Knowledge Graph Inference using Tensor Embedding

Ankur Padia[1] *, **Konstantinos Kalpakis**[2], **Francis Ferraro**[2], **Tim Finin**[2]
[1]Philips Research North Americas, Cambridge, MA 02141, USA
[2]University of Maryland, Baltimore County, Baltimore, MD 21250, USA
ankur.padia@philips.com, {kalpakis, ferraro, finin}@umbc.edu

1 Introduction

This extended abstract describes work reported in the Journal of Web Semantics (Padia et al. 2019).

Axiom based inference provides a clear and consistent way of reasoning to add more information to a knowledge graph. However, constructing a set of axioms is expensive and requires domain expertise, time, and money. It is also difficult to reuse or adapt a set of axioms to a knowledge graph in a new domain or even in the same domain but using a slightly different representation approach.

Representation learning (Bengio, Courville, and Vincent 2013) introduces a way to augment or even replace manually constructed ontology axioms and rules by using knowledge graph instances to discover common patterns and then apply them to suggest changes to the graph. The changes are often in the form of adding missing types and relations, but can also include schema modifications, removing incoherent instances, merging sets of instances describing the same real-world entity, or adding relation probabilities.

One popular approach for representation learning is based on learning how to embed a graph's entities and relations into a real-valued vector space, allowing both to be represented by dense, real-valued vectors. The entity and relation embeddings can be learned either independently or jointly, and then used to predict additional relations that are missing. Jointly learning the embeddings allows each to enhance the other (Nickel, Tresp, and Kriegel 2011).

There are several models that learn embedding of entities and relations to perform inference. Some use existing schemas (Krompass, Baier, and Tresp 2015) to regularize the quality of the embedding. These often give better performance compared to those that do not use schemas, as in Nickel (2011). However, schema-based embedding methods suffer from the above-mentioned limitations. We have developed a family of novel methods that improves the quality of the embeddings without using pre-defined schemas (Padia et al. 2019; Padia 2019a).

We divide statistical models that infer additional knowledge graph facts into two categories: *link ranking* systems that provide a ranked list of the top-K candidates and *link prediction or classification* systems that determine if a new fact holds or not. There are several approaches to link ranking (Socher et al. 2013), all of which involve an auxiliary problem of determining a threshold, either globally or per relation, that separates plausible from implausible relations. Since we are only interested in extending a knowledge graph with relations that are likely to hold (what we call facts), we designed an approach to solve it directly. Thus we have the fact or link prediction task: given a knowledge graph, learn a model that can classify relation instances that are very likely to hold. This task is more specific than link ranking and more directly solves an important problem.

We improve the quality of the relation and entities representations using data-driven constraints, hence our approach can be used when ontological axioms are not available or are expensive to create. We measured the quality of the learned embeddings by comparing our approaches with previous non-schema based methods as well as with neural models and found improvement ranging from 5% to 50%. We demonstrated its broad applicability using eight real-world data sets covering human language, medical data, and general world knowledge that are available online (Padia 2019b). This work makes three main contributions: it (1) provides a family of representation learning algorithms and an extensive analysis on eight datasets; (2) yields better results than existing tensor and neural models; and (3) includes a provably convergent factorization algorithm.

We use the initial knowledge graph to pre-compute a similarity matrix C for the relations that will help constrain the learning of embeddings. To better understand the idea, consider the WordNet knowledge graph where entities are words that are connected with relations like *hypernym* and *similar*. The graph has no schema and there are many missing relations. We create a similarity matrix quantifying the similarity between two relations as the number of overlapping words. The cells in Figure 1 show the number of subjects or objects that are shared by the relations, with darker cells indicating a smaller overlap.

2 Models Using Axiomatic-like constraints

We represent a multi-relational knowledge graph of N_r binary relations and N_e entities by the order-3 tensor $\mathcal{X}$ of dimension $N_e \times N_e \times N_r$. This binary tensor is often both very large and sparse. Our goal is to construct dense, infor-

*This work was done when the first author was doing Ph.D. at the University of Maryland, Baltimore County and before joining Philips Research North America.

Model Name	Kinship	UMLS	WN18	FB13	DB10	Framenet	WN18RR	fb15-237
Previous tensor factorization models								
RESCAL	93.24	**88.53**	62.13	65.37	61.27	82.54	66.63	92.56
NN-RESCAL	92.19	88.37	83.93	79.13	**81.72**	82.6	68.49	93.03
Linear/Quadratic Regularized/Constrained tensor factorization models								
LR	**93.99**	88.22	81.86	80.07	80.79	78.11	69.15	90.00
QR	93.89	88.11	84.41	79.12	80.47	82.34	66.73	**93.07**
LC	92.87	84.71	80.18	75.79	80.67	73.64	66.46	81.88
QC	93.84	86.17	**91.07**	**85.15**	81.69	**86.24**	**72.62**	86.47

Table 1: AUC for all models; LR and LC are linear regularized/constrained models; QR and QC are quadratic regularized/constrained.

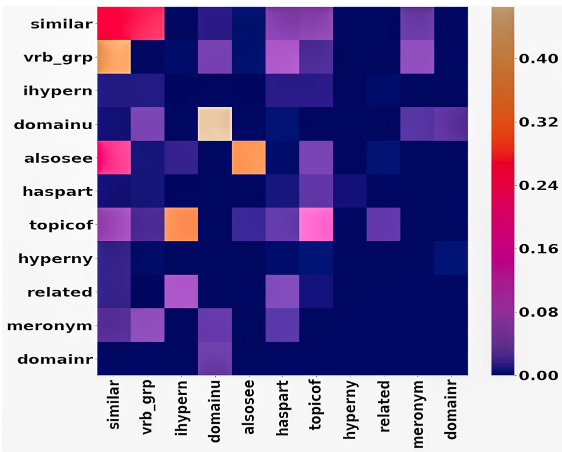

Figure 1: The computed similarity matrix for the WordNet dataset using the transitivity metric.

mative, p-dimensional embeddings, where p is much smaller than either the number of entities or relations. We represent the collection of p-dimensional entity embeddings by $\mathcal{A}$, the collection of relation embeddings by $\mathcal{R}$, and the similarity matrix by C. Mathematically, our objective is to reconstruct each of the k relation slices of $\mathcal{X}$, $\mathbf{X}_k$, as the product

$$\mathbf{X}_k \approx \mathcal{A}\mathbf{R}_k\mathcal{A}^{\intercal}. \tag{1}$$

All of the models realize the objective function structure.

$$\min_{\mathcal{A},\mathcal{R}} \underbrace{f(\mathcal{A},\mathcal{R})}_{\text{reconstruction loss}} + \overbrace{g(\mathcal{A},\mathcal{R})}^{\text{numerical regularization of the embeddings}} + \underbrace{f_s(\mathcal{A},\mathcal{R},C)}_{\text{capturing axiomatic-like pattern}}. \tag{2}$$

The first term of (2) reflects each of the k relational criteria given by (1). The second term employs standard regularizations of the embeddings, such as Frobenius minimization, that enhance the algorithm's numerical stability and support the interpretability of the resulting embeddings. The third term helps capture axiomatic-like patterns by using the similarity matrix C to enrich the learning process with our extra knowledge. We currently support five measures for computing the similarity matrix C: (1) *symmetry*, a measure of the overlap in the entities observed with each relation; (2) *agency*, how often both relations share the same subject; (3) *patient*, how often both relations have the same object; (4) *transitivity*, how often the object of one relation is the subject of the other; and (5) *reverse transitivity*, how often one relation's subject is the other relation's object. Details can be found in Padia et al. (2019).

3 Experiments and Results

We used *area under the curve* to compare the performance of our models with other non-schema based ones. Using constraining embedding for axiomatic-like patterns helps achieve gains of 5% to 50% compared to other tensor and neural models. Table 1 summarizes the results for the eight datasets. We also conducted a detailed study of the effect of (i) how the relation similarity matrix is computed, (ii) the way constraints are applied, and (iii) the effect of sparsity/graph density on the prediction task. See (Padia et al. 2019) for detailed results and model objective functions. We found that our constrained models performed better compared to other models for link prediction. We also showed that tensor-based models performed better than neural network based ranking models, indicating that the objective functions designed to learn the embeddings are important. When the graph density is high, the presence of additional constraints does not provide added advantage and the models perform like other non-schema based methods but constraints are helpful when the graph sparse.

References

Bengio, Y.; Courville, A.; and Vincent, P. 2013. Representation learning:A review and new perspectives. *IEEE TPAMI*.

Krompass, D.; Baier, S.; and Tresp, V. 2015. Type-constrained representation learning in knowledge graphs. In *International Semantic Web Conference*. Springer.

Nickel, M.; Tresp, V.; and Kriegel, H.-P. 2011. A three-way model for collective learning on multi-relational data. In *ICML 2011*, 809–816.

Padia, A.; Kalpakis, K.; Ferraro, F.; and Finin, T. 2019. Knowledge graph fact prediction via knowledge-enriched tensor factorization. *Journal of Web Semantics* v.59. access: final, preprint, ArXiv.org,.

Padia, A. 2019a. *Joint Models to Refine Knowledge Graphs*. Ph.D. Dissertation, University of Maryland, Baltimore County.

Padia, A. 2019b. KGFP GitHub repository.

Socher, R.; Chen, D.; Manning, C. D.; and Ng, A. 2013. Reasoning with neural tensor networks for knowledge base completion. In *NeurIPS*.