Look beyond the Surface: A Demo for Explaining Knowledge Graph Embeddings and Entity Similarity

Huu Tan Mai[1,2], Youmna Ismaeil[1,3], Trung-Kien Tran[1], Hendrik Bloeckeel[3] and Daria Stepanova[1]

[1]*Bosch Center for Artificial Intelligence, Robert Bosch Campus 1, 71272 Renningen, Germany*
[2]*Télécom Paris, Palaiseau, France*
[3]*Department of Computer Science, KU Leuven, Leuven BE 3000, Belgium*

Abstract

Knowledge Graph embedding (KGE) methods are concerned with mapping entities and relations in a KG into a low-dimensional vector space. KGEs have been effectively used for a variety of tasks such as link prediction, and entity classification or entity similarity. However, these methods are often considered as black boxes, providing users with no insights into the information captured by the embeddings and justifications for the computed outcome on a particular task. Recently, FeaBI, a framework for interpreting pre-computed entity embeddings relying on entity neighborhoods, has been proposed. In this paper we present a demo for this work. Our intuitive and interactive demo allows users to conveniently exploit the respective framework for computing embedding-based similarity between KG entities as well as generating and visualizing explanations for the respective similarity.

Keywords

Explainable Entity Similarity, Knowledge Graphs, Knowledge Graph Embeddings

1. Introduction

Knowledge Graph embeddings (KGEs) (see, e.g., [1]) represent entities and relations in a low-dimensional vector space. They have been useful in a range of tasks, including link prediction (e.g., [2, 3, 4]), entity classification (e.g., [5, 6]) or entity similarity. However, despite their success, KG embeddings are often regarded as black boxes. Lack of transparency and interpretability of KGEs limits users' understanding of their inner mechanisms, and undermines the trust in these models. E.g., given an entity, embedding-based suggestions regarding other entities similar to it might be less convincing if the user cannot examine the reasons behind the similarities.

Recently, a framework named FeaBI [7] has been proposed for explaining pre-computed entity embeddings. More specifically, given a KG and its embedding model FeaBI employs embedded feature selection techniques to extract from the KG propositional features in the form of relations and entities that are important for a given KG embedding model. These features are treated as KG embedding model explanations. FeaBI can be conveniently used for explaining similarities between entities.

ISWC 2023 Posters and Demos: 22nd International Semantic Web Conference, November 6–10, 2023, Athens, Greece
huu.mai@telecom-paris.fr (H. T. Mai); youmna.ismaeil@de.bosch.com (Y. Ismaeil); trungkien.tran@de.bosch.com (T. Tran); hendrik.blockeel@kuleuven.be (H. Bloeckeel); daria.stepanova@de.bosch.com (D. Stepanova)

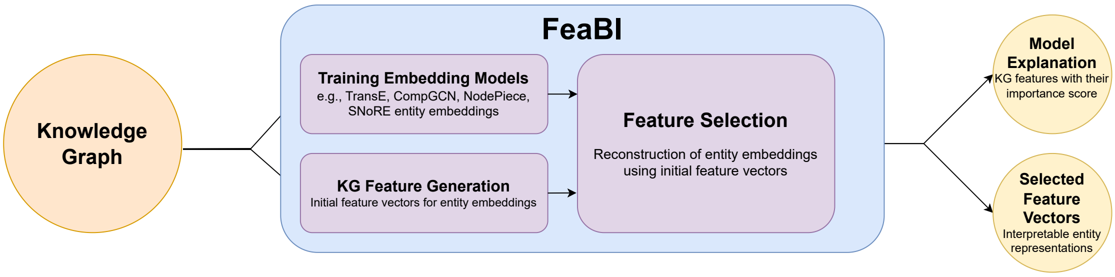

Figure 1: Overview of the Feature Selection-Based Framework

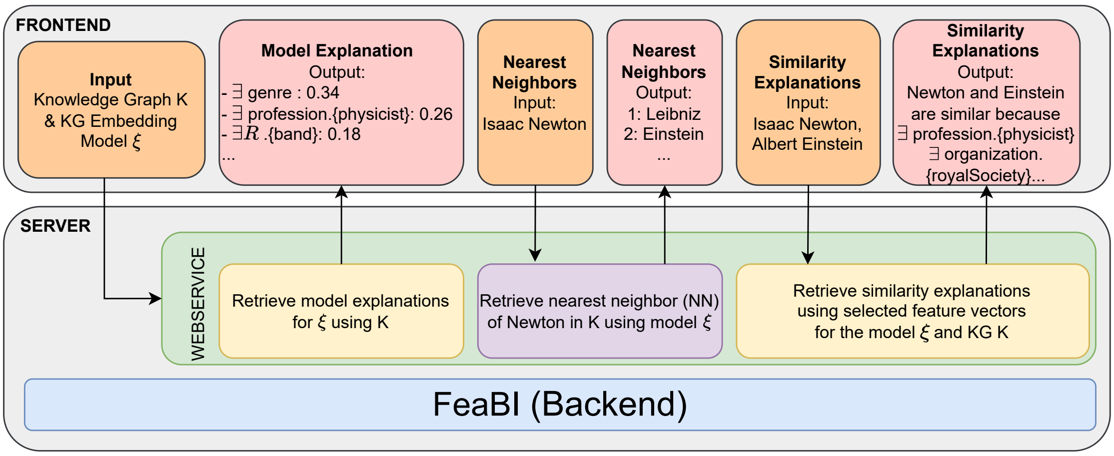

Figure 2: Structure of the Demo for the Framework (local setup)

In this paper, we present a demo for FeaBI, which offers a user-friendly graphical interface to facilitate user experience in exploring the computed explanations. This is achieved through the visualizations of the relevant graph-based entity surroundings and textual descriptions of explanations. The demo can be used for the analysis and comparison of different embedding models in terms of KG features that they capture, thus supporting the users in deciding which embedding model would suit their purpose best.

The recorded video of our demo is available at https://figshare.com/s/b941c7e0c800c23f5d82.

2. Demo Overview

Figure 2 presents an overview of our demo which is designed as a web application. In what follows, we describe the server and the frontend components of the demo in details.

The sever side is divided into two parts: 1) the backend (based on FeaBI [7]), which given a KG and its embedding model computes the explanations, and 2) the webservice, which communicates the KG and the embedding model chosen by the user to FeaBI and presents the received results to the user via the frontend.

KG	Model	FeaBI total runtime (s)
DBpedia50	TransE	10.39 ± 0.37s
	CompGCN	9.28 ± 0.31s
	NodePiece	9.96 ± 0.19s
	SNoRe	16.94 ± 0.20s
FB15k237	TransE	30.89 ± 2.02s
	CompGCN	26.45 ± 1.0s
	NodePiece	30.53 ± 0.82s
	SNoRe	33.01 ± 0.68s

Table 1
The runtime of feature construction and feature selection steps of FeaBI

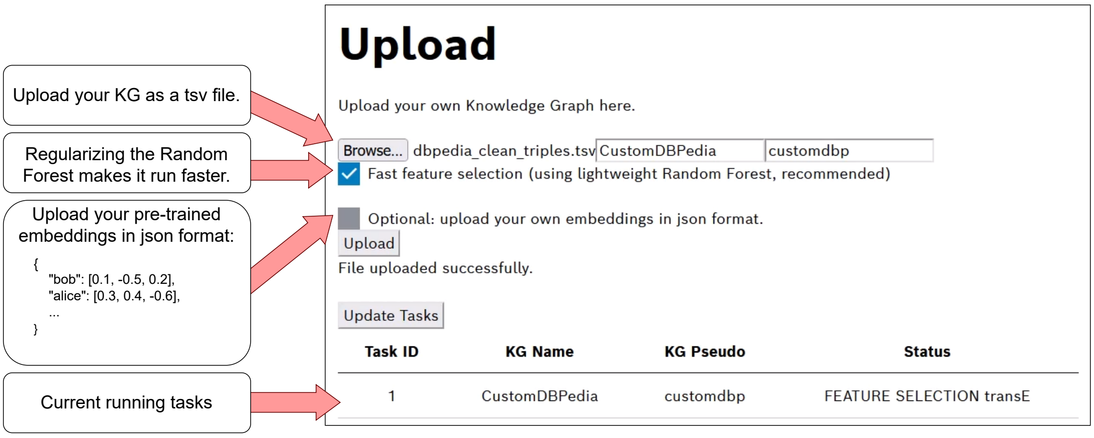

Figure 3: Illustration for different customizations, e.g., uploading a custom KG to the demo

Feabi (Backend). For a given KG and its embedding model, FeaBI computes KG embedding explanations defined as a list of KG features ranked based on their importance for the generation of the KG embedding. The top most important features are then used to build interpretable representations of the KG entity embeddings. The main components of FeaBI are KG embedding training, feature construction and feature selection (see [7] for details). The training of the KG embedding model is naturally the most time-consuming step, which typically takes up to 5 hours (e.g., for CompGCN on FB15K237 dataset). Therefore, in our demo we provide a number of pre-trained embedding models. At the moment we support 4 popular embedding models: TransE [2], CompGCN [8], NodePiece [9] and SNoRe [10], but other pretrained embeddings can also be provided by users as illustrated in Figure 3.

Table 1 shows the running time of the feature construction and feature selection steps of FeaBI for two popular KGs and embedding models available in the demo.

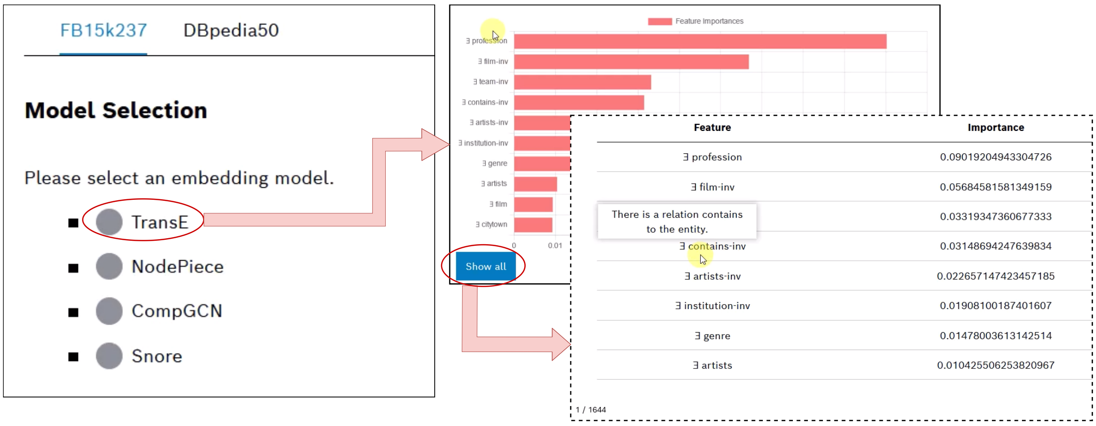

Figure 4: Model explanations as a ranked list of selected KG features

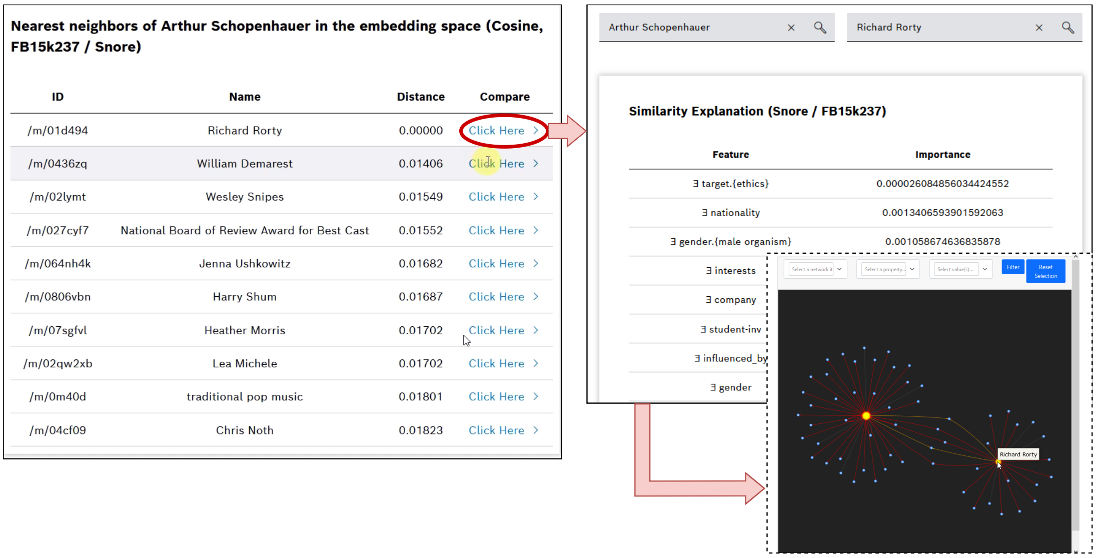

Figure 5: Similarity explanations and their visualization

Webservice. The webservice handles the communication of FeaBI with the frontend. In the frontend, the KG and KG embedding models are first selected by the user, and then passed to FeaBI via the webservice. Subsequently, FeaBI computes the results, which are then sent to the webservice and presented to the user via the frontend.

Frontend. The frontend allows users to conveniently explore the model explanations for a given embedding model, entity embedding explanations, as well as explanations for similarities between a pair of selected entities retrieved by the webservice.

The workflow of the demo proceeds as follows. First, the user selects a KG and an embedding model from the provided list (or uploads custom ones) via the visual interface. Then, a model explanation (i.e., a list of symbolic features ranked by their importance) is automatically generated and presented to the user (see Figure 4).

Additionally, the demo offers a possibility to compare entities in the KG in terms of their similarity relying on the given embedding model. As shown in Figure 5, for a given entity provided by the user, similar entities can be retrieved based on the distance metric in the embedding space (cosine similarity and Euclidean distance are currently supported). The user can select any pair of entities and use the system to generate explanations for their similarity, i.e., a list of selected KG features that the entities share along with their graph-based visualizations.

3. Conclusion

We presented a demo for FeaBI [7], which is a recently proposed framework for explaining KG embedding models. While the work in [7] focuses on technical details of the method, our demo system allows the users to easily analyse KG features captured by an embedding model as well as reasons behind embedding-based entity similarities. Future directions include the analysis of explanations for relation embeddings as well as the consideration of ontologies and KG schemes within the studied framework.

Acknowledgements. This work was partially funded by the grant ANR-20-CHIA-0012-01 ("NoRDF") and the European project SMARTEDGE (grant number 101092908).

References

[1] Q. Wang, Z. Mao, B. Wang, L. Guo, Knowledge graph embedding: A survey of approaches and applications, IEEE Transactions on Knowledge and Data Engineering 29 (2017) 2724–2743.

[2] A. Bordes, N. Usunier, A. García-Durán, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, in: NeurIPs, 2013, pp. 2787–2795.

[3] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, G. Bouchard, Complex embeddings for simple link prediction, in: ICML, 2016, pp. 2071–2080.

[4] B. Yang, W. Yih, X. He, J. Gao, L. Deng, Embedding entities and relations for learning and inference in knowledge bases, in: ICLR, 2015.

[5] P. Ristoski, J. Rosati, T. D. Noia, R. D. Leone, H. Paulheim, Rdf2vec: RDF graph embeddings and their applications, Semantic Web 10 (2019) 721–752.

[6] T. N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: ICLR 2017, OpenReview.net, 2017.

[7] Y. Ismaeil, D. Stepanova, T. K. Tran, H. Blockeel, Feabi: A feature selection-based framework for interpreting kg embeddings, ISWC (2023).

[8] S. Vashishth, S. Sanyal, V. Nitin, P. P. Talukdar, Composition-based multi-relational graph convolutional networks, in: ICLR, 2020.

[9] M. Galkin, E. G. Denis, J. Wu, W. L. Hamilton, Nodepiece: Compositional and parameter-efficient representations of large knowledge graphs, in: ICLR, 2022.

[10] S. Mežnar, N. Lavrač, B. Škrlj, Snore: Scalable unsupervised learning of symbolic node representations, IEEE Access 8 (2020) 212568–212588.