Noname manuscript No.
(will be inserted by the editor)

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

Antoine Bordes · Xavier Glorot · Jason Weston · Yoshua Bengio

Received: date / Accepted: date

Abstract Large-scale relational learning becomes crucial for handling the huge amounts of structured data generated daily in many application domains ranging from computational biology or information retrieval, to natural language processing. In this paper, we present a new neural network designed to embed multi-relational graphs into a flexible continuous vector space in which the original data is kept and enhanced. The network is trained to encode the semantics of these graphs in order to assign low energy values to plausible components. We demonstrate that it can scale up to hundreds of thousands of nodes and types of relation while reaching state-of-the-art performance on benchmark tasks such as link prediction or entity resolution. This is assessed on standard datasets from the literature as well as on data from a real-world knowledge base (WordNet). Besides, we present how our method can be applied to perform open-text semantic parsing i.e. to learn to assign a structured meaning representation to almost any sentence of free text.

Keywords Neural Networks · Multi-relational data · Semantic parsing

Antoine Bordes
CNRS - Heudiasyc UMR 7253
Université de Technologie de Compiègne, France
E-mail: antoine.bordes@utc.fr

Xavier Glorot
Université de Montréal
Montéal, QC, Canada
E-mail: glorotxa@iro.umontreal.ca

Jason Weston
Google,
New York, NY, USA
E-mail: jweston@google.com

Yoshua Bengio
Université de Montréal
Montéal, QC, Canada
E-mail: bengioy@iro.umontreal.ca

1 Introduction

Multi-relational data refers to graphs whose nodes represent entities and edges correspond to relations that link these entities. It plays a pivotal role in many areas such as recommender systems, the Semantic Web, or computational biology and is now the core of the field of statistical relational learning (Getoor and Taskar, 2007). Relations are modeled as triplets of the form (subject, predicate, object), where a predicate either models the relationship between two entities or between an entity and an attribute value; predicates are thus of several types. Such data sources are equivalently termed multi-relational graphs. They can also be represented by 3-dimensional tensors, for which each slice represents an adjacency matrix for one predicate. Multi-relational graphs are popular tools for encoding data via knowledge bases, semantic networks or any kind of database following the Resource Description Framework (RDF) format. Hence, they are widely used in the Semantic Web (Freebase, OpenCyc or YAGO[1]) but also for knowledge management in bioinformatics (GeneOntology, UMLS semantic network[2]) or natural language processing (WordNet[3]), to name a few. Social networks can also be represented using RDF.

In spite of their strong ability for representing complex data, multi-relational graphs remain complicated to manipulate for several reasons. First of all, interactions are of multiple types and heterogeneous (some may be frequent or not, some concern only subsets of entities and some all of them, etc.). Besides, most databases have been built either collaboratively or (partly) automatically. As a consequence, data is noisy and incomplete: relations can be missing or be invalid, there can be redundancy among entities because several nodes actually refer to the same concept, etc. Finally, most multi-relational graphs are of very large dimensions in terms of numbers of entities and of relation types. For example, Freebase contains more than 20 millions entities and so does YAGO, DBpedia is composed of 1 billion triplets linking around 4 millions entities, GeneOntology contains more than 350k verified biological entities. Hence, there is a deep need for methods able to handle these databases in order to represent, summarize, complete or merge them.

In this paper, we propose a new model to learn multi-relational semantics, that is, to encode multi-relational data into representations of entities and relations that preserve and enhance its inherent complex structure, while simplifying its manipulation. Our work is based on an original energy function (see Lecun et al (2006) for a review on energy-based learning), 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 distributed representation of the multi-relational data: any element (entity or relation type) is represented into a relatively low (e.g. 50) dimensional embedding vector space. The embeddings are learnt and established by a neural

[1] Respect. available from `freebase.com`, `opencyc.org` and `mpi-inf.mpg.de/yago-naga`.
[2] Respect. available from `geneontology.org` and `semanticnetwork.nlm.nih.gov`.
[3] Available from `wordnet.princeton.edu`.

network whose particular architecture allows to integrate the original data structure. Hence, our model provides a compact and fully distributed representation of the original data. We show empirically that this allows reaching state-of-the-art performance on benchmark tasks of link prediction, entity resolution or entity ranking, i.e., generalizing outside of the set of given valid triplets. This framework is also flexible as demonstrated by its application to the task of open-text semantic parsing: using multi-task training (Caruana, 1995; Collobert and Weston, 2008) on various data sources, we are able to jointly learn representations for words and for entities of a lexical knowledge base (WordNet) and to assign energies to triplets of both kinds of symbols.

This paper is an extension of Bordes et al (2012), which first introduced the model. However, the previous paper was only focused on the application to open-text semantic parsing, whereas the present version has a wider scope and considers many more problems involving multi-relational data. Hence, many new elements are provided: a new and cleaner form of the bilinear formulation, new experiments comparing this model to the state-of-the-art in link prediction and entity resolution, a more comprehensive literature review, and more insights and details on the model formulation and on the training procedure. We also provide a link to an associated open-source implementation.

The paper is organized as follows. Section 2 presents a review of previous work on learning with multi-relational data. Section 3 introduces the semantic matching energy function and Section 4 its training procedure. Extensive experimental results are given in Section 5. Finally, the application to open-text semantic parsing is described in Section 6 and Section 7 concludes and sketches future work.

2 Previous Work

Several ways have been explored to represent and encode multi-relational data, such as Bayesian methods for instance. Kemp et al (2006) introduced the Infinite Relational Model, IRM, a nonparametric Bayesian model whose latent variables are used to discover meaningful partitions among entities and relations. This model provides a great interpretability of the data but suffers from a poor predictive power. Sutskever et al (2009) proposed a refinement with the Bayesian Tensor Clustered Factorization model, BCTF, in which the nonparametric Bayesian framework is coupled with the learning of distributed representations for the entities and relation types. Other proposals have consisted in improving the original model by adding first-order formula with Markov Logic. Hence, MRC, for Multiple Relation Clustering(Kok and Domingos, 2007), is a model able to perform clustering of entities through several relation types simultaneously. Singla and Domingos (2006) presented another model based on Markov Logic Networks (denoted MLN in this paper) for the task of entity resolution.

All these methods share the ability of providing an interpretation of the data but can be slow and costly to train. Models based on tensor factoriza-

tion can be faster and scale to larger data because of their continuous optimization. Standard methods like CANDECOMP/PARAFAC (CP) (Harshman and Lundy, 1994) or those from Tucker (1966) have been applied on multi-relational graphs. For instance, Franz et al (2009) used CP for ranking data from RDF Knowledge Bases. The approach of Singh and Gordon (2008) aims at jointly factoring a relational database and a users-items rating matrix to improve collaborative filtering. Other directions have also been proposed derived from probabilistic matrix factorization for multi-dimensional data (Chu and Ghahramani, 2009) or by adapting dimension reduction techniques such as SVD (Speer et al, 2008; Cambria et al, 2009). Recently, Nickel et al (2011) presented RESCAL, an upgrade over previous tensor factorization methods, which achieves strong predictive accuracies on various problems. Besides, RESCAL has been applied to the large-scale knowledge base YAGO (Nickel et al, 2012). Hence, it is currently considered as the state-of-the-art for factorizing multi-relational data.

Some approaches described above (e.g. BCTF, RESCAL) end up with a distributed representation of the entities and relation types. A line of work consists in actually focusing on learning such representations, usually termed embeddings. The embedding idea has been successful in Natural Language Processing (NLP) via the framework of language models (Bengio et al, 2003) where an embedding per word is learnt: it has been shown that such representations help to improve performance on standard NLP tasks (Bengio, 2008; Collobert et al, 2011). Bordes et al (2010) adapted a related model to a small custom knowledge base for language understanding. For multi-relational data, Linear Relational Embeddings (Paccanaro, 2000; Paccanaro and Hinton, 2001) learn a mapping from the entities into a feature-space by imposing the constraint that relations in this feature-space are modeled by linear operations. In other words, entities are modeled by real-valued vectors and relations by matrices and parameters of both are learnt. This idea has been further improved in the Structured Embeddings (SE) framework of Bordes et al (2011).

Our work lies in the same trend of work since we aim at learning distributed representations of multi-relational data. However, a major difference in our approach is that, contrary to most previous work (including BCTF, RESCAL or SE) we do not model a relation type by a matrix (or a pair of matrices). As we show in the next section, a relation type is represented by a vector and thus shares the status and number of parameters of entities. This is convenient when the number of relation types is large or when relation types can also play the role of entities in relations as we illustrate in Section 6. Note that we empirically compare our model with IRM, BCTF, MRC, MLN, CP, RESCAL and SE in our experiments displayed in Section 5.

3 Semantic Matching Energy Function

This section introduces our model designed to embed multi-relational data into fully distributed representations via a custom energy function.

3.1 Notations

This work considers multi-relational databases as graph models. The data structure is defined by a set of nodes and a set of links. 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. In the remainder of the paper, 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.

3.2 Main Ideas

Inspired by the framework introduced by Bordes et al (2011) as well as by recent work of Bottou (2011), the main idea behind our semantic matching energy function is the following.

- Named symbolic entities (entities *and* relation types) are associated with a d-dimensional vector space, termed the "embedding space", following previous work in neural language models (see Bengio (2008) for a review). The i^{th} entity is assigned a vector $E_i \in \mathbb{R}^d$.
- 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 structured fashion. Our model is termed "semantic matching" because, as we describe in the following, $\mathcal{E}$ relies on a matching criterion computed between both sides of the triple.
- The energy function $\mathcal{E}$ is optimized to be lower for training examples than for other possible configurations of symbols. Hence the semantic energy function can distinguish plausible combinations of entities from implausible ones, and can be used, for instance, to answer questions, e.g. corresponding to a triplet $(lhs, rel, ?)$ with a missing rhs, by choosing among the possible entities a rhs with a relatively lower energy.

3.3 Neural Network Parametrization

The energy function $\mathcal{E}$ (denoted SME) is encoded using a neural network, whose parallel architecture is based on the intuition 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 (see Figure 1). Hence, pairs (lhs, rel) and (rel, rhs) are first combined separately and then, these semantic combinations are *matched*.

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

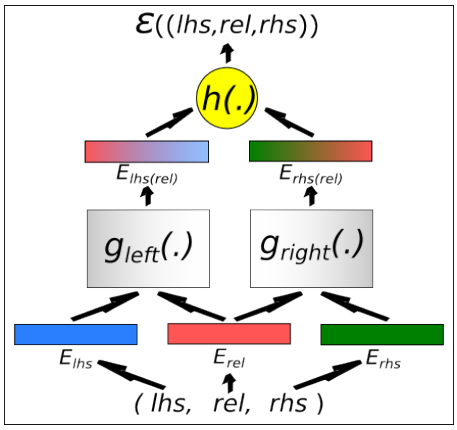


Fig. 1 Semantic matching energy function. A triple of entities (lhs, rel, rhs) is first mapped to its embeddings E_{lhs}, E_{lhs} and E_{lhs}. Then E_{lhs} and E_{rel} are combined using $g_{left}(.)$ to output $E_{lhs(rel)}$ (similarly $E_{rhs(rel)} = g_{right}(E_{rhs}, E_{rel})$). Finally the energy $\mathcal{E}((lhs, rel, rhs))$ is obtained by matching $E_{lhs(rel)}$ and $E_{rhs(rel)}$ with the $h(.)$ function.

(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. Even if it remains low-dimensional, nothing forces the dimension of $E_{lhs(rel)}$ and $E_{rhs(rel)}$, which we denote p, to be equal to d, the one of the entity embedding space.

(3) The energy is computed by "matching" the transformed embeddings of the left-hand side and right-hand side: $\mathcal{E}((lhs, rel, rhs)) = h(E_{lhs(rel)}, E_{rhs(rel)})$, where h can be a simple operator such as a dot product or a more complex function whose parameters are learnt.

Different types of parametrizations can be used for the g and h functions. We chose to use a dot product for the output h function because it is simple, reasonable and has shown to work well in related work (e.g. in Weston et al (2010)). For the g functions, we studied two options, a linear and a bilinear, which lead to two versions of SME detailed below:

– *Linear form* (denoted SME(linear) in the following), 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}, $W_{r2} \in \mathbb{R}^{p \times d}$ (weights), b_l, $b_r \in \mathbb{R}^p$ (biases) and $E^{\intercal}$ denotes the transpose of E. This leads to the following form for the energy:

$$\mathcal{E}((lhs, rel, rhs)) = \left(W_{l1}E_{lhs}^{\intercal} + W_{l2}E_{rel}^{\intercal} + b_l^{\intercal}\right)^{\intercal} \left(W_{r1}E_{rhs}^{\intercal} + W_{r2}E_{rel}^{\intercal} + b_r^{\intercal}\right). \quad (1)$$

– *Bilinear form* (denoted SME(bilinear) in the following), in this case g functions are using 3-modes tensors as core weights:

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

with W_l, $W_r \in \mathbb{R}^{p \times d \times d}$ (weights), b_l, $b_r \in \mathbb{R}^p$ (biases) and $\bar{\times}_3$ denotes the n-mode vector-tensor product along the 3^{rd} mode (see Kolda and Bader (2009) for more details). This leads to the following form for the energy:

$$\mathcal{E}((lhs, rel, rhs)) = \left((W_l \bar{\times}_3 E_{rel}^\intercal)\, E_{lhs}^\intercal + b_l^\intercal\right)^\intercal \left((W_r \bar{\times}_3 E_{rel}^\intercal)\, E_{rhs}^\intercal + b_r^\intercal\right). \quad (2)$$

3.4 Interpretation

We can notice that Equation (1), defining the energy for SME(linear), can be re-written as (bias terms are removed for clarity):

$$\mathcal{E}((lhs, rel, rhs)) = E_{lhs}\tilde{W}_1 E_{rhs}^\intercal + E_{lhs}\tilde{W}_2 E_{rel}^\intercal + E_{rel}\tilde{W}_3 E_{rhs}^\intercal + E_{rel}\tilde{W}_4 E_{rel}^\intercal,$$

with $\tilde{W}_1 = W_{l1}^\intercal W_{r1}$, $\tilde{W}_2 = W_{l1}^\intercal W_{r2}$, $\tilde{W}_3 = W_{l2}^\intercal W_{r1}$ and $\tilde{W}_4 = W_{l2}^\intercal W_{r2} \in \mathbb{R}^{d \times d}$. Hence, the energy can be decomposed into three terms coding for pairs (lhs, rhs), (lhs, rel) and (rel, rhs), and an additional term for rel. This shows that SME(linear) actually represents a triplet as a combination of bigrams.

Similarly, Equation (2), defining the energy for SME(bilinear) can be re-written as:

$$\mathcal{E}((lhs, rel, rhs)) = E_{lhs}\tilde{W}(rel)E_{rhs}^\intercal,$$

with $\tilde{W}(rel) = (W_l \bar{\times}_3 E_{rel}^\intercal)^\intercal\, (W_r \bar{\times}_3 E_{rel}^\intercal) \in \mathbb{R}^{d \times d}$. In this case, the energy is composed of a single term, which depends on all three entities, with a central role for rel. Hence, SME(bilinear) represents a triplet as a trigram. This illustrates that the choice between a linear or a bilinear form for g leads to very different formulation overall. But, the trigram formulation has a cost: SME(bilinear) has d times more parameters to learn than SME(linear).

Finally, we can remark that, in Equation (2), $W_l \bar{\times}_3 E_{rel}^\intercal$ and $W_r \bar{\times}_3 E_{rel}^\intercal$ act as a pair of matrices coding for rel: this can be seen as a distributed version of what previous methods like BCTF, RESCAL or SE proposed.

4 Training

This section details the training procedure for the semantic matching energy function, SME.

4.1 Training Criterion

We are given a training set $\mathcal{D}$ containing m triplets $x = (x_{lhs}, x_{rel}, x_{rhs})$, where $x_{lhs} \in \mathcal{C}$, $x_{rel} \in \mathcal{R}$, and $x_{rhs} \in \mathcal{C}$. We recall that the energy of a triplet is denoted $\mathcal{E}(x) = \mathcal{E}(x_{lhs}, x_{rel}, x_{rhs})$. Ideally, we would like to perform maximum likelihood over $P(x) \propto e^{-\mathcal{E}(x)}$ but this is intractable. The approach we follow here has already been used successfully in ranking settings Collobert et al (2011); Weston et al (2010) and corresponds to performing two approximations. First, like in pseudo-likelihood we only consider one input at a time given the others, e.g. lhs given rel and rhs, which makes normalization tractable. Second, instead of sampling a negative example from the model posterior[4], we use a ranking criterion (that is based on uniformly sampling a negative example).

Intuitively, if one of the elements of a given triplet were missing, then we would like the model to be able to predict the correct entity. The objective of training is to learn the semantic energy function $\mathcal{E}$ such that it can successfully rank the training samples x below all other possible triplets:

$$\mathcal{E}(x) < \mathcal{E}((i, x_{rel}, x_{rhs})) \quad \forall i \in \mathcal{C} : (i, x_{rel}, x_{rhs}) \notin \mathcal{D} \tag{3}$$

$$\mathcal{E}(x) < \mathcal{E}((x_{lhs}, j, x_{rhs})) \quad \forall j \in \mathcal{R} : (x_{lhs}, k, x_{rhs}) \notin \mathcal{D} \tag{4}$$

$$\mathcal{E}(x) < \mathcal{E}((x_{lhs}, x_{rel}, k)) \quad \forall k \in \mathcal{C} : (x_{lhs}, x_{rel}, j) \notin \mathcal{D} \tag{5}$$

Towards achieving this, the following stochastic criterion is minimized:

$$\sum_{x \in \mathcal{D}} \sum_{\tilde{x} \sim Q(\tilde{x}|x)} \max\left(\mathcal{E}(x) - \mathcal{E}(\tilde{x}) + 1, 0\right) \tag{6}$$

where $Q(\tilde{x}|x)$ is a corruption process that transforms a training example x into a corrupted *negative example*. Note that $\max\left(\mathcal{E}(x) - \mathcal{E}(\tilde{x}) + 1, 0\right)$ is similar in shape to the negative log-likelihood $-\log \frac{e^{-\mathcal{E}(x)}}{e^{-\mathcal{E}(x)} + e^{-\mathcal{E}(\tilde{x})}} = -\log \text{sigmoid}(\mathcal{E}(\tilde{x}) - \mathcal{E}(x))$, which corresponds to the probability of sampling x given that only x and $\tilde{x}$ are considered. In the experiments Q only changes one of the three members of the triplet (as in a pseudo-likelihood setup), replacing it by an entity uniformly sampled either from $\mathcal{R}$ if the replaced entity is a relation type, or from $\mathcal{C}/\mathcal{R}$ otherwise. We do not actually check if the negative example is in $\mathcal{D}$. Note that this is not necessary because if we have the symmetry $Q((\tilde{a}, b, c)|(a, b, c)) = Q((a, b, c)|(\tilde{a}, b, c))$ etc. for all elements of the triplet, and it is true here, then the expected contribution to the total expected gradient due to cases where $\tilde{x} \in \mathcal{D}$ is 0. This is because if we consider only the pairs $x, \tilde{x} \in \mathcal{D}$, the average over $\mathcal{D}$ of the gradients $\frac{\partial \mathcal{E}(x)}{\partial \theta}$ equals the average over $\mathcal{D}$ of the gradients $\frac{\partial \mathcal{E}(\tilde{x})}{\partial \theta}$, by our symmetry assumption.

[4] In an energy-based model such as the Boltzmann machine, the gradient of the negative log-likelihood is equal to the gradient of the energy of a positive example (observed and valid) minus the expected value of the gradient of a negative example (sampled from the model). In the case of pseudo-likelihood training one would consider conditional likelihoods $P(x_i|x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_d)$, and only the x_i part of the positive example needs to be resampled for constructing the negative example, using this same posterior.

4.2 Ranking Algorithm

To train the parameters of the energy function $\mathcal{E}$ we loop over all of the training data resources and use stochastic gradient descent (SGD) (Robbins and Monro, 1951). That is, we iterate the following steps:

1. Select a positive training triplet $x_i = (lhs_i, rel_i, rhs_i)$ at random from $\mathcal{D}$.
2. Select at random resp. constraint (3), (4) or (5).
3. Create a negative triplet $\tilde{x}$ by sampling one or two entities either from $\mathcal{R}$ to replace rel_i or from $\mathcal{C}/\mathcal{R}$ to replace lhs_i or rhs_i.
4. If $\mathcal{E}(x_i) > \mathcal{E}(\tilde{x}) - 1$, make a stochastic gradient step to minimize (6).
5. Enforce the constraint that each embedding is normalized, $||E_i|| = 1$, $\forall i$.

The gradient step requires a learning rate λ. The constant 1 in step 4 is the *margin* as is commonly used in many margin-based models such as SVMs (Boser et al, 1992). The normalization in step 5 helps remove scaling freedoms from the model and also makes the effect of the margin actually effective.

Matrix E contains the representations of the entities and is learnt via a *multi-task learning* procedure (Caruana, 1995; Collobert and Weston, 2008) because the same embedding matrix is used for all relation types (each corresponding to a different distribution of entities, i.e., a different task). As a result, the embedding of an entity contains factorized information coming from all the relations in which the entity is involved as lhs, rhs or even rel. For each entity, the model is forced to learn how it interacts with other entities in many different ways. One can think of the elements $E_{i,j}$ of each embedding vector E_i as *learned attributes* for entity i or relation type i. Different tasks may demand different attributes, so that entities that have a common behavior[5] in some way will get the same values for some of their attributes. If the same attributes can be useful for several tasks, then statistical power is gained through parameter sharing, and transfer of information between tasks can happen, making the data of some task informative for generalizing properly on another task.

4.3 Implementation Details

All experimental code have been implemented in Python and using the Theano library (Bergstra et al, 2010).[6] Training is carried out using mini-batches (we create 50 mini-batches for each dataset, independent of its size). All hyperparameter values are set using a validation set. The dimension of the embeddings (d) and the dimension of the output space of g functions (p) are selected among $\{10, 25, 100\}$. There is a different learning rate for the embedding matrix E and for the parameters of g. It is chosen among $\{0.03, 0.01, 0.003, 0.001, 0.0003\}$ for E and among $\{3., 1., 0.3, 0.1, 0.03\}$ for g. Training stops using early stopping on the validation set error.

[5] e.g., appear in semantically similar contexts, i.e., in instances containing the same entities or ones with close-by values of their embedding.

[6] Our code is freely available from `https://github.com/glorotxa/WakaBST`.

Table 1 Statistics of datasets used in our experiments. The top three are fully observed, very sparse i.e. only a small minority of relations are valid and hence are used in a cross-validation scheme. Only a fraction of relations are observed in Nations and WordNet.

Dataset	**Relation types**	**Entities**	**Observed relations**	**% valid relations**
Kinships	26	104	281,216	3.84
UMLS	49	135	893,025	0.76
Cora	7	2,497	43,617,355	0.09
Nations	57	125	11,191	22.9
WordNet	18	41,024	220,924	100

5 Empirical Evaluation

This section proposes an experimental comparison of SME with current state-of-the-art methods for learning representations of multi-relational data.

5.1 Datasets

In order to evaluate against existing methods, we performed experiments on four benchmarks from the literature. These datasets are fully observed, i.e. for each relation type and each potential pair of entities, it has been observed whether the given triplet is valid or not. They are also sparse, i.e. only a small fraction of triplets are valid. We also illustrate the properties of our model on WordNet. Unlike the other benchmarks, this dataset is only partially observed: we only observe some valid triplets, and the rest is unknown, that is, missing triplets can be valid or not. And of course, in that case only a tiny fraction of potential triplets are observed. We describe all datasets below, with some statistics displayed in Table 1.

Kinships Australian tribes are renowned among anthropologists for the complex relational structure of their kinship systems. This dataset, created by Denham (1973), focuses on the Alyawarra, a tribe from Central Australia. 104 tribe members were asked to provide kinship terms for each other. This results in graph of 104 entities and 26 relation types, each of them depicting a different kinship term, such as Adiadya or Umbaidya. See Denham (1973) or Kemp et al (2006) for more details.

UMLS This dataset contains data from the Unified Medical Language System semantic work gathered by McCray (2003). This consists in a graph with 135 entities and 49 relation types. The entities are high-level concepts like 'Disease or Syndrome', 'Diagnostic Procedure', or 'Mammal.' The relations represent verbs depicting causal influence between concepts like 'affect' or 'cause'.

Nations This dataset includes 14 countries, 54 binary predicates representing interactions between countries, and 90 features of the countries. See Rummel (1999) for details.

Table 2 Relation types of WordNet used in our experiments.

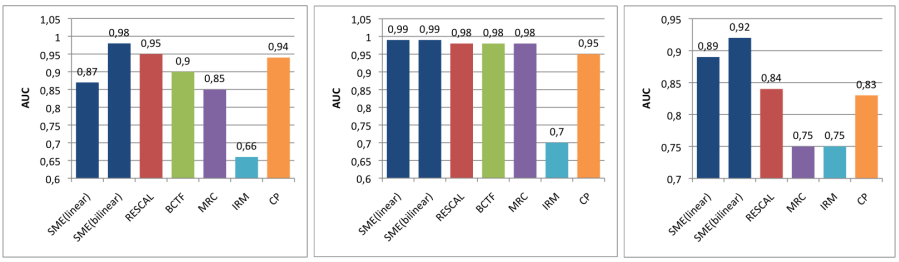
WordNet
_hypernym, *_hyponym*, *_instance_hyponym*, *_instance_hypernym*, *_related_form*, *_has_part*, *_part_of*, *_member_has_part*, *_member_part_of*, *_also_see*, *_attribute*, *_synset_domain_region*, *_synset_domain_usage*, *_synset_domain_topic*, *_verb_group*, *_member_of_domain_region*, *_member_of_domain_usage*, *_member_of_domain_topic*

Cora This dataset is a collection of 1295 different citations to computer science research papers from the Cora Computer Science Research Paper Engine. The original citations have been segmented into fields (author, venue, title, publisher, year, etc.) by Bilenko and Mooney (2003) using an information extraction system. Singla and Domingos (2006) further cleaned the data by correcting some labels and provided a version containing references to 132 different research papers. This leads to a graph with 2497 entities and 7 relation types. Entities include original citation ID, venue names (including conferences, journals, workshops, etc.), first authors, titles and words (composing titles, authors and venue names). Three relations indicate if an author, a title or a venue is mentioned in a citation. Three others state if an author, a title or a venue contains a given word. The last relation specifies if two entities are equivalent (i.e. they refer to written forms of the same concept).

WordNet This Knowledge Base is designed to produce intuitively usable dictionary and thesaurus, and supports automatic text analysis. It encompasses comprehensive knowledge within its graph structure, whose entities (termed *synsets*) correspond to senses, and relation types define lexical relations between those senses. We considered all the entities that were connected with the relation types given in Table 2, although we did remove some entities for which we have too little information. Indeed we filtered out the synsets appearing in less that 15 triplets, as well as relation types appearing in less than 5000 triplets. We obtain a graph with 41,024 synsets and 18 relations types. Examples of WordNet triplets are (*_score_NN_1*, *_hypernym*, *_evaluation_NN_1*) or (*_score_NN_2*, *_has_part*, *_musical_notation_NN_1*). As WordNet is composed of words with different meanings, here we describe its entities by the concatenation of the word, its part-of-speech tag ('NN' for noun, 'VB' for verb, 'JJ' for adjective and 'RB' for adverb) and a number indicating which sense it refers to i.e. *_score_NN_1* is the entity encoding the first meaning of the noun "score". This version of WordNet is different from that used in Bordes et al (2011).

5.2 Link Prediction

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 Kinships, UMLS



KINSHIPS UMLS NATIONS

Fig. 2 Link Prediction. AUC is computed in a 10-fold cross validation setting on Kinships (left), UMLS (center) and Nations (right).

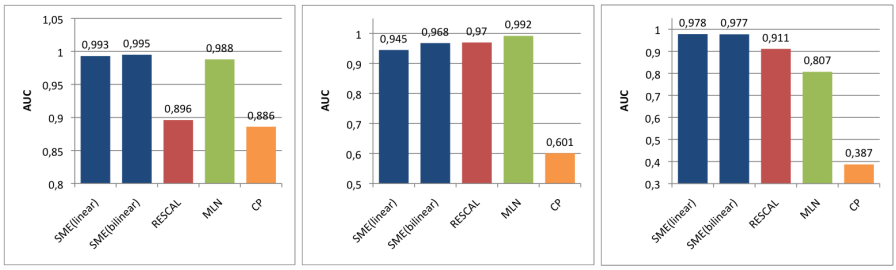
and Nations, following the setting introduced in Kemp et al (2006): data tensors are split in ten folds of valid configurations using (lhs, rel, rhs) triplets as statistical units, and experiments are performed by cross-validation. The standard evaluation metric is AUC. For our own models, we used one of the nine training folds for validation.

Figure 2 presents results of SME along with performances RESCAL, BCTF, MRC, IRM and CP, which have been extracted from Kemp et al (2006); Kok and Domingos (2007); Sutskever et al (2009); Nickel et al (2011). Even if the linear formulation is efficient, it is outperformed by SME(bilinear) on all three tasks. SME(bilinear) performs statistically significantly better than previously published methods on Kinships and Nations and similarly than RESCAL, BCTF and MRC on UMLS (the difference between 0.98 and 0.99 in AUC is not statistically significant according to a z-test). Note that, in almost all our experiments, we experienced that SME(bilinear) was always as good or better than SME(linear). The largest differences for Kinships and Nations indicate that, for these problems, a joint interaction between both lhs, rel and rhs is crucial to well represent the data: relations cannot be simply decomposed as a sum of bigrams.

5.3 Entity Resolution

A second standard test-bed for multi-relational data encoding is entity resolution on the Cora dataset. Many of the nodes of this graph actually refer to identical objects (e.g. the name of an author or a venue can be written in many ways), entity resolution refers to the task of de-duplicating citations, authors and venues as described in Singla and Domingos (2006). The original data contains 7 relation types, including one stating whether two entities are equivalent, denoted $\_equiv$, which is used for de-duplication.

Evaluation is performed in a 5-folds cross-validation scheme (3 folds for training, 1 for validation and 1 for test) and AUC is computed on pairs of entities of the same category (citation, author or venue). Our cross-validation



CITATIONS AUTHORS VENUES

Fig. 3 Entity Resolution. AUC is computed in a 5-fold cross validation setting on Cora for de-duplicating citations (left), authors (center) and venues (right).

scheme process follows that of Nickel et al (2011): splits are created using entities as the statistical unit, and for each split, all *_equiv* triplets involving at least one entity from the test set are removed from the training set and serve only for evaluation. The embedding for the *_equiv* relation type, which is crucial for prediction, is trained on triplets involving only training entities.

Results are presented in Figure 3. SME performs significantly better for resolving venues, but is significantly worse than MLN for de-duplicating authors, while performance of SME and MLN on citations are statistically equivalent. A reason why SME is not performing as well as on link prediction benchmarks might come from the different statistics of Cora (see Table 1). Indeed, this dataset has a much larger number of entities but fewer relation types: our method, which is more designed for large sets of entities and relation types, might be penalized in this case.

5.4 Entity Ranking

Performing an evaluation based on link prediction for WordNet is problematic because only positive triplets are observed. Hence, in this case, there is no negative triplet but only unknown ones for which it is impossible to state whether they are valid or not. For this setting, for which we only have access to positive training and test examples, AUC is not a satisfying metric anymore. Hence, we propose to evaluate our model on this data using the ranking setting proposed in Bordes et al (2011) and described below, because this setup allows an analysis on positive samples only.

We measure the mean predicted rank and the prediction@10, computed with the following procedure. For each test triplet, the left entity is removed and replaced by each of the entities of the dictionary in turn. Energies of those corrupted triplets are computed by the model and sorted by ascending order and the rank of the correct synset is stored. This is done for both the left-hand and right-hand arguments of the relation. The mean predicted rank is the average of those predicted ranks and the precision@10 (or p@10) is

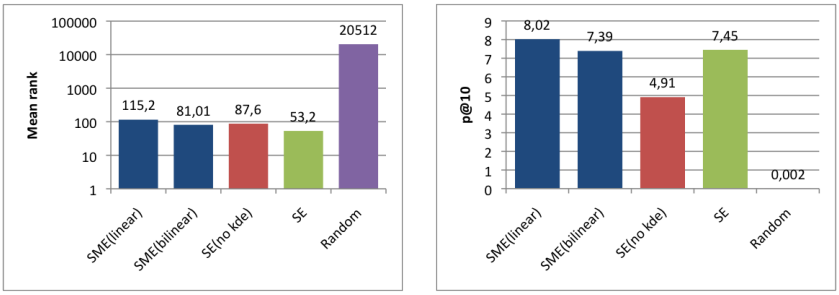


Fig. 4 Entity ranking Mean predicted rank (left - in log-scale) and precision@10 (right) are computed on the WordNet test data.

the proportion of ranks within 1 and 10, divided by 10. WordNet data was split in training/test sets with 211,017 observed triplets for training, 5,000 for validation and 5,000 for testing.

Figure 4 presents comparative results on the test set, together with performance of SE (Bordes et al, 2011). SME is outperformed by SE for the rank measure (left) but performs similarly for the p@10 (SME(linear) is surprisingly better). It is worth nothing that for SE, a Kernel Density Estimator (KDE) is stacked on top of the structured embeddings to improve prediction. Compared with SE without KDE, denoted SE (no KDE), our model is clearly competitive for both metrics.

5.5 Entity Embeddings

The matrix E factorizes information from all relations in which the entity appears. We propose here to illustrate the kind of semantics that can be captured by the representation.

We selected 115 entities from WordNet corresponding to countries from all over the world and to U.S. states. We selected this subset because we know that there exist an underlying structure among them. Then, we projected the corresponding embeddings learnt by the linear and bilinear versions of SME and created 2D plots using t-SNE (van der Maaten and Hinton, 2008). They are given in Figure 5: a different color is used for a each continent; suffixes depicting POS tag and sense indices have been removed for clarity).

The representations learnt by the linear model seem to nicely reflect the geographical semantics, hence encoding the "part-of" information contained in WordNet: nice clusters are formed for each continent. To assess more objectively the quality of this plot, Figure 6 proposes the one obtained for the same entities with the *lesk* similarity measure of the WordNet::Similarity package (Banerjee and Pedersen, 2002).[7] We tried several measures and chose *lesk* because it gave the best result. Comparing both plots tends to indicate that

[7] Freely available from `wn-similarity.sourceforge.net`.

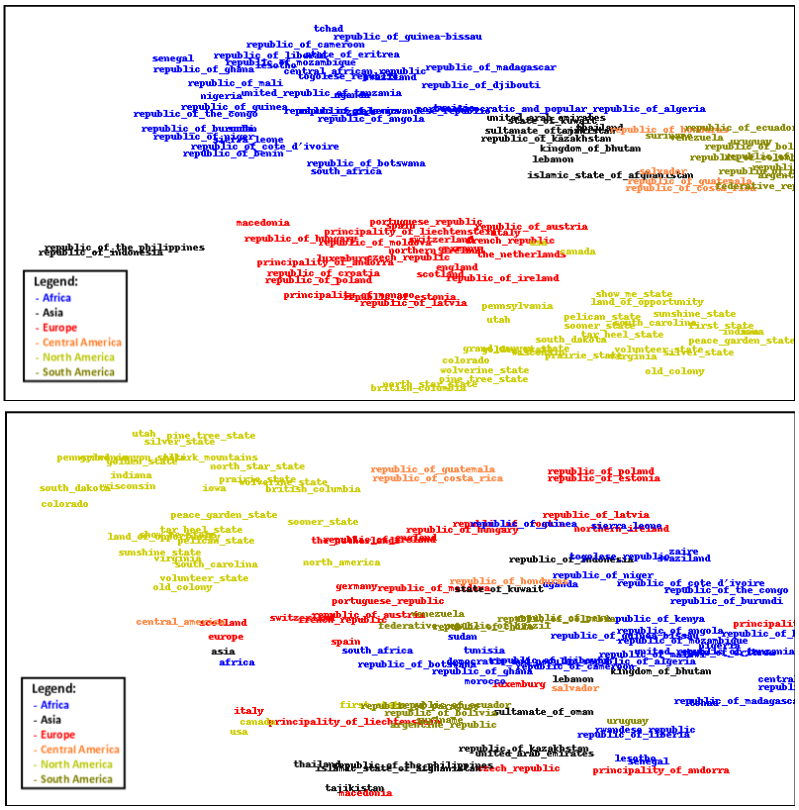


Fig. 5 Entity embeddings. Plots of embeddings (matrix E), learnt by SME(linear) (top) and SME(bilinear) (bottom), for 115 countries selected from WordNet and projected in 2D by t-SNE.

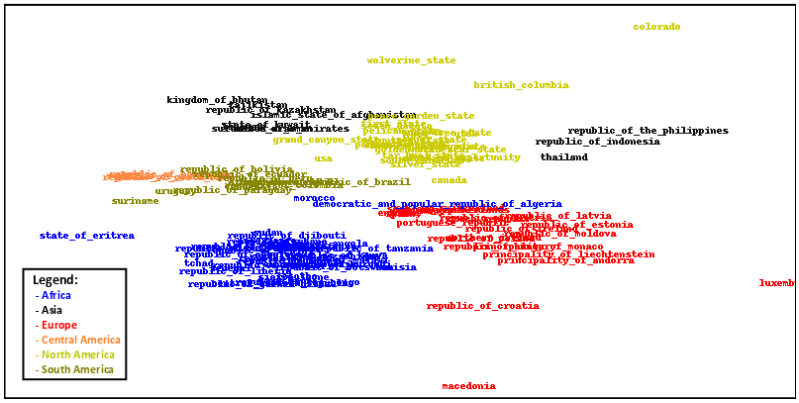


Fig. 6 WordNet::Similarity. Plots of the similarities computed with the Lesk measure among 115 countries selected from WordNet projected in 2D by t-SNE.

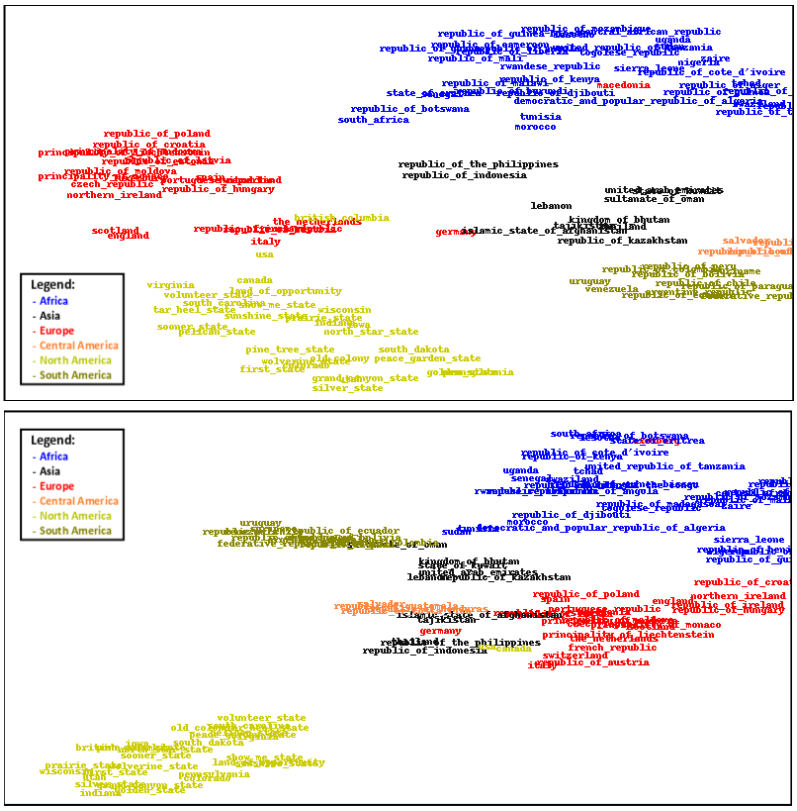


Fig. 7 Entity-relation embeddings. Plots of $E_{lhs(rel)}$ representations, learnt by SME(linear) (top) and SME(bilinear) (bottom), with 115 countries selected from WordNet as *lhs* and *_part_of* as *rel*, projected in 2D by t-SNE.

embeddings learnt by SME(linear) could be used to form a very decent similarity measure on WordNet. But, the comparison is not fair because the *lesk* measure does not only rely on the WordNet graph but also uses *glosses* (i.e. definitions) to improve its similarity measure. Performing the same experiment with WordNet::Similarity measures based only on the graph gives much worse results. Only SME seems able to capture the multi-relational semantics of WordNet.

However, the picture changes with the representations learnt by the bilinear models: the plot (bottom of Figure 5) is much harder to interpret and suggests that the interactions occurring in the SME(bilinear) neural network are more complex, with a more invasive role for the relation type. This intuition is confirmed by the plots of Figure 7. They still display t-SNE projections of representations of the same models for the same entities but not taken at the same level in the network. In this case, we projected the representations obtained by the embeddings when combined with the embedding of the re-

lation type *_part_of* by the g_{left} function. In other words, these are plots of $E_{lhs(rel)}$. The top plot corresponds to the linear model and resemble to the one of Figure 5: as expected, the linear g_{left} does not have dramatic effect on the embedding landscape. The bottom plot, depicting SME(bilinear), is much more interesting because it shows that what was messy at the root level is much more organized: clusters are now formed for continents with the one corresponding to U.S. states further apart from the countries. Embeddings of SME(bilinear) are more interpretable *given a relation type*. The bilinear g functions can drastically modify the distances within the embedding space.

6 Application for Open-text Semantic Parsing

We have introduced a new neural network architecture for learning multi-relational semantics. Its stochastic learning process and its distributed representation of entities *and* relations allow it to scale to large graphs in terms of nodes and links. In this section, we illustrate these appealing properties by applying our model for learning to perform open-text semantic parsing.

6.1 Open-text Semantic Parsing

Context Semantic parsing (Mooney, 2004) aims at building systems able to read text and express its meaning in a formal representation i.e. able to interpret statements expressed in natural language. The purpose of a semantic parser is to analyze the structure of sentence meaning and, formally, this consists of mapping a natural language sentence into a logical *meaning representation* (MR). This task seems too daunting to carry out manually (because of the vast quantity of knowledge engineering that would be required) so machine learning seems an appealing avenue. On the other hand, machine learning models usually require many labeled examples, which can also be costly to gather, especially when labeling properly requires the expertise of a linguist.

Hence, research in semantic parsing can be roughly divided in two tracks. The first one, which could be termed *in-domain*, aims at learning to build highly evolved and comprehensive MRs (Ge and Mooney, 2009; Zettlemoyer and Collins, 2009; Liang et al, 2011). Since this requires sophisticated labeling, such approaches have to be applied to text from a specific domain with restricted vocabulary (a few hundred words). Alternatively, a second line of research, which is termed *open-domain*, works towards learning to associate a MR to any kind of natural language sentence (Shi and Mihalcea, 2004; Giuglea and Moschitti, 2006; Poon and Domingos, 2009). In this case, the supervision is much weaker because it is unrealistic and infeasible to label data for large-scale, open-domain semantic parsing. As a result, models usually infer simpler MRs; this is sometimes referred to as *shallow* semantic parsing. In the following, we show how our model can be applied to build a system for the open-domain category.

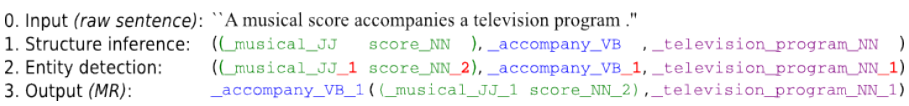


Fig. 8 Open-text semantic parsing. To parse an input sentence (step 0), a preprocessing (lemmatization, POS, chunking, SRL) is first performed (step 1) to clean data and uncover the MR structure. Then, to each lemma is assigned a corresponding WordNet synset (step 2), hence defining a complete meaning representation (step 3).

Joint Learning of Words and Meaning Representations We aim to produce MRs of the following form: $relation(subject, object)$, i.e. relations with subject and object arguments, where each component of the resulting triplet refers to a disambiguated entity. For a given sentence, we propose to infer a MR in two stages: (1) a semantic role labeling (SRL) step predicts the semantic structure, and (2) a disambiguation step assigns a corresponding entity to each relevant word, so as to minimize an energy given to the whole input. The process is illustrated in Figure 8. We consider simple MR structures and rely on an existing method to perform SRL because we focus on step (2). Indeed, in order to go open-domain, a large number of entities must be considered. For this reason, the set of entities considered is defined from WordNet. This results in a dictionary of more than 70,000 words that can be mapped to more than 40,000 possible entities. For each word, WordNet provides a list of candidate senses (synsets) so step (2) reduces to detecting the correct one and can be seen as a challenging all-words word-sense disambiguation (WSD) step. We train our semantic matching energy function via multi-task learning across different knowledge sources including WordNet, ConceptNet (Liu and Singh, 2004) and raw text. In this way MRs induced from text and MRs for WordNet entities are embedded (and hence integrated) in the same space.

6.2 Methodology

We consider MRs which are simple logical expressions $REL(A_0, \ldots, A_n)$, where REL is the relation symbol, and A_0, ..., A_n are its arguments. Note that several such forms can be recursively constructed to build more complex structures. We wish to parse open-domain raw text so a large set of relation types and arguments must be considered. We employ WordNet for defining REL and A_i arguments as proposed in Shi and Mihalcea (2004). Our semantic parsing consists of two stages which we detail below.

Step (1): MR structure inference The first stage consists in preprocessing the text and inferring the structure of the MR. For this stage we use standard approaches, the major novelty of our work lies in applying SME for step (2).

We use the SENNA software[8] (Collobert et al, 2011) to perform part-of-speech (POS) tagging, chunking, lemmatization[9] and semantic role labeling (SRL). In the following, we call a *lemma* the concatenation of a lemmatized word and a POS tag (such as *_score_NN* or *_accompany_VB*). Note the absence of an integer suffix, which distinguishes a lemma from a synset: a lemma is allowed to be semantically ambiguous. The SRL step consists in assigning a semantic role label to each grammatical argument associated with a verb for each proposition. It will be used to infer the *structure* of the MR.

We only consider sentences that match the template (*subject*, *verb*, *direct object*). Here, each of the three elements of the template is associated with a tuple of lemmatized words (i.e. a multi-word phrase). SRL is used to structure the sentence into the (lhs = subject, rel = verb, rhs = object) template. The order is not necessarily subject / verb / direct object in the raw text (e.g. in passive sentences). Clearly, the subject-verb-object structure causes the resulting MRs to have a straightforward structure (with a single relation), but this pattern is the most common and a good choice to test our ideas at scale. Learning to infer more elaborate grammatical patterns is left as future work: we chose here to focus on handling the large scale of the set of entities.

To summarize, this step starts from a sentence and either rejects it or outputs a triplet of lemma tuples, one tuple for the subject, one for the relation or verb, and one for the direct object. To complete our semantic parse (or MR), lemmas must be converted into synsets, that is, we still have to perform disambiguation, which takes place in step (2).

Step (2): Detection of MR entities This second step aims at identifying each semantic entity expressed in a sentence. Given a relation triplet (lhs^{lem}, rel^{lem}, rhs^{lem}) where each element of the triplet is associated with a tuple of lemmas, a corresponding triplet (lhs^{syn}, rel^{syn}, rhs^{syn}) is produced, where the lemmas are replaced by synsets. This step is a form of all-words word-sense disambiguation (WSD) in a particular setup, i.e., w.r.t. the logical form of the semantic parse from step (1). Depending on the lemmas, this can be either straightforward (some lemmas such as *_television_program_NN* or *_world_war_ii_NN* correspond to a single synset) or very challenging (*_run_VB* can be mapped to 33 different synsets and *_run_NN* to 10). Hence, in our proposed framework, MRs correspond to triplets of synsets (lhs^{syn}, rel^{syn}, rhs^{syn}), which can be reorganized to the form rel^{syn} (lhs^{syn}, rhs^{syn}), as shown in Figure 8.

Since the model is structured around relation triplets, MRs and WordNet relations are cast into the same scheme. For example, the WordNet relation (*_score_NN_2* , *_has_part*, *_musical_notation_NN_1*) fits the same pattern as our MRs, with the relation type *_has_part* playing the role of the verb, and the same entities being present in WordNet relations and MRs.

The semantic matching energy function is trained to assign energies to triplets of lemmas and synsets. However, the architecture introduced in Sec-

[8] Freely available from `ml.nec-labs.com/senna/`.

[9] Lemmatization is not carried out with SENNA but with the NLTK toolkit, `nltk.org`.) and transforms a word into its canonical or base form.

tion 3.3 cannot be applied directly. Indeed, here $\mathcal{E}$ must be able to handle variable-size arguments, since for example there could be multiple lemmas in the subject part of the sentence. Hence, we add a pooling stage between steps (1) and (2) (of Section 3.3). The embeddings associated with all the symbols (synsets or lemmas) within the same tuple are aggregated by a pooling function π (we used the mean but other plausible candidates include the sum, the max, and combinations of several such element-wise statistics, such as in Hamel et al (2011)). This re-defines E_{lhs}, E_{rel} and E_{rhs} as follows:

$$E_{lhs} = \pi(E_{lhs_1}, E_{lhs_2}, \ldots),$$
$$E_{rel} = \pi(E_{rel_1}, E_{rel_2}, \ldots),$$
$$E_{rhs} = \pi(E_{rhs_1}, E_{rhs_2}, \ldots),$$

where lhs_j denotes the j-th individual element of the left-hand side tuple, etc.

We use this slightly modified semantic matching energy function to solve the WSD step. We label a triplet of lemmas $((lhs_1^{lem}, lhs_2^{lem}, \ldots), (rel_1^{lem}, \ldots), (rhs_1^{lem}, \ldots))$ with synsets in a greedy fashion, one lemma at a time. For labeling lhs_2^{lem} for instance, we fix all the remaining elements of the triplet to their lemma and select the synset leading to the lowest energy:

$$lhs_2^{syn} = \text{argmin}_{S \in \mathcal{C}(syn|lem)} \mathcal{E}((lhs_1^{lem}, S, \ldots), (rel_1^{lem}, \ldots), (rhs_1^{lem}, \ldots))$$

with $\mathcal{C}(syn|len)$ the set of allowed synsets to which lhs_2^{lem} can be mapped. We repeat that for all lemmas. We always use lemmas as context, and never the already assigned synsets. Future work should investigate more advanced inference schemes, which would probably be iterative and would gradually refine the estimated set of synsets taking into account their mutual agreement.

This is an efficient process as it only requires the computation of a small number of energies, equal to the number of senses for a lemma, for each position of a sentence. However, it requires good representations (i.e. good embedding vectors E_i) for synsets and lemmas because they are used jointly to perform this crucial step. Hence, the multi-task training presented in the next section attempts to learn good embeddings jointly for synsets and lemmas (and good parameters for the g functions).

6.3 Multi-task Training

This section describes how we adapted the training scheme presented in Section 4 for learning synsets and lemmas embeddings using various data sources.

6.3.1 Multiple Data Resources

In order to endow the model with as much common-sense knowledge as possible, the following heterogeneous data sources are combined. Their statistics are summarized in Table 3.

Table 3 Multiple data sources used for learning representations of lemmas and synsets. "Labeled" indicates when triplets consist of text lemmas for which the corresponding synsets are known.

Dataset	**Train. size**	**Test size**	**Labeled**	**Symbols**
WordNet	216,017	5,000	No	synsets
ConceptNet	11,332	0	No	lemmas
Wikipedia	1,498,298	0	No	lemmas
Extended WordNet	786,105	5,000	Yes	lemmas+synsets
Unambig. Wikipedia	981,841	0	Yes	lemmas+synsets

WordNet v3.0 (WN). Already described in Section 5, this is the main resource, defining the dictionary of 41,024 entities. WordNet contains only relations between synsets. However, the disambiguation process needs embeddings for synsets and for lemmas. Following Havasi et al (2010), we created two other versions of this dataset to leverage WN in order to also learn lemma embeddings: "Ambiguated" WN and "Bridge" WN. In "Ambiguated" WN both synset entities of each triplet are replaced by one of their corresponding lemmas. "Bridge" WN is designed to teach the model about the connection between synset and lemma embeddings, thus in its relations the lhs or rhs synset is replaced by a corresponding lemma. Sampling training examples from WN involves actually sampling from one of its three versions, resulting in a triplet involving synsets, lemmas or both.

ConceptNet v2.1 (CN). CN (Liu and Singh, 2004) is a common-sense knowledge base in which lemmas or groups of lemmas are linked together with rich semantic relations as, for example, (*_kitchen_table_NN, _used_for, _eat_VB _breakfast_NN*). It is based on *lemmas* and not synsets, and it does not make distinctions between different senses of a word. Only triplets containing lemmas from the WN dictionary are kept, to finally obtain a total of 11,332 training lemma triplets.

Wikipedia (Wk). This resource is simply raw text meant to provide knowledge to the model in an unsupervised fashion. In this work 50,000 Wikipedia articles were considered, although many more could be used. Using the protocol of the first paragraph of Section 6.2, we created a total of 1,484,966 triplets of lemmas. Imperfect training triplets (containing a mix of lemmas and synsets) are produced by performing the disambiguation step on one of the lemmas. This is equivalent to MAP (Maximum A Posteriori) training, i.e., we replace an unobserved latent variable by its mode according to a posterior distribution (i.e. to the minimum of the energy function, given the observed variables). We have used the 50,000 articles to generate more than 3M examples.

EXtended WordNet (XWN) XWN (Harabagiu and Moldovan, 2002) is built from WordNet *glosses*, syntactically parsed and with content words semantically linked to WN synsets. Using the protocol of Section 6.2, we processed

these sentences and collected 47,957 lemma triplets for which the synset MRs were known. We removed 5,000 of these examples to use them as an evaluation set for the MR entity detection/word-sense disambiguation task. With the remaining 42,957 examples, we created unambiguous training triplets to help the performance of the disambiguation algorithm: for each lemma in each triplet, a new triplet is created by replacing the lemma by its true corresponding synset and by keeping the other members of the triplet in lemma form (to serve as examples of lemma-based context). This led to a total of 786,105 training triplets, from which we removed 10,000 examples for validation.

Unambiguous Wikipedia (Wku) We added to this training set some triplets extracted from the Wikipedia corpus which were modified with the following trick: if one of its lemmas corresponds unambiguously to a synset, and if this synset maps to other ambiguous lemmas, we create a new triplet by replacing the unambiguous lemma by an ambiguous one. Hence, we know the true synset in that ambiguous context. This allowed to create 981,841 additional triplets with supervision (as detailed in the next section), and we named this data set Unambiguous Wikipedia.

6.3.2 Training Procedure

The training algorithm described in Section 4 was used for all the data sources except XWN and Wku. In those two cases, positive triplets are composed of lemmas (as context) and of a disambiguated lemma replaced by its synset. Unlike for Wikipedia, this is labeled data, so we are certain that this synset is the valid sense. Hence, to increase training efficiency and yield a more discriminant disambiguation, in step 3 with probability $\frac{1}{2}$ we either sample randomly from the set of all entities or we sample randomly from the set of remaining candidate synsets corresponding to this disambiguated lemma (i.e. the set of its other meanings).

More precisely, during training, we sequentially alternate between all sources, performing an update of the model parameters with one mini-batch of examples each time. Sizes of mini-batches differ between sources because we split each of them in 50 mini-batches. We always loop over sources in the same order. Training is stopped after 8000 epochs on all sources (or 7 computation days). There is one learning rate for the g functions and another for the embeddings: their values are set using a grid search, choosing among $\{3., 1., 0.3, 0.1, 0.03, 0.01\}$ and $\{0.03, 0.01, 0.003, 0.001, 0.0003, 0.0001\}$ respectively. The model selection criterion is the mean rank from the entity ranking task on the WordNet validation set. Dimensions of embeddings and of the g output space are equal for these experiments and set to 50 (i.e. $d = p = 50$).

6.4 Related Work

Our approach is original by the way that it connects many tasks and many training resources within the same framework. However, it is highly related

with many previous works. Shi and Mihalcea (Shi and Mihalcea, 2004) proposed a rule-based system for open-text semantic parsing using WordNet and FrameNet (Baker et al, 1998) while Giuglea and Moschitti (2006) proposed a model to connect WordNet, VerbNet and PropBank (Kingsbury and Palmer, 2002) for semantic parsing using tree kernels. Poon and Domingos (2009, 2010) recently introduced a method based on Markov-Logic Networks for unsupervised semantic parsing that can be also used for information acquisition. However, instead of connecting MRs to an existing ontology as done here, it constructs a new ontology and does not leverage pre-existing knowledge.

Automatic information extraction is the topic of many models and demos (Snow et al, 2006; Yates et al, 2007; Wu and Weld, 2010; Suchanek et al, 2008) but none of them relies on a joint embedding model. In this trend, some approaches have been directly targeting to enrich existing resources, as we do here with WordNet, (Agirre et al, 2000; Cuadros and Rigau, 2008; Cimiano, 2006) but these never use learning. Finally, several previous works have targeted to improve WSD by using extra-knowledge by either automatically acquiring examples (Martinez et al, 2008) or by connecting different knowledge bases (Havasi et al, 2010).

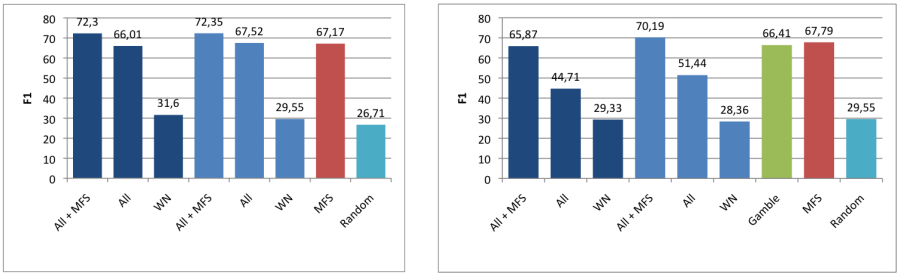
6.5 Experiments

To assess the performance w.r.t. the multi-task training and the diverse data sources, we evaluated models trained with several combinations of data sources. **WN** denotes SME models trained on WordNet, "Ambiguated" WordNet and "Bridge" WordNet, and **All** models are trained on all sources.

6.5.1 Entity Ranking

We evaluated SME (linear) and SME (bilinear) trained with all data sources (**All**) on the WordNet test set with the setting used in Section 5.4. This multi-task training still allows to encode WordNet knowledge, even if it is slightly worse than with WordNet alone. Hence, the linear model achieves a mean rank of 80.01 and a p@10 of 5.740 (instead of 67.57 and 7.675 originally – see Figure 4) and the bilinear model a mean rank of 102.97 and a p@10 of 5.202 (instead of 63.51 and 7.084). This relative loss in performance indicates that it is hard for our model to obtain a good encoding of both text and WordNet triplets.

By multi-tasking with raw text, the numbers of relation types grows from 18 to several thousands. Our model learns similarities, which are more complex with so many relations: by adding text relations, the problem of extracting knowledge from WordNet becomes harder. This degrading effect is a current limitation of the multi-tasking process, even if performance is still very good if one keeps in mind that ranks are over 41,024 entities. Besides, this offers the ability to combine multiple training sources, which is crucial for WSD and eventually for semantic parsing.



XWN SENSEVAL3

Fig. 9 Word Sense Disambiguation results. MFS is just using the Most Frequent Sense. **All+MFS** is our best system, combining all sources of information. **Random** chooses uniformly among allowed synsets.

6.5.2 Word Sense Disambiguation

Performance on WSD is assessed on two test sets: the XWN test set and a subset of English All-words WSD task of SensEval-3.[10] For the latter, we processed the original data using the protocol of Section 6.2 and obtained a total of 208 words to disambiguate (out of $\approx$ 2000 originally). The performance of the most frequent sense (**MFS**) based on WordNet frequencies is also evaluated. Finally, we also report the results of **Gamble** (Decadt et al, 2004), winner of Senseval-3, on our subset of its data. A side effect of our preprocessing of SensEval-3 data is that our subset contains mostly frequent words. This is easier for **MFS** than for **Gamble** because **Gamble** is efficient on rare terms. Hence, **Gamble** performs worse than during the challenge and is outperformed by **MFS**.

F1 scores are presented in Figure 9. The difference between **All** and **WN** indicates that the information from Wikipedia, XWN and Wku is crucial (+40%) and yields performance a bit better than **MFS** (a strong baseline in WSD) on the XWN test set. Actually, performances of the model trained on **WN** alone are roughly equivalent to that of Random. This confirms that knowledge from WordNet and free text are difficult to combine. Still, it is interesting to see that SME is able to train on these various sources of information and to somewhat capture information from them all.

Performance can be greatly improved by combining the **All** sources model and the MFS score. To do so, we converted the frequency information into an energy by taking minus the log frequency and used it as an extra energy term. The total energy function is used for disambiguation. This yields the results denoted by **All+MFS** which achieves the best performance of all the methods tried.

[10] `www.senseval.org/senseval3`.

Table 4 Lists of entities reported by SME(bilinear) trained on **All** and by TextRunner.

	SME(bilinear)	TextRunner
lhs	_army_NN_1	army
rel	_attack_VB_1	attacked
top ranked *rhs*	_troop_NN_4	Israel
	_armed_service_NN_1	the village
	_ship_NN_1	another army
	_territory_NN_1	the city
	_military_unit_NN_1	the fort
top ranked *lhs*	_business_firm_NN_1	People
	_person_NN_1	Players
	_family_NN_1	one
	_payoff_NN_3	Students
	_card_game_NN_1	business
rel	_earn_VB_1	earn
rhs	_money_NN_1	money

6.5.3 WordNet Enrichment

Experiments on WordNet and ConceptNet use a limited number of relation types (less than 20 of them, e.g. *_has_part* and *_hypernym*), and so they do not consider most verbs as relations. Thanks to our multi-task training and unified representation for MRs and WordNet/ConceptNet relations, our model is potentially able to generalize to such relations that do not exist in WordNet.

As illustration, predicted lists of synsets for relation types that do not exist in the two knowledge bases are given in Table 4. We also compare with lists returned by TextRunner (Yates et al, 2007) (an information extraction tool having extracted information from 100M web pages, to be compared with our 50k Wikipedia articles). Lists from both systems seem to reflect commonsense. However, contrary to our system, TextRunner does not disambiguate different senses of a lemma, and thus it cannot connect its knowledge to an existing resource to enrich it.

7 Conclusion

This paper presented a new neural network architecture for learning multi-relational semantics. This model can encode multi-relational graphs or tensors into a flexible continuous vector space in which the original data is kept and enhanced. We empirically showed that it can scale up to hundreds of thousands of nodes and types of relation and that it achieves state-of-the-art performance on benchmark tasks of tensor factorization such as link prediction or entity resolution.

Besides, we present how our method can be applied to perform open-text semantic parsing using a dictionary of more than 70,000 words based on WordNet. We demonstrated our that with multi-task learning over several resources, the proposed model can be used to learn disambiguated meaning representations from raw text. Our system, trained on WordNet and free text (and other sources), can potentially capture the deep semantics of sentences in its energy function and obtained positive experimental results on several tasks that appear to support this assertion. Future work should explore the capabilities of such systems further including other semantic tasks, and more evolved grammars, e.g. with FrameNet (Baker et al, 1998; Coppola and Moschitti, 2010).

An interesting extension of the model presented here extends its applicability to domains where the objects of interest are not all symbolic (i.e., from a finite set). In that case, one cannot associate a free parameter (its embedding vector) to each possible object. An example of such objects are image patches, and they are generally described by a "raw" feature vector. What we propose is to learn a mapping (possibly simply linear) from this raw feature space to the embedding space where the symbolic objects are mapped. Whereas for discrete object, one can view the object's embedding as the product of the embedding matrix by a one-hot vector (with a 1 at the position associated with the object symbol), for continuous objects, in the linear mapping case, the "embedding matrix" maps the raw features (which may be richer than one-hot) to the embedding vector. In this way, relations could involve objects some of which may be sometimes discrete, sometimes continuous.

Acknowledgements The authors would like to acknowledge Léon Bottou, Ronan Collobert, Nicolas Usunier, Nicolas Le Roux, Rodolphe Jenatton and Guillaume Obozinski for inspiring discussions. This work was supported by the DARPA Deep Learning Program, NSERC, CIFAR, the Canada Research Chairs, and Compute Canada.

References

Agirre E, Ansa O, Hovy E, Martinez D (2000) Enriching very large ontologies using the WWW. In: Proceedings of the ECAI 2000 Ontology Learning Worskshop

Baker C, Fillmore C, Lowe J (1998) The berkeley FrameNet project. In: ACL '98, pp 86–90

Banerjee S, Pedersen T (2002) An adapted lesk algorithm for word sense disambiguation using wordnet. In: Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing, Springer-Verlag, CICLing '02, pp 136–145

Bengio Y (2008) Neural net language models. Scholarpedia 3(1):3881

Bengio Y, Ducharme R, Vincent P, Jauvin C (2003) A neural probabilistic language model. JMLR 3:1137–1155

Bergstra J, Breuleux O, Bastien F, Lamblin P, Pascanu R, Desjardins G, Turian J, Warde-Farley D, Bengio Y (2010) Theano: a CPU and GPU

math expression compiler. In: Proceedings of the Python for Scientific Computing Conference (SciPy), URL http://www.iro.umontreal.ca/~lisa/pointeurs/theano_scipy2010.pdf, oral Presentation

Bilenko M, Mooney RJ (2003) Adaptive duplicate detection using learnable string similarity measures. In: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, KDD '03, pp 39–48

Bordes A, Usunier N, Collobert R, Weston J (2010) Towards understanding situated natural language. In: Proc. of the 13th Intern. Conf. on Artif. Intel. and Stat., vol 9, pp 65–72

Bordes A, Weston J, Collobert R, Bengio Y (2011) Learning structured embeddings of knowledge bases. In: Proceedings of the 25th Conference on Artificial Intelligence (AAAI-11), San Francisco, USA

Bordes A, Glorot X, Weston J, Bengio Y (2012) Joint learning of words and meaning representations for open-text semantic parsing. In: Proc. of the 15th Intern. Conf. on Artif. Intel. and Stat., JMLR, vol 22, pp 127–135

Boser B, Guyon I, Vapnik V (1992) A training algorithm for optimal margin classifiers. Proceedings of the fifth annual workshop on Computational learning theory pp 144–152

Bottou L (2011) From machine learning to machine reasoning. Tech. rep., arXiv.1102.1808, URL http://arxiv.org/abs/1102.1808

Cambria E, Hussain A, Havasi C, Eckl C (2009) Affectivespace: Blending common sense and affective knowledge to perform emotive reasoning. In: WOMSA at CAEPIA, pp 32–41

Caruana R (1995) Learning many related tasks at the same time with backpropagation. In: Tesauro G, Touretzky D, Leen T (eds) Advances in Neural Information Processing Systems 7 (NIPS'94), MIT Press, Cambridge, MA, pp 657–664

Chu W, Ghahramani Z (2009) Probabilistic models for incomplete multi-dimensional arrays. Journal of Machine Learning Research - Proceedings Track 5:89–96

Cimiano P (2006) Ontology Learning and Population from Text: Algorithms, Evaluation and Applications. Springer-Verlag

Collobert R, Weston J (2008) A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. In: Proc. of the 25th Inter. Conf. on Mach. Learn.

Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa P (2011) Natural language processing (almost) from scratch. Journal of Machine Learning Research 12:2493–2537

Coppola B, Moschitti A (2010) A general purpose FrameNet-based shallow semantic parser. In: Proceedings of the International Conference on Language Resources and Evaluation (LREC'10)

Cuadros M, Rigau G (2008) Knownet: using topic signatures acquired from the web for building automatically highly dense knowledge bases. In: Proceedings of COLING'08

Decadt B, Hoste V, Daeleamns W, van den Bosh A (2004) Gamble, genetic algorithm optimization of memory-based WSD. In: Proceeding of ACL/SIGLEX Senseval-3

Denham W (1973) The detection of patterns in alyawarra nonverbal behavior. PhD thesis

Franz T, Schultz A, Sizov S, Staab S (2009) Triplerank: Ranking semantic web data by tensor decomposition. In: Proceedings of the 8th International Semantic Web Conference, ISWC '09, pp 213–228

Ge R, Mooney RJ (2009) Learning a Compositional Semantic Parser using an Existing Syntactic Parser. In: Proc. of the 47th An. Meeting of the ACL

Getoor L, Taskar B (2007) Introduction to Statistical Relational Learning (Adaptive Computation and Machine Learning). The MIT Press

Giuglea A, Moschitti A (2006) Shallow semantic parsing based on FrameNet, VerbNet and PropBank. In: Proceeding of the 17th European Conference on Artificial Intelligence (ECAI'06), pp 563–567

Hamel P, Lemieux S, Bengio Y, Eck D (2011) Temporal pooling and multiscale learning for automatic annotation and ranking of music audio. In: In Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR11)

Harabagiu S, Moldovan D (2002) Knowledge processing on extended WordNet. In: Fellbaum C (ed) WordNet: An Electronic Lexical Database and Some of its Applications, MIT Press, pp 379–405

Harshman RA, Lundy ME (1994) Parafac: parallel factor analysis. Comput Stat Data Anal 18(1):39–72

Havasi C, Speer R, Pustejovsky J (2010) Coarse Word-Sense Disambiguation using common sense. In: AAAI Fall Symposium Series

Kemp C, Tenenbaum JB, Griffiths TL, Yamada T, Ueda N (2006) Learning systems of concepts with an infinite relational model. In: Proceedings of the 21st national conference on Artificial intelligence - Volume 1, AAAI Press, AAAI'06, pp 381–388

Kingsbury P, Palmer M (2002) From Treebank to PropBank. In: Proc. of the 3rd International Conference on Language Resources and Evaluation

Kok S, Domingos P (2007) Statistical predicate invention. In: Proceedings of the 24th international conference on Machine learning, ACM, New York, NY, USA, ICML '07, pp 433–440

Kolda TG, Bader BW (2009) Tensor decompositions and applications. SIAM Rev 51(3):455–500

Lecun Y, Chopra S, Hadsell R, marc'aurelio R, Huang f (2006) A tutorial on Energy-Based learning. In: Bakir G, Hofman T, schölkopf B, Smola A, Taskar B (eds) Predicting Structured Data, MIT Press

Liang P, Jordan MI, Klein D (2011) Learning dependency-based compositional semantics. In: Association for Computational Linguistics (ACL)

Liu H, Singh P (2004) Focusing on conceptnet's natural language knowledge representation. In: Proc. of the 8th Intl Conf. on Knowledge-Based Intelligent Information and Engineering Syst.

Martinez D, de Lacalle O, Agirre E (2008) On the use of automatically acquired examples for all-nouns word sense disambiguation. J Artif Int Res 33:79–107

McCray AT (2003) An upper level ontology for the biomedical domain. Comparative and Functional Genomics 4:80–88

Mooney R (2004) Learning Semantic Parsers: An Important But Under-Studied Problem. In: Proc. of the 19th AAAI Conf. on Artif. Intel.

Nickel M, Tresp V, Kriegel HP (2011) A three-way model for collective learning on multi-relational data. In: Getoor L, Scheffer T (eds) Proceedings of the 28th International Conference on Machine Learning (ICML-11), ACM, ICML '11, pp 809–816

Nickel M, Tresp V, Kriegel HP (2012) Factorizing yago: scalable machine learning for linked data. In: Proceedings of the 21st international conference on World Wide Web, WWW '12, pp 271–280

Paccanaro A (2000) Learning distributed representations of concepts from relational data. IEEE Transactions on Knowledge and Data Engineering 13:200–0

Paccanaro A, Hinton G (2001) Learning distributed representations of concepts using linear relational embedding. IEEE Trans on Knowl and Data Eng 13:232–244

Poon H, Domingos P (2009) Unsupervised semantic parsing. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, Singapore, pp 1–10

Poon H, Domingos P (2010) Unsupervised ontology induction from text. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Uppsala, Sweden, pp 296–305

Robbins H, Monro S (1951) A stochastic approximation method. Annals of Mathematical Statistics 22:400–407

Rummel RJ (1999) Dimensionality of nations project: Attributes of nations and behavior of nation dyads. In: ICPSR data file, pp 1950–1965

Shi L, Mihalcea R (2004) Open text semantic parsing using FrameNet and WordNet. In: HLT-NAACL 2004: Demonstration Papers, Boston, Massachusetts, USA, pp 19–22

Singh AP, Gordon GJ (2008) Relational learning via collective matrix factorization. In: Proc. of SIGKDD'08, pp 650–658

Singla P, Domingos P (2006) Entity resolution with markov logic. In: Proceedings of the Sixth International Conference on Data Mining, IEEE Computer Society, pp 572–582

Snow R, Jurafsky D, Ng A (2006) Semantic taxonomy induction from heterogenous evidence. In: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, pp 801–808

Speer R, Havasi C, Lieberman H (2008) Analogyspace: reducing the dimensionality of common sense knowledge. In: Proceedings of the 23rd national conference on Artificial intelligence - Volume 1, AAAI Press, AAAI'08, pp 548–553

Suchanek F, Kasneci G, Weikum G (2008) Yago: A large ontology from Wikipedia and WordNet. Web Semant 6:203–217

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

Tucker LR (1966) Some mathematical notes on three-mode factor analysis. Psychometrika 31:279–311

van der Maaten L, Hinton G (2008) Visualizing high-dimensional data using t-sne. Journal of Machine Learning Research 9:2579–2605

Weston J, Bengio S, Usunier N (2010) Large scale image annotation: learning to rank with joint word-image embeddings. Machine Learning 81:21–35

Wu F, Weld D (2010) Open information extraction using Wikipedia. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Uppsala, Sweden, pp 118–127

Yates A, Banko M, Broadhead M, Cafarella M, Etzioni O, Soderland S (2007) TextRunner: Open information extraction on the Web. In: Proceedings of NAACL-HLT '07, pp 25–26

Zettlemoyer L, Collins M (2009) Learning Context-Dependent Mappings from Sentences to Logical Form. In: Proceedings of the 47th Annual Meeting of the ACL