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

Alberto García-Durán
Sorbonne universités, Université de technologie de Compiègne, CNRS, Heudiasyc UMR 7253
CS 60 319, 60 203 Compiègne cedex, France
alberto.garcia-duran@utc.fr

Antoine Bordes
Facebook AI Research
770 Broadway, New York, NY 10003. USA
abordes@fb.com

Nicolas Usunier*
Facebook AI Research
112 Avenue de Wagram, 75017 Paris, France
usunier@fb.com

Yves Grandvalet
Sorbonne universités, Université de technologie de Compiègne, CNRS, Heudiasyc UMR 7253
CS 60 319, 60 203 Compiègne cedex, France
yves.grandvalet@utc.fr

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 a 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 ×10\(^9\) km) using an internal

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


©2015 AI Access Foundation. All rights reserved.
KB. 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 nowadays being devoted 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 be 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 the 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 addresses.
Figure 1: Example of (incomplete) Knowledge Base with 6 entities, 2 relationships and 7 facts.

as tail but pretty much everything else as head. A link prediction algorithm should be able to adapt to these different settings.

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 \((h, \ell, t)\) 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 \(\ell\). 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 \(\ell\) 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 in RESCAL, i.e. each relationship corresponds to a matrix, whereas in TRANSE they are also represented by vectors. Besides the different interactions across the arguments of a triple to explain its validity, these models also differ in their training objective. 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) use combinations of a 3-way model with a more constrained 2-way model, and in that sense are closer to our algorithm TATEC. There are important differences between these algorithms and TATEC, though. First, both LFM and NTN share the entity embeddings in the 2-way and the 3-way models, while we learn different entity embeddings. The use of different embeddings for the 2-way and the 3-way models does not increase the model expressiveness, because it is equivalent to a combination with shared embeddings in a higher dimensional embedding space, with additional constrains on the relation parameters. As we show in the experiments however, these additional constraints lead to very significant improvements. The second main difference between our approach and LFM is that 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. Lastly, LFM seeks to maximize the likelihood function given a set of positive and negative facts. 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 these works ignore the type-constraints present in the relationships (i.e. some entities are not legitimate arguments of a given relationship), other approaches present extensions making use of this side information. Hence, (Chang et al., 2014) proposes a modification of RESCAL to avoid incompatible entity-relation triples to participate in the loss function and (Krompass, Baier, & Tresp, 2015) proposes a similar framework for the models optimized by iterating through small batches. We did not consider to use that side information since we wanted to keep our approach as general as possible, and applicable on data where such type information is not available (as in the SVO benchmark).

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 out 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.

Symbolic approaches, as the aforementioned that of (Kok & Domingos, 2007), are also worth mentioning in this context. Though their rule-based inference of new links may lead to great expressiveness, they are usually limited by the quality and coverage of their handcrafted rules. Still, the Path Ranking Algorithm (PRA) (Lao et al., 2011) presented a model able to discover rules automatically by performing random walks from training data, with the own limitation of the connectivity between nodes; i.e. if there is no short-enough path connecting two nodes, then the model is not able to infer a relation between them. Recently, (Gardner, Talukdar, Krishnamurthy, & Mitchell, 2014) cover this gap by combining that model with pre-trained embeddings. 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.

Even though we present and evaluate our algorithms in the context of knowledge bases, this work also applies in the broader context of RDF data. RDF is a standard model for data interchange on the Web. It is at the core of the Linked Data initiative (Bizer, Heath, Idehen, & Berners-Lee, 2008) that aims to extend the linking structure of the Web to use URIs to name the relationship between things as well as the two end of the link. This linking structure forms a directed, labeled graph, where the edges represent the named link between two objects. Thus such data is essentially made of triples. In RDF-terminology a triple is defined as (subject, predicate, object). The Freebase dump is available in this format. Figure 2 shows an example of a RDF file of Freebase, where m.02mjm8 is an identifier for the resource representing Barack Obama. This identifier has several predicates as ns:influence.influence_node.influenced_by or ns:people.person.religion, whose objects are ns:m.01d1n (Reinhold Niebuhr) and ns:m.01lp8 (Christianity), respectively.

3. TATEC

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

3.1 Scoring function

The data $S$ is a set of relations between entities in a fixed set of entities in $E = \{e^1, ..., e^E\}$. Relations are represented as triples $(h, \ell, t)$ where the head $h$ and the tail $t$ are indexes of entities (i.e. $h, t \in [E] = \{1, ..., E\}$), and the label $\ell$ is the index of a relationship in $L = \{l^1, ..., l^L\}$, 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 $E \times L \times E$ 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 $\mathbb{R}^{d_1}$, and parameters of operators on $\mathbb{R}^{d_1} \times \mathbb{R}^{d_1}$, most of these operators being associated to a single relationship. More precisely, the score given by Tatec to a triple $(h, \ell, t)$, 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:

$$s_1(h, \ell, t) = \langle r^\ell_1 | e^h_1 \rangle + \langle r^\ell_2 | e^t_1 \rangle + \langle e^h_1 | D | e^t_1 \rangle, \tag{2}$$

where $e^h_1, e^t_1$ are embeddings in $\mathbb{R}^{d_1}$ of the head and tail entities of $(h, \ell, t)$ respectively, $r^\ell_1$ and $r^\ell_2$ are vectors in $\mathbb{R}^{d_1}$ that depend on the relationship $\ell$, and $D$ is a diagonal matrix that does not depend on the input triple.

As a general notation throughout this section, $\langle \cdot | \cdot \rangle$ is the canonical dot product, and $\langle x | A | y \rangle = \langle x | Ay \rangle$ where $x$ and $y$ are two vectors in the same space and $A$ is a square matrix of appropriate dimensions.

We use two different relation vectors for the subject and the object in order to model asymmetric relationships; for instance, if $r^\ell_1 = r^\ell_2$, then (Paris, capital_of, France) would have the same score as (France, capital_of, Paris).

**(T)** Trigram or the 3-way interaction term:

$$s_2(h, \ell, t) = \langle e^h_2 | R^\ell | e^t_2 \rangle, \tag{3}$$

where $R^\ell$ is a matrix of dimensions $(d_1, d_2)$, and $e^h_2$ and $e^t_2$ 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 relationship-dependent vectors $r^\ell_1$ and $r^\ell_2$. It can also be seen as a relaxation of the translation model of (Bordes et al., 2013b), which is the special case where $r^\ell_1 = -r^\ell_2$, $D$ 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 \( r_1^\ell \) and \( r_2^\ell \) were constrained to be in the image of the matrix (\( D = 0 \) 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). Table 1 details the scoring function of some of the aforementioned models.

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).

\[
s_{FT}(h, \ell, t) = \langle r_1^\ell | e_1^h \rangle + \langle r_2^\ell | e_1^t \rangle + \langle e_1^h | D | e_1^t \rangle + \langle e_1^h | R^\ell | e_2^t \rangle
\]

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:

\[
s_{LC}(h, \ell, t) = \delta_1^\ell \langle r_1^\ell | e_1^h \rangle + \delta_2^\ell \langle r_2^\ell | e_1^t \rangle + \delta_3 \langle e_1^h | D | e_1^t \rangle + \delta_4 \langle e_1^h | R^\ell | e_2^t \rangle
\]

The combination weights \( \delta_\ell \) depend on the relationship and are learned by optimizing the ranking loss (defined later in (6)) using L-BFGS, with an additional quadratic penalization term, \( \sum_\ell \sigma_\ell \| \delta_\ell \|^2_2 \), where \( \delta_\ell \) contains the combination weights for relation \( \ell \), and \( \sigma \) are constrained to \( \sum_\ell \sigma_\ell = \alpha \) (\( \alpha \) is a hyperparameter). 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 FIBER BIASES

As a first motivation for having both a 2-way and a 3-way model, we use an analogy with matrix factorization. It is common in matrix factorization techniques for 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)):

$$R_{ui} \approx \langle P_u | Q_i \rangle + \alpha_u + \beta_i + \mu$$  \hspace{1cm} (4)

where $P \in \mathbb{R}^{U \times k}$, with each row $P_u$ containing the $k$-dimensional embedding of the user ($U$ is the number of users), $Q \in \mathbb{R}^{I \times k}$ 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 ($\mu$ 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 $\langle P_u | Q_i \rangle$ of the matrix factorization model for collaborative filtering (4).

The bigram term $(B)$ then plays the role of biases for each fiber of the tensor, i.e.

$$s_1(h, \ell, t) \approx B^1_{l,h} + B^2_{l,t} + B^3_{h,t}$$  \hspace{1cm} (5)

and thus is the analogue for tensors to the term $\alpha_u + \beta_i$ in the matrix factorization model (4). 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 $B^1 = \left[ B^1_{l,h} \right]_{l \in [L], h \in [E]}$, $B^2$ and $B^3$ of Equation (5). 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

---

5. Fibers are the higher order analogue of matrix rows and columns for tensors and are defined by fixing every index but one.
propose in (B) should allow to share relevant information between different biases. Note that whereas a tensor of dimensions $n \times m \times n$ (in general, in this problem the same set of entities is considered for both the head and the tail) has $2mn + n^2$ biases, Tatec computes such biases by means of linear combinations of $n + 2m$ embeddings, which allows a learning transfer across them.

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 (4), 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 assumption that embeddings should have low rank. The user/item-biases in (4) only boil down to adding two rank-1 matrices $\alpha_1^T$ and $\beta_1^T$ 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 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 (4) are trained using a squared error as a measure of approximation and regularized by $\lambda \left( \| P_u \|_2^2 + \| Q_i \|_2^2 + \alpha_u^2 + \beta_i^2 \right)$, where $\lambda > 0$ is the regularization factor. This kind of regularization is a weighted trace norm regularization (Salakhutdinov & Srebro, 2010) on $PQ^T$. Leaving aside the “weighted” part, the idea is that at convergence, the quantity $\lambda \left( \sum_u \| P_u \|_2^2 + \sum_i \| Q_i \|_2^2 \right)$ is equal to $2\lambda$ times the sum of the singular values of the matrix $PQ^T$. However, $\lambda \| \alpha \|_2^2$, which is the regularization applied to user biases, is not $2\lambda$ times the singular value of the rank-one matrix $\alpha 1^T$, which is equal to $\sqrt{7} \| \alpha \|_2$, and can be much larger than $\| \alpha \|_2^2$. 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 $\langle r_1^e | e_i^h \rangle$ and $\langle r_2^e | e_i^h \rangle$ can be added to the trigram term by adding constant features in the entities’ embeddings, and $\langle e_1^h | D | e_i^h \rangle$ 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 $\langle r_1^1 | e_h^1 \rangle + \langle r_2^1 | e_t^1 \rangle$ of 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, $\langle e_h^1 | D | e_t^1 \rangle$, 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 $D$ 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 $D$. 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 $r_1^f$ and $r_2^f$ 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.

---

6. We can see the equivalence by taking the eigenvalue decomposition of a symmetric $D$: apply the change of basis to the embeddings to keep only the diagonal part of $D$ in the term $\langle e_h^1 | D | e_t^1 \rangle$, and apply the reverse transformation to the vectors $r_1^f$ and $r_2^f$. Note that since rotations preserve Euclidean distances, the equivalence still holds under 2-norm regularization of the embeddings.
We use two different learning rates \( \lambda \) of minibatch setting. At each epoch the data set is shuffled and split into disjoint minibatches and thus to a poor 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

<table>
<thead>
<tr>
<th>Model</th>
<th>Score ( (s(h, \ell, t)) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRANSF</td>
<td>( |e_h^s + r^\ell - e_t^r|_2 )</td>
</tr>
<tr>
<td>TRANSH</td>
<td>( |e_h^s - (w_\ell^e e_h^e w_\ell^t) + r^\ell - (e_t^r - (w_\ell^e e_t^r w_\ell^t))|_2^2 )</td>
</tr>
<tr>
<td>TRANSR</td>
<td>( |e_h^s</td>
</tr>
<tr>
<td>RESCAL</td>
<td>( \langle y</td>
</tr>
<tr>
<td>LFM</td>
<td>( \langle y</td>
</tr>
</tbody>
</table>

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 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 \( S \) be the set of positive triples provided by the KB, we optimize the following ranking loss function:

\[
\sum_{(h, \ell, t) \in S} \sum_{(h', \ell', t') \in C(h, \ell, t)} \left[ \gamma - s(h, \ell, t) + s(h', \ell', t') \right]_+ \tag{6}
\]

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

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

The margin \( \gamma \) 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 \( \lambda_1 \) and \( \lambda_2 \), 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 may lead to bad local minima and thus to a poor 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
by constraining the entity embeddings to have, at most, a 2-norm of value \( \rho \). Previous work on embedding models have used two different regularization strategies: either a hard or soft adaptation of the two regularization schemes: in the hard scheme, we force the relation matrices to have, at most, a 2-norm of value \( \rho_r \), and in the soft one, we include a penalization term of the form \( \| R \|_F^2 - \rho_r^2 \) + to the loss function (6). As a result, in the soft scheme the following regularization term is added to the loss function (6): \( C_1 \| e_1 \|_2^2 - \rho_e^2 \) + + \( C_2 \| e_2 \|_2^2 - \rho_e^2 \) + + \( \| R \|_F^2 - \rho_r^2 \) + , where \( C_1 \) and \( C_2 \) are hyperparameters that weight the importance of 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 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 \( \rho_e \) (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 (6). 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 \( \rho_e \). In the latter, which we denote as soft regularization, a penalization term of the form \( \| e \|_2^2 - \rho_e^2 \) + for the entity embeddings \( e \) is added. The soft scheme allows the 2-norm of the embeddings to grow further than \( \rho_e \), 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 \( \rho_r \), and in the soft one, we include a penalization term of the form \( \| R \|_F^2 - \rho_r^2 \) + to the loss function (6). As a result, in the soft scheme the following regularization term is added to the loss function (6): \( C_1 \| e_1 \|_2^2 - \rho_e^2 \) + + \( C_2 \| e_2 \|_2^2 - \rho_e^2 \) + + \( \| R \|_F^2 - \rho_r^2 \) + , where \( C_1 \) and \( C_2 \) are hyperparameters that weight the importance of 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

---

**Algorithm 1** Learning unregularized Tatec.

```plaintext
input Training set \( S = \{(h, l, t)\} \), margin \( \gamma \), learning rates \( \lambda_1 \) and \( \lambda_2 
1: initialization
2: - for Bigrams: \( e_1 \leftarrow \text{uniform}(-\frac{6}{\sqrt{d_1}}, \frac{6}{\sqrt{d_1}}) \) for each entity \( e \)
3: \( r_1, r_2 \leftarrow \text{uniform}(-\frac{6}{\sqrt{d_1}}, \frac{6}{\sqrt{d_1}}) \) for each \( \ell \)
4: \( D \leftarrow \text{uniform}(-\frac{6}{\sqrt{d_1}}, \frac{6}{\sqrt{d_1}}) \)
5: - for Trigram: \( e_2 \leftarrow \text{uniform}(-\frac{6}{\sqrt{d_2}}, \frac{6}{\sqrt{d_2}}) \) for each entity \( e \)
6: \( R \leftarrow \text{uniform}(-\frac{6}{\sqrt{d_2}}, \frac{6}{\sqrt{d_2}}) \) for each \( \ell \)
7: - for Tatec-ft: pre-trained weights of Bigrams and Trigram
8: All the embeddings are normalized to have a 2- or Frobenius-norm equal to 1.
9: loop
10: \( S_{\text{batch}} \leftarrow \text{sample}(S, m) \) // sample a training minibatch of size \( m \)
11: \( T_{\text{batch}} \leftarrow \emptyset \) // initialize a set of pairs of examples
12: for \((h, l, t) \in S_{\text{batch}} \) do
13: \((h', l', t') \leftarrow \text{sample a negative triple according to the selected strategy} \ C(h, l, t) \)
14: \( T_{\text{batch}} \leftarrow T_{\text{batch}} \cup \{(h, l, t), (h', l', t')\} \) // record the pairs of examples
15: end for
16: Update parameters using gradients \( \sum_{(h, l, t), (h', l', t') \in T_{\text{batch}}} [\gamma - s(h, l, t) + s(h', l', t')]_+ \):
17: - for Bigrams (Eq. 2): \( s = s_1 \)
18: - for Trigram (Eq. 3): \( s = s_2 \)
19: - for Tatec-ft (Eq. 1): \( s = s_1 + s_2 \)
20: end loop
```
Table 2: **Statistics of the data sets** used in this paper and extracted from four knowledge bases: FB15k, SVO, Kinships and UMLS.

<table>
<thead>
<tr>
<th>Data set</th>
<th>FB15k</th>
<th>SVO</th>
<th>Kinships</th>
<th>UMLS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entities</td>
<td>14,951</td>
<td>30,605</td>
<td>104</td>
<td>135</td>
</tr>
<tr>
<td>Relationships</td>
<td>1,345</td>
<td>4,547</td>
<td>26</td>
<td>49</td>
</tr>
<tr>
<td>Training examples</td>
<td>483,142</td>
<td>1,000,000</td>
<td>224,973</td>
<td>102,612</td>
</tr>
<tr>
<td>Validation examples</td>
<td>50,000</td>
<td>50,000</td>
<td>28,122</td>
<td>89,302</td>
</tr>
<tr>
<td>Test examples</td>
<td>59,071</td>
<td>250,000</td>
<td>28,121</td>
<td>89,302</td>
</tr>
</tbody>
</table>

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:

$$\min \sum_{(h,\ell,t)\in S} \sum_{(h',\ell',t')\in C(h,\ell,t)} \left[ \gamma - s(h,\ell,t) + s(h',\ell',t') \right]_+$$

s.t. $||e^i_1||_2 \leq \rho_e \forall i \in [E]$  

$||e^i_2||_2 \leq \rho_e \forall i \in [E]$  

$||R^\ell||_F \leq \rho_l \forall \ell \in [L]$  

And in the soft regularization case it is:

$$\min \sum_{(h,\ell,t)\in S} \sum_{(h',\ell',t')\in C(h,\ell,t)} \left[ \gamma - s(h,\ell,t) + s(h',\ell',t') \right]_+ + C_1 \sum_{i\in[E]} (||e^i_1||^2_2 - \rho_e^2)_+$$

$$+ C_2 \left( \sum_{i\in[E]} (||e^i_2||^2_2 - \rho_e^2)_+ + \sum_{\ell\in[L]} (||R^\ell||^2_2 - \rho_l^2)_+ \right)$$

where $s(h,\ell,t) = \langle r^i_1 | e^i_1 \rangle + \langle r^i_2 | e^i_1 \rangle + \langle e^i_1 | D | e^i_1 \rangle + \langle e^i_2 | R^\ell | e^i_2 \rangle$ in both cases.

5. **Experiments**

This section presents various experiments to illustrate how competitive Tatec is 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.
Embeddings Models for Link Prediction in KBs

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. 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

---

7. This replication process is carried out only in training.
8. Results on both FB15K and SVO with TransE and Tatec were already provided in (García-Durán et al., 2014), however in these works the hyperparameters were validated on a smaller validation set and a not wide enough grid search, which led to suboptimal results. We hence decided to re-run them and got major improvements.
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 $\rho_e$ determining the value from which the $L_2$-norm of the entity embeddings are penalized has been fixed to 1, but the radius $\rho_l$ of the Trigram model has 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 $\delta$ parameters via L-BFGS, and update of $\sigma$ parameters using $\sigma^*_t = \frac{\alpha ||\delta^t||_2}{\sum_k ||\delta^t||_2}$, until some stopping criterion is reached. The $\delta$ parameters are initialized to 1 and the $\alpha$ value is validated among $\{0.1, 1, 10, 50, 100, 200, 500, 1000\}$.

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.
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.

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

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.

**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
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.

<table>
<thead>
<tr>
<th>Model</th>
<th>FB15k</th>
<th>SVO</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean Rank</td>
<td>Hits@10</td>
</tr>
<tr>
<td>Counts</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>SME(linear)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>LFM</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>TransH</td>
<td>87</td>
<td>64.4</td>
</tr>
<tr>
<td>TransR</td>
<td>77</td>
<td>68.7</td>
</tr>
<tr>
<td>cTransR</td>
<td>75</td>
<td>70.2</td>
</tr>
<tr>
<td>TransE-soft</td>
<td>50.7 ± 2.0</td>
<td>71.5 ± 0.3</td>
</tr>
<tr>
<td>TransE-hard</td>
<td>50.6 ± 2.0</td>
<td>71.5 ± 0.3</td>
</tr>
<tr>
<td>Tatec-no-pretrain</td>
<td>97.1 ± 3.9</td>
<td>65.7 ± 0.2</td>
</tr>
<tr>
<td>Tatec-shared</td>
<td>94.8 ± 3.2</td>
<td>63.4 ± 0.3</td>
</tr>
<tr>
<td>Bigrams-hard</td>
<td>94.5 ± 2.9</td>
<td>67.5 ± 0.4</td>
</tr>
<tr>
<td>Trigram-hard</td>
<td>137.7 ± 7.1</td>
<td>56.1 ± 0.4</td>
</tr>
<tr>
<td>Tatec-ft-hard</td>
<td>59.8 ± 2.6</td>
<td>77.3 ± 0.3</td>
</tr>
<tr>
<td>Bigrams-soft</td>
<td>87.7 ± 4.1</td>
<td>70.0 ± 0.2</td>
</tr>
<tr>
<td>Trigram-soft</td>
<td>121.0 ± 7.2</td>
<td>58.0 ± 0.3</td>
</tr>
<tr>
<td>Tatec-ft-soft</td>
<td>57.8 ± 2.3</td>
<td>76.7 ± 0.3</td>
</tr>
<tr>
<td>Tatec-lc-soft</td>
<td>68.5 ± 3.2</td>
<td>72.8 ± 0.2</td>
</tr>
</tbody>
</table>

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 $D$, 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 $P(\text{head}|\text{tail})$ and $P(\text{tail}|\text{head})$ 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
Table 5: **Test results on FB15k.** Proportion of entities ranked in the Top 1.

<table>
<thead>
<tr>
<th>Model</th>
<th>Hits@1</th>
</tr>
</thead>
<tbody>
<tr>
<td>TransE-soft</td>
<td>28.1</td>
</tr>
<tr>
<td>Bigrams-soft</td>
<td>27.2</td>
</tr>
<tr>
<td>Trigram-soft</td>
<td>24.9</td>
</tr>
<tr>
<td>Tatec-ft-soft</td>
<td><strong>37.8</strong></td>
</tr>
</tbody>
</table>

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 (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 there better than simply using a linear combination even if Tatec-lc still performs 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 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. Table 5 displays the Hits@1 for several of these models. Whereas the differences in the performance of Bigrams and Trigram are not so large as the ones shown in hits@10, Tatec is still the best model by a wide margin.

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 6. 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.
Table 6: **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**).

<table>
<thead>
<tr>
<th>Task</th>
<th>Predicting head</th>
<th>Predicting tail</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1-to-1</td>
<td>1-to-M.</td>
</tr>
<tr>
<td><strong>TransE-soft</strong></td>
<td>76.2</td>
<td>93.6</td>
</tr>
<tr>
<td>TransH</td>
<td>66.8</td>
<td>87.6</td>
</tr>
<tr>
<td>TransR</td>
<td>78.8</td>
<td>89.2</td>
</tr>
<tr>
<td>cTransR</td>
<td>81.5</td>
<td>89</td>
</tr>
<tr>
<td><strong>Bigrams-soft</strong></td>
<td>76.2</td>
<td>90.3</td>
</tr>
<tr>
<td>Trigram-soft</td>
<td>56.4</td>
<td>79.6</td>
</tr>
<tr>
<td>Tatec-ft-soft</td>
<td>79.3</td>
<td>93.2</td>
</tr>
</tbody>
</table>

### 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. Table 7 depicts training times of various models on FB15k, presenting the relative time w.r.t. to TransE for one training epoch. To speedup training, we could follow one or several of the following strategies:

- use adaptive learning rates in order to make convergence faster;
- train the model on GPUs, which is quite usual in the deep learning community when working with large datasets;
- parallelize training with Hogwild (Recht, Re, Wright, & Niu, 2011).

We could also speed up the validation. For example, since the entities and relationships are usually strongly typed (i.e. given a relationship, only a subset of entities are real candidates for both the subject and object), we might consider only entities of the suitable type for a given relationship and role. Nevertheless, given that scalability is not a major issue on the datasets used in this paper we did not look for any speed optimization here.

### 5.3 Illustrative experiments

This last experimental section provides some illustrations and insights on the performance of Tatec and TransE.
Table 7: Relative training times with respect to TransE on FB15k for running one epoch on a single core.

<table>
<thead>
<tr>
<th>Model</th>
<th>Relative train. time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bigrams-soft</td>
<td>× 1.4</td>
</tr>
<tr>
<td>Trigram-soft</td>
<td>× 3.6</td>
</tr>
<tr>
<td>Tatec-ft-soft</td>
<td>× 4.0</td>
</tr>
</tbody>
</table>

Table 8: 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.

<table>
<thead>
<tr>
<th>Triple</th>
<th>Top-10 predictions</th>
</tr>
</thead>
<tbody>
<tr>
<td>(poland_national_football_team, /sports_team/location, ?)</td>
<td>Mexico, South_Africa, Republic_of_Poland, Belgium, Puerto_Rico, Austria, Georgia, Uruguay, Colombia, Hong_Kong</td>
</tr>
<tr>
<td>(?, /file/file_subject/films, remember_the_titans)</td>
<td>racism, vietnam_war, aviation, capital_punishment, television, filmmaking, christmas, female, english_language, korean_war</td>
</tr>
<tr>
<td>(noam_chomsky, /people/person/religion, ?)</td>
<td>atheism, agnosticism, catholicism, ashenkani_jews, buddhism, islam, protestantism, baptist, episcopal_church, hinduism</td>
</tr>
<tr>
<td>(?, /webpage/category, official_website)</td>
<td>supreme_court_of_canada, butch_hartman, robm_hitchcoc, mercer_university, clancy_brown, dana_gelany, hornets, grambling_state_university, dniprospetrovsk, juanes</td>
</tr>
</tbody>
</table>

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_subject/films 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 a symmetrical triple in the training set. If we split the test triples into two subsets, one containing the test triples for which a symmetrical triple has been used in the learning stage and the other containing those ones for which a symmetrical triple does not exist 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.
5.3.2 Anecdotal examples

Some examples of predictions by TATEC on FB15k are displayed in Table 8. 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 correlation among relationships, predicting the left-hand side argument is nearly impossible.

Figures 4a and 4b 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 4a that all attorneys are clustered and separated from the singers, except one, which corresponds to the multifaceted Fred Thompson. 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 4a (Trigram). In Figure 4b (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 4b, we find entities highly connected to Japan like yoko_ono – born in Japan, vic_mignogna,

---

9. Apart from being an attorney, he is an actor, a radio personality, a lawyer and a politician.
Table 9: **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.

<table>
<thead>
<tr>
<th>TRIPLE</th>
<th>TOP-10 PREDICTIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>(bus, ? , service)</td>
<td>use, provide, run, have, include carry, offer, enter, make, take</td>
</tr>
<tr>
<td>(emigrant, ? , country)</td>
<td>flee, become, enter, leave, form dominate, establish, make, move, join</td>
</tr>
<tr>
<td>(minister, ?, protest)</td>
<td>lead, organize, join, involve, make participate, conduct, stag, begin, attend</td>
</tr>
<tr>
<td>(vessel, ?, coal)</td>
<td>use, transport, carry, convert, send make, provide, supply, sell, contain</td>
</tr>
<tr>
<td>(tv_channel, ?, video)</td>
<td>feature, make, release, use, produce have, include, call, base, show</td>
</tr>
<tr>
<td>(great_britain, ?, north_america)</td>
<td>include, become, found, establish, dominate name, have, enter, form, run</td>
</tr>
</tbody>
</table>

**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 9 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 ((emigrant, 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 5a, 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 \( \rho_l \) in the regularized case (\( \rho_e \) 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 \( \rho_l \) and \( C_2 \) to enforce a stronger constraint.
Figure 5: Indicators of the behavior of Tatec-ft on FB15k according to to the number of training triples of each relationship.

Table 10: Examples of predictions on SVO for a regularized and an unregularized Trigram. In bold is the expected correct answer.

<table>
<thead>
<tr>
<th>TRIPLE</th>
<th>UNREGULARIZED</th>
<th>REGULARIZED (SOFT)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(bus, ? , service)</td>
<td>use, operate, offer, call, build, include, have, know, make, create</td>
<td>provide, use, have, include, make, offer, take, carry, serve, run</td>
</tr>
<tr>
<td>(emigrant, ? , country)</td>
<td>use, represent, save, flee, visit, come, make, leave, create, know</td>
<td>flee, become, come, enter, found, include, form, make, leave, join</td>
</tr>
<tr>
<td>(minister, ? , protest)</td>
<td>bring, lead, reach, have, become, say, include, help, leave, appoint</td>
<td>lead, organize, conduct, participate, join, make, involve, support, suppress, raise</td>
</tr>
<tr>
<td>(vessel, ? , coal)</td>
<td>take, use, have, carry, make, hold, move, become, fill, serve</td>
<td>use, transport, make, carry, deliver, send, contain, supply, leave, provide</td>
</tr>
<tr>
<td>(tv_channel, ?, video)</td>
<td>make, include, write, know, have, produce, use, play, give, become</td>
<td>release, make, feature, produce, have, include, use, take, show, base</td>
</tr>
<tr>
<td>(great_britain, ?, north_america)</td>
<td>have, use, include, make, leave, become, know, take, call, build</td>
<td>include, found, become, run, name, move, annex, form, establish, dominate</td>
</tr>
</tbody>
</table>

Figure 10 shows the effect of the verb frequency in these two models when predicting the same missing verb as in Table 9.

Breaking down the performance by relationship, this is translated into a strong relation between the performance of a relationship and its frequency (see Figure 5b). 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, whose scoring function, as we argue in Section 3.1, is tightly connected to other energy-based model of the literature such as TransE,
RESCAL or LFM. 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.

Our work uses FB15k as its main testbed but our model is not tied to any KB schema; as long as data is formatted as triples, Tatec can be applied. Hence, it could be directly applied to many Linked Data settings especially if there are in RDF format.

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: $d = 50, d_2 = 50, \lambda_1 = \lambda_2 = 0.001, \gamma = 1$;
- 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


