AUTOENCODING FOR JOINT RELATION FACTORIZATION AND DISCOVERY FROM TEXT

Diego Marcheggiani and Ivan Titov
Institute for Logic, Language and Computation
University of Amsterdam
{marcheggiani,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. We use a variational autoencoding objective and estimate the two components jointly so as to minimize errors in recovering arguments. We study factorization models inspired by previous work in relation factorization. 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 semantic relations present between two entities in text. For example, in the sentence

Ebert is the first journalist to win the Pulitzer prize. (1)

the extractor should predict the semantic relation r = *AWARDED* expressed between the entities e_1 = `Ebert` and e_2 = `Pulitzer prize`.

Existing methods for RE either do not scale well on open-domain scenarios (e.g., the entire Web) like supervised approaches, or rely on simple features and make strong modeling assumptions, like generative and agglomerative clustering models (Lin & Pantel, 2001; Yao et al., 2011).

In this work, we introduce a new model for unsupervised relation extraction that takles the aforementioned challenges. 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 learning to predict left-out arguments, the inference algorithm will search for latent relations which simplify the argument prediction task as much as possible. Roughly, such objective will favour inducing relations which 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.

Interestingly, the use of reconstruction-error objective, previously considered primarily in the context of training neural autoencoders (Hinton, 1989), gives us an opportunity to borrow ideas from the area of relation factorization (Bordes et al., 2011; Riedel et al., 2013). In our work, we also adopt a fairly standard RESCAL factorization (Nickel et al., 2011) and use it within our reconstruction component.

In contrast to relational learning, rather than factorizing existing relations, our method simultaneously discovers the relational schema and a mapping from text to the relations, and it does it in such way as to maximize performance on reconstruction tasks. Unlike generative models, the proposed model is more expressive (by relying on rich contextual features), and makes minimal independence assumptions.

1.1 Our Approach

We approach the problem by introducing a latent variable model which defines the interactions between a latent relations r and the observables: the entity pair (e_1, e_2) and other features of the sentence x. The main idea of latent variable modeling is that a good latent representation is the one which helps us to reconstruct input (i.e., x, including (e_1, e_2)). Thus, it is crucial to design the model in such a way that good r (the one predictive of x) indeed encodes relations rather than some other form of abstraction.

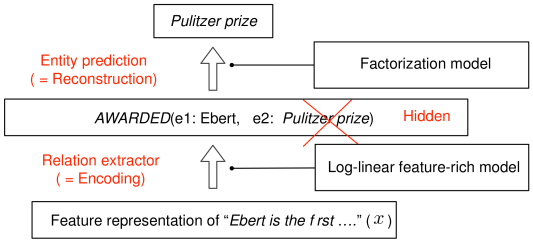

Figure 1: Inducing relations with discrete-state autoencoders.

In our approach, we encode this reconstruction idea very explicitly. As a motivating example, consider sentence 1. As shown in Figure 1, let us assume that we hide one argument: for example, $e_2 = \texttt{Pulitzer prize}$. Now the purpose of the reconstruction component is to reconstruct (i.e., infer) this argument relying on another argument ($e_1 = \texttt{Ebert}$), the latent relations r and nothing else. At the learning time, our inference algorithm will search through the space of potential relation clusterings to find the one which 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.

Though in the previous paragraph, we described it as clustering of patterns, 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 given sentence and an entity pair in this sentence. 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.

2 Reconstruction Error Minimization

In order to implement the desiderata sketched in the previous section, we take an inspiration from neural autoencoders (Hinton, 1989). Though popular within the neural network community (i.e., y is a real-valued vector), they have recently been applied to the discrete-state setting (i.e., y is a categorical random variable) (Daumé III, 2009; Ammar et al., 2014). The most related previous work (Titov & Khoddam, 2015) considers induction of semantics roles of verbal arguments, though no grouping of predicates into relations has been considered. We refer to such models as *discrete-state autoencoders*.

Differently from neural autoencoders, in this work 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.

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

$$q(r|x, \mathbf{w}) \propto \exp(\mathbf{w}^T \mathbf{g}(r, x)), \tag{2}$$

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 relation posteriors $q(r|x, \mathbf{w})$ and their gradients can be efficiently computed or approximated.

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; in this way we ensure that all the essential information is encoded by the relation variable. This bottleneck is needed as it forces the learning algorithm to induce informative relations.

Let us assume that we predict e_1. We write the conditional probability models in the following form

$$p(e_1|e_2, r, \theta) = \frac{\exp(\psi(e_1, e_2, r, \theta))}{\sum_{e' \in \mathcal{E}} \exp(\psi(e', e_2, r, \theta))}, \tag{3}$$

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, 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 for the decoding component: a tensor factorization model, a simple selectional-preference model, and a combination of the two.

ψ^{RS}: RESCAL The first reconstruction model we consider is RESCAL, a model very successful in the relational modeling context (Nickel et al., 2011). It is defined as

$$\psi^{RS}(e_1, e_2, r, \theta) = \mathbf{u}_{e_1}^T C_r \mathbf{u}_{e_2}, \tag{4}$$

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 two arguments of the relation.

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

$$\psi^{SP}(e_1, e_2, r, \theta) = \sum_{i=1}^{2} \mathbf{u}_{e_i}^T \mathbf{c}_{ir}, \tag{5}$$

where $\mathbf{c}_{1r}$ and $\mathbf{c}_{2r} \in \mathbb{R}^d$ encode selectional preferences for each argument of the relation r. Differently from the previous model, it does not model interaction between arguments.

ψ^{HY}: Hybrid model The RESCAL model may be too expressive to be accurately estimated for infrequent relations, whereas the selectional preference model cannot capture interdependencies between arguments, so it seems natural to try their combination ψ^{HY}.

2.1 LEARNING

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 marginalizing our relations

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

where the last term H denotes the entropy. 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. This objective can be formally justified by drawing connections to variational inference (Jaakkola & Jordan, 1996) and, more specifically, to variational autoencoders (Kingma & Welling, 2014).

The objective (6) cannot be efficiently optimized as the partition function of expression (3) 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 (3) and substitute $\log p(e_1|e_2, r, \theta)$ in the objective (6) 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. In the end, instead of directly optimizing the expression (6), we use the following objective

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.

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

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 above approximation, their computation 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.

3 Experiments

In this work we evaluate how effective our model is in discovering relations between pairs of entities in a sentence. 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. We tested our model on the New York Times corpus (Sandhaus, 2008) using articles from 2000 to 2007. We obtained about 2 million entity pairs (i.e., potential relation realizations) after preprocessing. In order to evaluate our models, we aligned each entity pair with Freebase, and evaluated induced relations against gold standard relations in Freebase. As the scoring function, we use the F_1 of the B^3 metric (Bagga & Baldwin, 1998), a standard measure for clustering tasks.

All model parameters $(\mathbf{w}, \theta)$ are initialized randomly. The embedding dimensionality d was set to 30. The number of relations to induce is 100, 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. We compared our models with the Rel-LDA model of Yao et al. (2011) and hierarchical agglomerative clustering (HAC) as in Yao et al. (2012). 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).[1]

3.1 Results and discussion

The results we report on Table 1 are averages across 3 runs with different random initialization of the parameters (except for the deterministic HAC approach), we also report the standard deviation. First, we can observe that all the proposed models substantially outperform all baselines: the best result is 35.8% F_1.

The selectional preference model on average performs better than the baseline (33.4% vs. 29.6% F_1). As we predicted in Section 2, compared with the RESCAL model, the sectional 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. Combining RESCAL and selection preference models, as we expected, gave some advantage in terms of performance. The hybrid model is the best performing model with 35.8% F_1, and it is in average 6.2% more accurate than Rel-LDA.

References

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

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

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

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

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

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

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

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

Tommi S. Jaakkola and Michael I. Jordan. Computing upper and lower bounds on likelihoods in intractable networks. In *UAI*, 1996.

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

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

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

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

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

Evan Sandhaus. The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia*, 6(12), 2008.

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

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

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