Towards Combined Matrix and Tensor Factorization for Universal Schema Relation Extraction

Sameer Singh
University of Washington
Seattle, WA
sameer@cs.washington.edu

Tim Rocktäschel* **Sebastian Riedel**
University College London
London, UK
{t.rocktaschel,s.riedel}@cs.ucl.ac.uk

Abstract

Matrix factorization of knowledge bases in universal schema has facilitated accurate distantly-supervised relation extraction. This factorization encodes dependencies between textual patterns and structured relations using low-dimensional vectors defined for each entity pair; although these factors are effective at combining evidence for an entity pair, they are inaccurate on rare pairs, or for relations that depend crucially on the entity types. On the other hand, tensor factorization is able to overcome these shortcomings when applied to link prediction by maintaining entity-wise factors. However these models have been unsuitable for universal schema. In this paper we first present an illustration on synthetic data that explains the unsuitability of tensor factorization to relation extraction with universal schemas. Since the benefits of tensor and matrix factorization are complementary, we then investigate two hybrid methods that combine the benefits of the two paradigms. We show that the combination can be fruitful: we handle ambiguously phrased relations, achieve gains in accuracy on real-world relations, and demonstrate that entity embeddings encode entity types.

1 Introduction

Distantly-supervised relation extraction has gained prominence as it utilizes automatically aligned data to train accurate extractors. Universal schema, in particular, has found impressive accuracy gains by (1) treating the distant-supervision as a knowledge-base (KB) containing both *structured* relations such as `bornIn` and surface form relations such as "was born in" extracted from text, and (2) by *completing* the entries in such a KB using joint and compact encoding of the dependencies between the relations (Riedel et al., 2013; Fan et al., 2014; Chang et al., 2014). Matrix factorization is at the core of this completion: Riedel et al. (2013) convert the KB into a binary matrix with entity-pairs forming the rows and relations forming the columns. Factorization of this matrix results in low-dimensional factors for entity-pairs and relations, which are able to effectively combine multiple evidence for each entity pair to predict unseen relations.

An important shortcoming of this matrix factorization model for universal schema is that no information is shared between the rows that contain the same entity. This can significantly impact accuracy on pairs of entities that are not mentioned together frequently, and for relations that depend crucially on fine-grained entity types, such as `schoolAttended`, `nationality`, and `bookAuthor`. On the other hand, tensor factorization for knowledge-base completion maintains per-entity factors that combine evidence from all the relations an entity participates in, to predict its relations to other entities – a task known as link prediction (Nickel et al., 2012; Bordes et al., 2013). These entity factors, as opposed to pairwise factors in matrix factorization, can be quite effective in identifying the latent, fine-grained entity types. Thus, in the light of the above problems of matrix factorization, the use of tensor factorization for universal schema is tempting. However, directly applying tensor factorization to universal schema has not been successful. Strong results were obtained only through a combination with matrix factorization predictions, and the use of predefined type information (Chang et al., 2014).

*First two authors contributed equally to the paper.

In this paper, we explore the application of matrix and tensor factorization for universal schema data. On simple, synthetic relations, we contrast the representational capabilities of these methods (in § 3.1) and investigate their benefits and shortcomings.We then propose two hybrid tensor and matrix factorization approaches that, by combining their complementary advantages, is able to overcome the shortcomings on synthetic data. We also present improved accuracy on real-world relation extraction data, and demonstrate that the entity embeddings are effective at encoding entity types.

2 Matrix and Tensor Factorization

In this section we introduce universal schemas and various factorization models that can be used to complete knowledge bases of such schemas.

2.1 Universal Schema

A universal schema is defined as the union of all OpenIE-like surface form patterns found in text and fixed canonical relations that exist in a knowledge base (Riedel et al., 2013). The task here is to complete this schema by jointly reasoning over surface form patterns and relations. A successful approach to this joint reasoning is to embed both kinds of relations into the same low-dimensional embedding space, which can be achieved by matrix or tensor factorization methods. We will study such representations for universal schema in this paper.

2.2 Matrix Factorization with Factors over Entity-Pairs

In matrix factorization for universal schema, Riedel et al. (2013) construct a sparse binary matrix of size $|\mathcal{P}| \times |\mathcal{R}|$ whose rows are indexed by entity-pairs $(a, b) \in \mathcal{P}$ and columns by surface form and Freebase relations $s \in \mathcal{R}$. Subsequently, generalized PCA (Collins et al., 2001) is used to find a rank-k factorization, *i.e.*, with relation factors $\mathbf{r} \in \mathbb{R}^{|\mathcal{R}| \times k}$ and entity-pair factors $\mathbf{p} \in \mathbb{R}^{|\mathcal{P}| \times k}$, the probability of a relation s and two entities a and b is:

$$P(s(a, b)) = \sigma(\mathbf{r}_s \cdot \mathbf{p}_{ab}) \quad (1)$$

where σ is the sigmoid function. Using this factorization, similar entity-pairs and relations are embedded close to each other in a k-dimensional vector space. Since this model uses embeddings for pairs of entities, as opposed to per-entity embeddings, we refer to such models as *pairwise* models. Pairwise embeddings are especially suitable when working with universal schema data, since they can represent correlations between surface pattern relations and structured relations compactly. Furthermore, they combine multiple evidences specific to an entity-pair to predict a relation between them. Since the observed data matrix contains only *true* entries, the parameters are learned using Bayesian personalized Ranking (Rendle et al., 2009) that supports implicit feedback.

Riedel et al. (2013) explore a number of variants of this factorization, including a neighborhood model that learns local classifiers, and an entity model that includes entity representations (we revisit this formulation in Section 2.3.4). In the rest of this paper we will only use the basic factorization model (referred to as **Model F**) as the primary pairwise embedding model, however the ideas apply directly to these variants as well.

There are a few shortcomings of models that rely solely on pairwise embeddings. To learn an appropriate representation of an entity-pair, the two entities need to be mentioned together frequently, which is not the case for many entity-pairs of interest. Since predicting relations often relies on the entity types, this lack of ample relational evidence for an entity pair can result in poor estimation of their types, and hence, of their relations. Further, a large number of pairwise relation instances (relative to the number of entities) results in a large number of model parameters, leading to scalability concerns.

2.3 Tensor Factorization with Entity Factors

Instead of using a matrix, it can be natural to represent the binary relations in universal schema as a mode-3 tensor. Here we allocate one mode for relations, one for entities appearing as first argument of relations, and the last mode for entities as second argument. This formulation allows the use of tensor factorization approaches that we will describe here. We use $\mathbf{e}_a \in \mathbb{R}^k$ to refer to the embedding of an entity a. In cases where the position of the entity requires different embeddings, we use $\mathbf{e}_{a,1}$ and $\mathbf{e}_{a,2}$ to represent its occurrence as first and second argument, respectively.

2.3.1 CANDECOMP/PARAFAC-Decomposition

In CANDECOMP/PARAFAC-decomposition (Harshman, 1970) the data tensor is approximated using a finite sum of rank one tensors, *i.e.*,

$$P(s(a,b)) = \sigma\left(\sum_k r_s^{(k)} e_a^{(k)} e_b^{(k)}\right). \quad (2)$$

This decomposition was originally introduced without the logistic function, *i.e.*, in its linear form. However since the additional non-linearity is beneficial for factorizing for binary data (Collins et al., 2001; Bouchard et al., 2015), we use the version above for our relational data.

2.3.2 Tucker2 Decomposition and RESCAL

CP-decomposition is quite restrictive since it does not take advantage of correlations between multiple entities and relations (Nickel et al., 2012). A more expressive factorization is Tucker decomposition (Tucker, 1966), where in its standard formulation, a mode-3 tensor is decomposed into a core tensor and three matrices. However, it is computationally expensive to estimate the core tensor, thus in practice the data tensor is often factorized only along two (instead of three) modes, which is referred to as Tucker2 decomposition. A natural choice for relational data is to keep the relational mode fixed, and thus represent each relation as a $k \times k$ matrix (*e.g.* $\mathbf{R}_s$ for relation s) and entities as k-vectors:

$$P(s(a,b)) = \sigma((\mathbf{R}_s \times \mathbf{e}_{a,1}) \cdot \mathbf{e}_{b,2}). \quad (3)$$

Like PARAFAC, the Tucker2 model was originally introduced in the linear form, however we use the logistic version here. A variant of Tucker2 decomposition that has been applied very successfully in knowledge base completion is RESCAL (Nickel et al., 2012), where each entity in has a single shared embedding irrespective of its argument position. Although a logistic version of RESCAL has also been introduced by Nickel and Tresp (2013), we use the linear form since an open-source implementation of the logistic version is not available.

2.3.3 TransE

Another formulation that is based on entity representations is the translating embeddings model by Bordes et al. (2013). The idea is that if a relation s between two entities a and b holds, that relation's vector representation $\mathbf{r}_s$ should translate the representation $\mathbf{e}_a$ to the second argument $\mathbf{e}_b$, *i.e.*,

$$\text{score}(s(a,b)) = -\|(\mathbf{e}_a + \mathbf{r}_s) - \mathbf{e}_b\|_2. \quad (4)$$

In this work we use a variant of TransE in which different embeddings are learned for an entity for each argument position.

2.3.4 Model E

Furthermore, we isolate the entity factorization in Riedel et al. (2013) by viewing it as tensor factorization. In this model, each relation is assigned an embedding for each of its two arguments, *i.e.*,

$$P(s(a,b)) = \sigma(\mathbf{r}_{s,1} \cdot \mathbf{e}_a + \mathbf{r}_{s,2} \cdot \mathbf{e}_b). \quad (5)$$

Although not explored in isolation by Riedel et al. (2013), model E can be used on its own to predict relations between entities, even if they have not been observed to be in a relation.

3 Combined Tensor and Matrix Factorization for Universal Schema

In the previous section, we provided background on matrix factorization with pairwise factors, followed by a tensor factorization based formulation of universal schema. Although matrix factorization performs well for universal schema (Riedel et al., 2013), it is not robust to sparse data and does not capture latent entity types that can be crucial for accurate relation extraction. On the other hand, although tensor factorization models are able to compactly represent entity types using unary embeddings, they are unable to adequately represent the *pair*-specific information that is necessary for modeling relations. It is worth noting that tensor factorization for universal schema has been proposed by Chang et al. (2014), who also observed that tensor factorization by itself performs poorly (even with additional type constraints), and the predictions need to be combined with matrix factorization to be accurate. In this section we will present the fundamental differences between matrix and tensor factorization, and examine a few hybrid models that can address these concerns.

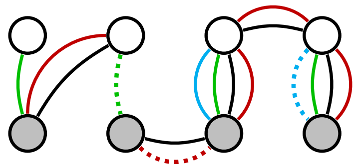


Figure 1: **RGB Relations:** Best viewed in color. **Black** is a sparsely observed relation between *any* pair of entities. **Red** relations correspond to each black edge, and a model that learns this implication can generalize to test instances (red dotted edge). **Green** relation exists between white and gray entities (we omit many of these edges for clarity), requiring the model to learn latent entity *types*. Finally, **Blue** relations exist for pairs where *both* a black and green relation is observed.

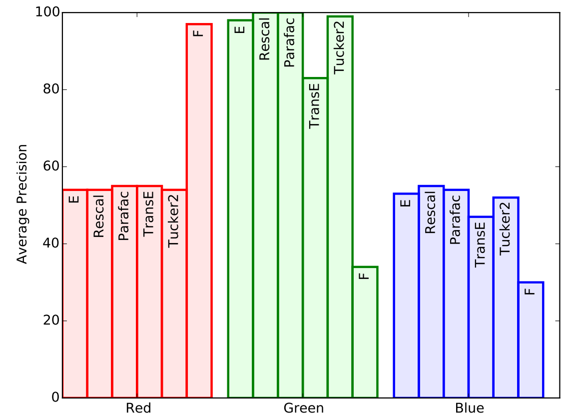


Figure 2: **Matrix versus Tensor Factorization on RGB Data:** illustrating that the tensor factorization approaches (*E*, RESCAL, PARAFAC, *TransE*, and *Tucker2*) are effective only on Green, while matrix factorization (*F*) only on Red. On Blue, both paradigms are unable to generalize.

3.1 Illustration Using Synthetic Relations

As an illustration of the limitations, we present experiments on a simple, synthetic relation extraction task. The generated data consists of entities that belong to one of two types, and the following four types of relations (see Figure 1 for an example): (a) *Black* relations that are observed randomly between any two entities (with probability 0.5), (b) *Red* relations that exist between all pairs for which a *Black* relation exists, similar to a `bornIn` relation corresponding to each observed "X was born in Y" surface pattern, (c) *Green* relations that appear between all pairs of entities of different types, and (d) *Blue* relations that appear between entity pairs that are of different types *and* a Black relation was observed between them. These Blue relation instances represent the relations that often occur in real-data: an ambiguous surface pattern such as "X went to Y" corresponds to `schoolAttended` relation only if the arguments are of certain types. We create such a dataset over 100 entities, and with 5 different sets of such relations (thus 20 total relations, and each entity is assigned 5 of 10 types), and hold out a random 10% of the Red, Green, and Blue relations for evaluation.

These relations target the strengths of the factorization representations. Red relations, as they directly correlate with observed Black instances, should be trivial for matrix factorization.[1] Similarly, Green relations are based on, and clearly define, the latent types of the entities, and thus tensor factorization with entity embeddings should be able to near-perfectly generalize these relations. The converse is more difficult to anticipate; it is unclear how matrix factorization can represent the types needed for Green relations, or whether tensor factorization can encode the Black-Red correspondence. Further, it is not easy to see how any of these approaches will generalize to the Blue relation.

We show the average precision curves on held-out relations for a pairwise embedding approach (matrix factorization *F* from §2.2) and many of the unary embeddings methods from §2.3, with rank 6 in Figure 2. As expected, matrix factorization (*F*) is able to capture the Red relation accurately, however unary embeddings are not able to generalize to it. On the other hand, unary embeddings are able to learn the Green relation which the pairwise approach fail to predict accurately. Blue relations, which most closely model many kinds of relations that occur in text, unfortunately, are not represented well by these approaches that use either unary or pairwise embeddings.

3.2 Hybrid Factorization Models

Since matrix and tensor factorization techniques are quite limited in their representations even on the simple, synthetic data, we now turn to hybrid matrix and

[1] In fact, the set of all Red relations can be represented by rank 2 factors, see Bouchard et al. (2015).

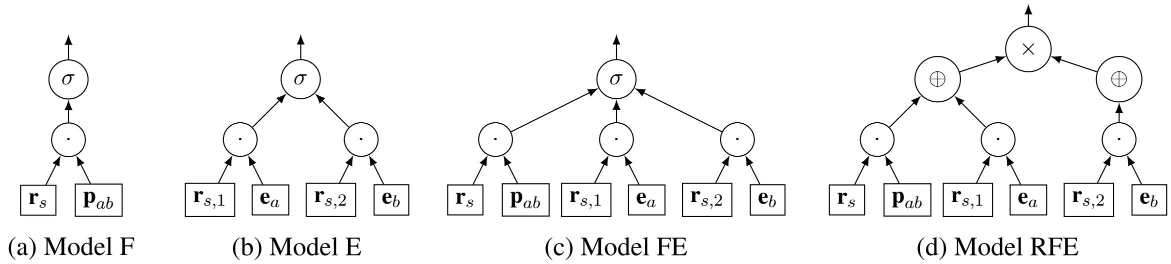


Figure 3: **Overview of the Models:** Some of the models explored in this work, showing pairwise (*F*) and unary (*E*) models, along with their combinations (*FE* and *RFE*), for computing $P(s(a,b))$.

tensor factorization models that represent entity types for universal schema. We describe two possible combinations, models FE and RFE, summarized in Figure 3. Note that these approaches are distinct from collective factorization (Singh and Gordon, 2008) that can be used when extra entity information is available as unary relations.

3.2.1 Combined Model (FE)

As the direct combination of a pairwise model (Eq. 1) with an entity model (Eq. 5), we consider the FE model from Riedel et al. (2013), *i.e.*, the additive combination of the two:

$$P(s(a,b)) = \sigma(\mathbf{r}_s \cdot \mathbf{e}_{ab} + \mathbf{r}_{s,1} \cdot \mathbf{e}_a + \mathbf{r}_{s,2} \cdot \mathbf{e}_b) \quad (6)$$

Both the matrix factorization model *F* and entity model *E* can de defined as special cases of this model, by setting $\mathbf{r}_{s,1/2}$ or $\mathbf{r}_s$ to zero, respectively.

3.2.2 Rectifier Model (RFE)

A problem with combining the two models additively, as in *FE*, is that one model can easily override the other. For instance, even if the type constraints of a relation are violated, a high score by the pairwise model score might still yield a high prediction for that triplet. To alleviate this shortcoming, we experimented with rectifier units (Nair and Hinton, 2010) so that a score of model *F* or model *E* first needs to reach a certain threshold to influence the overall prediction for a triplet. Specifically, we use the smooth approximation of a rectifier $\oplus(x) = \log(1 + e^x)$ and define the probability for a triplet as follows:

$$P(s(a,b)) = \oplus(\mathbf{r}_s \cdot \mathbf{p}_{ab}) \oplus (\mathbf{r}_{s,1} \cdot \mathbf{e}_a + \mathbf{r}_{s,2} \cdot \mathbf{e}_b)$$

3.3 Parameter Estimation

As by Riedel et al. (2013), we use a Bayesian personalized ranking objective (Rendle et al., 2009) to estimate parameters, *i.e.*, for each observed training fact, we sample an unobserved fact for the same relation, and maximize their relative ranking using AdaGrad. For all models we use $k = 100$ as dimension of latent representations, an initial learning rate of 0.1, and ℓ_2-regularization of all parameters with a weight of 0.01. For CANDECOMP/PARAFAC and RESCAL we use the open-source `scikit-tensor`[2] package with default hyper-parameters.

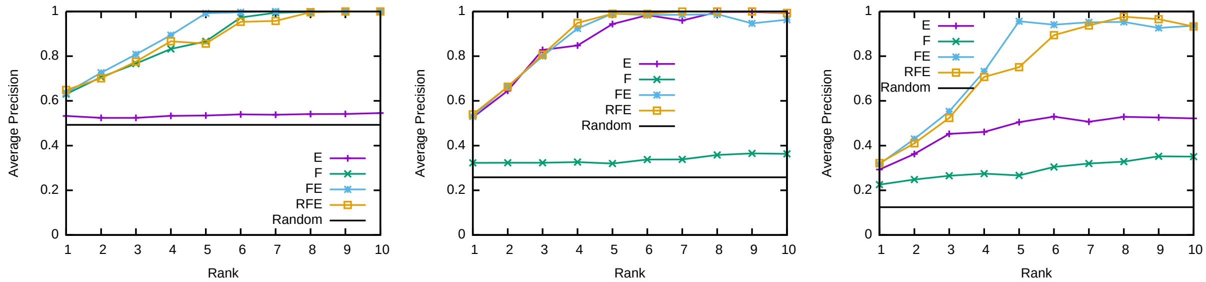
4 Experiments

In order to evaluate whether the hybrid models are able to effectively combine the benefits of matrix and tensor factorization, we first present experiments on synthetic data in Section 4.1. For a more real-world evaluation, we also experiment with universal schema for distantly-supervised relation extraction in Section 4.2.

4.1 Synthetic RGB Relations

In Section 3.1 we described a simple synthetic data set consisting of multiple *Red*, *Green*, and *Blue* relations constructed in order to illustrate the restrictions in the representation capabilities of matrix and tensor factorization models. Here we revisit the dataset using the proposed combined tensor and matrix factorization approaches to evaluate whether these hybrid models are able to compete with tensor and matrix factorization on the relations they are good at (*Green* and *Red*, respectively), but more importantly, whether the combined approaches can represent the *Blue* rela-

[2] `http://github.com/mnick/scikit-tensor`



(a) Red (b) Green (c) Blue

Figure 4: **Hybrid Methods on RGB Data:** Average precision as the rank is varied. *FE* and *RFE* perform as well (or better than) tensor factorization on Green and matrix factorization on Red, but importantly, are able to encode the Blue relations that matrix or tensor factorization fail to model.

	R13-F	TR-R13	E	F	RFE	FE
MAP	60	57	56	59	57	62
Weighted MAP	64	61	51	66	60	66

Table 1: **Distantly-Supervised Relation Extraction:** Weighted and unweighted mean average precision for Freebase relations, as achieved by a number of relation extractors, including pairwise (*F*), unary (*E*), and hybrid (*FE* and *RFE*) models.

tions that matrix and tensor factorization approaches fail to generalize to. In Figure 4 we present the average precision on the held-out data as the rank is varied for a number of approaches (we omit the remaining tensor factorization approaches for clarity since they perform similar to RESCAL and Model E). On the *Red* relation (Figure 4a), tensor factorization is close to random, while combined factorization approaches (FE and RFE) are competitive to, and often outperform, matrix factorization (F). Similarly, on the *Green* relation (Figure 4b), the combined approaches perform as well as tensor factorization, while matrix factorization is not much better than random. Finally, on the *Blue* relation on which matrix and tensor factorization fare poorly, the combined approaches are able to obtain high accuracy, in particular achieve close to 90% average precision with only a rank of 5. Although the same rank corresponds to different numbers of parameters for each method, the trend clearly indicates these results do not depend significantly on the number of parameters.

4.2 Universal Schema Relation Extraction

With the promising results shown on synthetic data, we now turn to evaluation on real-world information extraction. In particular, we evaluate the models on universal schema for distantly-supervised relation extraction. Following the experiment setup of Riedel et al. (2013), we instantiate the universal schema matrix over entity pairs and text/Freebase relations for New York Times data, and compare the performance using average precision of the presented models. Table 1 summarizes the performance of our models, as compared to existing approaches (see Riedel et al. (2013) for an overview). In particular, *TR-R13* takes the output predictions of matrix factorization, and combines it with an entity-type aware RESCAL model (Chang et al., 2014).[3] Tensor factorization approaches perform poorly on this data. We present results for Model *E*, but other formulations such as PARAFAC, TransE, RESCAL, and Tucker2 achieved even lower accuracy; this is consistent with the results in Chang et al. (2014). Models that use the matrix factorization (*F, FE, R13-F* and *RFE*) are significantly better, but more importantly, the hybrid appraoch FE achieves the highest accuracy. It is unclear why *RFE* fails to provide similar gains, in particular, performing slightly worse than matrix factorization. Note that we are not introducing a new state-of-art here, the neighborhood model (*NF*) that achieves a higher accuracy is omitted for clarity.

[3]Here, as in Riedel et al. (2013), we only evaluate on entity pairs that are linked to Freebase, thus the performance of Chang et al. (2014) is lower than their reported results.

LG Electronics
Genentech, Industrial and Commercial Bank of China, Broadway Video, Pollack, Bank Hapoalim, Caremark Rx, Mitchell Gold, Tellabs, Cathay Pacific, Eircom
La Stampa
Toronto Star, O Globo, The Daily Telegraph, El Diario, Le Devoir, Politika, The Straits Times, The Day, RedEye, The Globe
Fatherland
Answered Prayers, Age of Innocence, Auntie Mame, House of Meetings, Bergdorf Blondes, Berlin Diary, Clarissa, Eminent Victorians, Darkness Visible, Gossamer

Table 2: **Nearest-Neighbors** for a few randomly-selected entities based on their embeddings, demonstrating that similar entities are close to each other.

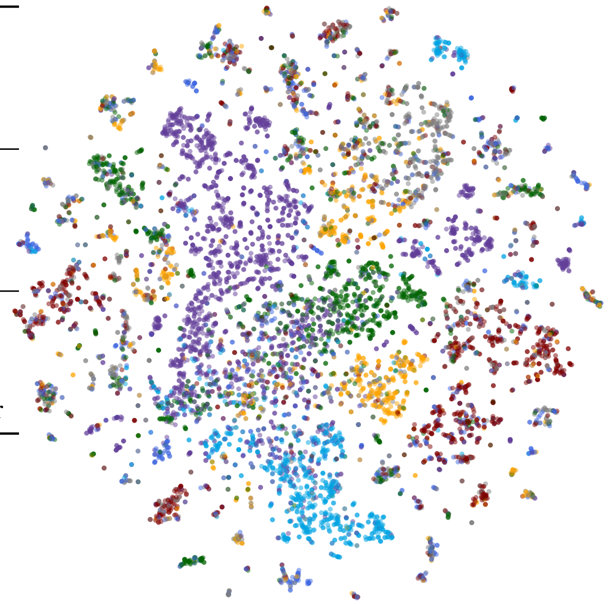


Figure 5: Visualizing entity embeddings, where the colors correspond to their types (**person**, **location**, **organization**, **author**, **actor/musician**, **sports person**, **politician**). Best viewed in color.

4.3 Entity Embeddings and Types

Although the focus of this work is relation extraction, and the models are trained primarily for finding relations, in this section we explore the learned entity embeddings. The low-dimensional entity embeddings have been trained to predict the binary relations that the entity participates in, and thus we expect entities that participate in similar relations to have similar embeddings. To investigate whether the embeddings capture this intuition, we compute similarities of a few randomly selected entities with every other entity using the cosine distance of the *FE* entity embeddings, and show the 10 nearest neighbors in Table 2. The nearest neighbors definitely capture the entity types, for example all the neighbors of "La Stampa" are newspapers in other parts of the world, which is quite impressive considering no explicit type information was available during training. However, the granularity of the types depends on the textual patterns and relations in the schema; for "LG Electronics", the neighbors are mostly generic commercial institutions, perhaps because the observed surface patterns are similar across these types of organizations.

Since the embeddings enable us to compute the similarity between any two entities, we also present a 2D visualization of the entities in the data using the *t-Distributed Stochastic Neighbor Embedding (t-SNE)* (van der Maaten and Hinton, 2008) technique for dimensionality reduction. Further, to investigate whether the embeddings represent correct entity types, we perform an automatic, error-prone alignment of the entity strings to Freebase by finding a prominent entity that has the string as its name, and extract its types. Figure 5 shows the projection for 10 000 randomly selected entities, colored as per their type. We see that the entity embeddings are able to separate most of the coarse level types, as locations are clustered quite separately from the organizations and people, but further, even fine-grained person types occur as distinct collections, for example politicians and sportsmen. There is some cluster overlap as well, especially between the different person types such as authors, actors/musicians, and politicians; it is unclear whether this arises due to incorrect entity linking, inexact two-dimensional projection, entities that belong to multiple types, or from inaccurate embeddings caused by insufficient data.

5 Conclusions and Future Work

Although tensor factorization has been widely used for knowledge-base completion for structured data, it performs poorly on universal schema for relation extraction. Matrix factorization, on the other hand, is appropriate for the task as it is able to compactly represent the correlations between surface pattern and structured KB relations, however learning pairwise factors is not effective for entity pairs with sparse observations or for identifying latent entity types. We illustrate the differences between these matrix and tensor factorization using simple relations, and further, construct an additional relation that none of these approaches are able to model. Motivated by this need for combining their complementary benefits, we explore two hybrid matrix and tensor factorization approaches. Along with being able to model our constructed relations, these approaches also provided improvements on real-world relation extraction. We further provide qualitative exploration of the entity embedding vectors, showing that the embeddings learn fine-grained entity types from relational data.

Our investigations suggest a number of possible avenues for future work. Foremost, we would like to investigate why the hybrid models, which perform significantly better on synthetic data, fail to achieve similar gains on real-world relations. Second, including tensor factorization in the universal schema model enables us to augment the model with external entity information such as observed unary patterns and Freebase types, in order to aid both relation extraction and entity type prediction. Lastly, these hybrid approaches also enable extension of universal schema directly to n-ary relations, allowing a variety of models based on the choice of matrix or tensor representation for each relation.

Acknowledgments

This work was supported in part by Microsoft Research through its PhD Scholarship Programme, an Allen Distinguished Investigator Award, a Marie Curie Career Integration Grant, and in part by the TerraSwarm Research Center, one of six centers supported by the STARnet phase of the Focus Center Research Program (FCRP) a Semiconductor Research Corporation program sponsored by MARCO and DARPA.

References

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Neural Information Processing Systems (NIPS)*.

Guillaume Bouchard, Sameer Singh, and Theo Trouillon. 2015. On approximate reasoning capabilities of low-rank vector spaces. In *AAAI Spring Syposium on Knowledge Representation and Reasoning (KRR): Integrating Symbolic and Neural Approaches*.

Kai-Wei Chang, Wen-tau Yih, Bishan Yang, and Christopher Meek. 2014. Typed tensor decomposition of knowledge bases for relation extraction. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Michael Collins, Sanjoy Dasgupta, and Robert E Schapire. 2001. A generalization of principal components analysis to the exponential family. In *Neural Information Processing Systems (NIPS)*, pages 617–624.

Miao Fan, Deli Zhao, Qiang Zhou, Zhiyuan Liu, Thomas Fang Zheng, and Edward Y. Chang. 2014. Distant supervision for relation extraction with matrix completion. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Richard A. Harshman. 1970. Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis. *UCLA Working Papers in Phonetics*, 16(84).

Vinod Nair and Geoffrey E. Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *International Conference on Machine Learning (ICML)*.

Maximilian Nickel and Volker Tresp. 2013. Logistic tensor factorization for multi-relational data. In *ICML Workshop - Structured Learning: Inferring Graphs from Structured and Unstructured Inputs (SLG)*.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2012. Factorizing yago: scalable machine learning for linked data. In *Proc. of International Conference on World Wide Web (WWW)*, pages 271–280.

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Uncertainty in Artificial Intelligence (UAI)*.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL HLT)*, pages 74–84.

Ajit P. Singh and Geoffrey J. Gordon. 2008. Relational learning via collective matrix factorization. In *Knowledge Discovery and Data Mining (KDD)*.

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

Laurens van der Maaten and Geoffrey E. Hinton. 2008. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605.