Link Prediction Using Graph Embeddings

Eleventh European Big Data Management & Analytics Summer School (eBISS 2023) Barcelona, Spain

Ahmad Ahmad - a.schigdar@gmail.com
Rishika Gupta - rishikagupta1999@gmail.com

1. RESEARCH PROBLEM

Problem Statement: Knowledge Graphs (KGs) have gained popularity in industry and academia, leading to extensive research on extracting information from various sources. But, even the most advanced KGs suffer from incompleteness, prompting research efforts in the field of Link Prediction (LP).

Why Graph Embeddings?[1]

- Enhancing Machine Learning on graphs
- Efficient storage and retrieval for processing
- Aid in simpler and faster computations

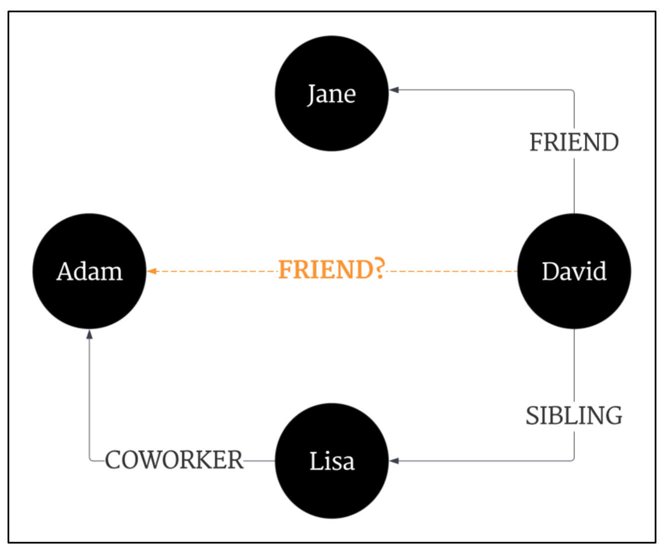

Fig. 1: Simple LP Example

Link Prediction: It is a challenging problem in several domains, for instance as shown in Figure 1, can suggest new friendships in social networks[1].

2. METHODOLOGY

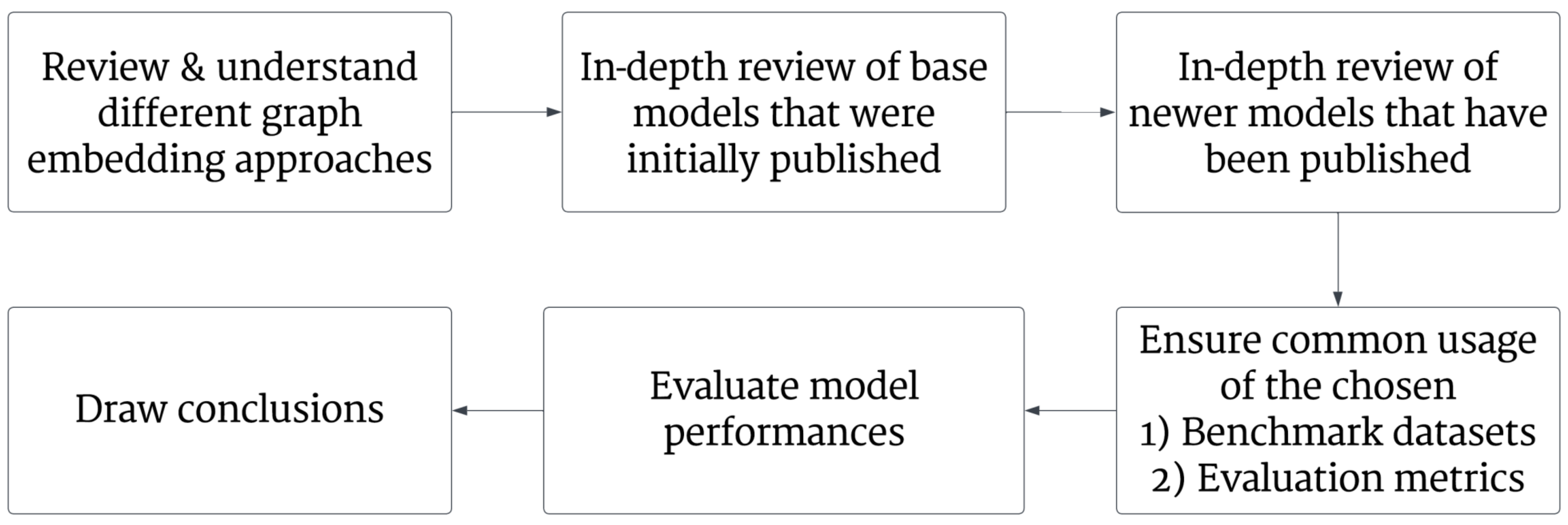

Fig. 2: Methodology Overview

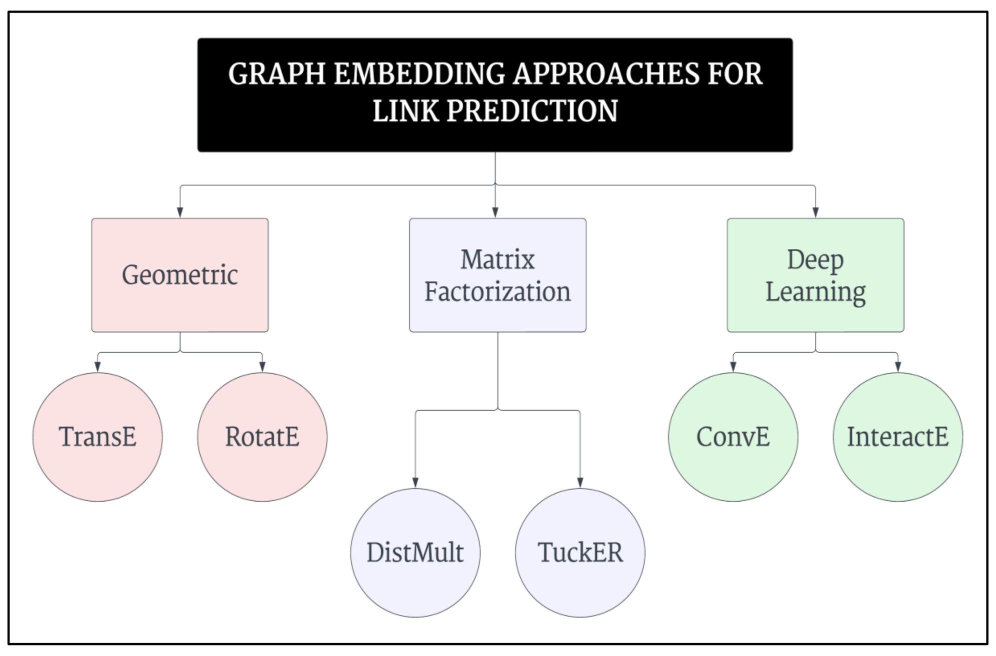

Fig. 3: Chosen Approaches & Models

Geometric[2]**:** Relations are taken as geometric operations.

$$\phi(h,r,t) = \delta(\tau(h,r),t)$$

τ: spatial transformation of h and r
δ (L1/L2): distance between (h,r) & t

Matrix Factorization[2]**:** Viewed as 3D matrices, KGs are decomposed into collection of low-dimension vectors.

Deep Learning[2]**:** Using weights and biases (or even complex neural networks) with the input relations, embeddings are learnt.

Benchmark Datasets[2]**:**

(1) Freebase (FB15k-237): Covers 14.5k entities and 237 relations from different topics like people, places, movies, and organizations, etc.

(2) WordNet (WN18RR): Includes 40k entities and 11 relations that captures different semantic relationships between words and concepts, such as hypernymy, hyponymy, etc.

These two datasets exclude inverse relation patters to deal with data leakage as compared to their original versions (FB15k and WN18).

Evaluation Metrics[2]**:**

(1) Mean Reciprocal Ratio (MRR): Represents the reciprocal of the average ranks for correct prediction on missing links.

$$MRR = \frac{1}{|Q|}\sum_{q \in Q}\frac{1}{q}$$

(2) Hits@K (H@10): Represents the ratio of predictions with a rank equal to or lower than a given threshold K. For example, Hits@10 measures the accuracy of the top ten predictions.

$$H@10 = \frac{|q \in Q : q \leq 10|}{Q}$$

3. TOP MODELS SUMMARY

RotatE

- Represents each relation as a rotation (Figure 5) from the source to its target entity within the complex vector space (instead of a translation in TransE like in Figure 4).
- Able to infer multiple types of relational patterns which are symmetry / anti-symmetry, inversion, as well as composition.
- Distance function for each triple (h, r, t) is denoted as: $d_r(h,t) = ||h \circ r - t||$ where $|r_i| = 1$, and $\circ$ is the Hadmard product.

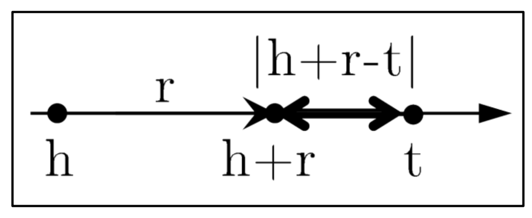

Fig. 4: TransE translation[3]

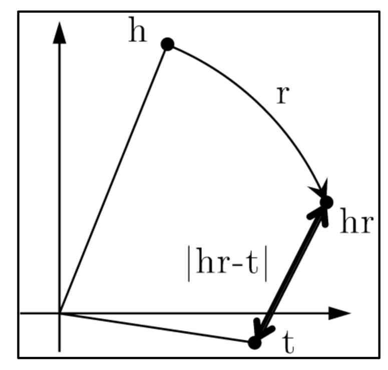

Fig. 5: RotatE rotation[3]

TuckER

- Utilizes Tucker decomposition for the binary tensor representation of triples.
- Score ϕ is obtained using true triples $\phi(e_s,r,e_o) = W \times_1 e_s \times_2 w_r \times_3 e_o$ where $e_s, e_o \in R^{d_e}, w_r \in R^{d_r}, W \in R^{d_e * d_r * d_e}$ and W is the core tensor, in order to accurately score all the missing triples.
- Learnt knowledge is encoded in embeddings & in core tensor (W as in Figure 6), unlike other simpler models like DistMult, ComplEx, etc.

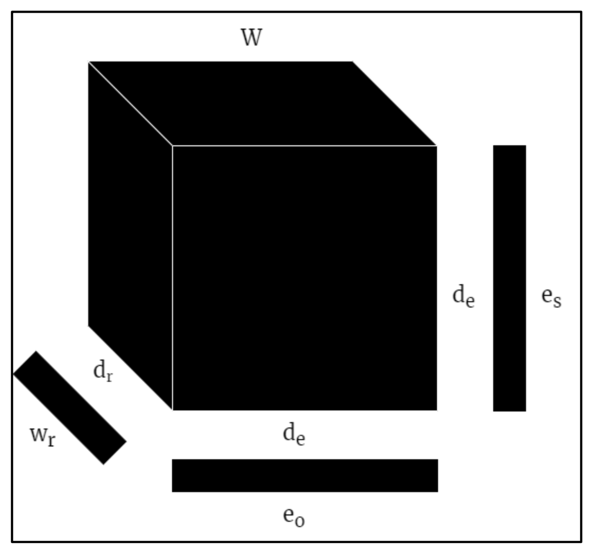

Fig. 6: TuckER architecture[4]

4. PERFORMANCE & RESULTS

Approach ↓	Dataset → / Metric → Model ↓	FB15k-237 MRR	FB15k-237 H@10	WN18RR MRR	WN18RR H@10
Geometric	TransE	0.294	0.354	0.226	0.501
	RotatE	0.338	0.533	**0.476**	**0.571**
Matrix Factorization	DistMult	0.241	0.419	0.430	0.490
	TuckER	**0.358**	**0.544**	0.470	0.526
Deep Learning	ConvE	0.325	0.501	0.430	0.520
	InteractE	0.354	0.535	0.463	0.528

Table 1: Consolidated Result Summary[3], [4], [5]

- **TuckER** can infer composition patterns very well which is the main relation pattern in FB15k-237.
- In WN18RR, **RotatE** is more dominant, due to its ability to infer symmetrical patterns very well.

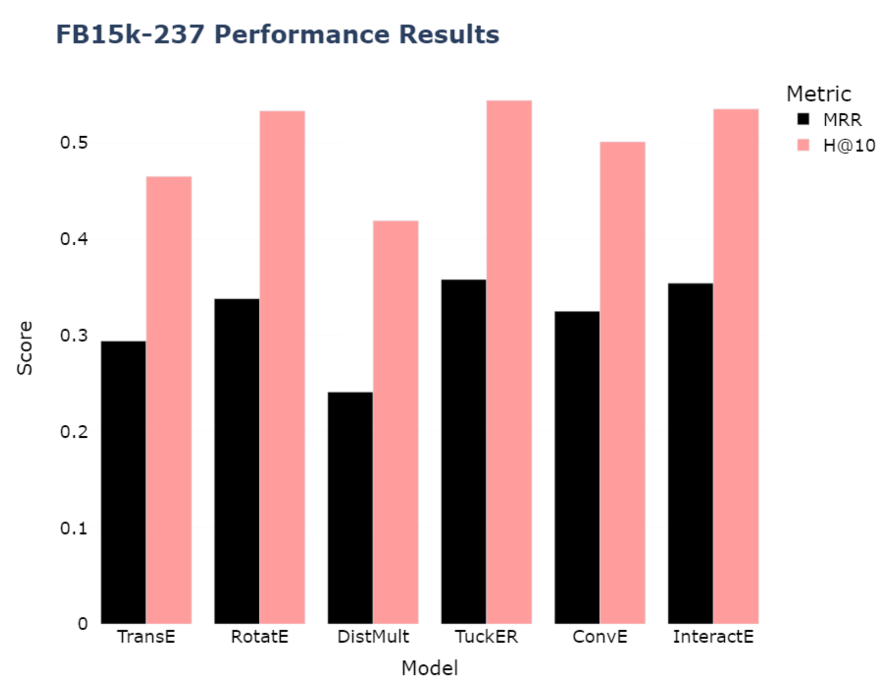

Fig. 7: FB15k-237 Results Comparison

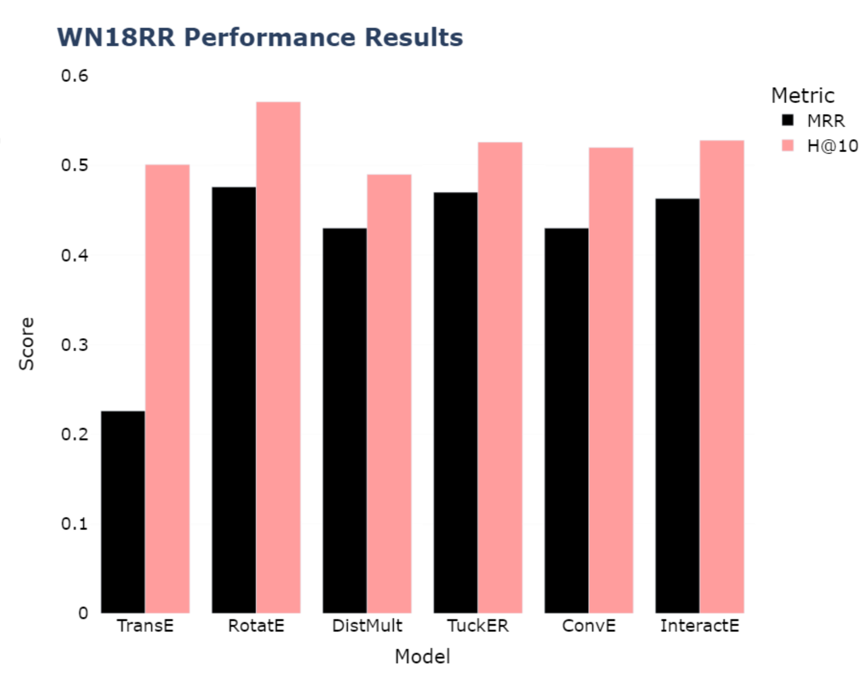

Fig. 8: WN18RR Results Comparison

5. CONCLUSIONS

- Presented six different state-of-the-art link prediction techniques used to infer missing links in knowledge graphs.
- Compared & analyzed performance for all the models on two benchmark datasets (FB15k-237 & WN18RR) using the evaluation metrics, MRR & Hits@10.
- Using approaches such as binary tensor decomposition (TuckER), complex vector spaces with rotations (RotatE), or deep learning approaches with increased feature interactions (InteractE) can infer missing links well.
- Furthermore, RotatE performs exceptionally well, while maintaining a linear space complexity as compared to TuckER and InteractE.

REFERENCES

[1] Godec, P. (2019, June 26). Graph embeddings - the summary. Medium. https://towardsdatascience.com/graph-embeddings-the-summary-cc6075aba007

[2] Rossi, A., Barbosa, D., Firmani, D., Matinata, A., & Merialdo, P. (2021). Knowledge graph embedding for link prediction: A comparative analysis. ACM Transactions on Knowledge Discovery from Data (TKDD), 15(2), 1-49.

[3] Sun, Z., Deng, Z. H., Nie, J. Y., & Tang, J. (2019). Rotate: Knowledge graph embedding by relational rotation in complex space. arXiv preprint arXiv:1902.10197.

[4] Balažević, I., Allen, C., & Hospedales, T. M. (2019). Tucker: Tensor factorization for knowledge graph completion. arXiv preprint arXiv:1901.09590.

[5] Vashishth, S., Sanyal, S., Nitin, V., Agrawal, N., & Talukdar, P. (2020, April). Interacte: Improving convolution-based knowledge graph embeddings by increasing feature interactions. In Proceedings of the AAAI conference on artificial intelligence (Vol. 34, No. 03, pp. 3009-3016).