SelfORE: Self-supervised Relational Feature Learning for Open Relation Extraction

Xuming Hu[1], Lijie Wen[1†], Yusong Xu[1], Chenwei Zhang[2†], Philip S. Yu[1,3]
[1]Tsinghua University, [2]Amazon, [3]University of Illinois at Chicago
[1]{hxm19,xys20}@mails.tsinghua.edu.cn
[1]wenlj@tsinghua.edu.cn [2]cwzhang@amazon.com
[3]psyu@uic.edu

Abstract

Open relation extraction is the task of extracting open-domain relation facts from natural language sentences. Existing works either utilize heuristics or distant-supervised annotations to train a supervised classifier over pre-defined relations, or adopt unsupervised methods with additional assumptions that have less discriminative power. In this work, we propose a self-supervised framework named `SelfORE`, which exploits weak, self-supervised signals by leveraging large pretrained language model for adaptive clustering on contextualized relational features, and bootstraps the self-supervised signals by improving contextualized features in relation classification. Experimental results on three datasets show the effectiveness and robustness of `SelfORE` on open-domain Relation Extraction when comparing with competitive baselines. Source code is available[1].

1 Introduction

With huge amounts of information people generate, Relation Extraction (RE) aims to extract triplets of the form (subject, relation, object) from sentences, discovering the semantic relation that holds between two entities mentioned in the text. For example, given a sentence *Derek Bell was born in Belfast*, we can extract a relation BORN IN between two entities *Derek Bell* and *Belfast*. The extracted triplets from the sentence are used in various downstream applications like web search, question answering, and natural language understanding.

Existing RE methods work well on pre-defined relations that have already appeared either in human-annotated datasets or knowledge bases. While in practice, human annotation can be labor-intensive to obtain and hard to scale up to a large

number of relations. Lots of efforts are made to alleviate the human annotation efforts in Relation Extraction. Distant Supervision (Mintz et al., 2009) is a widely-used method to train a supervised relation extraction model with less annotation as it only requires a small amount of annotated triplets as the supervision. However, distant supervised methods usually make strong assumptions on entity co-occurrence without sufficient contexts, which leads to noises and sparse matching results. More importantly, it works on a set of pre-defined relations, which prevents its applicability on open-domain text corpora.

Open Relation Extraction (OpenRE) aims at inferring and extracting triplets where the target relations cannot be specified in advance. Besides approaches that first identify relational phrases from open-domain corpora using heuristics or external labels via distant supervision and then recognize entity pairs (Yates et al., 2007; Fader et al., 2011), clustering-based unsupervised representation learning models get lots of attentions recently due to their ability to recognize triplets from meaningful semantic features with minimized or even no human annotation. Yao et al. (2011) regards OpenRE as a totally unsupervised task and uses clustering method to extract triplets with new relation types. However, it cannot effectively discard irrelevant information and select meaningful relations. Simon et al. (2019) trains expressive relation extraction models in an unsupervised setting. But it still requires that the exact number of relation types in the open-domain corpus is known in advance.

To further alleviate the human annotation efforts while obtaining high-quality supervision for open relation extraction, in this paper, we propose a self-supervised learning framework which obtains supervision from the data itself and learns to improve the supervision quality by learning better feature presentations in an iterative fashion. The proposed

---

[1]https://github.com/THU-BPM/SelfORE
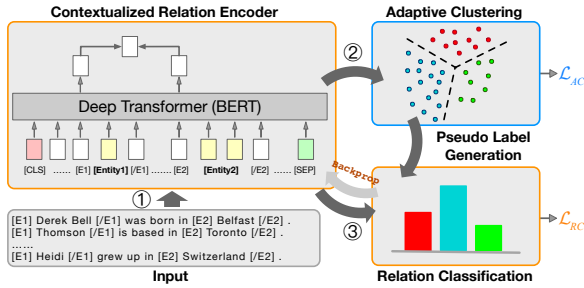[†]Corresponding Authors.

Figure 1: Open Relation Extraction via Self-supervised Learning.

framework has three modules, Contextualized Relation Encoder, Adaptive Clustering, and Relation Classification. As shown in Figure 1, the Contextualized Relation Encoder leverages pretrained BERT model to encode entity pair representations based on the context in which they are mentioned. To recognize and facilitate proximity of relevant entity pairs in the relational semantic space, the Adaptive Clustering module effectively clusters the contextualized entity pair representations generated by Contextualized Relation Encoder and generates pseudo-labels as the self-supervision. The Relation Classification module takes the cluster labels as pseudo-labels to train a relation classification module. The loss of Relation Classification on self-supervised pseudo labels helps improve contextualized entity pairs features in Contextualized Relation Encoder, which further improves the pseudo label quality in Adaptive Clustering in an iterative fashion.

To summarize, the main contributions of this work are as follows:

- We developed a novel self-supervised learning framework SelfORE for relation extraction from open-domain corpus where no relational human annotation is available.

- We demonstrated how to leverage pretrained language models to learn and refine contextualized entity pair representations via self-supervised training schema.

- We showed that the self-supervised model outperforms strong baselines, and is robust when no prior information is available on target relations.

2 Proposed Model

The proposed model SelfORE consists of three modules: Contextualized Relation Encoder, Adap-

tive Clustering, and Relation Classification. As illustrated in Figure 1, the Contextualized Relation Encoder takes sentences as the input, where named entities are recognized and marked in the sentence. Contextualized Relation Encoder leverages the pretrained BERT (Devlin et al., 2018) model to output contextualized entity pair representation. The Adaptive Clustering takes the contextualized entity pair representation as the input, aiming to perform clustering that determines the relational cluster an entity pair belongs to. Unlike traditional clustering methods which assign hard cluster labels to each entity pair and are sensitive to the number of clusters, Adaptive Clustering performs soft-assignment which encourages high confidence assignments and is insensitive to the number of clusters. The pseudo labels based on the clustering results are considered as the self-supervised prior knowledge, which further guides the Relation Classification and features learning in Contextualized Relation Encoder.

Before introducing details of each module, we briefly summarize the overall learning schema:

① Obtain contextualized entity pair representations based on entities mentioned in sentences using Contextualized Relation Encoder.

② Apply Adaptive Clustering based on updated entity pair representations in ① to generate pseudo labels for all relational entity pairs.

③ Use pseudo labels as the supervision to train and update both Contextualized Relation Encoder and Relation Classification. Repeat ②.

2.1 Contextualized Relation Encoder

The contextualized relation encoder aims to extract contextualized relational representations between two given entities in a sentence. In this work, we assume named entities in the text have been recognized ahead of time and we only focus on binary relations which involve two entities.

The type of relation between a pair of entities can be reflected by their contexts. Also, the nuances of expression in contexts also contribute to the relational representation of entity pairs. Therefore, we leverage pretrained deep bi-directional transformers networks (Devlin et al., 2018) to effectively encode entity pairs, along with their context information.

For a sentence $X = [x_1, .., x_T]$ where two entities $E1$ and $E2$ are mentioned, we follow the labeling schema adopted in Soares et al. (2019) and

augment X with four reserved tokens to mark the beginning and the end of each entity mentioned in the sentence. We introduce the $[E1_{start}]$, $[E1_{end}]$, $[E2_{start}]$, $[E2_{end}]$ and inject them to X:

$$X = \big[x_1, ..., [E1_{start}], x_i, ..., x_{j-1}, [E1_{end}], \\ ..., [E2_{start}], x_k, ..., x_{l-1}, [E2_{end}], ..., x_T\big] \tag{1}$$

as the input token sequence for Contextualized Relation Encoder.

The contextualized relation encoder is denoted as $f_\theta(X, E1, E2)$. To get the relation representation of two entities $E1$ and $E2$, instead of using the output of $[CLS]$ token from BERT which summarizes the sentence-level semantics, we use the outputs corresponding to $[E1_{start}]$, $[E2_{start}]$ positions as the contextualized entity representation and concatenate them to derive a fixed-length relation representation $\mathbf{h} \in \mathbb{R}^{2 \cdot h_R}$:

$$\mathbf{h} = [\mathbf{h}_{[E1_{start}]}, \mathbf{h}_{[E2_{start}]}]. \tag{2}$$

2.2 Adaptive Clustering

After we obtained $H = \{\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_N\}$ from N contextualized entity pair representations using Contextualized Relation Encoder, Adaptive Clustering aims to cluster entity pair representations into K semantically-meaningful clusters. Adaptive Clustering gives each entity pair a cluster label, which serves as the pseudo label for later stages.

Comparing with the traditional clustering method which gives hard label assignment for each entity pair (e.g. k-means), the Adaptive Clustering adopts a soft-assignment, adaptive clustering schema. The adaptive clustering encourages high-confidence assignments and is insensitive to the number of clusters. More specifically, Adaptive Clustering consists of two parts: (1) a non-liner mapping g_ϕ to convert the entity pair representation $\mathbf{h} \in \mathbb{R}^{h_R}$ to a latent representation $\mathbf{z} \in \mathbb{R}^{h_{AC}}$, (2) learning a set of K cluster centroids $\{\mu_k \in \mathbb{R}^{h_{AC}}\}_{k=1}^K$, and a soft-assignment of all N entity pairs to K cluster centroids.

For the first part, we simply adopt a set of fully connected layers as the non-linear mapping. Instead of initializing parameters randomly and training the mapping from scratch, the initial parameters are adopted from an encoder of an autoencoder model (Vincent et al., 2010). We pretrain an autoencoder model separately, which takes $\mathbf{h}$ as the input and minimizes the reconstruction loss over all N samples:

$$\widetilde{\mathbf{h}} = Dropout(\mathbf{h}) \tag{3}$$
$$\mathbf{z} = g(\mathbf{W}_\phi \widetilde{\mathbf{h}} + \mathbf{b}_\phi) \tag{4}$$
$$\widetilde{\mathbf{z}} = Dropout(\mathbf{z}) \tag{5}$$
$$\hat{\mathbf{h}} = d(\mathbf{W}_\sigma \widetilde{\mathbf{z}} + \mathbf{b}_\sigma). \tag{6}$$

For the second part, the module learns to optimize g_ϕ's parameters and assign each sample to a cluster with high confidence. We first perform standard k-means clustering in the feature space $\mathbb{R}^{h_{AC}}$ to obtain K initial centroids $\{\mu_k \in \mathbb{R}^{h_{AC}}\}_{k=1}^K$. Inspired by Xie et al. (2016), we use the Student's t-distribution as a kernel to measure the similarity between embedded point $\mathbf{z}_n$ and each centroid μ_k:

$$q_{nk} = \frac{(1 + ||\mathbf{z}_n - \mu_k||^2/\alpha)^{-\frac{\alpha+1}{2}}}{\sum_{k'}(1 + ||\mathbf{z}_n - \mu_{k'}||^2/\alpha)^{-\frac{\alpha+1}{2}}}, \tag{7}$$

where α represents the freedom of the Student's t-distribution and q_{nk} can be regarded as the probability of assigning sample n to cluster k as the soft assignment. We set $\alpha = 1$ for all experiments.

We normalize each cluster by frequency as an auxiliary target distribution in Equation 8 and iteratively refine clusters by learning from high-confidence assignments with the help of an auxiliary distribution:

$$p_{nk} = \frac{q_{nk}^2/f_k}{\sum_{k'} q_{nk'}^2/f_{k'}}, \tag{8}$$

where $f_k = \sum_n q_{nk}$ is the soft cluster frequency.

With the auxiliary distribution, we define KL divergence loss between the soft assignments q_n and the auxiliary distribution p_n as follows to train the Adaptive Clustering module:

$$\mathcal{L}_{AC} = KL(P||Q) = \sum_n \sum_k p_{nk} log \frac{p_{nk}}{q_{nk}}. \tag{9}$$

We use gradient descent based optimizer to minimize $\mathcal{L}_{AC}$. Note that only the parameters for g_ϕ will be updated —parameters in the Contextualized Relation Encoder (f_θ) are not effected when minimizing $\mathcal{L}_{AC}$. We assign the pseudo label s_n for the n-th entity pair by taking the label associated with the largest probability:

$$s_n = \underset{k \in \mathcal{K}}{\operatorname{argmax}} \, p_{nk}. \tag{10}$$

To alleviate the negative impact from choosing unideal initial centroids, Adaptive Clustering reselects a set of K initial centroids randomly if $\mathcal{L}_{AC}$ does not decrease after the first epoch.

In summary, comparing with traditional clustering methods such as k-means, Adaptive Clustering adopts an iterative, soft-assignment learning process which encourages high-confidence assignments and uses high-confidence assignments to improve low-confidence ones. Adaptive Clustering possesses the following advantages: 1) It improves clustering purity and benefits low-confidence assignment for an overall better relational clustering performance. 2) It prevents large relational clusters from distorting the hidden feature space. (3) It neither requires the actual number of target relations in advance (although it is good to have the target relations as the prior knowledge), nor the distribution of relations.

2.3 Relation Classification

Adaptive Clustering generates cluster labels $S = \{s_1, s_2, ..., s_N\}$ for all entity pairs as pseudo labels. With these pseudo labels as self-supervised signals derived from the corpora themselves, Relation Classification module aims to use pseudo labels to guide the relational feature learning in Contextualized Relation Encoder as well as relation classifier learning in Relation Classification.

Similar to supervised classifiers which learn to predict golden labels, the Relation Classification module learns to predict the pseudo labels generated by Adaptive Clustering:

$$\mathbf{l}_n = c_\tau(f_\theta(X_n, E1, E2)), \quad (11)$$

where c_τ denotes the relation classification module parameterized by τ and $\mathbf{l}_n$ is a probability distribution over K pseudo labels for the n-th sample. In order to find the best-performing parameters θ for Contextualized Relation Encoder and τ for Relation Classification, we optimize the following classification loss:

$$\mathcal{L}_{RC} = \min_{\theta,\tau} \frac{1}{N} \sum_{n=1}^{N} loss(\mathbf{l}_n, \text{one\_hot}(s_n)), \quad (12)$$

where $loss$ is the cross entropy loss function and $\text{one\_hot}(s_n)$ returns a one-hot vector indicating the pseudo label assignments.

2.4 The Bootstrapping Self-Supervision Loop

After optimizing $\mathcal{L}_{RC}$, we repeat Adaptive Clustering and Relation Classification in an iterative fashion, shown as ②,③ in Figure 1. Overall, the Adaptive Clustering exploits weak, self-supervised

signals from data and Relation Classification bootstraps the discriminative power of the Contextualized Relation Encoder by improving contextualized relational features for Relation Classification. Note that for Adaptive Clustering, although it does not update Contextualized Relation Encoder, it always utilizes the updated θ to get the most up-to-date entity pair feature representations $\mathbf{h}$ for clustering. Hence it generates stronger self-supervision as the loop goes on, by providing pseudo labels with higher quality for the Relation Classification module.

We stop the clustering and classification loop when current pseudo labels have less than 10% difference with the former epoch. To get the surface-form relation name for each cluster, if there is one, we get words between $[E1]$ and $[E2]$ and calculate the most frequent n-gram as the surface form. For quantitative evaluation, we assign the majority ground truth label within each cluster as the predict relation label for each relation cluster.

3 Experiments

We conduct extensive experiments on real-world datasets to show the effectiveness of our self-supervised learning rationale on relation extraction, and give a detailed analysis to show its advantages.

3.1 Datasets

Three datasets are used to evaluate our model: NYT+FB, T-REx SPO, and T-REx DS. NYT+FB dataset aligns sentences from the New York Times corpus (Sandhaus, 2008) with Freebase (Bollacker et al., 2008) triplets. It has been widely used in previous RE works (Yao et al., 2011; Marcheggiani and Titov, 2016; Simon et al., 2019). We follow the setting in Simon et al. (2019) and filter out sentences with non-binary relations. We get 41,000 labeled sentences containing 262 target relations from 2 million sentences. 20% of these sentences will be used as validation datasets for hyperparameter tuning and 80% will be used for model training.

Both T-REx SPO and T-REx DS datasets come from T-REx (Elsahar et al., 2018) which is generated by aligning Wikipedia corpus with Wikidata (Vrandečić, 2012). We filter triplets and keep sentences where both entities appear in the same sentence — a sentence will appear multiple times if it contains multiple binary relations associated with different entity pairs. We built two datasets T-REx SPO and T-REx DS depending on whether

the dataset has surface-form relations or not. For example, the relation *give birth to* could be conveyed by surface-forms like *born in*, *date of birth*, etc. T-REx SPO contains 615 relations and 763,000 sentences, where all sentences contain triplets having the surface form relation in the sentence. T-REx DS is generated where the surface-form of relation is not necessarily contained in the sentence. T-REx DS contains 1189 relations and nearly 12 million sentences. The dataset still contains some misalignment, but should nevertheless be easier for models to extract the correct semantic relation. 20% of these sentences will be used as the validation dataset and 80% will be used for model training.

3.2 Baseline and Evaluation metrics

We use standard unsupervised evaluation metrics for comparisons with other three baseline algorithms Yao et al. (2011); Marcheggiani and Titov (2016); Simon et al. (2019) where no human annotation is available for Relation Extraction from the open-domain data. For all models, we assume the number of target relations is known to the model in advance. We set the number of clusters to the number of ground-truth categories and evaluate performance with B^3, V-measure and ARI.

Additionally, we evaluate the performance of our proposed model in a practical, yet more challenging setting: we assume the size of target relations is not known. A much larger cluster size $\hat{K}$ such as 1000 is adopted. When $\hat{K} \gg K$, we use unsupervised approaches such as k-means to further merge $\hat{K}$ clusters into K clusters (the size of ground-truth categories) for a fair evaluation.

For baselines, rel-LDA is a generative model proposed by Yao et al. (2011). We consider two variations of rel-LDA which only differ in the number of features they considered. rel-LDA uses the 3 simplest features and rel-LDA-full is trained with a total number of 8 features listed in Marcheggiani and Titov (2016). UIE (Simon et al., 2019) is the state-of-the-art method that trains a discriminative relation extraction model on unlabeled datasets by forcing the model to predict each relation with confidence and encourage all relations to be predicted on average. Two base model architectures (UIE-March and UIE-PCNN) are considered. To make it fair comparison, we further introduce UIE-BERT, which is trained with losses introduced in Simon et al. (2019) but we replace the PCNN classifier + GloVe embedding with our BERT-based Relation

Encoder and Classification module.

To convert pseudo labels indicating the clustering assignment to relation labels for evaluation purposes, we follow the setting in the previous work (Simon et al., 2019) and assign the majority of ground truth relation labels in each cluster to all samples in that cluster as the prediction label. For evaluation metrics, we use B^3 precision and recall to measure the correct rate of putting each sentence in its cluster or clustering all samples into a single class. More specifically, B^3 F_1 is the harmonic mean of precision and recall:

$$B^3 \text{ Prec.} = \mathop{\mathbb{E}}_{X,Y} P(g(X) = g(Y)|c(X) = c(Y))$$

$$B^3 \text{ Rec.} = \mathop{\mathbb{E}}_{X,Y} P(c(X) = c(Y)|g(X) = g(Y)).$$

We use V-measures (Rosenberg and Hirschberg, 2007) to calculate homogeneity and completeness, which is analogous to B^3 precision and recall, but with the conditional entropy:

$$\text{Homogeneity} = 1 - H(c(X)|g(X))/H(c(X))$$

$$\text{Completeness} = 1 - H(g(X)|c(X))/H(g(X))$$

where these two metrics penalize small impurities in a relatively "pure" cluster more harshly than in less pure ones. We also report F1, which is the harmonic mean of Homogeneity and Completeness.

Adjusted Rand Index (Hubert and Arabie, 1985) measures the degree of agreement between two data distributions. The range of ARI is [-1,1], the larger the value, the more consistent the clustering result is with the real situation.

3.3 Implementation Details

Following the settings used in Simon et al. (2019), all models are trained with 10 relation classes. Although it is lower than the number of true relations in the dataset, it still reveals important insights as the distribution of target relations is very unbalanced. Also, this allows us to do a fair comparison with baseline results.

For Contextualized Relation Encoder, we use the default tokenizer in BERT to preprocess dataset and set max-length as 128. We use the pretrained `BERT-Base_Cased` model to initialize parameters for Contextualized Relation Encoder and use BertAdam to optimize the loss.

For Adaptive Clustering, we use an autoencoder with fully connected layers with the following dimensions $2h_R$-500-500-200 as the *encoder* and

Dataset	Model	B^3			V-measure			ARI
		F1	Prec.	Rec.	F1	Hom.	Comp.	
NYT+FB	rel-LDA(Yao et al., 2011)	29.1	24.8	35.2	30.0	26.1	35.1	13.3
	rel-LDA-full(Yao et al., 2011)	36.9	30.4	47.0	37.4	31.9	45.1	24.2
	March(Marcheggiani and Titov, 2016)	35.2	23.8	67.1	27.0	18.6	49.6	18.7
	UIE-March(Simon et al., 2019)	37.5	31.1	47.4	38.7	32.6	47.8	27.6
	UIE-PCNN(Simon et al., 2019)	39.4	32.2	50.7	38.3	32.2	47.2	33.8
	UIE-BERT	41.5	34.6	51.8	39.9	33.9	48.5	35.1
	SelfORE w/o Classification	30.7	28.2	33.8	23.7	21.9	25.6	20.0
	SelfORE w/o Adaptive Clustering	46.2	45.1	47.4	44.1	43.2	45.0	37.6
	SelfORE	**49.1**	47.3	51.1	**46.6**	45.7	47.6	**40.3**
T-REx SPO	rel-LDA(Yao et al., 2011)	11.9	10.2	14.1	5.9	4.9	7.4	3.9
	rel-LDA-full(Yao et al., 2011)	18.5	14.3	26.1	19.4	16.1	24.5	8.6
	March(Marcheggiani and Titov, 2016)	24.8	20.6	31.3	23.6	19.1	30.6	12.6
	UIE-March(Simon et al., 2019)	29.5	22.7	42.0	34.8	28.4	45.1	20.3
	UIE-PCNN(Simon et al., 2019)	36.3	28.4	50.3	41.1	33.7	53.6	21.3
	UIE-BERT	38.1	30.7	50.3	39.1	37.6	40.8	23.5
	SelfORE w/o Classification	32.7	28.3	38.6	25.3	23.1	28.0	22.5
	SelfORE w/o Adaptive Clustering	34.5	31.2	38.5	29.2	27.4	31.2	28.3
	SelfORE	**41.0**	39.4	42.8	**41.4**	40.3	42.5	**33.7**
T-REx DS	rel-LDA(Yao et al., 2011)	9.7	6.8	17.0	8.3	6.6	11.4	2.2
	rel-LDA-full(Yao et al., 2011)	12.7	8.3	26.6	17.0	13.3	23.5	3.4
	March(Marcheggiani and Titov, 2016)	9.0	6.4	15.5	5.7	4.5	7.9	1.9
	UIE-March(Simon et al., 2019)	19.5	13.3	36.7	30.6	24.1	42.1	11.5
	UIE-PCNN (Simon et al., 2019)	19.7	14.0	33.4	26.6	20.8	36.8	9.4
	UIE-BERT	22.4	17.6	30.8	31.2	26.3	38.3	12.3
	SelfORE w/o Classification	31.5	23.2	49.1	14.1	10.9	19.8	7.7
	SelfORE w/o Adaptive Clustering	32.0	26.3	41.0	16.9	14.3	20.8	12.7
	SelfORE	**32.9**	29.7	36.8	**32.4**	30.1	35.1	**20.1**

Table 1: Quantitative performance evaluation on three datasets.

200-500-500-$2h_R$ for the *decoder*. We randomly initialize weights using a Gaussian distribution with zero-mean and a standard deviation of 0.01. The autoencoder is pretrained for 20 epoches with $1e-3$ learning rate and $1e-5$ weight-decay with Adam Optimizer. To get the initial centroids, we applied k-means and set K as 10.

For Relation Classification, we use a fully connected layer as c_τ and set dropout rate to 10%, learning rate to $1e-5$ and warm-up rate to 0.1. We fixed the parameters in f_θ for the first three epochs to allow the classification layer to warm up.

3.4 Results

Table 1 shows the experimental results. UIE-PCNN is considered as the previous state-of-the-art result. We enhance this baseline by replacing PCNN and GloVe embedding with the proposed BERT-based encoder and classifier. The enhanced state-of-the-art model, namely UIE-BERT, achieves the best performance among baselines. The proposed SelfORE model outperforms all baseline models consistently on B^3 F1/Precision, V-measure F1/Homogeneity and ARI. SelfORE on average achieves 7.0% higher in B^3 F1, 3.4% higher in V-

measure F1 and 7.7% higher in ARI among three datasets when comparing with UIE-BERT. Unlike baseline methods which achieve high B^3 Recall but low Precision, or high V-measure Completeness but low Homogeneity, our model obtains a more balanced performance while achieving the highest Precision and Homogeneity, although B^3 Recall and V-measure Completeness are less satisfactory. Having high precision and homogeneity scores can be a quite appealing property for precision-oriented applications in the real-world.

Ablation Study

We conduct ablation study to show the effectiveness of different modules of SelfORE to the overall improved performance. SelfORE w/o Classification is the proposed model without Relation Classification and only uses the Contextualized Relation Encoder for Adaptive Clustering. SelfORE w/o Adaptive Clustering replaces the proposed soft-assignment clustering methods with k-means clustering as a hard-assignment alternative.

A general conclusion from ablation rows in Table 1 is that all modules contribute positively to the improved performance. More specifically, without self-supervised signals for relational feature learn-
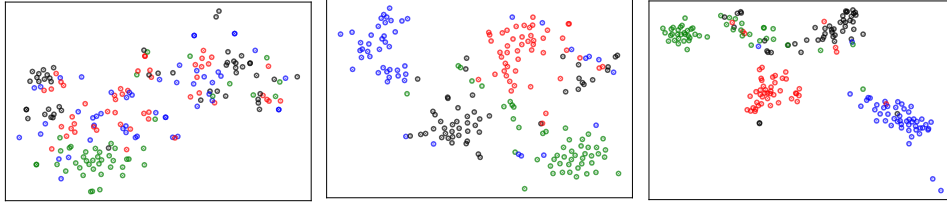
Figure 2: Visualizing contextualized entity pair features after t-SNE dimension reduction for `SelfORE` w/o classification (left), `SelfORE` w/o Adaptive Clustering (middle) and `SelfORE` (right) on NYT+FB dataset.

ing, `SelfORE` w/o Classification gives us 14.4% less performance averaged over all metrics on all datasets. Similarly, Adaptive Clustering gives 6.2% performance boost in average over all metrics when comparing with the hard-assignment alternative (`SelfORE` w/o Adaptive Clustering).

Visualize Contextualized Features
To intuitively show how self-supervised learning helps learn better contextualized relational features on entity pairs, we visualize the contextual representation $\mathbb{R}^{2 \cdot h_R}$ after dimension reduction using t-SNE (Maaten and Hinton, 2008). We randomly choose 4 relations from NYT+FB dataset and sample 50 entity pairs. The visualization results are shown in Figure 2. Features are colored with their ground-truth relation labels.

From Figure 2 we can see that the features obtained through the raw BERT model (left) can already give meaningful semantics to entity pairs having different relations. But these features are not tailored for the relation extraction task. When Adaptive Clustering is not applied (middle) and simply using k-means, which performs hard-assignment on samples, the proposed model without Adaptive Clustering gives decent results but does not provide confident cluster assignments. The proposed model (right) uses soft-assignment and a self-supervised learning schema to improve the relational feature learning —we learn denser clusters and more discriminitaive features.

Sensitivity analysis: when K is unknown
The Adaptive Clustering gives `SelfORE` enough flexibility to model relational features without knowing any prior information on the number of target relations or the relation distribution. This property is appealing when the number of target relations is not available for Relation Extraction on an open-domain corpus.

The proposed model does require an intial cluster size $\hat{K}$ as the scope for pseudo labels. A general guideline for choosing $\hat{K}$ is to choose a value that

is larger than the actual number of relations in the corpora as over-specifying the cluster size should not hurt the model performance. We set an initial $\hat{K}$ (for example $\hat{K} = 1000$), and use an unsupervised method, here we use k-means, to merge $\hat{K}$ cluster centroids into K clusters for evaluation.

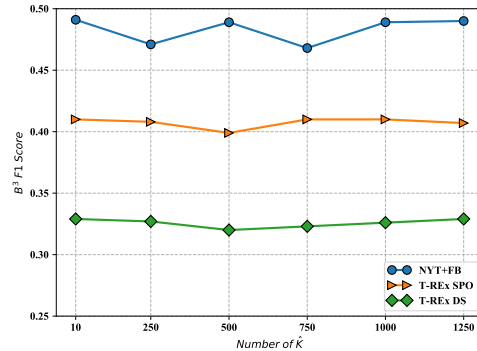

Figure 3: F1 Score with different $\hat{K}$.

We vary $\hat{K}$ from 10 to 1250 and report the B^3 F1 score when comparing the predicted relation type (based on K clusters after merging) with the golden relation type. As shown in Figure 3, the best performance is obtained when $\hat{K} = 10$, indicating that `SelfORE` actually leverages the number of target relations as a useful prior knowledge. Thanks to the self-learning schema and the Adaptive Clustering, when we very $\hat{K}$ from 10 to 1250, the model achieves stable F1 score and is not sensitive to the initial choice of $\hat{K}$ on all three datasets. The results also further indicate the applicability of the proposed model when being applied to an open-domain corpus when the number of target relations is not available in advance. We can assign a larger $\hat{K}$ value than needed and the model is still robust. Note that merging $\hat{K}$ clusters into K clusters is mainly for evaluation purpose: when K is unknown in advance and we simply use a large K directly, it does result in K clusters where clusters tend to be smaller, and multiple clusters may correspond to entity pairs having the same relation.

Extracted surface-form	Golden surface-form
are close to	shares border with
the state of	country
capital city	capital
son of	child
member of	member of

Table 2: Extracted and golden surface-form relation names on T-REx SPO.

Surface-form Relation Names

We provide a brief case study to show the surface-form relation names we extracted for each cluster (introduced in Section 2.4). We randomly select 5 relations in T-REx SPO and report the extracted surface-form relation names using frequent n-gram in Table 2. The surface-form relation name extraction gives SelfORE an extended ability to not only discriminate between entity pairs having different relations, but also derive surface-forms for relation clusters as the final Relation Extraction results. However, evaluating the quality of relation surface-forms is out-of-scope for this work.

4 Related Works

Relation extraction focuses on identifying the relation between two entities in a given sentence. Traditional closed-domain relation extraction methods are supervised models. They need a set of pre-defined relation labels and require large amounts of annotated triplets, making them less ideal to work on open-domain corpora. Distant supervision (Mintz et al., 2009; Surdeanu et al., 2012) is a widely used method to alleviate human annotation: if multiple sentences contain two entities that have a certain relation in a knowledge base, at least one sentence is believed to convey the related relation. However, entities convey semantic meanings also according to the contexts, distant supervised models do not explicitly consider contexts and the models cannot discover new relations as the supervision is purely adopted from knowledge bases.

Unsupervised relation extraction (Stanovsky et al., 2018; Saha et al., 2018; Yu et al., 2017) gets lots of attention, due to the ability to discover relational knowledge without access to annotations and external resources. Unsupervised models either 1) cluster the relation representation extracted from the sentence; 2) make more assumptions that provide learning signals for classification models. Among clustering models, an important milestone is the OpenIE approach (Banko et al., 2007), assuming the surface form of relations will appear be-

tween two entities in its dependency tree. However, these works heavily rely on surface-form relation and have less ideal generalization capabilities. To solve this problem, Roy et al. (2019) proposes a system that learns to supervise unsupervised OpenIE model, which combines the strength and avoids the weakness in each individual OpenIE system. Relation knowledge transfer system (Wu et al., 2019) learns similarity metrics of relations from labeled data, and then transfers the relational knowledge to identify novel relations in unlabeled data.

Marcheggiani and Titov (2016) proposes a variational autoencoder approach (VAE): the encoder part extracts relations from labeled features, and the decoder part predicts one entity when given the other entity and the relation with the function of triplet scoring (Nickel et al., 2011). This scoring function could provide a signal since it is known to predict relation triplets when given their embeddings. However, posterior distribution and prior uniform distribution based on KL divergence is unstable. Simon et al. (2019) proposes a model to solve instability and trains the features on classifiers such as PCNN model (Zeng et al., 2015).

Inspired by the success of self-supervised learning in computer vision (Wiles et al., 2018; Caron et al., 2018), and large pretrained language models that show great potential in encoding meaningful semantics for various downstream tasks (Devlin et al., 2018), we proposed a self-supervised learning schema for open-domain relation extraction. It has the advantages of unsupervised learning to handle the cases where the number of relations is not known in advance, but also keeps the advantage of supervised learning that has strong discriminative power for relational feature learning.

5 Conclusions

We propose a self-supervised learning model SelfORE for open-domain relation extraction. Different from conventional distant-supervised models which require labeled instances for Relation Extraction in a closed-world setting, our model does not require annotations and is able to work on open-domain scenarios when target relation number and relation distributions are not known in advance. Comparing with unsupervised models, our model exploits the advantages of supervised models and bootstraps the discriminative power using self-supervised signals via learning improved contextualized relational features. Experiments on three real-world datasets show effectiveness and ro-

bustness of `SelfORE` over competitive baselines.

Acknowledgments

References

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.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD*, pages 1247–1250. AcM.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. 2018. Deep clustering for unsupervised learning of visual features. In *ECCV*, pages 132–149.

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, Frédérique Laforest, and Elena Simperl. 2018. T-rex: A large scale alignment of natural language with knowledge base triples. In *LREC-2018*.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *EMNLP*, pages 1535–1545. Association for Computational Linguistics.

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

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *JMLR*, 9(Nov):2579–2605.

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.

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

Andrew Rosenberg and Julia Hirschberg. 2007. V-measure: A conditional entropy-based external cluster evaluation measure. In *EMNLP-CoNLL*, pages 410–420.

Arpita Roy, Youngja Park, Taesung Lee, and Shimei Pan. 2019. Supervising unsupervised open information extraction models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 728–737.

Swarnadeep Saha et al. 2018. Open information extraction from conjunctive sentences. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2288–2299.

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

Étienne Simon, Vincent Guigue, and Benjamin Piwowarski. 2019. Unsupervised information extraction: Regularizing discriminative approaches with relation distribution losses. In *ACL*, pages 1378–1387.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. *arXiv preprint arXiv:1906.03158*.

Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. 2018. Supervised open information extraction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 885–895.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D Manning. 2012. Multi-instance multi-label learning for relation extraction. In *EMNLP*, pages 455–465. Association for Computational Linguistics.

Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *JMLR*, 11(Dec):3371–3408.

Denny Vrandečić. 2012. Wikidata: A new platform for collaborative data collection. In *WWW*, pages 1063–1064. ACM.

Olivia Wiles, A Koepke, and Andrew Zisserman. 2018. Self-supervised learning of a facial attribute embedding from video. *arXiv preprint arXiv:1808.06882*.

Ruidong Wu, Yuan Yao, Xu Han, Ruobing Xie, Zhiyuan Liu, Fen Lin, Leyu Lin, and Maosong Sun. 2019. Open relation extraction: Relational knowledge transfer from supervised data to unsupervised data. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 219–228.

Junyuan Xie, Ross Girshick, and Ali Farhadi. 2016. Unsupervised deep embedding for clustering analysis. In *ICML*, pages 478–487.

Limin Yao, Aria Haghighi, Sebastian Riedel, and Andrew McCallum. 2011. Structured relation discovery using generative models. In *EMNLP*, pages 1456–1466. Association for Computational Linguistics.

Alexander Yates, Michael Cafarella, Michele Banko, Oren Etzioni, Matthew Broadhead, and Stephen Soderland. 2007. Textrunner: open information extraction on the web. In *NAACL*, pages 25–26. Association for Computational Linguistics.

Dian Yu, Lifu Huang, and Heng Ji. 2017. Open relation extraction and grounding. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 854–864.

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.