.1, the interpretable representation provided by one-sided path features for an certain entity e_h is $\phi_h = [e_\pi : \pi \in \Pi_L]$. Note the parameter L represents a complexity constraint, because it limits both the path's maximum length and the number of features.

Example 2 *For instance, consider the comparison between the interpretable representation of Dom Pedro II ϕ_{pedro}, and its real-valued vector $\hat{e}_{pedro}$:*

$$\Pi_2 = \begin{bmatrix} religion \\ nationality \\ \vdots \\ spouse - gender \end{bmatrix} : \phi_{pedro} = \begin{bmatrix} catholic \\ brazil \\ \vdots \\ female \end{bmatrix}, \hat{e}_{pedro} = \begin{bmatrix} 0.9 \\ 1.2 \\ \vdots \\ 0.1 \end{bmatrix}. \tag{5}$$

In order to answer the second question, we propose to use the knowledge embedding itself to extract the graph features. Since each path π_l is compound by a sequence of relations $r_1 - r_2 - ... - r_l$, where $r_i \in \mathcal{R}, \forall i \in \{1, 2, ...l\}$, we can use our classifiers $g_r, r \in \mathcal{R}$ (the knowledge embedding itself) to extract the interpretable features for a given entity embedding.

Example 3 *Back to our toy example, we illustrate the embedding feature extraction for the compound feature $spouse - gender$ of the entity dom_pedro_ii. First we discover the spouse of Dom Pedro II using the function g_{spouse}, then we inquire for her gender using g_{gender}. That is,*

$$g_{spouse}(pedro|\Theta) = teresa \rightarrow g_{gender}(teresa|\Theta) = female. \tag{6}$$

Once we defined how to map the embeddings to their interpretable representations, we are ready to proceed. Suppose we wish to explain why e_t is a plausible tail entity for the triple $\langle e_h, r, ?\rangle$. First, we sample K data points around $\hat{e_h}$, similarly to LIME applied to tabular data [Ribeiro et al. 2016], thus generating a dataset $\mathcal{Z}$ of perturbed samples $\hat{z_k}$. It is worth to mention that unlike LIME, we sample around the input original representation, instead of its interpretable one.

Next, for each perturbed sample $\hat{z_k} \in \mathcal{Z}$ we realize the feature extraction procedure previously described. That is,

Algorithm .1 Explanation Generation

procedure EXTRACT-FEATURES($\hat{z}_k, \Pi_L, \mathcal{G}$)
  $\phi_k = \{\}$
  for all $\pi \in \Pi_L$ **do** ▷ Equivalent to Equation (7)
    $e_\pi \leftarrow \hat{z}_k$
    for each edge $r_j \in \pi$ **do**
      $e_\pi \leftarrow g_{r_j}(e_\pi)$
    $\phi_k \leftarrow \phi_k \cup e_\pi$
  return ϕ_k
procedure EXPLAIN-INSTANCE($\hat{e}_h, r, t, L, \mathcal{G}$)
  $\Phi \leftarrow \{\}$
  $\Pi_L \leftarrow$ GRAPH-FEATURES(L) ▷ Generate set of path features
  for $k \in 1, 2, 3, ..., K$ **do**
    $\hat{z}_k \leftarrow$ SAMPLE-AROUND$(\hat{e}_h)$ ▷ Generate perturbed sample
    $\phi_k \leftarrow$ EXTRACT-FEATURES$(\hat{z}_k, \Pi_L, \mathcal{G})$
    $\Phi \leftarrow \Phi \cup \langle \phi_k, g_r(\hat{z}_k), d(\hat{e}_h, \hat{z}_k) \rangle$
  $g'_r \leftarrow SLR(\Phi)$ ▷ Train interpretable classifier with ϕ_k as features and t as target
  Draw explanations from g'_r in terms of feature importance

$$\phi_k = [e_\pi : g_\pi(\hat{z}_k), \pi \in \Pi_L]. \tag{7}$$

As a result of the previous step, we have the interpretable representation $\phi_k \in \Phi$ for each perturbed sample $z \in \mathcal{Z}$. Finally, we train a intrinsically interpretable classifier, such as a sparse logistic regression (SLR), $g'_r \leftarrow SLR(\Phi)$ and draw explanations from it in terms of feature importance, e.g. the top n high-valued coefficients, following Algorithm .1.

6. Experiments

In this section we present and discuss our empirical results.

6.1. Set-Up

Before we properly evaluate our proposed explanation method, we first need to generate a knowledge embedding. Therefore, we start by describing the selection of dataset and model for the embedding, followed by the training procedures and results.

We selected the subset FB13 of Freebase [Bollacker et al. 2008], as it has been consistently used as benchmark [Gardner and Mitchell 2015] [Gusmão et al. 2018]. FB13 dataset contains $75,043$ entities, 13 relations and a total of $345,873$ triples that are divided into $316,232$, $5,908$ and $23,733$ triples for training, validation and testing, respectively.

We selected the model TransE [Bordes et al. 2013] to train our embedding, as it is a commonly used baseline [Gusmão et al. 2018] [Zhang et al. 2019] and many popular embeddings models are inspired by it [Wang et al. 2017]. TransE models relationships as translations on embedding space and the plausibility score function as a distance measure,

Table 2. Embedding Model Training Parameters

Parameter	Train times	Batches	Alpha	Margin	Dimension	Optimizer
Value	1000	100	0.001	1.0	100	ADAGRAD

Table 3. Link Prediction results

Metric	MR		MRR (%)		Hits@1(%)		Hits@10(%)	
Eval. setting	Raw	Filt.	Raw	Filt.	Raw	Filt.	Raw	Filt.
No Type Const.	14,571	7,279	26.30	28.26	20.81	23.34	36.47	37.32
Type Const.	12,338	5,047	26.32	28.29	20.82	23.36	36.51	37.38

so that if the summation of the head entity vector and the relation vector is close to the tail vector in the real-valued space, the triple holds.

The whole system was implemented in Python, with the implementation of TransE from OpenKE [1]. We also employed packages matplotlib [2] and pandas [3].

The TransE model was trained on FB13 following the parameters presented in Table 2. The resulting knowledge embedding was evaluated on common knowledge base completion tasks (triple classification and link prediction) [Wang et al. 2019] following the procedure of [Bordes et al. 2013]. The triple classification accuracy obtained was 79.52%, while the summary overall metrics for link prediction is presented in Table 3. The metrics obtained by our knowledge embedding are similar to those reported in previous works [Gusmão et al. 2018].

As a complement to the objective metrics for triple classification and link prediction, we also realized a qualitative analysis on our embedding. Table 4 presents the top 3 entities for sample link predictions using our trained knowledge embedding. One can see that the predicted tails are coherent, what suggests that the embedding is appropriate.

6.2. Results Discussion

In this section we evaluate our explanation method, so as to answer:

1. **Q1**: Can meaningful and human-friendly explanations be drawn from the proposed method?
2. **Q2**: Does the explanation give reasons to both support or deny a prediction?
3. **Q3**: Are the presented reasons coherent?

As these questions are inherently subjective, their evaluation relies on qualitative analysis. We assumed that manual inspection of embedding predictions and their correspondent explanations is a suitable evaluation technique for this purpose. Our proposed explanation method used a $L = 1$ as maximum path length and a number of samples $K = 5000$. The distance function d_x considered was an exponential kernel defined on *L1* distance. The interpretable classifier used in our test was a sparse logistic regression.

[1] https://github.com/thunlp/OpenKE

[2] https://matplotlib.org/

[3] https://pandas.pydata.org/

Table 4. Link Prediction ranking sample triples

Head	Relation	Predicted Tails (Top-3)
Jesus	religion	Judaism, Christian, Christianity
Jesus	nationality	Israel, Roman Empire, Iudaea Province
Gandhi	profession	Independence Activist, Political Prisioner, Statesman
Anne Frank	profession	Writer, Author, Poet

Table 5 presents the top 3 reasons (feature importance on interpretable classifier) [4]. For instance, the reasons that supports "Judaism" as Jesus religion are that his ethnicity is "Jew", his profession is "Rabbi" and his nationality is from the country "Israel". The intuition behind this explanation is that the presented reasons represent features that are most correlated to the tail entity. The features that are indeed correlated to the tail are highlighted. Because for all three samples at least one feature is highlighted, we can observe that our method successfully identified meaningful explanations, answering the first question (**Q1**).

In addition, as the interpretable classifier assigns negative weights for features that denies the tail entity, the most negative coefficients can be interpreted as reasons that refute the prediction. It is desired for an explanation to present both the pros and cons reasons for a given prediction, because it offers a broader understanding than presenting only one perspective. For example, the cause of death (cod) being crucifixion supports that Jesus religion is "Christian", while the fact that his ethnicity is "Jew" negates. Our proposed method successfully identified meaningful pros and cons reasons for all the samples, what answer the second question (**Q2**).

It is worth noting that, for a given relation, a positive reason for one tail entity tends to be a negative against another one. For instance, while ethnicity "Jew" is a feature that supports the religion "Judaism", it at the same time discourages the alternative "Christian". This characteristic can be interpreted as coherence between features, thus addressing the third question (**Q3**).

7. Conclusion

This paper has proposed a novel model-agnostic method for faithfully explaining knowledge embedding predictions. We have also also presented a feature extraction technique that provides an interpretable representation for multi-relational embeddings. The results produced by our method support local-surrogate techniques as offering a promising approach to explain embedding predictions in a human-friendly manner.

The present work is a step towards explainable knowledge embeddings. Future work should include a comparison between different types of graph features and the exploration of other approaches for explanation generation, such as counterfactuals, and an evaluation of explanations with human subjects.

[4] "ethn", "prof", "nat", "rel", "cod" and "pob" are abbreviations for "ethnicity", "profession", "nationality", "cause_of_death" and "place_of_birth", respectively.

Table 5. Explanation samples in FB13

Head	Jesus	Jesus	Jesus
Relation	Religion	Religion	Cause of Death
Tail	Judaism	Christian	Crucifixion
Pros (Positive coefficients)			
Reason 1	1.03 *ethn.* → **jew**	1.03 *cod* → **crucifixion**	0.57 *rel.* → **christian**
Reason 2	0.82 *prof.* → **rabbi**	0.75 *nat.* → puerto rico	0.50 *prof.* → **prophet**
Reason 3	0.73 *nat* → **israel**	0.50 *cod.* → air. crash	0.41 *pob.* → cluj-napoca
Cons (Negative coefficients)			
Reason 1	(0.92) *nat.* → puerto rico	(1.05) *nat.* → **israel**	(1.22) *rel.* → **judaism**
Reason 2	(0.4) *cod.* → **crucifixion**	(0.97) *pob.* → lahore	(0.91) *pob.* → kabul
Reason 3	(0.27) *prof.* → prophet	(0.75) *ethn.* → **jew**	(0.77) *ethn.* → javanese

Acknowledgements

This work was carried out with the support of Itaú Unibanco S.A.; the first author has been supported by the Itaú Scholarship Program (PBI), linked to the Data Science Center (C2D) of the Escola Politcnica da Universidade de São Paulo.

The second author has been partially supported by the Conselho Nacional de Desenvolvimento Cientifico e Tecnologico (CNPq), grant 312180/2018-7. The work was also supported by the Fundação de Amparo a Pesquisa do Estado de São Paulo (FAPESP), grant 2016/18841-0, and also by the Coordenação de Aperfeiçoamento de Pessoal de Nivel Superior (CAPES) - finance code 001. We are also grateful to the Center for Innovation at Universidade de São Paulo (InovaUSP) for hosting our lab.

References

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. (2008). Freebase: A Collaboratively Created Graph Database For Structuring Human Knowledge. Technical report, Metaweb Technologies.

Bordes, A., Usunier, N., García-Durán, A., and Weston, J. (2013). Translating Embeddings for Modeling Multi-Relational Data. *Advances in Neural Information Processing Systems*, pages 2787–2795.

Gardner, M. and Mitchell, T. (2015). Efficient and Expressive Knowledge Base Completion Using Subgraph Feature Extraction. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1488–1498. Association for Computational Linguistics.

Gusmão, A. C., Correia, A. C., De Bona, G., and Cozman, F. G. (2018). Interpreting Embedding Models of Knowledge Bases : A Pedagogical Approach. In *2018 ICML Workshop on Human Interpretability in Machine Learning (WHI 2018)*, number Whi, pages 79–86.

He, R., Kang, W.-C., and McAuley, J. (2017). Translation-based recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, RecSys '17, pages 161–169, New York, NY, USA. ACM.

Huang, X., Zhang, J., Li, D., and Li, P. (2019). Knowledge graph embedding based question answering. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, WSDM '19, pages 105–113, New York, NY, USA. ACM.

Kazemi, S. M. and Poole, D. (2018). Simple embedding for link prediction in knowledge graphs. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31*, pages 4284–4295. Curran Associates, Inc.

Lao, N., Mitchell, T., and Cohen, W. W. (2011). Random Walk Inference and Learning in A Large Scale Knowledge Base. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 529—-539, Edinburgh, United Kingdom. Association for Computational Linguistics.

Murphy, B., Talukdar, P., and Mitchell, T. (2012). *Learning Effective and Interpretable Semantic Models using Non-Negative Sparse Embedding*, pages 1933–1950. The COLING 2012 Organizing Committee, Mumbai, India.

Nickel, M., Murphy, K., Tresp, V., and Gabrilovich, E. (2016). A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 1135–1144, New York, NY, USA. ACM.

W3. RDF 1.1 Concepts and Abstract Syntax.

Wang, Q., Mao, Z., Wang, B., and Guo, L. (2017). Knowledge Graph Embedding : A Survey of Approaches and Applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743.

Wang, Y., Ruffinelli, D., Gemulla, R., Broscheit, S., and Meilicke, C. (2019). On evaluating embedding models for knowledge base completion. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 104–112, Florence, Italy. Association for Computational Linguistics.

Wang, Z., Zhang, J., Feng, J., and Chen, Z. (2014). Knowledge Graph Embedding by Translating on Hyperplanes. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 1112, 1119.

Xie, Q., Ma, X., Dai, Z., and Hovy, E. H. (2017). An interpretable knowledge transfer model for knowledge base completion. In *ACL*.

Zhang, W., Paudel, B., Zhang, W., Bernstein, A., and Chen, H. (2019). Interaction embeddings for prediction and explanation in knowledge graphs. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, WSDM '19, pages 96–104, New York, NY, USA. ACM.