Evaluation of Multi-Hop Knowledge Graph Reasoning with Reward Shaping Via Replication Study

Lama Khalil
B.Sc.[H] Computer Science
School of Computer Science
University of Windsor
Windsor, ON, Canada
Khali121@uwindsor.ca

Dr. Ziad Kobti
Professor and Director
School of Computer Science
University of Windsor
Windsor, ON, Canada
kobti@uwindsor.ca

Dr. Kalyani Selvarajah
Assistant Professor
School of Computer Science
University of Windsor
Windsor, ON, Canada
selva111@uwindsor.ca

Table of Contents

1 Abstract

It has become evident that Knowledge Graphs (KG) are enormously helpful in transforming data into knowledge. The problem is that they are often incomplete and lack the ability to reason beyond the linkages within their entities.

Due to the increasing use of KG applications in the science and economics fields, a complete COVID-19 KG could be beneficial to the individuals who are trying to make informed decisions during this unprecedented time. The availability of COVID-19-related datasets allowed experts to build various COVID-19 Knowledge Graphs. However, inputting the data into a KG is not enough to provide intelligent answers, especially when the question contains numerous attributes and inferences. For this reason, new trends in KG research have rapidly emerged. A variety of KG completion methods were applied to COVID-19 Knowledge Graphs to overcome this challenge and generate the missing links automatically.

An efficient KG completion method may help the researchers to discover relevant information in the battle against COVID-19. Therefore, the aim of this study is to test the suitability of the state-of-the-art graph completion method in (Lin et. al., 2018) for real-world KGs. This is done through replicating the results by the application of this approach to the same benchmark

datasets used in the study. Our experiment confirmed that the query answering performance of this algorithm deteriorates significantly when it was tested on FB15k-237, compared to other benchmark datasets. This occurs because the inverse relationship triples, which occurred in the original dataset FB15K, were removed in the subset FB15k-237 to resemble the real-word datasets. Thus, our result could mean that this method is not going to perform well on a real-world dataset such as a COVID-19 pathophysiology KG. This KG was recently generated using coronavirus-related scientific literature by (Domingo-Fernandez et. al., 2020).

2 Key Words

- Covid-19
- Pandemic
- Knowledge Graph
- Knowledge Graph Completion
- COVID-19 KG

3 Acknowledgment

I want to thank Dr. Ziad Kobti for teaching me the proper scientific research methods and for being approachable and available to answer my questions all the time. I also want to thank Dr. Kalyani Selvarajah for suggesting the idea of research in the field of knowledge graphs and for the help she has provided during the whole process.

4 Introduction

The persistent COVID-19 pandemic poses significant concerns on the complexity of the real-world's data and exposes the challenges pertaining to data use (Kejriwal et. al., 2020). Information about the Coronavirus has a direct impact on the health, educational, economics fields and much more. Thus, COVID-19 complex multi-relational data has led to the construction of numerous, extensive but incomplete knowledge graphs (Akrami et. al., 2018).

Unlike past pandemics, data science and AI advances provided meaningful support to the study and understanding of this virus by health practitioners, epidemiologists, researchers, and policymakers (Kejriwal et. al., 2020). It is obvious that during this crisis, the world's greatest challenge is to analyze information and obtain answers to study and decide all aspects of life, with emphasis on finding a cure and providing adequate supplies for the healthcare sector. Knowledge graphs play a pivotal role in question answering, however, responding to complex questions involves data analysis, knowledge logical reasoning and making inferences. It is a difficult task even for human beings.

A notable example of coronavirus case counts shows how difficult it is to draw conclusions without analyzing multiple entities in a dataset. It is not possible to simply scan the data or perform a keyword-based search to answer complex questions when some relational-links are missing within and between various knowledge graphs (Domingo-Fernandez et. al., 2020). It is unlikely that any knowledge graph completion algorithm can achieve human reasoning and data linking accuracy in the near future (Lin et. al., 2018). Nonetheless, testing the available KG completion methodologies is one step forward.

In this paper we replicate and examine the algorithm on KG complication in (Lin et. al. 2018). Our goal is to verify the accuracy and effectiveness of this method for real-world KGs, such as the leading COVID-19 pathophysiology Knowledge graphs. We hope that this review can assist researchers, who are battling COVID-19, to find the answers that will help them discover improved methods to deal with this disease.

5 Literature review

We started by reviewing (Lin et. al. 2018). The focus of this study was the interpretability issue of the graph embedding method using Tensor Neural Network.

Although this method produces highly accurate results, it lacks interpretability. Therefore, this paper introduced the Multi-Hop Knowledge Graph Reasoning with Reward Shaping as a solution. In the later method an additional layer was added to the Tensor Neural Network to create a model that can sequentially search the KG until it reaches its target. The issue was that the new model was not as accurate as the original one.

While we were trying to investigate this issue, we came across (Akrami et. al. 218). This was a novel study. No other study discussed the issue with the benchmark datasets that have been extensively used to validate the performance of KG completion methods, such as F15k. This issue was that these methods contain the inverse of most triples. So, if the triple (s, r1, o) $\in$ G $\Rightarrow$ (o , r2, s) $\in$ G, this causes the inverse of many test triples to occur in the training set. Thus, the model would get the answers without predicting new linkage. To remedy this issue (Akrami et. al. 218) created a subset of FK15 (FK237), where the inverse relations were removed to resemble a real-world KG. Their study showed that the performance of most KG completion methods deteriorated once they were tested on FK237.

This motivated us to look further to locate a real-world dataset that can be used for KG completion methods validation. The one that we found was (Domingo-Fernandez et. al., 2020). When this KG was created it included 4016 nodes, covering 10 entity types (proteins, genes, chemicals and biological processes) and 10232 relationships. Additionally, the KG is available in multiple formats including CVS. Thus, we believe that this KG could be a good candidate to be added to the benchmark validation datasets.

6 Problem Statement

With the enormous amount of new data produced each day about COVID-19, it is necessary for researchers to determine the validity of the information in a COVID-19 KG and to complete the missing links. To overcome this challenge, several approaches have been applied on COVID-19 Knowledge graphs. More recently (Ioannidis et. al. 2020) and (Zhang et. al. 2021) used graph Neural Networks on COVID-19 KG for drug repurposing. The Graph Neural Networks like Neural Tensor Networks yield accurate and very efficient results, however they lack interpretability. This problem has been resolved in (Lin et. al. 2018).

Motivated by our literature review our goal is to verify if the Multi-Hop Knowledge Graph Reasoning with Reward Shaping can be used to infer accurate and interpretable linkage in the COVID-19 pathophysiology Knowledge graph. A clear understanding of the various strengths and limitations of this approach may lead future researchers to the discovery of new and inspiring methods that can be applied on COVID-19 Knowledge Graphs and yield better results.

7 General Concepts

7.1 COVID-19

(COVID-19) is a recently identified infectious respiratory disease caused by SARS-CoV-2 coronavirus (Canada Public Heath, 2021). The virus was first discovered in December 2019. In March 2020, it was declared a global pandemic by World Health Organization (WHO). To date, over two (2) million people have died of COVID-19 worldwide (WHO, 2021). The world has changed rapidly in every area of life in attempting to stop its further spread and limit the resulting illness and deaths.

7.2 Knowledge Graph

Knowledge Graphs (KGs) are computable databases that store the knowledge in a graph in which the entities are represented as nodes and their relationships as edges. Other than using graph data structure instead of tables, the primary difference between KGs and the traditional relational Database Management System (DBMS) rests mainly in the referential methodology used to connect the data.

In the DBMS foreign keys are used as a static referential tool, however, a static method is not suitable for the dynamic and fast-changing information in the world. Unlike relational DBMS, KG database is structured entirely around data relationships and how to dynamically update existing relationships (links) and infer new ones. Although KGs seem to be the perfect solution to store the facts and update the relationships dynamically, still KGs are incomplete by design. Formally speaking, KGs are defined as "G = (E, R), where E is the set of entities and R is the set of relations" (Lin et. al. 2018).

7.3 Knowledge Graph Incompleteness

Knowledge Graphs are incomplete by design, but what does this mean?

Let us take an example that will help us to understand the problem:

Our brains acquire new information each day. The information is stored in a way that resembles a Knowledge Graph, although it is not necessary to create links between this information as we acquire it. For example, we could learn that Justine Trudeau is the 23rd Prime Minister of Canada. Justine Trudeau will be associated with being the 23rd Prime Minister of Canada when this information is stored in our brains. A new piece of information could be learned in a few days, namely; that Sophie Trudeau is married to Justine Trudeau. The link between Sophie Trudeau and Justine Trudeau will be established when this information is stored in our memory. However, at that time there is not necessarily a connection between Sophie Trudeau and the 23rd Prime Minister in our brains. Something may later trigger the need to link the two pieces of information. For instance, the question: "who is the wife of Canada's 23rd Prime Minister?" Some neurons will be activated at this point to check the stored information and to detect missing links.

This example helps us to understand how the Knowledge graphs are incomplete by nature. Therefore, we need a computational method that could resemble human neurons in order to predict the missing links in a Knowledge Graph.

In (Lin et. al. 2018) the problem was formally defined as the need to perform an efficient search over a KG and collect the set of all possible answers. Each directed link in the KG $l = (e_s, r_q,\ e_o) \in \mathrm{G}$, where e_s is the source entity, r_q is the relation, and e_o is the set of all possible answers represents a fact and is called a triple. When a query such as $(e_s, r_q\ ,\ ?)$ is asked, then the results might comes back as $(e_s, r_q,\ e_o) \notin \mathrm{G}$ due to the KG's incompleteness.

7.4 Graph Embedding Using Neural Tensor Networks

In (Lin et. al. 2018), Neural Tensor Networks (NTN) were defined as the tool that utilized tensor objects to predict relationships between the vectors that represent the entities in the graph. The goal of using NTN is to be able to accurately determine if two entities are in a certain relationship (Socher et. al., 2013).

For example, to successfully check the truth and accuracy of the relation $(e_1, \mathrm{R},\ e_2)$= (Margaret Trudeau, was married to, Pierre Trudeau) a vector is used to represent the features of two entities. Then, the vectors are placed on the vector space according to the weights that were assigned to them based on the number of features they have in common. After that, the tensor layer in NTN relates the two entities across multiple dimensions. The final score of how likely the two entities are related is computed using the following function (Socher et. al., 2013). (Socher et. al., 2013):

$$g(e_1, R, e_2) = u_R^T f\left(e_1^T W_R^{[1:k]} e_2 + V_R \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} + b_R\right)$$

This method yields accurate results, but it lacks interpretability (Socher et. al., 2013).

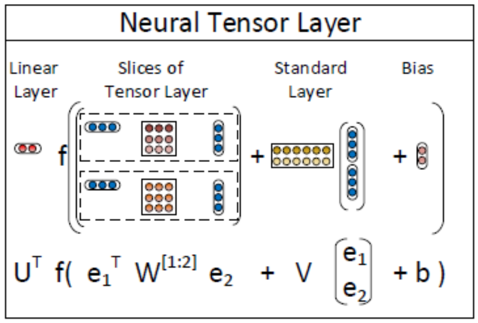

Figure 1: Visualization of NTN (Socher et. al., 2013).

7.5 Interpretability in Machine Learning

The extent to which a cause and effect within a Machine Learning (ML) system can be observed defines the interpretability in ML. In other words, it is the ability to understand what the model has learned to be able to predict what is going to happen when the input or the parameters of the algorithm are changed (Montavon et. al., 2018). The interpretability of a model becomes extremely important when the accuracy of the predicted information must be guaranteed (Montavon et. al., 2018).

Although the Neural Tensor Networks provide highly accurate results in KG completion, their functionality is considered a Blackbox. For this reason, graph embedding using Neural Tensor Networks do not provide a discreet chain of reasoning behind its decisions.

7.6 Reinforcement Learning

7.6.1 Reinforcement Learning Definition

Reinforcement Learning (RL) technique is analogous to dog training. In RL the agent can act in the environment and receive numerical rewards for its actions. The main goal of this process is to learn how to take actions that will maximize the reward.

7.6.2 Reinforcement Learning Set-up

The agent and the environment are the two main components in a Reinforcement Learning model. The agent and environment go through a series of actions and rewards until the environment ends the loop.

This episode begins when the environment sends the agent a $state_t$. The agent will then take an action. Next, the environment sends the reward to the agent along with the next state. Finally, the environment sends the agent a final state to terminate the session.

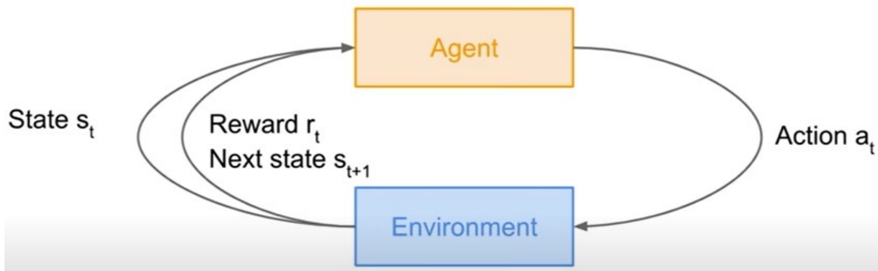

Figure 2: Reinforcement Learning Set-up - Stanford reinforcement learning CS234

7.6.3 Reinforcement Learning Mathematical Formulation in KG

Reinforcement Learning (RL) is mathematically formulated by a Markov Decision Process (MDP). This process is defined by five (5) tuples: ($\boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{R}}, \boldsymbol{\mathcal{P}}, \boldsymbol{\gamma}$), where $\boldsymbol{\mathcal{S}}$ is the set of possible states, $\boldsymbol{\mathcal{A}}$ is the set of possible actions, is the distribution of reward given (state, action) pair, $\boldsymbol{\mathcal{P}}$ is the transition probability over next state given (state, action) pair, and $\boldsymbol{\gamma}$ is the discount factor.

7.6.4 Reinforcement Learning Graph Search MDP

The Markav Decision Process (MDP) was defined in (Sutton and Barto, 1998) for the use of RL in Knowledge Graph's research. There were three sets in the MDP setup: States, Actions and Rewards. Each of them was defined as follows:

States: is the set S that contains all states, where $s_t = \left(e_t, (e_s,\ r_q)\right) \in$ S. S is a tuple where e_t is the entity visited at step t and $(e_s,\ r_q)$ are the source entity and the relation of interest.

Actions: is the set A that contains all possible actions, where $A_t = \{(r', e') | (e_t,\ r', e') \in \mathrm{G}\}$. This definition can be seen as a problem straight away. It specifies that the visited entity must be in the graph for an action to be in set A. However, the Knowledge graphs are incomplete by design. To give the agent a termination option a self loop was added to every A_t (Lin et. al. 2018).

Transition: The following transition function was used δ: S × A → S. This function was identified by $\delta(s_t, A_t) = \delta(e_t, (e_s, r_q), A_t)$.
Rewards: When the agent arrives at the right target entity, it receives a final reward of 1, or 0 otherwise (Lin et. al. 2018). This can be formally defined as: $R_b(s_t) = 1\{(e_s, r_q, e_t) \in G\}$. (1)

7.7 Multi-Hop Knowledge Graph Reasoning method

In this approach the power and accuracy of neural networks was combined with a highly interpretable model using reinforcement learning. This method was introduced to solve the problem of a lack of interpretability of the graph embedding method. In this method, RL has been employed to add a sequential decision-making layer to the highly accurate NTN graph embedding approach. This allowed the model to sequentially hop around the entities in the knowledge graph and to track its actions on the way to a final outcome.

The problem with this method can be seen in the above reward equation. The agent receives a binary reward of "1" if it reaches the right entity in the graph, and zero otherwise. This is a problem because KGs are naturally incomplete. Thus, the false negative search results are rewarded the same way as the true negatives.

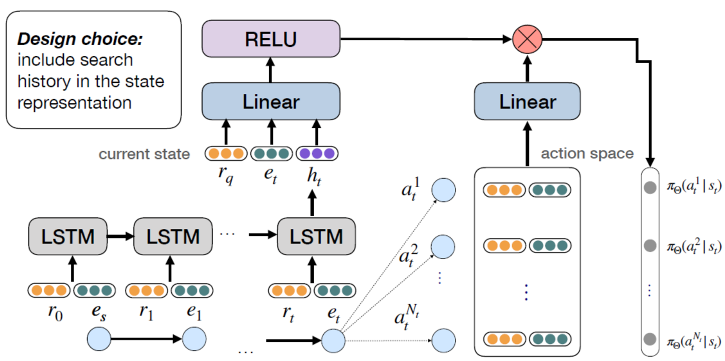

Figure 3: Multi-Hop Knowledge Graph Reasoning with Reward Shaping(Lin et. al. 2018)

7.8 Multi-Hop Knowledge Graph Reasoning with Reward Shaping

(Lin et. al. 2018) proposed a reward shaping method to solve the issue caused by equation (1). The study utilized the existing KG embedding model to compute a reward value for target entities whose correctness is unknown.

Formally speaking, if the agent arrives at the correct entity it receives a reward of "1", otherwise it receive an estimated fact correctness score. This score is estimated by the embedding model that maps E and R in a vector space, and estimates the likelihood of each fact $l = (e_s, r_q, e_t) \in G$ using a composition function $f\ (e_s, r, e_t)$. f is trained by maximizing the likelihood of all facts in G. The scoring function that was used in (Lin et. al. 2018) is: $R(s_T) = R_b(s_T\ ) + \ (1\ -\ R_b(s_T\ ))f\ (e_s, r_q, e_T\ )$.

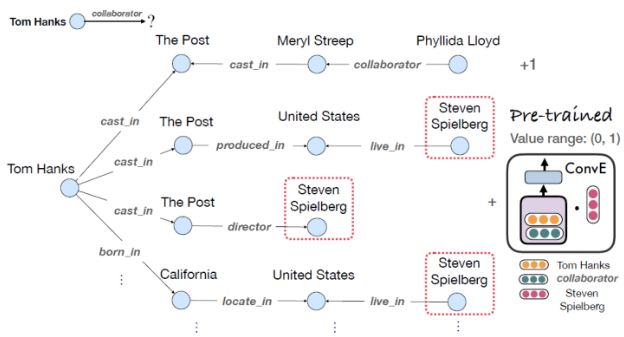

Figure 4: Multi-Hop Knowledge Graph Reasoning with Reward Shaping (Lin et. al. 2018)

8 Approach

We have followed the framework and testing methods identified in (Lin et. al. 2018) and (Sochar et. al., 2013).The code has been cloned from the GitHub repositories available at this link and this link. Below is a detailed overview of our procedures.

8.1 Framework setup

We started by examining the code and checking all the required dependencies.
There were two approaches for code execution. The first one, was by using the docker image and the second one was to unzip the datasets and install the dependencies manually. After considering a variety of options, we decided to run the code using its docker image on Amazon Web Service (AWS).

8.2 Framework Tools and Dependences

To set the required tools and dependences we used the following steps:

8.2.1 Setting-up the Server (Cloud Computing Environment)

a. Created a regular AWS account. Then, created an Educational AWS account that offered limited free access to Amazon Elastic Compute Cloud (EC2) services.
b. Launched a free Tier EC2 Deep Learning instance, with ubuntu16.04 with docker and Nvidia CUDA 9.0 GPU.
c. Installed AWS command Line Interface (CLI) for Windows 64-bits.
d. Installed AWS SAM command Line Interface for Windows 64-bits.
e. Created a user with Administrator access using AWS Identity and Access Management

(IAM)

f. Integrated the AWS user credentials with our remote Windows machine using *aws configure* command to access our EC2 instance using the AWS CLI.
g. Created the 4960Project.pem authentication code to access our EC2 instance remotely.

8.2.2 Setting-up the SDK and Dependencies

h. Installed pycharm professional with free educational licence.
i. connected pycharm to GitHub and cloned the code from (Lin et. al. 2018) repository.
j. Installed AWS Toolkit plug-in in pycharm.
k. Installed Docker Desktop for Windows.
l. Added to the datasets to the graph embedding model.
m. Deployed the code on EC2 instance as an AWS serverless application as a docker image.

8.2.3 Trying different Environments

Before we decided to utilize AWS services, we explored a variety of options and faced some challenges when we tried to run the code. We list them in the following sections.

a. **Microsoft Windows [version 10.0.18363.1440] with Intel (R) HD Graphics 620**

i. **Using Docker Image**

We followed the instructions on the Readme file posted with the original code GitHub repository. As per the instructions, we installed Docker Desktop for Windows. To build the docker image we used the command *docker build -< Dockerfile -t multi_hop_kg:v1.0* which completed successfully. However, when we tried to run the code, we received the following error message: *AttributeError: module 'torch._C' has no attribute '_cuda_setDevice'*. This means that Nvidia CUDA 9.0 GPU is required.

ii. **Manually**

We installed Pytorch 4.0.1 with CUDA Toolkit. Then we unpacked the dataset using *tar xvzf data-release.tgz*. After that we needed to manually install all the dependences, as they were all older versions and did not install automatically. This method did not succeed as for the lack of CUDA 9.0 GPU as well.

b. **macOS Catalina 10.15**

We followed the same procedure mentioned above. Similarly, we received error messages stating that nvidia-docker is not available for Mac operation system.

c. **University of Windsor Linux Server**

We used No Machine to access the University’s Linux system. After setting up the environment by installing Docker and building the docker image, we also received the same error message that says cuda GPU is required.

9 Dataset

9.1 Datasets used in this Experiment

The benchmark FB15k dataset was widely used for assessing the query answering performance of Knowledge Graph completion methods (Akrami et. al., 2020). The FB15k dataset, however, contains many revered relationships. This means that the inverse of many relationships in the training set also exists in the testing set (Akrami et. al., 2020). For instance, if FB15k contains the triple (s, r1, o), it will also contain the triple (o, r2, s) (Akrami et. al., 2020). Basically, this means that the Graph completion method predicting accuracy is not very accurate, as the model would already have the answers. To remedy this issue, FB15k-237 was created as a subset of FB15k. The reversed relationships were removed from the new subset to provide a better simulation of the real-world Knowledge Graphs' structure. To verify the suitability of the method introduced in (Lin et. al. 2018) for real-world KGS, we retested it on the following datasets:

Dataset	Number of Entities	Number of Relations	Number of Facts	Mean	Median
FB15k-237	14,505	237	272,115	19.74	14
NELL-995	75,492	200	154,213	4.07	1
WN18RR	40,945	11	86,835	2.19	2
Kinship	104	25	8,544	85.15	82
UMLS	135	46	5,216	38.63	28

Table 1: Datasets used in this study listed in ascending order bases on the number of relations

9.2 Datasets for future work

Our original intention was to test this method on the COVID-19 pathophysiology KG. This KG was recently generated using coronavirus-related scientific literature by (Domingo-Fernandez et. al., 2020). However, due to time and resource constraints we were not able to use this dataset for testing in this study.

The code required using Nvidia CUDA 9.0 GPU, which was available to us for a limited time via Amazon Web Server. Unfortunately, after running the model three times, using the above benchmark datasets, we received a usage alert from AWS stating that our AWS account has exceeded 85% of the usage limit for AWS Free Tier-eligible services.

We are hoping to test the model on this dataset in the future when more resources are available. Although we did not test the model using this dataset, we can clearly see that its query answering performance deteriorated when it was applied to FB15k-237, which is a dataset that is similar to the real-world datasets. This means that there is a high probability this method will not perform well on a real-world KG such as the COVID-19 pathophysiology KG.

10 Comparison of Tested Models

10.1 Evaluation metrics

1. The results are the average of 3 runs.
2. The results were multiplied by 100.
3. Hits@k, the percentage of possible correct answers, where the rank of e_o is less than k. e_o, is one of the possible correct answers in the set E.
4. Mean Reciprocal Rank (MRR) which is the mean of 1/ rank of e_o for all examples in the test set.
5. Ours(ComplEx) and Ours(ConvE) are the embedding methods that were used.

10.2 Results of testing Multi-Hop Method

FB15k-237

Model	Hits @1	Hits @10	Mean reciprocal Rank (MRR)
Ours(ComplEx)	29.7	57.4	43.5
Ours(ConvE)	35.8	59.5	47.6

WN18RR

Model	Hits @1	Hits @10	Mean reciprocal Rank (MRR)
Ours(ComplEx)	39.7	50.3	45.2
Ours(ConvE)	44.3	54.8	49.5

UMLS

Model	Hits @1	Hits @10	Mean reciprocal Rank (MRR)
Ours(ComplEx)	89.1	94.4	91.7
Ours(ConvE)	95.7	97.2	96.4

Kinship

Model	Hits @1	Hits @10	Mean reciprocal Rank (MRR)
Ours(ComplEx)	84.2	96.1	90.1
Ours(ConvE)	81.4	95.9	88.6

NELL-995

Model	Hits @1	Hits @10	Mean reciprocal Rank (MRR)
Ours(ComplEx)	67.3	80.9	74.1
Ours(ConvE)	67.4	81.7	74.5

10.3 Results of testing Graph Embedding with NTN

FB15k-237

Model	Hits @1	Hits @10	Mean reciprocal Rank (MRR)
ComplEx	30.1	59.8	45.7
ConvE	32.6	65.3	45.1

WN18RR

Model	Hits @1	Hits @10	Mean reciprocal Rank (MRR)
ComplEx	43.0	51.7	41.6
ConvE	44.3	55.2	47.8

UMLS

Model	Hits @1	Hits @10	Mean reciprocal Rank (MRR)
ComplEx	88.6	98.9	93.8
ConvE	97.3	99.7	98.2

Kinship

Model	Hits @1	Hits @10	Mean reciprocal Rank (MRR)
ComplEx	85.2	99.0	92.1
ConvE	83.6	94.1	89.5

NELL-995

Model	Hits @1	Hits @10	Mean reciprocal Rank (MRR)
ComplEx	63.7	84.9	74.3
ConvE	69.6	90.3	79.8

10.4 Results Discussion

Our results confirmed that the Multi-Hop Knowledge Graph Reasoning with Reward Shaping method is not ready to handle a real-world KG, yet. We can see that this method's performance worsened when it was applied on FB15k-237 and WN18RR. The original research paper noted this significant drop in the performance. However, it discusses the performance issue based on the difference between the embedding based methods and the reinforcement path-based model. (Lin et. al. 2018) does not mention the fact that the performance of the reinforcement worsens due to the absence of the inverse relationships in FB15k-237. In addition, our results confirm the original findings in(Lin et. al. 2018) regarding the accuracy of the graph embedding method over the multi-hop reasoning with reward shaping method.

11 Conclusion

Our experiment confirmed the effect of removing the inverse triples from the benchmark datasets on the Multi-hop with reward shaping graph completions method. When the method was tested on the new subset of F15K, it showed a significant decrease in the performance compared to other datasets.
Our literature review revealed that most researchers have overlooked the fact that their results are biased by the impact of the inverse relationships in the benchmark datasets. For example, WN18 and FB15k are widely used for validation in KG completions research. However, they contain a substantial number of inverse relationships (Akrami et. al. 218). This is a very important fact that cannot be ignored as the real-life KGs may not have the inverse relations in them. These relations are supposed to be inferred by the graph completion methods. There is a great possibility that the inverse of the relations in the training set will appear in the testing set. This weakens the learning process of the model.

Thus, we conclude that Multi-hop with reward shaping method is not suitable for real-world KGs yet. In addition, we propose that there should be a shift in the attention of the researchers into developing more realistic datasets that can resemble the real-world KGs. FB15k-237 is a good starting point; however, it is much smaller than real-world KGs (Akrami et. al. 218). Thus, we need to add a bigger and more realistic dataset to the validation package. For instance, considerable efforts are being made to create the COVID-19 pathophysiology KG to support both scientists and researchers. We believe that adding the COVID-19 pathophysiology KG as one of the benchmark datasets could possibly result in a better and more realistic validation system.

12 Future Work

This research is just a starting point for numerous future works which includes, but is not limited to, improving the benchmark datasets. This is done by examining possible candidates from the real-world KGs, such as, COVID-19 pathophysiology and checking the possibility of including it in the benchmark validation datasets package. Another addition to this effort, could be a study about possible ways to improve the accuracy of the multi-hope reasoning with reward shaping. Clearly, it is still not up to the level of accuracy of the embedding methods with NTN.

13 References

[1] Lin, Xi Victoria, Richard Socher, and Caiming Xiong. "Multi-hop knowledge graph reasoning with reward shaping." arXiv preprint arXiv:1808.10568 (2018).

[2] Domingo-Fernández, Daniel, et al. "COVID-19 Knowledge Graph: a computable, multi-modal, cause-and-effect knowledge model of COVID-19 pathophysiology." BioRxiv (2020).

[3] Kejriwal, Mayank. "Knowledge Graphs and COVID-19: Opportunities, Challenges, and Implementation." Special Issue 1-COVID-19 (2020).

[4] Akrami, Farahnaz, et al. "Realistic re-evaluation of knowledge graph completion methods: An experimental study." Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data (2020).

[5] Ioannidis, Vassilis N., Da Zheng, and George Karypis. "Few-shot link prediction via graph neural networks for covid-19 drug-repurposing." arXiv preprint arXiv:2007.10261 (2020).

[6] Zhang, Rui, et al. "Drug repurposing for COVID-19 via knowledge graph completion." Journal of Biomedical Informatics 115 (2021).

[7] Trivedi, Rakshit, et al. "Linknbed: Multi-graph representation learning with entity linkage." arXiv preprint arXiv:1807.08447 (2018).

[8] Socher, Richard, et al. "Reasoning with neural tensor networks for knowledge base completion." Advances in neural information processing systems (2013).

[9] Montavon, Grégoire, Wojciech Samek, and Klaus-Robert Müller. "Methods for interpreting and understanding deep neural networks." Digital Signal Processing 73 (2018).

[10] Richard S. Sutton and Andrew G. Barto. "Reinforcement learning - an introduction. Adaptive computation and machine learning." MIT Press (1998).

[11] Canada, Public Health Agency of. "Government of Canada." Canada.ca, / Gouvernement Du Canada, 20 Jan. 2021, www.canada.ca/en/public-health/services/diseases/2019-novel-coronavirus-infection/health-professionals.html.

[12] "Coronavirus." World Health Organization, World Health Organization, www.who.int/health-topics/coronavirus#tab=tab_1.