Unsupervised Information Extraction: Regularizing Discriminative Approaches with Relation Distribution Losses

Étienne Simon and **Vincent Guigue** and **Benjamin Piwowarski**
Sorbonne Université, CNRS, Laboratoire d'Informatique de Paris 6
LIP6, F-75005 Paris, France
{etienne.simon, vincent.guigue, benjamin.piwowarski}@lip6.fr

Abstract

Unsupervised relation extraction aims at extracting relations between entities in text. Previous unsupervised approaches are either generative or discriminative. In a supervised setting, discriminative approaches, such as deep neural network classifiers, have demonstrated substantial improvement. However, these models are hard to train without supervision, and the currently proposed solutions are unstable. To overcome this limitation, we introduce a skewness loss which encourages the classifier to predict a relation with confidence given a sentence, and a distribution distance loss enforcing that all relations are predicted in average. These losses improve the performance of discriminative based models, and enable us to train deep neural networks satisfactorily, surpassing current state of the art on three different datasets.

1 Introduction

Information extraction models aim at discovering the underlying semantic structure linking entities mentioned in a text. This can be used to build knowledge bases, which are widely used in several applications such as question answering (Yih et al., 2015; Berant et al., 2013), document retrieval (Dalton et al., 2014) and logical reasoning (Socher et al., 2013).

In the relation extraction (RE) task, we are interested in discovering the semantic (binary) relation that holds between two entities mentioned in text. The end goal is to extract triplets of the form (subject, relation, object). A considerable amount of work has been conducted on supervised or weakly-supervised relation extraction (Kambhatla, 2004; Zeng et al., 2015; Lin et al., 2016), with recent state-of-the-art models using deep neural networks (NN).

Developing unsupervised relation extraction models is interesting for three reasons: they (1) do not necessitate labeled data except for validating the models; (2) can uncover new relation types; and (3) can be trained from large unlabeled datasets, and then fine-tuned for specific relations.

The first unsupervised models used a clustering (Hasegawa et al., 2004; Banko et al., 2007) or generative (Yao et al., 2011, 2012) approach. The latter, which obtained state-of-the-art performance, still makes a lot of simplifying hypotheses, such as assuming that the entities are conditionally independent between themselves given the relation. To train more expressive models, a shift to discriminative approaches was necessary. The open question then becomes how to provide a sufficient learning signal to the classifier. To the best of our knowledge, only Marcheggiani and Titov (2016) followed this path by leveraging representation learning for modeling knowledge bases, and proposed to use an auto-encoder model: their encoder extracts the relation from a sentence, that the decoder uses to predict a missing entity. However, their encoder is still limited compared to its supervised counterpart (e.g. Zeng et al. (2015)) and relies on hand-crafted features extracted by natural language processing tools, containing errors and unable to discover new patterns, which might hinder performances.

More importantly, our initial experiments showed that the above model was unstable, especially when using a deep NN relation classifier. It converged to either of the two following regimes, depending on hyper-parameter settings: always predicting the same relation, or predicting a uniform distribution. To overcome these limitations, we propose to use two new losses alongside a link prediction loss based on a fill-in-the-blank task, and show experimentally that this is key to learning deep neural network models. Our contributions are the following:

- We propose two RelDist losses: a skewness loss, which encourages the classifier to pre-

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 1378–1387
Florence, Italy, July 28 - August 2, 2019. ©2019 Association for Computational Linguistics

dict a class with confidence for a single sentence, and a distribution distance loss, which encourages the classifier to scatter a set of sentences into different classes;

- We perform extensive experiments on the usual NYT+FB dataset, as well as two new datasets;
- We show that our RelDist losses allow us to train a deep PCNN classifier (Zeng et al., 2015) as well as improve performance of feature-based models (Marcheggiani and Titov, 2016).

In the following, we first discuss related works (Section 2) before describing our model (Section 3) and presenting experimental results (Section 4).

2 Related work

Relation extraction is a standard language classification task: given a sentence containing two entities, the goal is to predict what is the relation linking these two entities. Most relation extraction systems need to be trained on a labeled dataset. However human annotation is expensive, and virtually impractical when a large number of relations is involved.

As a result, most systems are trained on datasets built through distant supervision (Mintz et al., 2009), a compromise between the supervised and unsupervised settings. It makes the following assumption: if a sentence contains two entities linked in a knowledge base, this sentence necessarily conveys that relation. For example, distant supervision aligns the sentence "*Hubel*$_{e_1}$ *received the Nobel Prize*$_{e_2}$ *for his discovery*" with the triplet (Hubel, award received, Nobel Prize), thus supervising the sentence with the label "award received". The resulting alignment are of a poorer quality, and even though this method can leverage large amounts of unlabeled text, the relation ontology is still fixed by a knowledge base, the resulting model being unable to discover new relations.

In the supervised setting, neural network models have demonstrated substantial improvement over approaches using hand-crafted features. In particular, piecewise convolutional neural networks (PCNN, Zeng et al., 2015) are now widely used as a basis for other improvements, such as the instance-level selective attention mechanism of Lin et al. (2016) which follows the multi-instance multi-label framework (Hoffmann et al., 2011; Surdeanu et al., 2012). The recent NN approaches however need large amount of data to achieve good performances.

In the unsupervised setting, models have no access to annotated sentences or to a knowledge base: other regularity hypotheses have to be made. The resulting models can be categorized into either the generative/clustering or discriminative approaches. The former try to cluster regularities in the text surrounding two entities, while the latter use discriminative models but have to make further hypotheses, namely that a pair of given entities always share the same relation, to provide a learning signal for the classifier.

Among clustering models, one of the earliest work is from Hasegawa et al. (2004) who propose building clusters by using cosine similarity on TF-IDF vectors for the surrounding text. Later, the OpenIE approaches (Banko et al., 2007; Angeli et al., 2015) relied upon the hypothesis that the surface form of the relation conveyed by a sentence appears in the path between the two entities in its dependency tree. However, these latter works are too dependent on the raw surface form and suffer from bad generalization. In our previous example, OpenIE will extract the triplet (Hubel, received, Nobel Prize), but simply replacing "received" by "was awarded" might produce a different relation even though the semantic remains the same.

Related to these clustering approaches, the Rel-LDA models (Yao et al., 2011, 2012) use a generative model inspired by LDA to cluster sentences: each relation defines a distribution over a high-level handcrafted set of features describing the relationship between the two entities in the text (e.g. the dependency path). However, these models are limited in their expressiveness. More importantly, depending on the set of features, they might focus on features not related to the relation extraction task.

We posit that discriminative approaches can help in going further in expressiveness, especially considering recent results with neural network models. To the best of our knowledge, the only discriminative approach to unsupervised relation extraction is the variational autoencoder approach (VAE) proposed by Marcheggiani and Titov, 2016): the encoder extracts the semantic relation from hand-crafted features of the sentence (related to those of Rel-LDA), while the decoder

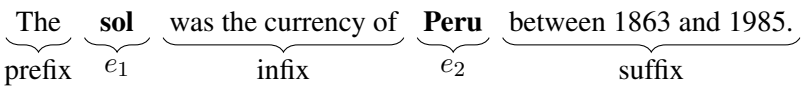

Figure 1: A sentence from Wikipedia where the conveyed relation is "currency used by". We call s the sentence with the two entities removed: $s = (\text{prefix}, \text{infix}, \text{suffix})$.

tries to predict one of the two entities given the relation and the other entity, using a general triplet scoring function (Nickel et al., 2011). This scoring function provides a signal since it is known to predict to some extent relation triplets given their embeddings. Among the input features of the classifiers are the entities themselves, the resulting model can thus be interpreted as an autoencoder where the encoder part benefits from an additional context. The proposed loss, based on the KL divergence between the posterior distribution over relations and a uniform prior on the relation distribution, is very unstable in practice. Our proposed approaches solve this unstability, and allows us to train expressive classifiers such as the PCNN model (Zeng et al., 2015).

3 Model description

Our model focuses on extracting the relation between two entities in textual data, and assumes that a recognition tool has identified named entities in the text. Furthermore, like most works on relation extraction, we limit ourselves to binary relations and therefore consider sentences with two tagged entities, as shown in Figure 1.

To provide a supervision signal to our relation classifier, we follow Marcheggiani and Titov (2016) and use a *fill-in-the-blank* task, i.e. "*The sol$_{e_1}$ was the currency of __?__$_{e_2}$ between 1863 and 1985.*". To correctly fill in the blank, we could directly learn to predict the missing entity, but in this case we would not be able to learn a relation classifier. Instead, we want to first learn that this sentence expresses the semantic relation "currency used by" before using this information for a supervised task:

(i) We suppose that the relation can be predicted by the text surrounding the two entities alone (see Figure 1);

(ii) We then try to predict the missing entity given the predicted relation and the other entity – this gives the supervision signal.

These hypotheses lead to the following formulation of the fill-in-the-blank task:

$$p(e_{-i} \mid s, e_i) = \sum_r \underbrace{p(r \mid s)}_{\text{(i) classifier}} \underbrace{p(e_{-i} \mid r, e_i)}_{\text{(ii) link predictor}} \quad (1)$$

where e_1 and e_2 are the two entities, s is the text surrounding them and r is the relation linking them. As the link predictor can consider either entity, we use e_i to designate the given entity, and $e_{-i} = \{e_1, e_2\} \setminus \{e_i\}$ the one to predict.

The relation classifier $p(r \mid s)$ and link predictor $p(e_{-i} \mid r, e_i)$ are trained jointly to reconstruct a missing entity, but the link predictor cannot access the input sentence directly. Thus, all the required information must be condensed into r, which acts as a bottleneck. We advocate that this information is the semantic relation between the two entities.

Note that Marcheggiani and Titov (2016) did not make our first independence hypothesis. Instead, their classifier is conditioned on both e_i and e_{-i}, strongly relying on the fact that r is an information bottleneck.

In the following, we first describe the relation classifier $p(r \mid s)$ in section 3.1, before introducing the link predictor $p(e_{-i} \mid r, e_i)$ in section 3.2. Arguing that the resulting model is unstable, we describe the two new RelDist losses in section 3.3.

3.1 Unsupervised Relation Classifier

Our model for $p(r \mid s)$ follows current state-of-the-art practices for supervised relation extraction by using a piecewise convolutional neural network (PCNN, Zeng et al., 2015). The input sentence can be split into three parts separated by the two entities (see Figure 1). In a PCNN, the model outputs a representation for each part of the sentence. These are then combined to make a prediction. Figure 2 shows the network architecture that we now describe.

First, each word of s is mapped to a real-valued vector. In this work, we use standard word embedding, initialized with GloVe[1] (Pennington et al., 2014), and fine-tune them during training. Based on those embeddings, a convolutional layer detects

[1] 6B.50d from https://nlp.stanford.edu/projects/glove/

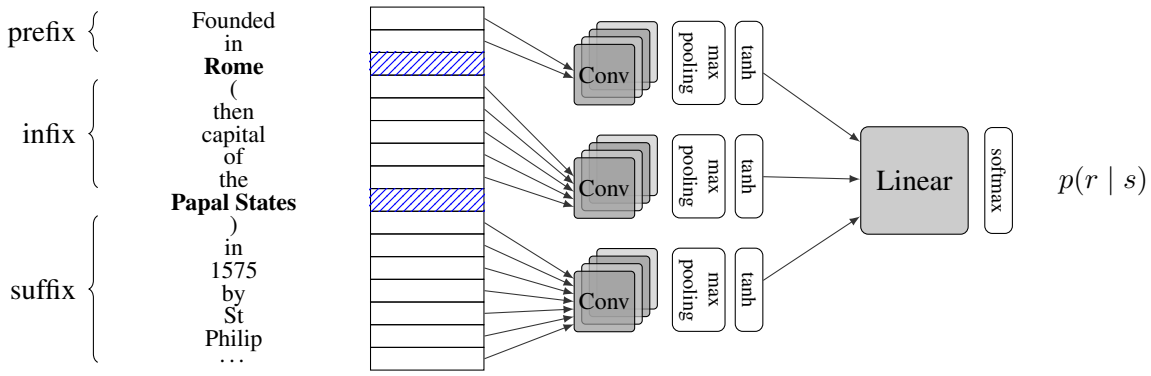

Figure 2: Our relation extraction model. Its input is the sentence with the entities removed $s = \{\text{prefix}, \text{infix}, \text{suffix}\}$. Each part is run through a convolutional layer to give a fixed-size representation, which are then fed to a softmax layer to make a prediction.

patterns in subsequences of words. Then, a max-pooling along the text length combines all features into a fixed-size representation. Note that in our architecture, we obtained better results by using three distinct convolutions, one for each sentence part (i.e. the weights are not shared). We then apply a non-linear function ($\tanh$) and sum the three vectors into a single representation for s. Finally, this representation is fed to a softmax layer to predict the distribution over the relations. This distribution can be plugged into equation (1). Denoting f_{PCNN} our classifier, we have:

$$p(r \mid s) = f_{\text{PCNN}}(r; s, \theta_{\text{PCNN}})$$

where θ_{PCNN} are the parameters of the classifier. Note that we can use the PCNN to predict the relationship for any pair of entities appearing in any sentence, since the input will be different for each pair selected (see Figure 2).

3.2 Link Predictor

The purpose of the link predictor is to provide supervision for the relation classifier. As such, it needs to be differentiable. We follow Marcheggiani and Titov (2016) to model $p(e_i \mid r, e_{-i})$, and use an energy-based formalism, where $\psi(e_1, r, e_2)$ is the energy associated with (e_1, r, e_2). The probability is obtained as follows:

$$p(e_1 \mid r, e_2) \propto \exp(\psi(e_1, r, e_2)) \qquad (2)$$

where ψ is expressed as the sum of two standard relational learning models:

$$\psi(e_1, r, e_2) = \underbrace{\mathbf{u}_{e_1}^T \mathcal{A}_r \mathbf{u}_{e_2}}_{\text{RESCAL}} + \underbrace{\mathbf{u}_{e_1}^T B_r + \mathbf{u}_{e_2}^T C_r}_{\text{Selectional Preferences}}$$

where $\mathbf{u} \in \mathbb{R}^{|E| \times m}$ is an entity embedding matrix, $\mathcal{A} \in \mathbb{R}^{|R| \times m \times m}$ is a three-way tensor encoding the entities interaction and $B, C \in \mathbb{R}^{|R| \times m}$ are two matrices encoding the preferences of each relation of certain entities, and the hyper-parameter m is the dimension of the embedded entities. The function ψ also depends on the energy functions parameters $\theta_\psi = \{\mathcal{A}, B, C, \mathbf{u}\}$ that we omit for legibility. RESCAL (Nickel et al., 2011) uses a bilinear tensor product to gauge the compatibility of the two entities, whereas in the Selectional Preferences model only the predisposition of an entity to appear as the subject or object of a relation is captured.

Negative Sampling

The number of entities being very large, the partition function of equation (2) cannot be efficiently computed. To avoid the summation over the set of entities, we follow Marcheggiani and Titov (2016) and use negative sampling (Mikolov et al., 2013): instead of training a softmax classifier, we train a discriminator which tries to recognize real triplets ($D = 1$) from fake ones ($D = 0$):

$$p(D = 1 \mid e_1, e_2, r) = \sigma\left(\psi(e_1, r, e_2)\right)$$

where $\sigma(x) = 1/(1 + \exp(-x))$ is the sigmoid function. This model is then trained by generating negative entities for each position and optimizing

the negative log likelihood:

$$\mathcal{L}_{\text{LP}} = \mathop{\mathbb{E}}_{\substack{(e_1,e_2,s)\sim\chi \\ r\sim f_{\text{PCNN}}(s)}} \Big[-2\log\sigma\left(\psi(e_1,r,e_2)\right) - \sum_{j=1}^{k} \mathop{\mathbb{E}}_{e'\sim\mathcal{E}} \left[\log\sigma\left(-\psi(e_1,r,e')\right)\right] - \sum_{j=1}^{k} \mathop{\mathbb{E}}_{e'\sim\mathcal{E}} \left[\log\sigma\left(-\psi(e',r,e_2)\right)\right] \Big] \tag{3}$$

This loss is defined over the data distribution χ, i.e. the samples (e_1, e_2, s) follow a uniform distribution over sentences tagged with two entities. The distribution of the relation r for the sentence s is then given by the classifier $f_{\text{PCNN}}(s)$, which corresponds to the $\sum_r p(r \mid s)$ in equation (1). Following standard practice, during training, the expectation on negative entities is approximated by sampling k random entities following the empirical entity distribution $\mathcal{E}$ for each position.

3.3 RelDist losses

Training the classifier through equation (3) alone is very unstable and dependent on precise hyper-parameter tuning. More precisely, according to our early experiments, the training process usually collapses into one of two regimes:

($\mathcal{P}1$) The classifier is very uncertain about which relation is expressed and outputs a relation following a uniform distribution ;

($\mathcal{P}2$) All sentences are classified as conveying the same relation.

In both cases, the link predictor can do a good job minimizing $\mathcal{L}_{\text{LP}}$ by ignoring the output of the classifier, simply exploiting entities co-occurrences. More precisely, many entities only appear in one relationship with a single other entity. In this case, the link predictor can easily ignore the relationship r and predict the missing entity – and there is a pressure for this as the classifier's output is not yet reliable at the beginning of the optimization process.

This instability problem is particularly true since the two components (classifier and link predictor) are strongly interdependent: the classifier cannot be trained without a good link predictor, which itself cannot take r into account without a good classifier resulting in a bootstrap problem. To overcome these pitfalls, we developed two additional losses, that we now describe.

Skewness. Firstly, to encourage the classifier to be confident in its output, we minimize the entropy of the predicted relation distribution. This addresses $\mathcal{P}1$ by forcing the classifier toward outputting one-hot vectors for a given sentence using the following loss:

$$\mathcal{L}_{\text{S}} = \mathbb{E}_{(e_1,e_2,s)\sim\chi}\left[H(R \mid e_1, e_2, s)\right] \tag{4}$$

where R is the random variable corresponding to the predicted relation. Following our first independence hypothesis, the entropy of equation (4) is equivalent to $H(R \mid s)$.

Dispersion. Secondly, to ensure that the classifier predicts several relations, we minimize the KL-divergence between the prior $p(R)$ and the uniform distribution U, that is:

$$\mathcal{L}_{\text{D}} = D_{\text{KL}}(p(R) \,\|\, U) \tag{5}$$

Note that contrary to $\mathcal{L}_{\text{S}}$, in order to have a good approximation of $p(R)$, the loss $\mathcal{L}_{\text{D}}$ measures the un-conditionnal distribution over R, i.e. the distribution of predicted relations over all sentences. This addresses $\mathcal{P}2$ by forcing the classifier toward predicting each class equally often over a set of sentences.

To satisfactorily and jointly train the link predictor and the classifier, we use the two losses at the same time, resulting in the final loss:

$$\mathcal{L} = \mathcal{L}_{\text{LP}} + \alpha\mathcal{L}_{\text{S}} + \beta\mathcal{L}_{\text{D}} \tag{6}$$

where α and β are both positive hyper-parameters.

All three losses are defined over the real data distribution, but in practice they are approximated at the level of a mini-batch. First, both $\mathcal{L}_{\text{LP}}$ and $\mathcal{L}_{\text{S}}$ can be computed for each sample independently. To optimize $\mathcal{L}_{\text{D}}$ however, we need to estimate $p(R)$ at the mini-batch level, and maximize the entropy of the mean predicted relation. Formally, let s_i for $i = 1, \ldots, B$ be the i-th sentence in a batch of size B, we approximate $\mathcal{L}_{\text{D}}$ as:

$$\sum_r \left(\sum_{i=1}^{B} \frac{f_{\text{PCNN}}(r; s_i)}{B}\right) \log\left(\sum_{i=1}^{B} \frac{f_{\text{PCNN}}(r; s_i)}{B}\right)$$

Learning We optimize the empirical estimation of (6), learning the PCNN parameters and word embeddings θ_{PCNN} as well as the link predictor parameters and entity embeddings θ_ψ jointly.

Comparison to VAE When computing the loss of the VAE model (Marcheggiani and Titov, 2016), aside from the reconstruction term $\mathcal{L}_{\text{LP}}$, the following regularization term is derived:

$$\mathcal{L}_{\text{VAEreg}} = \mathbb{E}_{(e_1,e_2,s)\sim\chi}\left[-H(R \mid e_1, e_2, s)\right]$$

This term results from the KL between $p(R \mid e_1, e_2, s)$ and the uniform distribution. Its purpose is to prevent the classifier from always predicting the same relation, i.e. it has the same purpose as our distance loss $\mathcal{L}_{\text{D}}$. However its expression is equivalent to $-\mathcal{L}_{\text{S}}$, and indeed, minimizing the opposite of our skewness loss increases the entropy of the classifier output, addressing $\mathcal{P}2$. Yet, using $\mathcal{L}_{\text{VAEreg}} = -\mathcal{L}_{\text{S}}$ alone, draws the classifier into the other pitfall $\mathcal{P}1$. This causes a drop in performance, as we will show experimentally.

4 Experiments

4.1 Datasets

To evaluate our model we use labeled datasets, the labels being used for validation[2] and evaluation. The first dataset is the one of Marcheggiani and Titov (2016), which is similar to the one used in Yao et al. (2011). This dataset was built through distant supervision (Mintz et al., 2009) by aligning sentences from the New York Times corpus (NYT, Sandhaus, 2008) with Freebase (FB, Bollacker et al., 2008) triplets. Several sentences were filtered out based on features like the length of the dependency path between the two entities, resulting in 2 million sentences with only 41,000 (2%) of them labeled with one of 262 possible relations. 20% of the labeled sentences were set aside for validation, the remaining 80% are used to compute the final results.

We also extracted two datasets from T-REx (Elsahar et al., 2017) which was built as an alignment of Wikipedia with Wikidata (Vrandečić, 2012). We only consider triplets where both entities appear in the same sentence. If a single sentence contains multiple triplets, it will appear multiple times in the dataset, each time with a different pair of target entities. We built the first dataset *DS* by extracting all triplets of T-REx where the two entities are linked by a relation in Wikidata. This is the usual distant supervision method. It resulted in 1189 relations and nearly 12 million sentences, all of them labeled with a relation.

[2] As in other unsupervised RE papers.

In Wikidata, each relation is annotated with a list of associated surface forms, for example "shares border with" can be conveyed by "borders", "adjacent to", "next to", etc. The second dataset we built, *SPO*, only contains the sentences where a surface form of the relation also appears, resulting in 763,000 samples (6% of the unfiltered) and 615 relations. This dataset still contains some misalignment, but should nevertheless be easier for models to extract the correct semantic relation.

4.2 Baseline and Model

We compare our model with two state-of-the-art approaches, two generative rel-LDA models of Yao et al. (2011) and the VAE model of Marcheggiani and Titov (2016).

The two rel-LDA models only differ by the number of features considered. We use the 8 features listed in Marcheggiani and Titov (2016). Rel-LDA uses the first 3 simplest features defined in their paper, while rel-LDA1 is trained by iteratively adding more features until all 8 are used.

To assess our two main contributions individually, we evaluate the PCNN classifier and our additional losses separately.

More precisely, we first study the effect of the RelDist losses by looking at the differences between models optimizing $\mathcal{L}_{\text{LP}} - \alpha\mathcal{L}_{\text{S}}$ and the ones optimizing $\mathcal{L}_{\text{LP}} + \alpha\mathcal{L}_{\text{S}} + \beta\mathcal{L}_{\text{D}}$. Second, we study the effect of the relation classifier by comparing the feature-based classifier and the PCNN trained with the same losses. We thus have four models: March$-\mathcal{L}_{\text{S}}$ (which corresponds to the model of Marcheggiani and Titov (2016)), March$+\mathcal{L}_{\text{S}}+\mathcal{L}_{\text{D}}$, PCNN$-\mathcal{L}_{\text{S}}$ and PCNN$+\mathcal{L}_{\text{S}} + \mathcal{L}_{\text{D}}$.

All models are trained with 10 relation classes, which, while lower than the number of true relations, allows to compare faithfully the models since the distribution of gold relations is very unbalanced. For feature-based models, the size of the features domain range from 1 to 10 million values depending on the dataset. We train our models with Adam using L_2 regularization on all parameters. To have a good estimation of $p(R)$ in the computation of $\mathcal{L}_{\text{D}}$, we use a batch size of 100. Words embeddings are of size 50, entities embeddings of size $m = 10$. We sample $k = 5$ negative samples to estimate $\mathcal{L}_{\text{LP}}$. Lastly, we set $\alpha = 0.01$ and $\beta = 0.02$. All three datasets come with a validation set, and following Marcheggiani and Titov (2016), we used it for cross-validation to optimize

Dataset	Model		B^3			V-measure			ARI
	Classifier	Reg.	F_1	Prec.	Rec.	F_1	Hom.	Comp.	
NYT+FB	rel-LDA		29.1	24.8	35.2	30.0	26.1	35.1	13.3
	rel-LDA1		36.9	30.4	47.0	37.4	31.9	45.1	24.2
	March.	$-\mathcal{L}_S$	35.2	23.8	67.1	27.0	18.6	49.6	18.7
	PCNN	$-\mathcal{L}_S$	27.6	24.3	31.9	24.7	21.2	29.6	15.7
	March.	$\mathcal{L}_S + \mathcal{L}_D$	37.5	31.1	47.4	**38.7**	32.6	47.8	27.6
	PCNN	$\mathcal{L}_S + \mathcal{L}_D$	**39.4**	32.2	50.7	38.3	32.2	47.2	**33.8**
T-REx SPO	rel-LDA		11.9	10.2	14.1	5.9	4.9	7.4	3.9
	rel-LDA1		18.5	14.3	26.1	19.4	16.1	24.5	8.6
	March.	$-\mathcal{L}_S$	24.8	20.6	31.3	23.6	19.1	30.6	12.6
	PCNN	$-\mathcal{L}_S$	25.3	19.2	37.0	23.1	18.1	31.9	10.8
	March.	$\mathcal{L}_S + \mathcal{L}_D$	29.5	22.7	42.0	34.8	28.4	45.1	20.3
	PCNN	$\mathcal{L}_S + \mathcal{L}_D$	**36.3**	28.4	50.3	**41.4**	33.7	53.6	**21.3**
T-REx DS	rel-LDA		9.7	6.8	17.0	8.3	6.6	11.4	2.2
	rel-LDA1		12.7	8.3	26.6	17.0	13.3	23.5	3.4
	March.	$-\mathcal{L}_S$	9.0	6.4	15.5	5.7	4.5	7.9	1.9
	PCNN	$-\mathcal{L}_S$	12.2	8.6	21.1	12.9	10.1	18.0	2.9
	March.	$\mathcal{L}_S + \mathcal{L}_D$	19.5	13.3	36.7	**30.6**	24.1	42.1	**11.5**
	PCNN	$\mathcal{L}_S + \mathcal{L}_D$	**19.7**	14.0	33.4	26.6	20.8	36.8	9.4

Table 1: Results (percentage) on our three datasets. The rel-LDA and rel-LDA1 models come from Yao et al. (2011). The model of Marcheggiani and Titov (2016) is March $-\mathcal{L}_S$.

the B^3F_1 (described below).

4.3 Evaluation metrics

We used the B^3 metric used in Yao et al. (2011) and Marcheggiani and Titov (2016), and complemented it with two more metrics commonly seen in clustering task evaluation: V-measure (Rosenberg and Hirschberg, 2007) and ARI (Hubert and Arabie, 1985), allowing us to capture the characteristics of each approach more in detail.

To clearly describe the different metrics, we propose a common probabilistic formulation of those (in practice, they are estimated on the validation and test sets), and use the following notations. Let X (or Y) be a random variable corresponding to a sentence. We denote $c(X)$ the predicted cluster of X and $g(X)$ its conveyed gold relation.

B-cubed. The first metric we compute is a generalization of F_1 for clustering tasks called B^3 (Bagga and Baldwin, 1998). The B^3 precision and recall are defined as follows:

$$\begin{aligned} B^3 \text{ Precision} &= \mathop{\mathbb{E}}_{X,Y} P\left(g(X) = g(Y) \mid c(X) = c(Y)\right) \\ B^3 \text{ Recall} &= \mathop{\mathbb{E}}_{X,Y} P\left(c(X) = c(Y) \mid g(X) = g(Y)\right) \end{aligned}$$

As precision and recall can be trivially maximized by putting each sample in its own cluster or by clustering all samples into a single class, the main metric B^3 F_1 is defined as the harmonic mean of precision and recall.

V-measure. We also consider an entropy-based metric (Rosenberg and Hirschberg, 2007); this metric is defined by the homogeneity and completeness, which are akin to B^3 precision and recall, but rely on conditional entropy:

$$\begin{aligned} \text{Homogeneity} &= 1 - H\left(c(X) \mid g(X)\right) / H\left(c(X)\right) \\ \text{Completeness} &= 1 - H\left(g(X) \mid c(X)\right) / H\left(g(X)\right) \end{aligned}$$

As B^3, the V-measure is summarized by the F1 value. Compared to B^3, the V-measure penalizes small impurities in a relatively "pure" cluster more harshly than in less pure ones. Symmetrically, it penalizes more a degradation of a well clustered relation than of a less well clustered one.

Adjusted Rand Index. Finally, the Rand Index is defined as the probability that cluster and gold assignments are compatible:

$$\text{RI} = \mathop{\mathbb{E}}_{X,Y}\left[P\left(c(X) = c(Y) \Leftrightarrow g(X) = g(Y)\right)\right]$$

The Adjusted Rand Index (ARI, Hubert and Arabie, 1985) is a normalization of the Rand Index such that a random assignment has an ARI of 0, and the maximum is 1. Compared to the previous metrics, ARI will be less sensitive to a discrepancy between precision/homogeneity and recall/completeness since it is not an harmonic mean of both.

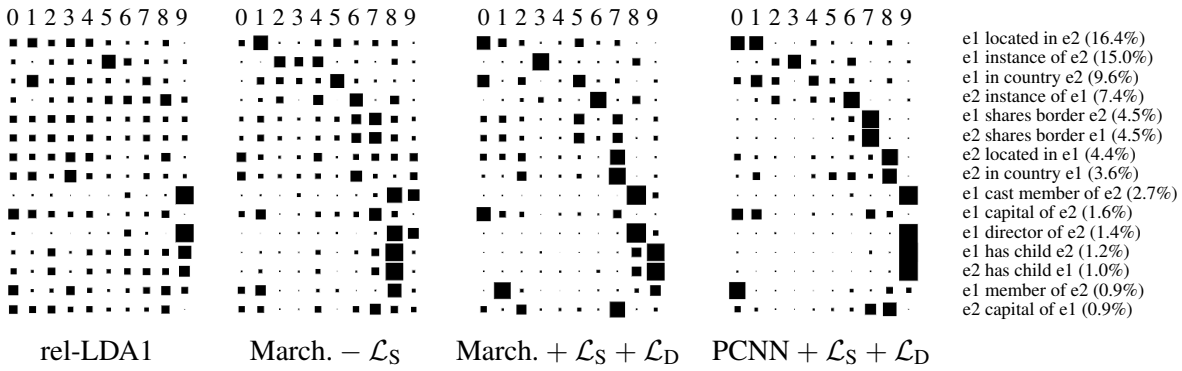

Figure 3: Normalized contingency tables for the TREx SPO dataset. Each of the 10 columns corresponds to a predicted relation cluster, which were sorted to ease comparison. The rows identify Wikidata relations sorted by frequency in the TREx SPO corpus. The area of each square is proportional to the number of sentences in the cell. The matrix was normalized so that each row sum to 1, thus it is more akin to a B^3 per-item recall than a true contingency table.

4.4 Results

The results reported in Table 1 are the average test scores of three runs on the NYT+FB and T-REx SPO datasets, using different random initialization of the parameters – in practice the variance was low enough so that reported results can be analyzed. We observe that regardless of the model and metrics, the highest measures are obtained on T-REx SPO, then NYT+FB and finally T-REx DS. This was to be expected, since T-REx SPO was built to be easy, and hard-to-process sentences were filtered out of NYT+FB (Yao et al., 2011; Marcheggiani and Titov, 2016). We also observe that main metrics agree in general (B^3, V-measure and ARI) in most cases. Performing a PCA on the measures, we observed that V-measure forms a nearly-orthogonal axis to B^3, and to lesser extent ARI. Hence we can focus on B^3 and V-measure in our analysis.

We first measure the benefit of our RelDist losses: on all datasets and metrics, the two models using $+\mathcal{L}_S+\mathcal{L}_D$ are systematically better than the ones using $-\mathcal{L}_S$ alone: (1) The PCNN models consistently gain between 7 and 11 points in B^3 F_1 from these additional losses; (2) The feature-based classifier benefits from the RelDist losses to a lesser extent, except on the T-REx DS dataset on which the March$-\mathcal{L}_S$ model without the RelDist losses completely collapses – we hypothesize that this dataset is too hard for the model given the number of parameters to estimate.

We now restrict to discriminative models based on $+\mathcal{L}_S+\mathcal{L}_D$. We note that both (March/PCNN) exhibit better performances than generative ones (Rel-LDA, Rel-LDA1) with a difference ranging from 2.5/0.6 (NYT, for March/PCNN) to 11/17.8 (on SPO). However, the advantage of PCNN over feature-based classifier is not completely clear. While the PCNN version has a systematically better B^3 F_1 on all datasets (Δ of 0.2/1.9/6.8 respectively for DS/NYT/SPO), the V-measure decreases by 0.4/4.0 on respectively NYT/DS, and ARI by 2.1 on DS. As B^3 F_1 was used for validation, this shows that the PCNN models overfit this metric by polluting relatively clean clusters with unrelated sentences or degrades well clustered gold relations by splitting them within two clusters.

4.5 Qualitative Analysis

Since all the metrics agree on the SPO dataset, we plot the contingency tables of our models in Figure 3. Each row is labeled with the gold Wikidata relation extracted through distant supervision. Since relations are generally not symmetric, each Wikidata relation appears twice in the table, once for each disposition of the entities in the sentence. This is particularly problematic with symmetric relations like "shares border" which are two different gold relations that actually convey the same semantic.

To interpret Figure 3, we have to see whether a predicted cluster (column) contains different gold relations – paying attention to the fact that the most important gold relations are listed in the top rows (the top 5 relations account for 50% of sentences). The first thing to notice is that the contingency tables of both models using our RelDist losses are sparser (for each columnn), which means that our models better separate relations from each other. We observe that March$-\mathcal{L}_S$ is

affected by the pitfall $\mathcal{P}1$ (uniform distribution) for many gold clusters. The $-\mathcal{L}_S$ loss forces the classifier to be uncertain about which relation is expressed, translating into a dense contingency table and resulting in poor performances. The RelLDA1 model is even worse, and fails to identify clear clusters, showing the limitations of a purely generative approach that might focus on clusters not linked with any relation.

Focusing on our proposed model, PCNN+$\mathcal{L}_S$ + $\mathcal{L}_D$ (rightmost figure), we looked at two different mistakes. The first is a gold cluster divided in two (low recall). When looking at clusters 0 and 1, we did not find any recognizable pattern. Moreover, the corresponding link predictor parameters are very similar. This seems to be a limitation of the distance loss: splitting a large cluster in two may improve $\mathcal{L}_D$ but worsen all the evaluation metrics. The model is then penalized by the fact that it lost one slot to transmit information between the classifier and the link predictor. The second type of mistake is when a predicted cluster corresponds to two gold ones (low precision). Here, most of the mistakes seem understandable: "shares border" is symmetric (cluster 7), “located in” and “in country” (cluster 8) or “cast member” and “director of” (cluster 9) are clearly related.

5 Conclusion

In this paper, we show that discriminative relation extraction models can be trained efficiently on unlabeled datasets. Unsupervised relation extraction models tends to produce impure clusters by enforcing a uniformity constrain at the level of a single sample. We proposed two losses (named RelDist) to effectively train expressive relation extraction models by enforcing the distribution over relations to be uniform – note that other target distributions could be used. In particular, we were able to successfully train a deep neural network classifier that only performed well in a supervised setting so far. We demonstrated the effectiveness of our RelDist losses on three datasets and showcased its effect on cluster purity.

Future work will investigate more complex and recent neural network models such as Devlin et al. (2018), as well as alternative losses. In particular, while forcing an uniform distribution with the distance loss $\mathcal{L}_D$ might be meaningful with a low number of predicted clusters, it might not generalize to larger number of relations. Preliminary experiments seem to indicate that this can be addressed by replacing the uniform distribution in equation 5 with the empirical distribution of the relations in the validation set, or any other appropriate law if no validation set is available.

Acknowledgments

We are grateful to Diego Marcheggiani for for sharing his dataset with us. Furthermore, we would like to thank Alexandre Allauzen, Xavier Tannier as well as the anonymous ACL reviewers for their valuable remarks. This work was lead with the support of the FUI-BInD Project.

References

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 344–354.

Amit Bagga and Breck Baldwin. 1998. Entity-based cross-document coreferencing using the vector space model. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*, pages 79–85. Association for Computational Linguistics.

Michele Banko, Michael J Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *IJCAI*, volume 7, pages 2670–2676.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *EMNLP*.

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

Jeffrey Dalton, Laura Dietz, and James Allan. 2014. Entity query feature expansion using knowledge base links. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '14, pages 365–374, New York, NY, USA. ACM.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Elena Simperl, and Frederique Laforest. 2017. T-rex: A large scale alignment of natural language with knowledge base triples. *Proceedings of the 11th International Conference on Language Resources and Evaluation*.

Takaaki Hasegawa, Satoshi Sekine, and Ralph Grishman. 2004. Discovering relations among named entities from large corpora. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 415. Association for Computational Linguistics.

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

Lawrence Hubert and Phipps Arabie. 1985. Comparing partitions. *Journal of classification*, 2(1):193–218.

Nanda Kambhatla. 2004. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, page 22. Association for Computational Linguistics.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2124–2133, Berlin, Germany. Association for Computational Linguistics.

Diego Marcheggiani and Ivan Titov. 2016. Discrete-state variational autoencoders for joint discovery and factorization of relations. *Transactions of the Association for Computational Linguistics*, 4:231–244.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *ICML*, volume 11, pages 809–816.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Andrew Rosenberg and Julia Hirschberg. 2007. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*.

Evan Sandhaus. 2008. The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia*, 6(12):e26752.

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

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D Manning. 2012. Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 455–465. Association for Computational Linguistics.

Denny Vrandečić. 2012. Wikidata: A new platform for collaborative data collection. In *Proceedings of the 21st international conference on world wide web*, pages 1063–1064. ACM.

Limin Yao, Aria Haghighi, Sebastian Riedel, and Andrew McCallum. 2011. Structured relation discovery using generative models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1456–1466. Association for Computational Linguistics.

Limin Yao, Sebastian Riedel, and Andrew McCallum. 2012. Unsupervised relation discovery with sense disambiguation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 712–720. Association for Computational Linguistics.

Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. 2015. Semantic parsing via staged query graph generation: Question answering with knowledge base. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1321–1331. Association for Computational Linguistics.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Emnlp*, pages 1753–1762.