Journal of Artificial Intelligence Research 1 (2015) 1-15 Submitted 6/91; published 9/91

Combining Two And Three-Way Embeddings Models for Link Prediction in Knowledge Bases

Alberto García-Durán ALBERTO.GARCIA-DURAN@UTC.FR
Sorbonne universités, Université de technologie de Compiègne, CNRS, Heudiasyc UMR 7253
CS 60 319, 60 203 Compiègne cedex, France

Antoine Bordes ABORDES@FB.COM
Facebook AI Research
770 Broadway, New York, NY 10003. USA

Nicolas Usunier[*] USUNIER@FB.COM
Facebook AI Research
112 Avenue de Wagram, 75017 Paris, France

Yves Grandvalet YVES.GRANDVALET@UTC.FR
Sorbonne universités, Université de technologie de Compiègne, CNRS, Heudiasyc UMR 7253
CS 60 319, 60 203 Compiègne cedex, France

Abstract

This paper tackles the problem of endogenous link prediction for Knowledge Base completion. Knowledge Bases can be represented as directed graphs whose nodes correspond to entities and edges to relationships. 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 TATEC a happy medium obtained by complementing a high-capacity model with a simpler one, both pre-trained separately and then combined. We present several variants of this model with different kinds of regularization and combination strategies and show that this approach outperforms existing methods on different types of relationships by achieving state-of-the-art results on four benchmarks of the literature.

1. Introduction

Knowledge bases (KBs) are crucial tools to deal with the rise of data, since they provide ways to organize, manage and retrieve all digital knowledge. These repositories can cover any kind of area, from specific domains like biological processes (e.g. in GENEONTOLOGY[1]), to very generic purposes. FREEBASE[2], a huge collaborative KB which belongs to the Google Knowledge Graph, is an example of the latter kind which provides expert/common-level knowledge and capabilities to its users. An example of knowledge engine is WOLFRAMALPHA[3], an engine which answers to any natural language question, like `how far is saturn from the sun?`, with human-readable answers (`1,492` $\times 10^9$ `km`) using an internal KB.

[*] Part of this work was done while Nicolas Usunier was with Sorbonne universités, Université de technologie de Compiègne, CNRS, Heudiasyc UMR 7253.

[1] `http://geneontology.org`

[2] `http://www.freebase.com`

[3] `http://www.wolframalpha.com`

Such KBs can be used for question answering, but also for other natural language processing tasks like word-sense disambiguation (Navigli & Velardi, 2005), co-reference resolution (Ponzetto & Strube, 2006) or even machine translation (Knight & Luk, 1994).

KBs can be formalized as directed multi-relational graphs, whose nodes correspond to entities connected with edges encoding various kinds of relationship. Hence, one can also refer to them as multi-relational data. In the following we denote connections among entities via triples or *facts* (*head, label, tail*), where the entities *head* and *tail* are connected by the relationship *label.* Any information of the KB can be represented via a triple or a concatenation of several ones. Note that multi-relational data are not only present in KBs but also in recommender systems, where the nodes would correspond to users and products and edges to different relationships between them, or in social networks for instance.

A main issue with KBs is that they are far from being complete. Freebase currently contains thousands of relationships and more than 80 millions of entities, leading to billions of facts, but this remains only a very small portion out of all the human knowledge, obviously. And since question answering engines based on KBs like WolframAlpha are not capable of generalizing over their acquired knowledge to fill in for missing facts, they are *de facto* limited: they search for matches with a question/query in their internal KB and if this information is missing they can not provide a correct answer, even if they correctly interpreted the question. Consequently, huge efforts are devoted nowadays towards KBs construction or completion, via manual or automatic processes, or a mix of both. This is mainly divided in two tasks: entity creation or extraction, which consists in adding new entities to the KB and link prediction, which attempts to add connections between entities. This paper focuses on the latter case. Performing link prediction can be formalized as filling in incomplete triples like (*head*, *label*, ?) or (?, *label*, *tail*), by predicting the missing argument of the triple when such triple does not exist in the KB, yet. For instance, given the small example KB of Figure 1, made of 6 entities and 2 different relationships, and containing facts like (`Jared Leto, influenced_by, Bono`) or (`Michael Buble, profession, singer`), we would like to able to predict new links such as (`Frank Sinatra, profession, singer`), by using the fact that he influenced the singer `Michael Buble` for instance.

Link prediction in KBs is complex due to several issues. The entities are not homogeneously connected: some of them will have a lot of links with other entities, whereas others will be rarely connected. To illustrate the diverse characteristics present in the relationships we can take a look at FB15k, a subset of Freebase introduced in (Bordes, Usunier, García-Durán, Weston, & Yakhnenko, 2013b). In this data set of ∼14k entities and 1k types of relationships, entities have a mean number of triples of ∼400, but a median of 21 indicating that a large number of them appear in very few triples. Besides, roughly 25% of connections are of type 1-to-1, that is, a head is connected to at most one tail, and around 25% are of type Many-to-Many, that is, multiple heads can be linked to a tail and vice versa. As a result, diverse problems coexist in the same database. Another property of relationships that can have a big impact on the performance is the typing of their arguments. On FB15k, some relationships are very strongly typed like `/sports/sports_team/location`, where one always expects a football team as head and a location as tail, and some are far less precise such as `/common/webpage/category` where one expects only web page adresses as tail but pretty much everything else as head. A link prediction algorithm should be able to adapt to these different settings.

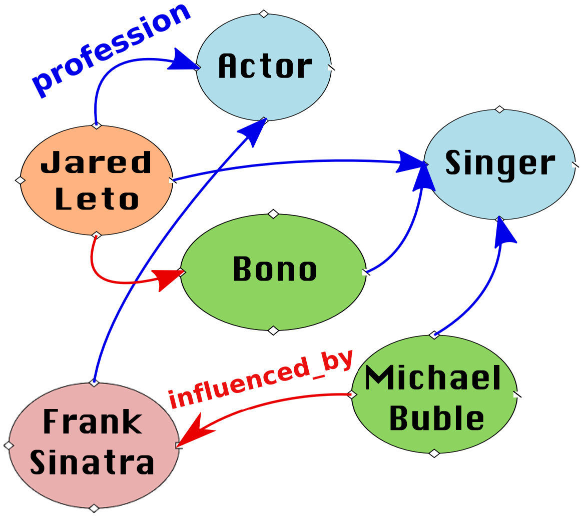


Figure 1: **Example of (incomplete) Knowledge Base** with 6 entities, 2 relationships and 7 facts.

Though there exists (pseudo-) symbolic approaches for link prediction based on Markov-logic networks (Kok & Domingos, 2007) or random walks (Lao, Mitchell, & Cohen, 2011), learning latent features representations of KBs constituents - the so-called *embedding methods* - have recently proved to be more efficient for performing link prediction in KBs, e.g. in (Bordes et al., 2013b; Wang, Zhang, Feng, & Chen, 2014b; Lin, Liu, Sun, Liu, & Zhu, 2015; Chang, Yih, Yang, & Meek, 2014; Wang, Zhang, Feng, & Chen, 2014a; Zhang, Salwen, Glass, & Gliozzo, 2014; Yang, Duan, Zhou, & Rim, 2014b). In all these works, entities are represented by low-dimensional vectors - the embeddings - and relationships act as operators on them: both embeddings and operators define a scoring function that is learned so that triples observed in the KBs have higher scores than unobserved ones. The embeddings are meant to capture underlying features that should eventually allow to create new links successfully. The scoring function is used to predict new links: the higher the score, the more likely a triple is to be true. Representations of relationships are usually specific (except in LFM (Jenatton, Le Roux, Bordes, & Obozinski, 2012) where there is a sharing of parameters across relationships), but embeddings of entities are shared for all relationships and allow to transfer information across them. The learning process can be considered as multi-task, where one task concerns each relationship, and entities are shared across tasks.

Embedding models can be classified according to the interactions that they use to encode the validity of a triple in their scoring function. If the joint interaction between the head, the label and the tail is used then we are dealing with a *3-way* model; but when the binary interactions between the head and the tail, the head and the label, and the label and the tail are the core of the model, then it is a *2-way* model. Both kinds of models represent the entities as vectors, but they differ in the way they model the relationships: 3-way models generally use matrices, whereas 2-way models use vectors. This difference in the capacity leads to a difference in the expressiveness of the models. The larger capacity of 3-way models (due to the large number of free parameters in matrices) is beneficial for the relationships appearing in a lot of triples, but detrimental for rare ones even if regularization is applied. Capacity is not the only difference between 2- and 3-way models, the information encoded by these two models is also different: we show in Sections 3 and 5.3.2 that both kinds of models assess the validity of the triple using different data patterns.

In this paper we introduce Tatec that encompass previous works by combining well-controlled 2-way interactions with high-capacity 3-way ones. We aim at capturing data patterns of both approaches by separately pre-training the embeddings of 2-way and 3-way models and using different embedding spaces for each of the two of them. We demonstrate in the following that otherwise – with no pre-training and/or no use of different embedding spaces – some features cannot be conveniently captured by the embeddings. Eventually, these pre-trained weights are combined in a second stage, leading to a combination model which outperforms most previous works in all conditions on four benchmarks from the literature, UMLS, Kinships, FB15k and SVO. Tatec is also carefully regularized since we systematically compared two different regularization schemes: adding penalty terms to the loss function or hard-normalizing the embedding vectors by constraining their norms.

This paper is an extension of (García-Durán, Bordes, & Usunier, 2014): we added a much more thorough study on regularization and on combination strategies for Tatec. Besides we propose experiments on several new benchmarks and a more complete comparison of our proposed method w.r.t. the state-of-the-art. We also give examples of predictions and projections in 2D of the obtained embeddings to provide some insights into the behavior of Tatec. The paper is organized as follows. Section 2 discusses previous works. Section 3 presents our model and justifies its choices. Detailed explanations of both the training procedure and the regularization schemes are given in Section 4. Finally, we present our experimental results on four benchmarks in Section 5.

2. Related work

In this section, we discuss the state-of-the-art of modeling large multi-relational databases, with a particular focus on embedding methods for knowledge base completion.

One of the simplest and most successful 2-way models is TransE (Bordes et al., 2013b). In that model, relationships are represented as translations in the embedding space: if (h, ℓ, t) holds, then the embedding of the *tail* t should be close to the embedding of *head* h plus some vector that depends on the *label* ℓ. This is a natural approach to model hierarchical and asymmetric relationships, which are common in knowledge bases such as Freebase. Several modifications to TransE have been proposed recently, TransH (Wang et al., 2014b) and TransR (Lin et al., 2015). In TransH, the embeddings of the entities are projected onto a hyperplane that depends on ℓ before the translation. The second algorithm, TransR, follows the same idea, except that the projection operator is a matrix that is more general than an orthogonal projection to a hyperplane. As we shall see in the next section, TransE corresponds to our Bigrams model with additional constraints on the parameters.

While 2-way models were shown to have very good performances on some KB datasets, they have limited expressiveness and they can fail dramatically on harder datasets. In contrast, 3-way models perform some form of low-rank tensor factorization, and in that respect can have extremely high expressiveness depending on the rank constraints. In the context of link prediction for multi-relational data, RESCAL (Nickel, Tresp, & Kriegel, 2011) follows natural modeling assumptions. Similarly to TransE, RESCAL learns one low-dimensional embedding for each entity. However, relationships are represented as a bilinear operator in the embedding space, i.e. each relationship corresponds to a matrix. The training objective of RESCAL is the Frobenius norm between the original data tensor and its low-rank reconstruction, whereas Tatec uses the margin ranking criterion of TransE. Another related 3-way model is SME(bilinear) (Bordes, Glorot, Weston, & Bengio, 2013a). The parameterization of SME(bilinear) is a constrained version of RESCAL, and also uses a ranking criterion as training objective.

The Latent Factor Model (LFM) (Jenatton et al., 2012) and the Neural Tensor Networks (NTN) (Socher, Chen, Manning, & Ng, 2013) can be seen as combinations of 2-way and 3-way interaction models, and in that sense are closer to our model Tatec. There are important differences between these algorithms and Tatec, though. Indeed, in addition to a different objective function and another kind of regularization, LFM uses a different parameterization than Tatec. First, LFM uses the same entity embeddings for the 2-way and 3-way interaction terms, whereas we show that using different entity embeddings for the two models can lead to very significant improvements. Second, in LFM, some parameters of the relationships between the 2-way and the 3-way interaction terms are also shared, which is not the case in Tatec. Indeed, such joint parameterization might reduce the expressiveness of the 2-way interaction terms which, as we argue in Section 3.3, should be left with maximum degrees of freedom. The NTN has a more general parameterization than LFM, but still uses the same entity embeddings for the 2-way and 3-way interaction terms. Also, NTN has two layers and a non-linearity after the first layer, while our model does not add any nonlinearity after the embedding step. In order to have a more precise overview of the differences between the approaches, we give in Section 3 (Table 1) the formulas of the scoring functions of these related works.

While there has been a lot of focus recently on algorithms purely based on learning embeddings for entities and/or relationships, many earlier alternatives had been proposed. We discuss works carried ou in the Bayesian clustering framework, as well as approaches that explicitly use the graph structure of the data. The Infinite Relational Model of (Kemp, Tenenbaum, Griffiths, Yamada, & Ueda, 2006), which is a nonparametric extension of the Stochastic Blockmodel (Wang & Wong, 1987), is a Bayesian clustering approach that learns clusters of entities of the same kind, i.e. groups of entities that can have similar relationships with other entities. This work was followed by (Sutskever, Salakhutdinov, & Tenenbaum, 2009), a 3-way tensor factorization model based on Bayesian clustering in which entities within a cluster share the same distribution of embeddings.

The last line of work we discuss here are the approaches that use the structure of the graph in a symbolic way. In that line of work, the Path Ranking Algorithm (PRA) (Lao et al., 2011) estimates the probability of an unobserved fact as a function of the different paths that go from the subject to the object in the multi-relational graph; learning consists in finding, for each relationship, a weight associated to a kind of path (represented as a sequence of relationships) linking two entities. The PRA is used in the Knowledge Vault project (Dong, Gabrilovich, Heitz, Horn, Lao, Murphy, Strohmann, Sun, & Zhang, 2014) in conjunction with an embedding approach. Thus, even though we do not consider these symbolic approaches here, they could also be combined with our embedding model if desired.

3. TATEC

We now describe our model and the motivations underlying our parameterization.

3.1 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 ones. Our proposed model, Tatec, learns embeddings of entities in a low dimensional vector space, say $\mathbb{R}^d$, and parameters of operators on $\mathbb{R}^d \times \mathbb{R}^d$, most of these operators being associated to a single relationship. 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) Bigram or the 2-way interaction term:

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

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, 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 interaction term:

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

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 even have different dimensions.

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 interaction term of the model is similar to that of (Bordes et al., 2013a), 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 (Bordes et al., 2013b), which is the special case where $\mathbf{r}_1^\ell = -\mathbf{r}_2^\ell$, $\mathbf{D}$ is the identity matrix, and the entity embeddings are constrained to lie on the unit sphere.

The 3-way term corresponds exactly to the model used by the collective factorization method RESCAL (Nickel et al., 2011), and we chose it for its high expressiveness on complex relationships. Indeed, as we said earlier in the introduction, 3-way models can basically represent any kind of interaction among entities. The combination of 2- and 3-way terms has already been used in (Jenatton et al., 2012; Socher et al., 2013), but, besides a different parameterization, Tatec contrasts with them by the additional freedom brought by using different embeddings in the two interaction terms. In LFM (Jenatton et al., 2012), 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 in the image of the matrix ($\mathbf{D} = \mathbf{0}$ in their work). These global constraints severely limited the expressiveness of the 3-way model, and act as a stringent regularization that reduces the expressiveness of the 2-way model, which, as we explain in Section 3.3, should be left with maximum degrees of freedom. We are similar to NTN (Socher et al., 2013) in the respect that we do not share any parameter between relations. 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 (for instance, the gradients have a larger magnitude).

3.2 Term combination

We study two strategies for combining the bigram and trigram scores as indicated in Equation (1). In both cases, both s_1 and s_2 are first trained separately as we detail in Section 4 and then combined. The difference between our two strategies depends on whether we jointly update (or fine-tune) the parameters of s_1 and s_2 in a second phase or not.

Fine tuning This first strategy, denoted Tatec-ft, simply consists in summing both scores following Equation (1).

$$s_{FT}(h,\ell,t) = \langle \mathbf{r}_1^\ell | \mathbf{e}_1^h \rangle + \langle \mathbf{r}_2^\ell | \mathbf{e}_1^t \rangle + \langle \mathbf{e}_1^h | \mathbf{D} | \mathbf{e}_1^t \rangle + \langle \mathbf{e}_2^h | \mathbf{R}^\ell | \mathbf{e}_2^t \rangle$$

All parameters of s_1 and s_2 (and hence of s) are then fine-tuned in a second training phase to accommodate for their combination. This version could be trained directly without pre-training s_1 and s_2 separately but we show in our experiments that this is detrimental.

Linear combination The second strategy combines the bigram and trigram terms using a linear combination, without jointly fine-tuning their parameters that remain unchanged after their pre-training. The score s is hence defined as follows:

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

The combination weights δ_i^ℓ depend on the relationship and are learned by optimizing the ranking loss (defined later in (4)) using L-BFGS, with an additional quadratic penalization term, $\sum_k \frac{||\delta^k||_2^2}{\sigma_k+\epsilon}$, subject to $\sum_k \sigma_k = \alpha$. The intuition behind this particular penalization for the δs is that it is equivalent to a LASSO penalization (Grandvalet, 1998) and our initial idea was to enforce sparsity among δ parameters, since we thought that some of the interactions, depending on the relationship, were rather noisy than beneficial. However in the following we will see that the validated value of α does not yield a sparse solution. This version of Tatec is denoted Tatec-lc in the following.

3.3 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.

3.3.1 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. (Koren, Bell, & Volinsky, 2009)):

$$R_{ui} \approx \langle \mathbf{P}_u | \mathbf{Q}_i \rangle + \alpha_u + \beta_i + \mu \tag{2}$$

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 interaction 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 parameterization of the RESCAL algorithm (Nickel et al., 2011) and plays a role analogous to the term $\langle \mathbf{P}_u | \mathbf{Q}_i \rangle$ of the matrix factorization model for collaborative filtering (2). The bigram term **(B)** then plays the role of biases for each fiber of the tensor,[4] i.e.

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

and thus is the analogue for tensors to the term $\alpha_u + \beta_i$ in the matrix factorization model (2). 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 (3). 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.

3.3.2 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 (Jenatton et al., 2012; Socher et al., 2013). We motivate this choice in this section.

It is important to notice that biases in the matrix factorization model (2), 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 (2) 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 compared to

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

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 (Koren et al., 2009) for instance, all terms of (2) are trained using a squared error as a measure 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 (Salakhutdinov & Srebro, 2010) on $\mathbf{P}\mathbf{Q}^T$. 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 $\mathbf{P}\mathbf{Q}^T$. 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 sometimes turns out to be more detrimental than effective in experiments. 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.

3.3.3 2-way interactions as entity types+similarity

Having a part of the model that is less expressive, but less regularized (see Subsection 4.2) 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 interaction 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

Table 1: **Scoring function for several models of the literature.** Capitalized letters denote matrices and lower cased ones, vectors.

Model	Score ($s(h, \ell, t)$)
TransE	$\|\mathbf{e}^h + \mathbf{r}^\ell - \mathbf{e}^t\|_2$
TransH	$\|(\mathbf{e}^h - \langle \mathbf{w}^\ell \vert \mathbf{e}^h \mathbf{w}^\ell \rangle) + \mathbf{r}^\ell - (\mathbf{e}^t - \langle \mathbf{w}^\ell \vert \mathbf{e}^t \mathbf{w}^\ell \rangle)\|_2^2$
TransR	$\|\langle \mathbf{e}^h \vert \mathbf{M}_\ell \rangle + \mathbf{r}^\ell - \langle \mathbf{e}^\mathbf{t} \vert \mathbf{M}_\ell \rangle\|_\mathbf{2}$
RESCAL	$\langle \mathbf{e}^h \vert \mathbf{R}^\ell \vert \mathbf{e}^t \rangle$
LFM	$\langle y \vert \mathbf{R}^\ell \vert y' \rangle + \langle \mathbf{e}^h \vert \mathbf{R}^\ell \vert \mathbf{z} \rangle + \langle \mathbf{z} \vert \mathbf{R}^\ell \vert \mathbf{e}^t \rangle + \langle \mathbf{e}^h \vert \mathbf{R}^\ell \vert \mathbf{e}^t \rangle$

the 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, the capital of France can easily be predicted by looking 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 $\mathbf{D}$.[5] 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. Training

4.1 Ranking objective

Training Tatec is carried out using stochastic gradient descent over a ranking objective function, which is designed to give higher scores to positive triples (facts that express true and verified information from the KB) than to negative ones (facts that are supposed to express false information). These negative triples can be provided by the KB, but often they are not, so we need a process to turn positive triples into corrupted ones to carry out our discriminative training. A simple approach consists in creating negative examples by

[5] 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 Euclidean distances, the equivalence still holds under 2-norm regularization of the embeddings.

replacing one argument of a positive triple by a random element. This way is simple and efficient in practice but may introduce noise by creating wrong negatives.

Let $\mathcal{S}$ be the set of positive triples provided by the KB, we optimize the following ranking loss function:

$$\sum_{(h,\ell,t)\in\mathcal{S}} \sum_{(h',\ell',t')\in\mathcal{C}(h,\ell,t)} \big[\gamma - s(h,\ell,t) + s(h',\ell',t')\big]_+ \tag{4}$$

where $[z]_+ = \max(z,0)$ and $\mathcal{C}(h,\ell,t)$ is the set of corrupted triples. Depending on the application, this set can be defined in 3 different ways:

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

The margin γ is an hyperparameter that defines the minimum gap between the score of a positive triple and its negative one's. The stochastic gradient descent is performed in a minibatch setting. At each epoch the data set is shuffled and split into disjoint minibatches of m triples and 1 or 2 (see next section) negative triples are created for every positive one. We use two different learning rates λ_1 and λ_2, one for the Bigrams and one the Trigram model; they are kept fixed during the whole training.

We are interested in both Bigrams and Trigram terms of Tatec to capture different data patterns, and using a random initialization of all weights can not necessarily lead to such a solution. Hence, we first pre-train separately $s_1(h,\ell,t)$ and $s_2(h,\ell,t)$, and then we use these learned weights to initialize that of the full model. Training of Tatec is hence carried out in two phases: a (disjoint) pre-training and either a (joint) fine-tuning for Tatec-ft or a learning of the combination weights for Tatec-lc. Both pre-training and fine-tuning are stopped using early stopping on a validation set, and follow the training procedure that is summarized in Algorithm 1, for the unregularized case. Training of the linear combination weights of Tatec-lc is stopped at convergence of L-BFGS.

4.2 Regularization

Previous work on embedding models have used two different regularization strategies: either by constraining the entity embeddings to have, at most, a 2-norm of value ρ_e (García-Durán et al., 2014) or by adding a 2-norm penalty on the weights (Wang et al., 2014b; Lin et al., 2015) to the objective function (4). In the former, which we denote as *hard regularization*, regularization is performed by projecting the entity embeddings after each minibatch onto the 2-norm ball of radius ρ_e. In the latter, which we denote as *soft regularization*, a penalization term of the form $[||\mathbf{e}||_2^2 - \rho_e^2]_+$ for the entity embeddings $\mathbf{e}$ is added. The soft scheme allows the 2-norm of the embeddings to grow further than ρ_e, with a penalty.

To control the large capacity of the relation matrices in the Trigram model, we have adapted the two regularization schemes: in the *hard* scheme, we force the relation matrices to have, at most, a Frobenius norm of value ρ_l, and in the *soft* one, we include a penalization term of the form $[||\mathbf{R}||_F^2 - \rho_l^2]_+$ to the loss function (4) . As a result, in the *soft* scheme the following regularization term is added to the loss function (4): $C_1[||\mathbf{e}_1||_2^2 - \rho_e^2]_+ + C_2\big([||\mathbf{e}_2||_2^2 - \rho_e^2]_+ + [||\mathbf{R}||_F^2 - \rho_l^2]_+\big)$, where C_1 and C_2 are hyperparameters that weight the importance of

Algorithm 1 Learning unregularized Tatec.

```
input  Training set $S = \{(h, l, t)\}$, margin $\gamma$, learning rates $\lambda_1$ and $\lambda_2$
initialization
    - for Bigrams: $\mathbf{e}_1 \leftarrow$ uniform$(-\frac{6}{\sqrt{d_1}}, \frac{6}{\sqrt{d_1}})$ for each entity $e$
                   $\mathbf{r}_1, \mathbf{r}_2 \leftarrow$ uniform$(-\frac{6}{\sqrt{d_1}}, \frac{6}{\sqrt{d_1}})$ for each $\ell$
                   $\mathbf{D} \leftarrow$ uniform$(-\frac{6}{\sqrt{d_1}}, \frac{6}{\sqrt{d_1}})$
    - for Trigram: $\mathbf{e}_2 \leftarrow$ uniform$(-\frac{6}{\sqrt{d_2}}, \frac{6}{\sqrt{d_2}})$ for each entity $e$
                   $\mathbf{R} \leftarrow$ uniform$(-\frac{6}{\sqrt{d_2}}, \frac{6}{\sqrt{d_2}})$ for each $\ell$
    - for Tatec-ft: pre-trained weights of Bigrams and Trigram
All the embeddings are normalized to have a 2- or Frobenius-norm equal to 1.
loop
    $S_{batch} \leftarrow sample(S, m)$ // sample a training minibatch of size $m$
    $T_{batch} \leftarrow \emptyset$ // initialize a set of pairs of examples
    for $(h, \ell, t) \in S_{batch}$ do
        $(h', \ell', t') \leftarrow$ sample a negative triple according to the selected strategy $\mathcal{C}(h, \ell, t)$
        $T_{batch} \leftarrow T_{batch} \cup \{((h, \ell, t), (h', \ell', t'))\}$ // record the pairs of examples
    end for
    Update parameters using gradients $\sum_{((h,\ell,t),(h',\ell',t')) \in T_{batch}} \nabla[\gamma - s(h, \ell, t) + s(h', \ell', t')]_+$:
        - for Bigrams: $s = s_1$
        - for Trigram: $s = s_2$
        - for Tatec-ft: $s = s_1 + s_2$
end loop
```

each soft constraint. In terms of practicality, the bigger flexibility of the soft version comes with one more hyperparameter. In the following, the suffixes soft and hard are used to refer to either of those regularization scheme. Tatec has also an other implicit regularization factor since it is using the same entity representation for an entity regardless of its role as head or tail.

To sum up, in the hard regularization case, the optimization problem for Tatec-ft is:

$$\begin{aligned} \min & \sum_{(h,\ell,t)\in\mathcal{S}} \sum_{(h',\ell',t')\in\mathcal{C}(h,\ell,t)} \big[\gamma - s(h, \ell, t) + s(h', \ell', t')\big]_+ \\ \text{s.t.} \quad & ||\mathbf{e}_1^i||_2 \le \rho_e \quad \forall i \in [\![E]\!] \\ & ||\mathbf{e}_2^i||_2 \le \rho_e \quad \forall i \in [\![E]\!] \\ & ||R^\ell||_F \le \rho_l \quad \forall \ell \in [\![L]\!] \end{aligned}$$

And in the soft regularization case it is:

$$\begin{aligned} \min \sum_{(h,\ell,t)\in\mathcal{S}} \sum_{(h',\ell',t')\in\mathcal{C}(h,\ell,t)} & [\gamma - s(h, \ell, t) + s(h', \ell', t')]_+ + C_1 \sum_{i\in[\![E]\!]} [||\mathbf{e}_1^i||_2^2 - \rho_e^2]_+ \\ & + C_2\big(\sum_{i\in[\![E]\!]} [||\mathbf{e}_2^i||_2^2 - \rho_e^2]_+ + \sum_{\ell\in[\![L]\!]} [||\mathbf{R}^\ell||_F^2 - \rho_l^2]_+ \big) \end{aligned}$$

where $s(h, \ell, t) = \langle \mathbf{r}_1^\ell | \mathbf{e}_1^h \rangle + \langle \mathbf{r}_2^\ell | \mathbf{e}_1^t \rangle + \langle \mathbf{e}_1^h | \mathbf{D} | \mathbf{e}_1^t \rangle + \langle \mathbf{e}_2^h | \mathbf{R}^\ell | \mathbf{e}_2^t \rangle$ in both cases.

Table 2: **Statistics of the data sets** used in this paper and extracted from four knowledge bases: FB15k, SVO, Kinships and UMLS.

Data set	FB15k	SVO	Kinships	UMLS
Entities	14,951	30,605	104	135
Relationships	1,345	4,547	26	49
Training examples	483,142	1,000,000	224973	102612
Validation examples	50,000	50,000	28122	89302
Test examples	59,071	250,000	28121	89302

5. Experiments

This section presents various experiments to illustrate how competitive is Tatec with respect to several state-of-the-art models on 4 benchmarks from the literature: UMLS, Kinships, FB15k and SVO. The statistics of these data sets are given in Table 2. All versions of Tatec and of its components Bigrams and Trigram are compared with the state-of-the-art models for each database.

5.1 Experimental setting

This section details the protocols used in our various experiments.

5.1.1 Datasets and metrics

Our experimental settings and evaluation metrics are borrowed from previous works, so as to allow for result comparisons.

UMLS/Kinships Kinships (Denham, 1973) is a KB expressing the relational structure of the kinship system of the Australian tribe Alyawarra, and UMLS (McCray, 2003) is a KB of biomedical high-level concepts like diseases or symptoms connected by verbs like `complicates`, `affects` or `causes`. For these data sets, the whole set of possible triples, positive or negative, is observed. We used the area under the precision-recall curve as metric. The dataset was split in 10-folds for cross-validation: 8 for training, 1 for validation and the last one for test. Since the number of available negative triples is much bigger than the number of positive triples, the positive ones of each fold are replicated to match the number of negative ones. These negative triples correspond to the first setting of negative examples of Section 4.1. The number of training epochs was fixed to 100. Bigrams, Trigram and Tatec models were validated every 10 epochs using the AUC under the precision-recall curve as validation criterion over 1,000 randomly chosen validation triples - keeping the same proportion of negative and positive triples. For TransE, which we ran as baseline, we validated every 10 epochs as well.

FB15k Introduced in (Bordes et al., 2013b), this data set is a subset of Freebase, a very large database of generic facts gathering more than 1.2 billion triples and 80 million entities. For evaluation on it, we used a ranking metric. The head of each test triple is replaced by each of the entities of the dictionary in turn, and the score is computed for each of them. These scores are sorted in descending order and the rank of the correct entity is stored. The same procedure is repeated when removing the tail instead of the head.

The mean of these ranks is the *mean rank*, and the proportion of correct entities ranked in the top 10 is the *hits@10*. This is called the *raw* setting. In this setting correct positive triples can be ranked higher than the target one and hence be counted as errors. Following (García-Durán et al., 2014), in order to reduce this noise in the measure, and thus granting a clearer view on ranking performance, we remove all the positive triples that can be found in either the training, validation or testing set, except the target one, from the ranking. This setting is called *filtered*.

Since FB15k is made up only of positive triples, the negative ones have to be generated. To do that, in each epoch we generate two negative triples per positive by replacing a single unit of the positive triple by a random entity (once the head and once the tail). This corruption approach implements the prior knowledge that unobserved triples are likely to be invalid, and has been widely used in previous work when learning embeddings of knowledge bases or words in the context of language models. These negative triples correspond to the second setting of negative examples of Section 4.1. We ran 500 training epochs for both TransE, Bigrams, Trigram and Tatec, and using the final filtered mean rank as validation criterion. If several models statistically have similar filtered mean ranks, we take the *hits*@10 as secondary validation criterion.[6] Since for this dataset, training, validation and test sets are fixed, to give a confidence interval to our results, we randomly split the test set into 4 subsets before computing the evaluation metrics. We do this 5 times, and finally we compute the mean and the standard deviation over these 20 values for mean rank and hits@10.

SVO SVO is a database of nouns connected by verbs through subject-verb-direct object relations and extracted from Wikipedia articles. It has been introduced in (Jenatton et al., 2012). For this database we perform a verb prediction task, where one has to assign the correct verb given two nouns acting as subject and direct object; in other words, we present results of ranking *label* given *head* and *tail*. As for FB15k, two ranking metrics are computed, the *mean rank* and the *hits@5%*, which is the proportion of predictions for which the correct verb is ranked in the top 5% of the total number of verbs, that is within the top 5% of 4,547 $\approx$ 227. We use the *raw* setting for SVO. Due to the different kind of task (predicting *label* instead of predicting *head*/*tail*), the negative triples have been generated by replacing the label by a random verb. These negative triples correspond to the third setting of negative examples of Section 4.1. For TransE, Bigrams and Trigram the number of epochs has been fixed to 500 and they were validated every 10 epochs. For Tatec we ran only 10 epochs, and validated for each. The mean rank has been chosen as validation criterion over 1,000 random validation triples.

5.1.2 Implementation

To pre-train our Bigrams and Trigram models we validated the learning rate for the stochastic gradient descent among $\{0.1, 0.01, 0.001, 0.0001\}$ and the margin among $\{0.1, 0.25, 0.5, 1\}$. The radius ρ_e determining the value from which the L_2-norm of the entity embeddings are penalized has been fixed to 1, but the radius ρ_l of the Trigram model has

[6] Results on both FB15k and SVO with TransE and Tatec are provided in (García-Durán et al., 2014), however in these works the hyperparameters were validated on a smaller validation set, that led to suboptimal results.

been validated among $\{0, 1, 5, 10, 20\}$. Due to the different size of these KBs, the embedding dimension d has been validated in different ranges. For SVO it has been selected among $\{25, 50\}$, among $\{50, 75, 100\}$ for FB15k and among $\{10, 20, 40\}$ for UMLS and Kinships. When the soft regularization is applied, the regularization parameter has been validated among $\{0, 0.0001, 0.001, 0.01, 0.1, 1, 10, 100\}$. For fine-tuning Tatec, the learning rates were selected among the same values for learning the Bigrams and Trigram models in isolation, independent of the values chosen for pre-training, and so are the margin and for the penalization terms C_1 and C_2 if the soft regularization is used. The configurations of the model selected using their performance on the validation set are given in Appendix A.

Training of the combination weights of Tatec-lc is carried out in an iterative way, by alternating optimization of δ parameters via L-BFGS, and update of σ parameters using $\sigma_i^* = \frac{\alpha||\delta_i||_2}{\sum_i ||\delta_i||_2}$, until some stopping criterion is reached. The δ parameters are initialized to 1 and the α value is validated among $\{0.1, 1, 10, 100, 250, 500, 1000\}$.

5.1.3 Baselines

Variants We performed breakdown experiments with 2 different versions of Tatec to assess the impact of its various aspects. These variants are:

- Tatec-ft-no-pretrain: Tatec-ft without pre-training $s_1(h, l, t)$ and $s_2(h, l, t)$.
- Tatec-ft-shared: Tatec-ft but sharing the entities embeddings between $s_1(h, l, t)$ and $s_2(h, l, t)$ and without pre-training.

The experiments with these 3 versions of Tatec have been performed in the soft regularization setting. Their hyperparameters were chosen using the same grid as above.

Previous models We retrained TransE ourselves with the same hyperparameter grid as for Tatec and used it as a running baseline on all datasets, using either soft or hard regularization. In addition, we display the results of the best performing methods of the literature on each dataset, with values extracted from the original papers.

On UMLS and Kinships, we also report the performance of the 3-way models RESCAL, LFM and the 2-way SME(linear). On FB15k, recent variants of TransE, such as TransH, TransR and cTransR (Lin et al., 2015) have been chosen as main baselines. Both in TransH and TransR/cTransR, the optimal values of the hyperparameters as the dimension, the margin or the learning rate have been selected within similar ranges as those for Tatec. On SVO, we compare Tatec with three different approaches: Counts, the 2-way model SME(linear) and the 3-way LFM. 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 (Jenatton et al., 2012), and we followed their experimental setting. Since the results in this paper are only available in the raw setting, we restricted our experiments to this configuration on SVO as well.

5.2 Results

We recall that the suffixes soft or hard refer to the regularization scheme used, and the suffixes ft and lc to the combination strategy of Tatec.

Table 3: **Test AUC under the precision-recall curve on UMLS and Kinships** for models from the literature (top) and Tatec (bottom).. Best performing methods are in bold.

Model	UMLS	Kinships
SME(linear)	0.983 ± 0.003	0.907 ± 0.008
RESCAL	0.98	**0.95**
LFM	**0.990** ± 0.003	**0.946** ± 0.005
TransE-soft	0.734 ± 0.033	0.135 ± 0.005
TransE-hard	0.706 ± 0.034	0.134 ± 0.005
Bigrams-hard	0.936 ± 0.020	0.140 ± 0.004
Trigram-hard	0.980 ± 0.006	**0.943** ± 0.009
Tatec-ft-hard	0.984 ± 0.004	0.876 ± 0.012
Bigrams-soft	0.936 ± 0.018	0.141 ± 0.003
Trigram-soft	0.983 ± 0.004	**0.948** ± 0.008
Tatec-ft-soft	**0.985** ± 0.004	0.919 ± 0.008
Tatec-lc-soft	**0.985** ± 0.004	**0.941** ± 0.009

Table 4: **Test results on FB15k and SVO** for models from the literature (top), Tatec (middle) and variants (bottom). Best performing methods are in bold. The *filtered* setting is used for FB15k and the *raw* setting for SVO.

	FB15k		SVO	
Model	Mean Rank	Hits@10	Mean Rank	Hits@5%
Counts	-	-	517.4	72
SME(linear)	-	-	199.6	77
LFM	-	-	195	78
TransH	87	64.4	-	-
TransR	77	68.7	-	-
cTransR	75	70.2	-	-
TransE-soft	**50.7** ± 2.0	71.5 ± 0.3	282.5 ± 1.7	70.6 ± 0.2
TransE-hard	**50.6** ± 2.0	71.5 ± 0.3	282.8 ± 2.3	70.6 ± 0.2
Tatec-no-pretrain	97.1 ± 3.9	65.7 ± 0.2	-	-
Tatec-shared	94.8 ± 3.2	63.4 ± 0.3	-	-
Bigrams-hard	94.5 ± 2.9	67.5 ± 0.4	219.2 ± 1.9	77.6 ± 0.1
Trigram-hard	137.7 ± 7.1	56.1 ± 0.4	187.9 ± 1.2	79.5 ± 0.1
Tatec-ft-hard	59.8 ± 2.6	**77.3** ± 0.3	188.5 ± 1.9	79.8 ± 0.1
Bigrams-soft	87.7 ± 4.1	70.0 ± 0.2	211.9 ± 1.8	77.8 ± 0.1
Trigram-soft	121.0 ± 7.2	58.0 ± 0.3	189.2 ± 2.1	79.5 ± 0.2
Tatec-ft-soft	57.8 ± 2.3	**76.7** ± 0.3	185.4 ± 1.5	**80.0** ± 0.1
Tatec-lc-soft	68.5 ± 3.2	72.8 ± 0.2	**182.6** ± 1.2	**80.1** ± 0.1

5.2.1 UMLS and Kinships

The results for these two knowledge bases are provided in Table 3. In UMLS, most models are performing well. The combination of the Bigrams and Trigram models is slightly better than the Trigram alone but it is not significant. It seems that the constituents of Tatec, Bigrams and Trigram, do not encode very complementary information and their combination does not bring much improvement. Basically, on this dataset, many methods are somewhat as efficient as the best one, LFM. The difference between TransE and Bigrams on this dataset illustrates the potential impact of the diagonal matrix $\mathbf{D}$, which does not constrain embeddings of both head and tail entities of a triple to be similar.

Regarding Kinships, there is a big gap between 2-way models like TransE and 3-way models like RESCAL. The cause of this deterioration comes from a peculiarity of the positive triples of this KB: each entity appears 104 times – the number of entities in this KB – as head and it is connected to the 104 entities – even itself – only once. In other words, the conditional probabilities $P(head|tail)$ and $P(tail|head)$ are totally uninformative. This has a very important consequence for the 2-way models since they highly rely on such information: for Kinships, the interaction head-tail is, at best, irrelevant, though in practice this interaction may even introduce noise.

Due to the poor performance of the Bigrams model, when it is combined with the Trigram model this combination can turn out to be detrimental w.r.t. to the performance of Trigram in isolation: 2-way models are quite noisy for this KB and we cannot take advantage of them. On the other side the Trigram model logically reaches a very similar performance to RESCAL, and similar to LFM as well. Performance of Tatec versions based on fine-tuning of the parameters (Tatec-ft) are worse than that of Trigram because Bigrams degrades the model. Tatec-lc, using a – potentially sparse – linear combination of the models, does not have this drawback since it can completely cancel out the influence of bigram model. As a conclusion from the experiments in this KB, when one of the components of Tatec is quite noisy, we should directly remove it and Tatec-lc can do it automatically. The soft regularization setting seems to be slightly better also.

5.2.2 FB15k

Table 4 (left) displays results on FB15k. Unlike for Kinships, here the 2-way models outperform the 3-way models in both mean rank and hits@10. The simplicity of the 2-way models seems to be an advantage in FB15k: this is something that was already observed in cite (Yang, Yih, He, Gao, & Deng, 2014a). The combination of the Bigrams and Trigram models into Tatec leads to an impressive improvement of the performance, which means that for this KB the information encoded by these 2 models are complementary. Tatec outperforms all the existing methods – except TransE in mean rank – with a wide margin in hits@10. Bigrams-soft performs roughly like cTransR, and better than its counterpart Bigrams-hard. Though Trigram-soft is better than Trigram-hard as well, Tatec-ft-soft and Tatec-ft-hard converge to very similar performances. Fine-tuning the parameters is this time better than simply using a linear combination even if Tatec-lc is still performing well.

Tatec-ft outperforms both variants Tatec-shared and Tatec-no-pretrain by a wide margin, which confirms that both pre-training and the use of different embeddings spaces are essential to properly collect the different data patterns of the Bigrams and Trigram models: by sharing the embeddings we constrain too much the model, and without pre-training Tatec is not able to encode the complementary information of its constituents. The performance of Tatec in these cases is in-between the performances of the soft version

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.
TransE-soft	76.2	93.6	47.5	70.2	76.7	50.9	93.1	72.9
TransH	66.8	87.6	28.7	64.5	65.5	39.8	83.3	67.2
TransR	78.8	89.2	34.1	69.2	79.2	37.4	90.4	72.1
cTransR	81.5	89	34.7	71.2	80.8	38.6	90.1	73.8
Bigrams-soft	76.2	90.3	37.4	70.1	75.9	44.4	89.8	72.8
Trigram-soft	56.4	79.6	30.2	57	53.1	28.8	81.6	60.8
Tatec-ft-soft	79.3	93.2	42.3	77.2	78.5	51.5	92.7	80.7

of the Bigrams and Trigram models, which indicates that they converge to a solution that is not even able to reach the best performance of their constituent models.

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. Bigrams and Trigram models cooperate in a constructive way for all the types of relationship when predicting both the head and tail. Tatec-ft is remarkably better for M-to-M relationships.

5.2.3 SVO

Tatec achieves also a very good performance on this task since it outperforms all previous methods on both metrics. As before, both regularization strategies lead to very similar performances, but the soft setting is slightly better. In terms of hits@5%, Tatec outperforms its constituents, however in terms of mean rank the Bigrams model is considerably worse than Trigram and Tatec. The performance of LFM is in between the Trigram and Bigrams models, which confirms the fact that sharing the embeddings in the 2- and 3-way terms can actually prevent to make the best use of both types of interaction.

As for Kinships, since here the performance of Bigrams is much worse than that of Trigram, Tatec-lc is very competitive. It seems that when Bigrams and Trigram perform well for different types of relationships (such as in FB15k), then combining them via fine-tuning (i.e. Tatec-ft) allows to get the best of both; however, if one of them is consistently performing worse on most relationships as it seems to happen for Kinships and SVO, then Tatec-lc is a good choice since it can cancel out any influence of the bad model. However, Table 6, depicting training times of various models on FB15k, shows that training Tatec-lc is around twice as slow as training Tatec-ft.

5.3 Illustrative experiments

This last experimental section provides some illustrations and insights on the performance of Tatec and TransE.

Table 6: **Training times on FB15k** on a single core.

Model	Train. time
Bigrams-soft	∼ 6h
Trigram-soft	∼ 12h
Tatec-ft-soft	∼ 13h
Tatec-lc-soft	∼22h

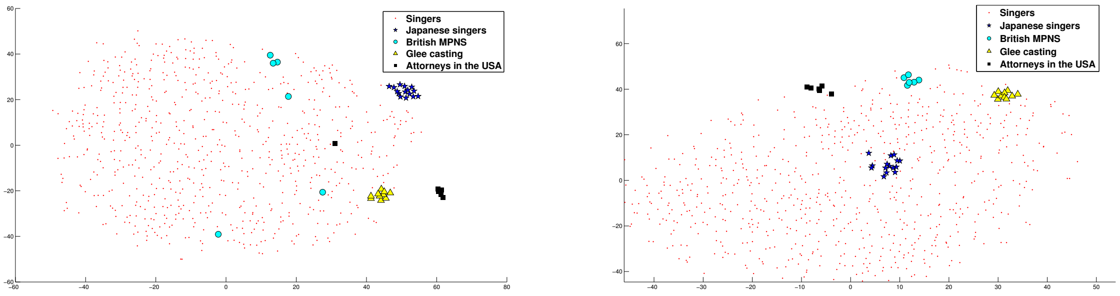
Table 7: **Examples of predictions on FB15k.** Given an entity and a relation type from a test triple, Tatec fills in the missing slot. In bold is the expected correct answer.

Triple	Top-10 predictions
(poland_national_football_team, /sports_team/location, ?)	Mexico, South_Africa, **Republic_of_Poland** Belgium, Puerto_Rico, Austria, Georgia Uruguay, Colombia, Hong_Kong
(?, /film/film_subject/films , remember_the_titans)	**racism**, vietnam_war, aviation, capital_punishment television, filmmaking, Christmas female, english_language, korean_war
(noam_chomsky, /people/person/religion, ?)	**atheism**, agnosticism, catholicism, ashkenazi_jews buddhism, islam, protestantism baptist, episcopal_church, Hinduism
(?, /webpage/category, official_website)	supreme_court_of_canada, butch_hartman, robyn_hitchcoc, mercer_university clancy_brown, dana_delany, hornets grambling_state_university, dnipropetrovsk, juanes

5.3.1 TransE and symmetrical relationships

TransE has a peculiar behavior: it performs very well on FB15k but quite poorly on all the other datasets. Looking in detail at FB15k, we noticed that this database is made up of a lot of pairs of symmetrical relationships such as `/film/film/subjects` and `/film/film_subject/films`, or `/music/album/genre` and `/music/genre/albums`. The simplicity of the translation model of TransE works well when, for predicting the validity of an unknown triple, the model can make use of its symmetrical counterpart if it was present in the training set. Specifically, 45,817 out of 59,071 test triples of FB15k have their symmetrical triple in the training set. If we split the test triples into two subsets, one containing the test triples for which the symmetrical triple has been used in the learning stage and the other containing those ones whose symmetrical triple does not exit in the training set, the overall mean rank of TransE of 50.7 is decomposed into a mean rank of 17.5 and 165.7, and the overall hits@10 of 71.5 is decomposed into 76.6 and 53.7, respectively. TransE makes a very adequate use of this particular feature. In the original TransE paper (Bordes et al., 2013b), the algorithm is shown to perform well on FB15k and on a dataset extracted from the KB WordNet (Miller, 1995): we suspect that the WordNet dataset also contains symmetrical counterparts of test triples in the training set (such as hyperonym vs hyponym, meronym vs holonym).

Tatec can also make use of this information and is, as expected, much better on relations with symmetrical counterparts in train: on FB15k, the mean rank of Tatec-ft-soft is of 17.5 for relations with symmetrical counterparts 197.4 instead and hits@10 is of 84.4% instead of 50%. Yet, as results on other datasets show, Tatec is also able to generalize when more complex information needs to be taken into account.



(a) Embeddings of Trigram

(b) Embeddings of Bigrams

Figure 2: **Embeddings obtained by Trigram and Bigrams models** and projected in 2-D using t-SNE. MPNS stands for `Main Profession is Not Singer`.

5.3.2 Anecdotal examples

Some examples of predictions by Tatec on FB15k are displayed in Table 7. In the first row, we want to know the answer to the question `What is the location of the polish national football team?`; among the possible answers we find not only locations, but more specifically countries, which makes sense for a national team. For the question `What is the topic of the film 'Remember the titans'?` the top-10 candidates may be potential film topics. Same for the answers to the question `Which religion does Noam Chomsky belong to?` that can all be typed as religions. In these examples, both sides of the relationship are clearly typed: a certain type of entity is expected in head or tail (country, religion, person, movie, etc.). The operators of Tatec may then operate on specific regions of the embedding space. On the contrary, the relationship `/webpage/category` is an example of non-typed relationship. This one, which could actually be seen as an attribute rather than a relationship, indicates if the entity head has a topic website or an official website. Since many types of entities can have a webpage and there is little to no correlated relationships, predicting the left-hand side argument is nearly impossible.

Figures 2a and 2b show 2D projections of embeddings of selected entities for the Trigram and Bigrams models trained on FB15k, respectively, obtained by projecting them using t-SNE (Van der Maaten & Hinton, 2008). This projection has been carried out only for Freebase entities whose profession is either `singer` or `attorney` in the `USA`. We can observe in Figure 2a that all attorneys are clustered and separated from the singers, except one, which corresponds to the multifaceted `Fred Thompson`[7]. However, embeddings of the singers are not clearly clustered: since singers can appear in a multitude of triples, their layout is the result of a compendium of (sometimes heterogeneous) categories. To illustrate graphically the different data patterns to which Bigrams and Trigram respond, we focus on the small cluster made up of Japanese singers that can be seen in Figure 2a (Trigram). In Figure 2b (Bigrams) however, these same entities are more diluted in the whole set of singers. Looking at the neighboring embeddings of these Japanese singers entities in Figure 2b, we find entities highly connected to `japan` like `yoko_ono` – born in Japan, `vic_mignogna`,

[7] Apart from being an attorney, he is an actor, a radio personality, a lawyer and a politician

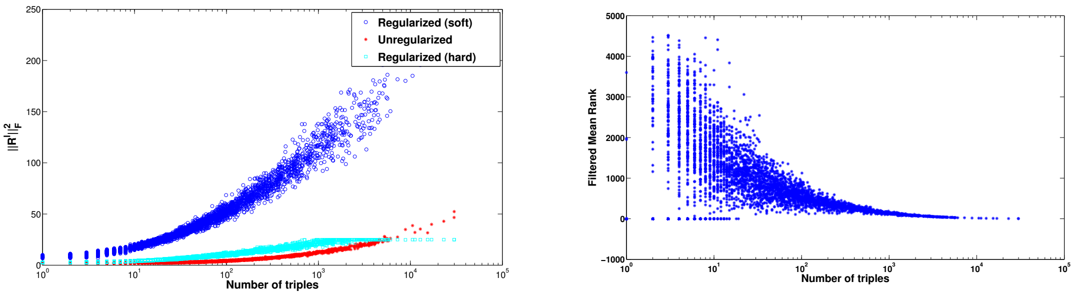
Table 8: **Examples of predictions on SVO.** Given two nouns acting as subject and direct object from a test triple, Tatec predicts the best fitting verb. In bold is the expected correct answer.

Triple	Top-10 predictions
(bus, ? , service)	use, provide, **run**, have, include carry, offer, enter, make, take
(emigrant, ? , country)	flee, become, **enter**, leave, form dominate, establish, make, move, join
(minister, ?, protest)	lead, organize, **join**, involve, make participate, conduct, stag, begin, attend
(vessel, ?, coal)	use, **transport**, carry, convert, send make, provide, supply, sell, contain
(tv_channel, ?, video)	feature, make, release, use, produce have, include, call, base, show
(great_britain, ?, north_america)	include, become, found, establish, dominate name, have, enter, form, run

greg_ayres, chris_patton or laura_bailey – all of them worked in the dubbing industry of Japanese *anime* movies and television series. This shows the impact of the interaction between heads and tails in the Bigrams model: it tends to push together entities connected in triples whatever the relation. In this case, this forms a Japanese cluster.

Table 8 shows examples of predictions on SVO. In the first example, though run is the target verb for the pair (bus, service), other verbs like provide or offer are good matches as well. Similarly, non-target verbs like establish or join, and lead, participate or attend are good matches for the second and third examples ((migrant, country) and (minister, protest)) respectively. The fourth and fifth instances show an example of very heterogeneous performance for a same relationship (the target verb is transport in both cases) which can be easily explained from a semantic point of view: transport is a very good fit given the pair (vessel, coal), whereas a TV channel transports video is not a very natural way to express that one can watch videos in a TV channel, and hence this leads to a very poor performance – the target verb is ranked #696. The sixth example is particularly interesting, since even if the target verb, colonize, is ranked very far in the list (#344), good candidates for the pair (Great Britain, North America) can be found in the top-10. Some of them have a similar representation as colonize, because they are almost synonyms, but they are ranked much higher. This is an effect of the verb frequency.

As illustrated in Figure 3a, the more frequent a relationship is, the higher its Frobenius norm is; hence, verbs with similar meanings but unbalanced frequencies can be ranked differently, which explains that a rare verb, such as colonize, can be ranked much worse than other semantically similar words. A consequence of this relation between the Frobenius norm and the appearance frequency is that usual verbs tend to be highly ranked even though sometimes they are not good matches, due to the influence of the norm in the score. In that figure, we can see that the Frobenius norm of the relation matrices are larger in the regularized (*soft*) case than in the unregularized case. This happens because we fixed a very large value for both C_2 and ρ_l in the regularized case (ρ_e is fixed to 1). It imposes a strong constraint on the norm of the entities but not on the relationship matrices and makes the Frobenius norm of these matrices absorb the whole impact of the norm of the score, and, thus, the impact of the verb frequency. We could down-weight the importance of the verb frequency by tuning the parameters ρ_l and C_2 to enforce a stronger constraint.



(a) Frobenius norm of the rel. matrices according to the number of training triples of each rel.

(b) Test mean rank according to the number of training triples of each relationship.

Figure 3: Indicators of the behavior of Tatec-ft on FB15k according to to the number of training triples of each relationship.

Table 9: **Examples of predictions on SVO for a regularized and an unregularized Trigram.** In bold is the expected correct answer.

Triple	Top-10 predictions	
	Unregularized	Regularized (soft)
(bus, ? , service)	use, operate, offer, call, build, include, have, know, make, create	provide, use, have, include, make, offer, take, carry, serve, **run**
(emigrant, ? , country)	use, represent, save, flee, visit, come, make, leave, create, know	flee, become, come, **enter**, found, include, form, make, leave, join
(minister, ? , protest)	bring, lead, reach, have, become, say, include, help, leave, appoint	lead, organize, conduct, participate, **join** make, involve, support, suppress, raise
(vessel, ? , coal)	take, use, have, carry, make, hold, move, become, fill, serve	use, **transport**, make, carry, deliver, send, contain, supply, leave, provide
(tv_channel, ?, video)	make, include, write, know, have, produce, use, play, give, become	release, make, feature, produce, have, include, use, take, show, base
(great_britain, ?, north_america)	have, use, include, make, leave, become, know, take, call, build	include, found, become, run,name, move, annex, form, establish, dominate

Figure 9 shows the effect of the verb frequency in these two models when predicting the same missing verb as in Table 8.

Breaking down the performance by relationship, this is translated into a strong relation between the performance of a relationship and its frequency (see Figure 3b). However, the same relation between the 2-norm of the entities embeddings and their frequency is not observed, which can be explained given that an entity can appear in the left and right argument in an unbalanced way.

6. Conclusion

This paper presents Tatec, a tensor factorization method that satisfactorily combines 2- and 3-way interaction terms to obtain a performance better than the best of either constituent. Different data patterns are properly encoded thanks to the use of different embedding spaces and of a two-phase training (pre-training and fine-tuning/linear-combination). Experiments on four benchmarks for different tasks and with different quality measures prove the strength and versatility of this model, which could actually be seen as a generalization of a lot of existing works. Our experiments also allow us to draw some conclusions about the two usual regularization schemes used so far in these embedding-based models: they both achieve similar performances, even if soft regularization appears slightly more efficient but with one extra-hyperparameter.

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).

Appendices

Appendix A. Optimal hyperparameters

The optimal configurations for UMLS are:
- TransE-soft: $d = 40, \lambda = 0.01, \gamma = 0.5, C = 0$;
- Bigrams-soft: $d_1 = 40, \lambda_1 = 0.01, \gamma = 0.5, C = 0.1$;
- Trigram-soft: $d_2 = 40, \lambda_2 = 0.01, \gamma = 1, C = 0.1, \rho_l = 5$;
- Tatec-soft: $d_1 = 40, d_2 = 40, \lambda_1 = \lambda_2 = 0.001, \gamma = 1, C_1 = C_2 = 0.01, \rho_l = 5$;
- TransE-hard: $d = 40, \lambda = 0.01, \gamma = 0.1$;
- Bigrams-hard: $d_1 = 40, \lambda_1 = 0.01, \gamma = 0.5$;
- Trigram-hard: $d_2 = 40, \lambda_2 = 0.01, \gamma = 1, \rho_l = 10$;
- Tatec-hard: $d_1 = 40, d_2 = 40, \lambda_1 = \lambda_2 = 0.001, \gamma = 1, \rho_l = 10$.
- Tatec-linear-comb: $d_1 = 40, d_2 = 40, \gamma = 0.5, \alpha = 50$.

The optimal configurations for Kinships are:
- TransE-soft: $d = 40, \lambda = 0.01, \gamma = 1, C = 0$;
- Bigrams-soft: $d_1 = 40, \lambda_1 = 0.01, \gamma = 1, C = 1$;
- Trigram-soft: $d_2 = 40, \lambda_2 = 0.01, \gamma = 0.5, C = 0.1, \rho_l = 5$;
- Tatec-soft: $d_1 = 40, d_2 = 40, \lambda_1 = \lambda_2 = 0.001, \gamma = 1, C_1 = 100, C_2 = 0.0001, \rho_l = 10$;
- TransE-hard: $d = 40, \lambda = 0.01, \gamma = 1$;
- Bigrams-hard: $d_1 = 40, \lambda_1 = 0.01, \gamma = 1$;
- Trigram-hard: $d_2 = 40, \lambda_2 = 0.01, \gamma = 0.5, \rho_l = 10$;
- Tatec-hard $d_1 = 40, d_2 = 40, \lambda_1 = \lambda_2 = 0.001, \gamma = 1, \rho_l = 10$.
- Tatec-linear-comb: $d_1 = 40, d_2 = 40, \gamma = 1, \alpha = 10$.

The optimal configurations for FB15k are:
- TransE-soft: $d = 100, \lambda = 0.01, \gamma = 0.25, C = 0.1$;
- Bigrams-soft: $d_1 = 100, \lambda_1 = 0.01, \gamma = 1, C = 0$;
- Trigram-soft: $d_2 = 50, \lambda_2 = 0.01, \gamma = 0.25, C = 0.001, \rho_l = 1$;
- Tatec-soft: $d_1 = 100, d_2 = 50, \lambda_1 = \lambda_2 = 0.001, \gamma = 0.5, C_1 = C_2 = 0$;
- TransE-hard: $d = 100, \lambda = 0.01, \gamma = 0.25$;
- Bigrams-hard: $d_1 = 100, \lambda_1 = 0.01, \gamma = 0.25$;
- Trigram-hard: $d_2 = 50, \lambda_2 = 0.01, \gamma = 0.25, \rho_l = 5$;

- Tatec-hard: $d_1 = 100, d_2 = 50, \lambda_1 = \lambda_2 = 0.001, \gamma = 0.25, \rho_l = 5$;
- Tatec-no-pret: $d_1 = 100, d_2 = 50, \lambda_1 = \lambda_2 = 0.01, \gamma = 0.25, C_1 = 0, C_2 = 0.001, \rho_l = 1$;
- Tatec-shared: $d_1 = d_2 = 75, \lambda_1 = \lambda_2 = 0.01, \gamma = 0.25, C_1 = C_2 = 0.001, \rho_l = 5$;
- Tatec-linear-comb: $d_1 = 100, d_2 = 50, \gamma = 0.25, \alpha = 200$.

The optimal configurations for SVO are:
- TransE-soft: $d = 50, \lambda = 0.01, \gamma = 0.5, C = 1$;
- Bigrams-soft: $d_1 = 50, \lambda_1 = 0.01, \gamma = 1, C = 0.1$;
- Trigram-soft: $d_2 = 50, \lambda_2 = 0.01, \gamma = 1, , C = 10, \rho_l = 20$;
- Tatec-soft: $d_1 = 50, d_2 = 50, \lambda_1 = \lambda_2 = 0.0001, \gamma = 1, C_1 = 0.1, C_2 = 1, \rho_l = 20$;
- TransE-hard: $d = 50, \lambda = 0.01, \gamma = 0.5$;
- Bigrams-hard: $d_1 = 50, \lambda_1 = 0.01, \gamma = 1$;
- Trigram-hard: $d_2 = 50, \lambda_2 = 0.01, \gamma = 1, \rho_l = 20$;
- Tatec-hard: $d_1 = 50, d_2 = 50, \lambda_1 = \lambda_2 = 0.0001, \gamma = 1, \rho_l = 20$.
- Tatec-linear-comb: $d_1 = 50, d_2 = 50, \gamma = 1, \alpha = 50$.

References

Bordes, A., Glorot, X., Weston, J., & Bengio, Y. (2013a). A semantic matching energy function for learning with multi-relational data. *Machine Learning*, *94*, 233–259.

Bordes, A., Usunier, N., García-Durán, A., Weston, J., & Yakhnenko, O. (2013b). Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, pp. 2787–2795.

Chang, K.-W., Yih, W.-t., Yang, B., & Meek, C. (2014). Typed tensor decomposition of knowledge bases for relation extraction. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1568–1579.

Denham, W. (1973). *The detection of patterns in Alyawarra nonverbal behavior*. Ph.D. thesis, University of Washington.

Dong, X., Gabrilovich, E., Heitz, G., Horn, W., Lao, N., Murphy, K., Strohmann, T., Sun, S., & Zhang, W. (2014). Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 601–610. ACM.

García-Durán, A., Bordes, A., & Usunier, N. (2014). Effective blending of two and three-way interactions for modeling multi-relational data. In *ECML PKDD 2008*. Springer Berlin Heidelberg.

Grandvalet, Y. (1998). Least absolute shrinkage is equivalent to quadratic penalization. In *The International Conference on Artificial Neural Networks*.

Jenatton, R., Le Roux, N., Bordes, A., & Obozinski, G. (2012). A latent factor model for highly multi-relational data. In *NIPS 25*.

Kemp, C., Tenenbaum, J. B., Griffiths, T. L., Yamada, T., & Ueda, N. (2006). Learning systems of concepts with an infinite relational model. In *Proc. of the 21st national conf. on Artif. Intel. (AAAI)*, pp. 381–388.

Knight, K., & Luk, S. K. (1994). Building a large-scale knowledge base for machine translation. In *AAAI*, Vol. 94, pp. 773–778.

Kok, S., & Domingos, P. (2007). Statistical predicate invention. In *Proceedings of the 24th international conference on Machine learning*, ICML '07, pp. 433–440.

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, *42*(8), 30–37.

Lao, N., Mitchell, T., & Cohen, W. W. (2011). 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.

Lin, Y., Liu, Z., Sun, M., Liu, Y., & Zhu, X. (2015). Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of AAAI'15*.

McCray, A. T. (2003). An upper level ontology for the biomedical domain. *Comparative and Functional Genomics*, *4*, 80–88.

Miller, G. (1995). WordNet: a Lexical Database for English. *Communications of the ACM*, *38*(11), 39–41.

Navigli, R., & Velardi, P. (2005). Structural semantic interconnections: a knowledge-based approach to word sense disambiguation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *27*(7), 1075–1086.

Nickel, M., Tresp, V., & Kriegel, H.-P. (2011). 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.

Ponzetto, S. P., & Strube, M. (2006). Exploiting semantic role labeling, wordnet and wikipedia for coreference resolution. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pp. 192–199. Association for Computational Linguistics.

Salakhutdinov, R., & Srebro, N. (2010). Collaborative filtering in a non-uniform world: Learning with the weighted trace norm. *tc (X)*, *10*, 2.

Socher, R., Chen, D., Manning, C. D., & Ng, A. Y. (2013). Reasoning With Neural Tensor Networks For Knowledge Base Completion. In *Advances in Neural Information Processing Systems 26*.

Sutskever, I., Salakhutdinov, R., & Tenenbaum, J. (2009). Modelling relational data using bayesian clustered tensor factorization. In *Adv. in Neur. Inf. Proc. Syst. 22*.

Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, *9*(2579-2605), 85.

Wang, Y. J., & Wong, G. Y. (1987). Stochastic blockmodels for directed graphs. *Journal of the American Statistical Association*, *82*(397).

Wang, Z., Zhang, J., Feng, J., & Chen, Z. (2014a). Knowledge graph and text jointly embedding. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Wang, Z., Zhang, J., Feng, J., & Chen, Z. (2014b). Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pp. 1112–1119.

Yang, B., Yih, W.-t., He, X., Gao, J., & Deng, L. (2014a). Learning multi-relational semantics using neural-embedding models. *CoRR*, *abs/1411.4072*.

Yang, M.-C., Duan, N., Zhou, M., & Rim, H.-C. (2014b). Joint relational embeddings for knowledge-based question answering. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 645–650.

Zhang, J., Salwen, J., Glass, M., & Gliozzo, A. (2014). Word semantic representations using bayesian probabilistic tensor factorization. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.