Open Relation Extraction: Relational Knowledge Transfer from Supervised Data to Unsupervised Data

Ruidong Wu[1*], **Yuan Yao**[1*], **Xu Han**[1], **Ruobing Xie**[2],
Zhiyuan Liu[1†], **Fen Lin**[2], **Leyu Lin**[2], **Maosong Sun**[1]
[1]Department of Computer Science and Technology, Tsinghua University, Beijing, China
Institute for Artificial Intelligence, Tsinghua University, Beijing, China
State Key Lab on Intelligent Technology and Systems, Tsinghua University, Beijing, China
[2]Search Product Center, WeChat Search Application Department, Tencent, China
`mooninsiderain@gmail.com`

Abstract

Open relation extraction (OpenRE) aims to extract relational facts from the open-domain corpus. To this end, it discovers relation patterns between named entities and then clusters those semantically equivalent patterns into a united relation cluster. Most OpenRE methods typically confine themselves to unsupervised paradigms, without taking advantage of existing relational facts in knowledge bases (KBs) and their high-quality labeled instances. To address this issue, we propose Relational Siamese Networks (RSNs) to learn similarity metrics of relations from labeled data of pre-defined relations, and then transfer the relational knowledge to identify novel relations in unlabeled data. Experiment results on two real-world datasets show that our framework can achieve significant improvements as compared with other state-of-the-art methods. Our code is available at `https://github.com/thunlp/RSN`.

1 Introduction

Relation extraction (RE) aims to extract relational facts between two entities from plain texts. For example, with the sentence *“Hayao Miyazaki is the director of the film ‘The Wind Rises’"*, we can extract a relation “`director_of`" between two entities “*Hayao Miyazaki*" and “*The Wind Rises*".

Recent progress in supervised methods to RE has achieved great successes. Supervised methods can effectively learn significant relation semantic patterns based on existing labeled data, but the data constructions are time-consuming and human-intensive. To lower the level of supervision, several semi-supervised approaches have been developed, including bootstrapping, active learning, label propagation (Pawar et al., 2017). Mintz (2009) also proposes distant supervision to generate training data automatically. It assumes that if two entities have a relation in KBs, all sentences that contain these two entities will express this relation. Still, all these approaches can only extract pre-defined relations that have already appeared either in human-annotated datasets or KBs. It is hard for them to cover the great variety of novel relational facts in the open-domain corpora.

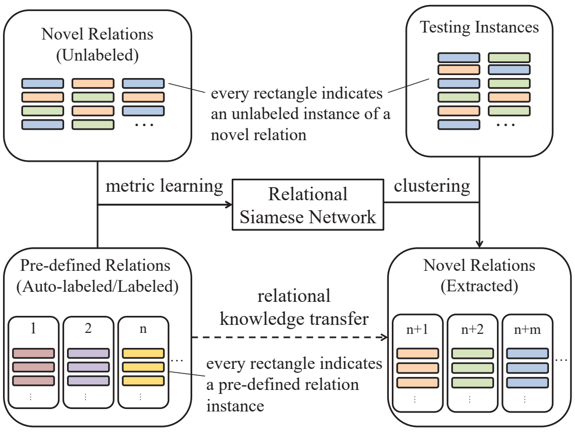

Novel Relations (Unlabeled)
Testing Instances
every rectangle indicates an unlabeled instance of a novel relation
metric learning
Relational Siamese Network
clustering
Pre-defined Relations (Auto-labeled/Labeled)
1
2
n
relational knowledge transfer
every rectangle indicates a pre-defined relation instance
Novel Relations (Extracted)
n+1
n+2
n+m

Figure 1: A flowchart of our framework. Our model RSN learns from both labeled instances of pre-defined relations and unlabeled instances of new relations, and tries to cluster testing instances of new relations.[1]

Open relation extraction (OpenRE) aims to extract relational facts on the open-domain corpus, where the relation types may not be pre-defined. There are some efforts concentrating on extracting triples with new relation types. Banko (2008) directly extracts words or phrases in sentences to represent new relation types. However, some relations cannot be explicitly represented with tokens in sentences, and it is hard to align different relational tokens that exactly have the same meanings. Yao (2011) consid-

[*] indicates equal contribution
[†] Corresponding author: Z.Liu(liuzy@tsinghua.edu.cn)

[1] To highlight our model’s ability to extract new relations, testing instances only contain new relations.

*Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing
and the 9th International Joint Conference on Natural Language Processing*, pages 219–228,
Hong Kong, China, November 3–7, 2019. ©2019 Association for Computational Linguistics

ers OpenRE as a clustering task for extracting triples with new relation types. However, previous clustering-based OpenRE methods (Yao et al., 2011, 2012; Marcheggiani and Titov, 2016; Elsahar et al., 2017) are mostly unsupervised, and cannot effectively select meaningful relation patterns and discard irrelevant information.

In this paper, we propose to take advantage of high-quality supervised data of pre-defined relations for OpenRE. The approach is non-trivial, however, due to the considerable gap between the pre-defined relations and novel relations of interest in open domain. To bridge the gap, we propose **Relational Siamese Networks** (RSNs) to learn transferable relational knowledge from supervised data for OpenRE. Specifically, RSNs learn relational similarity metrics from labeled data of pre-defined relations, and then transfer the metrics to measure the similarity of unlabeled sentences for open relation clustering. We describe the flowchart of our framework in Figure 1.

Moreover, we show that RSNs can also be generalized to various weakly-supervised scenarios. We propose **Semi-supervised RSN** to learn from both supervised data of pre-defined relations and unsupervised data with novel relations, and **Distantly-supervised RSN** to learn from distantly-supervised data and unsupervised data.

We conduct experiments on real-world RE datasets, FewRel and FewRel-distant, by splitting relations into seen and unseen set, and evaluate our models in supervised, semi-supervised, and distantly-supervised scenarios. The results demonstrate that our models significantly outperform state-of-the-art baseline methods in all scenarios without using external linguistic tools. To summarize, the main contributions of this work are as follows:

(1) We develop a novel relational knowledge transfer framework RSN for OpenRE, which can effectively transfer existing relational knowledge to novel-relation data and accurately identify novel relations. To the best of our knowledge, RSN is the first model to consider knowledge transfer in clustering-based OpenRE task.

(2) We further propose Semi-supervised RSNs and Distantly-supervised RSNs that can learn from various weakly supervised scenarios. The experimental results show that all these RSN models achieve significant improvements in F-measure compared with state-of-the-art baselines.

2 Related Work

Open Relation Extraction. Relation extraction (RE) is an important task in NLP. Traditional RE methods mainly concentrate on classifying relational facts into pre-defined relation types (Mintz et al., 2009; Yu et al., 2017). Zeng (2014) utilizes CNN encoders to build sentence representations with the help of position embeddings. Lin (2016) further improves RE performance on distantly-supervised data via instance-level attention. These methods take advantage of supervised or distantly-supervised data to learn neural sentence encoders for distributed representations, and have achieved promising results. However, these methods cannot handle the open-ended growth of new relation types in the open-domain corpora.

To solve this problem, recently many efforts have been invested in exploring methods for open relation extraction (OpenRE), which aims to discover new relation types from unsupervised open-domain corpora. OpenRE methods can be roughly divided into two categories: tagging-based and clustering-based. Tagging-based methods cast OpenRE as a sequence labeling problem, and extract relational phrases consisting of words from sentences in unsupervised (Banko et al., 2007; Banko and Etzioni, 2008) or supervised paradigms (Jia et al., 2018; Cui et al., 2018; Stanovsky et al., 2018). However, tagging-based methods often extract multiple overly-specific relational phrases for the same relation type, and cannot be readily utilized for downstream tasks.

In comparison, conventional clustering-based OpenRE methods extract rich features for relation instances via external linguistic tools, and cluster semantic patterns into several relation types (Lin and Pantel, 2001; Yao et al., 2011, 2012). Marcheggiani (2016) proposes a reconstruction-based model discrete-state variational autoencoder for OpenRE via unlabeled instances. Elsahar (2017) utilizes a clustering algorithm over linguistic features. In this paper, we focus on the clustering-based OpenRE methods, which have the advantage of discovering highly distinguishable relation types.

Few-shot Learning. Few-shot learning aims to classify instances with a handful of labeled samples. Many efforts are devoted to few-shot image classification (Koch et al., 2015) and relation classification (Yuan et al., 2017; Han et al., 2018). Notably, (Koch et al., 2015) introduces Convolu-

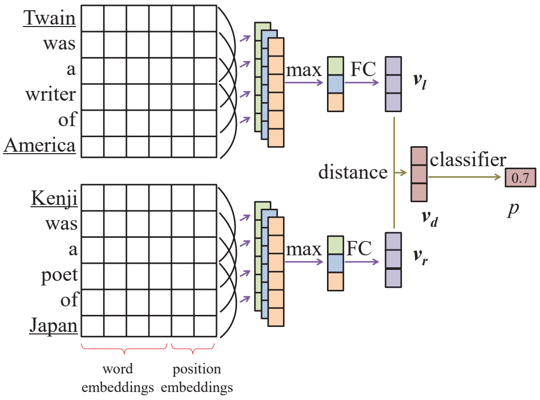


Figure 2: The architecture of Relational Siamese Networks. The output is the similarity between two relational instances.

tional Siamese Neural Network for image metric learning, which inspires us to learn relational similarity metrics for OpenRE.

Semi-supervised Clustering. Semi-supervised clustering aims to cluster semantic patterns given instance seeds of target categories (Bair, 2013; Hongtao Lin, 2019). Differently, our proposed Semi-supervised RSN only leverages labeled instances of pre-defined relations, and does not need any seed of new relations.

3 Methodology

Our OpenRE framework mainly consists of two modules, the relation similarity calculation module and the relation clustering module. For relation similarity calculation, we propose Relational Siamese Networks (RSNs), which learn to predict whether two sentences mention the same relation. To utilize large-scale unsupervised data and distantly-supervised data, we further propose Semi-supervised RSN and Distantly-supervised RSN. Finally, in the relation clustering module, with the learned relation metric, we utilize hierarchical agglomerative clustering (HAC) and Louvain clustering algorithms to cluster target relation instances of new relation types.

3.1 Relational Siamese Network (RSN)

The architecture of our Relational Siamese Networks is shown in Figure 2. CNN modules encode a pair of relational instances into vectors, and several shared layers compute their similarity.

Sentence Encoder. We use a CNN module as the sentence encoder. The CNN module includes an embedding layer, a convolutional layer, a max-pooling layer, and a fully-connected (FC) layer. The embedding layer transforms the words in a sentence x and the positions of entities e_{head} and e_{tail} into pre-trained word embeddings and random-initialized position embeddings. Following (Zeng et al., 2014), we concatenate these embeddings to form a vector sequence. Next, a one-dimensional convolutional layer and a max-pooling layer transform the vector sequence into features. Finally, an FC layer with sigmoid activation maps features into a relational vector $\boldsymbol{v}$. To summarize, we obtain a vector representation $\boldsymbol{v}$ for a relational sentence with our CNN module:

$$\boldsymbol{v} = \mathrm{CNN}(\mathrm{s}), \tag{1}$$

in which we denote the joint information of a sentence x and two entities in it e_{head} and e_{tail} as a data sample s. And with paired input relational instances, we have:

$$\boldsymbol{v_l} = \mathrm{CNN}(\mathrm{s_l}), \boldsymbol{v_r} = \mathrm{CNN}(\mathrm{s_r}), \tag{2}$$

in which two CNN modules are identical and share all the parameters.

Similarity Computation. Next, to measure the similarity of two relational vectors, we calculate their absolute distance and transform it into a real-number similarity $p \in [0, 1]$. First, a distance layer computes the element-wise absolute distance of two vectors:

$$\boldsymbol{v_d} = |\boldsymbol{v_l} - \boldsymbol{v_r}|. \tag{3}$$

Then, a classifier layer calculates a metric p for relation similarity. The layer is a one-dimensional-output FC layer with sigmoid activation:

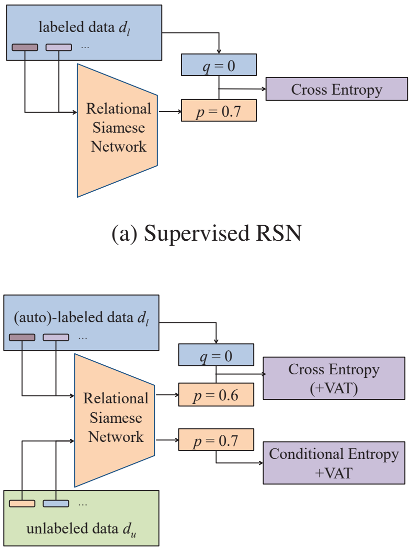
$$p = \sigma(\boldsymbol{k}\boldsymbol{v_d} + b), \tag{4}$$

in which σ denotes the sigmoid function, $\boldsymbol{k}$ and b denote the weights and bias. To summarize, we obtain a good similarity metric p of relational instances.

Cross Entropy Loss. The output of RSN p can also be explained as the probability of two sentences mentioning two different relations. Thus, we can use binary labels q and binary cross entropy loss to train our RSN:

$$\mathcal{L}_l = \mathbb{E}_{d_l \sim \mathcal{D}_l}[q \ln(p_\theta(d_l)) + (1-q)\ln(1-p_\theta(d_l))], \tag{5}$$

in which θ indicates all the parameters in the RSN.



(a) Supervised RSN

(b) Weakly-supervised RSNs

Figure 3: The comparison of (a) Supervised RSN and (b) Weakly-supervised RSNs. Weakly-supervised RSNs, including Semi-supervised RSN and Distantly-supervised RSN, further learn from unlabeled data with conditional entropy minimization and virtual adversarial training (VAT). In figures, p indicates the predicted similarity of two relational sentences, while q indicates the ground-truth label between them.

3.2 Semi-supervised RSN

To discover relation clusters in the open-domain corpus, it is beneficial to not only learn from labeled data, but also capture the manifold of unlabeled data in the semantic space. To this end, we need to push the decision boundaries away from high-density areas, which is known as the cluster assumption (Chapelle and Zien, 2005).

We try to achieve this goal with several additional loss functions. In the following paragraphs, we denote the labeled training dataset as $\mathcal{D}_l$ and a couple of labeled relational instances as d_l. Similarly, we denote the unlabeled training dataset as $\mathcal{D}_u$ and a couple of unlabeled instances as d_u.

Conditional Entropy Loss. In classification problems, a well-classified embedding space usually reserves large margins between different classified clusters, and optimizing margin can be a promising way to facilitate training. However, in clustering problems, type labels are not available during training. To optimize margin without explicit supervision, we can push the data points away from the decision boundaries. Intuitively, when the distance similarity p between two relational instances equals 0.5, there is a high probability that at least one of two instances is near the decision boundary between relation clusters. Thus, we use the conditional entropy loss (Grandvalet and Bengio, 2005), which reaches the maximum when $p = 0.5$, to penalize close-boundary distribution of data points:

$$\mathcal{L}_u = \mathbb{E}_{d_u \sim \mathcal{D}_u}[p_\theta(d_u)\ln(p_\theta(d_u)) + (1 - p_\theta(d_u))\ln(1 - p_\theta(d_u))]. \tag{6}$$

Virtual Adversarial Loss. Despite its theoretical promise, conditional entropy minimization suffers from shortcomings in practice. Due to neural networks' strong fitting ability, a very complex decision hyperplane might be learned so as to keep away from all the training samples, which lacks generalizability. As a solution, we can smooth the relational representation space with locally-Lipschitz constraint.

To satisfy this constraint, we introduce virtual adversarial training (Miyato et al., 2016) on both branches of RSN. Virtual adversarial training can search through data point neighborhoods, and penalize most sharp changes in distance prediction. For labeled data, we have

$$\mathcal{L}_{vl} = \mathbb{E}_{d_l \sim \mathcal{D}_l}[\mathrm{D}_{\mathrm{KL}}(p_\theta(d_l)||p_\theta(d_l, t_1, t_2))], \tag{7}$$

in which D_{KL} indicates the Kullback-Leibler divergence, $p_\theta(d_l, t_1, t_2)$ indicates a new distance estimation with perturbations t_1 and t_2 on both input instances respectively. Specifically, t_1 and t_2 are worst-case perturbations that maximize the KL divergence between $p_\theta(d_l)$ and $p_\theta(d_l, t_1, t_2)$ with a limited length. Empirically, we approximate the perturbations the same as the original paper (Miyato et al., 2016). Specifically, we first add a random noise to the input, and calculate the gradient of the KL-divergence between the outputs of the original input and the noisy input. We then add the normalized gradient to the original input and get the perturbed input. And for unlabeled data, we have

$$\mathcal{L}_{vu} = \mathbb{E}_{d_u \sim \mathcal{D}_u}[\mathrm{D}_{\mathrm{KL}}(p_\theta(d_u)||p_\theta(d_u, t_1, t_2))], \tag{8}$$

in which the perturbations t_1 and t_2 are added to word embeddings rather than the words themselves.

To summarize, we use the following loss function to train Semi-supervised RSN, which learns from both labeled and unlabeled data:

$$\mathcal{L}_{all} = \mathcal{L}_l + \lambda_v \mathcal{L}_{vl} + \lambda_u(\mathcal{L}_u + \lambda_v \mathcal{L}_{vu}), \tag{9}$$

in which λ_v and λ_u are two hyperparameters.

3.3 Distantly-supervised RSN

To alleviate the intensive human labor for annotation, the topic of distantly-supervised learning has attracted much attention in RE. Here, we propose Distantly-supervised RSN, which can learn from both distantly-supervised data and unsupervised data for relational knowledge transfer. Specifically, we use the following loss function:

$$\mathcal{L}_{all} = \mathcal{L}_l + \lambda_u(\mathcal{L}_u + \lambda_v \mathcal{L}_{vu}), \tag{10}$$

which treats auto-labeled data as labeled data but removes the virtual adversarial loss on the auto-labeled data.

The reason to remove the loss is simple: virtual adversarial training on auto-labeled data can amplify the noise from false labels. Indeed, we do find that the virtual adversarial loss on auto-labeled data can harm our model's performance in experiments.

We do not use more denoising methods, since we think RSN has some inherent advantages of tolerating such noise. Firstly, the noise will be overwhelmed by the large proportion of negative sampling during training. Secondly, during clustering, the prediction of a new relation cluster is based on areas where the density of relational instances is high. Outliers from noise, as a result, will not influence the prediction process so much.

3.4 Open Relation Clustering

After RSN is learned, we can use RSN to calculate the similarity matrix of testing instances. With this matrix, several clustering methods can be applied to extract new relation clusters.

Hierarchical Agglomerative Clustering. The first clustering method we adopt is hierarchical agglomerative clustering (HAC). HAC is a bottom-up clustering algorithm. At the start, every testing instance is regarded as a cluster. For every step, it agglomerates two closest instances. There are several criteria to evaluate the distance between two clusters. Here, we adopt the complete-linkage criterion, which is more robust to extreme instances.

However, there is a significant shortcoming of HAC: it needs the exact number of clusters in advance. A potential solution is to stop agglomerating according to an empirical distance threshold, but it is hard to determine such a threshold. This problem leads us to consider another clustering algorithm Louvain (Blondel et al., 2008).

Louvain. Louvain is a graph-based clustering algorithm traditionally used for detecting communities. To construct the graph, we use the binary approximation of RSN's output, with 0 indicating an edge between two nodes. The advantage of Louvain is that it does not need the number of potential clusters beforehand. It will automatically find proper sizes of clusters by optimizing community modularity. According to the experiments we conduct, Louvain performs better than HAC.

After running, Louvain might produce a number of singleton clusters with few instances. It is not proper to call these clusters new relation types, so we label these instances the same as their closest labeled neighbors.

Finally, we want to explain the reason why we do not use some other common clustering methods like K-Means, Mean-Shift and Ward's (Ward Jr, 1963) method of HAC: these methods calculate the centroid of several points during clustering by merely averaging them. However, the relation vectors in our model are high-dimensional, and the distance metric described by RSN is non-linear. Consequently, it is not proper to calculate the centroid by simply averaging the vectors.

4 Experiments

In this section, we conduct several experiments on real-world RE datasets to show the effectiveness of our models, and give a detailed analysis to show its advantages.

4.1 Dataset

In experiments, we use FewRel (Han et al., 2018) as our first dataset. FewRel is a human-annotated dataset containing 80 types of relations, each with 700 instances. An advantage of FewRel is that every instance contains a unique entity pair, so RE models cannot choose the easy way to memorize the entities.

We use the original train set of FewRel, which contains 64 relations, as labeled set with predefined relations, and the original validation set of FewRel, which contains 16 new relations, as the unlabeled set with novel relations to extract. We then randomly choose $1,600$ instances from the unlabeled set as the test set, with the rest labeled and unlabeled instances considered as the train set.

The second dataset we use is FewRel-distant, which contains the distantly-supervised data obtained by the authors of FewRel before human annotation. We follow the split of FewRel to obtain the auto-labeled train set and unlabeled train set. For evaluation, we use the human-annotated test set of FewRel with $1,600$ instances. Unlabeled instances already existing in this test set are removed from the unlabeled train set of FewRel-distant. Finally, the auto-labeled train set contains $323,549$ relational instances, and the unlabeled train set contains $60,581$ instances.

A previous OpenRE work reports performance on an unpublic dataset called NYT-FB (Marcheggiani and Titov, 2016). However, it has several shortcomings compared with FewRel-distant. First, NTY-FB's test set is distantly-supervised and is noisy for instance-level RE. Moreover, instances in NYT-FB often share entity pairs or relational phrases, which makes it much easier for relation clustering. Therefore, we think the results on FewRel-distant are convincing enough for Distantly-supervised OpenRE.

4.2 Implementation Details

Data Sampling. The input of RSN should be a pair of sampled instances. For the unlabeled set, the only possible sampling method is to select two instances randomly. For the labeled set, however, random selection would result in too many different-relation pairs, and cause severe biases for RSN. To solve this problem, we use downsampling. In our experiments, we fix the percentage of same-relation pairs in every labeled data batch as 6%.

Let us denote this percentage number as the sample ratio for convenience. Experimental results show that the sample ratio decides RSN's tendency to predict larger or smaller clusters. In other words, it controls the granularity of the predicted relation types. This phenomenon suggests a potential application of our model in hierarchical relation extraction. However, we leave any serious discussion to future work.

Hyperparameter Settings. Following (Lin et al., 2016) and (Zeng et al., 2014), we fix the less influencing hyperparameters for sentence encoding as their reported optimal values. For word embeddings, we use pre-trained 50-dimensional Glove (Pennington et al., 2014) word embeddings. For position embeddings, we use random-initialized 5-dimensional position embeddings. During training, all the embeddings are trainable. For the neural network, the number of feature maps in the convolutional layer is 230. The filter length is 3. The activation function after the max-pooling layer is ReLU, and the activation functions after FC layers are sigmoid. Besides, we adopt two regularization methods in the CNN module. We put a dropout layer right after the embedding layer as (Miyato et al., 2016). The dropout rate is 0.2. We also impose L2 regularization on the convolutional layer and the FC layer, with parameters of 0.0002 and 0.001 respectively. Hyperparameters for virtual adversarial training are just the same as (Miyato et al., 2016) proposed.

At the same time, major hyperparameters are selected with grid search according to the model performance on a validation set. Specifically, the validation set contains 10,000 randomly chosen sentence pairs from the unlabeled set (i.e. 16 novel relations) and does not overlap with the test set. The model is evaluated according to the precision of binary classification of sentence pairs on the validation set, which is an estimation for models' clustering ability. We do not use F1 during model validation because the clustering steps are time-consuming.

For optimization, we use Adam optimizer (Kingma and Ba, 2014) with a learning rate of 0.0001, which is selected from $\{0.1, 0.01, 0.001, 0.0001, 0.00001\}$. The batch size is 100 selected from $\{25, 50, 100\}$. For hyperparameters in Equation 9 and Equation 10, λ_v is 1.0 selected from $\{0.1, 0.5, 1.0, 2.0\}$ and λ_u is 0.03 selected from $\{0.01, 0.02, 0.03, 0.04, 0.05\}$.

For baseline models, original papers do grid search for all possible hyperparameters and report the best result during testing. We follow their settings and do grid search directly on the test set.

4.3 Experiment Results on OpenRE

In this section, we demonstrate the effectiveness of our RSN models by comparing our models with state-of-the-art clustering-based OpenRE methods. We also conduct ablation experiments to detailedly investigate the contributions of different mechanisms of Semi-supervised RSN and Distantly-supervised RSN.

Baselines. Conventional clustering-based OpenRE models usually cluster instances by either clustering their linguistic features (Lin and Pantel, 2001; Yao et al., 2012; Elsahar et al., 2017) or imposing reconstruction constraints (Yao et al., 2011; Marcheggiani and Titov, 2016). To demonstrate the effectiveness of our RSN models, we compare our models with two state-of-the-art models:

(1) HAC with re-weighted word embeddings (RW-HAC) (Elsahar et al., 2017): RW-HAC is the state-of-the-art feature clustering model for OpenRE. The model first extracts KB types and NER tags of entities as well as re-weighted word embeddings from sentences, then adopts principal component analysis (PCA) to reduce feature dimensionality, and finally uses HAC to cluster the concatenation of reduced feature representations.

(2) Discrete-state variational autoencoder (VAE) (Marcheggiani and Titov, 2016): VAE is the state-of-the-art reconstruction-based model for OpenRE via unlabeled instances. It optimizes a relation classifier by reconstructing entities from pairing entities and predicted relation types. Rich features including entity words, context words, trigger words, dependency paths, and context POS tags are used to predict the relation type.

RW-HAC and VAE both rely on external linguistic tools to extract rich features from plain texts. Specifically, we first align entities to Wikidata and get their KB types. Next, we preprocess the instances with part-of-speech (POS) tagging, named-entity recognition (NER), and dependency parsing with Stanford CoreNLP (Manning et al., 2014). It is worth noting that these features are only used by baseline models. Our models, in contrast, only use sentences and entity pairs as inputs.

Evaluation Protocol. In evaluation, we use B^3 metric (Bagga and Baldwin, 1998) as the scoring function. B^3 metric is a standard measure to balance the precision and recall of clustering tasks, and is commonly used in previous OpenRE works (Marcheggiani and Titov, 2016; Elsahar et al., 2017). To be specific, we use F_1 measure, the harmonic mean of precision and recall.

First, we report the result of supervised RSN with different clustering methods. Specifically, **SN** represents the original RSN structure, **HAC** and **L** indicate HAC and Louvain clustering introduced in Sec. 3.3. The result shows that Louvain performs better than HAC, so in the following experiments we focus on using Louvain clustering.

Next, for Semi-supervised and Distantly-supervised RSN, we conduct various combinations of different mechanisms to verify the contribution of each part. **(+C)** indicates that the model is powered up with conditional entropy minimization, while **(+V)** indicates that the model is powered up with virtual adversarial training.

	FewRel			FewRel-distant		
Approach	P	R	F_1	P	R	F_1
VAE	17.9	69.7	28.5	17.9	69.7	28.5
RW-HAC	31.8	46.0	37.6	31.8	46.0	37.6
SN-HAC	36.2	53.3	43.1	34.5	53.3	41.5
SN-L	36.5	69.2	47.8	34.6	59.8	43.9
SN-L+V	46.1	77.3	57.8	40.7	52.4	45.8
SN-L+C	47.1	**78.1**	58.8	**42.3**	66.0	51.5
SN-L+CV[1]	**48.9**	77.5	**59.9**	40.8	**74.0**	**52.6**

Table 1: Precision, recall and F1 results (%) for different models. The first two models are baselines. The next five models are different variants of our model.

Experimental Result Analysis. Table 1 shows the experimental results, from which we can observe that:

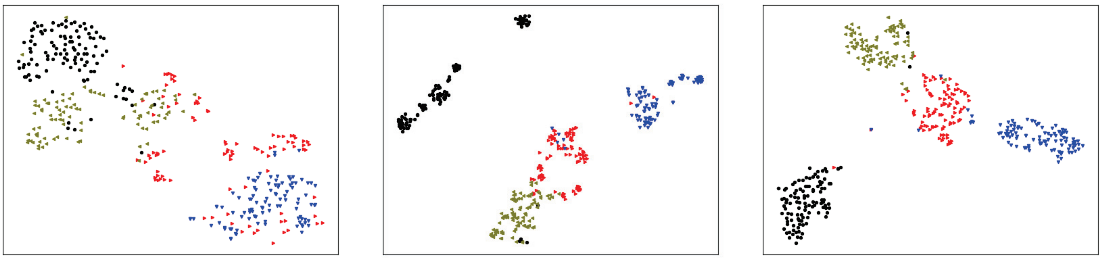
(1) RSN models outperform all baseline models on precision, recall, and F1-score, among which Weakly-supervised RSN (SN-L+CV) achieves state-of-the-art performances. This indicates that RSN is capable of understanding new relations' semantic meanings within sentences.

(2) Supervised and distantly-supervised relational representations improve clustering performances. Compared with RW-HAC, SN-HAC achieves better clustering results because of its supervised relational representation and similarity metric. Specifically, unsupervised baselines mainly use sparse one-hot features. RW-HAC uses word embeddings, but integrates them in a rule-based way. In contrast, RSN uses distributed feature representations, and can optimize information integration process according to supervision.

(3) Louvain outperforms HAC for clustering with RSN, comparing SN-HAC with SN-L. One explanation is that our model does not put additional constraints on the prior distribution of relational vectors, and therefore the relation clusters might have odd shapes in violation of HAC's assumption. Moreover, when representations are not distinguishable enough, forcing HAC to find fine-grained clusters may harm recall while contributing minimally to precision. In practice, we do observe that the number of relations SN-L extracts is constantly less than the true number 16.

(4) Both SN-L+V and SN-L+C improve the performance of supervised or distantly-supervised

[1] Here for FewRel-distant we use Equation 10 rather than Equation 9 as loss, which corresponds to Distantly-supervised RSN, and this brings a minor improvement on F_1 from 52.0% to 52.6%.



(a) RSN

(b) Semi-supervised RSN

(c) Supervised CNN

Figure 4: The t-SNE visualization of the output vectors of CNN modules in our (a) OpenRE model RSN, (b) Semi-supervised RSN facilitated by unlabeled novel-relation data and in (c) a classical RE baseline trained with labeled novel-relation data. All figures visualize the clustering result for 402 instances of 4 novel relations.

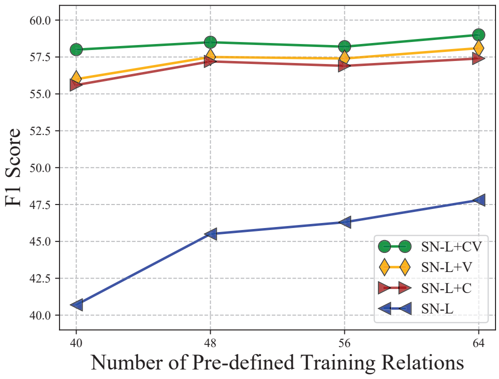


Figure 5: The clustering results with different numbers of pre-defined training relations.

RSN by further utilizing unsupervised corpora. Both semi-supervised approaches bring significant improvements for F_1 scores by increasing the precision and recall, and combining both can further increase the F_1 score.

(5) One interesting observation is that SN-L+V does not outperform SN-L so much on FewRel-distant. This is probably because VAT on the noisy data might amplify the noise. In further experiments, we perform VAT only on unlabeled set and observe improvements on F_1, with SN-L+V from 45.8% to 49.2% and SN-L+CV from 52.0% to 52.6%, which proves this conjecture.

4.4 The Influence of Pre-defined Relation Diversity on Generalizability

In this subsection, we mainly focus on analyzing the influence of pre-defined relation diversity, i.e., the number of relations in the labeled train set. To study this influence, we use FewRel for evaluation and change the number of relations in the labeled train set from 40 to 64 while fixing the total number of labeled instances to $25,000$, and report the clustering results in Figure 5.

Several conclusions can be drawn according to Figure 5. Firstly, a rich variety of labeled relations do improve the performance of our models, especially RSN. The models trained on 64 relations perform better than those trained on 40 relations constantly. Secondly, while the performance of supervised RSN is very sensitive to pre-defined relation diversity, its semi-supervised counterparts suffer much less from the relation number limit. This phenomenon suggests that Semi-supervised RSNs succeed in learning from unlabeled novel-relation data and are more generalizable to novel relations.

4.5 Relational Knowledge Representation Visualization

To intuitively evaluate the knowledge transfer effects of RSN and Semi-supervised RSN, we visualize their relational knowledge representation spaces in the last layer of CNN encoders with t-SNE(Maaten and Hinton, 2008) in Figure 4. We also compare with a supervised CNN trained on $9,600$ labeled instances of novel relations, which suggests the optimal relational knowledge representation. In each figure, we plot 402 relation instances of 4 randomly-chosen relation types in the test set, and points are colored according to their ground-truth labels.

As we can see from Figure 4, RSN is able to roughly distinguish different relations, and Semi-supervised RSN further facilitated knowledge transfer by optimizing the margin between potential relation clusters during training. As a result, Semi-supervised RSN can extract more distinguishable novel relations, and gains comparable relational knowledge representation ability with supervised CNN.

5 Conclusions and Future Work

In this paper, we propose a new model Relational Siamese Network (RSN) for OpenRE. Different from conventional unsupervised models, our model learns to measure relational similarity from supervised/distantly-supervised data of predefined relations, as well as unsupervised data of novel relations. There are mainly two innovative points in our model. First, we propose to transfer relational similarity knowledge with RSN structure. To the best of our knowledge, we are the first to propose knowledge transfer for OpenRE. Second, we propose Semi/Distantly-supervised RSN, to further perform semi-supervised and distantly-supervised transfer learning. Experiments show that our models significantly surpass conventional OpenRE models and achieve new state-of-the-art performance.

For future research, we plan to explore the following directions: (1) Besides CNN, there are some other popular sentence encoder structures like piecewise convolutional neural network (PCNN) and Long Short-Term Memory (LSTM) for RE. In the future, we can try different sentence encoders in our model. (2) As mentioned above, our model has the potential ability to discover the hierarchical structure of relations. In the future, we will try to explore this application with additional experiments.

6 Acknowledgement

This work is supported by the National Key Research and Development Program of China (No. 2018YFB1004503) and the National Natural Science Foundation of China (NSFC No. 61572273, 61661146007). Ruidong Wu is also supported by Tsinghua University Initiative Scientific Research Program.

References

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *Proceedings of the First Iternational Conference on Language Resources and Evaluation Workshop on Linguistics Coreference*.

Eric Bair. 2013. Semi-supervised clustering methods. *Wiley Interdisciplinary Reviews: Computational Statistics*.

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

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

Vincent D Blondel, Jean Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics*.

Olivier Chapelle and Alexander Zien. 2005. Semi-supervised classification by low density separation. In *Proceedings of AISTATS*.

Lei Cui, Furu Wei, and Ming Zhou. 2018. Neural open information extraction. *arXiv*.

Hady Elsahar, Elena Demidova, Simon Gottschalk, Christophe Gravier, and Frederique Laforest. 2017. Unsupervised open relation extraction. In *Proceedings of European Semantic Web Conference*.

Yves Grandvalet and Yoshua Bengio. 2005. Semi-supervised learning by entropy minimization. In *Proceedings of NIPS*.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. Fewrel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of EMNLP*.

Meng Qu Xiang Ren Hongtao Lin, Jun Yan. 2019. Learning dual retrieval module for semi-supervised relation extraction. In *Proceedings of WWW*.

Shengbin Jia, Yang Xiang, and Xiaojun Chen. 2018. Supervised neural models revitalize the open relation extraction. *arXiv*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv*.

Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. 2015. Siamese neural networks for one-shot image recognition. In *Proceedings of ICML Deep Learning Workshop*, volume 2.

Dekang Lin and Patrick Pantel. 2001. Dirt - discovery of inference rules from text. In *Proceedings of KDD*.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of ACL*.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *JMLRJournal of Statistical Mechanics*.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of ACL*.

Diego Marcheggiani and Ivan Titov. 2016. Discrete-state variational autoencoders for joint discovery and factorization of relations. *Transactions of ACL*.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of ACL-IJCNLP*.

Takeru Miyato, Andrew M Dai, and Ian Goodfellow. 2016. Adversarial training methods for semi-supervised text classification. *arXiv*.

Sachin Pawar, Girish K. Palshikar, and Pushpak Bhattacharyya. 2017. Relation extraction : A survey. *arXiv*.

Jeffrey Pennington, Richard Socher, and Christoper Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of EMNLP*.

Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. 2018. Supervised open information extraction. In *Proceedings of NAACL*.

Joe H Ward Jr. 1963. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*.

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

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

Dian Yu, Lifu Huang, and Heng Ji. 2017. Open relation extraction and grounding. In *Proceedings of IJCNLP*.

Jianbo Yuan, Han Guo, Zhiwei Jin, Hongxia Jin, Xianchao Zhang, and Jiebo Luo. 2017. One-shot learning for fine-grained relation extraction via convolutional siamese neural network. In *Proceedings of BigData*.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, Jun Zhao, et al. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING*.