Discrete-State Variational Autoencoders for Joint Discovery and Factorization of Relations

Discrete-State Variational Autoencoders for Joint Discovery and Factorization of Relations

Diego Marcheggiani
ILLC
University of Amsterdam
marcheggiani@uva.nl

Ivan Titov
ILLC
University of Amsterdam
titov@uva.nl

Abstract

We present a method for unsupervised open-domain relation discovery. In contrast to previous (mostly generative and agglomerative clustering) approaches, our model relies on rich contextual features and makes minimal independence assumptions. The model is composed of two parts: a feature-rich relation extractor, which predicts a semantic relation between two entities, and a factorization model, which reconstructs arguments (i.e., the entities) relying on the predicted relation. The two components are estimated jointly so as to minimize errors in recovering arguments. We study factorization models inspired by previous work in relation factorization and selectional preference modeling. Our models substantially outperform the generative and agglomerative-clustering counterparts and achieve state-of-the-art performance.

1 Introduction

The task of Relation Extraction (RE) consists of detecting and classifying the semantic relations present in text. RE has been shown to benefit a wide range of NLP tasks, such as information retrieval (Liu et al., 2014), question answering (Ravichandran and Hovy, 2002) and textual entailment (Szpektor et al., 2004).

Supervised methods for RE have been successful when small restricted sets of relations are considered. However, human annotation is expensive and time-consuming, and consequently these approaches do not scale well to the open-domain setting where a large number of relations need to be detected in a heterogeneous text collection (e.g., the entire Web). Though weakly-supervised approaches, such as distantly supervised methods and bootstrapping (Mintz et al., 2009; Agichtein and Gravano, 2000), reduce the amount of necessary supervision, they still require examples for every relation considered.

These limitations led to the emergence of unsupervised approaches for RE. These methods extract surface or syntactic patterns between two entities and either directly use these patterns as substitutes for semantic relations (Banko et al., 2007; Banko and Etzioni, 2008) or cluster the patterns (sometimes in context-sensitive way) to form relations (Lin and Pantel, 2001; Yao et al., 2011; Nakashole et al., 2012; Yao et al., 2012). The existing methods, given their generative (or agglomerative clustering) nature, rely on simpler features than their supervised counterparts and also make strong modeling assumptions (e.g., assuming that arguments are conditionally independent of each other given the relation). These shortcomings are likely to harm their performance.

In this work, we tackle the aforementioned challenges and introduce a new model for unsupervised relation extraction. We also describe an efficient estimation algorithm which lets us experiment with large unannotated collections. Our model is composed of two components:

- *an encoding component*: a feature-rich relation extractor which predicts a semantic relation between two entities in a specific sentence given contextual features;
- *a reconstruction component*: a factorization model which reconstructs arguments (i.e., the entities) relying on the predicted relation.

The two components are estimated jointly so as to minimize errors in reconstructing arguments. While

Transactions of the Association for Computational Linguistics, vol. 4, pp. 231–244, 2016. Action Editor: Sebastian Riedel.
Submission batch: 10/2015; Revision batch: 2/2016; Published 6/2016.

learning to predict left-out arguments, the inference algorithm will search for latent relations that simplify the argument prediction task as much as possible. Roughly, such an objective will favour inducing relations that maximally constrain the set of admissible argument pairs. Our hypothesis is that relations induced in this way will be interpretable by humans and useful in practical applications. Why is this hypothesis plausible? Primarily because humans typically define relations as an abstraction capturing the essence of the underlying situation. And the underlying situation (rather than surface linguistic details like syntactic functions) is precisely what imposes constraints on admissible argument pairs.

This framework allows us to both exploit rich features (in the encoding component) and capture interdependencies between arguments in a flexible way (both in the reconstruction and encoding components).

The use of a reconstruction-error objective, previously considered primarily in the context of training neural autoencoders (Hinton, 1989; Vincent et al., 2008), gives us an opportunity to borrow ideas from the well-established area of statistical relational learning (Getoor and Taskar, 2007), and, more specifically, relation factorization. In this area, tensor and matrix factorization methods have been shown to be effective for inferring missing facts in knowledge bases (Bordes et al., 2011; Riedel et al., 2013; Chang et al., 2014; Bordes et al., 2014; Sutskever et al., 2009). In our work, we also adopt a fairly standard RESCAL factorization (Nickel et al., 2011) and use it within our reconstruction component.

Though there is a clear analogy between statistical relational learning and our setting, there is also a very significant difference. In contrast to relational learning, rather than factorizing existing relations (an existing 'database'), our method simultaneously discovers the relational schema (i.e., an inventory of relations) and a mapping from text to the relations (i.e., a relation extractor), and it does it in such a way as to maximize performance on reconstruction (i.e., inference) tasks. This analogy also highlights one important property of our framework: unlike generative models, we explicitly force our semantic representations to be useful for at least the most basic form of semantic inference (i.e., inferring an argument based on the relation and another argument). It is important to note that the model is completely agnostic about the real semantic relation between two arguments, as the relational schema is discovered during learning.

We consider both a factorization method inspired by previous research in knowledge base modeling (as discussed above) and another, even simpler one, based on ideas from previous research on modeling selectional preferences (e.g., Resnik (1997); Ó Séaghdha (2010); Van de Cruys (2010)), plus their combination. Our models are applied to a version of the New York Times corpus (Sandhaus, 2008). In order to evaluate our approach, we follow Yao et al. (2011) and align named entities in our collection to Freebase (Bollacker et al., 2008), a large collaborative knowledge base. In this way we can evaluate a subset of our induced relations against relations in Freebase. Note that Freebase has not been used during learning, making this a fair evaluation scenario for an unsupervised relation induction method. We also qualitatively evaluate our model by both considering several examples of induced relations (both appearing and not appearing in Freebase) and visualizing embeddings of named entities induced by our model. As expected, the choice of a factorization model affects the model performance. Our best models substantially outperform the state-of-the-art generative Rel-LDA model of Yao et al. (2011): 35.8% F_1 and 29.6% F_1 for our best model and Rel-LDA, respectively.

The rest of the paper is structured as follows. In the following section, we formally describe the problem. In Section 3, we motivate our approach. In Section 4, we formally describe the method. In Section 5 we describe our experimental setting and discuss the results. We give more background on RE, knowledge base completion and autoencoders in Section 6.

2 Problem Definition

In the most standard form of RE considered in this work, an extractor, given a sentence and a pair of named entities e_1 and e_2, needs to predict the underlying semantic relation r between these entities. For example, in the sentence

Roger Ebert wrote a review of The Fall

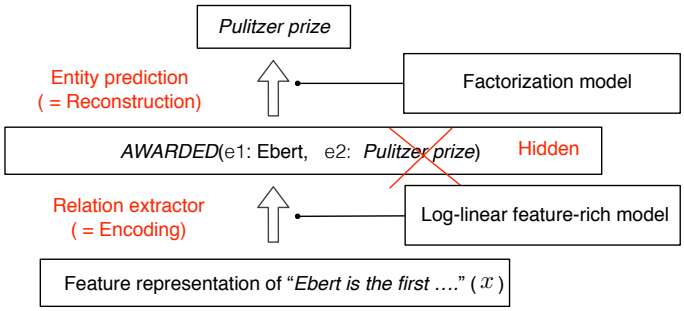

Figure 1: Inducing relations with discrete-state autoencoders.

we have two entities e_1 = `Roger Ebert` and e_2 = `The Fall`, and the extractor should predict the semantic relation r = *REVIEWED*.[1] The standard approach to this task is to either rely on human annotated data (i.e., supervised learning) or use data generated automatically by aligning knowledge bases (e.g., Freebase) with text (called distantly-supervised methods). Both classes of approaches assume a predefined inventory of relations and a manually constructed resource.

In contrast, the focus of this paper is on open-domain unsupervised RE (also known as relation discovery) where no fixed inventory of relations is provided to the learner. The methods induce relations from the data itself. Previous work on this task (Banko et al., 2007), as well as on its generalization, called unsupervised semantic parsing (Poon and Domingos, 2009; Titov and Klementiev, 2011), groups patterns between entity pairs (e.g., *wrote a review*, *wrote a critique* and *reviewed*) and uses these clusters as relations. Other approaches (e.g., Shinyama and Sekine (2006); Yao et al. (2011); Yao et al. (2012); de Lacalle and Lapata (2013)), including the one introduced in this paper, perform context-sensitive clustering, that is, they treat relations as latent variables and induce them for each entity-pair occurrence individually. Rather than relying solely on a pattern between entity pairs, the latter class of methods can use additional context to decide that *Napoleon reviewed the Old Guard* and the above sentence about Roger Ebert should not be labeled with the same relation.

[1] In some of our examples we will use relation names, although our method, as virtually any other latent variable model, will not induce names but only indices.

Unsupervised relation discovery is an important task because existing knowledge bases (e.g., Freebase, Yago (Suchanek et al., 2007), DBpedia (Auer et al., 2007)) do not have perfect coverage even for most standard domains (e.g., music or sports), and, arguably more importantly, because there are many domains not covered by these resources. Though one option is to provide a list of relations with seed examples for each of them and then use bootstrapping (Agichtein and Gravano, 2000), it requires domain knowledge and may thus be problematic. In these cases unsupervised relation discovery is the only non-labour-intensive way to construct a relation extractor. Moreover, unsupervised methods can also aid in building new knowledge bases by providing an initial set of relations which can then be refined.

As is common, in this work we limit ourselves to only considering binary relations between entities occurring in the same sentence. We focus only on extracting semantic relations, assuming that named entities have already been recognized by an external method (Finkel et al., 2005). As in previous work (Yao et al., 2011), we are not trying to detect if there is a relation between two entities or not; our aim is to detect a relation between each pair of entities appearing in a sentence. In principle, heuristics (i.e., based on the syntactic dependency paths connecting arguments) can be used to get rid of unlikely pairs.

3 Our Approach

We approach the problem by introducing a latent variable model which defines the interactions between a latent relation r and the observables: the

entity pair (e_1, e_2) and other features of the sentence x. The idea which underlies much of latent variable modeling is that a good latent representation is the one that helps us to reconstruct the input (i.e., x, including (e_1, e_2)). In practice, we are not interested in predicting x, as x is observable, but rather in inducing an appropriate latent representation (i.e., r). Thus, it is crucial to design the model in such a way that a good r (the one predictive of x) indeed encodes relations rather than some other form of abstraction.

In our approach, we encode this reconstruction idea very explicitly. As a motivating example, consider the following sentence:

Ebert is the first journalist to win the Pulitzer prize.

As shown in Figure 1, let us assume that we hide one argument, chosen at random: for example, $e_2 =$ `Pulitzer prize`. Now the purpose of the reconstruction component is to reconstruct (i.e., infer) this argument relying on another argument ($e_1 =$ `Ebert`), the latent relations r and nothing else. At learning time, our inference algorithm will search through the space of potential relation clusterings to find the one that makes these reconstruction tasks as simple as possible. For example, if the algorithm clusters expressions *is the first journalist to win* together with *was awarded*, the prediction is likely to be successful, assuming that the passage *Ebert was awarded the Pulitzer prize* has been observed elsewhere in the training data. On the contrary, if the algorithm clustered *is the first journalist to win* with *presented*, we are likely to make a wrong inference (i.e., predict `Golden Thumb award`). Given that we optimize the reconstruction objective, the former clustering is much more likely than the latter. Reconstruction can be seen as a knowledge base factorization approach similar to the ones of Bordes et al. (2014). Notice that the model's final goal is to learn a good relation clustering, and that the reconstruction objective is used as a means to reach this goal. For reasons which will be clear in a moment, we will refer to the model performing the prediction of entities relying on other entities and relations as a *decoder* (a.k.a. *the reconstruction component*).

Despite our description of the model as pattern-clustering, it is important to stress that we are inducing clusters in a context-sensitive way. In other words, we are learning an *encoder*: a feature-rich classifier, which predicts a relation for a specific sentence and an entity pair in this sentence. Clearly, this is a better approach because some of the patterns between entities are ambiguous and require extra features to disambiguate them (recall the example from the previous section), whereas other patterns may not be frequent enough to induce reliable clustering (e.g., *is the first journalist to win*). The encoding and reconstruction components are learned jointly so as to minimize the prediction error. In this way, the encoder is specialized to the defined reconstruction problem.

4 Reconstruction Error Minimization

In order to implement the desiderata sketched in the previous section, we take inspiration from a framework popular in the neural network community, namely autoencoders (Hinton, 1989). Autoencoders are composed of two components: an encoder which predicts a latent representation y from an input x, and a decoder which relies on the latent representation y to recover the input ($\tilde{x}$). In the learning phase, the parameters of both the encoding and reconstruction part are chosen so as to minimize a reconstruction error (e.g., the Euclidean distance $||x - \tilde{x}||_2$).

Although popular within the neural network community (where y is defined as a real-valued vector), autoencoders have recently been applied to the discrete-state setting (where y is defined as a categorical random variable, a tuple of variables or a graph). For example, such models have been used in the context of dependency parsing (Daumé III, 2009), or in the context of POS tagging and word alignment (Ammar et al., 2014; Lin et al., 2015a). The most related previous work (Titov and Khoddam, 2015) considers induction of semantic roles of verbal arguments (e.g., an agent, a performer of an action vs. a patient, an affected entity), though no grouping of predicates into relations was considered. We refer to such models as *discrete-state autoencoders*.

We use different model families for the decoding and reconstruction components. The encoding part is a log-linear feature-rich model, while the reconstruction part is a tensor (or matrix) factorization

model which seeks to reconstruct entities, relying on the outcome of the encoding component.

4.1 Encoding component

The encoding component, that is, the actual relation extractor that will be used to process new sentences, is a feature-rich classifier that, given a set of features extracted from the sentence, predicts the corresponding semantic relation $r \in \mathcal{R}$. We use a log-linear model ('softmax regression')

$$q(r|x, \mathbf{w}) = \frac{\exp(\mathbf{w}^T \mathbf{g}(r, x)))}{\sum_{r' \in \mathcal{R}} \exp(\mathbf{w}^T \mathbf{g}(r', x))}, \quad (1)$$

where $\mathbf{g}(r, x)$ is a high-dimensional feature representation and $\mathbf{w}$ is the corresponding vector of parameters. In principle, the encoding model can be any model as long as the relation posteriors $q(r|x, \mathbf{w})$ and their gradients can be efficiently computed or approximated. We discuss the features we use in the experimental section (Section 5).

4.2 Reconstruction component

In the reconstruction component (i.e., decoder), we seek to predict an entity $e_i \in \mathcal{E}$ in a specific position $i \in \{1, 2\}$ given the relation r and another entity e_{-i}, where e_{-i} denotes the complement $\{e_1, e_2\} \backslash \{e_i\}$. Note that this model does not have access to any features of the sentence; this is crucial since in this way we ensure that all the essential information is encoded by the relation variable. This bottleneck forces the learning algorithm to induce informative relations rather than cluster relation occurrences in a random fashion or assign them all to the same relation.

To simplify our notation, let us assume that we predict e_1; the model for e_2 will be analogous. We write the conditional probability models in the following form

$$p(e_1|e_2, r, \theta) = \frac{\exp(\psi(e_1, e_2, r, \theta))}{\sum_{e' \in \mathcal{E}} \exp(\psi(e', e_2, r, \theta))}, \quad (2)$$

where $\mathcal{E}$ is the set of all entities; ψ is a general scoring function which, as we will show, can be instantiated in several ways; θ represents its parameters. The actual set of parameters represented by θ will depend on the choice of scoring function. However, in all the cases we consider in this paper, the parameters will include entity embeddings ($\mathbf{u}_e \in \mathbb{R}^d$ for every $e \in \mathcal{E}$). These embeddings will be learned within our model.

In this work we explore three different factorizations ψ for the decoding component: a tensor factorization model inspired by previous work on relation factorization, a simple selectional-preference model which scores each argument independently of the other, and a third model which is a combination of the first two.

4.2.1 ψ^{RS}: RESCAL

The first reconstruction model we consider is RESCAL, a model very successful in the relational modeling context (Nickel et al., 2011; Chang et al., 2014). It is a restricted version of the classic Tucker tensor decomposition (Tucker, 1966; Kolda and Bader, 2009) and is defined as

$$\psi^{RS}(e_1, e_2, r, \theta) = \mathbf{u}_{e_1}^T C_r \mathbf{u}_{e_2}, \quad (3)$$

where $\mathbf{u}_{e_1}, \mathbf{u}_{e_2} \in \mathbb{R}^d$ are the entity embeddings corresponding to the entities e_1 and e_2. $C_r \in \mathbb{R}^{d \times d}$ is a matrix associated with the latent semantic relation r; it evaluates (i.e., scores) the compatibility between the two arguments of the relation.

4.2.2 ψ^{SP}: Selectional preferences

The second factorization ψ^{SP} scores how well each argument fits the selectional preferences of a given relation r

$$\psi^{SP}(e_1, e_2, r, \theta) = \sum_{i=1}^{2} \mathbf{u}_{e_i}^T \mathbf{c}_{ir}, \quad (4)$$

where $\mathbf{c}_{1r}$ and $\mathbf{c}_{2r} \in \mathbb{R}^d$ encode selectional preferences for the first and second argument of the relation r, respectively. This factorization is also known as model E in Riedel et al. (2013). In contrast to the previous model, it does not model the interaction between arguments: it is easy to see that $p(e_1|e_2, r, \theta)$ for this model (expression (2)) does not depend on e_2 (i.e., on $\mathbf{u}_{e_2}$ and $\mathbf{c}_{2r}$). Consequently, such a decoder would be more similar to generative models of relations which typically assume that arguments are conditionally independent (Yao et al., 2011). Note however that our joint model can still capture argument interdependencies in the encoding component. Still, this approach does not fully implement the desiderata described in the previous section, so

we generally expect this model to be weaker on reasonably-sized collections (this hypothesis will be confirmed in our experimental evaluation).

4.2.3 ψ^{HY}: Hybrid model

The RESCAL model may be too expressive to be accurately estimated for infrequent relations, whereas the selectional preference model cannot, in turn, capture interdependencies between arguments. Thus it seems natural to hope that their combination ψ^{HY} will be more accurate overall:

$$\psi^{HY}(e_1, e_2, r, \theta) = \mathbf{u}_{e_1}^T C_r \mathbf{u}_{e_2} + \sum_{i=1}^{2} \mathbf{u}_{e_i}^T \mathbf{c}_{ir}. \quad (5)$$

This model is very similar to the tensor factorization approach proposed in Socher et al. (2013).

4.3 Learning

We first provide an intuition behind the objective we optimize. We derive it more formally in the subsequent section, where we show that it can be regarded as a variational lower bound on pseudolikelihood (Section 4.3.1). As the resulting objective is still computationally expensive to optimize (due to a summation over all potential entities), we introduce further approximations in Section 4.3.2.

The parameters of the encoding and decoding components (i.e., $\mathbf{w}$ and θ) are estimated jointly. Our general idea is to optimize the quality of argument prediction while averaging over relations

$$\sum_{i=1}^{2} \sum_{r \in \mathcal{R}} q(r|x, \mathbf{w}) \log p(e_i|e_{-i}, r, \theta). \quad (6)$$

Though this objective seems natural, it has one serious drawback: the induced posteriors $q(r|x, \mathbf{w})$ end up being extremely sharp which, in turn, makes the search algorithm more prone to getting stuck in local minima. As we will see in the experimental results, this version of the objective results in lower average performance. This behaviour can be explained by drawing connections with variational inference. Roughly speaking, direct optimization of the above objective behaves very much like using hard EM for generative latent-variable models. Intuitively, one solution is, instead of optimizing expression (6), to consider an entropy-regularized version that favours more uniform posterior distributions $q(r|x, \mathbf{w})$

$$\sum_{i=1}^{2} \sum_{r \in \mathcal{R}} q(r|x,\mathbf{w}) \log p(e_i|e_{-i},r,\theta) + H(q(\cdot|x,\mathbf{w})), \quad (7)$$

where the last term H denotes the entropy over q. The entropy term can be seen as posterior regularization (Ganchev et al., 2010) which pushes the posterior $q(r|x, \mathbf{w})$ to be more uniform. As we will see in a moment, this approach can be formally justified by drawing connections to variational inference (Jaakkola and Jordan, 1996) and, more specifically, to variational autoencoders (Kingma and Welling, 2014).

4.3.1 Variational inference

This subsection presents a justification for the objectives (6) and (7); however, a reader not interested in this explanation can safely skip it and proceed directly to Section 4.3.2.

For the moment let us assume that we perform generative modeling, and we consider optimization of the following pseudo-likelihood (Besag, 1975) objective

$$\sum_{i=1}^{2} \log \sum_{r} p(e_i|e_{-i}, r, \theta) p_u(r), \quad (8)$$

where $p_u(r)$ is the uniform distribution over relations. Note that currently the encoding model is not part of this objective. The pseudo-likelihood (by Jensen's inequality) can be lower-bounded by the following *variational bound*

$$\sum_{i=1}^{2} \sum_{r \in \mathcal{R}} q_i(r) \log p(e_i|e_{-i}, r, \theta) p_u(r) + H(q_i), \quad (9)$$

where q_i is an arbitrary distribution over relations. Note that $p_u(r)$ can be dropped from the expression as it corresponds to a constant with respect to the choice of both the variational distributions q_i and the (reconstruction) model parameters θ.

In variational inference, the maximization of the original (pseudo-)likelihood objective (8) is replaced with the maximization of expression (9) both with respect to q_i and θ. This is typically achieved with an EM-like step-wise procedure: steps where q_i is

selected for a given θ are alternated with steps where the parameters θ are updated while keeping q_i fixed. One idea, introduced by Kingma and Welling (2014) for the continuous case, is to replace the search for an optimal q_i with a predictor (a classifier in our discrete case) trained within the same optimization procedure. Our encoding model $q(r|x, \mathbf{w})$ is exactly such a predictor. With these two modifications (dropping the nuisance term p_u and replacing q_i with $q(r|x, \mathbf{w})$), we obtain the objective (7).

4.3.2 Approximation

The objective (7) cannot be efficiently optimized in its exact form as the partition function of expression (2) requires the summation over the entire set of possible entities $\mathcal{E}$. In order to deal with this challenge we rely on the negative sampling approach of Mikolov et al. (2013). Specifically we avoid the softmax in expression (2) and substitute $\log p(e_1|e_2, r, \theta)$ in the objective (7) with the following expression

$$\log \sigma(\psi(e_1, e_2, r, \theta)) + \sum_{e_1^{neg} \in S} \log \sigma(-\psi(e_1^{neg}, e_2, r, \theta)),$$

where S is a random sample of n entities from the distribution of entities in the collection and σ is the sigmoid function. Intuitively, this expression pushes up the scores of arguments seen in the text and pushes down the scores of 'negative' arguments. When there are multiple entities e_1 which satisfy the relation r with e_2 (for example, `Natasha Obama` and `Malia Ann Obama`, in relation *CHILD_OF* with `Barack Obama`) the scores for all such entities will be pushed up. Assuming both daughters are mentioned with a similar frequency, they will get similar scores. Generally, arguments more frequently mentioned in text will get higher scores.

In the end, instead of directly optimizing expression (7), we use the following objective

$$\sum_{i=1}^{2} \mathbb{E}_{q(\cdot|x,\mathbf{w})} \big[\log \sigma(\psi(e_i, e_{-i}, r, \theta)) + \sum_{e_i^{neg} \in S} \log \sigma(-\psi(e_i^{neg}, e_{-i}, r, \theta)) \big] + \alpha H(q(\cdot|x, \mathbf{w})), \quad (10)$$

where $\mathbb{E}_{q(\cdot|x,\mathbf{w})} \big[\dots \big]$ denotes an expectation computed with respect to the encoder distribution $q(r|x,\mathbf{w})$. Note the non-negative parameter α: after substituting the softmax with the negative sampling term, the entropy parameter and the expectation are not on the same scale anymore. Though we could try estimating the scaling parameter α, we chose to tune it on the validation set.

The gradients of the above objective can be calculated using backpropagation. With the proposed approximation, the computation of the gradients is quite efficient since the reconstruction model has a fairly simple form (e.g., bilinear) and learning the encoder is no more expensive than learning a supervised classifier. We used AdaGrad (Duchi et al., 2011) as an optimization algorithm.

5 Experiments

In this section we evaluate how effective our model is in discovering relations between pairs of entities in a sentence. We consider the unsupervised setting, so we use clustering measures for evaluation.

Since we want to directly compare to Rel-LDA (Yao et al., 2011), we use the transductive set-up: we train our model on the entire training set (with labels removed) and we evaluate the estimated model on a subset of the training set. Given that we train the relation classifier (i.e., the encoding model), unlike some of the previous approaches, there is nothing in our approach which prevents us from applying it in an inductive scenario (i.e., to unseen data).

Towards the end of this section we also provide qualitative evaluation of the induced relations and entity embeddings.

5.1 Data and evaluation measures

We tested our model on the New York Times corpus (Sandhaus, 2008) using articles from 2000 to 2007. We use the same filtering and preprocessing steps (POS tagging, NER, and syntactic parsing) as the ones described in Yao et al. (2011). In that way we obtained about 2 million entity pairs (i.e., potential relation realizations).

In order to evaluate our models, we aligned each entity pair with Freebase, and, as in Yao et al. (2012), we discarded unaligned ones from the eval-

uation. We consider Freebase relations as gold-standard clusterings and evaluated induced relations against them. Note that we use the micro-reading scenario (Nakashole and Mitchell, 2014), that is, we predict a relation on the basis of a single occurrence of an entity pair rather than aggregating information across all the occurrences of the pair in the corpus. Though it is likely to harm our performance when evaluating against Freebase, this is a deliberate choice as we believe extracting relations about less frequent entities (where there is little redundancy in a collection) and modelling content of specific documents is a more challenging and important research direction. Moreover, feature-rich models are likely to be especially beneficial in these scenarios, as for micro-reading the information extraction systems cannot fall back to easier non-ambiguous contexts.

We use the B^3 metric (Bagga and Baldwin, 1998) as the scoring function. B^3 is a standard measure for evaluating precision and recall of clustering tasks (Yao et al., 2012). As the final evaluation score we use F_1, the harmonic mean of precision and recall.

5.2 Features

The crucial characteristic of the learning method we propose is the ability to handle a rich (and overlapping) set of features. With this in mind we adopted the following set of features:

1. bag of words between e_1 and e_2;
2. the surface form of e_1 and e_2;
3. the lemma of the 'trigger'[2] (i.e., for the passage *Microsoft is based in Redmond*, the trigger is *based* and its lemma is *base*);
4. the part-of-speech sequence between e_1 and e_2;
5. the entity type of e_1 and e_2 (as a pair);
6. the entity type of e_1;
7. the entity type of e_2;
8. words on the syntactic dependency path between e_1 and e_2, i.e., the lexicalized path between the entities stripped of dependency labels and their direction.

[2]We define triggers as in Yao et al. (2011), namely "all the words on the dependency path except stop words".

For example, from the sentence

`Stephen Moore`*, director of fiscal policy studies at the conservative* `Cato Institute`,

we would extract the following features:

1. BOW:director, BOW:of, BOW:fiscal, BOW:policy, BOW:studies, BOW:at, BOW:the;
2. E1:Stephen_Moore, E2:Cato_Institute;
3. Trigger:director;
4. PoS:NN_IN_JJ_NN_NNS_IN_DT_JJ;
5. PairType:PERSON_ORGANIZATION;
6. E1Type:PERSON;
7. E2Type:ORGANIZATION;
8. Path:director_at.

5.3 Parameters and baselines

All model parameters $(\mathbf{w}, \theta)$ were initialized randomly. The embedding dimensionality d was set to 30. We induced 100 relations, the same as used for Rel-LDA in Yao et al. (2011). We also set the mini batch size to 100, the initial learning rate of AdaGrad to 0.1 and the number of negative samples n to 20. The results reported in Table 1 are average results of three runs obtained after 5 iterations over the entire training set. For each model we tuned the weight for the L2 regularization penalty and chose 0.1 as it worked well across all the models. We tuned the α coefficient (i.e., the weight for the entropy term) for each model: we chose 0.25 for RESCAL, 0.01 for the selectional preferences, and 0.1 for the hybrid model. All model selection was performed on a validation set: we selected a random 20% of the entire dataset, and considered all entity pairs aligned to Freebase. The final evaluation was done on the remaining 80%.

In order to compare our models with the state of the art in unsupervised RE, we used as a baseline the Rel-LDA model introduced in Yao et al. (2011). Rel-LDA is an application of the LDA topic model (Blei et al., 2003) to the relation discovery task. In Rel-LDA topics correspond to latent relations, and, instead of relying on words as LDA does,

RESCAL	Selectional Pref.	Hybrid	Rel-LDA (our feats)	Rel-LDA (Yao et al., 2012) feats	HAC (DIRT)
34.5 ± 1.3	33.4 ± 1.1	$\mathbf{35.8} \pm 2.0$	29.6 ± 0.9	26.3 ± 0.8	28.3

Table 1: Average F_1 results (%), and the standard deviation, across 3 runs of different models on the test set.

Rel-LDA uses predefined features, including argument words. In a similar fashion to our selectional-preference decoder, it assumes that arguments are conditionally independent given the relation. As another baseline, following Yao et al. (2012), we used hierarchical agglomerative clustering (HAC). This baseline is very similar to the standard unsupervised relation extraction method DIRT (Lin and Pantel, 2001). The HAC cut-off parameter was set to 0.95 based on the development set performance. We used the same feature representation for all the models, including the baselines. We also report results of Rel-LDA using the features from Yao et al. (2012).[3]

5.4 Results and discussion

The results we report in Table 1 are mean and standard deviations across 3 runs with different random initialization of the parameters (except for the deterministic HAC approach). First, we can observe that using richer features is beneficial for the generative baseline. It leads to a substantial improvement in F_1 (from 26.3% to 29.6% F_1). The HAC baseline is outperformed by Rel-LDA (28.3% vs. 29.6% F_1). However, all our proposed models substantially outperform all 3 baselines: the best result is 35.8% F_1.

The selectional preference model on average performs better than the best baseline (33.4% vs. 29.6% F_1). As we predicted in Section 4, compared with the RESCAL model, the selectional preference model has slightly lower performance (34.5% vs. 33.4% F_1). This is not surprising as the argument independence assumption is very strong, and the general motivation we provided in Section 2 does not really apply to the selectional preference model.

Combining RESCAL and selection preference models, as we expected, gives some advantage in terms of performance. The hybrid model is the best performing model with 35.8% F_1, and it is, on average, 6.2% more accurate than Rel-LDA.

The introduction of entropy in expression (7) does not only add an extra justification to the objective we optimize, but also helps to improve the models' performance. In fact, as shown in Figure 2 for the Hybrid model, the difference between having or not the entropy term makes a big difference, going from 23.9% without regularization to 34.3% F_1 with regularization. Note that the method is quite stable within the range $\alpha \in [0.1, 1]$, and more fine-grained tuning of α seems only mildly beneficial. However the performance with small values of α (0.01) is more problematic: Hybrid both does not outperform Rel-LDA and has a large variance across runs. Somewhat counter-intuitively, with $\alpha = 0$ (no entropy regularization) the variance is almost negligible. However, given the low performance in this regime, it probably just means that we get consistently stuck in equally bad local minima.

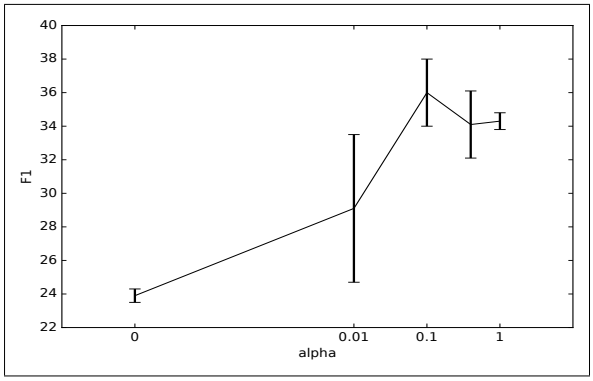

Figure 2: Results of the hybrid model on the validation set, with different α.

Though it may seem that the need to tune the entropy term weight is an unfortunate side effect of using the non-probabilistic objective from Section 4.3.2, the reality is more subtle. In fact, even for fully probabilistic variational autoencoders with real-valued states y, using the weight of 1, as prescribed by their variational interpretation (see Section 4.3.1), does not lead to stable performance (Bowman et al., 2016). Instead, annealing over α seems necessary. Though annealing is likely

[3] Yao et al. (2012) is a follow-up work for Yao et al. (2011).

Relation 66	Relation 62	Relation 19
president	review	professor
director	review_restaurant	dean
chairman	review_production	graduate
executive	review_book	director
spokesman	review_performance	specialist
manager	column_review	attend
analyst	review_concert	expert
owner	review_revival	professor_study
professor	review_rise	chairman

Table 2: Relation clusters ordered from left to right by their frequency.

to benefit our method as well, we leave it for future work.

Since the proposed model is unsupervised, it is interesting to inspect the relations induced by our best model. In order to do so, we select the most likely relation according to our relation extractor (i.e., encoding model) for every context in the validation set and then, for every relation, we count occurrences of every trigger. The most-frequent trigger for three induced relations are presented in Table 2. Relation 62 encodes the relation *REVIEWED* (not present in Freebase), as in

Anthony Tommasini reviews Metropolitan Opera's production of Cosi Fan Tutte.

Clusters 19 and 66 are examples of more coarse relations. Relation 19 represents a broader *ACADEMIC* relation, as in the passage

Dr. Susan Merritt, dean of the School of Computer Science and Information Systems.

or as in the passage

George Myers graduated from Yale University.

Cluster 66 instead groups together expressions such as *leads* or *president (of)*, so it can vaguely be described as a *LEADERSHIP* relation, but it also contains the relation triggered by the word *professor (in)*. In fact, this is the most frequent relation induced by our model. We can check further by looking at the learned embeddings of named entities visualized with the t-SNE algorithm (Van der Maaten and Hinton, 2008). In Figure 3, we can see that entities representing universities and non-academic organizations end up being very close in the embedding space. This artefact is likely to be related to the coarseness of Relations 66 and 19, though it does not provide a good explanation for why this has happened, since the entity embeddings are also induced within our model.

Semi-sup RESCAL	**62.3**
Semi-sup Selectional Pref.	58.1
Semi-sup Hybrid	61.5
Unsup Hybrid	34.3

Table 3: Average F_1 results (%) for semi-supervised and unsupervised models, across 3 runs of different models tested on Te.

However, overlaps in embeddings do not seem to be a general problem: the t-SNE visualization shows that most entities are well clustered into fine-grained types, for example, football teams, nations, and music critics.

5.5 Decoder influence

In order to examine the influence of the decoder on the model performance, we performed additional experiments in a more controlled setting. We reduced the dataset to entity pairs participating in Freebase relations, ending up with a total of about 42,000 relation realizations. We randomly split the dataset in two. We used the first half as a test set Te, while we used the second half as a training set Tr. We further randomly split the training set Tr in two parts, Tr_1 and Tr_2. We use Tr_1 as a (distantly) labeled dataset to learn only the decoding part for each proposed model. To make it comparable to our unsupervised models with 100 induced relations, we trained the decoder on the 99 most frequent Freebase relations plus a further *OTHER* relation, which is a union of the remaining less frequent relations. This approach is similar to the KB factorization adopted in Bordes et al. (2011). With the decoder learned and fixed, we trained the encoder part on unlabeled examples in Tr_2, while leveraging the previously trained decoder. In other words, we optimize the objective (10) on Tr_2 but update only the encoder parameters $\mathbf{w}$.[4] In this setting the decoder provides a learning signal for the encoder. The better the gen-

[4]We also update embeddings of entities not appearing in Tr_1.

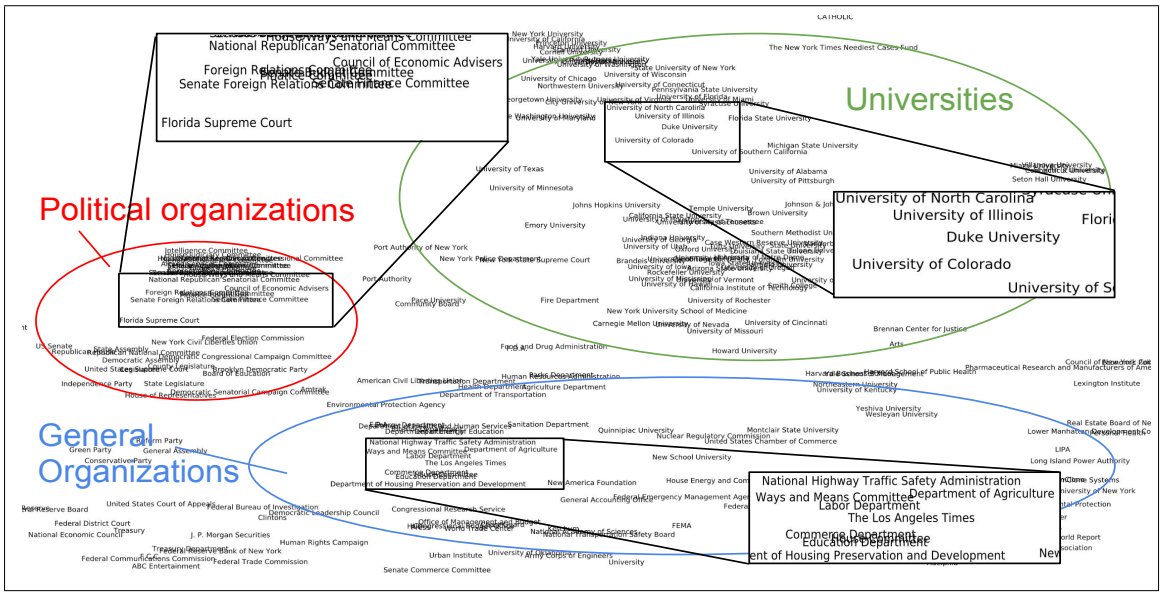

Figure 3: t-SNE visualization of entity embeddings learned during the training process.

eralization properties of the decoder are, the better the resulting encoder should be. We expect more expressive decoders (i.e., RESCAL and Hybrid) to be able to capture relations better than the selectional preference model and, thus, yield better encoders. In order to have a lower bound for the semi-supervised models, we also trained our best model from the previous experiments (Hybrid) on Tr_2 in a fully unsupervised way. All models are tested on the test set Te.

As expected, all models with a supervised decoder are much more accurate than the unsupervised model (Table 3). The best results with a supervised decoder are obtained by the RESCAL model with 62.3% F_1, while the result of the unsupervised hybrid model is 34.3% F_1. As expected the RESCAL and Hybrid outperform the selectional preference model in this setting (62.3% and 61.5% vs. 58.1% F_1 respectively). Somewhat surprisingly, the RESCAL model is slightly more accurate (0.8% F_1) than the hybrid model. These experiments confirm that more accurate decoder models lead to better performing encoders. The results also hint at a potential extension of our approach to a more realistic semi-supervised setting, though we leave any serious investigation of this set-up for future work.

6 Additional Related Work

In this section, we mainly consider lines of related work not discussed in other sections of the paper, and we emphasize their relationship to our approach.

Distant supervision. These methods can be regarded as a half-way point between unsupervised and supervised methods. Distantly supervised models are trained on data generated automatically by aligning knowledge bases (e.g., Freebase and Wikipedia infoboxes) with text (Mintz et al., 2009; Riedel et al., 2010; Surdeanu et al., 2012; Zeng et al., 2015). Similarly to our method they can use feature-rich models without the need for manually annotated data. However, a relation extractor trained in this way will only be able to predict relations already present in a knowledge base. These methods cannot be used to discover new relations. The framework we propose is completely unsupervised and does not have this shortcoming.

Bootstrapping. Bootstrapping methods for relation extraction (Agichtein and Gravano, 2000; Brin, 1998; Batista et al., 2015) iteratively label new examples by finding the ones which are the most similar, according to some similarity function, to a seed set of labeled examples. The process continues until some convergence criteria is met. Even though this approach is not very labor-intensive (i.e., it requires

only few manually labeled examples for the initial seed set), it requires some domain knowledge from the model designer. In contrast, unsupervised models are domain-agnostic and require only unlabeled text.

Knowledge base factorization. Knowledge base completion via matrix or tensor factorization has received a lot of attention in the past few years (Bordes et al., 2011; Jenatton et al., 2012; Weston et al., 2013; Bordes et al., 2013; Socher et al., 2013; García-Durán et al., 2014; Bordes et al., 2014; Lin et al., 2015b; Chang et al., 2014; Nickel et al., 2011). But in contrast to what we propose here, namely, induction of new relations, these models factorize relations already present in knowledge bases.

Universal schema methods (Riedel et al., 2013) use factorization models to infer facts (e.g., predict missing entities), but they do not attempt to induce relations. In other words, they consider each given context as a relation and induce an embedding for each of them. They do not attempt to induce a clustering over the contexts. Our work can be regarded as an extension of these methods.

Autoencoders with discrete states. Aside from the work cited above (Daumé III, 2009; Ammar et al., 2014; Titov and Khoddam, 2015; Lin et al., 2015a), we are not aware of previous work using autoencoders with discrete states (i.e., a categorical latent variable or a graph). The semisupervised version of variational autoencoders (Kingma et al., 2014) used a combination of a real-valued vector and a categorical variable as its hidden representation and yielded impressive results on the MNIST image classification task. However, their approach cannot be directly applied to unsupervised classification, as there is no reason to believe that latent classes would be captured by the categorical variable rather than in some way represented by the real-valued vector.

The only other application of variational autoencoders to natural language is the very recent work of Bowman et al. (2016). They study language modeling with recurrent language models and consider only real-valued vectors as states.

Generative models with rich features have also been considered in the past (Berg-Kirkpatrick et al., 2010). However, autoencoders are potentially more flexible than generative models as they can use very different encoding and decoding components and can be faster to train.

7 Conclusions and Discussion

We presented a new method for unsupervised relation extraction.[5] The model consists of a feature-rich classifier that predicts relations, and a tensor factorization component that relies on the predicted relations to infer left-out arguments. These models are jointly estimated by optimizing the argument reconstruction objective.

We studied three alternative factorization models building on ideas from knowledge base factorization and selectional preference modeling. We empirically showed that our factorization models yield relation extractors that are more accurate than state-of-the-art generative and agglomerative clustering baselines.

As the proposed modeling framework is quite flexible, the model can be extended in many different ways. Our approach can be regarded as learning semantic representations that are informative for basic inference tasks (in our case, the inference task was recovering individual arguments). More general classes of inference tasks can be considered in future work. Moreover, it would be interesting to evaluate the proposed model on how accurately it infers these facts (rather than only on the quality of the induced latent representations). The work presented in this paper can also be combined with the approach of Titov and Khoddam (2015) to induce both relations and semantic roles (i.e., essentially to induce semantic frames (Fillmore, 1976)). Another potential direction is the use of labeled data: our feature-rich model (namely its discriminative encoding component) is likely to have much better asymptotic performance than its generative counterpart, and, consequently, labeled data should be much more beneficial.

Acknowledgments

This work is supported by NWO Vidi Grant 016.153.327, Google Focused Award on Natural Language Understanding and partially supported by ISTI Grant for Young Mobility. The authors thank

[5] github.com/diegma/relation-autoencoder

the action editor and the anonymous reviewers for their valuable suggestions and Limin Yao for answering our questions about data and baselines.

References

Eugene Agichtein and Luis Gravano. 2000. *Snowball*: Extracting relations from large plain-text collections. In *5th ACM Conference on Digital Libraries*.

Waleed Ammar, Chris Dyer, and Noah A. Smith. 2014. Conditional random field autoencoders for unsupervised structured prediction. In *NIPS*.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary G. Ives. 2007. Dbpedia: A nucleus for a web of open data. In *6th International Semantic Web Conference (ISWC)*.

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *LREC*.

Michele Banko and Oren Etzioni. 2008. The tradeoffs between open and traditional relation extraction. In *ACL*.

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *IJCAI*.

David S. Batista, Bruno Martins, and Mário J. Silva. 2015. Semi-supervised bootstrapping of relationship extractors with distributional semantics. In *EMNLP*.

Taylor Berg-Kirkpatrick, Alexandre Bouchard-Côté, John DeNero, and Dan Klein. 2010. Painless unsupervised learning with features. In *HLT - NAACL*.

Julian Besag. 1975. Statistical analysis of non-lattice data. *The statistician*, pages 179–195.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Kurt D. Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *SIGMOD*.

Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. 2011. Learning structured embeddings of knowledge bases. In *AAAI*.

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Irreflexive and hierarchical relations as translations. In *Structured Learning: Inferring Graphs from Structured and Unstructured Inputs (SLG-ICML)*.

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

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Józefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *ICLR*.

Sergey Brin. 1998. Extracting patterns and relations from the world wide web. In *The World Wide Web and Databases Workshop (WebDB)*.

Kai-Wei Chang, Wen-tau Yih, Bishan Yang, and Christopher Meek. 2014. Typed tensor decomposition of knowledge bases for relation extraction. In *EMNLP*.

Hal Daumé III. 2009. Unsupervised search-based structured prediction. In *ICML*.

Oier Lopez de Lacalle and Mirella Lapata. 2013. Unsupervised relation extraction with general domain knowledge. In *EMNLP*.

John C. Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159.

Charles J. Fillmore. 1976. Frame semantics and the nature of language. *Annals of the New York Academy of Sciences*, 280(1):20–32.

Jenny Rose Finkel, Trond Grenager, and Christopher D. Manning. 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. In *ACL*.

Kuzman Ganchev, Joao Graca, Jennifer Gillenwater, and Ben Taskar. 2010. Posterior regularization for structured latent variable models. *Journal of Machine Learning Research*, 11:2001–2049.

Alberto García-Durán, Antoine Bordes, and Nicolas Usunier. 2014. Effective blending of two and three-way interactions for modeling multi-relational data. In *European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD)*.

Lise Getoor and Ben Taskar. 2007. *Introduction to statistical relational learning*. MIT press.

Geoffrey E. Hinton. 1989. Connectionist learning procedures. *Artificial Intelligence*, 40(1-3):185–234.

Tommi S. Jaakkola and Michael I. Jordan. 1996. Computing upper and lower bounds on likelihoods in intractable networks. In *12th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*.

Rodolphe Jenatton, Nicolas Le Roux, Antoine Bordes, and Guillaume Obozinski. 2012. A latent factor model for highly multi-relational data. In *NIPS*.

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational Bayes. In *ICLR*.

Diederik P. Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. 2014. Semi-supervised learning with deep generative models. In *NIPS*.

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

Dekang Lin and Patrick Pantel. 2001. DIRT - discovery of inference rules from text. In *SIGKDD*.

Chu-Cheng Lin, Waleed Ammar, Chris Dyer, and Lori S. Levin. 2015a. Unsupervised POS induction with word embeddings. In *NAACL HLT*.

Yankai Lin, Zhiyuan Liu, Huan-Bo Luan, Maosong Sun, Siwei Rao, and Song Liu. 2015b. Modeling relation paths for representation learning of knowledge bases. In *EMNLP*.

Xitong Liu, Fei Chen, Hui Fang, and Min Wang. 2014. Exploiting entity relationship for query expansion in enterprise search. *Information Retrieval*, 17(3):265–294.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *ICLR*.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *ACL*.

Ndapandula Nakashole and Tom M. Mitchell. 2014. Micro reading with priors: Towards second generation machine readers. In *AKBC at NIPS*.

Ndapandula Nakashole, Gerhard Weikum, and Fabian M. Suchanek. 2012. PATTY: A taxonomy of relational patterns with semantic types. In *EMNLP*.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *ICML*.

Diarmuid Ó Séaghdha. 2010. Latent variable models of selectional preference. In *ACL*.

Hoifung Poon and Pedro M. Domingos. 2009. Unsupervised semantic parsing. In *EMNLP*.

Deepak Ravichandran and Eduard H. Hovy. 2002. Learning surface text patterns for a question answering system. In *ACL*.

Philip Resnik. 1997. Selectional preference and sense disambiguation. In *ACL SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How*.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *ECML-PKDD*.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M. Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *NAACL*.

Evan Sandhaus. 2008. The New York Times annotated corpus. *Linguistic Data Consortium, Philadelphia*, 6(12).

Yusuke Shinyama and Satoshi Sekine. 2006. Preemptive information extraction using unrestricted relation discovery. In *NAACL HLT*.

Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Y. Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *NIPS*.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: A core of semantic knowledge. In *WWW*.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D. Manning. 2012. Multi-instance multi-label learning for relation extraction. In *EMNLP-CoNLL*.

Ilya Sutskever, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. 2009. Modelling relational data using Bayesian clustered tensor factorization. In *NIPS*.

Idan Szpektor, Hristo Tanev, Ido Dagan, and Bonaventura Coppola. 2004. Scaling web-based acquisition of entailment relations. In *EMNLP*.

Ivan Titov and Ehsan Khoddam. 2015. Unsupervised induction of semantic roles within a reconstruction-error minimization framework. In *NAACL*.

Ivan Titov and Alexandre Klementiev. 2011. A Bayesian model for unsupervised semantic parsing. In *ACL*.

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

Tim Van de Cruys. 2010. A non-negative tensor factorization model for selectional preference induction. *Journal of Natural Language Engineering*, 16(4):417–437.

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

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *ICML*.

Jason Weston, Antoine Bordes, Oksana Yakhnenko, and Nicolas Usunier. 2013. Connecting language and knowledge bases with embedding models for relation extraction. In *EMNLP*.

Limin Yao, Aria Haghighi, Sebastian Riedel, and Andrew McCallum. 2011. Structured relation discovery using generative models. In *EMNLP*.

Limin Yao, Sebastian Riedel, and Andrew McCallum. 2012. Unsupervised relation discovery with sense disambiguation. In *ACL*.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *EMNLP*.