Predicting Antiviral Drugs over COVID-19 Knowledge Graph

Tianqing Fang, Changlong Yu
Department of Computer Science and Engineering
HKUST, Hong Kong, China
{tfangaa, cyuaq}@cse.ust.hk

Abstract

We study the link prediction problem on a novel COVID-19 related knowledge graph that represents relationships between viruses, proteins, and drugs. Several representative knowledge graph embedding models, such as TransE, ComplEx, and RotatE are studied to perform link prediction. We also perform node type classification using the acquired embeddings by both knowledge graph embedding models and a random-walk-based network embedding method. Experments show that the RotatE model performs the best under the COVID-19 knowledge graph for both link prediction and node classification. Random-walk-based network embeddings, which are learned without leveraging detailed relational types, performs poorly on the node classification test. Visualization regarding the acquired embeddings by different models also demonstrate the superiority of RotatE on clustering different node types.

1 Introduction and Background

The virus of COVID-19 (SARS-CoV-2) is an RNA-based virus that parasites to the host, replicating itself and produce virus proteins inside human cells to infect humans. The genetic materials in the virus will be injected into the cells of the host and control the production of virus protein. In modern biology, antiviral drugs focus on eventually preventing the process of producing virus protein during certain biological processes. Extensive drug experiments are needed to test the effects of drugs on different viruses. However, there may be known relationships between different viruses, and also between different proteins. For example, different viruses may produce similar virus proteins, and there may also be interactions between different proteins. With such knowledge between biological entities, machine learning models can be applied to predict missing relationships for unseen viruses or proteins.

Knowledge graphs (KGs) are such heterogeneous structured representations of real-world knowledge, where nodes could be named entities [1], concepts [2] or events [3] and edges represent the semantics relations between nodes. KGs have been widely used for downstream tasks such as query understanding, question answering, and recommendation systems. In the context of drug development, Knowledge Graphs can be used to store existing knowledge among the interactions between proteins, the host-protein relations, virus-drug effects, and so on. For example, a real knowledge graph regarding related proteins and drugs of HIV is shown in Figure 1. The Virus protein "NP" is the one that HIV produces and affects host proteins like "ITA4" and "DJB11". By incorporating more diverse relationships between human proteins we may be able to infer the effects of certain drugs on some unknown proteins. For COVID-19, with the newly developed knowledge graph on COVID-related entities [1], we can adopt artificial intelligence techniques to predict missing links between virus and drugs, thus helping the process of drug development.

[1] https://www.biendata.xyz/competition/ccks_2020_7_3/

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

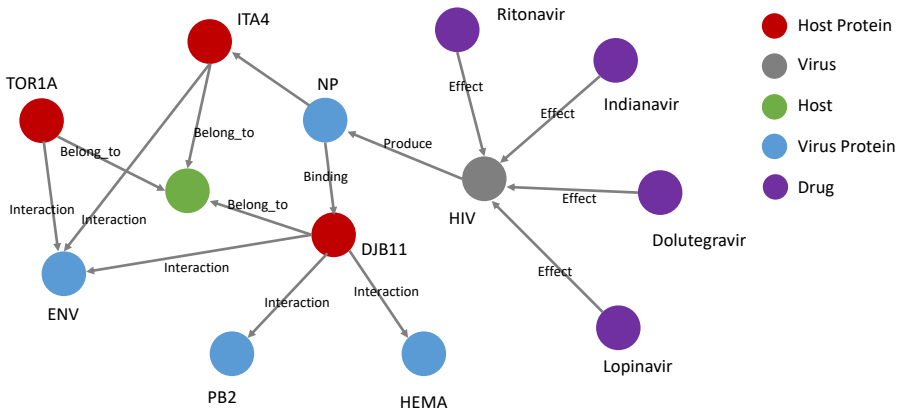


Figure 1: A subgraph from the COVID19 knowledge graph. Nodes with different colors represent different types and nodes are connected with different relations.

Types	Numbers	Relations
Drug	72	Drug *effect* Virus
Virus	240	Virus *produce* Protein
Virus protein	821	Virus protein *binding* Host protein
Host protein	6,711	Host protein *interaction* Virus protein

Table 1: The statistics of entities with different types and the relation constraints between types of entities. Note that proteins include virus proteins and host proteins.

2 Problem Formulation & Approaches

2.1 Data

The statistics of datasets are shown in the Table 1. Generally the knowledge graph defines four types of entities as nodes, i.e., Drugs, Viruses, Virus Proteins, and Host Proteins. We can see that most of entities belong to *Host Proteins*. Also the types between entities and constraints are indicated in the "Relations" columns in the table.

2.2 Knowledge Graph Embeddings

We introduce several off-the-shelf knowledge graph embedding methods used in our projects and normally the knowledge graph is stored as triplets $\mathcal{D} = \{(h, r, t)\}$ where h and t are the head entity and tail entity respectively. r is the relation between them. A typical KG embedding method represents entities ($\mathbf{h}$, $\mathbf{t}$) and relations ($\mathbf{r}$) as vectors or matrices. Then a score function $f_r(h, t)$ is defined to measure the feasibility of the triplet and further used to optimize the maximum of total scores. The common scoring functions are listed in the Table 2 and explained as follows.

TransE [4] is the most representative work of using translation-based knowledge graph embeddings. It represents the relations and entities in the same topological space. The score function is defined as the negative distance between $\mathbf{h} + \mathbf{r}$ and $\mathbf{t}$. The score should be larger if the relation holds.

TransR [5] extended TransE by introducing relation-specific spaces and the head/tail entities are firstly projected into the space specific to relation r, i.e., $\boldsymbol{h}\boldsymbol{M}_r$ and $\boldsymbol{h}\boldsymbol{M}_r$ where $\boldsymbol{M}_r$ is the projection matrix. The other optimization keeps the same as TransE.

TransD [6] further simplified TransR by decomposing the projection matrix into the product of two vectors, i.e., $\boldsymbol{M}_r = \boldsymbol{r}_p \boldsymbol{h}_p^T + \boldsymbol{I}$. Compared with TransR, it requires less parameters and is more efficient. But topologically those two resulted embeddings are similar (the visualization in Fig 2).

Model	Score Function	Conditions
TransE	$-\|\|\boldsymbol{h}+\boldsymbol{r}-\boldsymbol{t}\|\|$	$\boldsymbol{h},\boldsymbol{r},\boldsymbol{t}\in\mathbb{R}^k$
TransR	$-\|\|\boldsymbol{h}\boldsymbol{M}_r+r-\boldsymbol{t}\boldsymbol{M}_r\|\|^2$	$\boldsymbol{h},\boldsymbol{r},\boldsymbol{t}\in\mathbb{R}^k,\boldsymbol{M}_r\in\mathbb{R}^{k\times k}$
TransH	$-\|\|(\boldsymbol{h}-\boldsymbol{w}_r^T\boldsymbol{h}\boldsymbol{w}_r)+\boldsymbol{d}_r-(\boldsymbol{t}-\boldsymbol{w}_r\boldsymbol{t}\boldsymbol{w}_r)\|\|^2$	$\boldsymbol{h},\boldsymbol{r},\boldsymbol{t}\in\mathbb{R}^k,\boldsymbol{w}_r\in\mathbb{R}^{k\times k}$
TransD	$-\|\|(\boldsymbol{r}_p\boldsymbol{h}_p^T+\boldsymbol{I})\boldsymbol{h}+\boldsymbol{r}-(\boldsymbol{r}_p\boldsymbol{t}_p^T+\boldsymbol{I})\boldsymbol{t}\|\|^2$	$\boldsymbol{h},\boldsymbol{h}_p\boldsymbol{t},\boldsymbol{t}_p\in\mathbb{R}^n,\boldsymbol{r},\boldsymbol{r}_p\in\mathbb{R}^m,\boldsymbol{I}\in\mathbb{R}^{m\times n}$
ComplEx	$\mathrm{Re}(\langle\boldsymbol{r},\boldsymbol{h},\bar{\boldsymbol{t}}\rangle)$	$\boldsymbol{h},\boldsymbol{r},\boldsymbol{t}\in\mathbb{C}^k$
RotatE	$-\|\|\boldsymbol{h}\circ\boldsymbol{r}-t\|\|^2$	$\boldsymbol{h},\boldsymbol{r},\boldsymbol{t}\in\mathbb{R}^k,\|r_i\|=1$

Table 2: Scoring functions with respect to different KG embedding methods.

TranH [7] also extended TransE by introducing relation-specific hyperplanes. It models entities again as vectors but relation r as a vector $\boldsymbol{r}$ on a hyperplane $\boldsymbol{w}_r$. Specially $\boldsymbol{h}=\boldsymbol{h}-\boldsymbol{w}_r^T\boldsymbol{h}\boldsymbol{w}_r$ and $\boldsymbol{t}=\boldsymbol{t}-\boldsymbol{w}_r^T\boldsymbol{t}\boldsymbol{w}_r$. Different from TransE, TransH models an entity with different relations effectively.

Complex [8] embeds entities and relations not in a real space but a complex space. The score function is defined as:

$$f_r(h,t)=\mathrm{Re}(\boldsymbol{h}^T\mathrm{diag}(\boldsymbol{r})\bar{\boldsymbol{t}}) \tag{1}$$

where $\bar{\boldsymbol{t}}$ is the conjugate of t in the complex space and $\mathrm{Re}()$ means the real part of a complex value.

RotatE [9] defines each relation as a rotation from the source entity to the target entity. The scoring function is defined as $\boldsymbol{h}\circ\boldsymbol{r}-\boldsymbol{t}$, where $\circ$ is the element-wise product. RotatE is currently the state-of-the-art kg embedding method, which can infer more relation patterns (symmetry or anti-symmetry).

The learning objective is to minimize a margin-based ranking criterion over the training:

$$\mathcal{L}=\sum_{(h,r,t)\in\mathcal{D}}\sum_{(h',r,t')\in\mathcal{D}'_{(h',r,t')}}[\gamma+f_r(h,t)-f_r(h',t')]_+ \tag{2}$$

Here, (h,r,t) is a ground-truth triple in the training set. $\mathcal{D}'_{(h',r,t')}$ is the set of triples by replacing a ground-truth h with another random head h', and t with a random t'.

In summary, the aforementioned embedding methods could map relational graphical structures into continuous vector spaces. Specially we could make use of trained entity embeddings or relation embeddings to further investigate the interesting problems inside the knowledge graph. In this project, we study one COVID19 knowledge graph to infer the antivirus drug (link prediction) and predict entity types (node classification). Finally we use dimensional deduction technique to understand the embedding spaces from different methods.

2.3 Link Prediction and Node Classification

Link prediction is typically to predict h given $(?,r,t)$ or t given $(h,r,?)$. For example, (?, *Effect*, HIV) in the Fig 1, we can take every entity h' in the KG as a candidate result and calculate a score $f_r(h',t)$ according to the score functions in the Table 2. And the link prediction task is normally framed as ranking problems.

Node classification is to predict entity types (i.e., *Drugs*, *Viruses*, *Virus Proteins*, and *Host Proteins* in the Table 1) with respect to their node features from entity embeddings. Besides translation-based KG embeddings aforementioned, we are also interested in whether random-walk based embedding methods, such as Node2Vec [10] could well preserve the structured similarity and perform well on the link prediction tasks on the relational graph. The idea is that random walk would explore the neighborhood information and enable to maximize the likelihood of preserving diverse neighborhoods. Hence we also compare with the node embeddings from Node2Vec and though it does not contain relational connections, it provides rich neighborhood information.

$$P(c_i=x|c_{i-1}=v)=\begin{cases}\pi_{vx} & if(v,x)\in E\\ 0 & \boldsymbol{o}therwise\end{cases}$$

Model	MRR ↑	MR ↓	HITS@10 ↑	HITS@3 ↑	HITS@1 ↑
TransE	0.051284	765.906006	0.152021	0.034774	0.005874
TransR	0.040611	905.984253	0.115367	0.022556	0.003994
TransH	0.042955	811.696411	0.125000	0.024906	0.003759
TransD	0.043083	800.839050	0.125000	0.025846	0.003524
Complex	0.039259	858.044861	0.102679	0.026316	0.004229
RotatE	**0.060584**	**611.709351**	**0.175047**	**0.038064**	**0.008224**

Table 3: Link Prediction Results (Raw)

Model	MRR ↑	MR ↓	HITS@10 ↑	HITS@3 ↑	HITS@1 ↑
TransE	0.131119	702.265259	0.308271	0.158365	0.041353
TransR	0.117260	840.832947	0.275141	0.140742	0.039004
TransH	0.116027	746.440796	0.284774	0.140273	0.031955
TransD	0.118609	735.667542	0.282190	0.145912	0.034070
Complex	0.115814	792.442932	0.243186	0.125470	0.053102
RotatE	**0.217327**	**546.357117**	**0.414474**	**0.242716**	**0.122180**

Table 4: Link Prediction Results (Filtered)

3 Experiments

3.1 Setup and Evaluation

For both link prediction and node classification tasks, we randomly split the datasets as 80% training, 10% development and 10% testing set. We use three commonly-used classifiers (linear regress, Support vector machines and two-layer neural networks) for node classification taks and report averaged *accuracy*.

For evaluation of link prediction, we generate candidate triplets by by corrupting head or tail, i.e., (h', r, t) or (h, r, t') to be ranked by score functions. Specially we adopt "raw" and "filtered" setting, where the difference is whether the candidates appear in the training, validation, testing set or not. Mean Reciprocal Rank (MRR: the average of reciprocal ranks), Mean Rank (MR: the average of predicted ranks), and HITS@N (the proportion of ranks no larger than N, we take N as 1, 3, 10) are used for the evaluation metrics.

3.2 Results of Link Prediction

The performance of link prediction on the COVID-19 dataset is shown in Table 3 and 4. From all of the two tables, we can draw the following observations: (1). RotatE outperforms all the baselines and the results show that it could model the relational data well. (2). The "Filtered" setting is more reasonable as evaluation metrics. However the performances could be further improved if relation constraints are considered. For example the relation "produce" could only exist between "Virus" and "Protein".

3.3 Results of Node Classification

Classifier	TransE	TransR	TransH	TransD	ComplEx	RotatE	Node2Vec
LR	0.899	0.911	0.878	0.892	0.898	**0.978**	0.855
SVM	0.918	0.922	0.898	0.913	0.906	**0.980**	0.855
MLP	0.946	0.954	0.955	0.961	0.952	**0.978**	0.888

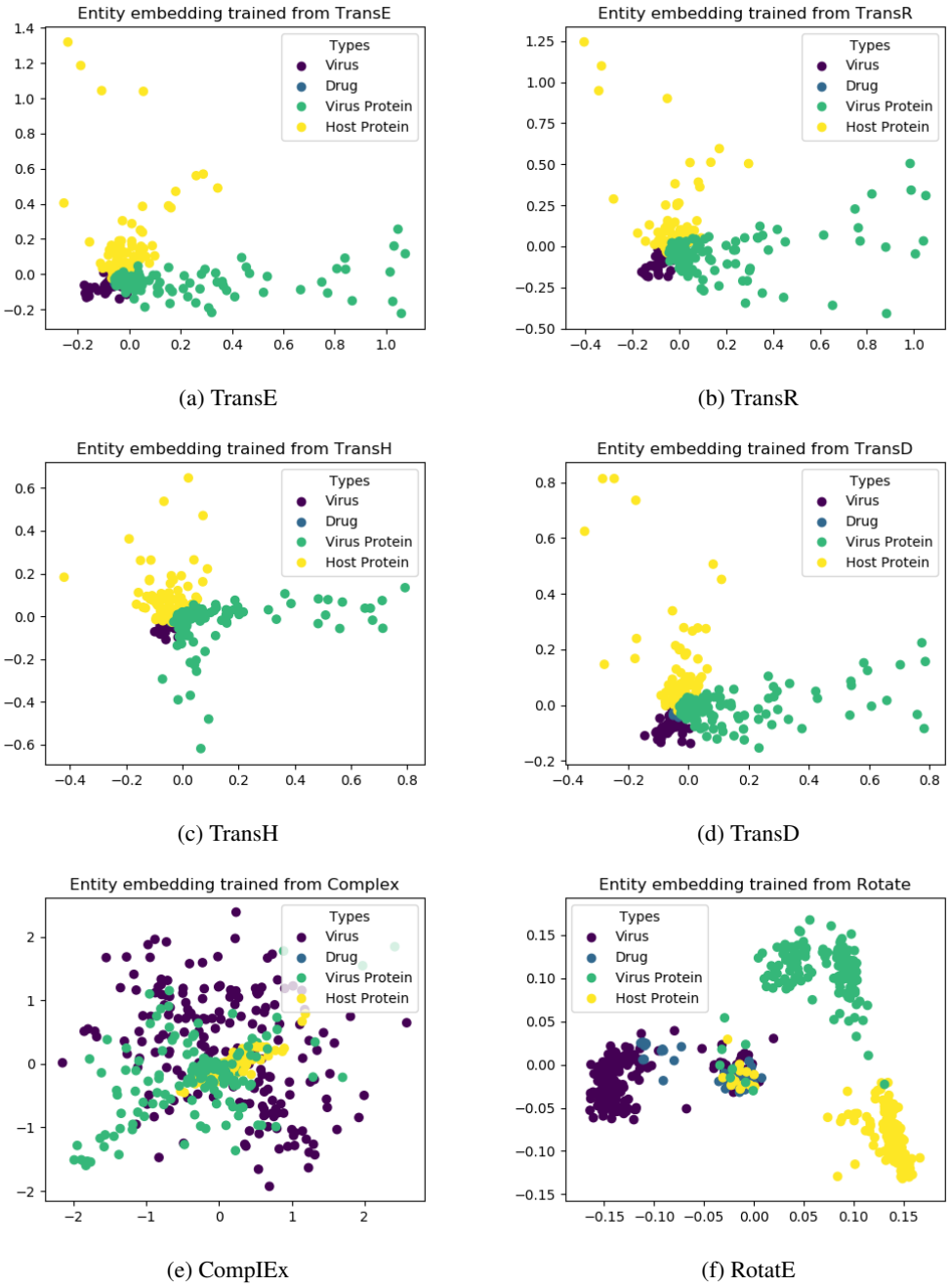
Table 5: The classification accuracy of using different node embeddings.

We show the classification results in the Table 5 using different kg embeddings. The RotatE model shows the consistent superiority in the node classification task and achieves all the best accuracy with

different classifiers. It is surprising that the Node2Vec with random walk performs the worst and the reason might be that it lacks of relational information which is important for knowledge graph.

3.4 Data Visualization

We used principal component analysis (PCA) to visualize embeddings of the entities using different knowledge graph embedding methods. The plots are shown in the Fig 2. We could observe that RotatE could take different types of nodes apart well and it shows that the embeddings learned are informative. Another interesting point is that TransR and TransD have similar point distributions as they learn embeddings in the similar vector spaces.



(a) TransE

(b) TransR

(c) TransH

(d) TransD

(e) CompIEx

(f) RotatE

Figure 2: Plot of entity embeddings with different knowledge graph embedding methods using PCA.

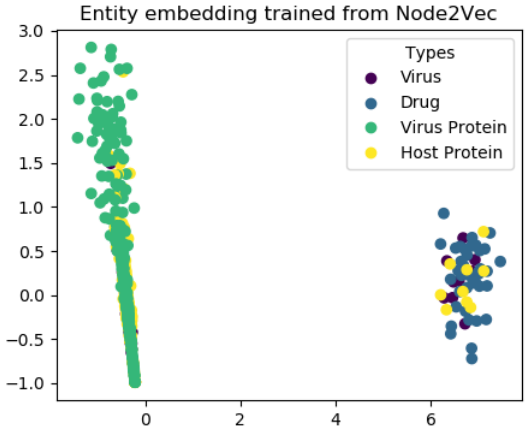


Figure 3: PCA visualization of Node2Vec nodes.

For random-walk based kg embedding, we could see from Figure 3 that the connections are weaker than translation-based kg embedding from the co-occurrence information. One take-away is that more fine-grained relation interactions should be involved for relational graph modeling. And different kg embeddings actually learn the node distribution from different spaces (shown in the above plot).

4 Conclusion

For COVID-19 Knowledge Graph Prediction, RotatE performs the best as it can encode all three frequent relational patterns, symmetric, inversion, and composition [9]. Visualization and Node classification results also demonstrate the superiority of RotatE regarding different node types. Random-walk-based methods are not suitable for the task as relational information is ignored, which is of vital importance in knowledge graph link prediction.

5 Contribution

Tianqing Fang: Link prediction models, report writing, video recording.

Changlong Yu: Node classification models, data visualization, report writing.

References

[1] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250, 2008.

[2] Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Q Zhu. Probase: A probabilistic taxonomy for text understanding. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pages 481–492, 2012.

[3] Hongming Zhang, Xin Liu, Haojie Pan, Yangqiu Song, and Cane Wing-Ki Leung. Aser: A large-scale eventuality knowledge graph. In *Proceedings of The Web Conference 2020*, pages 201–211, 2020.

[4] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Neural Information Processing Systems (NIPS)*, pages 1–9, 2013.

[5] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

[6] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of ACL*, pages 687–696, 2015.

[7] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28, 2014.

[8] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *International Conference on Machine Learning (ICML)*, volume 48, pages 2071–2080, 2016.

[9] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations*, 2019.

[10] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.