Predicting Asymmetric Transitive Relations in Knowledge Bases

Pushpendre Rastogi[1] **Benjamin Van Durme**[1,2]
pushpendre@jhu.edu vandurme@cs.jhu.edu

[1]Center for Language and Speech Processing
[2]Human Language Technology Center of Excellence
Johns Hopkins University

Abstract

Knowledge Base Completion (KBC), or link prediction, is the task of inferring missing edges in an existing knowledge graph. Although a number of methods have been evaluated empirically on select datasets for KBC, much less attention has been paid to understanding the relationship between the logical properties encoded by a given KB and the KBC method being evaluated. In this paper we study the effect of the logical properties of a relation on the performance of a KBC method, and we present a theorem and empirical results that can guide researchers in choosing the KBC algorithm for a KB.

1 Introduction

Large-scale and highly accurate knowledge bases (KB) such as Freebase and YAGO2, have been recognized as essential for high performance on natural language processing tasks such as Relation extraction [Dalton et al., 2014], Question Answering [Yao and Van Durme, 2014], and Entity Recognition in informal domains [Ritter et al., 2011]. Because of this importance of large scale KBs and because the recall of even Freebase, one of the largest open source KB, is low[1] a large number of researchers have presented models for knowledge base completion (KBC).

A popular strategy for KBC is to *embed* the entities and relations in low dimensional continuous vector spaces and to then use the learnt *embeddings* for link prediction. In other words, continuous real valued vectors and matrices are automatically learnt that can represent the entities and edges in a knowledge base, and at the time of inference these real valued representations are used to predict whether a particular edge exists between two entities. This general strategy can be implemented in many different ways and we refer the reader to the survey by [Nickel et al., 2016a] for more details. Even though the strategy of embedding the elements of a graph is popular for knowledge base completion, theoretical studies of such methods are scarce.

In this paper we demonstrate *theoretically*, and *experimentally*, the adverse effect that asymmetric, transitive relations can have on a KBC method that relies on a single vector embedding of a KB entity. Transitive-asymmetric relations such as the `type of` relation in Freebase [Bollacker et al., 2008] and, the *hyponym* relation in WordNet [Miller, 1995] are ubiquitous in KBs and therefore very important [Guha, 2015]. For our theoretical result, we analyze a widely cited KBC algorithm called RESCAL [Nickel et al., 2011, Toutanova et al., 2015] and we prove theoretically that on large KBs that contain a large proportion of asymmetric, transitive relations, methods such as RESCAL will wrongly predict the existence of edges that are the reverse of edges in the training data. We also present a way to mitigate this problem, by using role sensitive embeddings for entities and we empirically verify that our proposed solution improves performance. Through our experiments we also discover a drawback in the prevalent evaluation methodology, of randomly sampling unseen edges, for testing KBC models and show that random sampling can overlook errors on special types of edges.

In: L. Dietz, C. Xiong, E. Meij (eds.): Proceedings of the First Workshop on Knowledge Graphs and Semantics for Text Retrieval and Analysis (KG4IR), Tokyo, Japan, 11-Aug-2017, published at http://ceur-ws.org

[1]It was reported by [Dong et al., 2014] in October 2013, that 71% of people in Freebase had no known place of birth and that 75% had no known nationality.

2 Theoretical Analysis

Notation: A KB contains *(subject, relation, object)* triples. Each triple encodes the fact that a *subject* entity is related to an *object* through a particular *relation*. Let $\mathcal{V}$ and $\mathcal{R}$ denote the set of entities and relationships. We use $\mathcal{V}$ to denote entities to evoke the notion that an entity corresponds to a vertex in the knowledge graph. We assume that $\mathcal{R}$ includes a type for the *null relation* or *no relation*. Let $\mathrm{V} = |\mathcal{V}|$ and $\mathrm{R} = |\mathcal{R}|$ denote the number of entities and relations. We use v and r to denote a generic entity and relation respectively. The shorthand $[n]$ denotes $\{x | 1 \leq x \leq n, x \in \mathbb{N}\}$. Let $\mathcal{E}$ denote the entire collection of facts and let e denote a generic element of $\mathcal{E}$. Each instance of e is an edge in the knowledge graph. We refer to the subject, object and relation of e as $e^{sub}, e^{obj} \in \mathcal{V}$ and $e^{rel} \in \mathcal{R}$ respectively. $\mathrm{E} = |\mathcal{E}|$ is the number of known triples.

RESCAL: The RESCAL model associates each entity v with the vector $a_v \in \mathbb{R}^d$ and it represents the relation r through the matrix $M_r \in \mathbb{R}^{d \times d}$. Let v and v' denote two entities whose relationship is unknown, and let $s(v, r, v') = a_v^T M_r a_{v'}$, then the RESCAL model predicts the relation between v and v' to be: $\hat{r} = \operatorname{argmax}_{r \in \mathcal{R}} s(v, r, v')$. Note that in general if the matrix M_r is asymmetric then the score function s would also be asymmetric, i.e., $s(v, r, v') \neq s(v', r, v)$. Let $\Theta = \{a_v | v \in \mathcal{V}\} \cup \{M_r | r \in \mathcal{R}\}$.

Transitive Relations and RESCAL: In addition to relational information about the binary connections between entities, many KBs contain information about the relations themselves. For example, consider the toy knowledge base depicted in Figure 2a. Based on the information that `Fluffy` *is-a* `Dog` and that a `Dog` *is-a* `Animal` and that *is-a* is a transitive relations we can infer missing relations such as `Fluffy` *is-a* `Animal`.

Let us now analyze what happens when we encode a transitive, asymmetric relation. Consider the situation where the set $\mathcal{R}$ only contains two relations $\{r_0, r_1\}$. r_1 denotes the presence of the *is-a* relation and r_0 denotes the absence of that relation. The embedding based model can only follow the chain of transitive relations and infer missing edges using existing information in the graph if for all triples of vertices v, v', v'' in $\mathcal{V}$ for which we have observed *(v, is-a, v′)* and *(v′, is-a, v′′)* the following holds true:

$$s(v, r_1, v') > s(v, r_0, v') \texttt{ and } s(v', r_1, v'') > s(v', r_0, v'') \implies s(v, r_1, v'') > s(v, r_0, v'')$$

$$\text{I.e. } a_v^T (M_{r_1} - M_{r_0}) a_{v'} > 0 \texttt{ and } a_{v'}^T (M_{r_1} - M_{r_0}) a_{v''} > 0 \implies a_v^T (M_{r_1} - M_{r_0}) a_{v''} > 0 \quad (1)$$

We now define a *transitive matrix* and state a theorem that we prove in Section 6.

Definition A matrix $M \in \mathbb{R}^{d \times d}$ is transitive if $a^T M b > 0$ and $b^T M c > 0$ implies $a^T M c > 0$.

Theorem 1. *Every transitive matrix is symmetric.*

If we enforce the constraint in Equation 1 to hold for all possible vectors and not just a finite number of vectors then $M_{r_1} - M_{r_0}$ is a transitive matrix. By Theorem 1, $M_{r_1} - M_{r_0}$ must be symmetric. This further implies that if $s(v, r_1, v') > s(v, r_0, v')$ then $s(v', r_1, v) > s(v', r_0, v)$. In terms of the toy KB shown in Figure 2a; if the RESCAL model predicts that `Fluffy` *is-a* `Animal` then it will also predict that `Animal` *is-a* `Fluffy`.

Augmenting RESCAL to Encode Transitive Relations: The analysis above points to a simple way for improving RESCAL's performance on asymmetric, transitive relations. The reason that the original method fails to satisfactorily encode transitive asymmetric relations is because if the score $s(v, r_1, v')$ is high then $s(v', r_1, v)$ will also be high. We can avoid this situation by using two different embeddings for all the entities and compute the score of a relation through those role specific embeddings; i.e. we can use the embeddings a_v^1, a_v^2 to represent vertex v and let $s(v, r_1, v') = a_v^1 M_{r_1} a_{v'}^2$ and $s(v', r_1, v) = a_{v'}^1 M_{r_1} a_v^2$. This idea of using role specific embeddings has been known for a long time starting from [Tucker, 1966]. [2] In fact the specific method that we have just explained is generally known to KBC researchers as the Tucker2 decomposition [Singh et al., 2015]. In order to encode more than one relations, only the matrix M_r needs to change but the entity embeddings can be shared across all relations.

3 Related Work

Due to the large body of work that has been done for the task of KBC it is not possible to cover all of the related work on KBC in this section. Instead, we refer the reader to the survey [Nickel et al., 2016a] for an overview of the empirical work that has been done in the area of KBC and link prediction.

Since we focus on the analysis of RESCAL, our work is most closely related to the paper [Nickel et al., 2014]. This paper proves an important theorem that shows that the dimensionality required by the RESCAL model[3] for exactly representing a weighted adjacency matrix of a knowledge graph must be greater than the number of strongly connected components in the graph. In our setting where we consider data sets that contain only transitive-asymmetric relations, the number of

[2] Recently [Yoon et al., 2016] used this idea of using role specific embeddings to preserve the properties of symmetry and transitivity in *translation based* knowledge base embeddings.

[3] Actually their theorem provides a lower bound for a more general model than RESCAL which automatically applies to RESCAL.

strongly connected components in the graph equal the number of vertices in the graph. Therefore their theorem proves that the dimensionality required for exactly representing a dataset such as WordNet using an algorithm such as RESCAL must be greater than the number of entities in the knowledge graph. In contrast to this result, our analysis gives an explicit example of a type of query for which the RESCAL algorithm will make wrong inferences.

Our analysis trivially extends to a few other factorization based algorithms e.g. the Holographic embedding algorithm by [Nickel et al., 2016b]. The holographic embedding method can be rewritten as a constrained form of RESCAL with a "holographically constrained" matrix M. Figure 1 shows an example of a 3×3 holographically constrained matrix with the constraint that elements with the same color must hold the same value. Since such a matrix is asymmetric by construction, our theorem proves that there will exist vectors a, b, and c for which M will violate transitivity.

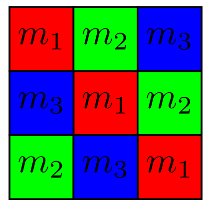


Figure 1: An illustration of a "holographically constrained" matrix.

Recently [Bouchard et al., 2015] argued that the phenomenon of transitivity of relations between vertices in a knowledge graph can be modeled with high accuracy if the knowledge graph is modeled as a thresholded version of a latent low rank real matrix, and the vertex embeddings are learnt as a low rank factorization of that latent matrix. Based on this argument they claimed that factorizing a knowledge graph with a squared loss was less appropriate in comparison to factorizing it with a hinge loss or logistic loss. In this work we provide an argument based on the symmetry of transitive matrices to show that the method of RESCAL which minimizes the squared reconstruction error must fail to capture phenomenon like transitivity in large knowledge bases. In this way, our results complement the work by [Bouchard et al., 2015].

4 Experiments

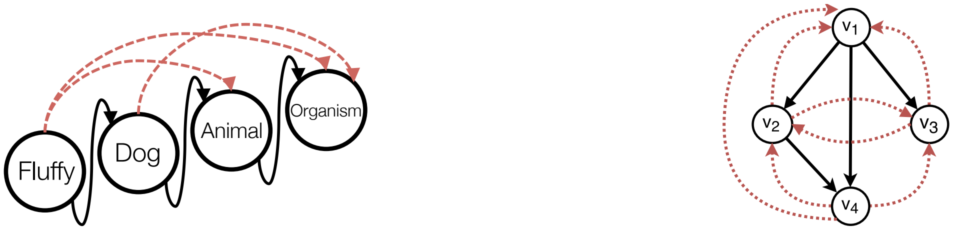
Our theoretical result in Section 2 was derived under the assumption that the constraint 1 held over all vectors in $\mathbb{R}^d$ instead of just the finite number of vector triples used to encode the KB triples. It is intuitive that as the number of entities inside a KB increases our assumption will become an increasingly better approximation of reality. Therefore our theory predicts that the performance of the RESCAL model will degrade as the number of entities inside the KB increases and the dimensionality of the embeddings remains constant. We perform experiments to test this prediction of our theory.

4.1 Experiments On Simulated Data

We tested the applicability of our analysis by the following experiments: We started with a complete, balanced, rooted, directed binary tree $\mathcal{T}$, with edges directed *from* the root *to* its children. We then augmented $\mathcal{T}$ as follows: For every tuple of distinct vertices v, v' we added a new edge to $\mathcal{T}$ if there already existed a directed path starting at v and ending at v' in $\mathcal{T}$. We stopped when we could not add any more edges without creating multi-edges. For the rest of the paper we denote this resulting set of ordered pairs of vertices as $\mathcal{E}$ and those pairs of vertices that are not in $\mathcal{E}$ as $\mathcal{E}^c$. For a tree of depth 11, $\mathrm{V} = 2047, \mathrm{E} = 18,434$ and $|\mathcal{E}^c| = 4,171,775$. See Figure 2b for an example of $\mathcal{E}, \mathcal{E}^c$. We trained the RESCAL model under two settings: In the first setting, called *FullSet*, we used entire $\mathcal{E}$ and $\mathcal{E}^c$ for training. In the second setting, called *SubSet*, we randomly sample $\mathcal{E}^c$ and select only $\mathrm{E} = |\mathcal{E}|$ edges from $\mathcal{E}^c$. All the edges in $\mathcal{E}$ including all the edges in the original tree are always used during both *FullSet* and *SubSet*. For both the settings of *FullSet* and *SubSet* we trained RESCAL 5 times and evaluated the models' predictions on $\mathcal{E}$, $\mathcal{E}^c$ and $\mathcal{E}^{(rev)}$. $\mathcal{E}, \mathcal{E}^c$ have already been defined, and $\mathcal{E}^{(rev)}$ is the set of reversed ordered pairs in $\mathcal{E}$. I.e., $\mathcal{E}^{rev} = \{(u, v) | (v, u) \in \mathcal{E}\}$

For every edge in these three subsets we evaluated the model's performance under $0 - 1$ loss. Specifically, to evaluate the performance of RESCAL on an edge $(v, v') \in \mathcal{E}$ we checked whether the model assigns a higher score to (v, r_1, v') than (v, r_0, v') and rewarded the model by 1 point if it made the right prediction and 0 otherwise. As before, r_1 and r_0 denote the presence and absence of relationship respectively.

Note that low performance on $\mathcal{E}^{rev}$ and high performance on $\mathcal{E}$ will indicate exactly the type of failure predicted from our analysis. We vary the dimensionality of the embedding d, and the number of entities V, since they influence the performance of the model, and present the results in Table 1a–1b. The right most column of Table 1b is the most direct empirical evidence of our theoretical analysis. The performance of RESCAL embeddings is substantially lower on $\mathcal{E}^{rev}$ in comparison to $\mathcal{E}, \mathcal{E}^c$. The last row with $d = 400$ however shows a very sharp drop in the accuracy on $\mathcal{E}^c$ while the performance of $\mathcal{E}^{rev}$ increases



(a) A toy knowledge base containing only *is-a* relations. The dashed edges indicate unobserved relations that can be recovered using the observed edges and the fact that *is-a* is a transitive relation.

(b) Assume that the black edges constitute $\mathcal{E}$ and the red dotted denote $\mathcal{E}^c$, then $\mathcal{E}^{rev}$ contains the edges $(v_4, v_1), (v_4, v_2), (v_2, v_1)$, and (v_3, v_1).

Figure 2: Illustrative Diagrams

slightly. We believe that this happens because of higher overfitting to the forward edges as the number of parameters increases.

FullSet

d	V = 2047			4095			8191		
	$\mathcal{E}$	$\mathcal{E}^c$	$\mathcal{E}^{rev}$	$\mathcal{E}$	$\mathcal{E}^c$	$\mathcal{E}^{rev}$	$\mathcal{E}$	$\mathcal{E}^c$	$\mathcal{E}^{rev}$
50	66	100	100	60	100	100	54	100	100
100	76	100	100	69	100	100	63	100	100
200	86	100	100	79	100	100	72	100	100
400	94	100	100	88	100	100	81	100	100

(a) Accuracy in percentage of RESCAL with all the edges as training data (denoted as *FullSet*) on $\mathcal{E}, \mathcal{E}^c, \mathcal{E}^{rev}$.

SubSet

d	V = 2047			4095			8191		
	$\mathcal{E}$	$\mathcal{E}^c$	$\mathcal{E}^{rev}$	$\mathcal{E}$	$\mathcal{E}^c$	$\mathcal{E}^{rev}$	$\mathcal{E}$	$\mathcal{E}^c$	$\mathcal{E}^{rev}$
50	100	93	52	100	91	48	100	89	44
100	100	78	58	100	92	56	100	89	52
200	100	60	72	100	71	61	100	90	59
400	100	54	67	100	57	70	100	65	62

(b) Accuracy in percentage of RESCAL trained with all positive edges and subsampled negative edges as training data (together called *SubSet*).

Table 1: V denotes the number of nodes in the tree. d denotes the number of dimensions.

4.2 Experiments On WordNet

WordNet is a KB that contains vertices called *synsets* that are arranged in a tree like hierarchy under the *hyponymy* relation. The hyponym of a synset is another synset that contains elements that have a more specific meaning. For example, the *dog* synset[4] is a hyponym of the *animal* synset and an *animal* is a hyponym of *living_thing* therefore a *dog* is a hyponym of *living_thing*. We extracted all the hyponyms of the *living_thing.n.01* synset as the vertices of $\mathcal{T}$ and we used the transitive closure of the direct hyponym relationship between two synsets as the edges of $\mathcal{T}$. Quantitatively, the *living_thing* synset contained $16,255$ hyponyms, and $16,489$ edges. After performing the transitive closure E became $128,241$.

We performed two experiments with the WordNet graphs, using the same *FullSet* and *SubSet* protocols described earlier. The results are in the left half of Table 2. We see that even though the accuracy on $\mathcal{E}$ and $\mathcal{E}^c$ is high, the performance on $\mathcal{E}^{rev}$ is much lower. This trend is in line with our theoretical prediction that the RESCAL model will fail on "reverse relations" as the KB's size increases.

d	*FullSet*			*SubSet*			*SubSet*		
	$\mathcal{E}$	$\mathcal{E}^c$	$\mathcal{E}^{rev}$	$\mathcal{E}$	$\mathcal{E}^c$	$\mathcal{E}^{rev}$	$\mathcal{E}$	$\mathcal{E}^c$	$\mathcal{E}^{rev}$
50	71	100	100	100	93	58	100	55	65
100	79	100	100	100	94	60	100	56	56
200	84	100	100	100	93	63	100	56	75
400	89	100	100	100	68	69	100	97	91

Table 2: Results from experiments on WordNet. We used the subtree rooted at the *living_things* synset from the WordNet hierarchy. d indicates the dimensionality of the embeddings used and the triple of numbers under *FullSet* and *SubSet* indicates the accuracy of RESCAL on $\mathcal{E}, \mathcal{E}^c, \mathcal{E}^{rev}$. V is $16,255$ for all columns. The right half shows results from experiments on WordNet with role dependent embeddings for entities.

Finally we present the results of augmenting RESCAL with role specific embeddings in the right half of Table 2. The results show that using role specific embeddings increases the performance over the performance of the RESCAL algorithm

[4]A synset must be qualified by the word sense and the part of speech. So a valid synset called *dog.n.01*. For simplicity we skip this detail in our explanation but our implementation distinguishes between the synset *dog.n.01* and *dog.n.02*.

and with a high dimensionality of embeddings it is possible to encode both the forward and the reverse relations in the embeddings. Please note that we do not claim that our proposed augmentation for RESCAL will empirically be any better than the much more recently proposed methods such as ARE [Nickel et al., 2014], or Poincaré embeddings [Nickel and Kiela, 2017]. We leave a careful empirical comparison of these techniques for future work.

5 Conclusions

The information present in large scale knowledge bases has helped in moving information retrieval beyond retrieval of documents to more specific entities and objects. And in order to further improve coverage of knowledge bases it is important to research knowledge base completion methods. Since many knowledge bases contain information about real world artifacts that obey hierarchical relations and logical properties, it is important to keep such properties in mind while designing knowledge base completion algorithms. In this paper we demonstrate a close connection between logical properties of relations such as asymmetry, and transitivity, and the performance of KBC algorithms used to predict those relations. Specifically, we theoretically analyzed a popular KBC algorithm named RESCAL, and our analysis showed that the performance of that model in encoding transitive and asymmetric relations must degrade as the size of the KB increases. Our experimental results in Table 1a,1b and 2 confirmed our theoretical hypothesis, and most strikingly we observed that the accuracy of RESCAL on $\mathcal{E}^{rev}$ was substantially lower than its performance on either $\mathcal{E}$ or $\mathcal{E}^c$, even though $\mathcal{E}^{rev}$ is a subset of $\mathcal{E}^c$.

In Table 3, we visualize the errors made by RESCAL by listing a few edges in $\mathcal{E}^{rev}$ that were wrongly predicted as true edges. These examples show that the trained RESCAL model can predict that *fruit tree* is a hyponym of *mango* or that every *accountant* is a *bean counter*. Such wrong predictions can be harmful. Based on our analysis, we advocated for role specific embeddings as a way of alleviating this shortcoming of RESCAL and we empirically showed its efficacy in Table 2.

Our results also highlight a problem with the commonly employed KBC evaluation protocol of randomly dividing the edge set of a graph into train and test sets for measuring knowledge base completion accuracy. For example with $d = 50$ the average accuracy on both $\mathcal{E}$ and $\mathcal{E}^c$ is quite high but on $\mathcal{E}^{rev}$ accuracy is low even though $\mathcal{E}^{rev}$ is a subset of $\mathcal{E}^c$. Such a failure will stay undetected with existing evaluation methods.

Argument 1	*Argument 2*
draftsman.n.02	cartoonist
fruit tree	mango
taster	wine taster
accountant	bean counter
scholar.n.03	rhodes scholar

Table 3: Examples of wrong predictions for the hyponym relations by the RESCAL model with $d = 400$ when trained under the *SubSet* setting. The default synset is *n.01*.

6 Proof of Theorem 1

We now present our novel proof of Theorem 1 beginning with a lemma. [5]

Lemma 2. *Every transitive matrix is PSD.*

Proof. Consider the triplet of vectors $c := x, b := Mc, a := Mb$. Then $a^T(Mb) = ||Mb||^2 \geq 0$ and $b^T(Mc) = ||b||^2 \geq 0$ and $a^T Mc = b^T Mb$. Three cases are possible, either $b = 0$, or $Mb = 0$, or both $Mb \neq 0$ and $b \neq 0$. In the third case transitivity applies and we conclude that $b^T Mb > 0$. In all cases $b^T Mb \geq 0$ which implies M is PSD. □

The next lemma proves that if M is transitive then $x^T My$ and $x^T M^T y$ must have the same sign.

Lemma 3. *If $\exists x, y\ x^T My > 0$ but $x^T M^T y < 0$ then M is not transitive.*

Proof. Let x, y be two vectors that satisfy $x^T My > 0$ and $x^T M^T y < 0$. Since $x^T M^T y = y^T Mx$ therefore $y^T M(-x) > 0$. If we assume M is transitive, then $x^T M(-x) > 0$ by transitivity, but Lemma 2 shows such an x cannot exist. □

Lemma 4 is a general statement about all matrices which states that if the two bilinear forms have the same sign for all inputs then they have to be scalar multiples of each other. We omit its proof due to space constraint.

Lemma 4. *Let $M_1, M_2 \in \mathbb{R}^{d\times d} \setminus \{0\}$. If $x^T M_1 y > 0 \implies x^T M_2 y > 0$ then $M_1 = \lambda M_2$ for some $\lambda > 0$.*

Finally we use Lemma 3–4 to prove Theorem 1.

Proof. Let M be a transitive matrix and let x, y be two vectors such that $x^T My > 0$. By transitivity of M and Lemma 3 $x^T M^T y > 0$. Therefore by Lemma 4 we get $M = \lambda M^T$ for some $\lambda > 0$. Clearly $\lambda = 1$, this concludes the proof that M is symmetric. □

[5] Theorem 1 was first proven by [Grinberg, 2015](unpublished). Our proof is more elementary and direct.

References

[Bollacker et al., 2008] Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. (2008). Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the SIGMOD*, pages 1247–1250. ACM.

[Bouchard et al., 2015] Bouchard, G., Singh, S., and Trouillon, T. (2015). On approximate reasoning capabilities of low-rank vector spaces. In *2015 AAAI Spring Symposium Series*.

[Dalton et al., 2014] Dalton, J., Dietz, L., and Allan, J. (2014). Entity query feature expansion using knowledge base links. In *Proceedings of the 37th SIGIR*, pages 365–374. ACM.

[Dong et al., 2014] Dong, X., Gabrilovich, E., Heitz, G., Horn, W., Lao, N., Murphy, K., Strohmann, T., Sun, S., and Zhang, W. (2014). Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th SIGKDD*, pages 601–610. ACM.

[Grinberg, 2015] Grinberg, D. (2015). Existence and characterization of transitive matrices? MathOverflow. URL:http://mathoverflow.net/q/212808 (version: 2015-08-01).

[Guha, 2015] Guha, R. (2015). Towards a model theory for distributed representations. In *AAAI Spring Symposium Series*.

[Miller, 1995] Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

[Nickel et al., 2014] Nickel, M., Jiang, X., and Tresp, V. (2014). Reducing the rank of relational factorization models by including observable patterns. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, pages 1179–1187. MIT Press.

[Nickel and Kiela, 2017] Nickel, M. and Kiela, D. (2017). Poincaré embeddings for learning hierarchical representations. *arXiv preprint arXiv:1705.08039*.

[Nickel et al., 2016a] Nickel, M., Murphy, K., Tresp, V., and Gabrilovich, E. (2016a). A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33.

[Nickel et al., 2016b] Nickel, M., Rosasco, L., and Poggio, T. (2016b). Holographic embeddings of knowledge graphs. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 1955–1961. AAAI Press.

[Nickel et al., 2011] Nickel, M., Tresp, V., and Kriegel, H.-P. (2011). A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th ICML*, pages 809–816.

[Ritter et al., 2011] Ritter, A., Clark, S., Mausam, and Etzioni, O. (2011). Named entity recognition in tweets: An experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1524–1534, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Singh et al., 2015] Singh, S., Rocktäschel, T., and Riedel, S. (2015). Towards combined matrix and tensor factorization for universal schema relation extraction. In *Workshop on VSM for NLP*, pages 135–142. ACL.

[Toutanova et al., 2015] Toutanova, K., Chen, D., Pantel, P., Poon, H., Choudhury, P., and Gamon, M. (2015). Representing text for joint embedding of text and knowledge bases. In *Proceedings of the EMNLP*, pages 1499–1509. ACL.

[Tucker, 1966] Tucker, L. R. (1966). Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311.

[Yao and Van Durme, 2014] Yao, X. and Van Durme, B. (2014). Information extraction over structured data: Question answering with freebase. In *Proceedings of the 52nd ACL*, pages 956–966, Baltimore, Maryland. ACL.

[Yoon et al., 2016] Yoon, H.-G., Song, H.-J., Park, S.-B., and Park, S.-Y. (2016). A translation-based knowledge graph embedding preserving logical property of relations. In *Proceedings of NAACL-HLT*, pages 907–916.