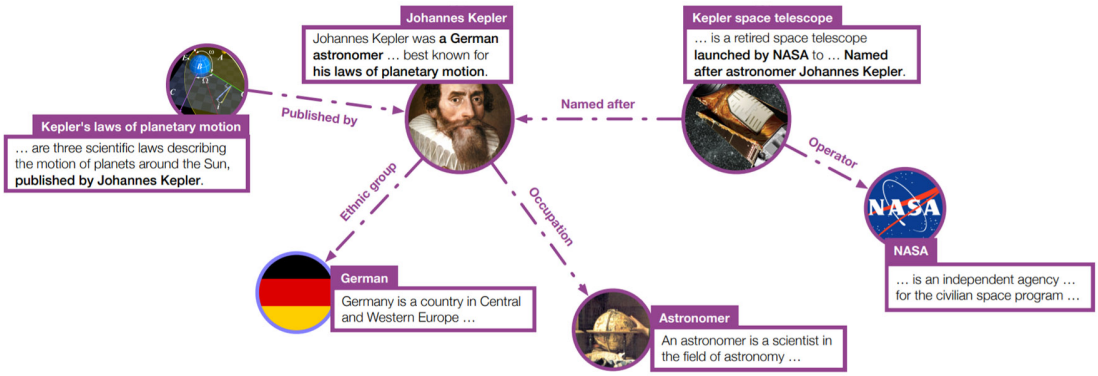
Assignment-2 : KG Completion

Learning on Graphs and its Applications (CSL7870)
Khushal Damor (B21AI018)
October 2024

1. Introduction

We assume the reader to have some basic knowledge about knowledge graphs. If not, then the reader can refer to [1]. Knowledge graphs are multi-relational graphs that express facts about the world by connecting entities via different types of relationships [2]. For example, the graph in Figure 1 is a knowledge graph. The fact that "Johannes Kepler" is an "Astronomer" is expressed by the relation "Occupation" connecting "Johannes Kepler" and "Astronomer".

Figure 1: An example of a knowledge graph [3].



Knowledge graphs are notorious for being massive (millions of nodes and facts) and incomplete (many facts are missing) [1].

The knowledge graph completion problem can be stated as follows : given an enormous and incomplete knowledge graph, can we complete that knowledge graph ?

The following definition of a knowledge graph has been taken from [2] to ensure consistency throughout this text.

A knowledge graph $\mathcal{G}$ is a multi-relational graph consisting of a set of entities $\mathcal{E}$, relationships $\mathcal{R}$, and factual statements in the form of (head, relation, tail) triplets (h, r, t) $\in \mathcal{E} \times \mathcal{R} \times \mathcal{E}$.

2. Dataset

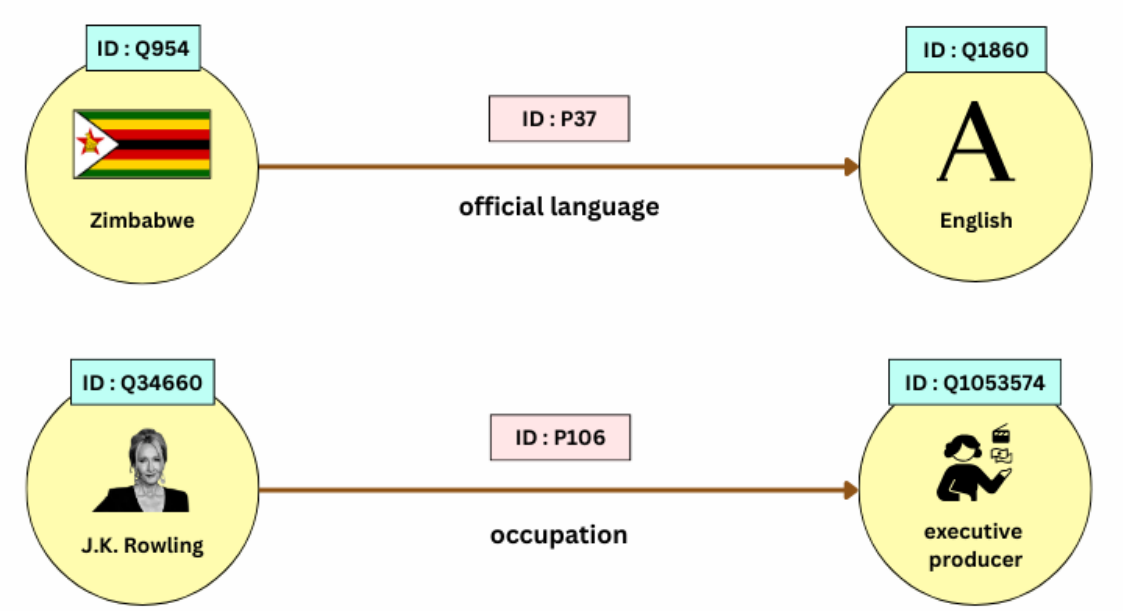
The dataset that we'll be considering for the knowledge graph completion task is CoDEx-S from CoDEx [2]. Statistics of CoDEx-S are given in Table 1.

Table 1: Statistics of the CoDEx-S dataset.

$\lvert\mathcal{E}\rvert$	2034
$\lvert\mathcal{R}\rvert$	42
#Positive Train Triplets	32,888
#Positive Validation Triplets	1827
#Positive Test Triples	1828
Language	English (en)

It contains descriptions of entities (and relationships). These descriptions contain abundant information about entities that can help to represent the relational facts between them [3].

Figure 2: Example triplets from the CoDEx-S dataset.



For example, the description of "English" is given as "West Germanic language originating in England with linguistic roots in French, German and Vulgar Latin". And the description of "J.K. Rowling" is given as "English novelist".

3. KG Completion with TransE

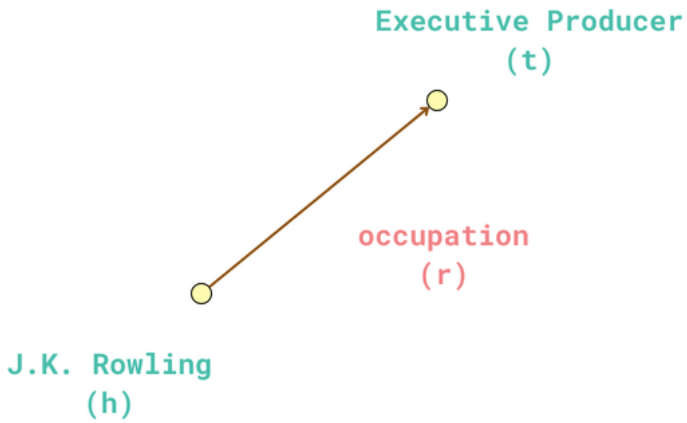
3.1. Methodology

The TransE method is capable of handling multi-relational data and suitable for the knowledge graph completion task [4]. It's an energy-based method to learn the low-dimensional embeddings of the entities. In this method, relationships are represented as translations in the embedding space.

If (h,r,t) is a fact, then the low-dimensional embedding of t should be as close as possible to the low-dimensional embedding of h plus some vector that depends on r.

$$\mathbf{h} + \mathbf{r} \approx \mathbf{t} \tag{1}$$

Figure 3: J.K. Rowling has the occupation of an Executive Producer. In other words, (J.K. Rowling,occupation,Executive Producer) is a fact. Therefore, ideally, the low-dimensional embedding of Executive Producer should be equal to the low-dimensional embedding of J.K. Rowling plus some vector that depends on occupation.



More details about the TransE method can be found in [4]. It's a simple method that performs quite well. However, it fails with symmetric, one-to-many, many-to-one and many-to-many relationships. In the CoDEx-S dataset, 17.46% of the triplets contain a symmetric relation. [2]

3.2. Training

Given a training set (T_{train}) of triplets (h,r,t), our goal is to learn low-dimensional embeddings of the entities and relationships. The embeddings will be from a low-dimensional vector space $\mathbb{R}^k$, where k is a hyperparameter. If (h,r,t) is a fact, then we want $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$. Otherwise, we want $\mathbf{h} + \mathbf{r}$ to be far way from $\mathbf{t}$.

The energy of a triplet is d($\mathbf{h}$ + $\mathbf{r}$,$\mathbf{t}$), where d is a dissimilarity measure that can be chosen to be either the L_1 norm or the L_2 norm (d is a hyperparameter).

To learn such embeddings, the following margin-ranking criterion is minimized over the training set (T_{train}).

$$\mathcal{L} = \sum_{(h,r,t)\in T_{train}} \sum_{(h',r,t')\in T'_{train}} [\gamma + d(\mathbf{h}+\mathbf{r},\mathbf{t}) - d(\mathbf{h}^{'}+\mathbf{r},\mathbf{t}^{'})]_+ \tag{2}$$

where, $[x]_+$ denotes the positive part of x, $\gamma > 0$ is a hyperparameter and

$$T'_{train} = \{(h^{'},r,t)|h^{'} \in \mathcal{E}\} \cup \{(h,r,t^{'})|t^{'} \in \mathcal{E}\} \tag{3}$$

The set of corrupted triplets (T'_{train}) contains triplets from the training set (T_{train}) with either the head or the tail (but not both at the same time) replaced by a random entity from $\mathcal{E}$. The margin-ranking criterion favors lower energy values for the training triplets than for the corrupted triplets. More details about the training process can be found in [4].

3.3. Evaluation

After the training is complete, the performance of the trained model is evaluated on an evaluation set of triplets (T_{eval}). This could be the validation set of triplets or the test set of triplets.

The evaluation protocol is as follows :

(1) For each triplet in the evaluation set (T_{eval}), its head is removed and replaced with each entity from $\mathcal{E}$. This will produce a set of corrupted triplets.

(2) From this set, all triplets that are already part of the knowledge graph are removed/filtered (excluding the positive evaluation triplet of interest).

(3) The energy values of these filtered corrupted triplets are calculated. After that, they are sorted in ascending order and the rank of the positive evaluation triplet is noted. Then, two values are stored. First, the reciprocal of the rank. Second, a boolean value that is true only if the rank is less than or equal to K, where K is a non-negative integer that is chosen appropriately.

(4) The above process is repeated but this time the tail is replaced instead of the head.

(5) After steps (1) to (4) are completed for each triplet in the evaluation set (T_{eval}), the Mean Reciprocal Rank (MRR) value is reported along with the Hits@K value.

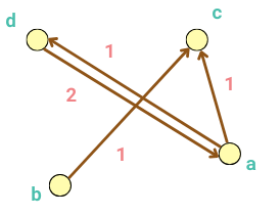
A toy example illustrating the evaluation process is given in Figure 4. Step (2) is also known as filtering. It's used to ensure that known positive triples do not artificially lower the performance by getting ranked above the positive evaluation triplet. More details about the evaluation process can be found in [4].

Figure 4: A toy example to illustrate the evaluation process.

E = {a,b,c,d}, R = {1,2}

T_{train} = {(b,1,c),(d,2,a),(a,1,d)}

T_{eval} = {(a,1,c)}



Now, consider the following positive evaluation triplet :

(a,1,c)

First, operate on the head.

1.Replacing a	2.Filtering	3.Sorted Energy	
(a,1,c)	(a,1,c)	(d,1,c)	
(b,1,c)	(d,1,c)	(a,1,c)	Rank = 2
(d,1,c)	(c,1,c)	(c,1,c)	
(c,1,c)			

Then, operate on the tail.

1.Replacing c	2.Filtering	3.Sorted Energy	
(a,1,c)	(a,1,c)	(a,1,c)	Rank = 1
(a,1,d)	(a,1,b)	(a,1,b)	
(a,1,b)	(a,1,a)	(a,1,a)	
(a,1,a)			

Finally, report the evaluation metrics.

Mean Reciprocal Rank = 0.75
Hits@1 = 0.50

4. Experiments

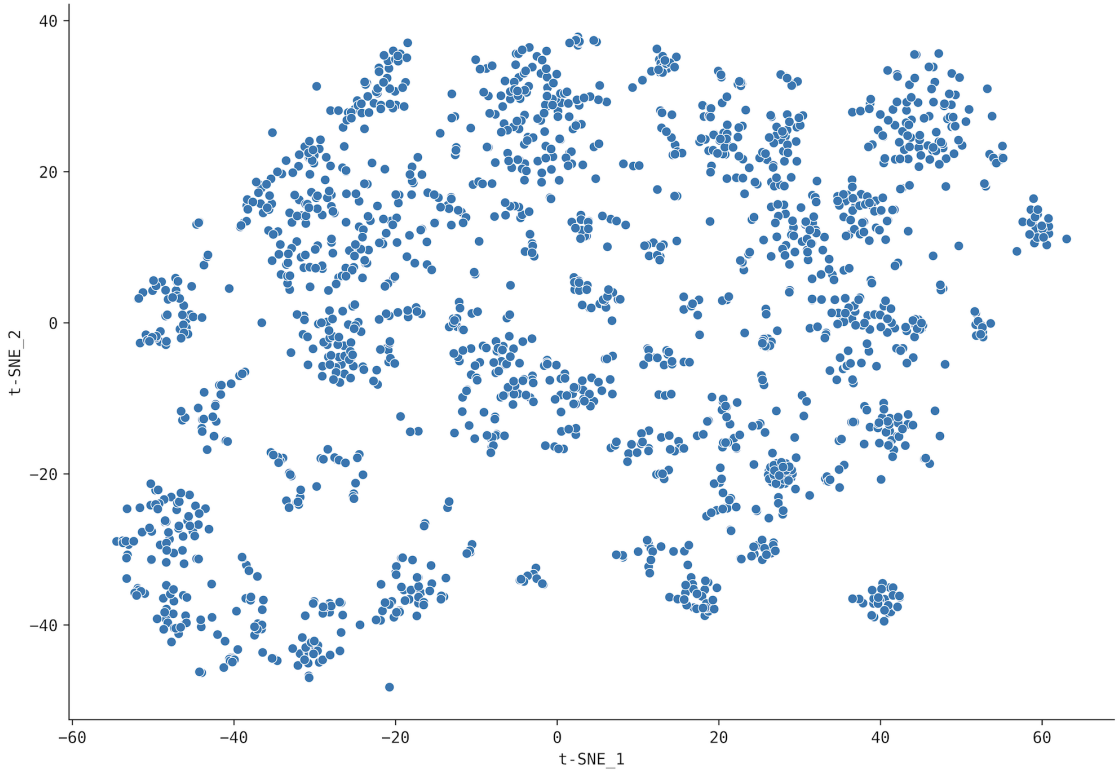
We performed several experiments to evaluate the performance of the TransE method on the knowledge graph completion task (specifically on the CoDEx-S dataset).

These experiments were performed on a Tesla T4 GPU (with 16 GB of GDDR6 VRAM) that was available on the Google Colaboratory platform.

For training, we selected the Adam optimizer and set its learning rate to 0.001. The β_1 and β_2 values were set to 0.9 and 0.999, respectively. The margin (γ) value was set to 0.5. The dissimilarity measure was set as the L_4 norm. The batch size was set to 128. The margin-ranking loss was minimized for 250 epochs. The best model was selected using the MRR value on the validation set of triples.

The relationship embeddings were initialized randomly as given in [4]. For the entity embeddings, we experimented with two different initialization methods. First, random initialization. Second, LLM-Based (Large Language Model) initialization. In LLM-Based initialization, we took the descriptions that came with each entity and passed them to a pre-trained sentence transformer [5] to get a 768-dimensional embedding. Then we used these embedding to initialize the embeddings of the entities. This also implies that we have taken the value of k as 768 i.e. the entity embeddings and relationship embeddings come from $\mathbb{R}^{768}$.

Figure 5: The embeddings of each entity (from CoDEx-S) generated from its textual description by a pre-trained sentence transformer. Visualized in 2-dimensions with t-SNE.



4.1. Results

From Table 2, we can observe that results are almost identical for both of the initialization methods (for the entity embeddings). This could mean that the descriptions of the entities are not useful in the modeling process and/or the quality of the embeddings generated by the pre-trained sentence transformer are not good. Table 3 and Table 4 illustrate few example predictions on the test set of triplets. These also suggest that LLM-Based initialization of the entity embeddings don't make any significant difference on the results.

Table 2: Results of the knowledge graph completion task on the CoDEx-S dataset. x indicates that x is the best value in that column.

Initialization of Entity Embeddings	**MRR**	**Hits@1**	**Hits@3**	**Hits@10**
Random	0.292127	0.170131	0.336159	0.542122
LLM-Based	0.271215	0.151258	0.310995	0.513402

Table 3: Example predictions on the test set of triplets (Random initialization of entity embeddings). The predictions are sorted in ascending order of the energy of the implied triplet. t indicates that t is the true tail of the test triplet and the implied triplet is a part of the knowledge graph. t indicates that the implied triplet is a part of the knowledge graph. For other tails, we don't know whether the implied triplet is a part of the knowledge graph or not.

Head	**Relationship**	**Predicted Tails (Top-3)**
Gaspard Monge	country of citizenship	France , Poland, Argentina
Leon Russell	occupation	musician , singer-songwriter , guitarist
Mos Def	occupation	actor , singer , composer
Max Frisch	genre	play , novel, short story
Bulat Okudzhava	location of formation	location of formation, Moscow, Saint Petersburg

Table 4: Example predictions on the test set of triplets (LLM-Based initialization of entity embeddings). The predictions are sorted in ascending order of the energy of the implied triplet. t indicates that t is the true tail of the test triplet and the implied triplet is a part of the knowledge graph. t indicates that the implied triplet is a part of the knowledge graph. For other tails, we don't know whether the implied triplet is a part of the knowledge graph or not.

Head	**Relationship**	**Predicted Tails (Top-3)**
Gaspard Monge	country of citizenship	France , Argentina, Lebanon
Leon Russell	occupation	singer , pianist , guitarist
Mos Def	occupation	actor , film producer , film actor
Max Frisch	genre	play , essay, novel
Bulat Okudzhava	location of formation	Moscow, Saint Petersburg, Los Angeles

4.2. Impact of LLM Initialization

Table 5 and Table 6 together indicate that with both of the initialization methods (for the entity embeddings), the top performing relationships are more or less the same suggesting that LLM-Based initialization of the entity embeddings may not have any significant impact on the modeling process.

Table 5: Top performing relationships (Random initialization of entity embeddings) on the basis of MRR with triplet count greater than or equal to 5. r indicates that r is also top performing (in the top-5) when the entity embeddings are initialized with an LLM.

Relationship	Count	MRR	Hits@1	Hits@3	Hits@10
part of	6	0.57	0.50	0.58	0.66
continent	10	0.53	0.40	0.65	0.70
instrument	71	0.45	0.35	0.50	0.66
member of political party	8	0.43	0.31	0.56	0.62
member of	270	0.42	0.28	0.47	0.74

Table 6: Top performing relationships (LLM-Based initialization of entity embeddings) on the basis of MRR with triplet count greater than or equal to 5. r indicates that r is also top performing (in the top-5) when the entity embeddings are initialized randomly.

Relationship	Count	MRR	Hits@1	Hits@3	Hits@10
part of	6	0.58	0.41	0.75	0.83
continent	10	0.53	0.40	0.70	0.70
ethnic group	19	0.50	0.42	0.52	0.68
member of political party	8	0.47	0.43	0.50	0.50
official language	14	0.41	0.32	0.50	0.57

4.3. Hyperparameter Tuning

For hyperparameter tuning, we explored the space of two hyperparameters : the margin (γ) value and the dissimilarity measure (d). We picked γ from [0.1,0.5,3.0] and d from [L_1,L_2,L_4]. For each of these experiments, we initialized the embeddings of the entities with the LLM-based method. The training was done for 100 epochs.

Results of the hyperparameter tuning experiments are given in Table 7. We can observe that ($\gamma = 0.5$) produced the best results. And using the L_4 norm led to much better results when compared to the L_1 norm and the L_2 norm.

Therefore, the best hyperparameter set is $\{\gamma = 0.5, L_4 \text{ Norm}\}$. These are the exact hyperparameters that we selected initially and the results for which are also given in Table 2.

Table 7: Results of hyperparameter tuning. x indicates that x is the best value in that column.

γ	Norm	MRR	Hits@1	Hits@3	Hits@10
0.5	L_1	0.025463	0.000000	0.024617	0.065919
0.5	L_2	0.245268	0.140591	0.271061	0.463895
0.5	L_4	0.276428	0.162746	0.307166	0.510667
1.0	L_1	0.024761	0.000000	0.028720	0.067013
1.0	L_2	0.192349	0.099289	0.203501	0.397976
1.0	L_4	0.251830	0.139497	0.291028	0.476477
3.0	L_1	0.017264	0.000821	0.018873	0.045405
3.0	L_2	0.082378	0.027626	0.064278	0.177243
3.0	L_4	0.165950	0.087527	0.167943	0.351477

5. Conclusion

We have found that initializing embeddings of entities with an LLM does not show any significant improvement to the knowledge graph completion task using the TransE method, specifically on the CoDEx-S dataset. Generating embeddings for millions of entities with an LLM requires computational resources. Even though it's a one time investment, it may not be feasible in low-resource settings. Therefore, random initialization of the entity embeddings should be preferred.

As for the knowledge graph completion task on the CoDEx-S dataset, we were able to achieve an MRR of 0.292 with the TransE method (Random Initialization of entity embeddings), which is lower than 0.354 (reported in [2]).

TransE was introduced in 2013 [4]. Since then, several new methods have been proposed like TransR [6], ComplEx [7], DistMult [8], RotatE [9], R-GCN [10] etc. (and many more) for the task of knowledge graph completion. Future work could experiment with these methods along with the LLM-Based initialization of entity embeddings. So far we have only considered using the descriptions of entities and not of the relationships. Future work could analyze whether the descriptions of relationships can be used with or without the entity descriptions in the modeling process for the knowledge graph completion task with the TransE method or any other method.

References

[1] Jure Leskovec. *CS224W : Knowledge Graph Embeddings*. https://web.stanford.edu/class/cs224w/slides/10-kg.pdf. [Online; accessed 31-October-2024]. 2024.

[2] Tara Safavi and Danai Koutra. "CoDEx: A Comprehensive Knowledge Graph Completion Benchmark". In: *CoRR* abs/2009.07810 (2020). arXiv: 2009.07810. URL: https://arxiv.org/abs/2009.07810.

[3] Xiaozhi Wang et al. "KEPLER: A unified model for knowledge embedding and pre-trained language representation". In: *Transactions of the Association for Computational Linguistics* 9 (2021), pp. 176–194.

[4] Antoine Bordes et al. "Translating Embeddings for Modeling Multi-relational Data". In: *Advances in Neural Information Processing Systems*. Ed. by C.J. Burges et al. Vol. 26. Curran Associates, Inc., 2013. URL: https://proceedings.neurips.cc/paper_files/paper/2013/file/1cecc7a77928ca8133fa24680a88d2f9-Paper.pdf.

[5] Kaitao Song et al. *MPNet: Masked and Permuted Pre-training for Language Understanding*. 2020. arXiv: 2004.09297 [cs.CL]. URL: https://arxiv.org/abs/2004.09297.

[6] Yankai Lin et al. "Learning entity and relation embeddings for knowledge graph completion". In: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. AAAI'15. Austin, Texas: AAAI Press, 2015, pp. 2181–2187. ISBN: 0262511290.

[7] Théo Trouillon et al. *Knowledge Graph Completion via Complex Tensor Factorization*. 2017. arXiv: 1702.06879 [cs.AI]. URL: https://arxiv.org/abs/1702.06879.

[8] Bishan Yang et al. *Embedding Entities and Relations for Learning and Inference in Knowledge Bases*. 2015. arXiv: 1412.6575 [cs.CL]. URL: https://arxiv.org/abs/1412.6575.

[9] Zhiqing Sun et al. *RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space*. 2019. arXiv: 1902.10197 [cs.LG]. URL: https://arxiv.org/abs/1902.10197.

[10] Michael Schlichtkrull et al. *Modeling Relational Data with Graph Convolutional Networks*. 2017. arXiv: 1703.06103 [stat.ML]. URL: https://arxiv.org/abs/1703.06103.