Project proposal
Background Knowledge for TuckER

Joss Rakotobe, Haoyu Zhang

April 2020

A knowledge graph is a dataset of relations between entities coming from real-word knowledge. It is represented as an oriented labeled graph where the vertices are entities and the edges are relations. Since they are typically incomplete, there is an active area of research trying to make link predictions on how to complete it. Several models have recently come to light to tackle this problem. In particular, Balazevic et al. [1] proposed a state-of-the-art model called TuckER, based on the Tucker decomposition.

Their idea was to represent a dataset with a one-hot encoding tensor $\mathcal{T} \in \mathbb{R}^{n_e \times n_r \times n_e}$ with n_e (resp. n_r) the number of entities (resp. relations) and model it in the form $\mathbf{W} \times_1 \mathbf{E} \times_2 \mathbf{R} \times_3 \mathbf{E}$, with $\mathbf{E}$ (resp. $\mathbf{R}$) the entity (resp. relation) embedding, and $\mathbf{W}$ the core tensor. They defined a scoring function $\phi(s, r, o) = \mathbf{W} \times_1 e_s \times_2 w_r \times_3 e_o$, where e_s, w_r and e_o are the embedding vectors of the entity subject s, the relation r and the entity object o respectively.

Our objective is to improve the performance of this model. Sometimes, some relations are known to have particular properties such as symmetry, anti-symmetry or being the inverse of another relation. We refer to this type of knowledge as "background knowledge". We believe that enforcing these properties into a model should improve its performance. This kind of work was already performed by Minervini et al. [2] on three linear models, namely TransE, DistMult and ComplEx, and also by Kazemi et al. [3] on SimplE. The results they obtained support our assumption.

We will investigate different ways to make use of background knowledge through parameter tying into a TuckER model. Our main focus will be to incorporate symmetric and anti-symmetric relations and compare the efficiency of the new models on the WordNet (WN18-RR) dataset.

Our first method to incorporate background knowledge will be to put a hard constraint on the core tensor $\mathbf{W}$ and the relation embeddings of the TuckER model.
Formally, we will force the first d_a mode-2 slices of $\mathbf{W}$ to be anti-symmetric,

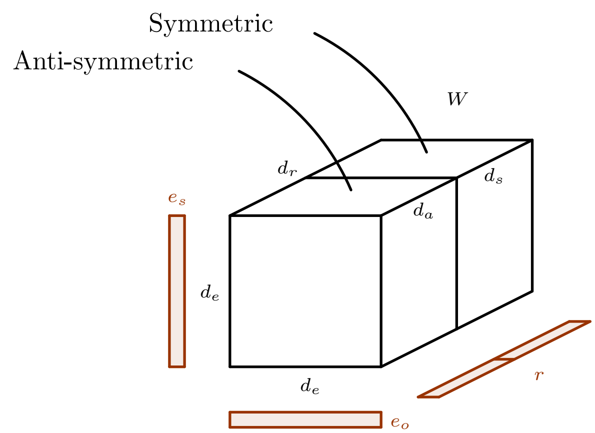

Figure 1: Visualization of the new TuckER architecture.

its last d_s mode-2 slices to be symmetric, and for each symmetric (resp. anti-symmetric) relation, we'll force the first d_a (resp. last d_s) components of its embedding to be zero (see Figure). By doing so, we can see that the mode-2 product of the core tensor with the relation embedding will be a symmetric (resp. anti-symmetric) matrix. For a general relation, it suffices to see that any matrix can be formed by a linear combination of a symmetric and an anti-symmetric matrix.

Our second method will be to put a hard constraint on the scoring function by setting a new scoring function

$$\phi_\lambda(s,r,o) = \begin{cases} \frac{1}{2}(\phi(s,r,o) + \phi(o,r,s)) & \text{if r is symmetric} \\ \frac{1}{2}(\phi(s,r,o) - \phi(o,r,s)) & \text{if r is anti-symmetric} \\ \phi(s,r,o) & \text{otherwise} \end{cases}$$

We will use the algorithm proposed by Balazevic et al.[1] as our stepping stone.

References

[1] I. Balazevic, C. Allen, and T. M. Hospedales, "Tucker: Tensor factorization for knowledge graph completion," *CoRR*, vol. abs/1901.09590, 2019.

[2] P. Minervini, L. Costabello, E. Muñoz, V. Novácek, and P.-Y. Vandenbussche, *Regularizing Knowledge Graph Embeddings via Equivalence and Inversion Axioms*, pp. 668–683. 01 2017.

[1] `https://github.com/ibalazevic/TuckER`

[3] S. M. Kazemi and D. Poole, "Simple embedding for link prediction in knowledge graphs," in *Advances in Neural Information Processing Systems 31* (S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, eds.), pp. 4284–4295, Curran Associates, Inc., 2018.