Blending of Two and Three-way Interactions for Modeling Multi-relational Data

Alberto García-Durán, Antoine Bordes, Nicolas Usunier

Effective Blending of Two and Three-way Interactions for Modeling Multi-relational Data

Alberto García-Durán, Antoine Bordes, Nicolas Usunier

Université de Technologie de Compiègne - CNRS
Heudiasyc UMR 7253
Compiègne, France
{agarciad,bordesan,nusunier}@utc.fr

Abstract. Much work has been recently proposed to model relational data, especially in the multi-relational case, where different kinds of relationships are used to connect the various data entities. Previous attempts either consist of powerful systems with high capacity to model complex connectivity patterns, which unfortunately usually end up overfitting on rare relationships, or in approaches that trade capacity for simplicity in order to fairly model all relationships, frequent or not. In this paper, we propose a happy medium obtained by complementing a high-capacity model with a simpler one, both pre-trained separately and jointly fine-tuned. We show that our approach outperforms existing models on different types of relationships, and achieves state-of-the-art results on two benchmarks of the literature.

Keywords: Representation learning; Multi-relational data.

1 Introduction

Predicting new links in multi-relational data plays a key role in many areas and hence triggers a growing body of work. Multi-relational data are defined as directed graphs whose nodes correspond to *entities* and *edges* are in the form of triples (*head*, *label*, *tail*) (denoted (h, ℓ, t)), each of which indicates that there exists a relationship of name *label* between the entities *head* and *tail*. Figure 1 displays an example of such data with six entities (*Jane*, *Patti*, *John*, *Mom*, *Miami* and *Austin*) and two relationships (born_in and child_of). Link prediction in this context consists in attempting to create new connections between entities and to determine their type; this is crucial in social networks, knowledge management or recommender systems to name a few.

Performing predictions in multi-relational data is complex because of their heterogeneous nature. Any such data can equivalently be seen as a set of directed graphs that share the same nodes but that usually present drastically different properties in terms of sparsity or connectivity. As illustration, we can look at some statistics of a subset of the knowledge base FREEBASE, named FB15K in the following, which we use for our experiments. This data set contains ∼15k

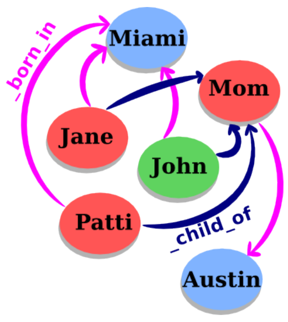

Fig. 1. Example of multi-relational data with 6 entities and 2 relationships.

entities connected by ∼1.5k relationships to form a graph of ∼500k triples. Even if FB15K is just a small sample, it is likely that its characteristics are shared with most real-world multi-relational data, but at a different scale. The relationships of FB15K have a mean number of triples of ∼400 and a median of 21: a vast number of relationships appear in very few triples, while others provide a large majority of the connections. Besides, roughly 25% of the relationships are of type 1-to-1, that is a *head* is connected to at most one *tail* (think of a spouse_of link for instance), but on the opposite 25% of the relationships are of type Many-to-Many, that is, multiple *head* can be linked to a *tail* and vice-versa (for instance, a like_product link). Creating new connections for Many-to-Many relationships can be possible by relying on several existing links of the same kind, whereas in the 1-to-1 case, the only way to be able to generalize is to count on the other relationships, especially if the relationship of interest happens to be rare.

In contrast to (pseudo-) symbolic approaches for link prediction based on Markov-logic networks [11] or random walks [13], most recent effort towards solving this problem concern latent factor models (e.g. [19, 10, 22, 16, 2, 17, 9]) because they tend to scale better and to be more robust w.r.t. the heterogeneity of multi-relational data. These models represent entities with latent factors (usually low-dimensional vectors or *embeddings*) and relationships as operators destined to combine those factors. Operators and latent factors are trained to fit the data using reconstruction [17], clustering [22] or ranking costs [2]. The multi-relational quality of the data is modeled through the sharing of the latent factors across relationships which grants a transfer of information from one relationship to another; operators are normally specific to each relationships, except in [9] where some parameters are shared among relationships.

Learning these latent factor models can be seen as a kind of multitask training, with one task per relationship: one model is fitted for each task and some parameters are shared across tasks. All existing latent factor approaches define the same formulation for each task. This is natural since hand-crafting a particular model for each relationship seems daunting since there can be several thousands of them. However, as all relationships have very different properties, this also induces important drawbacks.

The standard modeling assumption is to consider 3-way interactions between *head*, *label* and *tail*, i.e. to consider that they are all interdependent: the validity of a triple (h, ℓ, t) depends jointly on h, ℓ and t. This generally results in models where entities are represented by vectors and relationships by matrices. An exception is Parafac [8, 6] that models multi-relational data as a binary tensor and factorizes it as a sum of rank one tensors. Other tensor factorization methods derived from Tucker decompositions [23] or Dedicom [**?**] like RESCAL [17, 18] end up with vectorial latent factors for entities and low rank matrices for relationships. This formulation is also shared by many non-parametric Bayesian approaches such as extensions of the Stochastic Block Models [10, 16, 24], or joint entities and relationships clustering approaches [22], which end up modeling the data with similar architectures but with drastically different training and inference procedures. Linear Relational Embeddings [19] were proposed as a 3-way model trained using a regression loss: the vector representing t is learned so that it can be reconstructed using the embedding of h and the matrix encoding ℓ, if (h, ℓ, t) is valid. This work was later followed by the Structured Embeddings model (SE) [2] where the regression loss was replaced by a ranking loss for learning embeddings of entities.

Three-way models are appropriate for multi-relational data since they can potentially represent any kind of interaction. However, this comes at a price since they have to allocate the same large capacity to model each relationship. While this is beneficial for frequent ones, this can be problematic for rare relationships and cause major overfitting. Hence, one must control the capacity either by regularizing, which is not straightforward since different penalties might need to be used for each relationship, or by reducing the expressivity of the model. The second option is implemented in two recent embedding models, SME [3] and TransE [4], that choose to represent multi-relational data as combination of 2-way interactions. The idea is to assume that the validity of a triple (h, ℓ, t) is governed by binary interaction terms (h, t), (t, ℓ) and (h, ℓ), which allows to represent a relationship as a vector as with the other entities. Such a model, TransE [4], outperforms most 3-way approaches on various data sets, which indicates that less expressivity can be beneficial overall for a whole database, especially for relationships where the number of training triples is reduced. However, by design, methods based on 2-way interactions are limited and can not hope to represent all kinds of relations between entities.

In this paper, we introduce Tatec (for *Two And Three-way Embeddings Combination*), a latent factor model which successfully combines well-controlled 2-way interactions with high-capacity 3-way ones. We demonstrate in the following that our proposal is a generalization of many previous methods. Unlike recent work like the Latent Factor Model (LFM) of [9] or the Neural Tensor Model (NTN) of [21] that proposed similar joint formulations mixing several interaction terms, we deliberately choose not to share parameters between the 2- and 3-way interaction components of our model. Previous work use the same embeddings for entities in all terms of their models, which seems to be a sound and natural idea to obtain the possible latent representations. However, we discovered that 2- and 3-way models do not respond to the same data patterns, and that they do not necessarily encode the same kind of information in the embeddings: sharing them among interaction terms can hence be detrimental because it can make some features to be missed by the embeddings or to be destroyed. On the contrary, we explain in the following that using different embeddings for both terms allows to detect distinct kinds of patterns in the data. To ensure that Tatec satisfactorily collects both kinds of patterns, we pre-train separately a 2-way and a 3-way model, which are then combined and jointly fine-tuned in a second stage. We show in various experiments that this combination process is powerful since it allows to jointly enjoy both nice properties of 2 and of 3-way interactions. As a result, Tatec is more robust than previous work w.r.t. the number of training samples and the type of relationships. It consistently outperforms most models in all conditions and achieves state-of-the-art results on two benchmarks from the literature, FB15k [4] and SVO [9].

This paper is organized as follows. Section 2 introduces our formulation and our training procedure, divided in a pre-training phase followed by a fine-tuning step both conducted via stochastic gradient descent. We justify our particular modeling choices in Section 3. Finally, we display and discuss our experimental results on FB15k, SVO and a synthetic dataset in Section 4.

2 Model

We now describe our model, and the training algorithm associated to it. The motivation underlying our parameterization is given in the next section.

2.1 Scoring Function

The data $\mathcal{S}$ is a set of relations between entities in a fixed set of entities in $\mathcal{E} = \{e^1, ..., e^E\}$. Relations are represented as triples (h, ℓ, t) where the head h and the tail t are indexes of entities (i.e. $h, t \in [\![E]\!] = \{1, ..., E\}$), and the label ℓ is the index of a relationship in $\mathcal{L} = \{l^1, ..., l^L\}$, which defines the type of the relation between the entities e^h and e^t. Our goal is to learn a discriminant scoring function on the set of all possible triples $\mathcal{E} \times \mathcal{L} \times \mathcal{E}$ so that the triples which represent likely relations receive higher scores than triples that represent unlikely relations. Our proposed model, Tatec, learns embeddings of entities in low dimensional vector spaces, and parameters of operators on $\mathbb{R}^d \times \mathbb{R}^d$, most of them being associated to single relationships. More precisely, the score given by Tatec to a triple (h, ℓ, t), denoted by $s(h, \ell, t)$, is defined as:

$$s(h, \ell, t) = s_1(h, \ell, t) + s_2(h, \ell, t) \tag{1}$$

where s_1 and s_2 have the following form:

(B) Bigrams or the 2-way interactions terms:

$$s_1(h, \ell, t) = \left\langle \mathbf{r}_1^\ell \middle| \mathbf{e}_1^h \right\rangle + \left\langle \mathbf{r}_2^\ell \middle| \mathbf{e}_1^t \right\rangle + \left\langle \mathbf{e}_1^h \middle| \mathbf{D} \middle| \mathbf{e}_1^t \right\rangle ,$$

where $\mathbf{e}_1^h, \mathbf{e}_1^t$ are embeddings in $\mathbb{R}^{d_1}$ of the head and tail entities of (h, ℓ, t) respectively, $\mathbf{r}_1^\ell$ and $\mathbf{r}_2^\ell$ are vectors in $\mathbb{R}^{d_1}$ that depend on the relationship l^ℓ, and $\mathbf{D}$ is a diagonal matrix that does not depend on the input triple.
As a general notation throughout this section, $\langle . | . \rangle$ is the canonical dot product, and $\langle \mathbf{x} | \mathbf{A} | \mathbf{y} \rangle = \langle \mathbf{x} | \mathbf{A}\mathbf{y} \rangle$ where $\mathbf{x}$ and $\mathbf{y}$ are two vectors in the same space and $\mathbf{A}$ is a square matrix of appropriate dimensions.

(T) Trigram or the 3-way interactions term:

$$s_2(h, \ell, t) = \langle \mathbf{e}_2^h | \mathbf{R}^\ell | \mathbf{e}_2^t \rangle ,$$

where $\mathbf{R}^\ell$ is a matrix of dimensions (d_2, d_2), and $\mathbf{e}_2^h$ and $\mathbf{e}_2^t$ are embeddings in $\mathbb{R}^{d_2}$ of the head and tail entities respectively. The embeddings of the entities for this term are not the same as for the 2-way term; they can have different dimensions for instance.

The embedding dimensions d_1 and d_2 are hyperparameters of our model. All other vectors and matrices are learned without any additional parameter sharing.

The 2-way interactions term of the model is similar to that of [3], but slightly more general because it does not contain any constraint between the relation-dependent vectors $\mathbf{r}_1^\ell$ and $\mathbf{r}_2^\ell$. It can also be seen as a relaxation of the translation model of [4], which is the special case where $\mathbf{r}_1^\ell = -\mathbf{r}_2^\ell$, $\mathbf{D}$ is the identity matrix, and the 2-norm of the entities embeddings are constrained to equal 1.

The 3-way term corresponds exactly to the model used by the collective factorization method RESCAL [17], and we chose it for its high expressivity on complex relationships. Indeed, as we said earlier in the introduction, 3-way models can basically represent any kind of interaction among entities. The usage of combinations of 2-way and 3-way terms has already been used in [9, 21], but, besides a different parameterization, Tatec contrasts with them by the choice of not sharing the embeddings between the two models. In LFM [9], constraints were imposed on the relation-dependent matrix of the 3-way terms (low rank in a limited basis of rank-one matrices), the relation vectors $\mathbf{r}_1^\ell$ and $\mathbf{r}_2^\ell$ were constrained to be a constant linear function of the matrix ($\mathbf{D} = \mathbf{0}$ in their work). These global constraints severely limited the expressivity of the 3-way model, and act as powerful regularization in that respect. However, their global constraints also reduces the expressivity of the 2-way model, which, as we explain in Section 3, should be left with maximum degrees of freedom. The fact that we do not share any parameter between relations is similar to NTN [21]. Our overall scoring function is similar to this model with a single layer, with the fundamental difference that we use different embedding spaces and do not use any non-linear transfer function, which results in a facilitated training (the gradients have a larger magnitude, for instance).

2.2 Training

Training is carried out using gradient descent, with a ranking criterion as training objective. The optimization approach is similar to the one used for TransE [4], but the models are very different. Our loss function takes training triples, and tries to give them higher scores than to corrupted versions, where the corruption consists in either replacing the head or the tail of each triple by a random entity. Since we are learning with positive examples only, this kind of criterion implements the prior knowledge that unobserved triples are likely to be invalid. Such corruption approaches are widely used when learning embeddings of knowledge bases [2, 4] or words in the context of language models [5, 14].

Given a training triple (h, ℓ, t), the set of possible corrupted triples, denoted by $\mathcal{C}(h, \ell, t)$, is defined as:

$$\mathcal{C}(h, \ell, t) = \{(h', \ell, t') \in [\![E]\!] \times \{\ell\} \times [\![E]\!] | h' = h \text{ or } t' = t\} .$$

The loss function we optimize is then:

$$\sum_{(h,\ell,t)\in\mathcal{S}} \sum_{(h',\ell,t')\in\mathcal{C}(h,\ell,t)} \max(0, 1 - s(h, \ell, t) + s(h', \ell, t')) \tag{2}$$

Stochastic gradient is performed in a minibatch setting. The dataset $\mathcal{S}$ is shuffled at each epoch, minibatches of $m << |\mathcal{S}|$ training triples are selected, and, for each one of them, a corresponding mini-batch of corrupted triples is sampled at random to create ranking pairs. We only create a single corrupted triple per training sample. The learning rate of the stochastic gradient is kept constant, and optimization is stopped using early stopping on a validation set.

Several regularization schemes were tried during training: either by forcing the entity embeddings to have, at most, a 2-norm of r (for radius), or by adding 2-norm regularization inside the sum of (2) of the form $\lambda\| \mathbf{x} \|_2^2$ for each parameter $\mathbf{x}$ (relation vectors and diagonal matrix in the 2-way term or relation matrix in the 3-way term) that appears in $\max(0, 1 - s(h, \ell, t) + s(h', \ell, t'))$. The first kind of regularization is carried out after each minibatch by projecting the entities into the 2-norm ball of radius r.

A random initialization of the scoring function (1) can lead to a poor local minimum, but can also prevent the different embeddings used in the 2- and 3-way terms to capture different patterns as we expect (see next section). Hence, following many previous work on deep architecture, we decided to first pre-train separately the bigrams and trigram terms on the training set. When their pre-training is over (i.e. stopped using early stopping on a validation set), we initialize the parameter of the full score (1) using these learned weights and fine-tuned it by running stochastic gradient descent on the training set with the full model.

3 Interpretation and Motivation of the Model

This section discusses the motivations underlying the parameterization of Tatec, and in particular our choice of 2-way model to complement the 3-way term.

3.1 2-Way Interactions as one Fiber Biases

It is common in regression, classification or collaborative filtering to add biases (also called offsets or intercepts) to the model. For instance, a critical step of the best-performing techniques of the Netflix prize was to add user and item biases, i.e. to approximate a user-rating R_{ui} according to (see e.g. [12]):

$$R_{ui} \approx \langle \mathbf{P}_u | \mathbf{Q}_i \rangle + \alpha_u + \beta_i + \mu \tag{3}$$

where $\mathbf{P} \in \mathbb{R}^{U \times k}$, with each row $\mathbf{P}_u$ containing the k-dimensional embedding of the user (U is the number of users), $\mathbf{Q} \in \mathbb{R}^{I \times k}$ containing the embeddings of the I items, $\alpha_u \in \mathbb{R}$ a bias only depending on a user and $\beta_i \in \mathbb{R}$ a bias only depending on an item (μ is a constant that we do not consider further on).

The 2-way + 3-way interactions model we propose can be seen as the 3-mode tensor version of this "biased" version of matrix factorization: the trigram term **(T)** is the collective matrix factorization parametrization of the RESCAL algorithm [17] and plays a role analogous to the term $\langle \mathbf{P}_u | \mathbf{Q}_i \rangle$ of the matrix factorization model for collaborative filtering (3). The bigram term **(B)** then plays the role of biases for each fiber of the tensor,[1] i.e.

$$s_1(h, \ell, t) \approx B^1_{l,h} + B^2_{l,t} + B^3_{h,t} \tag{4}$$

and thus is the analogous for tensors to the term $\alpha_u + \beta_i$ in the matrix factorization model (3). The exact form of $s_1(h, \ell, t)$ given in **(B)** corresponds to a specific form of collective factorization of the fiber-wise bias matrices $\mathbf{B}^1 = \left[B^1_{l,h}\right]_{l \in [\![L]\!], h \in [\![E]\!]}$, $\mathbf{B}^2$ and $\mathbf{B}^3$ of Equation 4. We do not exactly learn one bias by fiber because many such fibers have very little data, while, as we argue in the following, the specific form of collective factorization we propose in **(B)** should allow to share relevant information between different biases.

3.2 The Need for Multiple Embeddings

A key feature of Tatec is to use different embedding spaces for the 2-way and 3-way terms, while existing approaches that have both types of interactions use the same embedding space [9, 21]. We motivate this choice in this section.

It is important to notice that biases in the matrix factorization model (3), or the bigram term in the overall scoring function (1) do not affect the model expressiveness, and in particular do not affect the main modeling assumptions that embeddings should have low rank. The user/item-biases in (3) only boil down to adding two rank-1 matrices $\boldsymbol{\alpha}\mathbf{1}^T$ and $\mathbf{1}\boldsymbol{\beta}^T$ to the factorization model. Since the rank of the matrix is a hyperparameter, one may simply add 2 to this hyperparameter and get a slightly larger expressiveness than before, with reasonably little impact since the increase in rank would remain small w.r.t. its original value (which is usually 50 or 100 for large collaborative filtering data sets). The critical feature of these biases in collaborative filtering is how they interfere with capacity control terms other than the rank, namely the 2-norm regularization: in [12] for instance, all terms of (3) are trained using a squared error as a measure

[1] Fibers are the higher order analogue of matrix rows and columns for tensors and are defnied by fixing every index but one.

of approximation and regularized by $\lambda \left(\| \mathbf{P}_u \|_2^2 + \| \mathbf{Q}_i \|_2^2 + \alpha_u^2 + \beta_i^2 \right)$, where $\lambda > 0$ is the regularization factor. This kind of regularization is a weighted trace norm regularization [20] on $\mathbf{PQ}^T$. Leaving aside the "weighted" part, the idea is that at convergence, the quantity $\lambda \left(\sum_u \| \mathbf{P}_u \|_2^2 + \sum_i \| \mathbf{Q}_i \|_2^2 \right)$ is equal to 2λ times the sum of the singular values of the matrix $\mathbf{PQ}^T$. However, $\lambda \| \boldsymbol{\alpha} \|_2^2$, which is the regularization applied to user biases, is *not* 2λ times the singular value of the rank-one matrix $\boldsymbol{\alpha}\mathbf{1}^T$, which is equal to $\sqrt{I} \| \boldsymbol{\alpha} \|_2$, and can be much larger than $\| \boldsymbol{\alpha} \|_2^2$. Thus, if the pattern user+item biases exists in the data, but very weakly because it is hidden by stronger factors, it will be less regularized than others and the model should be able to capture it. Biases, which are allowed to fit the data more than other factors, offer the opportunity of relaxing the control of capacity on some parts of the model but this translates into gains if the patterns that they capture are indeed useful patterns for generalization. Otherwise, this ends up relaxing the capacity to lead to more overfitting.

Our bigram terms are closely related to the trigram term: the terms $\langle \mathbf{r}_1^\ell | \mathbf{e}_1^h \rangle$ and $\langle \mathbf{r}_2^\ell | \mathbf{e}_1^t \rangle$ can be added to the trigram term by adding constant features in the entities' embeddings, and $\langle \mathbf{e}_1^h | \mathbf{D} | \mathbf{e}_1^t \rangle$ is directly in an appropriate quadratic form. Thus, the only way to gain from the addition of bigram terms is to ensure that they can capture useful patterns, but also that capacity control on these terms is less strict than on the trigram terms. In tensor factorization models, and especially 3-way interaction models with parameterizations such as **(T)**, capacity control through the regularization of individual parameters is still not well understood, and as it turns out in experiments is more detrimental than effective. The only effective parameter is the admissible rank of the embeddings, which leads to the conclusion that the bigram term can be really useful in addition to the trigram term if higher-dimensional embeddings are used. Hence, in absence of clear and concrete way of effectively controlling the capacity of the trigram term, we believe that different embedding spaces should be used.

3.3 2-Way Interactions as Entity Types+Similarity

Having a part of the model that is less expressive, but less regularized than the other part is only useful if the patterns it can learn are meaningful for the prediction task at hand. In this section, we give the motivation for our 2-way interactions term for the task of modeling multi-relational data.

Most relationships in multi-relational data, and in knowledge bases like FB15K in particular, are strongly typed, in the sense that only well-defined and specific subsets of entities can be either heads or tails of selected relationships. For instance, a relationship like `capital_of` expects a (big) city as head and a country as tail for any valid relation. Large knowledge bases have huge amounts of entities, but those belong to many different types. Identifying the expected types of head and tail entities of relationships, with an appropriate granularity of types (e.g. `person` or `artist` or `writer`), is likely to filter out 95% of the entity set during prediction. The exact form of the first two terms $\langle \mathbf{r}_1^\ell | \mathbf{e}_1^h \rangle + \langle \mathbf{r}_2^\ell | \mathbf{e}_1^t \rangle$ of the

Table 1. Statistics of the data sets used in this paper and extracted from an artificial database, FAMILY, and from two knowledge bases: FB15K and SVO.

DATA SET	FAMILY	FB15K [4]	SVO [9]
ENTITIES	721	14,951	30,605
RELATIONSHIPS	5	1,345	4,547
TRAINING EXAMPLES	5,748	483,142	1,000,000
VALIDATION EXAMPLES	1,935	50,000	50,000
TEST EXAMPLES	1,955	59,071	250,000

2-way interaction model **(B)**, which corresponds to a low-rank factorization of the per bias matrices (*head*, *label*) and (*tail*, *label*) in which *head* and *tail* entities have the same embeddings, is based on the assumption that the types of entities can be predicted based on few (learned) features, and these features are the same for predicting *head*-types as for predicting *tail*-types. As such, it is natural to share the entities embeddings in the first two terms of **(B)**.

The last term, $\langle \mathbf{e}_1^h | \mathbf{D} | \mathbf{e}_1^t \rangle$, is intended to account for a global similarity between entities. For instance, predicting the capital of France can easily be performed correctly by saying that we search for the city with strongest overall connections with France in the knowledge base. A country and a city may be strongly linked through their geographical positions, independent of their respective types. The diagonal matrix $\mathbf{D}$ allows to re-weight features of the embedding space to account for the fact that the features used to describe types may not be the same as those that can describe the similarity between objects of different types. The use of a diagonal matrix is strictly equivalent to using a general symmetric matrix in place of $\mathbf{D}$.[2] The reason for using a symmetric matrix comes from the intuition that the direction of many relationships is arbitrary (i.e. the choice between having triples "Paris is capital of France" rather than "France has capital Paris"), and the model should be invariant under arbitrary inversions of the directions of the relationships (in the case of an inversion of direction, the relations vectors $\mathbf{r}_1^\ell$ and $\mathbf{r}_2^\ell$ are swapped, but all other parameters are unaffected). For tasks in which such invariance is not desirable, the diagonal matrix could be replaced by an arbitrary matrix.

4 Experiments

This section presents a series of experiments that we conducted to compare Tatec to previous models from the literature on two benchmarks, FB15K, a subset of FREEBASE [4], and SVO, a database of nouns connected by verbs and introduced in [9], as well an artificial data set that we created (FAMILY). The statistics of these data sets are given in Table 1.

[2] We can see the equivalence by taking the eigenvalue decomposition of a symmetric $\mathbf{D}$: apply the change of basis to the embeddings to keep only the diagonal part of$\mathbf{D}$ in the term $\langle \mathbf{e}_1^h | \mathbf{D} | \mathbf{e}_1^t \rangle$, and apply the reverse transformation to the vectors $\mathbf{r}_1^\ell$ and $\mathbf{r}_2^\ell$. Note that since rotations preserve euclidian distances, the equivalence still holds under 2-norm regularization of the embeddings.

Table 2. Baselines. Rules used by our symbolic baselines, an upper and a lower one, to predict a *tail* given a *head* and a *label* on the Family data set. Similar symmetric rules have been used to predict a *head* given a *tail* and a *label* for these relations.

Relationship	Baseline	Rules for predicting a *tail* given a *head*
cousin_of	Lower	Any entity of the same layer as the *head* of the families where the *head* has a cousin.
	Upper	Any entity whose parent is a sibling(-in-law) of the parents of the *head*.
sibling_of	Lower	Any entitiy of the same layer as the *head* of the families where the *head* has a sibling.
	Upper	Any entity whose parent is the same as the parents of *head*.
married_to	Lower	Any entity of the same layer as the *head* whose marriage would not be forbidden with.
	Upper	Entity who has children in common with the *head*.
parent_of	Lower	Any entity of the lower layer of *head* of the families where *head* has a child.
	Upper	Any entity being children of the spouse of *head*.
uncle_of	Lower	Any entity of the lower layer of *head* of the families where *head* has a niece/nephew.
	Upper	Any entity being child of a sibling/sibling a law of *head* or the spouse of *head*.

For evaluation, we use a ranking procedure as in [2, 4]. For each test triplet, the head is removed and replaced by each of the entities of the dictionary in turn. Scores of those corrupted triplets are computed by the models and sorted by descending order and the rank of the correct entity is stored. This whole procedure is repeated when removing the tail instead or the head. We report the *mean* of those predicted ranks and the *hits@10*, i.e. the proportion of correct entities ranked in the top 10.

4.1 Synthetic Data

"Family" Data Set This database contains triples expressing family relationships among the members of 5 families along 6 generations, each family being organized in a layered tree structure where each layer refers to a generation. These 5 families are first created independent of each other by recursively sampling the number of children of each node of a layer using the normal distribution $\mathcal{N}(3, 1.5)$ to create a new generation. Then, families are connected by marriage links between two members. We use pre-defined rules to avoid non-typical situations, like marriages between cousins and brothers, as well as marriages between two members of different generations. To control the number of connections between families, only $i - 1$ marriages are allowed in the generation i.

After all families are created, we build a multi-relational data set by collecting the pairs of entities connected using the following relationships: cousin_of, married_to, parent_of, sibling_of and uncle_of. We end up with a data set with 721 entities, 5 relationships and ∼9k triples which is later split into training, validation and test sets. There is a large variation in the numbers of triples: there are only 30 examples with married_to but 5, 060 with cousin_of. Family is a realistic and challenging study case, but for which we know the underlying semantics.

Baselines We created two symbolic baselines since we know the underlying rules used to generate the data. These baselines can be used to assess what kind of pattern is caught by our model. Our first baseline, Lower, uses simplistic rules in order to find a set of potential candidates among all entities given a *label* and either a *head* or a *tail* as input. Our second baseline, Upper, returns candidate

Table 3. Synthetic data set. We compare our Bigrams, Trigram and Tatec models in terms of mean rank, with both baselines on the FAMILY dataset.

RELATION	cousin_of	married_to	parents_of	sibling_of	uncle_of
Lower	76	4	25	35	34
Upper	**4**	**3**	7	**4**	**3**
Bigrams	10	102	9	8	13
Trigram	7	162	7	5	8
Tatec	6	69	**5**	**4**	6

answers using a much more refined knowledge about the underlying semantics of the relationships, such as, if *John* and *Mary* have children together then they are likely to be married. We made sure than, for any triple from the data set, the correct missing element given a *label* and either a *head* or a *tail* as input, is always contained in the sets returned by Lower and Upper (the second being included in the first one). Then, for each test triple, we computed the mean rank of the answers given by the baselines, by sorting the elements of the returned candidate sets by the number of occurrences of each entity in the training set. It is worth noting that the entities of the Upper set always form triples expressing true knowledge. The rules used to defined both baselines are given in Table 2.

Implementation To pre-train our Bigrams and Trigram models we selected the learning rate for the stochastic gradient descent among $\{0.1, 0.01, 0.001, 0.0001\}$, and the embedding dimension among $\{5, 10, 20\}$. The margin was fixed to 1, and the radius determining the maximim 2-norm of the entity embeddings was validated among $\{1, 10, 100\}$. For fine-tuning Tatec, the learning rate was selected among the same values as above, independent of the values chosen for pre-training. For all three models, training was limited to a maximum of 500 epochs, and we used the mean rank on the validation set (computed every 50 epochs) as stopping criterion.

Results Table 3 presents our results. Tatec outperforms the best performance of Bigrams and Trigram counterparts for each relationship, indicating that the biases brought to the 3-way model by the bigram terms are indeed beneficial to detect complementary patterns in the data. As a result, Tatec gets a really close performance to that of the upper baseline, indicating that it can perform relatively sophisticated inference, as long as the amount of data is sufficient. Indeed, all embedding-based models fail completely on the married_to relationship: with very few training samples, those models cannot learn any meaningful information that would grant non-trivial predictions for this complex relationship.

4.2 Encoding Freebase

Freebase Data Set (FB15k) FREEBASE is a huge and growing database of general facts; there are currently around 1.2 billion triples and more than 80 million entities. We used FB15K, a data set based on FREEBASE introduced in

Table 4. Link prediction results. We compare Bigrams, Trigram and several version of Tatec with various methods from the literature on the FB15K dataset. Results are displayed in the filtered setting, see the text for more details.

INTERACTION	MODEL	MEAN RANK	HITS@10
2-way	SME(LINEAR) [3]	154	40.8
	TransE [4]	125	47.1
3-way	SE [2]	162	39.8
	SME(BILINEAR) [3]	158	41.3
	RESCAL [17]	683	44.1
2 + 3-way	NTN [21]	332	27.0
	LFM [9]	164	33.1
2-way	Bigram	133	44.7
3-way	Trigram	156	42.7
2 + 3-way	Tatec-NO-PRETRAIN	133	44.7
	Tatec-SHARED-EMBS	137	45.0
	Tatec-LINEAR-COMB	115	51.7
	Tatec	**111**	**52.6**

[4], This small data set is based on a subset of entities that are also present in the WIKILINKS database[3] and that also have at least 100 mentions in FREEBASE (for both entities and relationships). This results in $592,213$ triples with $14,951$ entities and $1,345$ relationship which were randomized and split.

Baselines We compare Tatec with various models from previous work: RESCAL [17], LFM [9] , SE [2] , SME [3] and TransE [4]. Results were extracted from [4] since we follow the same experimental protocol here. We also include comparisons with NTN [21]. For this method, we ran experiments with the code provided by the authors. The embedding dimension was selected between $\{25, 50\}$ and the number of slices of the tensor layer was fixed to 2 for computational considerations. We chose the regularization hyperparameter among $\{0, 0.1, 0.01, 0.001\}$ and tanh as nonlinear element-wise function. The negative triplets were generated as before in a proportion of 10 negative to 1 positive triple. The model ran for 700 iterations and was validated every 50 iterations.

Besides Bigrams, Trigram and Tatec, we also propose the performance of 3 other versions of Tatec:

– Tatec-NO-PRETRAIN: Tatec without pre-training $s_1(h, \ell, t)$ and $s_2(h, \ell, t)$.
– Tatec-SHARED: Tatec but sharing the embeddings between $s_1(h, \ell, t)$ and $s_2(h, \ell, t)$ and without pre-training.
– Tatec-LINEAR-COMB: this version simply combines the bigram and trigram terms using a linear combination, without jointly fine-tuning their parameters. The score is hence defined as follows:

$$s(h, \ell, t) = \delta_1^\ell \langle \mathbf{r}_1^\ell | \mathbf{e}_1^h \rangle + \delta_2^\ell \langle \mathbf{r}_2^\ell | \mathbf{e}_1^t \rangle + \delta_3^\ell \langle \mathbf{e}_1^h | \mathbf{D} | \mathbf{e}_1^t \rangle + \delta_4^\ell \langle \mathbf{e}_2^h | \mathbf{R}^\ell | \mathbf{e}_2^t \rangle$$

[3] `code.google.com/p/wiki-links`

Table 5. Detailed results by category of relationship. We compare our Bigrams, Trigram and Tatec models in terms of Hits@10 (in %) on FB15K in the filtered setting against other models of the literature. (M. stands for MANY).

TASK	PREDICTING *head*				PREDICTING *tail*			
REL. CATEGORY	1-TO-1	1-TO-M.	M.-TO-1	M.-TO-M.	1-TO-1	1-TO-M.	M.-TO-1	M.-TO-M.
SE [2]	35.6	62.6	17.2	37.5	34.9	14.6	68.3	41.3
SME(LINEAR) [3]	35.1	53.7	19.0	40.3	32.7	14.9	61.6	43.3
SME(BILINEAR) [3]	30.9	69.6	19.9	38.6	28.2	13.1	76.0	41.8
TransE [4]	43.7	65.7	18.2	47.2	43.7	**19.7**	66.7	**50.0**
Bigrams	55.4	73.2	25.5	49.3	51.3	11.4	78.5	37.4
Trigram	44.3	69.6	29.0	48.0	41.2	8.3	72.6	35.1
Tatec	**65.8**	**84.8**	**40.0**	**58.9**	**62.3**	15.1	**87.0**	42.3

The combination weights δ_i^ℓ depend on the relationship and are learned by optimizing the ranking loss defined in (2) using L-BFGS, with an additional quadratic penalization term, $\sum_k \frac{||\delta^k||_2^2}{\sigma_k+\epsilon}$, subject to $\sum_k \sigma_k = \lambda$. Training is carried out in an iterative way, by alternating optimization of δ parameters via L-BFGS, and update of σ parameters using $\sigma_i^* = \frac{\lambda||\delta_i||_2}{\sum_i ||\delta_i||_2}$, until some stopping criterion is reached. The δ parameters are initialized to 1 and the λ value is validated among $\{0.1, 1, 10, 100, 250, 500, 1000\}$. The intuition behind this particular penalization for the δs is that it is equivalent to a LASSO penalization [7] and our initial idea was to enforce sparsity among δ parameters. However we found experimentally that the best performance was obtained with a λ of 250, which does not yield a sparse solution.

Implementation Tatec, and its 3 alternative versions have been trained and validated in the same setting that was used for the FAMILY experiments, except that we chose the embedding dimensions among $\{25, 50\}$.

Results Table 4 displays the mean rank and hits@10 for all the aforementioned methods. These results have been computed in a *filtered setting* as defined in [4]: to reduce the error introduced by true triples that might be ranked above the target triple in test, all the entities forming existing triples in the train, validation and test sets but the target one are removed from the candidate set of entities to be ranked. This grants a clearer view on ranking performance.

First of all, we can notice that our plain 2- and 3$-$ way models (Bigrams and Trigram respectively) are performing comparably as other similarly expressive models: Bigrams is better than SME(linear) but worse than TransE, and Trigram performs roughly like SME(bilinear). RESCAL is interesting since it achieves a very poor mean rank but almost the best hits@10 value: we believe that this is due to overfitting. To make it scale on large data sets, RESCAL has to be ran without regularization, this causes a major overfitting on rare relationships and hence a poor mean rank. But, it seems that one can reach a very decent hits@10 nonetheless. Interestingly, Tatec is able to significantly outperform both its con-

Table 6. Verb prediction results. We compare our Bigrams, Trigram and Tatec models with baselines from the literature on the SVO dataset.

	Median/Mean Rank	Hits@5%	Hits@20%
Counts [9]	48/517	72	83
LFM [9]	50/195	78	95
SME [3]	56/199	77	95
Bigrams	52/210	78	98
Trigram	44/188	79	95
Tatec	**42/180**	**80**	**99**

stituents Bigrams and Trigram, which indicates that they can encode complementary information. This is confirmed by the comparison with the baseline 2+3-way models, LFM and NTN.[4] By sharing their embeddings between their 2- and 3-way terms, they constrain their model too much. We can see a similar behavior if we look at the results of Tatec-SHARED-EMBS, which are much worse than those of Tatec. The pre-training is very useful: without pre-training, Tatec only achieves the same performance as the 2-way term alone. Tatec-LINEAR-COMB performs only slightly worse than Tatec, which indicates that, with proper regularization, a simpler combination of 2- and 3-way terms can be efficient. Overall, Tatec outperforms all previous models by a wide margin, especially in hits@10.

We also broke down the results by type of relation, classifying each relationship according to the cardinality of their *head* and *tail* arguments. A relationship is considered as 1-to-1, 1-to-M, M-to-1 or M-M regarding the variety of arguments *head* given a *tail* and vice versa. If the average number of different *heads* for the whole set of unique pairs (*label*, *tail*) given a relationship is below 1.5 we have considered it as 1, and the same in the other way around. The number of relations classified as 1-to-1, 1-to-M, M-to-1 and M-M is 353, 305, 380 and 307, respectively. The results are displayed in the Table 5. Most results point out that Tatec consistently outperforms all models we compared it with, except for the relations 1-to-M and M-to-M when predicting the *tail*.

4.3 Predicting Verbs

In this last experimental section, we present results of ranking *label* given *head* and *tail*. We do so by working on a verb prediction task, where one has to assign the correct verb given two noun phrases acting subject and direct object.

Subject-Verb-Object Data Set (SVO) This data set was generated by extracting sentences from Wikipedia articles whose syntactic structure is (subject, verb, direct object) and where the verb appears in the WordNet lexicon [15] and where the subject and direct object are noun phrases from WordNet as well. Due to the high number of relations in this data set, this is an interesting benchmark for *label* prediction.

[4] Results of NTN are worse than expected. As we said earlier, we tried a large number of hyperparameter values but NTN might require to cover an even wider range.

Baselines We compare Tatec with 3 different approaches: LFM, Counts and SME(linear). Counts is based on the direct estimation of probabilities of triples (*head*, *label*, *tail*) by using the number of occurrences of pairs (*head*, *label*) and (*label*, *tail*) in the training set. The results for these models have been extracted from [9], and we followed the same experimental setting.

Implementation Due to the different nature of the application, the negative triples have been generated here by replacing the verb of a given positive triple by a random verb. The rest of the experimental setting is identical to the one used for Family and FB15k, but running only 100 epochs and validating every 10 epochs in the pre-training phase, since we found that the models were converging much faster. For Tatec, we even validated every epoch.

Results Table 6 shows the results for this database. The measure hits@z% indicates the proportion of predictions for which the correct verb is ranked in the top $z\%$ of the verb list. The performance of Tatec is also excellent in this case since it outperforms all previous methods on all metrics, including LFM, another model combining 2- and 3-way interactions.

5 Conclusion

This paper introduced Tatec, a new method for performing link prediction in multi-relational data, which is made of the combination a 2- and 3-way interactions terms. Both terms do not share their embeddings and this, along with a two-phase training (pre-training and fine-tuning), allows for the model to encode complementary information into its parameters. As a result, Tatec outperforms by a wide margin many methods from the literature, some based on 2-way, on 3-way interactions or on both.

Acknowledgments

This work was carried out in the framework of the Labex MS2T (ANR-11-IDEX-0004-02), and was funded by the French National Agency for Research (EVEREST-12-JS02-005-01).

References

1. Bader, B.W., Harshman, R.A., Kolda, T.G.: Temporal analysis of social networks using three-way dedicom. Sandia National Laboratories TR SAND2006-2161 119 (2006)
2. Bordes, A., Weston, J., Collobert, R., Bengio, Y.: Learning structured embeddings of knowledge bases. In: Proc. of the 25th Conf. on Artif. Intel. (AAAI) (2011)
3. Bordes, A., Glorot, X., Weston, J., Bengio, Y.: A semantic matching energy function for learning with multi-relational data. Machine Learning (2013)
4. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Advances in Neural Information Processing Systems. pp. 2787–2795 (2013)

5. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. Journal of Machine Learning Research 12, 2493–2537 (2011)
6. Franz, T., Schultz, A., Sizov, S., Staab, S.: Triplerank: Ranking semantic web data by tensor decomposition. In: Proceedings of the 8th International Semantic Web Conference. pp. 213–228. ISWC '09 (2009)
7. Grandvalet, Y.: Least absolute shrinkage is equivalent to quadratic penalization. In: The International Conference on Artificial Neural Networks (ICANN). pp. 201–206 (1998)
8. Harshman, R.A., Lundy, M.E.: Parafac: parallel factor analysis. Comput. Stat. Data Anal. 18(1), 39–72 (Aug 1994)
9. Jenatton, R., Le Roux, N., Bordes, A., Obozinski, G., et al.: A latent factor model for highly multi-relational data. In: NIPS 25 (2012)
10. Kemp, C., Tenenbaum, J.B., Griffiths, T.L., Yamada, T., Ueda, N.: Learning systems of concepts with an infinite relational model. In: Proc. of the 21st national conf. on Artif. Intel. (AAAI). pp. 381–388 (2006)
11. Kok, S., Domingos, P.: Statistical predicate invention. In: Proceedings of the 24th international conference on Machine learning. pp. 433–440. ICML '07 (2007)
12. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer 42(8), 30–37 (2009)
13. Lao, N., Mitchell, T., Cohen, W.W.: Random walk inference and learning in a large scale knowledge base. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. pp. 529–539. Association for Computational Linguistics (2011)
14. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems. pp. 3111–3119 (2013)
15. Miller, G.: WordNet: a Lexical Database for English. Communications of the ACM 38(11), 39–41 (1995)
16. Miller, K., Griffiths, T., Jordan, M.: Nonparametric latent feature models for link prediction. In: Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C.K.I., Culotta, A. (eds.) Advances in Neural Information Processing Systems 22, pp. 1276–1284 (2009)
17. Nickel, M., Tresp, V., Kriegel, H.P.: A three-way model for collective learning on multi-relational data. In: Proceedings of the 28th International Conference on Machine Learning (ICML-11). pp. 809–816 (2011)
18. Nickel, M., Tresp, V., Kriegel, H.P.: Factorizing yago: scalable machine learning for linked data. In: Proceedings of the 21st international conference on World Wide Web. pp. 271–280. WWW '12 (2012)
19. Paccanaro, A., Hinton, G.: Learning distributed representations of concepts using linear relational embedding. IEEE Trans. on Knowl. and Data Eng. 13, 232–244 (2001)
20. Salakhutdinov, R., Srebro, N.: Collaborative filtering in a non-uniform world: Learning with the weighted trace norm. tc (X) 10, 2 (2010)
21. Socher, R., Chen, D., Manning, C.D., Ng, A.Y.: Reasoning With Neural Tensor Networks For Knowledge Base Completion. In: Advances in Neural Information Processing Systems 26 (2013)
22. Sutskever, I., Salakhutdinov, R., Tenenbaum, J.: Modelling relational data using bayesian clustered tensor factorization. In: Adv. in Neur. Inf. Proc. Syst. 22 (2009)
23. Tucker, L.R.: Some mathematical notes on three-mode factor analysis. Psychometrika 31, 279–311 (1966)

24. Zhu, J.: Max-margin nonparametric latent feature models for link prediction. In: Proceedings of the 29th Intl Conference on Machine Learning (2012)