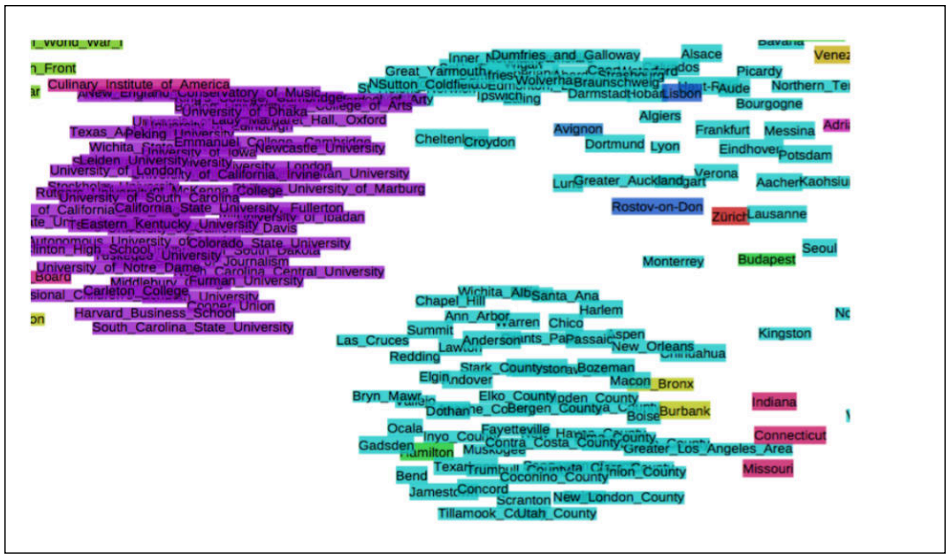
Learning representations in multi-relational graphs : algorithms and applications

Alberto García Durán

Par **Alberto GARCÍA DURÁN**

Learning representations in multi-relational graphs: algorithms and applications

Thèse présentée
pour l'obtention du grade
de Docteur de l'UTC



Soutenue le 6 avril 2016
Spécialité : Technologies de l'Information et des Systèmes

D2271

Learning representations in multi-relational graphs: algorithms and applications

Alberto García Durán Université de Technologie de Compiègne

France

Soutenue le 6 avril 2016 devant le jury composé de :

M. D. LENNE (Président)
M. A. BORDÈS (Directeur de thèse)
M. Y. GRANDVALET (Directeur de thèse)
M. G. BOUCHARD (Rapporteur)
M. L. DENOYER (Rapporteur)
M. N. USUNIER
M. G. VAROQUAUX

Acknowledgements

These ones are actually the last words I write, and the ones that put an end to this very nice stage. I am really grateful for these years. Thanks Antoine for your guidance and your personal treatment, you have been an excellent main supervisor. Thanks Nicolas for your always wise comments. Thanks Yves for all your help.

I would not like to forget those great people who I met along these years: Luis, Juan, Lola and my nice colleagues of Lisa Lab, Sungjin and Caglar. Shameem, I will miss a lot all the coffees and talks we have shared.

A mis padres, por haberme inculcado que el esfuerzo es todo. A mi hermanito, por todas las risas y bromas que hemos y seguiremos compartiendo.

Gracias Laura por todo el tiempo y la paciencia que me has dedicado en especial estos 3 años. Ahora nos espera una nueva aventura.

Forza, saluti a tutti, bacioni, auguri, in bocca al lupo, arrivederci e a presto pino !

Juan de Pablos

Contents

List of Figures

List of Tables

Part I.

Context and State-of-the-art

This manuscript is a paper-based thesis. It is divided in three parts. In Part I, we provide a description and introduction of the problem this thesis focuses on. The state of the art is also reviewed.

Part II encompasses the papers of this thesis. Firstly, generic information (task evaluation, databases...) that is used along these papers is introduced. Then, each chapter of this part will correspond to an adapted and extended version of a single paper along with its corresponding final discussion on related (previous and posterior) works.

Lastly, the Part III will summarize the main points of this thesis and future lines that are worth being investigated.

1. Context

Huge amounts of structured and relational data are available in many domains of engineering, industry or research ranging from the Semantic Web, or bioinformatics to recommender systems. As a result, Knowledge Bases, such as Freebase, WordNet or GeneOntology, became essential tools for storing, manipulating and accessing information, but they are also incomplete, imprecise and far too large to be used as efficiently and broadly as they could. Hence, there is need for methods able to summarize, complete or merge these large databases. This is our main motivation. Knowledge Bases can be represented as 3-dimensional tensors, and we will rely on tensor factorization methods to learn compact representations. The overall objective of the EVEREST project and this thesis is to bring a leap forward in factorization of large sparse tensors in order to improve the accessibility, completeness and reliability of real-world Knowledge Bases. This line of research could have a huge impact in industry (Semantic Web, biomedical applications, etc.). For that reason, Xerox Research Center Europe has supported this project and provided expertise and ease industrial transfer. This proposal is also consistent with the long-term research direction of its principal partner, Heudiasyc, since it contributes in several aspects of the 10-years LabEx program on Technological Systems of Systems started in 2011.

Figure 1.1.: EVEREST project logo

EVEREST has been presented at: Google NY (Feb'13), ICLR (Apr'13), SMAI congress (May'13), ICML (Jun'13), PFIA (Jul'13), UW MSR Summer Institute (Jul'13), Workshop on Computational Models of Early Language Acquisition and Zero Resource Speech Technologies (Jul'13), EMNLP (Oct'13), University of Edinburg (Nov'13), Criteo (Nov'13), NIPS (Dec'13), Google MTV (Dec'13), Facebook (Dec'13), GdR CNRS ISIS (Feb'14).

This thesis has been conducted within the context of the ANR-funded EVEREST project.

To see more details of the project, please visit https://everest.hds.utc.fr/doku.php.

2. State-of-the-art

2.1. Introduction

Internet provides a huge amount of information at hand in such a variety of topics, that now everyone is able to access to any kind of knowledge. Such a big quantity of information could bring a leap forward in many areas if used properly. This way, a crucial challenge of the Artificial Intelligence community has been to gather, organize and make intelligent use of this growing amount of available knowledge.

Fortunately, important efforts have been made in gathering and organizing knowledge for some time now, and a lot of structured information can be found in repositories called Knowledge Bases (KBs)[1], which we can be browsed on-line. For now, a key task is left: to take advantage of them in a intelligent and efficient way.

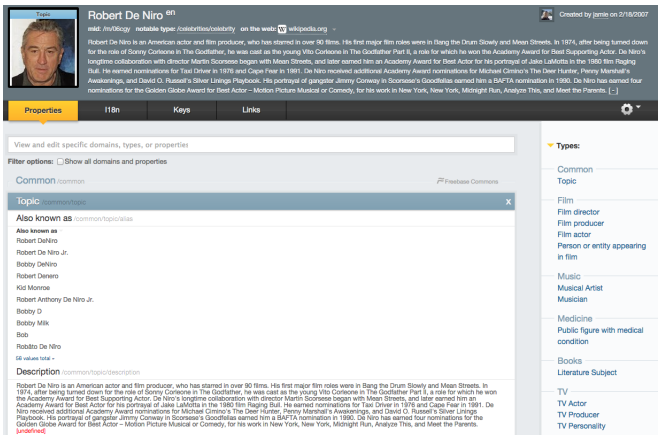


Figure 2.1.: Example of Freebase topic

These repositories can cover any kind of area, from specific domains like biological processes (e.g. in GeneOntology[2]) or lexical information (e.g. WordNet[3]), to very generic purposes. Freebase[4] (see Figure 2.1), 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[5], an engine which answers to

[1] Not all the KBs contain exclusively structured information
[2] http://geneontology.org
[3] http://wiki.dbpedia.org/
[4] http://www.freebase.com
[5] http://www.wolframalpha.com

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. Such KBs can be used for question answering, but also for other natural language processing tasks like word-sense disambiguation (Navigli and Velardi 2005), co-reference resolution (Ponzetto and Strube 2006) or even machine translation (Knight and 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*. Despite its simplicity, most of the core information of the spoken and written language can be represented via one or several triples. 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. Other than that, biological interactions representing the effects that organisms in a community have on one another are usually represented as multi-relational graphs, whose nodes corresponding to species, are connected by links that measure the strength of the interaction between these two species.

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 hundreds of millions 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. By finding new links between entities we are not only completing the KB with known information but we also may be discovering unknown relations between elements of the repository. For example, we may discover interactions between genes in GeneOntology or potential friendships among members of the social network Facebook[6].

2.1.1. Knowledge Bases

The structured information contained in the KBs is presented as a set of triples or facts (*head, label, tail*) that define a multi-relational graph as the one shown in Figure 2.2. Thus, this type of structure can cover any knowledge involving two entities connected by a label. Note that a triple actually defines a piece of 1-hop information of such graph.

In a small KB, such as in Figure 2.2, made up of 6 entities and 2 different labels,

[6] https://www.facebook.com/

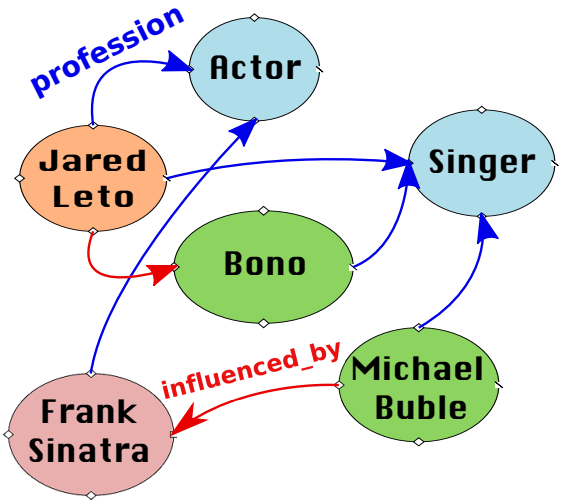


Figure 2.2.: **Example of (incomplete) Knowledge Base** with 6 entities, 2 labels and 7 facts

we can extract facts like (`Jared Leto, influenced_by, Bono`) or (`Michael Buble, profession, singer`). By performing link prediction in such graph, we could obtain new facts such as (`Frank Sinatra, profession, singer`), predicted by using the fact that he influenced the singer `Michael Buble`. This implies to perform some kind of logic over the graph. In this specific case, the reasoning to obtain (`Frank Sinatra, profession, singer`) is based on the transitive property: if a is related to an element b, and b is in turn related to an element c, then a is also related to c. Because of the nature of the problem, inductive logic programming, a field that tries to infer logic rules from the data to explain them, is an appealing approach to perform link prediction. In this thesis, we have used the naming *symbolic approaches* (Section 2.2.5) to encompass this field and other related works that try explain the data from rules.

We now present in more detail two well-known databases of the literature.

Freebase

FREEBASE (Bollacker et al. 2008) is a collaboratively created graph database for structuring human knowledge. Topics are represented by machine IDs (mid) (which play a critical role when topics are merged or split because of their uniqueness) that can be used in URLs. For example, the mid **/m/01vrncs** representing the topic Bob Dylan is accessible by the corresponding FREEBASE entry `http://www.freebase.com/m/01vrncs`.

Unlike other KBs, labels in FREEBASE provide information about both the category that relationship can be framed in and the expected categories of *head* and *tail*. For example, given (`/m/01smm, /travel/travel_destination/tourist_attractions, /m/0328cp`) we know that:

- the relation is categorized in the category *travel*,
- the head (`/m/01smm` corresponds to the topic *Columbus*) is a travel destination,

```
ns:m.02mjmr
    ns:architecture.building_occupant.buildings_occupied    ns:m.0dfzl4x:
    ns:award.award_nominee.award_nominations    ns:m.0k0xq8d:
    ns:common.topic.alias    "Barack Hussein Obama, Jr."@en;
    ns:film.person_or_entity_appearing_in_film.films    ns:m.0nczv2n:
    ns:influence.influence_node.influenced_by    ns:m.01d1nj;
    ns:people.person.religion    ns:m.01lp8:
```

Figure 2.3.: Example of RDF file of FREEBASE

- and the tail (`/m/0328cp` corresponds to the topic *Nationwide Arena*) is a tourist attraction.

Accordingly the human-readable string for the topic identified by its *mid* can be obtained from the fact (`mid`, `/type/object/name`,`any interpretable string`).

Wordnet

WORDNET (Fellbaum 2005) is a large lexical database of English. Nouns, verbs, adjectives... are grouped into sets of distinct concepts, called synsets. Examples are $\{car, automobile\}$, $\{hit, strike\}$ and $\{big, large\}$. These synsets are interlinked by means of conceptual-semantic and lexical relations (e.g. synonymy, hyponymy...). In turn, the same word may appear in several different synsets, reflecting polysemy or multiplicity of meaning. Consequently Wordnet has been proved to be useful for a range of Natural Language Processing (NLP) tasks that involve the challenge of word sense identification (Li et al. 1995, Nastase and Szpakowicz 2001).

Though we present and evaluate our algorithms in the context of knowledge bases, it 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[7] (Bizer et al. 2009) that aims to extend the linking structure of the Web to use URIs to name the relationship between things as well as the two ends 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 2.3 shows an example of a RDF file of FREEBASE, where `m.02mjmr` 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.

[7] http://linkeddata.org

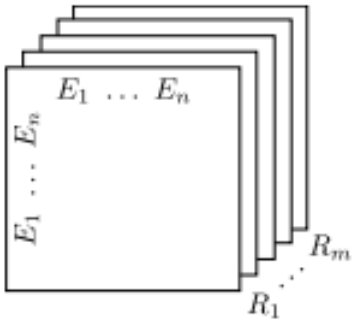


Figure 2.4.: Graphical view of a 3-mode tensor

2.2. Modeling multi-relational data

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

These relational data can be expressed as a tensor of dimensions $n \times n \times m$, where n and m are the number of entities and labels in the multi-relational graph, respectively. This is a collection of slices stacked as the one illustrated in Figure 2.4. The slice k corresponds to the binary adjacency of the graph defined by such label. This pile of binary adjacency matrices is a typical way of representing graphs, where the entry x_{ijk} points out whether the $k-th$ relation between the $i-$ and $j-th$ entities (nodes) holds. Formally:

$$x_{ijk} = \begin{cases} 1, & \text{if } \text{Rel}_k(\text{Entity}_i, \text{Entity}_j) \text{ holds} \\ 0, & \text{if } \text{Rel}_k(\text{Entity}_i, \text{Entity}_j) \text{ does not hold.} \end{cases} \tag{2.1}$$

Note that a 3-mode tensor representation is not exclusive of relational data, but any kind of data involving interactions between two sets along a third mode can be represented in such a way. For example, the slices of a tensor can contain the transactions between two sets of companies along time.

2.2.1. Classic tensor factorization methods

The most referenced and starting point of the current works on tensor decomposition are CANDECOMP/PARAFAC (CP) (Mocks 1988) and Tucker Decomposition (TD) (Tucker 1966). CP factorizes a tensor $X \in \mathbb{R}^{I \times J \times K}$ into a sum of R rank-one tensors $\mathbf{a}_r$,

$\mathbf{b}_r$ and $\mathbf{c}_r$:

$$x_{ijk} = \sum_{r=1}^{R} a_{ir} b_{jr} c_{kr} \tag{2.2}$$

This number R of rank-one tensors that are needed for an exact decomposition of a tensor is the tensor rank and, unfortunately there is no straightforward algorithm to determine the rank of a specific given tensor. As opposed to Singular Value Decomposition (SVD) (Golub and Reinsch 1970), here summing k of the rank-one tensors would not yield the best rank-k approximation, so this implies that the components of the best rank-k model may not be solved sequentially. Therefore, the way to do a CP decomposition is by doing multiple CP decompositions with different number of rank-one tensors until one to be "good" (Kolda and Bader 2009). Once the number of components is fixed alternating least squares (ALS) based methods (Carroll and Chang 1970, Harshman 1970, Navasca et al. 2008, Nion and De Lathauwer 2008, Rajih et al. 2008) happen to be the most usual way to solve it, though it can take many iterations to converge and it is not guaranteed to converge to a global minimum.

TD decomposes a tensor into a core tensor $G \in \mathbb{R}^{P \times Q \times R}$ multiplied by a factor matrix, $A \in \mathbb{R}^{I \times P}$, $B \in \mathbb{R}^{J \times Q}$ and $C \in \mathbb{R}^{K \times R}$ (which are usually constrained to be orthogonal), along each mode:

$$x_{ijk} = \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{r=1}^{R} g_{pqr} a_{ip} b_{jq} c_{kr}. \tag{2.3}$$

In fact, CP can be viewed as a special case of TD. As these factor matrices are usually orthogonal, this decomposition uses to be considered as a high-order PCA (Jolliffe 2002).

Another well-known tensor decomposition method is DEDICOM (Harshman 1978). It decomposes the slices X_ks of a 3-mode tensor $X \in \mathbb{R}^{I \times I \times K}$ as follows:

$$X_k \approx A D_k R D_k A^T \tag{2.4}$$

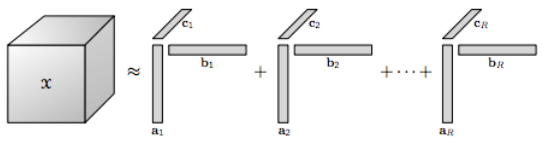
where $A \in \mathbb{R}^{I \times R}$, $R \in \mathbb{R}^{I \times R}$, D_ks $\in \mathbb{R}^{R \times R}$ are diagonal matrices and the entry $(D_k)_{rr}$ indicates the participation of the r-th latent component at slice k.

As CP, methods for computing TD and DEDICOM are mostly ALS based (De Lathauwer et al. 2000, Kroonenberg and De Leeuw 1980) and (Bader et al. 2007, Kiers 1993), respectively. Figure 2.5 provides a visual interpretation of these 3 methods.

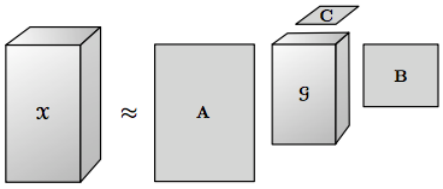
An extensive survey on both non-negative matrix and tensor factorization can be found in (Cichocki and Amari 2002).

2.2.2. RESCAL based models

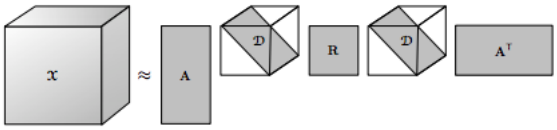
Nickel et al. (2011) proposed a tensor factorization technique, named RESCAL, based on a relaxed version of DEDICOM with a special focus for relational learning. The rank-reduced reconstructed tensor by RESCAL tries to capture the underlying structure of the



(a) CANDECOMP/PARAFAC



(b) Tucker decomposition



(c) DEDICOM

Figure 2.5.: Tensor factorization methods

relational data by detecting connections between different entities and relations. This method approximates each slice X_k by two matrices, one global and one relation-specific matrix, as follows:

$$X_k \approx AR_kA^T, \tag{2.5}$$

and unlike DEDICOM, RESCAL does not set any constraint on the structure of either of them.

The entities are represented by the global common matrix $A \in \mathbb{R}^{n \times d}$ regardless of the relation, which is actually the latent-component representation of the entities. The relation matrices ($R_k \in \mathbb{R}^{d \times d}$) model the interactions of the latent components in the k-th predicate. d is the rank of the predicted tensor, which is an hyperparameter of the model. The relation matrices and latent representations of the entities capture simultaneously not only the interactions and intra-actions between the relations and entities, but also information regarding the different role (head or tail) of an entity. Nickel et al. (2014) upper bounds the rank required to recover adjacency tensors, which does not only increase the predictive performance but also reduces meaningfully the required runtime.

While this tensor product is able to capture rich interactions, a main problem lies in the large number of parameters of the relation matrices, which (as we will see later in Chapter 5) can be problematic in terms of overfitting and computational demands. Yang et al. (2014a) proposed to use diagonal relation matrices to reduce the number of parameters, at the cost of not being able to model asymmetric relations (i.e. $\tilde{x}_{ijk} = \tilde{x}_{kji}$ $\forall i, k$).

As RESCAL is free of constraints in its parameters, its computation is much simpler, being able to be solved directly with any nonlinear optimization algorithm. Nickel et al. (2011) use a modification of an ALS approach, named ASALSAN (Bader et al. 2007), to minimize the least-squares error between the predicted and real tensors. The corresponding probabilistic interpretation is that the random variation of the

data follows a Gaussian distribution. Given the binary nature of the tensor of these multi-relational data, Nickel and Tresp (2013) frame RESCAL as a logistic regression problem rather than a least-squares one, which assumes the data come from a Bernoulli distribution.

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 (as shown in Section 2.1.1, some KBs as Freebase provide information about the expected category of the left and right arguments. KBs in RDF format also offer the predicates `rdfs:domain` and `rdfs:range` to specify the expected type of entities in the head and tail, respectively). For example, given the label *born_in* only the set of person and location entities are compatible entity pairs for the left and right argument, respectively. Ignoring these type-constraints in RESCAL, for example, makes the whole embedding matrix of entities (contained in the global matrix A) has to be updated when learning every single slice of the tensor. Chang et al. (2014) propose a modification of RESCAL to avoid incompatible entity-relation pairs to participate in the loss function by selecting sub-matrices of each slice of the tensor during training, leading to a considerable improvement in terms of convergence time, and also better accuracy. They approximate the adjacency matrix of the relation k X_k by:

$$\tilde{X}_k \approx A_{[\text{domain}_k,:]} R_k A^T_{[\text{range}_k,:]} \tag{2.6}$$

where $\tilde{X}_k \in \mathbb{R}^{n_k \times m_k}$ is the subgraph defined by relation-type k, and $A_{[\text{domain}_k,:]} \in \mathbb{R}^{n_k \times d}$, $A_{[\text{range}_k,:]} \in \mathbb{R}^{m_k \times d}$ are the indices that agree with the domain and range constraints of relation-type k, where n_k and m_k are the the number of indices for each, respectively.

Krompaß et al. (2015) propose a similar framework for the model optimized by iterating through mini-batch Stochastic Gradient Descent (SGD), instead of by an ALS-based approach.

The data definition given by Equation 2.1 is usually replaced by

$$x_{ijk} = \begin{cases} 1, & \text{if } \text{Rel}_k(\text{Entity}_i, \text{Entity}_j) \text{ holds} \\ 0, & \text{otherwise.} \end{cases} \tag{2.7}$$

Consequently, given a multi-relational graph defined by a KB, all the links among entities other than the ones contained in the KB are processed in an identical way, regardless of their veracity. In consequence, eventually these approaches really reconstruct a noisy version of the real tensor. If the observed tensor is very sparse (as it is often in real data), then these approaches will result in fitting a large number of *phantom zeros*.

London et al. (2013), Gao et al. (2011) have addressed this increment of noise in the data at setting all the unknown entries of the tensor to 0 by learning only from observed data by multiplying the entries of the reconstructed tensor with the corresponding entries of a non-negative matrix that adjust the importance of the values in the learning. In consequence, the learning complexity will be reduced, since it is limited to a reduced

set of samples out of the total.

2.2.3. Collective matrix factorization

Collective matrix factorization is an extension of matrix factorization in domains with more than one relation matrix with some degree of overlapping in the entity sets. In this case, the information is not a pile of relation matrices where the entities are always the same, but a collection of matrices sharing some entity type. For instance: an integer matrix X representing users' rating of movies on a scale (movies vs users), and a binary matrix Y representing the genres each movies belongs to (movies vs genres). Tensor factorization would be a particular case of collective matrix factorization, where there is a perfect overlap among entity types across the relation matrices.

A low-rank factorization of a matrix W has the form $W \approx f(U, V^T)$, as a consequence of minimizing an expression including a loss function, constraints, regularization terms, etc. Collective matrix factorization methods (Singh and Gordon 2008) would address the previous example by finding low rank approximations of such matrices as $X \approx f_1(U, V^T)$ and $Y \approx f_2(V, Z^T)$, with V as factor common given the fact that they share an entity type (movies). These approximations are found jointly at minimizing an overall loss function that averages both single factorization expressions.

Other interesting works within this category are those of Mukherjee et al. (2013) and Bouchard et al. (2013). The former extended the non-negative matrix factorization of Ding et al. (2006) to a collective matrix factorization framework, while the second proposes a convex formulation to resolve a collective matrix factorization.

2.2.4. Energy-based models

Energy-based models (see LeCun et al. 2006, for a review) are functions trained to assign low energy values to plausible triples/facts of a multi-relational graph, and high values otherwise. These functions rely on a distributed representation of the multi-relational data: any element (entity or label) is encoded into a low dimensional embedding space. The embeddings are learned and established by a neural network whose particular architecture allow to integrate the original data structure, while preserving and enhancing the complex structure of such data. By learning these embeddings, the model can be used to predict the plausibility of unknown facts.

Note that from above definition there is not a meaningful difference with respect to the works discussed so far, nevertheless, we prefer to use this category for referring to works that are presented in a clear neural network framework. Other than that, these works learn the graph triple by triple, unlike previous works, where the whole or a subset of the graph is updated at once. Consequently, the problem is defined as a set $D_x = \{x_i\}_{i=1}^N$ of N triples:

$$x_i = (h_i, l_i, t_i), \tag{2.8}$$

where h, l and t stand for *head*, *label* and *tail*, respectively.

Bordes et al. (2011) propose an architecture Structured Embeddings (SE) with the following energy function:

$$f(h, l, t) = ||R_l^{lhs} e_h - R_l^{rhs} e_t||_p, \tag{2.9}$$

where $e \in \mathbb{R}^d$, $R^{lhs}, R^{rhs} \in \mathbb{R}^{d \times d}$, d is the embedding dimension and p indicates the norm. The entity embeddings are transformed accordingly by the corresponding left- and right-relation matrix and then the similarity is measured in this transformed embedding space. Trying directly to minimize the objective function of Equation 2.9 would lead to a trivial solution (e.g. all zero embeddings for all the elements would give a zero-error). Instead, energy-based models define the following training methodology: given a triple (h_i,l_i,t_i) with a missing argument (typically either the left or right one) they would like their function f to predict the correct target entity. Formally:

$$\begin{aligned} f(h_i, l_i, t_i) &< f(h_j, l_i, t_i) + \gamma, \quad \forall j : (h_j, l_i, t_i) \notin D_x \\ f(h_i, l_i, t_i) &< f(h_i, l_i, t_j) + \gamma, \quad \forall j : (h_i, l_i, t_j) \notin D_x, \end{aligned}$$

where the scalar value γ is the *margin* as is commonly used in many margin-based models such as SVMs (Burges 1998). The difference between the energies of the correct answer and the corrupted one is penalized when larger than γ. This hyperparameter takes an important role to avoid overfitting and underfitting. A small value may lead to underfitting, whereas a high value may lead to overfitting.

As RESCAL, the entity embedding matrix is global and unique for all relationships, and contains factorized information coming from all the relations in which the entity is involved in a sort of multi-task learning.

Following this work, Bordes et al. (2014a) proposed the generic architecture Semantic Matching Energy (SME) of Figure 2.6. The first layer of the architecture maps each symbol of the triple to its embedding, then it is followed by a layer that combines the embeddings of the head and tail with the embedding of the relationship in order to obtain new relation-dependent embeddings ($E_{lhs}(rel)$ and $E_{rhs}(rel)$ in Figure 2.6). Finally the energy of such triple is computed via the dot product of both relation-dependent embeddings.

Different types of parametrizations can be used for the g and h functions, and consequently lead to two versions of SME[8]:

- SME(linear): a dot product for the output function h and a linear layer for the function g.
$$f(h, l, t) = (W_{h1} e_h^T + W_{h2} e_l^T)(W_{t1} e_t^T + W_{t2} e_l^T). \tag{2.10}$$
- SME(bilinear): a dot product for the output function h and a bilinear layer for the function g.
$$f(h, l, t) = ((W_l \bar{\times}_3 e_h^T) e_l^T)((W_t \bar{\times}_3 e_t^T) e_l^T), \tag{2.11}$$
where denotes $\bar{\times}_3$ the n-mode vector-tensor product along the 3rd mode.

[8] For clarity, bias terms are removed

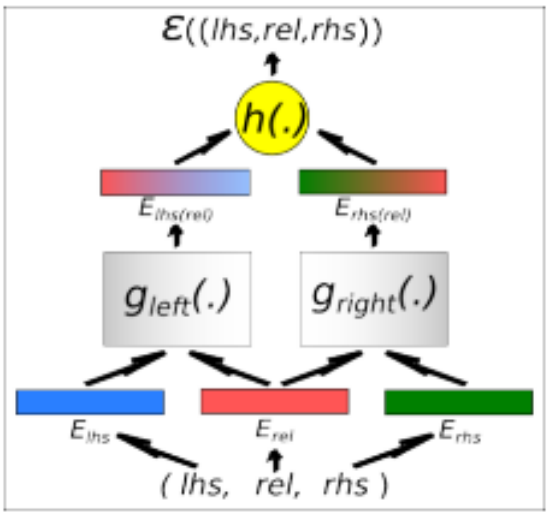


Figure 2.6.: SME architecture

Expanding and rearranging these 2 models, it is easy to show that SME(linear) is actually a sum of bigram terms, and SME(bilinear) is a trigram term similar to RESCAL (with different objective function and training procedures).

The Neural Tensor Network (NTN) (Socher et al. 2013) computes a score of how likely it is that two entities are in a certain relationship by the following energy function that encompasses both trigram and bigram terms wrapped up with a non-linear function:

$$f(h,l,t) = u_l^T g(e_h^T W_l^{[1:k]} e_t + V_l \begin{pmatrix} e_h \\ e_t \end{pmatrix} + b_l) \tag{2.12}$$

where $g = tanh$ is a standard nonlinearity applied element-wise, $W_l^{[1:k]} \in \mathbb{R}^{d \times d \times k}$, $V_l \in \mathbb{R}^{k \times 2d}$ and $u_l, b_l \in \mathbb{R}^k$. d and k are hyperparameters for the embedding dimension and the slices of the tensor layer respectively.

All the aforementioned methods within this category follow the same training procedure, namely:

1. Select a positive training triple x_i at random.

2. Select randomly one of the three arguments of the triple to corrupt it: an entity or relation is randomly drawn when the corrupted argument is the head/tail or the label, respectively, and a negative triple x_i^{neg} is constructed by replacing the corrupted argument by the randomly drawn element. This generation of negative facts does also introduce noise since the followed methodology cannot guarantee anything about its falseness, and consequently they can express true information (contained in D_x or not).

3. If $f(x_i)+\gamma > f(x_i^{neg})$ then the embeddings involved are updated by an optimization method (e.g. SGD). This is equivalent to a hinge loss function $\max(0, f(x_i) - f(x_i^{neg})+\gamma)$, and it only depends on the energy differences. Else, these embeddings are not updated.

This procedure is iterated over a fixed number of iterations.

A similar approach is followed by Jenatton et al. (2012), but with a clearer n-gram based parametrization framed in a probabilistic framework. The relation matrices are shared across all the n-gram terms and decomposed over a common set of R rank one matrices $\{\Theta_r\}_{r=1}^{R}$ representing some canonical relations $R_k = \sum_{r=1}^{n} \alpha_r^k \Theta_r$. Sharing parameters aims at avoiding overfitting, since the number of observations for some relations can be quite small, specially when the number of relations is high (which in real training data happens to be). This is called the Latent Factor Model (LFM).

The contributions of this thesis presented in Part II belong to this category.

Probabilistic interpretation

Energy-based models can be considered as an alternative to probabilistic approaches for learning. Nevertheless, we may turn these energy models into probabilistic based models in a straightforward, but costly, way. By obtaining the energy values when evaluating such function over the whole set of elements for the corrupted argument and applying a softmax layer over such values, we would end up with a probabilistic vision of the model.

For example, given the triple (Paris, capital_of, **?**) with **?** being the corrupted argument, we can evaluate the energy function for all the possible triples at replacing such argument with the whole set of entities, once at a time. After applying a softmax layer over these energy values, we would obtain the probabilities $P(? = e|(\text{Paris}, \text{capital\_of}, ?))$ for any $e \in V_e$, where V_e is the set of entities. A similar methodology is usually followed at test time to get the rank of the correct entity, but it can be followed at training time as well in order to optimize a log-likelihood function, instead of the classic ranking loss function used in these models.

2.2.5. Symbolic approaches

Symbolic approaches, as that of Kok and Domingos (2007), are also worth mentioning in this context. This approach, called Multiple Relational Clusterings (MRC), iteratively refines clusters of symbols[9] based on the clusters of symbols they appear in atoms with to eventually predict the probability of query atoms given the cluster membership of the symbols in them. 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. In the same spirit, the Infinite Relational Model (IRM) (Kemp et al. 2006),

[9] They use the terminology *symbols* for referring to *entities*, and *atoms* for *facts*

which is a nonparametric extension of the Stochastic Blockmodel (Wang and Wong 1987), assumes that the adjacency matrices are generated due to the presence of intrinsic clusters determining entity's tendency to participate in relations. Unlike MRC, in IRM the cluster assignment is unique. One limitation of IRM and MRC is that both models learn flat clustering rather hierarchical models, which is useful for human interpretability and for improving predictive performance. That approach is taken by Nath et al. (2012).

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 et al. (2014) cover this gap by combining that model with pre-trained embeddings. The PRA is also used in the KnowledgeVault project[10] (Dong et al. 2014) in conjunction with an embedding approach. However, an important drawback of PRA is at computing the probability of arriving to the target node given that the random walk began at a specific source node and it follows a specific path type, since it involves an exhaustive search of all the possible targets given that source node and that path type, count how frequently each target is seen and then normalize the probability distribution. The computation of these random walk probabilities have not shown a discernible benefit in the KB completion task, and consequently they are dropped (Gardner and Mitchell 2015).

For a good survey of rule-based algorithms for relational learning see (Getoor 2007).

2.2.6. Probabilistic models

Though most of the previous works could have been stated in a probabilistic framework, in this category we mention relevant works whose authors have explicitly presented and resolved in such framework.

Xiong et al. (2010) model temporal relational data where the third mode of the tensor corresponds to the time evolution. It decomposes the tensor as a CP instance, where the slices corresponding to the third mode depend only on its immediate predecessor, which is not suitable for modeling relational data where that dependency does not hold. In a similar way, Yoshii et al. (2013) aims at modeling non-binary and time-evolving data but without that strong dependency; instead the frontal slices of the tensor are approximated by a convex combination of global positive semidefinite matrices weighted by global and local weights.

A best fit for the kind of multi-relational data we deal with in this thesis are (Sutskever et al. 2009) and (Paccanaro and Hinton 2001). The former divides the relation and entity spaces in partitions, in a way that the truth-value of a triple is mainly determined by the

[10] It is a potential successor to Google's Knowledge Graph, bringing facts from across the entire web, including unstructured sources.

cluster assignments of the three arguments making up that fact, while in the latter the truth-value of a triple is given by a goodness function based on the goal that $R_l e_h \approx e_t$.

2.2.7. Collaborative filtering

Collaborative Filtering (CF) (Ekstrand et al. 2011) is a well-known technique for exploiting users' patterns in order to predict unknown patterns of a particular user. So, CF methods are widely used in recommender systems to, for instance, predict users' ratings on movies or songs. In this context, the underlying assumption is that if the relevant features (sometimes the own ratings) featuring two users are similar, then their preferences will also be. Here, the information is one-relation data that might be shaped as $Rel\ (user_x,\ item_y))$, where Rel expresses a degree of affinity between its arguments, and thus, the application of these methods are usually quite limited to multi-relational data.

2.3. Applications of Knowledge Graphs

Knowledge graphs represent a very common type of information, since we are surrounded by entities, which are connected by relations. Nevertheless, these knowledge graphs present several problems, namely:

- Incompleteness: not only link prediction methods tackle this problem, but also ontology matching and knowledge extraction methods address it in different ways.
- Correctness: it has to do not only with the veracity of the facts, but also with entities that are duplicated or have been wrongly merged (entity resolution).

The interest of completing their missing information and making them more accurate comes from its application to related fields such as information retrieval, natural language processing, machine learning or artificial intelligence. Some of its applications are:

- Exploratory search in search engines (e.g. New York sightseeing) or social networks (e.g. people who like Harvard University and Basketball and work at Facebook),
- Intelligence: such as question answering, where the performance of the model will be limited by the completeness and accurateness of the knowledge base, or knowledge discovery,
- NLP-related tasks, e.g. word-sense disambiguation.

Extracting structured facts from unstructured (text) and semi-structured sources (tables or pages with regular structure) may be, sometimes, unreliable. A good way to combat this is to use prior knowledge, derived from other kinds of data. This can be thought as link prediction in a graph, and hence that all these models that exploit existing triples in one or several knowledge bases can assign a probability to any possible triple. (Dong et al. 2014) follows this methodology to assign priors in order to build a Web-scale probabilistic knowledge base made up of confident facts called Knowledge Vault.

2.4. Contributions

In this section we break down the main contributions of this thesis.

2.4.1. Energy functions

Embedding-based models learn low-dimensional vectors for the entities, and the relationships act as operators on them to define scoring functions measuring the plausibility of triples. This operator defines the parametrization of relations as vectors or matrices.

In Chapter 4 we present TransE, a model based on translations in the embedding space, in such a way that it aims at $\mathbf{e}^h + \mathbf{r}^\ell \approx \mathbf{e}^t$ when (h, ℓ, t) holds, while $\mathbf{e}^h + \mathbf{r}^\ell$ should be far away from $\mathbf{e}^t$ otherwise, which is partially motivated by the work of Mikolov et al. (2013), where coincidentally they discovered this translation structure between different entity categories at learning word embeddings from text. Despite its simplicity it has proven very good performance for several datasets from the literature. We think that it is mainly for the vectorial representation of all the elements (both entities and relationships), which eases to regularize the model thanks to its reduced expressiveness.

The core of the scoring function of TransE are binary (2-way) interactions between the subject and the object, the subject and the relationship, and the object and the relationship. Three-way interaction is also a widely known modeling assumption taken in other works such as (Nickel et al. 2011, Bordes et al. 2014a). This approach parametrizes the relations as matrices, leading to an obvious increase in the capacity of the model, but subsequently to regularization problems. In Chapter 5, we show that these two kind of interactions respond to different data patterns and in order to take advantage of both patterns we propose TATEC, a model that aims at efficiently combining 2- and 3-way interactions. Unlike other works such as (Jenatton et al. 2012, Socher et al. 2013) we use two different embedding spaces to model each data pattern and consequently capture the complementary information they may encode. We do so by pretraining first the constituents of TATEC in a first stage, and then we train them jointly. We show in several benchmarks that this combination turns out beneficial, since TATEC outperforms always the best of its constituents. TATEC also outperforms TransE by a wide margin in these benchmarks, proving that the 3-way term encodes complementary information.

Due to the success of TransE and the ease in the interpretability of its results, we kept working on that approach. We noticed that TransE failed at making very basic reasoning when predicting the tail/head of a test triple. For example, given in the training set the facts (`John, born_in, London`) and (`London, contained_by, England`) it should be trivial for the model to predict the nationality of `John`. However, the prediction in these cases was not as good as expected. We attribute this to the training methodology. All the models until that moment learned the multi-relational graph with 1-hop facts, which along the effect of the ranking loss function, made the model to fail at inferring very simple reasoning. In the aforementioned example, TransE guaranteed that the embedding `John + born_in` is the closest one to `London`, and also that the embedding

`London + contained_by` is the closest one to `England`, but not that the distances are small. When going from `John` to `England` through `born_in + contained_by` the model is affected by the cascading error, making the model unable to get the relationship `nationality` to give a similar result as the composition `born_in` and `contained_by`. In Chapter 6 we propose a modification of TransE, called rTransE, that is more amenable to that kind of composionality that guarantees and favors this basic reasoning. We handle it through regularization on a training set augmented with relevant examples of such compositions, weighting the importance of each composition by a reliability factor.

Our future plans of experimenting with rTransE on paths other than "unambiguous" ones and different regularization strategies were not carried out because of the two extended versions (Gu et al. 2015, Lin et al. 2015a) of our model presented at the same conference as rTransE. As we will discuss in Section 6.4, these works include experiments with hops bigger than 2 and "ambiguous" paths leading to a superior performance.

2.4.2. Settings and protocols

Apart from proposals of energy functions, this thesis has also focused on improving the evaluation protocols and studying the effect of different regularization schemes in these models.

Regularization in these models is difficult, different schemes have been used ranging from enforcing unitary L2-norm (Bordes et al. 2011; 2013), to including penalization terms in the cost function (Nickel et al. 2011, Wang et al. 2014) (called *soft* regularization in this thesis) passing by projecting the embeddings into the L2-norm ball of a given radius (García-Durán et al. 2014) (called *hard* regularization in this thesis). Therefore, in Chapter 5 we systematically evaluate both TransE and TATEC on several benchmarks from the literature. Based on that study we conclude that both schemes show a similar performance, with the advantage of requiring one less hyperparameter brought by the *hard* scheme.

In terms of evaluation this thesis has also meant a leap forward. We have proposed the *filtered* setting in order to have a more accurate measure of the performance of the model in the link prediction task. In contrast to the *raw* setting, the *filtered* one removes all the triples ranked higher than the target ones that appear in either training, validation or test set. We have also established different categories in which the performance of the models can be broken down according to the nature of the triple to get a better understanding of it. Specifically:

1. Regarding the cardinality of the *head* and *tail* arguments, a relationship can be categorized as `1-to-1`, `Many-to-1`, `1-to-Many` and `Many-to-Many`,
2. Regarding the existence of a reciprocal triple in the training set or not, a test triple can be categorized as *easy* or *hard* to predict.

According to our experience while the first category has not proven to be very useful, since none of the studied models have shown a big advantage for one or a set of these specific categories, the second category has proven more interesting. For example, though the overall performance of TATEC is better that TransE's, the latter seems a better option in case of KBs where reciprocal information is not a majority.

We have also introduced a new evaluation protocol: link prediction on quadruples. Though in Chapter 6 we restrict ourselves to link prediction in quadruples, the proposed evaluation may be of interest for paths of arbitrary length.

2.4.3. A novel application: Question Generation

As previously mentioned, KBs have been widely used as essential side knowledge to address problems such as entity disambiguation (Zheng et al. 2012), information extraction (Hoffmann et al. 2011) or relation extraction (Weston et al. 2013, Chang et al. 2014).

In Chapter 7 we propose a novel application to make use of the learned embeddings of a KB. Given a set of question fact pairs, we have trained a machine translation model where the input is the concatenation of the embeddings of the three arguments making up a triple, and the output is an associated question to that fact in english language. The model learns to frame questions by making associations between the semantics of the subject and object, by means of a relationship, and then outputting an appropriate question based on these associations. Since the model learns these semantic associations and these semantics are captured in the embeddings (in this work by TransE), it is able to output a question given a fact made up of elements that have not previously be seen during training.

To our knowledge, this is the first work on question generation from structured data by means of neural networks. Our model generates questions that are preferred over the template-based ones, by a wide margin, according to human evaluation.

In the Chapters 4, 5 and 6 we present the energy-based models that form the core of this work. Finally, in Chapter 7 we make use of these learned representations as input in a machine translation model to generate questions in english language.

Part II.

Papers

3. Introduction

In this chapter we introduce some pieces of information, such as the benchmarks or the evaluation tasks, which are common to Chapters 4, 5 and 6.

3.1. Benchmarks

We report in this section the datasets over which we have evaluated our models. Following we briefly describe them:

- **FB15k**: Freebase is a huge and growing KB of general facts; there are currently around 1.2 billion triples and more than 80 million entities. Bordes et al. (2013) created two data sets from Freebase. First, to make a small data set to experiment on we selected the subset of entities that are also present in the Wikilinks database[1] and that also have at least 100 mentions in Freebase (for both entities and relationships) were selected. Relationships like !/people/person/nationality which just reverses the head and tail compared to the relationship /people/person/nationality were removed. This resulted in 592,213 triples with 14,951 entities and 1,345 relationships which were randomly split as shown in Table 3.1. This data set is denoted **FB15k** in the rest of this thesis. Apart from that, a large-scale data from Freebase were created by Bordes et al. (2013) by selecting the most frequently occurring 1 million entities. This led to a split with around 25k relationships and more than 17 millions training triples, which is referred to as **FB1M**.

- **SVO**: SVO is a database of nouns connected by verbs through subject-verb-direct object triples and extracted from Wikipedia articles. All triples are unique and the words appearing in the validation or test sets are occurring in the training set. It was introduced by Jenatton et al. (2012). Statistics are given in Table 3.1.

- **WordNet**: This KB is designed to produce an intuitively usable dictionary and thesaurus, and support automatic text analysis. Its entities (termed synsets) correspond to word senses, and relationships define lexical relations between them. We considered the data version used by Bordes et al. (2014a), which we denote WN in the following. Examples of triples are (score_NN_1, hypernym, evaluation_NN_1) or (score_NN_2, has part, musical_notation_NN_1)[2]. Table 3.1 provides statistics for this dataset.

[1] code.google.com/p/wiki-links

[2] WN is composed of senses, its entities are denoted by the concatenation of a word, its part-of-speech tag and a digit indicating which sense it refers to i.e. score NN_1 encodes the first meaning of the noun score.

Table 3.1.: **Statistics of the data sets** used in this thesis and extracted from five knowledge bases: Freebase, SVO, WordNet, Kinships and UMLS

Data set	FB15k	FB1M	SVO	WordNet	Kinships	UMLS
Entities	14,951	1×10^6	30,605	40,943	104	135
Relationships	1,345	23,382	4,547	18	26	49
Training examples	483,142	17.5×10^6	1,000,000	141,442	224,973	102,612
Validation examples	50,000	50,000	50,000	5,000	28,122	89,302
Test examples	59,071	177,404	250,000	5,000	28,121	89,302

- **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. See Table 3.1 for more information.

3.2. Evaluation tasks

Since the energy-based models use a ranking loss function (see Section 2.2.4), which compares the score of a positive triple against a negative one (one at a time), we need a methodology to generate such negative triples whenever they are not provided by the KB.

Let $\mathcal{S}$ be the set of positive triples provided by the KB, the set of negative triples $\mathcal{C}(h, l, t)$ is defined in 3 different ways depending on the application. Formally, these 3 methodologies are defined as follows:

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

where $[\![E]\!]$ and $\mathcal{L}$ are the set of entities and relationships, respectively.

We perform link prediction as evaluation task for experiments in Freebase (both FB15k and FB1M) and Wordnet. In the latter application we follow the second setting to generate negative triples. 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. 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*.

Another related evaluation task is verb prediction: for each test relationship, we rank all verbs using our energy-based models given a pair (subject, direct object). As before, these scores are sorted in descending order and the rank of the correct verb is stored. For this application, we generate negative triples following the third aforementioned setting. Experiments on SVO are evaluated in this manner.

The mean of those predicted ranks is the *mean rank*[3] and the *hits@n* is the proportion of correct entities ranked in the top n. Similarly, the *hits@n%* is the proportion of correct entities ranked in the top $n\%$ of the total number of elements. Therefore, the lower that value of *mean rank* is, the better that model performs; and the same in the other way around for *hits@n* and *hits@n%*.

For those datasets for which the whole set of triples, positive or negative, is observed (as UMLS and KINSHIPS), we formulate a binary classification evaluation task, i.e. we classify the triples as positives or negatives. In this case, we compare one random positive triple against a random negative one (that follows the first methodology). We compute the area under the precision-recall curve. This area is a single number summary of the information in such curve, and weights the importance of both precision (fraction of retrieved triples that are positive) and recall (fraction of positive triples that are successfully retrieved). Thus, the higher the better.

[3]Nevertheless we think that the median rank would be more appropriate given it is more robust than the mean, but for historical reasons we have stuck to this reference metric.

4. Translating Embeddings for Modeling Multi-relational Data

This chapter corresponds to the paper Bordes et al. (2013) *Translating Embedding for Multi-relational data*. Bordes, A., Usunier, N., **García-Durán, A.**, Weston, J., Yakhnenko, O. In Advances in Neural Information Processing Systems (pp. 2787-2795).

4.1. Introduction

Multi-relational data refers to directed graphs whose nodes correspond to *entities* and *edges* of the form (*head*, *label*, *tail*), each of which indicates that there exists a relationship of name *label* between the entities *head* and *tail*. Models of multi-relational data play a pivotal role in many areas such as social network analysis, where entities are members and edges (relationships) are friendship/social relationship links, recommender systems where entities are users and products and relationships are buying, rating, reviewing or searching for a product, to KBs such as Freebase, Google Knowledge Graph or GeneOntology, where each entity of the KB represents an abstract concept or concrete entity of the world and relationships are predicates that represent facts involving two of them. Our work focuses on modeling multi-relational data from KBs (Wordnet and Freebase in this paper), with the goal of providing an efficient tool to complete them by automatically adding new facts, without requiring extra-knowledge.

Modeling multi-relational data In general, the modeling process boils down to extracting local or global connectivity patterns between entities, and prediction is performed by using these patterns to generalize the observed relationship between a specific entity and all others. The notion of locality for a single relationship may be purely structural, such as the friend of my friend is my friend in social networks, but can also depend on the entities, such as those who liked Star Wars IV also liked Star Wars V, but they may or may not like Titanic. In contrast to single-relational data where ad-hoc but simple modeling assumptions can be made after some descriptive analysis of the data, the difficulty of relational data is that the notion of locality may involve relationships and entities of different types at the same time, so that modeling multi-relational data requires more generic approaches that can choose the appropriate patterns considering all heterogeneous relationships at the same time.

Following the success of user/item clustering or matrix factorization techniques in collaborative filtering to represent non-trivial similarities between the connectivity patterns of entities in single-relational data, most existing methods for multi-relational data

have been designed within the framework of *relational learning from latent attributes* as pointed out by Jenatton et al. (2012). That is, by learning and operating with latent representations (or embeddings) of the constituents (entities and relationships). Starting from natural extensions of these approaches to the multi-relational domain (see Chapter 2) such as a non-parametric Bayesian extension of the *stochastic blockmodel* (Kemp et al. 2006, Miller et al. 2009, Zhu 2012) and models based on tensor factorization (Harshman 1970) or collective matrix factorization (Singh and Gordon 2008) many of the most recent approaches have focused on increasing the expressivity and the universality of the model in either Bayesian clustering frameworks (Sutskever et al. 2009) or energy-based frameworks for learning embeddings of entities in low-dimensional spaces (Bordes et al. 2011, Sutskever et al. 2009). The greater expressivity of these models comes at the expense of substantial increases in model complexity which results in modeling assumptions that are hard to interpret, higher computational cost and, potentially, overfitting. Indeed, Bordes et al. (2014a) show that a simpler model (linear instead of bilinear) achieves almost as good performance as the most expressive models on several multi-relational datasets with a relatively large number of different relationships. This suggests that even in complex and heterogeneous multi-relational domains simple yet appropriate modeling assumptions can lead to better trade-offs between accuracy and scalability.

Relationships as translations in the embedding space In this paper, we introduce TransE, an energy-based model for learning low-dimensional embeddings of entities, in which relationships are represented as *translations in the embedding space*: if (h, ℓ, t) holds, then the embedding of t should be close to the embedding of h plus some vector that depends on ℓ. Our approach relies on a reduced set of parameters as it learns only one low-dimensional vector for each entity and each relationship.

The main motivation behind our translation-based parameterization is that hierarchical relationships are extremely common in KBs and translations are the natural transformations for representing them. Indeed, considering the natural representation of trees (i.e. embeddings of the nodes in dimension 2), the siblings are close to each other and nodes at a given height are organized on the x-axis, the parent-child relationship corresponds to a translation on the y-axis. Since a null translation vector corresponds to an equivalence relationship between entities, the model can then represent the sibling relationship as well. Hence, we chose to use our parameter budget per relationship (one low-dimensional vector) to represent what we considered to be the key relationships in KBs. Another, secondary, motivation comes from the recent work of Mikolov et al. (2013), who learn word embeddings from free text, and some *1-to-1* relationships between entities of different types, such "capital of" between countries and cities, are (coincidentally rather than willingly) represented by the model as translations in the embedding space. This suggests that there may exist embedding spaces in which *1-to-1* relationships between entities of different types may, as well, be represented by translations. The intention of our model is to enforce such a structure of the embedding space.

Our experiments in Section 4.3 demonstrate that this new model, despite its simplicity

and that it is primarily designed for modeling hierarchies, is actually very powerful on most kinds of relationships, and can significantly outperform previous methods in link prediction on real-world KBs. Besides, its light parameterization allows it to be successfully trained on the large scale split FB1M of Freebase containing 1M entities, 25k relationships and more than 17M training samples.

4.2. Translation-based model

Given a training set S of triples (h, ℓ, t), our model learns vector embeddings of the entities and the relationships. The embeddings take values in $\mathbb{R}^k$ (k is a model hyperparameter) and are denoted with the same letters, in boldface characters. The basic idea behind our model is that the functional relation induced by the ℓ-labeled arcs corresponds to a translation of the embeddings, i.e. we want that $\mathbf{e}^h + \mathbf{r}^\ell \approx \mathbf{e}^t$ when (h, ℓ, t) holds, while $\mathbf{e}^h + \mathbf{r}^\ell$ should be far away from $\mathbf{e}^t$ otherwise. Following an energy-based framework, the energy of a triple is equal to $d(\boldsymbol{h} + \boldsymbol{\ell}, \boldsymbol{t})$ for some dissimilarity measure d, which we take to be either the L_1 or the L_2-norm.

To learn such embeddings, we minimize the following margin-based ranking criterion over the training set:

$$\sum_{(h,\ell,t)\in S} \sum_{(h',\ell,t')\in S'_{(h,\ell,t)}} \left[\gamma + d(\boldsymbol{h} + \boldsymbol{\ell}, \boldsymbol{t}) - d(\boldsymbol{h'} + \boldsymbol{\ell}, \boldsymbol{t'})\right]_+ \tag{4.1}$$

where $[x]_+$ denotes the positive part of x, $\gamma > 0$ is the margin hyperparameter, and $S'_{(h,\ell,t)}$ is the set of negative triples, constructed according to the second setting of Section 3.2, which are basically the training (positive) triples with either the head or tail replaced by a random entity (but not both at the same time). The loss function (4.1) favors lower values of the energy for positive triples than for negative triples, and is thus a natural implementation of the intended criterion. Note that for a given entity, its embedding vector is the same when the entity appears as the head or as the tail of a triple.

The optimization is carried out by stochastic gradient descent (in minibatch mode), over the possible $\mathbf{e}^h, \mathbf{r}^\ell$ and $\mathbf{e}^t$, with the additional constraints that the L_2-norm of the embeddings of the entities is 1 (no regularization or norm constraints are given to the label embeddings $\mathbf{r}^\ell$).

The detailed optimization procedure is described in Algorithm 1. All embeddings for entities and relationships are first initialized following the random procedure proposed by Glorot and Bengio (2010). At each main iteration of the algorithm, the embedding vectors of the entities are first normalized. Then, a small set of triples is sampled from the training set, and will serve as the positive triples of the minibatch. For each such positive triple, we then sample a single negative triple. The parameters are then updated by taking a gradient step with constant learning rate. The algorithm is stopped based on its performance on the validation set.

Algorithm 1 Learning TransE

Input Training set $S = \{(h, \ell, t)\}$, margin γ, learning rate λ
initialize $\mathbf{r} \leftarrow$ Uniform$(-\frac{6}{\sqrt{k}}, \frac{6}{\sqrt{k}})$ for each ℓ
    $\mathbf{r} \leftarrow \mathbf{r}/\left\|\mathbf{r}\right\|$ for each ℓ ▷ This changes the boundaries of U
    $\mathbf{e} \leftarrow$ Uniform$(-\frac{6}{\sqrt{k}}, \frac{6}{\sqrt{k}})$ for each entity e
while some condition **do**
    $\mathbf{e} \leftarrow \mathbf{e}/\left\|\mathbf{e}\right\|$ for each entity e
    $S_{batch} \leftarrow sample(S, b)$ //sample minibatch of size b
    $T_{batch} \leftarrow \emptyset$ //initialize set of pairs
    for $(h, \ell, t) \in S_{batch}$ **do**
        $(h', \ell, t') \leftarrow$sample$(S'_{(h,\ell,t)})$ //sample negative triple
        $T_{batch} \leftarrow T_{batch} \cup \left\{\left((h, \ell, t), (h', \ell, t')\right)\right\}$
    end for
    Update embeddings w.r.t. $\sum_{\left((h,\ell,t),(h',\ell,t')\right) \in T_{batch}} \nabla^1 \left[\gamma + d(\boldsymbol{h} + \boldsymbol{\ell}, \boldsymbol{t}) - d(\boldsymbol{h'} + \boldsymbol{\ell}, \boldsymbol{t'})\right]_+$
end while

Table 4.1.: **Numbers of parameters** and their values for FB15k (in millions). n_e and n_r are the nb. of entities and relationships; d the embeddings dimension.

Method	Nb. of parameters	on FB15k
Unstructured	$O(n_e k)$	0.75
RESCAL	$O(n_e k + n_r k^2)$	87.80
SE	$O(n_e k + 2n_r k^2)$	7.47
SME(linear)	$O(n_e k + n_r k + 4k^2)$	0.82
SME(bilinear)	$O(n_e k + n_r k + 2k^3)$	1.06
LFM	$O(n_e k + n_r k + 10k^2)$	0.84
TransE	$O(n_e k + n_r k)$	0.81

4.3. Experiments

Our approach, TransE, is evaluated on data extracted from Wordnet and Freebase against several recent methods from the literature which have shown to achieve the best current performance on various benchmarks and to scale to relatively large data sets.

Datasets We have used Freebase and Wordnet as benchmarks. See Section 3.1 for a more detailed explanation of these KBs.

[1]For simplicity, we use the symbol ∇ to refer to the gradient w.r.t. the parameters of the model

4.3.1. Experimental setup

Evaluation protocol We use link prediction as evaluation task.

We report the *mean* of the predicted ranks and the *hits@10* in both *raw* and *filtered* settings on WN and FB15k. Only *raw* results are provided for experiments on FB1M.

Baselines The first method is Unstructured, the natural counterpart of TransE, which considers the data as mono-relational and sets all translations to $\mathbf{0}$. We also compare with RESCAL, the collective matrix factorization model presented by Nickel et al. (2011), and the energy-based models SE (Bordes et al. 2011), SME(linear) and SME(bilinear) (Bordes et al. 2014a) and LFM (Jenatton et al. 2012). RESCAL is trained via an alternating least-square method, whereas the others are trained by stochastic gradient descent, as is TransE. Table 4.1 compares the theoretical number of parameters of the baselines to our model, and gives the order of magnitude on FB15k. While SME(linear), SME(bilinear), LFM and TransE have about the same numbers of parameters as Unstructured for low dimensional embeddings, the other algorithms SE and RESCAL, which learn at least one $k \times k$ matrix for each relationship rapidly need to learn many parameters. RESCAL needs about 87 times more parameters on FB15k because it requires a much higher dimensional embedding than other models to achieve good performance. We did not experiment on FB1M with RESCAL, SME(bilinear) and LFM for scalability reasons in terms of numbers of parameters or training duration.

We trained all counterpart methods using the code provided by the authors. For RESCAL, we had to set the regularization parameter λ to 0 for scalability reasons, as it is indicated in the paper, and chose the latent dimension k among $\{50, 250, 500, 1000, 2000\}$ which provided the lowest mean predicted ranks on the validation sets (using the *raw* setting). For Unstructured, SE, SME(linear) and SME(bilinear), we selected the learning rate among $\{0.001, 0.01, 0.1\}$, k among $\{20, 50\}$, and early stopping using the mean rank on the validation (with a total of at most 1,000 epochs over the training data). For LFM, we also used the mean validation ranks to select the model and to choose the latent dimension among $\{25, 50, 75\}$, the number of factors among $\{50, 100, 200, 500\}$ and the learning rate among $\{0.01, 0.1, 0.5\}$.

Implementation For experiments with TransE, we selected the learning rate for the stochastic gradient descent among $\{0.001, 0.01, 0.1\}$ and chose the margin γ among $\{1, 2, 10\}$. The dissimilarity measure d was set either to the L_1 or L_2 distance. We fixed the latent dimension k of the embeddings to 20 on WN and 50 on FB15k and FB1M. For both datasets, the training time was limited to at most $1,000$ epochs over the training set. The best models were selected using the mean predicted ranks on the validation sets (*raw* setting).

4.3.2. Link prediction

Table 4.2.: **Link prediction results.** We compare our model, TransE, with several methods from the literature on three datasets. **Bold** indicates best results.

Dataset	WN				FB15k				FB1M	
Metric	Mean Rank		Hits@10 (%)		Mean Rank		Hits@10 (%)		Mean Rank	Hits@10 (%)
Eval. setting	*Raw*	*Filt.*	*Raw*	*Filt.*	*Raw*	*Filt.*	*Raw*	*Filt.*	*Raw*	*Raw*
Unstructured	315	304	35.3	38.2	1,074	979	4.5	6.3	15,139	2.9
RESCAL	1,180	1,163	37.2	52.8	828	683	28.4	44.1	-	-
SE	1,011	985	68.5	80.5	273	162	28.8	39.8	22,044	17.5
SME(linear)	545	533	65.1	74.1	274	154	30.7	40.8	-	-
SME(bilinear)	526	509	54.7	61.3	284	158	31.3	41.3	-	-
LFM	469	456	71.4	81.6	283	164	26.0	33.1	-	-
TransE	**263**	**251**	**75.4**	**89.2**	**243**	**125**	**34.9**	**47.1**	**14,615**	**34.0**

Overall results Tables 4.2 displays the results on all data sets for all compared methods. As expected, the *filtered* setting provides lower mean ranks and higher hits@10, which we believe are a clearer evaluation of the performance of the methods in link prediction. However, generally the trends between *raw* and *filtered* are the same.

Our method, TransE, outperforms all counterparts on all metrics, usually with a wide margin and reaches some promising absolute performance scores such as 89% of hits@10 on WN (over more than 40k entities) and 34% on FB1M (over 1M entities).

We believe that the good performance of TransE is due to an appropriate design of the model according to the data, but also to its relative simplicity. The latter means that it can be optimized efficiently with stochastic gradient. We show in Section 4.4 that SE is more expressive than our proposal. However, its complexity may make it quite hard to learn, resulting in worse performance. SME(bilinear) and LFM suffer from the same training issue: we never managed to train them well enough so that they could exploit their full capabilities. The poor results of LFM might also be explained by our evaluation setting, based on ranking entities, whereas LFM was originally proposed to predict relationships. RESCAL can achieve quite good hits@10 on FB15k but yields poor mean ranks, especially on WN, even when we used large latent dimensions (2000 on Wordnet).

The impact of the translation term is huge. When one compares performance of TransE and Unstructured (i.e. TransE without translation), mean ranks of Unstructured appear to be rather good (best runner-up on WN), but hits@10 are very poor. Unstructured simply clusters all entities co-occurring together, independent of the relationships involved, and hence can only make guesses of which entities are related. On FB1M the mean ranks of TransE and Unstructured are almost similar, but TransE places 10 times more predictions in the top 10.

Detailed results Table 4.3 breaks down the results in hits@10 on FB15k depending on the category of the relationships and the argument to predict for several of the methods. We categorized the relationships according the cardinalities of their *head* and *tail* arguments into four classes: 1-to-1, 1-to-Many, Many-to-1, Many-to-Many.

Table 4.3.: **Detailed results by category of relationship.** We compare Hits@10 (in %) on FB15k in the filtered evaluation setting for our model, TransE and counterparts. (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.
Unstructured	34.5	2.5	6.1	6.6	34.3	4.2	1.9	6.6
SE	35.6	62.6	17.2	37.5	34.9	14.6	68.3	41.3
SME(LINEAR)	35.1	53.7	19.0	40.3	32.7	14.9	61.6	43.3
SME(BILINEAR)	30.9	**69.6**	**19.9**	38.6	28.2	13.1	**76.0**	41.8
TransE	**43.7**	65.7	18.2	**47.2**	**43.7**	**19.7**	66.7	**50.0**

A given relationship is 1-TO-1 if a *head* can appear with at most one *tail*, 1-TO-MANY if a *head* can appear with many *tails*, MANY-TO-1 if many *heads* can appear with the same *tail*, or 1-TO-MANY if multiple *heads* can appear with multiple *tails*. We automatically classified the relationships into these four classes by computing, for each relationship ℓ, the averaged number of *heads* h (respect. *tails* t) appearing in the FB15k data set, given a pair (ℓ, t) (respect. a pair (h, ℓ)). If this average number was below 1.5 then the argument was labeled as 1 and MANY otherwise. For example, a relationship having an average of 1.2 *head* per *tail* and of 3.2 *tails* per *head* was classified as *1-to-Many*. We obtain that FB15k has 26.2% of 1-TO-1 relationships, 22.7% of 1-TO-MANY, 28.3% of MANY-TO-1, and 22.8% of MANY-TO-MANY.

These detailed results in Table 4.3 allow for a precise evaluation and understanding of the behavior of the methods. First, it appears that, as one would expect, it is easier to predict entities on the "side 1" of triples (i.e., predicting *head* in 1-TO-MANY and *tail* in MANY-TO-1), that is when multiple entities point to it. These are the well-posed cases. SME(bilinear) proves to be very accurate in such cases because they are those with the most positive examples. Unstructured performs well on 1-TO-1 relationships: this shows that arguments of such relationships must share common hidden types that Unstructured is able to somewhat uncover by clustering entities linked together in the embedding space. But this strategy fails for any other category of relationship. Adding the translation term (i.e. upgrading Unstructured into TransE) brings the ability to move in the embeddings space, from one entity cluster to another by following relationships. This is particularly spectacular for the well-posed cases.

Illustration Table 4.4 gives examples of nearest link prediction results of TransE on the FB15k test set (predicting *tail*). This illustrates the capabilities of our model. Given a head and a label, the top predicted tails (and the true one) are depicted. The examples come from the FB15k test set. Even if the good answer is not always top-ranked, the predictions reflect common-sense.

Table 4.4.: **Example predictions** on the FB15k test set using TransE. **Bold** indicates the test triple's true tail and *italics* other true tails present in the training set. Actual Freebase identifiers have been replaced by readable strings.

Input (Head and Label)	Predicted Tails
J. K. Rowling `influenced by`	*G. K. Chesterton*, J. R. R. Tolkien, *C. S. Lewis*, **Lloyd Alexander**, Terry Pratchett, Roald Dahl, Jorge Luis Borges, *Stephen King*, Ian Fleming
Anthony LaPaglia `performed in`	*Lantana*, *Summer of Sam*, *Happy Feet*, *The House of Mirth*, Unfaithful, **Legend of the Guardians**, Naked Lunch, X-Men, The Namesake
Camden County `adjoins`	**Burlington County**, *Atlantic County*, *Gloucester County*, Union County, Essex County, New Jersey, Passaic County, Ocean County, Bucks County
The 40-Year-Old Virgin `nominated for`	*MTV Movie Award for Best Comedic Performance*, *BFCA Critics' Choice Award for Best Comedy*, *MTV Movie Award for Best On-Screen Duo*, MTV Movie Award for Best Breakthrough Performance, **MTV Movie Award for Best Movie**, MTV Movie Award for Best Kiss, D. F. Zanuck Producer of the Year Award in Theatrical Motion Pictures, Screen Actors Guild Award for Best Actor - Motion Picture
David Foster `has the genre`	*Pop music*, *Pop rock*, Adult contemporary music, Dance music, **Contemporary R&B**, Soft rock, Rhythm and blues, Easy listening
Costa Rica football team `has position`	*Forward*, *Defender*, *Midfielder*, **Goalkeepers**, Pitchers, Infielder, Outfielder, Center, Defenseman
Lil Wayne `born in`	**New Orleans**, Atlanta, Austin, St. Louis, Toronto, New York City, Wellington, Dallas, Puerto Rico
WALL-E `has the genre`	Animations, Computer Animation, *Comedy film*, *Adventure film*, *Science Fiction*, **Fantasy**, Stop motion, *Satire*, Drama
Richard Crenna `has cause of death`	*Pancreatic cancer*, **Cardiovascular disease**, Meningitis, Cancer, Prostate cancers, Stroke, Liver tumour, Brain tumor, Multiple myeloma

4.3.3. Learning to predict new relationships with few examples

Using FB15k, we wanted to test how well methods could generalize new facts into KBs by checking how fast they were learning new relationships. To that end, we randomly selected 40 relationships and split the data into two sets: a set (termed *FB15k-40rel*) containing all triples containing these 40 relationships and another set (*FB15k-rest*) containing the rest. We made sure that both sets contained all entities. *FB15k-rest* has then been split into a training set of 353,788 triples and a validation set of 53,266, and *FB15k-40rel* into a training set of 40,000 triples (1,000 for each relationship) and a test set of 45,159. Using these data sets, we conducted the following experiment: (1) models were trained and selected using *FB15k-rest* training and validation sets, (2) they were subsequently trained on the training set *FB15k-40rel* but only to learn the parameters related to the fresh 40 relationships, (3) they were evaluated in link prediction on the test set of *FB15k-40rel* (containing only relationships unseen during phase (1)). We repeated this procedure while using 0, 10, 100 and 1000 examples of each relationship in phase (2).

Results for Unstructured, SE, SME(linear), SME(bilinear) and TransE are presented in Figure 4.1. The performance of Unstructured is the best when no example of the unknown relationship is provided, because it does not use this information to predict. But, of course, this does not change while providing labeled examples. TransE is the fastest method to learn: with only 10 examples of a new relationship, the hits@10 is already 18% and it improves monotonically with the number of provided samples. We

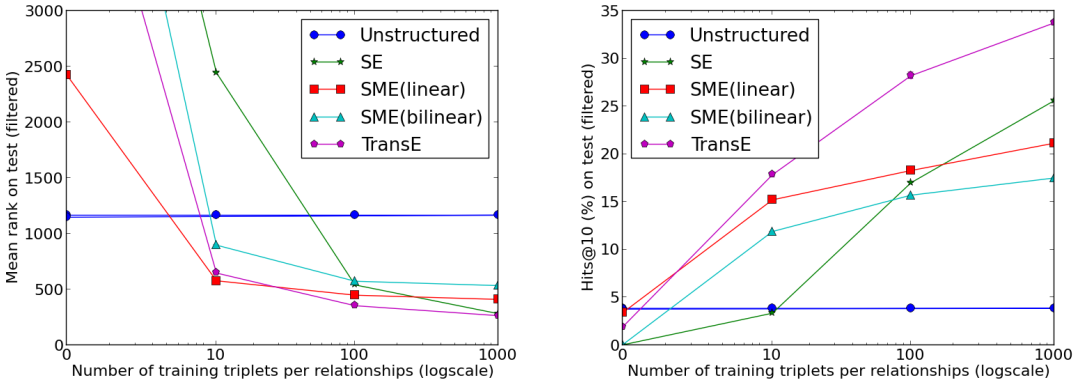


Figure 4.1.: **Learning new relationships with few examples**. Comparative experiments on FB15k data evaluated in mean rank (left) and hits@10 (right). More details in the text.

believe the simplicity of the TransE model makes it able to generalize well, without having to modify any of the already trained embeddings.

4.4. Discussion on related works

We detail here the relationships between our model and those of Bordes et al. (2011) (SE) and Socher et al. (2013) (NTN).

SE Bordes et al. (2011) embeds entities into $\mathbb{R}^k$, and relationships into two matrices $\mathbf{R}^\ell_{\mathbf{lhs}} \in \mathbb{R}^{\mathbf{k}\times\mathbf{k}}$ and $\mathbf{R}^\ell_{\mathbf{rhs}} \in \mathbb{R}^{\mathbf{k}\times\mathbf{k}}$ such that $d(\mathbf{R}^\ell_{\mathbf{lhs}}\mathbf{e}^{\mathbf{h}}, \mathbf{R}^\ell_{\mathbf{lhs}}\mathbf{e}^{\mathbf{t}})$ is small for positive triples (h, ℓ, t) (and large otherwise). The basic idea is that when two entities belong to the same triple, their embeddings should be close to each other in some subspace that depends on the relationship. Using two different projection matrices for the head and for the tail is intended to account for the possible asymmetry of relationship ℓ. When the dissimilarity function takes the form of $d(\boldsymbol{x}, \boldsymbol{y}) = g(\boldsymbol{x} - \boldsymbol{y})$ for some $g : \mathbb{R}^k \to \mathbb{R}$ (e.g. g is a norm), then the model of SE with an embedding of size $k+1$ is strictly more expressive than our model with an embedding of size k, since linear operators in dimension $k+1$ can reproduce affine transformations in a subspace of dimension k (by constraining the $k+1$st dimension of all embeddings to be equal to 1). SE, with $\mathbf{R}^\ell_{\mathbf{rhs}}$ as the identity matrix and $\mathbf{R}^\ell_{\mathbf{lhs}}$ taken so as to reproduce a translation is then equivalent to our model. Despite the lower expressiveness of our model, we still reach better performance than this model in our experiments. We believe this is because (1) our model is a more direct way to represent the true properties of the relationship, and (2) regularization, and more generally any form of capacity control, is difficult in embedding models; greater expressiveness may then be more synonymous to overfitting than to better performance.

As previously mentioned, another related model is the NTN (Socher et al. 2013). A

Table 4.5.: **Scoring function for several models related to TransE.** 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	$\|\|(\mathbf{e}^h - \langle \mathbf{w}^\ell \| \mathbf{e}^h \mathbf{w}^\ell \rangle) + \mathbf{r}^\ell - (\mathbf{e}^t - \langle \mathbf{w}^\ell \| \mathbf{e}^t \mathbf{w}^\ell \rangle)\|\|_2$
TransR	$\|\|\langle \mathbf{e}^h \| \mathbf{M}^\ell \rangle + \mathbf{r}^\ell - \langle \mathbf{e}^t \| \mathbf{M}^\ell \rangle\|\|_2$
SE	$\|\|\langle \mathbf{R}^\ell_{\mathbf{lhs}} \| \mathbf{e}^h \rangle - \langle \mathbf{R}^\ell_{\mathbf{rhs}} \| \mathbf{e}^t \rangle\|\|_1$
NTN	$\langle u^\ell \| f(\langle \mathbf{e}^t \| \mathbf{W}^{\ell\ [1:k]} \| \mathbf{e}^t \rangle + \langle \mathbf{V}^\ell \| \begin{pmatrix} \mathbf{e}^h \\ \mathbf{e}^t \end{pmatrix} \rangle \rangle + \mathbf{b}^\ell)$

special case of that model corresponds to learn scores $s(h,\ell,t)$ (higher scores for positive triples) of the form:

$$s(h,\ell,t) = \langle \mathbf{e}^h \| \mathbf{R}^\ell \| \mathbf{e}^t \rangle + \langle \mathbf{r}^\ell_1 \| \mathbf{e}^h \rangle + \langle \mathbf{r}^\ell_2 \| \mathbf{e}^t \rangle \tag{4.2}$$

where $\mathbf{R}^\ell \in \mathbb{R}^{\mathbf{k}\times\mathbf{k}}$, $\mathbf{r}^\ell_1 \in \mathbb{R}^k$, $\mathbf{r}^\ell_2 \in \mathbb{R}^k$, all of them depending on ℓ, $\langle .|. \rangle$ is the canonical dot product, and $\langle \mathbf{x} \| \mathbf{A} \| \mathbf{y} \rangle = \langle \mathbf{x} \| \mathbf{A}\mathbf{y} \rangle$.

If we consider our model with the squared distance as the dissimilarity function, we have:

$$d(\boldsymbol{h}+\boldsymbol{\ell},\boldsymbol{t}) = \| \mathbf{e}^h \|^2 + \| \mathbf{r}^\ell \|^2 + \| \mathbf{e}^t \|^2 - 2\big(\langle \mathbf{e}^h \| \mathbf{e}^t \rangle + \langle \mathbf{r}^\ell \| (\mathbf{e}^t - \mathbf{e}^h) \rangle\big)\,.$$

Considering our norm constraints ($\| \mathbf{e}^h \|^2 = \| \mathbf{e}^t \|^2 = 1$) and the ranking criterion (4.1), in which $\| \mathbf{r}^\ell \|^2$ does not play any role in comparing positive and negatives triples, our model thus involves scoring the triples with $\langle \mathbf{e}^h \| \mathbf{e}^t \rangle + \langle \mathbf{r}^\ell \| (\mathbf{e}^t - \mathbf{e}^h) \rangle$, and hence corresponds to the NTN model of Socher et al. (2013) (Equation (4.2)) where $\mathbf{R}^\ell$ is the identity matrix, and $\mathbf{r}^\ell = \mathbf{r}^\ell_1 = -\mathbf{r}^\ell_2$. We could not run experiments with that model, but once again our model has much fewer parameters: this should simplify the training and prevent overfitting, and may compensate for a lower expressiveness.

As a consequence of TransE, a lot of translation based models as TransH (Wang et al. 2014) and TransR (Lin et al. 2015b) were proposed later. In TransH, the embeddings of the entities h and t 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 (also relation-dependent) that is more general than an orthogonal projection to a hyperplane. Table 4.5 displays the scoring functions of the aforementioned works related to TransE.

Translations have proven a simple but accurate modeling assumption to learn these multi-relational graphs. Additionally, this simplicity in the expressiveness of the model provides a powerful regularization that helps to avoid overfitting. These translations are applied in 1-hop pieces of information (triples), but we will see later in Chapter 6 generalizations of TransE over hops bigger than 1.

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

This chapter corresponds to the papers (García-Durán et al. 2014) *Effective Blending of Two and Three-way Interactions for Modeling Multi-relational Data.* **García-Durán, A.**, Bordes A., Usunier N. In Machine Learning and Knowledge Discovery in Databases (pp. 434-449) and (García-Durán et al. 2016) *Combining Two And Three-Way Embeddings Models for Link Prediction in Knowledge Bases.* **García-Durán, A.**, Bordes A., Usunier N., Grandvalet Y. Accepted at Journal of Artificial Intelligence Research.

5.1. Introduction

This paper tackles the problem of endogenous link prediction for KB completion. KB can be represented as directed graphs whose nodes correspond to entities and edges to relationships. 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 by Bordes et al. (2013). 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 Freebase, 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 as tail but pretty much everything else as head.

Though there exists (pseudo-) symbolic approaches for link prediction based on Markov-logic networks (Kok and Domingos 2007) or random walks (Lao et al. 2011), learning latent features representations of KBs constituents - the so-called *embedding methods* - have recently proved to be an alternative for performing link prediction in KBs (Bordes et al. 2013, Wang et al. 2014, Lin et al. 2015b, Chang et al. 2014, Zhang et al. 2014, Yang et al. 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 by Jenatton et al. (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) may be 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 5.2 and 5.4.3 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.

5.2. TATEC

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

5.2.1. Scoring function

The data $\mathcal{S}$ is a set of relations between entities in a fixed set of entities in $\mathcal{E} = \{e^1, ..., e^E\}$. Relations are represented as triples (h, ℓ, t) where the head h and the tail t are indexes of entities (i.e. $h, t \in [\![E]\!] = \{1, ..., E\}$), and the label ℓ is the index of a relationship in $\mathcal{L} = \{l^1, ..., l^L\}$, which defines the type of the relation between the entities e^h and e^t. Our goal is to learn a discriminant scoring function on the set of all possible triples $\mathcal{E} \times \mathcal{L} \times \mathcal{E}$ so that the triples which represent likely relations receive higher scores than triples that represent unlikely 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:

$$s(h, \ell, t) = s_1(h, \ell, t) + s_2(h, \ell, t) \tag{5.1}$$

where s_1 and s_2 have the following form:

(B) Bigram or the 2-way interaction term:

$$s_1(h, \ell, t) = \left\langle \mathbf{r}_1^\ell \middle| \mathbf{e}_1^h \right\rangle + \left\langle \mathbf{r}_2^\ell \middle| \mathbf{e}_1^t \right\rangle + \left\langle \mathbf{e}_1^h \middle| \mathbf{D} \middle| \mathbf{e}_1^t \right\rangle, \tag{5.2}$$

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.

We use two different relation vectors for the subject and the object in order to model asymmetric relationships; for instance, if $\mathbf{r}_1^\ell = \mathbf{r}_2^\ell$, then (Paris, capital_of, France) would have the same score as (France, capital_of, Paris).

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

$$s_2(h, \ell, t) = \left\langle \mathbf{e}_2^h \middle| \mathbf{R}^\ell \middle| \mathbf{e}_2^t \right\rangle, \tag{5.3}$$

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.

5.2.2. Term combination

We study two strategies for combining the bigram and trigram scores as indicated in Equation (5.1). In both cases, both s_1 and s_2 are first trained separately as we detail in Section 5.3 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 (5.1).

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

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 (5.6)) using L-BFGS, with an additional quadratic penalization term, $\sum_\ell \frac{||\boldsymbol{\delta}^\ell||_2^2}{\sigma_\ell + \epsilon}$, where $\boldsymbol{\delta}^\ell$ contains the combination weights for relation ℓ, and σ are constrained to $\sum_\ell \sigma_\ell = \alpha$ (α is a hyperparameter). This version of TATEC is denoted TATEC-LC in the following.

5.2.3. 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.

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 (Koren et al. 2009):

$$R_{ui} \approx \langle \mathbf{P}_u | \mathbf{Q}_i \rangle + \alpha_u + \beta_i + \mu \tag{5.4}$$

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, $\alpha_u \in \mathbb{R}$

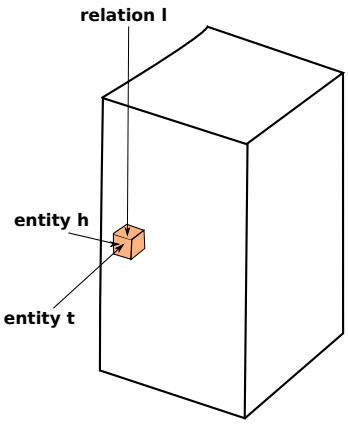


Figure 5.1.: The entry (h,l,t) of the tensor indicates if the relation l holds between the entities h and t

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 (5.4).

The bigram term **(B)** then plays the role of biases for each fiber of the tensor,[1] i.e.

$$s_1(h, \ell, t) \approx B^1_{l,h} + B^2_{l,t} + B^3_{h,t} \tag{5.5}$$

and thus is the analogue for tensors to the term $\alpha_u + \beta_i$ in the matrix factorization model (5.4). 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 $\mathbf{B}^1 = \left[B^1_{l,h}\right]_{l \in [\![L]\!], h \in [\![E]\!]}$, $\mathbf{B}^2$ and $\mathbf{B}^3$ of Equation (5.5). 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 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 $n(n+2m)$ biases, TATEC computes such biases by means of linear combinations of $n+2m$ embeddings, which allows a learning transfer across them.

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

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

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 (5.4), or the bigram term in the overall scoring function (5.1) 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 (5.4) 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 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 (5.4) are trained using a squared error as a measure of approximation and regularized by $\lambda \left(\| \mathbf{P}_u \|^2 + \| \mathbf{Q}_i \|^2 + \alpha_u^2 + \beta_i^2 \right)$, where $\lambda > 0$ is the regularization factor. This kind of regularization is a weighted trace norm regularization (Srebro and Salakhutdinov 2010) on $\mathbf{P}\mathbf{Q}^T$. Leaving aside the "weighted" part, the idea is that at convergence, the quantity $\lambda \left(\sum_u \| \mathbf{P}_u \|^2 + \sum_i \| \mathbf{Q}_i \|^2 \right)$ is equal to 2λ times the sum of the singular values of the matrix $\mathbf{P}\mathbf{Q}^T$. However, $\lambda \| \boldsymbol{\alpha} \|^2$, 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 $\sqrt{I} \| \boldsymbol{\alpha} \|$, and can be much larger than $\| \boldsymbol{\alpha} \|^2$. 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.

2-way interactions as entity types+similarity

Having a part of the model that is less expressive, but less regularized (see Subsection 5.3.2) 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 Freebase in particular, are strongly typed, in the sense that only well-defined and specific subsets of entities can be either heads or tails of selected relationships. For instance, a relationship like `capital_of` expects a (big) city as head and a country as tail for any valid relation. Large knowledge bases have huge amounts of entities, but those belong to many different types. Identifying the expected types of head and tail entities of relationships, with an appropriate granularity of types (e.g. `person` or `artist` or `writer`), is likely to filter out 95% of the entity set during prediction. The exact form of the first two terms $\langle \mathbf{r}_1^\ell | \mathbf{e}_1^h \rangle + \langle \mathbf{r}_2^\ell | \mathbf{e}_1^t \rangle$ of the 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 $\mathbf{D}$.[2] The reason for using a symmetric matrix comes from the intuition that the direction of many relationships is arbitrary (i.e. the choice between having triples "Paris is capital of France" rather than "France has capital Paris"), and the model should be invariant under arbitrary inversions of the directions of the relationships (in the case of an inversion of direction, the relations vectors $\mathbf{r}_1^\ell$ and $\mathbf{r}_2^\ell$ are swapped, but all other parameters are unaffected). For tasks in which such invariance is not desirable, the diagonal matrix could be replaced by an arbitrary matrix.

5.3. Training

5.3.1. 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

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

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

$$\sum_{(h,\ell,t)\in\mathcal{S}} \sum_{(h',\ell',t')\in\mathcal{C}(h,\ell,t)} \left[\gamma - s(h,\ell,t) + s(h',\ell',t')\right]_+ \tag{5.6}$$

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 (see Section 3.2). 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 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 procedure that is summarized in Algorithm 2, for the unregularized case. Training of the linear combination weights of TATEC-LC is stopped at convergence of L-BFGS.

5.3.2. 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. 2014, Lin et al. 2015b) to the objective function (5.6). 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 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

Algorithm 2 Learning unregularized TATEC.

Input Training set $S = \{(h, l, t)\}$, margin γ, learning rates λ_1 and λ_2
initialization
  - for BIGRAMS: $\mathbf{e}_1 \leftarrow \text{uniform}(-\frac{6}{\sqrt{d_1}}, \frac{6}{\sqrt{d_1}})$ for each entity e
    $\mathbf{r}_1, \mathbf{r}_2 \leftarrow \text{uniform}(-\frac{6}{\sqrt{d_1}}, \frac{6}{\sqrt{d_1}})$ for each ℓ
    $\mathbf{D} \leftarrow \text{uniform}(-\frac{6}{\sqrt{d_1}}, \frac{6}{\sqrt{d_1}})$
  - for TRIGRAM: $\mathbf{e}_2 \leftarrow \text{uniform}(-\frac{6}{\sqrt{d_2}}, \frac{6}{\sqrt{d_2}})$ for each entity e
    $\mathbf{R} \leftarrow \text{uniform}(-\frac{6}{\sqrt{d_2}}, \frac{6}{\sqrt{d_2}})$ for each ℓ
  - 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.
while some condition **do**
  $S_{batch} \leftarrow sample(S, m)$ // 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 $\sum_{((h,\ell,t),(h',\ell',t')) \in T_{batch}} \nabla^2 \left[\gamma - s(h, \ell, t) + s(h', \ell', t')\right]_+$:
    - for BIGRAMS (Eq. 5.2): $s = s_1$
    - for TRIGRAM (Eq. 5.3): $s = s_2$
    - for TATEC-FT (Eq. 5.1): $s = s_1 + s_2$
end while

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 (5.6) . As a result, in the *soft* scheme the following regularization term is added to the loss function (5.6): $C_1[||\mathbf{e}_1||_2^2 - \rho_e^2]_+ + C_2\big([||\mathbf{e}_2||_2^2 - \rho_e^2]_+ + [||\mathbf{R}||_F^2 - \rho_l^2]_+\big)$, where C_1 and C_2 are hyperparameters that weight the importance of 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:

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

[2]For simplicity, we use the symbol ∇ to refer to the gradient w.r.t. the parameters of the model

And in the soft regularization case it is:

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

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.

5.4. 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 (see Section 3.1 for more details on these data sets). All versions of TATEC and of its components BIGRAMS and TRIGRAM are compared with the state-of-the-art models for each database.

5.4.1. Experimental setting

This section details the protocols used in our various experiments.

Datasets and metrics

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

UMLS/Kinships 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.[3] These negative triples correspond to the first setting of negative examples of Section 3.2. 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 We used a ranking metric evaluated in both *raw* and *filtered* setting, following the first setting of Section 3.2 to generate negative triples. 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,

[3] This replication process is carried out only in training.

we take the *hits*@10 as secondary validation criterion.[4] 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 For this database we evaluate our models in the verb prediction task. As for FB15k, two ranking metrics are computed, the *mean rank* and the *hits@5%* (the 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 3.2. 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.

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 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.

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 $\sigma_\ell^* = \frac{\alpha||\boldsymbol{\delta}^\ell||_2}{\sum_k ||\boldsymbol{\delta}^k||_2}$, until some stopping criterion is reached. The δ parameters are initialized to 1 and the α value is validated among $\{0.1, 1, 10, 50, 100, 200, 500, 1000\}$.

[4] 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. Results on FB15k with TransE are also provided in Bordes et al. (2013) (see Chapter 4), but again the hyperparameters were validated on a not wide enough grid search. Specifically, the margin and the embedding dimension were validated in not good enough ranges.

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 $s_1(h, \ell, t)$ and $s_2(h, \ell, t)$.
- TATEC-FT-SHARED: TATEC-FT but sharing the entities embeddings between $s_1(h, \ell, t)$ and $s_2(h, \ell, t)$ 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. 2015b) 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.

5.4.2. 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.

UMLS and Kinships

The results for these two knowledge bases are provided in Table 5.1. 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

Table 5.1.: **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

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. 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 degrade 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.

Table 5.2.: **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

FB15k

Table 5.2 (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 et al. (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

Table 5.3.: **Test results on FB15k.** Proportion of entities ranked in the Top 1.

Model	Hits@1
TransE-soft	28.1
Bigrams-soft	27.2
Trigram-soft	24.9
Tatec-ft-soft	**37.8**

Table 5.4.: **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 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 5.3 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 5.4. 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.

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

Table 5.5.: **Relative training times with respect to TransE on FB15k** for running one epoch on a single core

Model	Relative train. time
Bigrams-soft	× 1.4
Trigram-soft	× 3.6
Tatec-ft-soft	× 4.0

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 5.5 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;
- parallellize training with Hogwild (Recht et al. 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.

5.4.3. Illustrative experiments

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

Table 5.6.: **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

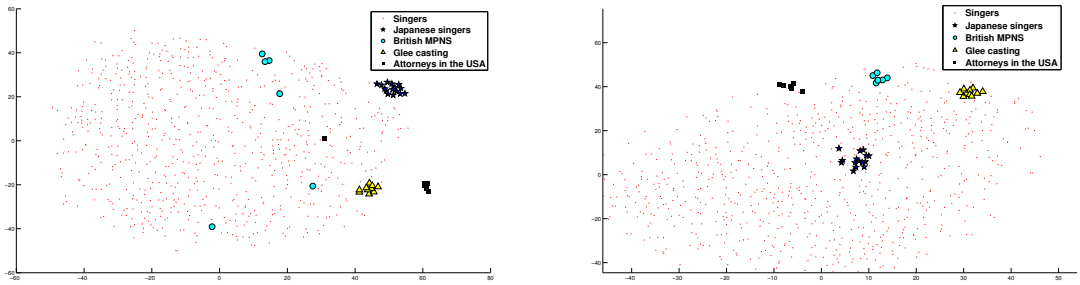
TransE and symmetrical relationships

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 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. 2013), the algorithm is shown to perform well on FB15k and on a dataset extracted from the KB WordNet: 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.

Anecdotal examples

Some examples of predictions by TATEC on FB15k are displayed in Table 5.6. 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



(a) Embeddings of TRIGRAM (b) Embeddings of BIGRAMS

Figure 5.2.: **Embeddings obtained by Trigram and Bigrams models** and projected in 2-D using t-SNE. MPNS stands for `Main Profession is Not Singer`.

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 5.2a and 5.2b 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 and 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 5.2a that all attorneys are clustered and separated from the singers, except one, which corresponds to the multifaceted `Fred Thompson`[5]. 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 separate small cluster made up of Japanese singers that can be seen in Figure 5.2a (TRIGRAM). In Figure 5.2b (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 5.2b, we find entities highly connected to `japan` like `yoko_ono` – born in Japan, `vic_mignogna`, `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

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

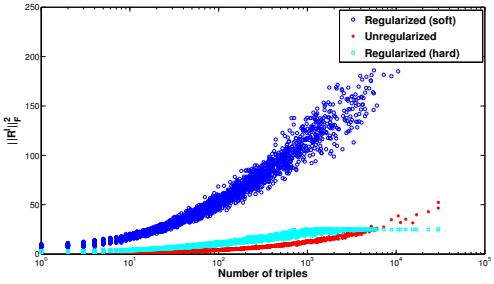
Table 5.7.: **Examples of predictions on SVO.** Given two nouns acting as subject and direct object from a test triple, TATEC predicts the best fitting verb. In bold is the expected correct answer.

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

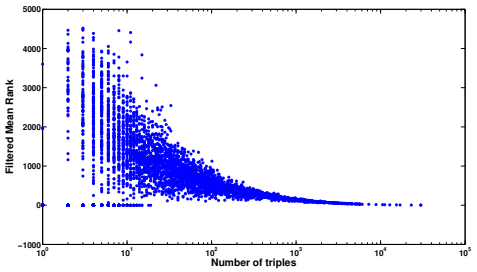
heads and tails in the BIGRAMS model: it tends to push together entities connected in triples whatever the relation. In this case, this forms a Japanese cluster.

Table 5.7 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 5.3a, 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. We can see in Figure 5.3a 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



(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 5.3.: Indicators of the behavior of TATEC-FT on FB15k according to to the number of training triples of each relationship.

Table 5.8.: **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

could down-weight the importance of the verb frequency by tuning the parameters ρ_l and C_2 to enforce a stronger constraint. Figure 5.8 shows the effect of the verb frequency in these two models when predicting the same missing verb as in Table 5.7.

Breaking down the performance by relationship, this is translated into a strong relation between the performance of a relationship and its frequency (see Figure 5.3b). 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.

5.5. Conclusion

This paper presents TATEC, a tensor factorization method that satisfactorily combines 2- and 3-way interaction terms to obtain a performance above the best of either constituent. Different data patterns are properly encoded thanks to the use of differ-

ent 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 5.6, 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.

5.6. Discussion on related works

The Latent Factor Model (LFM) (Jenatton et al. 2012) and the Neural Tensor Networks (NTN) (Socher et al. 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 5.2.3, 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 show in Table 5.9 the formulas of the scoring functions of these related works.

Specifically, the 2-way interaction terms of the model is similar to that of Bordes et al. (2014a) -SME(linear)-, but slightly more general because it does not contain any constraint of 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. (2013) -TransE-, 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, 3-way models can basically represent any kind of interaction among entities. In LFM (Jenatton et al. 2012), constraints were imposed on the relation-dependent matrix of the 3-way terms (low rank in a limited

Table 5.9.: **Scoring function for several models related to Tatec.** Capitalized letters denote matrices and lower cased ones, vectors.

Model	Score ($s(h, \ell, t)$)
Bigrams	$\langle \mathbf{r}_1^\ell \vert \mathbf{e}_1^h \rangle + \langle \mathbf{r}_2^\ell \vert \mathbf{e}_1^t \rangle + \langle \mathbf{e}_1^h \vert \mathbf{D} \vert \mathbf{e}_1^t \rangle$
Trigram	$\langle \mathbf{e}_2^h \vert \mathbf{R}^\ell \vert \mathbf{e}_2^t \rangle$
TransE	$\vert\vert \mathbf{e}^h + \mathbf{r}^\ell - \mathbf{e}^t \vert\vert_2$
RESCAL	$\langle \mathbf{e}^h \vert \mathbf{R}^\ell \vert \mathbf{e}^t \rangle$
LFM	$\langle y \vert \mathbf{R}^\ell \vert y' \rangle + \langle \mathbf{e}^h \vert \mathbf{R}^\ell \vert \mathbf{z} \rangle + \langle \mathbf{z} \vert \mathbf{R}^\ell \vert \mathbf{e}^t \rangle + \langle \mathbf{e}^h \vert \mathbf{R}^\ell \vert \mathbf{e}^t \rangle$
SME(linear)	$(\langle \mathbf{W}_\mathbf{1}^h \vert \mathbf{e}^h \rangle + \langle \mathbf{W}_\mathbf{2}^h \vert \mathbf{r}^\ell \rangle)(\langle \mathbf{W}_\mathbf{1}^t \vert \mathbf{e}^t \rangle + \langle \mathbf{W}_\mathbf{2}^t \vert \mathbf{r}^\ell \rangle)$
NTN	$\langle u^\ell \vert f(\langle \mathbf{e}^t \vert \mathbf{W}^{\ell\ [\mathbf{1:k}]} \vert \mathbf{e}^t \rangle + \langle \mathbf{V}^\ell \vert \begin{pmatrix} \mathbf{e}^h \\ \mathbf{e}^t \end{pmatrix} + \mathbf{b}^\ell \rangle) \rangle$

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).

In the same spirit, Nickel et al. (2015) try to combine the expressive power of the tensor product (3-way model) with the efficiency and simplicity of TransE (2-way model) by using the circular correlation of the embeddings that represent a pair of entities. This operator composes a new embedding for a pair of entities, where each component of this new embedding corresponds to the sum of pairwise interactions of the embedding features of the entities, keeping the memory complexity of TransE.

6. Composing Relationships with Translations

This chapter corresponds to the paper (García-Durán et al. 2015) *Composing Relationships with Translations*. **García-Durán A.**, Bordes A., Usunier N. In Empirical Methods on Natural Language Processing (pp. 286-290).

6.1. Introduction

Performing link prediction on multi-relational data is becoming increasingly important in order to complete the huge amount of missing information of the knowledge bases. This knowledge can be formalized as directed multi-relation graphs, whose node correspond to entities connected with edges encoding various kind of relationships. We denote these connections via triples *(head, label, tail)*. Link prediction consists in filling in incomplete triples like *(head, label, ?)* or *(?, label, tail)*.

In this context, embedding models that attempts to learn low-dimensional vector or matrix representations of entities and relationships have shown promising performance in recent years (Wang et al. 2014, Lin et al. 2015b, Jenatton et al. 2012, Socher et al. 2013). In particular, the basic model TransE (Bordes et al. 2013) has been proved to be very powerful. This model treats each relationship as a translation vector operating on the embedding representing the entities. Hence, for a triple *(head, label, tail)*, the vector embeddings of *head* and *tail* are learned so that they are connected through a translation parameterized by the vector associated with *label*. Many extensions have been proposed to improve the representation power of TransE while still keeping its simplicity, by adding some projections steps before the translation (Wang et al. 2014, Lin et al. 2015b).

In this paper, we proposed an extension of TransE that focuses on improving its representation of the underlying graph of multi-relational data by trying to learn compositions of relationships as sequences of translations in the embedding space. The idea is to train the embeddings by learning simple reasonings, such as the relationship `people/nationality` should give a similar result as the composition `people/city_of_birth` and `city/country`. In our approach, called rTransE, the training set is augmented with relevant examples of such compositions, and training so that sequences of translations lead to the desired result.

The idea of compositionality to model multi-relational data was previously introduced by Neelakantan et al. (2015). That work composes relationships by means of recurrent neural networks (RNN) (one per relationship) with non-linearities. However, we show

that there is a natural way to compose relationships by simply adding translation vectors and not requiring additional parameters, which makes it specially appealing because of its scalability.

We present experimental results that show the superiority of rTransE over TransE in terms of link prediction. A detailed evaluation, in which test examples are classified as *easy* or *hard* depending on their similarity with training data, highlights the improvement of rTransE on both categories. Our experiments include a new evaluation protocol, in which the model is directly asked to answer questions related to compositions of relations, such as *(head, label$_1$, label$_2$, ?)*. rTransE also achieves significantly better performances than TransE on this new dataset.

We describe rTransE in the next section, and present our experiments in Section 6.3.

6.2. Model

The model we propose is inspired by TransE (Bordes et al. 2013). In TransE, entities and relationships of a KB are mapped to low dimensional vectors, called embeddings. These embeddings are learnt so that for each fact (h, ℓ, t) in the KB, we have $\mathbf{e}^h + \mathbf{r}^\ell \approx \mathbf{e}^t$ in the embedding space.

Using translations for relationships naturally leads to embed the composition of two relationships as the sum of their embeddings: on a path $(h, \ell, t), (t, \ell', t')$, we should have $\mathbf{e}^h + \mathbf{r}^\ell + \mathbf{r}^{\ell'} \approx \mathbf{e}^{t'}$ in the embedding space. The original TransE does not enforce that the embeddings accurately reproduce such compositions. The *recurrent* TransE we propose here has a modified training stage to include such compositions. This should allow to model simple reasonings in the KB, such as `people/nationality` is similar to the composition of `people/city_of_birth` and `city/country`.

See Chapter 4 for a more detailed explanation of TransE.

6.2.1. Recurrent TransE

We describe in this section our model in its full generality, which allows to deal with compositions of an arbitrary number of relationships, even though in this first work we experimented only with compositions of two relationships.

Triples that are the result of a composition are denoted by $(h, \{\ell_i\}_{i=1}^p, t)$, where p is the number of relationships that are composed to go from h to t. Such a path means that there exist entities $e_1, ..., e_{p+1}$, with $e_1 = h$ and $e_{p+1} = t$ such that for all k, (e_k, ℓ_k, e_{k+1}) is a fact in the KB. Our model, rTransE for *recurrent* TransE, represents each step $s_k(h, \{\ell_i\}_{i=1}^p, t)$ along the path in the KB with the recurrence relationship (boldface characters denote embedding vectors):

$$\begin{aligned} \boldsymbol{s}_0(h, \{\ell_i\}_{i=1}^p, t) &= \mathbf{e}^h \\ \boldsymbol{s}_k(h, \{\ell_i\}_{i=1}^p, t) &= \boldsymbol{s}_{k-1}(h, \{\ell_i\}_{i=1}^p, t) + \mathbf{r}^{\ell_k} . \end{aligned}$$

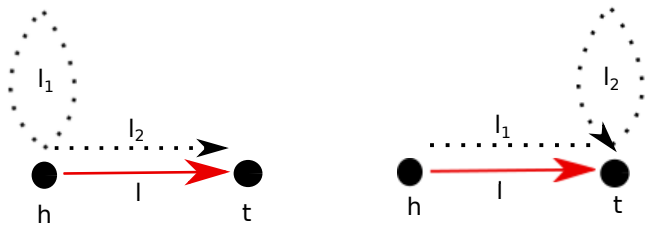


Figure 6.1.: Some of the paths filtered out to train rTransE

Then, the energy of a triple is computed as

$$d(h, \{\ell_i\}_{i=1}^p, t) = ||\boldsymbol{s}_p(h, \{\ell_i\}_{i=1}^p, t) - \mathbf{e}^t||_2 \,.$$

6.2.2. Path construction and filtering

The goal of the paper is motivated by learning simple reasonings in the KB through the compositions of relationships. Therefore, we restricted our analysis to paths of length 2 created as follows.

First, for each fact (h, ℓ, t), retrieve all paths $(h, \{\ell_1, \ell_2\}, t)$ such that there is e such that both (h, ℓ_1, e) and (e, ℓ_2, t) are in the KB. Then, we filter out paths where $(h, \ell_1, e) = (h, \ell, t)$ or $(e, \ell_2, t) = (h, \ell, t)$ (cases displayed in Figure 6.1), as well as the paths with $\ell_1 = \ell_2$ and $h = e = t$ (loops iterating over itself).

We focused on "unambiguous" paths, so that the reasoning might actually make sense. In particular, we considered only paths where ℓ_1 is either a 1-TO-1 or a 1-TO-MANY relationship, and where ℓ_2 is either a 1-TO-1 or a MANY-TO-1 relationship. In our experiments, the paths created for training only consider the training subset of facts.

In the remainder of the paper, such paths of length 2 are called *quadruples*.

6.2.3. Training and regularization

Our training objective is decomposed in two parts: the first one is the ranking criterion on triples of TRANSE, ignoring quadruples. Paths are then taken into account through additional regularization terms.

Denoting by $\mathcal{S}$ the set of facts in the KB, the first part of the training objective is the following ranking criterion that operates on triples

$$\sum_{\substack{(h,\ell,t)\in\mathcal{S} \\ (h',\ell,t')\in\mathcal{S}_{(h,\ell,t)}}} \left[\gamma + d(h,\ell,t) - d(h',\ell,t')\right]_+ ,$$

where $[x]_+ = \max(x, 0)$ is the positive part of x, γ is a margin hyperparameter and $\mathcal{S}_{(h,\ell,t)}$ is the set of corrupted triples created from (h, ℓ, t) by replacing either h or t with another KB entity (first setting of Section 3.2).

This ranking loss effectively trains so that the embedding of the tail is the nearest neighbor of the translated head, but it does not guarantee that the distance between the

tail and the translated head is small. The nearest neighbor criterion is sufficient to make inference over simple triples, but making sure that the distance is small is necessary for the composition rule to be accurate. In order to account for the compositionality of relationships, we add two additional regularization terms:

- $\lambda \sum_{(h,\ell,t)\in\mathcal{S}} d(h,\ell,t)^2$
- $\alpha \sum_{(h,\{\ell_1,\ell_2\},t)\in\mathcal{S}} N_{\ell\rightarrow\{\ell_1,\ell_2\}} d(h,\{\ell_1,\ell_2\},t)^2$.

The first criterion only applies to original facts of the KB, while the second term applies to quadruples. $N_{\ell\rightarrow\{\ell_1,\ell_2\}}$, which involves both the relationships of the quadruple and the relationship ℓ from which it was created, is the number of paths involving relationships $\{\ell_1,\ell_2\}$ created from a fact involving ℓ, normalized by the total number of quadruples created from facts involving ℓ. This criterion puts more weight on paths that are reliable as an alternative for a relationship, for instance {`people/city_of_birth`, `city/country`} is likely a better alternative to `people/nationality` than {`people/writer_of_the_film`, `film/film_release_region`}. Finally, a regularization term $\mu||\mathbf{e}||_2^2$ is added for each entity embedding.

6.3. Experiments

This section presents experiments on the benchmarks FB15K, introduced in Chapter 3, and FAMILY (described below), which is a good fit given the the compositional nature of its relationships. Statistics for these datasets for both tasks (link prediction on triples and quadruples) are given in Table 6.1.

Inspired by Hinton (1986), FAMILY is a database that contains triples expressing family relationships (`cousin_of`, `has_ancestor`, `married_to`, `parent_of`, `related_to`, `sibling_of`, `uncle_of`) among the members of 5 families along 6 generations. This dataset is artificial and each family is organized in a layered tree structure where each layer refers to a generation. Families are connected among them by marriage links between two members, randomly sampled from the same layer of different families. Interestingly on this dataset, there are obvious compositional relationships like `uncle_of` $\approx$ `sibling_of + parent_of` or *parent_of* $\approx$ *married_to* + *parent_of*, among others. We are the creators of this dataset.[1]

6.3.1. Experimental Protocol

Setting We followed the same experimental setting as in Chapter 4 and 5, using ranking metrics for evaluation for both FB15k and FAMILY.

The embedding dimensions were set to 20 for FAMILY and 100 for FB15K. Training was performed by stochastic gradient descent, stopping after for 500 epochs. On FB15K, we used the embeddings of TRANSE to initialize RTRANSE, and we set a learning rate of 0.001 to fine-tune RTRANSE. On FAMILY, both algorithms were initialized randomly

[1] https://everest.hds.utc.fr/doku.php?id=en:2and3ways

Data set	Family	FB15k
Entities	721	14,951
Relationships	7	1,345
Training triples	8,461	483,142
Training quad.	–	30,252
Validation triples	2,820	50,000
Test triples	2,821	59,071
Test quad.	–	1,852

Table 6.1.: Statistics of the datasets Family and FB15k: triples and quadruples

Model	TransE		rTransE	
	MR	H@10	MR	H@10
Easy	17.7	76.8	**12.5**	**82.2**
Hard	**191.0**	48.9	205.7	**51.0**
Easy w. Comp.	16.4	78.8	**11.6**	**83.0**
Easy w/o Comp.	21.6	71.3	**16.0**	**75.3**
Hard w. Comp.	**208.1**	46.8	212.2	**49.3**
Hard w/o Comp.	**122.9**	**57.0**	123.8	**57.5**
Overall	50.7	71.5	**49.5**	**76.2**

Table 6.2.: **Detailed performances on FB15k** of TransE and rTransE. H@10 are in %. w. Comp. indicates examples for which there exist quadruplets in train matching their relationship.

and used a learning rate of 0.01. The mean rank was used as a validation criterion, and the values of γ, λ, α and μ were chosen respectively among $\{0.25, 0.5, 1\}$, $\{1e^{-4}, 1e^{-5}, 0\}$, $\{0.1, 0.05, 0.1, 0.01, 0.005\}$ and $\{1e^{-4}, 1e^{-5}, 0\}$.

6.3.2. Results on triples

Overall performances Experiments on Family show a quantitative improvement of the performance of rTransE : where TransE gets a mean rank of 6.7 and a H@5 of 68.7, rTransE gets a performance of 6.3 and 72.3 respectively.

Similarly, on FB15k, Table 6.2 (last row) shows that training on longer paths (length 2 here) actually consistently improves the performance while predicting heads and tails of triples only: the overall H@10 improves by almost 5% from 71.5 for TransE to 76.2 for rTransE.

Detailed results In order to better understand the gains of rTransE, we performed a detailed evaluation on FB15k, by classifying the test triples along two axes: *easy* vs *hard* and *with composition* vs *without composition*. A test triple (h, ℓ, t) is *easy* if its head and tail are connected by a triple in the training set, i.e. if either (h, ℓ', t) or (t, ℓ', h) is seen in train for some relationship ℓ'. Otherwise, the triple is *hard*. Orthogonally, the test triple (h, ℓ, t) is *with composition* if there is a path $(h, \{\ell_1, \ell_2\}, t)$ that can be

	3 Nearest entities to $h + \ell_1 + \ell_2$	
	rTransE	TransE
h: madtv	**U.S.A.**	Ireland
ℓ_1:regular TV app.	Ireland	**U.S.A.**
ℓ_2: nationality	Japan	U.K.
h: stargate atlantis	Hawaii	Scotland
ℓ_1:regular TV app.	Scotland	Hawaii
ℓ_2:nationality	**U.S.A.**	U.K.
h: malay	southeast asia	taiwan
ℓ_1: language/main_country	malaysia	southeast asia
ℓ_2: continent	**asia**	philippines
h: indiana_state_university	**the_hoosier_state**	maryland
ℓ_1: institution/campuses	terre_haute	rhode_island
ℓ_2: location/state_province_region	rhode_island	the_constitution_state
h: university_of_victoria	**victoria**	kelowna
ℓ_1: institution/campuses	kurnaby	toronto
ℓ_2: location/citytown_province_region	kelowna	ottawa
h: law&order	**dick wolf**	michael crichton
ℓ_1: spin off	walon green	renny harlin
ℓ_2: program creator	ken burns	paul schrader

Table 6.3.: **Examples of predictions on quadruples** of TransE and rTransE

constructed from the training set (notice that $(h, \{\ell_1, \ell_2\}, t)$ will usually not be used for training because training paths are built upon training triples). If no such path exists, (h, ℓ, t) is *without composition*.

The detailed results are shown in Table 6.2. We can see that comparatively to TransE, rTansE particularly improves performances in terms of H@10 on triples *with composition*, improving on *easy* triples by 4.2% (from 78.8% to 83.0%) and *hard* triples by 2.5% (from 46.8% to 49.3%). The main gains are still on *easy* triples, and in fact the H@10 on *easy* triples *without composition* increases by 4%, from 71.3% to 75.3%. The mean rank also considerably improves on *easy triples*, and stays somehow still on *hard* ones. All in all, the results show that considering paths during training very significantly improves performances, and the results on triples *with composition* suggest that rTransE is indeed capable of capturing the evidence of links that exist in longer paths.

6.3.3. Results on quadruples

While usual evaluations for link prediction in KBs focus on predicting a missing element of a test triple, we propose here to extend the evaluation to answering more complex questions, such as $(h, \{\ell_1, \ell_2\}, ?)$ or $(?, \{\ell_1, \ell_2\}, t)$.

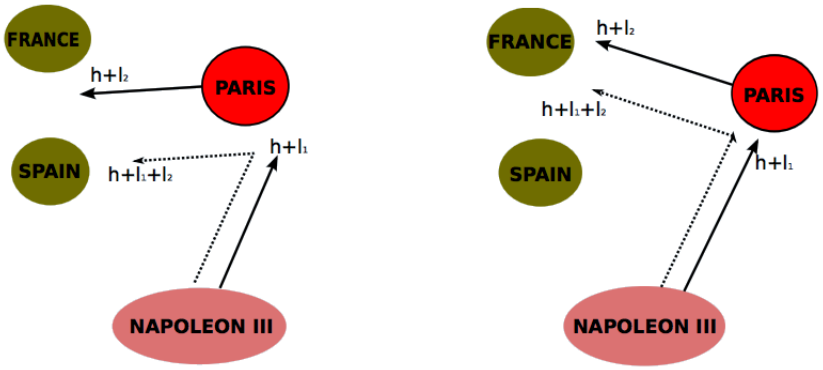


Figure 6.2.: TransE (left) *vs* rTransE(right)

Examples Table 6.3 presents examples of predictions of both TransE and rTransE on such quadruples.The two first examples try to predict the origin of two TV series from the nationality of the actors that regularly appear in them (`regular_tv_appearance`). In the first one, the american actor phil lamarr is the only entity connected to the american TV show madtv through the relationship regular_tv_appearance. rTransE is able to correctly infer the country of origin from this information since it forces `country_of_origin` $\approx$ `regular_tv_appearance + nationality`. On the other side TransE is affected by the cascading error since the ranking loss does not guarantee that the distance between $\mathbf{e}^h + \mathbf{r}^{\ell_1}$ and phil lamarr is small, so when summing $\mathbf{r}^{\ell_2}$ it eventually ends up closer to Ireland rather than USA. In contrast, the second example shows that answering that question by using that path is sometimes difficult: the members of the cast of that TV show have different nationalities, so rTransE lists the nationalities of these ones and the correct one is ranked third. TransE is again more affected than rTransE by the cascading error. In the third one, rTransE deducts the main region where malay is spoken from the continent of the country with the most number of speakers of that language. In the two following examples, rTransE infers the location of those universities by forcing an equivalence between their location and the location of their respective campus. Lastly, the producer of the TV series law&order is inferred from the program creator of the spin off of that TV show.

Figure 6.2 illustrates a graphical comparison of TransE against rTranse. rTranse reduces the cascading error at composing relationships. In that example, the relationship `people/nationality` is composed of `people/city_of_birth` and `city/country` and consequently rTranse correctly predicts France as the nationality of Napoleon III.

Prediction performance For a more quantitative analysis, we have generated new test data of link prediction on quadruples on FB15k. This test set was created by generating

the paths the usual test set (the triple test set) and removing those quadruples that are used for training. We obtain 1,852 quadruples. The overall experimental protocol is the same as before, trying to predict the head or tail of these quadruple in turn.

On that evaluation protocol, rTransE has a mean rank of 114.0 and a H@10 of 68.2%, while TransE obtains a mean rank of 159.9 and a H@10 of 65.2% (using the same models as in the previous subsection). We can see that learning on paths improves performances on both metrics, with a gain of 3% in terms of H@10 and an important gain of about 46 in mean rank, which corresponds to a relative improvement of about 30%.

6.4. Discussion on related works

Other than the obvious relatedness of rTransE with TransE and the modifications that came out of it (Wang et al. 2014, Lin et al. 2015b), simultaneously two works (Gu et al. 2015, Lin et al. 2015a) with the same spirit were presented in the same conference as rTransE.

pTransE (Lin et al. 2015a) handles composition through a ranking loss criterion, as opposed to rTransE that does it through regularization, in such a way that the relationship ℓ of the fact (h,ℓ,t) is formalized as $\ell_1 + \ell_2 + \cdots + \ell_n$, expressing that that path exists between these two entities h and t. Similarly to rTransE, pTransE also accounts for the fact that not all the relation paths are equally meaningful and reliable, however a big difference between these two models is that while in rTransE this reliability factor aims at scoring how good a path is at replacing a specific relation, in pTransE that factor measures the reliability of that path as a meaningful connection between the *head* and *tail* regardless the relationship is replacing. This work constrained itself to 2- and 3-hop paths, obtaining a filtered mean rank and hits@10 of 58 and 84.6, respectively, in FB15k.

Gu et al. (2015) propose a generic training procedure for any *composable* model. Instead of setting constraints as the initial and end node of the path at searching for them, it generates training examples just by performing random walks in the training graph. Then these generated examples $(h, \{ \ell_1, \ldots, \ell_n \}, t)$ can be applied to any *composable* model as TransE. Though this model lacks a proper reliability factor for the generated paths, note that these paths are randomly sampled from the training graph and this is actually a methodology that guarantees reliable paths have more weight during training, i.e. a reliable path is sampled more frequent than a not reliable one.

As mentioned in Section 6.1, the idea of compositionality for KB inference was introduced by Neelakantan et al. (2015). There, the vector representations of the paths (of any length) in the KB graph are computed by applying the composition function of the RNN recursively. Nevertheless, its applicability to real data is limited since they learn a separate composition matrix for every relation that is predicted. Applying a general composition matrix for all the relationships was also investigated by Lin et al. (2015a), proving an inferior performance with respect to that of pTransE, which seems to

confirm translations as a powerful modeling assumption for these multi-relational graphs.

7. Generating Factoid Questions With Recurrent Neural Networks

This is a novel application on how to make use of the vast amount information that KBs provide. We use the representations learned by TransE in Freebase as input to a machine translation model to generate questions in english language.

This chapter corresponds to the paper (Serban et al. 2016) *The 30M Factoid Question-Answer Corpus: Generating Factoid Questions with Recurrent Neural Networks.* Serban, I., **García-Durán, A.**, Gulcehre, C., Ahn, S., Chandar, S., Courville, A., Bengio, Y. Submitted to NAACL HLT 2016. Both Serban, I. and **García-Durán, A.** are the first authors.

7.1. Introduction

Large-scale supervised learning corpora have recently enabled machine learning researchers to make substantial advances in applications, ranging from automatic speech recognition, machine translation, language modeling, to object classification and image caption generation tasks (Hinton et al. 2012, Goodfellow et al. 2015, Mikolov et al. 2010, Auli et al. 2013, Sutskever et al. 2014, Xu et al. 2015, Kiros et al. 2014). Many of these successes are based on neural networks, and similar approaches are now being pursued for building information retrieval systems (Huang et al. 2013, Sordoni et al. 2015) and dialogue systems (Lowe et al. 2015).

A major obstacle for training end-to-end neural network based question-answering systems was the lack of labeled data. As a result, the question answering field mainly focused on building question-answering (QA) systems based on traditional information retrieval procedures (Lopez et al. 2011, Dumais et al. 2002, Voorhees and Tice 2000). More recently, researchers have started to utilize large-scale knowledge bases (KBs) (Lopez et al. 2011), such as Freebase (Bollacker et al. 2008), WikiData (Vrandečić and Krötzsch 2014) and Cyc (Lenat and Guha 1989). In order to make progress in spite of the lack of annotated QA pairs, researchers mainly have relied on hand-crafted rules and artificially synthesized QA corpora (Bordes et al. 2014b; 2015).

In this paper we focus on generating questions based on the Freebase KB. We frame question generation as a translation problem, starting from a Freebase fact represented by a triple consisting of a head, a label and a tail. Triples can be translated into a question about the head, where the tail is the correct answer (Bordes et al. 2015). We experimented with several models inspired by recent neural machine translation models

(Cho et al. 2014a, Sutskever et al. 2014, Bahdanau et al. 2015) and we used a methodology similar to Luong et al. (2015) to deal with the problem of rare words. We evaluate the produced questions with respect to automatic evaluation metrics, including BLEU, METEOR and word-embedding based evaluation metrics, and a human experiment. We find that our question-generation model outperforms the competing template-based baseline, and, when presented to untrained human evaluators, the produced questions appear to be indistinguishable from real human-generated questions. This suggests that the produced question-answer pairs will be very useful for training QA systems. Finally, we use our best performing model to construct a new factoid question answer corpus – The 30M Factoid Question-Answer Corpus – which is made freely available to the research community[1].

7.2. Task Definition

7.2.1. Knowledge Bases

In general, a KB can be viewed as a multi-relational graph, which consists of a set of nodes (entities) and a set of edges (relationships) linking nodes. In Freebase (Bollacker et al. 2008) these relationships are directed and always connect exactly two entities. For example, in Freebase the two entities `fires_creek` and `nantahala_national_forest` are linked together by the relationship `contained_by`. Since the triple (`fires_creek`, `contained_by`, `nantahala_national_forest`) represents a complete and self-contained piece of information, it is also called a *fact* where `fires_creek` is the head of the edge, `contained_by` is the relationship and `nantahala_national_forest` is the tail of the edge.

7.2.2. Translating Facts to Questions

We aim to translate a fact into a question, such that:

1. The question is concerned with the head and relationship of the fact, and
2. The tail of the fact represents a valid answer to the generated question.

We model this in a probabilistic framework as a directed graphical model:

$$P(Q|F) = \prod_{n=1}^{N} P(w_n|w_{<n}, F), \tag{7.1}$$

where $F = (head, label, tail)$ represents the fact, $Q = (w_1, \ldots, w_N)$ represents the question as a sequence of tokens $w_1, \ldots, w_N$, and $w_{<n}$ represents all the tokens generated before token w_n. In particular, w_N represents the question mark symbol '?'.

[1] The corpus will be made available very soon.

Questions	Entities	Relationships	Words
108,442	131,684	1,837	∼77k

Table 7.1.: Statistics of SimpleQuestions

7.2.3. Dataset

We use the SimpleQuestions dataset of Bordes et al. (2015) to train our models. This is by far the largest dataset of question-answer pairs created by humans based on a KB. It contains over 100K question-answer pairs created by annotators on Amazon Mechanical Turk[2] in English based on the Freebase KB. In order to create the questions, human participants were shown one whole Freebase fact at a time and they were asked to phrase a question such that the head of the presented fact becomes the answer of the question.[3] Consequently, both the head and the label are explicitly given in each question. But indirectly characteristics of the tail may also be given since the humans have an access to it as well. Often when phrasing a question the annotators tend to be more informative about the target tail by giving specific information about it in the question produced. For example, the question *What city is the American actress X from?* informs that the tail was born in America - information, which was not provided by either the head or label of the fact. We have also observed that the questions are mostly ambiguous: that is, one can easily come up with several possible answers that may fit the specifications of the question. Table 7.1 shows statistics of the dataset.

7.3. Model

We propose to attack the problem with the models inspired by the recent success of neural machine translation models (Sutskever et al. 2014, Bahdanau et al. 2015). Intuitively, one can think of this translation task as a "lossy translation" from structured knowledge (facts) to human language (questions in natural language), where certain aspects of the structured knowledge is intentionally left out (e.g. the name of the tail). These models typically consist of two components: an encoder, which encodes the source phrase into one or several fixed-size vectors, and a decoder, which decodes the target phrase based on the results of the encoder.

7.3.1. Encoder

In contrast to the neural machine translation framework, our source language is not a proper language but instead a sequence of three variables making up a fact. We propose an encoder sub-model, which encodes each atom of the fact into an embedding. Each atom $\{h, \ell, t\}$ (that stand for head, label and tail, respectively) of a fact $F = (h, \ell, t)$ is represented as a 1-of-K vector x_{atom}, whose embedding is obtained as $e_{\text{atom}} = E_{\text{in}} x_{\text{atom}}$,

[2] www.mturk.com

[3] It is not necessary for the tail to be the only answer, but it is required to be one of the possible answers.

where $E_{\text{in}} \in \mathbb{R}^{D_{\text{Enc}} \times K}$ is the embedding matrix of the input vocabulary and K is the size of that vocabulary. The encoder transforms this embedding into $\text{Enc}(F)_{\text{atom}} \in \mathbb{R}^{H_{\text{Dec}}}$ as $\text{Enc}(F)_{\text{atom}} = W_{\text{Enc}} e_{\text{atom}}$, where $W_{\text{Enc}} \in \mathbb{R}^{H_{\text{Dec}} \times D_{\text{Enc}}}$.

This embedding matrix, E_{in}, could be another parameter of the model to be learned, however, as discussed later (see Section 7.3.3), we have learned it separately and beforehand with TransE (Bordes et al. 2013), a model aimed at modeling this kind of multi-relational data. We fix it and do not allow the encoder to tune it during training.

We call *fact embedding* $\text{Enc}(F) \in \mathbb{R}^{3H_{\text{Dec}}}$ the concatenation $[\text{Enc}(F)_h, \text{Enc}(F)_\ell, \text{Enc}(F)_t]$ of the atom embeddings, which is the input for the next module.

7.3.2. Decoder

For the decoder, we propose to use a GRU recurrent neural network (RNN) (Cho et al. 2014b) with an attention-mechanism (Bahdanau et al. 2015) on the encoder representation to generate the associated question Q to that fact F. Recently, it has been shown that the GRU RNN performs equally well across a range of tasks compared to other RNN architectures, such as the LSTM RNN (Greff et al. 2015). The hidden state of the decoder RNN is computed at each time step n as:

$$g_n^r = \sigma(W_r E_{out} w_{n-1} + C_r c(F, h_{n-1}) + U_r h_{n-1}) \tag{7.2}$$

$$g_n^u = \sigma(W_u E_{out} w_{n-1} + C_u c(F, h_{n-1}) + U_u h_{n-1}) \tag{7.3}$$

$$\tilde{h} = \tanh(W E_{out} w_{n-1} + C c(F, h_{n-1}) + U(g_n^r \circ h_{n-1})) \tag{7.4}$$

$$h_n = g_n^u \circ h_{n-1} + (1 - g_n^u) \circ \tilde{h}, \tag{7.5}$$

where σ is the sigmoid function, s.t. $\sigma(x) \in [0, 1]$, and the circle, $\circ$, represents element-wise multiplication. The initial state h_0 of this RNN is given by the output of a feedforward neural network fed with the fact embedding. The product $E_{out} w_n \in \mathbb{R}^{D_{\text{Dec}}}$ is the decoder embedding vector corresponding to the word w_n (coded as a 1-of-V vector, with V being the size of the output vocabulary), the variables $U_r, U_u, U, C_r, C_u, C \in \mathbb{R}^{H_{\text{Dec}} \times H_{\text{Dec}}}$, $W_r, W_u, W \in \mathbb{R}^{H_{\text{Dec}} \times D_{\text{Dec}}}$ are the parameters of the GRU and $c(F, h_{n-1})$ is the context vector (defined below Eq. 7.6). The vector g^r is called the *reset gate*, g^u as the *update gate* and $\tilde{h}$ the *candidate activation*. By adjusting g^r and g^u appropriately, the model is able to create linear *skip-connections* between distant hidden states, which in turn makes the credit assignment problem easier and the gradient signal stronger to earlier hidden states. Then, at each time step n the set of probabilities of word tokens is given by applying a softmax layer over $V_o tanh(V_h h_{n-1} + V_w E_{out} w_n + V_c c(F, h_{n-1}))$, where $V_o \in \mathbb{R}^{V \times H_{\text{Dec}}}$, $V_h, V_c \in \mathbb{R}^{H_{\text{Dec}} \times H_{\text{Dec}}}$ and $V_w \in \mathbb{R}^{H_{\text{Dec}} \times D_{\text{Dec}}}$. Lastly, the function $c(F, h_{n-1})$ is computed using an attention-mechanism:

$$\begin{aligned} c(F, h_{n-1}) = {} & \alpha_{h,n-1} \text{Enc}(F)_h + \alpha_{\ell,n-1} \text{Enc}(F)_\ell \\ & + \alpha_{t,n-1} \text{Enc}(F)_t, \end{aligned} \tag{7.6}$$

where $\alpha_{h,n-1}, \alpha_{\ell,n-1}, \alpha_{t,n-1}$ are real-valued scalars, which weigh the contribution of the head, label and tail representations. They correspond to the *attention* of the model,

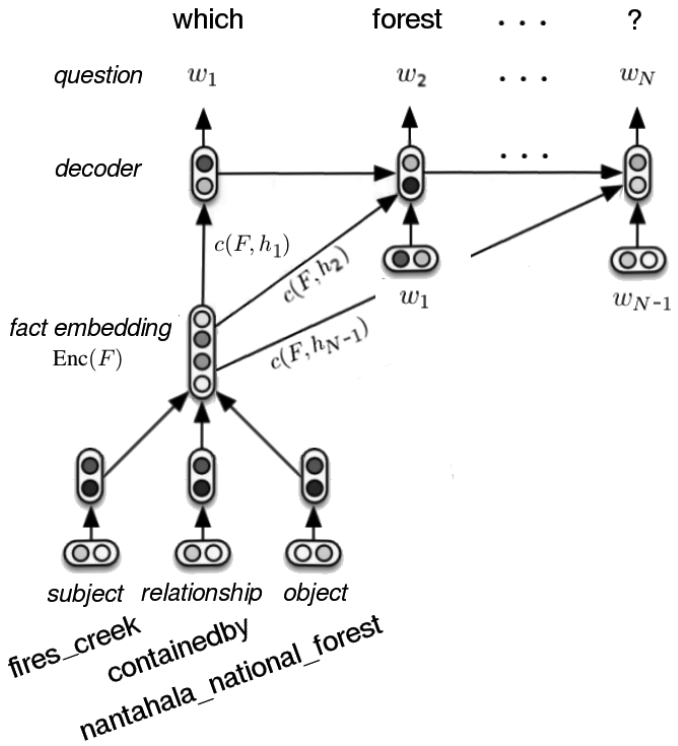


Figure 7.1.: Computational graph of the question generation model

and are computed by applying a one-layer neural network with tanh-activation function on the encoder representations of the fact, Enc(F), and the previous hidden state of the RNN, h_{n-1}, followed by the sigmoid function to restrict the attention values to be between zero and one. Formally:

$$\alpha_{\text{arg},n-1} = \sigma(W_{\text{Dec, att\_post}} \tanh(W_{\text{Dec, att\_pre}} h_{n-1} + W_{\text{Dec, att}} \text{Enc}(F)_{\text{arg}}) + b_{\text{Dec, att\_post}}) \tag{7.7}$$

where $W_{\text{Dec, att\_post}}, W_{\text{Dec, att\_pre}}, W_{\text{Dec, att\_pre}} \in \mathbb{R}^{H_{\text{Dec}} \times H_{\text{Dec}}}$ and $b_{\text{Dec, att\_post}} \in \mathbb{R}^{H_{\text{Dec}}}$.

The model is illustrated in Figure 7.1, where Enc(F) is the fact embedding produced by the encoder model, and $c(F, h_{n-1})$ for $n = 1, \ldots, N$ is the fact representation weighed according to the attention-mechanism, which depends on both the fact F and the previous hidden state of the decoder RNN h_{n-1}. For the sake of simplicity, the attention-mechanism is not shown explicitly.

7.3.3. Modeling the Source Language

A particular problem with the model presented above is related to the embeddings for the entities, relationships and tokens, which all have to be learned in one way or another. If we learn these naively on the SimpleQuestions training set, the model will perform

Closest neighbors to	Warner Bros. Entertainment	Manchester	hindi language
SQ	Billy Gibbons	Ricky Anane	nepali indian
	Jenny Lewis	Lee Dixon	Naseeb
	Lies of Love	Jerri Bryne	Ghar Ek Mandir
	Swordfish	Greg Wood	standard chinese
SQ + FB	Paramount Pictures	Oxford	dutch language
	Sony Pictures Entertainment	Sale	italian language
	Electronic Arts	Liverpool	danish language
	CBS	Guildford	bengali language

Table 7.2.: Examples of differences in the local structure of the vector space embeddings when adding more FB facts

poorly when it encounters previously unseen entities, relationships or tokens. Specifically the multi-relational graph defined by the facts in SimpleQuestions is extremely sparse, i.e. each node has very few edges to other nodes, as can be expected due to high ratio of unique entities over number of examples. Therefore, even for many of the entities in SimpleQuestions, the model may perform poorly if the embedding is learned solely based on the SimpleQuestions dataset alone.

On the source side, we can resolve this issue by initializing the head, relationship and tail embeddings to those learned by applying multi-relational embedding-based models to the knowledge base. Multi-relational embedding-based models (Bordes et al. 2011) have recently become popular to learn distributed vector embeddings for knowledge bases, and have shown to scale well and yield good performance. Due to its simplicity and good performance, we choose to use TransE (Bordes et al. 2013) embeddings. Embeddings for entities with few connections are easy to learn, yet the quality of these embeddings depends on how inter-connected they are. In the extreme case where the head and tail of a triple only appear once in the dataset, the learned embeddings of the head and tail will be semantically meaningless. This happens very often in SimpleQuestions, since only around 5% of the entities have more than 2 connections in the graph. Thus, by applying TransE directly over this set of triples, we would eventually end up with a layout of entities that does not contain clusters of semantically close concepts. In order to guarantee an effective semantic representation of the embeddings, we have to learn them together with additional triples extracted from the whole Freebase graph to complement the SimpleQuestions graph with relevant information for this task.

In particular, we only need a coarse representation for the entities contained in SimpleQuestions, capturing the specific information the annotators used when phrasing the questions, and accordingly we have looked for triples coming from the Freebase graph[4] regarding:

1. Category information: given by the `type/instance` relationship, this ensures that all the entities of the same semantic category are close to each other. Although one

[4]Extracted from one of the latest Freebase dumps (downloaded by mid-August 2015) https://developers.google.com/freebase/data

might think that the expected category of the head/tail could be inferred directly from the label, there are fine-grained differences in the expected types that can be extracted only directly by observing this category information. For example, the right argument of the relationship `location/contain` may be a continent, a country, a city...

2. Geographical information: sometimes the annotators have included information about nationality (e.g. *What French president...?*) or location (e.g. *Where in Germany...?*) of the head and/or tail. This information is given by the relationships `person/nationality` and `location/contained_by`. By including these facts in the learning, we ensure the existence of a fine-grained layout of the embeddings regarding this information within a same category.

3. Gender: similarly, sometimes annotators have included information about gender (e.g. *What male audio engineer...?*). This information is given by the relationship `person/gender`.

To this end, we have included more than 300,000 facts from Freebase in addition to the facts in SimpleQuestions for training. Table 7.2 shows the differences in the embeddings before and after adding additional facts for training the TransE representations.

7.3.4. Generating Questions

To resolve the problem of data sparsity and previously unseen words, we draw inspiration from the placeholders proposed for handling rare words in neural machine translation (Luong et al. 2015). For every question and answer pair, we search for words in the question which overlap with words in the head string of the fact.[5] These words are then replaced by the placeholder token <*placeholder*>. For example, given the fact (`fires_creek, contained_by, nantahala_national_forest` the original question *Which forest is Fires Creek in?* is transformed into the question *Which forest is <placeholder> in?*. The model is trained on these modified questions, which means that model only has to learn decoder embeddings for tokens which are not a good fit for the head string. At test time, after outputting a question, all placeholder tokens are replaced by the head string and then the outputs are evaluated. We call this the Single-Placeholder (SP) model. The main difference with respect to that of Luong et al. (2015) is that we do not use placeholder tokens in the input language, because then the entities and relationships in the input would not be able to transmit semantic (e.g. topical) information to the decoder. If we had included placeholder tokens in the input language, the model would not be able to generate informative words regarding the head in the question (e.g. it would be impossible for the model to learn that the head *Paris* may be accompanied by the words *French city* when generating a question, because it would not see *Paris* but a placeholder token).

[5] We use the tool difflib https://docs.python.org/2/library/difflib.html to find this match between the head string and the words of the question

Model	BLEU	METEOR	Emb. Avg.	Emb. Greedy	Emb. Extrema
Baseline	31.36	33.12	80.76	74.02	67.49
SP Triples	33.27	35.07	82.93	76.72	70.5
MP Triples	32.76	34.97	82.92	76.7	70.53
SP Triples TransE++	**33.32**	**35.38**	83.03	76.78	70.53
MP Triples TransE++	33.28	35.29	**83.08**	**77.01**	**70.82**

Table 7.3.: Test performance for all models w.r.t. BLEU, METEOR and word embedding-based performance metrics. The best performance on each metric is marked in bold font.

A single placeholder token for all question types could unnecessarily limit the model. We therefore also experiment with another model, called the Multi-Placeholder (MP) model, which uses 60 different placeholder tokens such that the placeholder for a given question is chosen based on the taxonomic category extracted from the relationship (e.g. `contained_by` is classified in the category location, and so the transformed question would be *Which forest is <location placeholder> in?*). This could make it easier for the model to learn to phrase questions about a diverse set of entities, but it also introduces additional parameters, since there are now 60 placeholder embeddings to be learned, and therefore the model may suffer from overfitting. This way of addressing the sparsity in the output reduces the vocabulary size to less than 7000 words, which represent the core vocabulary of the questions (e.g. *Wh-* pronouns, verbs, adjectives, common nouns as professions, nationalities...).

7.3.5. Template-based Baseline

To compare our neural network models, we propose a (non-parametric) template-based baseline model, which makes use of the entire training set when generating a question. The baseline operates on questions modified with the placeholder as in the preceding section. Given a fact F as input, the baseline picks a candidate fact F_c in the training set at uniformly random, such that the relationship is the same. As in the SP model, the placeholder token is finally replaced by the head string of the fact F.

7.4. Experiments

To investigate the performance of our models, we make use of both (objective) automatic evaluation metrics, and we conduct a human evaluation study.

7.4.1. Automatic Evaluation Metrics

BLEU (Papineni et al. 2002) and METEOR (Banerjee and Lavie 2005) are two widely used evaluation metrics in statistical machine translation and automatic image-caption generation. Recently, researchers have also started to apply them for evaluating image caption generation (Chen et al. 2015) and generative dialogue models (Galley et al. 2015).

We therefore use both BLEU and METEOR scores for evaluation.[6] In particular, we use METEOR for early-stopping on the validation set. Since METEOR makes use of the WordNet lexical database, it is able to relate semantically related words. We use the default split of the SimpleQuestions dataset into training, validation and test sets.

Unfortunately, neither METEOR nor BLEU are entirely satisfactory for our task, because many words, in particular words related to the entities, are not covered by WordNet. For example, paraphrases of the same entity will not be considered by either of the two metrics (e.g. "NYC" and "New York City"). Therefore, following (Liu et al. 2016), we make use of an additional set of evaluation metrics based on Word2Vec word embeddings (Mikolov et al. 2013). We make use of two different metrics, which embed model questions and human-generated questions into 300 dimensional real-valued vectors each. We then compute the cosine similarity between each pair of corresponding vectors, and take the mean cosine similarities over the test-set as the metric score. The first embedding method, called *Embedding Average* (Emb. Avg.), embeds a question into a real-valued vector by taking the mean over the word embeddings of the question. The second method, is also known as *Embedding Extrema* (Emb. Extrema), embeds a question into a vector by taking the extremum (maximum of the absolute value) across along each dimensionality of the word embeddings (Forgues et al. 2014). We also make use of another metric, called *Embedding Greedy* (Emb. Greedy), which uses the cosine similarity between word embeddings to find the closest word in the human-generated question for each word in the model question. Given the (non-exclusive) alignment between words in the two questions, the mean over the cosine similarities is computed for each pair of questions (Rus and Lintean 2012). The final metric score is the mean over the entire test set.

The results[7] are shown in Table 7.3. The neural network models outperform the template-based baseline by a clear margin across all metrics. Note that the template-based baseline is already a relatively strong model, because it makes use of a separate template for each relationship. This suggests that neural networks are generally better at the question generation task compared to hand-crafted template-based procedures, and therefore that they may be useful for generating question answering corpora. Furthermore, it appears that the best performing models are the models where TransE are trained on the largest set of triples (TransE ++). This set contains, apart from the supporting triples described in Section 7.3.3, triples involving entities which are highly connected to the entities found in the SimpleQuestions facts. In total, around 30 millions of facts, which have been used to generate the 30M Factoid Question-Answer Corpus.

[6] The BLEU evaluation is carried out at the corpus level, where the corpus is the entire test set, using the BLEU4 Moses evaluation script: https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl,
and the Moses tokenization script: https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/tokenizer.perl .
The METEOR evaluation is carried out using the METEOR 1.5 Java application developed by Michael Denkowski and Alon Lavie with its default settings: http://www.cs.cmu.edu/~alavie/METEOR/.

[7] $D_{\text{Enc}} = D_{\text{Dec}}$ and H_{Dec} were fixed to 200 and 600, respectively

Fact	SQ annotator	Baseline	MP Triples TransE++
bayuvi dupki – contained by – europe	where is bayuvi dupki?	what state is the city of bayuvi dupki located in?	what continent is bayuvi dupki in?
illinois – contains – ludlow township	what is in illinois?	what is a tributary found in illinois?	what is the name of a place within illinois?
neo contra – publisher – konami	who published neo contra?	which company published the game neo contra?	who is the publisher for the computer videogame neo contra?
harbord collegiate institute – notable types – school	what is harbord collegiate institute?	what is harbord collegiate institute known for being?	what type of building is the harbord collegiate institute?
pop music – artists – nikki flores	what artist is known for pop music?	An example of pop music is what artist?	who's an american singer that plays pop music?
11664 kashiwagi – orbits – sun	what does 11664 kashiwagi orbit?	which orbit has relationship with 11664 kashiwagi?	around which main star does 11664 kashiwagi gravitate?
cheryl hickey – profession – actor	what is cheryl hickey's profession?	what is cheryl hickey?	what is cheryl hickey's profession in the entertainment industry?

Table 7.4.: Test examples and corresponding questions using the template-based baseline and MP Triples TransE ++ model.

However, it is not clear whether the model with a single placeholder or the model with multiple placeholders performs best. Figure 7.2 show the projected word embeddings of the model with multiple placeholders into 2-D. Most of these tokens converge to a specific region of the embedding space, which seems to indicate that a single token might be sufficient.

Examples of the model with multiple placeholders are shown in Table 7.4. Finally, in comparison to image caption generation tasks the neural network models reach BLEU and METEOR scores which are relatively high (Xu et al. 2015).

We also ran an experiment using duples (*head, label*) as input to the machine translation model, obtaining similar performances as in the case of triples. Though the model is not able to see the *tail* and, thus, it is not able to correctly include information (fine-grained category, nationality...) regarding this element, it did not suffer clear deterioration in terms of these automatic evaluation metrics. This motivates the following human evaluation study.

7.4.2. Human Evaluation Study

It is not clear how accurately BLEU, METEOR and the word embedding-based performance metrics measure the quality of the questions. Therefore, we also carry out

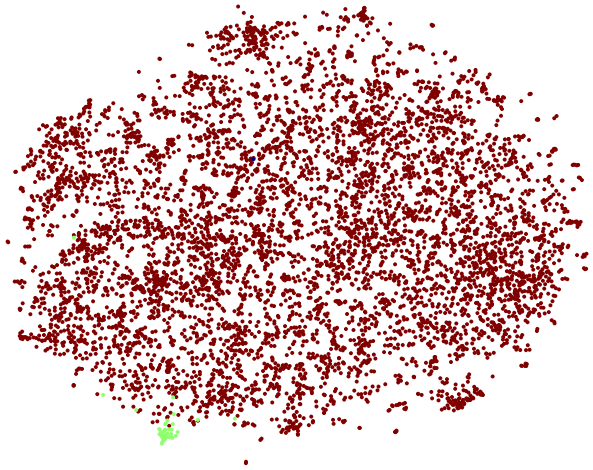


Figure 7.2.: Word embeddings projected in 2-D using t-SNE (Van der Maaten and Hinton 2008). Green points correspond to placeholders embeddings.

pairwise preference experiments on Amazon Mechanical Turk.[8] We setup experiments comparing: Human-Baseline (human and baseline questions), Human-MP (human and MP Triples TransE ++ questions) and Baseline-MP (baseline and MP Triples TransE ++ questions). We show human evaluators a fact along with two questions, one question from each model for the corresponding fact, and ask them to choose the question which is most relevant to the fact and most natural. The human evaluator also has the option of not choosing either question. This is important, for example, if both questions are equally good or if neither of the questions make sense. At the beginning of each experiment, we show the human evaluators two examples of statements and a corresponding pair of questions, where we briefly explain the form of the statements and how questions relate to those statements. Following the introductory examples, we present the facts and corresponding pair of questions one by one. To avoid presentation bias, we randomly shuffle the order of the examples and the order in which questions are shown by each model. During each experiment, we also show four check facts and corresponding check questions at random, which any attentive human annotator should be able to answer easily. We discard responses of human evaluators who fail any of these four checks.

The preference of each example is defined as the question which is preferred by the majority of the evaluators. Examples where neither of the two questions are preferred by the majority of the evaluators, i.e. when there is an equal number of evaluators who prefer each question, are in a separate preference class called "comparable".[9]

[8] www.mturk.com

[9] The probabilities for the "comparable" class in Table 7.5 can be computed in each row as 100 minus the third and fourth column in the table.

Model A	Model B	Model A Preference (%)	Model B Preference (%)	Fleiss' kappa
Human	Baseline	$^{*}\mathbf{56.329 \pm 5.4}$	34.177 ± 5.2	0.242
Baseline	MP Triples TransE++	32.484 ± 5.1	$^{*}\mathbf{60.828 \pm 5.4}$	0.234
Human	MP Triples TransE++	38.652 ± 5.6	$\mathbf{51.418 \pm 5.8}$	0.182

Table 7.5.: Pairwise human evaluation preferences computed across evaluators with 95% confidence intervals. The preferred model in each experiment is marked in bold font.

The results are shown in Table 7.5. An asterisk next to the preferred model indicates a statistically significance likelihood-ratio test, which shows that the model is preferred in at least half of the presented examples with 95% confidence. The last column shows the Fleiss' kappa averaged across batches (HITs) with different evaluators and questions. In total, 3,810 preferences were recorded by 63 independent human evaluators. The questions produced by each model model pair were evaluated in 5 batches. Each human evaluated 44-75 examples (facts and corresponding question pairs) in each batch, such that example was evaluated by 3-5 evaluators. In agreement with the automatic evaluation metrics, the human evaluators strongly prefer either the human or the neural network model over the template-based baseline. Furthermore, it appears that humans cannot distinguish between the human-generated questions and the neural network questions, even preferring the later over the former ones. We hypothesize that it is because our model penalizes uncommon and/or non-natural ways to frame questions[10] and, sometimes, includes specific information about the target that the humans do not (see that example of Table 7.4 where information about the nationality of the expected answer is included in the question. See also last example of that table). This confirms our earlier assertion, that the neural network questions can be used for building question answering systems.

7.5. Conclusion

Inspired by recent neural machine translation models, we propose neural network models to map knowledge base facts into corresponding natural language questions. The produced question and answer pairs are evaluated using automatic evaluation metrics, including BLEU, METEOR and word embedding-based similarity metrics, and are found to outperform a template-based baseline model. When evaluated by untrained human subjects, the question and answer pairs produced by our best performing neural network appears to be indistinguishable from real human-generated questions. Finally, we use our best performing neural network model to generate a corpus of 30M question and answer pairs, which we hope will enable future researchers to improve their question answering systems.

[10] We believe that some questions of the SimpleQuestions dataset have been produced by non-native English speakers

7.6. Discussion on related works

Question generation has attracted interest in recent years with notable work of Rus et al. (2010), followed by the increasing interest from the Natural Language Generation (NLG) community. A simple rule-based approach was proposed in different studies as *wh-fronting* or *wh-inversion* (Kalady et al. 2010, Ali et al. 2010). This comes at the disadvantage of not making use of the semantic content of words apart from their syntactic role. The problem of determining the *question type* (e.g. that a *Where-question* should be triggered for locations), which implies some knowledge of the category type of the elements involved in the sentence, has been addressed in two different ways: by using named entity recognizers (Mannem et al. 2010, Yao and Zhang 2010) or semantic role labelers (Chen et al. 2009). Curto et al. (2012) splits questions into classes according to their syntactic structure, prefix of the question and the category of the answer, and then a pattern is learned to generate questions for that class of questions. After the identification of key points, Chen et al. (2009) apply handcrafted-templates to generate questions framed in the right target expression by following the analysis of Graesser et al. (1992), who classify questions according to a taxonomy consisting of 18 categories.

The works that we discussed so far propose ways to map raw text to questions. This implies a 2-step process: first, transform a text into a symbolic representation (e.g. a syntactic representation of the sentence), and second, transform the symbolic representation of the text into the question (Yao et al. 2012). On the other hand, going from a symbolic representation (structured information) to a question, as we have done in this paper, only involves the second step. Closer to our approach is the work by Olney et al. (2012). They take triples as input, where the edge relation defines the question template and the head of the triple replaces the placeholder token in the selected question template. In the same spirit, Duma and Klein (2013) generate short descriptions from triples by using templates defined by the relationship and replacing accordingly the placeholder tokens for the head and tail.

Recent works in question answering (Bordes et al. 2014b, Berant and Liang 2014) have also used template-based approaches to generate synthetic questions to address the lack of question-answer pairs to train their models.

To our knowledge this is the first work on text generation from structured information by means of a neural network architecture.

Part III.

Summary

8. Conclusions

This thesis presents several contributions on learning representations of multi-relational data. These multi-relational data can be found in a multitude of domains such as social networks, recommendation systems or any repository where the information can be formalized as directed multi-relational graphs. Example of such repositories are the so-called Knowledge Bases, which range from very generic to very specific information and are collected in an automatic or collaborative way, or a mix of both. Nevertheless, except for very specific ones, most of them contain only a very small portion out of the total knowledge of the domain they focus on. For example, Freebase harvest data from many and heterogeneous sources and consequently, it encompasses a very reduced amount out of the total.

Therefore, automatic methods to complete KBs are required to improve the performance of tasks that make use of these knowledge sources. Such task encompass question answering (Bao et al. 2014), machine translation (Knight and Luk 1994) or word-sense disambiguation (Zheng et al. 2012).

Multi-relational graphs can be represented as a pile of adjacency matrices (one per relation) forming a tensor. Because of this nature, tensor factorization methods are a natural way to tackle the learning of representations of the elements involved in the KB. One of the most cited works in this category (formulated within this relational learning framework) is RESCAL, a relaxed version of other classic methods as CP and DEDICOM, which drew a lot of attention on these methods in this specific framework. Along with this category, rule-based methods (Getoor 2007) and other symbolic approaches (Kok and Domingos 2007, Lao et al. 2011) are another appealing and sound way for KB completion. A subcategory within the tensor factorization methods are the energy-based models (LeCun et al. 2006), which score the plausibility of facts. These models rely on the comparison of the score of a triple expressing true information against one that (hypothetically) expresses false information. Several energy functions have been proposed in the recent years (Bordes et al. 2011; 2014a) proving good performance and scalability. For example, they are not affected by the normalization problem of the probabilistic models. Still the analysis of these models remains unclear in some aspects (effect of the regularization, modeling assumption, performance).

In this manuscript we present new energy functions, and new settings and protocols to evaluate these models. We have also experimented different regularization schemes, and provided detailed explanations and evidences on the behavior of the proposed models. A novel application of this relational data on question generation is presented, which may lead to future works on text generation by using neural network architectures.

We hope this thesis brings insight into this problem.

8.1. Future work

We believe that according to the last findings, the right way to learn these graphs is globally (multi-hop fact learning) rather than locally (1-hop fact learning). Consequently we consider the works of Gu et al. (2015), Lin et al. (2015a) and García-Durán et al. (2015) as the ones to follow. There is room for improvement, specifically:

- All these works are translation based models, and consequently only binary interactions are taken into account at explaining the plausibility of a triple. In that sense, we have proven in Chapter 5 the benefits of combining 2- and 3-way interactions, since they (usually) encode supplementary information, and thus an interesting future work would be to apply TATEC in this multi-hop fact setting.

- All these works consider that the arguments *head* and *tail* of most of the relationships are well typed, and accordingly the relationship has to act as operator to "connect" both specific regions of the embedding space. Therefore, the expected entities for either the *head* or *tail* are very homogeneous, i.e. most of them could be classified in a very specific category such as country, automobile manufacturer or football player. This happens to be very common in Freebase, SVO or UMLS. Nevertheless we can easily come up with usual relationships whose arguments are not that well typed. For instance *contains* is a relationship susceptible to have any type of entity in both arguments as for example (`France, contains, Paris`),(`France, contains, The Alps`) or (`France, contains, Musee d'Orsay`). The *contains* operator would push `Paris`, `The Alps` and `Musee d'Orsay` to be close in some dimension of the embedding space, which will likely deteriorate the performance of other relations. For example, even though `Paris`, `The Alps` and `Musee d'Orsay` are the closest entities to `France + contains`, the cascading error will be very present in this kind of situations. We think that this type of relations forms the big next challenge to overcome for the embedding-based models.

Though for both type of datasets (those who contain the totality or only a portion of the facts) embedding-based approaches tend to employ a ranking criterion, it would also be reasonable to formulate it (and consequently to train it) as a classification problem, e.g. $\max(0, 1 - y(\mathbf{w}[\mathbf{e}^h, \mathbf{r}^\ell, \mathbf{e}^t] + b))$ where y is the label (+1: positive, -1: otherwise) associated to the triple (h, ℓ, t) (with parameters $\mathbf{e}^h, \mathbf{r}^\ell, \mathbf{e}^t$, respectively), and $\mathbf{w}$ and b are the parameters of the classifier. However this would come at the cost of not having control on the modeling assumption of the embedding space (on the other side, a ranking criterion allows us to define energy functions that explicitly express a certain modeling assumption). Nevertheless, for example we could also simply train $\max(0, y_i(f(x_i) - \eta))$, where x_i is a triple (either positive or negative), while keeping control on the modeling assumption. This classification formulation would allow us to play with different η values. For instance, we could use two different values of η whether the fact is observed in the dataset or not, in order to reduce the amount of noise introduced in the model by the lack of supervision at generating the negative triples.

This kind of formulation is something that we would like to investigate in the short-term.

In this thesis we have focused on embedding-based models, however as indicated in Section 2.2 there are other families of solutions. In particular, the *symbolic* approach community has been very active in recent years with works such as (Lao et al. 2011, Gardner et al. 2014, Gardner and Mitchell 2015). However, the benchmark datasets this community uses to test their models on are different to the ones used by the embedding-based models community. While we evaluate our models on FB15k, FB1M, SVO, UMLS and Kinships, they do it on NELL (Carlson et al. 2010) and a different partition of Freebase, and unfortunately we cannot compare experimentally the pros and cons of both families. Thus, a future work would be to evaluate and compare these families on several datasets. In that line, and following the work of Gardner et al. (2014) it would be interesting to combine the best of both worlds: the big expressiveness of the symbolic approaches with the no limitation of the connectivity between nodes of the embedding-based models.

Following the work of Chapter 7, we are also interested in generating more complex questions. In that work, the input is a single fact and the output is the associated question with the object of the triple being the answer. We would like to generate questions which result from the interaction of a set of related triples and require a more complex understanding and reasoning in order to find the correct answer. For example, given the facts (`DiCaprio, starred_in, Inception`), (`Cotillard, starred_in, Inception`) and (`Ellen Page, starred_in, Inception`) we would like to output the associated question *Which canadian female actress starred in Inception with DiCaprio and the french actress Cotillard?*. We think that this would allow the question answering community to address a more challenging problem. Nevertheless, the big drawback to tackle this task is the lack of data: to our knowledge, there are no available data to train this.

Bibliography

Husam Ali, Yllias Chali, and Sadid A Hasan. Automation of question generation from sentences. In *Proceedings of QG2010: The Third Workshop on Question Generation*, pages 58–67, 2010.

Michael Auli, Michel Galley, Chris Quirk, and Geoffrey Zweig. Joint language and translation modeling with recurrent neural networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1044–1054, 2013.

Brett W Bader, Richard Harshman, Tamara G Kolda, et al. Temporal analysis of semantic graphs using asalsan. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pages 33–42. IEEE, 2007.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*, 2015.

Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, volume 29, pages 65–72, 2005.

Junwei Bao, Nan Duan, Ming Zhou, and Tiejun Zhao. Knowledge-based question answering as machine translation. *Cell*, 2:6, 2014.

Jonathan Berant and Percy Liang. Semantic parsing via paraphrasing. In *Proceedings of ACL*, volume 7, pages 1415–1425, 2014.

Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data-the story so far. *Semantic Services, Interoperability and Web Applications: Emerging Concepts*, pages 205–227, 2009.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250, 2008.

Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. Learning structured embeddings of knowledge bases. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011*, 2011.

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, pages 2787–2795, 2013.

Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. A semantic matching energy function for learning with multi-relational data. *Machine Learning*, 94(2):233–259, 2014a.

Antoine Bordes, Jason Weston, and Nicolas Usunier. Open question answering with weakly supervised embedding models. In *Machine Learning and Knowledge Discovery in Databases - European Conference, (ECML PKDD)*, pages 165–180, 2014b.

Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. Large-scale simple question answering with memory networks. *arXiv preprint arXiv:1506.02075*, 2015.

Guillaume Bouchard, Dawei Yin, and Shengbo Guo. Convex collective matrix factorization. In *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, pages 144–152, 2013.

Christopher JC Burges. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167, 1998.

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R Hruschka Jr, and Tom M Mitchell. Toward an architecture for never-ending language learning. In *AAAI*, volume 5, page 3, 2010.

J Douglas Carroll and Jih-Jie Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of eckart-young decomposition. *Psychometrika*, 35(3):283–319, 1970.

Kai-Wei Chang, Wen-tau Yih, Bishan Yang, and Christopher Meek. Typed tensor decomposition of knowledge bases for relation extraction. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1568–1579, 2014.

Wei Chen, Gregory Aist, and Jack Mostow. Generating questions automatically from informational text. In *Proceedings of the 2nd Workshop on Question Generation (AIED 2009)*, pages 17–24, 2009.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollar, and C. Lawrence Zitnick. Microsoft COCO captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1724–1734, 2014a.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN Encoder-Decoder for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1724–1734, 2014b.

Andrzej Cichocki and Shun-ichi Amari. *Adaptive blind signal and image processing: learning algorithms and applications*, volume 1. John Wiley & Sons, 2002.

Sergio Curto, A Mendes, and Luisa Coheur. Question generation based on lexico-syntactic patterns learned from the web. *Dialogue and Discourse*, 3(2):147–175, 2012.

Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. A multilinear singular value decomposition. *SIAM journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.

Woodrow W Denham. *The detection of patterns in Alyawara nonverbal behavior*. PhD thesis, University of Washington, 1973.

Chris Ding, Tao Li, Wei Peng, and Haesun Park. Orthogonal nonnegative matrix t-factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 126–135. ACM, 2006.

Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 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*, pages 601–610. ACM, 2014.

Daniel Duma and Ewan Klein. Generating natural language from linked data: Unsupervised template extraction. *ACL*, pages 83–94, 2013.

Susan Dumais, Michele Banko, Eric Brill, Jimmy Lin, and Andrew Ng. Web question answering: Is more always better? In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 291–298, 2002.

Michael D Ekstrand, John T Riedl, and Joseph A Konstan. Collaborative filtering recommender systems. *Foundations and Trends in Human-Computer Interaction*, 4 (2):81–173, 2011.

Christiane Fellbaum. Wordnet and wordnets. In *Encyclopedia of Language and Linguistics*, pages 665–670. Elsevier, 2005.

Gabriel Forgues, Joelle Pineau, Jean-Marie Larchevêque, and Réal Tremblay. Bootstrapping dialog systems with word embeddings. In *Workshop on Modern Machine Learning and Natural Language Processing, Advances in Neural Information Processing Systems*, 2014.

Michel Galley, Chris Brockett, Alessandro Sordoni, Yangfeng Ji, Michael Auli, Chris Quirk, Margaret Mitchell, Jianfeng Gao, and Bill Dolan. deltaBLEU: A discriminative metric for generation tasks with intrinsically diverse targets. In *Proceedings of the 53rd Annual Meeting of the ACL and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL*, pages 445–450, 2015.

Sheng Gao, Ludovic Denoyer, and Patrick Gallinari. Link pattern prediction with tensor decomposition in multi-relational networks. In *Computational Intelligence and Data Mining (CIDM), 2011 IEEE Symposium on*, pages 333–340. IEEE, 2011.

Alberto García-Durán, Antoine Bordes, and Nicolas Usunier. Effective blending of two and three-way interactions for modeling multi-relational data. In *Machine Learning and Knowledge Discovery in Databases*, pages 434–449. Springer, 2014.

Alberto García-Durán, Antoine Bordes, and Nicolas Usunier. Composing relationships with translations. In *EMNLP*, pages 286–290. The Association for Computational Linguistics, 2015.

Alberto García-Durán, Antoine Bordes, Nicolas Usunier, and Yves Grandvalet. Combining two and three-way embeddings models for link prediction in knowledge bases. *Accepted at Journal of Artificial Intelligence Research*, 2016.

Matt Gardner and Tom Mitchell. Efficient and expressive knowledge base completion using subgraph feature extraction. *Proceedings of EMNLP. Association for Computational Linguistics*, 3, 2015.

Matt Gardner, Partha Pratim Talukdar, Jayant Krishnamurthy, and Tom Mitchell. Incorporating vector space similarity in random walk inference over knowledge bases. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, (EMNLP)*, pages 397–406, 2014.

Lise Getoor. *Introduction to statistical relational learning*. MIT press, 2007.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *International conference on artificial intelligence and statistics*, pages 249–256, 2010.

Gene H Golub and Christian Reinsch. Singular value decomposition and least squares solutions. *Numerische mathematik*, 14(5):403–420, 1970.

Ian Goodfellow, Aaron Courville, and Yoshua Bengio. Deep learning. Book in preparation for MIT Press, 2015. URL http://goodfeli.github.io/dlbook/.

Arthur C Graesser, Sallie E Gordon, and Lawrence E Brainerd. QUEST: A model of question answering. *Computers and Mathematics with Applications*, 23(6):733–745, 1992.

Klaus Greff, Rupesh Kumar Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. LSTM: A search space odyssey. *arXiv preprint arXiv:1503.04069*, 2015.

Kelvin Gu, John Miller, and Percy Liang. Traversing knowledge graphs in vector space. *arXiv preprint arXiv:1506.01094*, 2015.

Richard A. Harshman. Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis. *UCLA Working Papers in Phonetics*, 16(1), 1970.

Richard A Harshman. Models for analysis of asymmetrical relationships among n objects or stimuli. In *First Joint Meeting of the Psychometric Society and the Society for Mathematical Psychology, McMaster University, Hamilton, Ontario*, 1978.

Geoffrey Hinton, Li Deng, Dong Yu, George E. Dahl, Abdel-Rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara Sainath, and Brian Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine, IEEE*, 29(6):82–97, 2012.

Geoffrey E Hinton. Learning distributed representations of concepts. In *Proceedings of the eighth annual conference of the cognitive science society*, volume 1, page 12. Amherst, MA, 1986.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S Weld. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 541–550. Association for Computational Linguistics, 2011.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 2333–2338, 2013.

Rodolphe Jenatton, Nicolas L Roux, Antoine Bordes, and Guillaume R Obozinski. A latent factor model for highly multi-relational data. In *Advances in Neural Information Processing Systems*, pages 3167–3175, 2012.

Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2002.

Saidalavi Kalady, Ajeesh Elikkottil, and Rajarshi Das. Natural language question generation using syntax and keywords. In *Proceedings of QG2010: The Third Workshop on Question Generation*, pages 1–10. questiongeneration. org, 2010.

Charles Kemp, Joshua B Tenenbaum, Thomas L Griffiths, Takeshi Yamada, and Naonori Ueda. Learning systems of concepts with an infinite relational model. In *AAAI*, volume 3, page 5, 2006.

Henk AL Kiers. An alternating least squares algorithm for parafac2 and three-way dedicom. *Computational Statistics & Data Analysis*, 16(1):103–118, 1993.

Ryan Kiros, Ruslan Salakhutdinov, and Rich Zemel. Multimodal neural language models. In *Proceedings of the 31st International Conference on Machine Learning*, pages 595–603, 2014.

Kevin Knight and Steve K Luk. Building a large-scale knowledge base for machine translation. In *AAAI*, volume 94, pages 773–778, 1994.

Stanley Kok and Pedro Domingos. Statistical predicate invention. In *Proceedings of the 24th international conference on Machine learning*, pages 433–440. ACM, 2007.

Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 42(8):30–37, 2009.

Denis Krompaß, Stephan Baier, and Volker Tresp. Type-constrained representation learning in knowledge graphs. In *The Semantic Web–ISWC 2015*, pages 640–655. Springer, 2015.

Pieter M Kroonenberg and Jan De Leeuw. Principal component analysis of three-mode data by means of alternating least squares algorithms. *Psychometrika*, 45(1):69–97, 1980.

Ni Lao, Tom Mitchell, and William W Cohen. Random walk inference and learning in a large scale knowledge base. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 529–539. Association for Computational Linguistics, 2011.

Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. A tutorial on energy-based learning. *Predicting structured data*, 1:0, 2006.

Douglas B. Lenat and Ramanathan V. Guha. *Building large knowledge-based systems; representation and inference in the Cyc project*. Addison-Wesley Longman Publishing Co., Inc., 1989.

Xiaobin Li, Stan Szpakowicz, and Stan Matwin. A wordnet-based algorithm for word sense disambiguation. In *IJCAI*, volume 95, pages 1368–1374. Citeseer, 1995.

Yankai Lin, Zhiyuan Liu, and Maosong Sun. Modeling relation paths for representation learning of knowledge bases. *arXiv preprint arXiv:1506.00379*, 2015a.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of AAAI*, 2015b.

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. *arXiv preprint arXiv:1603.08023v1*, 2016.

Ben London, Theodoros Rekatsinas, Bert Huang, and Lise Getoor. Multi-relational learning using weighted tensor decomposition with modular loss. *arXiv preprint arXiv:1303.1733*, 2013.

Vanessa Lopez, Victoria Uren, Marta Sabou, and Enrico Motta. Is question answering fit for the semantic web? a survey. *Semantic Web*, 2(2):125–155, 2011.

Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *Special Interest Group on Discourse and Dialogue (SIGDIAL)*, 2015.

Minh-Thang Luong, Ilya Sutskever, Quoc V Le, Oriol Vinyals, and Wojciech Zaremba. Addressing the rare word problem in neural machine translation. In *Proceedings of ACL*, pages 11–19, 2015.

Prashanth Mannem, Rashmi Prasad, and Aravind Joshi. Question generation from paragraphs at upenn: Qgstec system description. In *Proceedings of QG2010: The Third Workshop on Question Generation*, pages 84–91, 2010.

Alexa T McCray. An upper-level ontology for the biomedical domain. *Comparative and Functional Genomics*, 4(1):80–84, 2003.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Annual Conference of the International Speech Communication Association*, pages 1045–1048, 2010.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013.

Kurt Miller, Michael I Jordan, and Thomas L Griffiths. Nonparametric latent feature models for link prediction. In *Advances in neural information processing systems*, pages 1276–1284, 2009.

J Mocks. Topographic components model for event-related potentials and some biophysical considerations. *IEEE transactions on biomedical engineering*, 6(35):482–484, 1988.

Tanmoy Mukherjee, Vinay Pande, and Stanley Kok. Extracting new facts in knowledge bases:-a matrix trifactorization approach. In *ICML Workshop on Structured Learning: Inferring Graphs from Structured and Unstructured Inputs*, volume 150, 2013.

Vivi Nastase and Stan Szpakowicz. Word sense disambiguation in roget's thesaurus using wordnet. In *Proc. of the NAACL WordNet and Other Lexical Resources Workshop. Pittsburgh*, 2001.

Aniruddh Nath, WASHINGTON EDU, and Pedro Domingos. Learning multiple hierarchical relational clusterings. In *ICML-12 Workshop on Statistical Relational Learning*, 2012.

Carmeliza Navasca, Lieven De Lathauwer, and Stefan Kindermann. Swamp reducing technique for tensor decomposition. In *Signal Processing Conference, 2008 16th European*, pages 1–5. IEEE, 2008.

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

Arvind Neelakantan, Benjamin Roth, and Andrew McCallum. Compositional vector space models for knowledge base completion. *arXiv preprint arXiv:1504.06662*, 2015.

Maximilian Nickel and Volker Tresp. Logistic tensor factorization for multi-relational data. *arXiv preprint arXiv:1306.2084*, 2013.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 809–816, 2011.

Maximilian Nickel, Xueyan Jiang, and Volker Tresp. Reducing the rank in relational factorization models by including observable patterns. In *Advances in Neural Information Processing Systems*, pages 1179–1187, 2014.

Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. Holographic embeddings of knowledge graphs. *arXiv preprint arXiv:1510.04935v2*, 2015.

Dimitri Nion and Lieven De Lathauwer. An enhanced line search scheme for complex-valued tensor decompositions. application in ds-cdma. *Signal Processing*, 88(3):749–755, 2008.

Andrew M Olney, Arthur C Graesser, and Natalie K Person. Question generation from concept maps. *Dialogue and Discourse*, 3(2):75–99, 2012.

Alberto Paccanaro and Geoffrey E Hinton. Learning distributed representations of concepts using linear relational embedding. *Knowledge and Data Engineering, IEEE Transactions on*, 13(2):232–244, 2001.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on ACL*, pages 311–318, 2002.

Simone Paolo Ponzetto and Michael Strube. 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*, pages 192–199. Association for Computational Linguistics, 2006.

Myriam Rajih, Pierre Comon, and Richard A Harshman. Enhanced line search: A novel method to accelerate parafac. *SIAM Journal on Matrix Analysis and Applications*, 30 (3):1128–1147, 2008.

Benjamin Recht, Christopher Re, Stephen Wright, and Feng Niu. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *Advances in Neural Information Processing Systems*, pages 693–701, 2011.

Vasile Rus and Mihai Lintean. A comparison of greedy and optimal assessment of natural language student input using word-to-word similarity metrics. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP, NAACL*, pages 157–162, 2012.

Vasile Rus, Brendan Wyse, Paul Piwek, Mihai Lintean, Svetlana Stoyanchev, and Cristian Moldovan. The first question generation shared task evaluation challenge. In *Proceedings of the 6th International Natural Language Generation Conference*, pages 251–257, 2010.

Iulian Serban, Alberto García-Durán, Caglar Gulcehre, Sungjin Ahn, Sarath Chandar, Aaron Courville, and Yoshua Bengio. The 30m factoid question-answer corpus: Generating factoid questionswith recurrent neural networks. 2016.

Ajit P Singh and Geoffrey J Gordon. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 650–658. ACM, 2008.

Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems*, pages 926–934, 2013.

Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 553–562, 2015.

Nathan Srebro and Ruslan R Salakhutdinov. Collaborative filtering in a non-uniform world: Learning with the weighted trace norm. In *Advances in Neural Information Processing Systems*, pages 2056–2064, 2010.

Ilya Sutskever, Joshua B Tenenbaum, and Ruslan R Salakhutdinov. Modelling relational data using bayesian clustered tensor factorization. In *Advances in neural information processing systems*, pages 1821–1828, 2009.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, 2014.

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

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(2579-2605):85, 2008.

Ellen M Voorhees and DM Tice. Overview of the trec-9 question answering track. In *TREC*, 2000.

Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014.

Yuchung J Wang and George Y Wong. Stochastic blockmodels for directed graphs. *Journal of the American Statistical Association*, 82(397):8–19, 1987.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 1112–1119. Citeseer, 2014.

Jason Weston, Antoine Bordes, Oksana Yakhnenko, and Nicolas Usunier. Connecting language and knowledge bases with embedding models for relation extraction. *arXiv preprint arXiv:1307.7973*, 2013.

Liang Xiong, Xi Chen, Tzu-Kuo Huang, Jeff G Schneider, and Jaime G Carbonell. Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *SDM*, volume 10, pages 211–222. SIAM, 2010.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, 2015.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*, 2014a.

Min-Chul Yang, Nan Duan, Ming Zhou, and Hae-Chang Rim. Joint relational embeddings for knowledge-based question answering. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 645–650, 2014b.

Xuchen Yao and Yi Zhang. Question generation with minimal recursion semantics. In *Proceedings of QG2010: The Third Workshop on Question Generation*, pages 68–75, 2010.

Xuchen Yao, Gosse Bouma, and Yi Zhang. Semantics-based question generation and implementation. *Dialogue and Discourse*, 3(2):11–42, 2012.

Kazuyoshi Yoshii, Ryota Tomioka, Daichi Mochihashi, and Masataka Goto. Infinite positive semidefinite tensor factorization for source separation of mixture signals. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 576–584, 2013.

Jingwei Zhang, Jeremy Salwen, Michael Glass, and Alfio Gliozzo. Word semantic representations using bayesian probabilistic tensor factorization. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, (EMNLP)*, pages 1522–1531, 2014.

Zhicheng Zheng, Xiance Si, Fangtao Li, Edward Y Chang, and Xiaoyan Zhu. Entity disambiguation with freebase. In *Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology-Volume 01*, pages 82–89. IEEE Computer Society, 2012.

Jun Zhu. Max-margin nonparametric latent feature models for link prediction. *arXiv preprint arXiv:1206.4659*, 2012.