A Semantic Matching Energy Function for Learning with Multi-relational Data

Xavier Glorot[1], Antoine Bordes[2], Jason Weston[3], Yoshua Bengio[1]

[1] DIRO, Université de Montréal, Montréal, QC, Canada — {glorotxa,bengioy}@iro.umontreal.ca
[2] CNRS - Heudiasyc, Université de Technologie de Compiègne, France — bordesan@hds.utc.fr
[3] Google, New York, NY, USA — jweston@google.com

1 Introduction

Multi-relational data, which refers to graphs whose nodes represent entities and edges correspond to relations that link these entities, plays a pivotal role in many areas such as recommender systems, the Semantic Web, or computational biology. Relations are modeled as triplets of the form (subject, relation, object), where a relation either models the relationship between two entities or between an entity and an attribute value; relations are thus of several types. In spite of their appealing ability for representing complex data, multi-relational graphs remain complicated to manipulate for several reasons (noise, heterogeneity, large-scale dimensions, etc.), and conveniently represent, summarize or de-noise this kind of data is now a central challenge in statistical relational learning [2].

In this work, we propose a new model to learn multi-relational semantics, that is, to encode multi-relational graphs into representations that capture the inherent complexity in the data, while seamlessly defining similarities among entities and relations and providing predictive power. Our work is based on an original energy function, which is trained to assign low energies to plausible triplets of a multi-relational graph. This energy function, termed *semantic matching energy*, relies on a compact distributed representation: all elements (entity and relation type) are represented into the same relatively low (e.g. 50) dimensional embedding vector space. The embeddings are learnt by a neural network whose particular architecture and training process force them to capture the structure implicit in the training data and generalize the graph formed from training triplets. Unlike in previous work [4, 6, 5, 3], in this model, relation types are modeled similarly as entities. In this way, entities can also play the role of relation type, as in natural language for instance, and this requires less parameters when the number of relation types grows. We show empirically that this model achieves competitive results on benchmark tasks of link prediction, i.e., generalizing outside of the set of given valid triplets.

2 Semantic Matching Energy Function

This work considers multi-relational databases as graph models. To each individual node of the graph corresponds an element of the database, which we term an *entity*, and each link defines a *relation* between entities. Relations are directed and there are typically several different kinds of relations. Let $\mathcal{C}$ denote the dictionary which includes all entities and relation types, and let $\mathcal{R} \subset \mathcal{C}$ be the subset of entities which are relation types. A relation is denoted by a triplet (lhs, rel, rhs), where lhs is the *left* entity, rhs the *right* one and rel the *type* of relation between them.

2.1 Main ideas

The main ideas behind our semantic matching energy function are the following.

- Named symbolic entities (entities *and* relation types) are associated with a d-dimensional vector space, termed the "embedding space". The ith entity is assigned a vector $E_i \in \mathbb{R}^d$. Note that more general mappings from an entity to its embedding are possible.

- The semantic matching energy value associated with a particular triplet (lhs, rel, rhs) is computed by a parametrized function $\mathcal{E}$ that starts by mapping all symbols to their embeddings and then combines them in a structured fashion. Our model is termed "semantic matching" because $\mathcal{E}$ relies on a matching criterion computed between both sides of the triplet.

- The energy function $\mathcal{E}$ is optimized to be lower for training examples than for other possible configurations of symbols.

2.2 Neural network parametrization

The energy function $\mathcal{E}$ (denoted SME) is encoded using a neural network, whose architecture first processes each entity in parallel, like in siamese networks [1]. The intuition is that the relation type should first be used to extract relevant components from each argument's embedding, and put them in a space where they can then be compared.

(1) Each symbol of the input triplet (lhs, rel, rhs) is mapped to its embedding E_{lhs}, E_{rel}, $E_{rhs} \in \mathbb{R}^d$.

(2) The embeddings E_{lhs} and E_{rel} respectively associated with the lhs and rel arguments are used to construct a new relation-dependent embedding $E_{lhs(rel)}$ for the lhs in the context of the relation type represented by E_{rel}, and similarly for the rhs: $E_{lhs(rel)} = g_{left}(E_{lhs}, E_{rel})$ and $E_{rhs(rel)} = g_{right}(E_{rhs}, E_{rel})$, where g_{left} and g_{right} are parametrized functions whose parameters are tuned during training. The dimension of $E_{lhs(rel)}$ and $E_{rhs(rel)}$, which we denote p, is low-dimensional but not necessarily equal to d, the dimension of the entity embedding space.

(3) The energy is computed by "matching" the transformed embeddings of the left-hand and right-hand sides: $\mathcal{E}((lhs, rel, rhs)) = h(E_{lhs(rel)}, E_{rhs(rel)})$, h is a dot product in our experiments.

We studied two options for the g functions, which lead to two versions of SME:

- *Linear form* (denoted SME(linear)), in this case g functions are simply linear layers:

$$
\begin{aligned}
E_{lhs(rel)} &= g_{left}(E_{lhs}, E_{rel}) &= W_{l1}E_{lhs}^\intercal + W_{l2}E_{rel}^\intercal + b_l^\intercal. \\
E_{rhs(rel)} &= g_{right}(E_{rhs}, E_{rel}) &= W_{r1}E_{rhs}^\intercal + W_{r2}E_{rel}^\intercal + b_r^\intercal.
\end{aligned}
$$

with $W_{l1}, W_{l2}, W_{r1}, W_{r2} \in \mathbb{R}^{p \times d}$, $b_l, b_r \in \mathbb{R}^p$ and $E^\intercal$ denotes the transpose of E. This leads to the energy: $\mathcal{E}((lhs, rel, rhs)) = -(W_{l1}E_{lhs}^\intercal + W_{l2}E_{rel}^\intercal + b_l^\intercal)^\intercal (W_{r1}E_{rhs}^\intercal + W_{r2}E_{rel}^\intercal + b_r^\intercal)$.

- *Bilinear form* (denoted SME(bilinear)), g functions are using 3-modes tensors as core weights:

$$
\begin{aligned}
E_{lhs(rel)} &= g_{left}(E_{lhs}, E_{rel}) &= (W_l \bar{\times}_3 E_{rel}^\intercal) E_{lhs}^\intercal + b_l^\intercal. \\
E_{rhs(rel)} &= g_{right}(E_{rhs}, E_{rel}) &= (W_r \bar{\times}_3 E_{rel}^\intercal) E_{rhs}^\intercal + b_r^\intercal.
\end{aligned}
$$

with $W_l, W_r \in \mathbb{R}^{p \times d \times d}$ (weights) and $b_l, b_r \in \mathbb{R}^p$ (biases). $\bar{\times}_3$ denotes the n-mode vector-tensor product along the 3rd mode. This leads to the following form for the energy: $\mathcal{E}((lhs, rel, rhs)) = -((W_l \bar{\times}_3 E_{rel}^\intercal) E_{lhs}^\intercal + b_l^\intercal)^\intercal ((W_r \bar{\times}_3 E_{rel}^\intercal) E_{rhs}^\intercal + b_r^\intercal)$.

Table 1: Statistics of datasets used in this paper.

Dataset	Nb. of relation types	Nb. of entities	Nb. of observed relations	% valid relations in obs. ones
UMLS	49	135	893,025	0.76
Kinships	26	104	281,216	3.84
Nations	56	14	11,191	22.9

To train the parameters of the energy function $\mathcal{E}$ we loop over all of the training data resources and use stochastic gradient descent with a ranking objective inspired by [7].

3   Empirical Evaluation

To evaluate against existing methods, we performed link prediction experiments on benchmarks from the literature, whose statistics are in Table 1.

The link prediction task consists in predicting whether two entities should be connected by a given relation type. This is useful for completing missing values of a graph, forecasting the behavior of a network, etc. but also to assess the quality of a representation. We evaluate our model on UMLS, Nations and Kinships, following the setting introduced in [4]. The standard evaluation metric is area under the precision-recall curve (AUC). Table 2 presents results of SME along with those of RESCAL, MRC, IRM, CP (CANDECOMP-PARAFAC) and LFM, which have been extracted from [5, 3].

The linear formulation of SME is outperformed by SME(bilinear) on all three tasks. The largest differences for Nations and Kinships indicate that, for these problems, a joint interaction between both lhs, rel and rhs is crucial to represent the data well: relations cannot be simply decomposed as a sum of bigrams. This is particularly true for the complex kinship systems of the Alyawarra. On the contrary, interactions within the UMLS network can be represented by simply considering the various (entity,entity) and (entity,relation type) bigrams. Compared to other methods, SME(bilinear) performs similarly to LFM on UMLS but is slightly outperfomed on Nations. On Kinships, it is outperformed by CP, RESCAL and LFM: on this dataset with complex ternary interactions, either the training process of the tensor factorization methods, based on reconstruction, or the combination of bigram and trigram interactions seems to be beneficial compared to our predictive approach. Compared to MRC, which is not using a matrix-based encoding, SME(bilinear) is highly competitive.

Table 2: Comparisons of area under the precision-recall curve (AUC) for link prediction.

Method	**UMLS**	**Nations**	**Kinships**
SME(linear)	$\mathbf{0.983} \pm 0.004$	0.777 ± 0.025	0.149 ± 0.003
SME(bilinear)	$\mathbf{0.985} \pm 0.003$	0.865 ± 0.015	0.894 ± 0.011
LFM	$\mathbf{0.990} \pm 0.003$	$\mathbf{0.909} \pm 0.009$	$\mathbf{0.946} \pm 0.005$
RESCAL	0.98	0.84	**0.95**
CP	0.95	0.83	**0.94**
MRC	0.98	0.75	0.85
IRM	0.70	0.75	0.66

Even if experimental results on these benchmarks are mixed, it is worth noting that, contrary to all previous methods, SME models relation types as vectors, lying in the same space as entities. From a conceptual viewpoint, this is powerful, since it models any relation types as a standard entity (and vice-versa). Hence, SME is the only method that could be directly applied on data for which any entity can also create relationships between other entities.

References

[1] Jame Bromley, Jim W. Bentz, Léon Bottou, Isabelle Guyon, Yann Le Cun, C. Moore, Eduard Säckinger, and Roopak Shah. Signature verification using a siamese time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(4), 1993.

[2] Lise Getoor and Ben Taskar. *Introduction to Statistical Relational Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2007.

[3] Rodolphe Jenatton, Nicolas Le Roux, Antoine Bordes, and Guillaume Obozinski. A latent factor model for highly multi-relational data. In *Advances in Neural Information Processing Systems 25*. 2012.

[4] Charles Kemp, Joshua B. Tenenbaum, Thomas L. Griffiths, Takeshi Yamada, and Naonori Ueda. Learning systems of concepts with an infinite relational model. In *Proceedings of the 21st national conference on Artificial intelligence - Volume 1*, AAAI'06, pages 381–388. AAAI Press, 2006.

[5] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In Lise Getoor and Tobias Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, ICML '11, pages 809–816. ACM, 2011.

[6] Ilya Sutskever, Ruslan Salakhutdinov, and Josh Tenenbaum. Modelling relational data using bayesian clustered tensor factorization. In *Adv. in Neur. Inf. Proc. Syst. 22*, 2009.

[7] Jason Weston, Samy Bengio, and Nicolas Usunier. Large scale image annotation: learning to rank with joint word-image embeddings. *Machine Learning*, 81:21–35, 2010.