A Unified Representation Learning Strategy for Open Relation Extraction with Ranked List Loss

Renze Lou[1]*, **Fan Zhang**[2]*, **Xiaowei Zhou**[3], **Yutong Wang**[4], **Minghui Wu**[1] and **Lin Sun**[1]†
[1]Department of Computer Science, Zhejiang University City College, Hangzhou, China
[2]Faculty of Economics, Hitotsubashi University, Tokyo, Japan
[3]Zhejiang Qianyue Digital Technology Co., Ltd, China
[4]Tsinghua Shenzhen International Graduate School, Tsinghua University, Shenzhen, China
marionojump0722@gmail.com; zhangoutstanding@hotmail.com;
tinycs@126.com; wangyt19@mails.tsinghua.edu.cn; {mhwu, sunl}@zucc.edu.cn

Abstract

Open Relation Extraction (OpenRE), aiming to extract relational facts from open-domain corpora, is a sub-task of Relation Extraction and a crucial upstream process for many other NLP tasks. However, various previous clustering-based OpenRE strategies either confine themselves to unsupervised paradigms or can not directly build a unified relational semantic space, hence impacting down-stream clustering. In this paper, we propose a novel supervised learning framework named MORE-RLL (**M**etric learning-based **O**pen **R**elation **E**xtraction with **R**anked **L**ist **L**oss) to construct a semantic metric space by utilizing Ranked List Loss to discover new relational facts. Experiments on real-world datasets show that MORE-RLL can achieve excellent performance compared with previous state-of-the-art methods, demonstrating the capability of MORE-RLL in unified semantic representation learning and novel relational fact detection.

1 Introduction

Relation Extraction (RE) aims to extract pre-defined relational facts from plain text (e.g., *"Mary gave birth to Keller in 1989s."*, RE can extract ***"gave_birth_to"*** between two named entities *"Mary"* and *"Keller"*). It is an important task that can structure a large amount of text data. Therefore, it can benefit for unstructured text data storing and the procedure of many other down-stream NLP tasks or applications, such as knowledge graph construction (Suchanek et al., 2007), information retrieval (Xiong et al., 2017), and logic reasoning (Socher et al., 2013). Nevertheless, with the rapid development of social media and human civilization, novel relationships and new knowledge in open-domain text data are also increasing. Accordingly, the relation types in the open-domain corpora may not be pre-defined, which is hard for RE to handle. To meet the rapid emergence of such novel knowledge, OpenRE emerged as the times required (Banko et al., 2007). The goal of OpenRE is to detect novel relational facts from open-domain datasets. It is a crucial task for updating the human knowledge base and the study of human civilization.

Existing OpenRE methods are divided into two main categories: tagging-based and clustering-based. The tagging-based strategies treat OpenRE as a sequence labeling problem (Banko et al., 2007; Banko and Etzioni, 2008). Still, these methods often extract surface forms that can not be utilized for down-stream tasks (i.e., some sequences have the same relational semantic type, but their phrases generated from tagging-based methods are different because of overly-specific). Comparatively, clustering-based methods aim to identify the rich semantic features in the text, then cluster them into certain relation types. Recently, many efforts have been devoted to exploring clustering-based methods, such as (Yao et al., 2012; Elsahar et al., 2017). Yet, those schemes are laborious and time-consuming because of the high dependence on well-designed features created by hand.

Profited from the substantial improvement of computing power in recent years, neural networks begin to be exploited in clustering-based OpenRE tasks to alleviate the above issues, such as (Simon et al., 2019; Hu et al., 2020; Gao et al., 2020). Even so, these strategies confine themselves to unsupervised

* Equal contribution
† Corresponding author

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1096-1108, Hohhot, China, August 13 - 15, 2021.
(c) Technical Committee on Computational Linguistics, Chinese Information Processing Society of China

or self-supervised paradigms and can not fully benefit from current high-quality human-labeled corpora. Although several unconventional works have gained phenomenal performance, such as (Zhang et al., 2021), the reliance on the extra knowledge for these strategies make it hard for us to compare with. Besides, another supervised scheme learned the similarity metrics from labeled instances and further transferred the relational knowledge to the open-domain scene, namely Relational Siamese Networks (RSNs) (Wu et al., 2019). However, RSNs target learning a similarity classifier rather than building relational representations directly. Thereby, this may impact the efficiency and effect of down-stream clustering.

In order to address these issues, we propose a novel supervised learning framework via a clustering-based scheme driving neural encoder to build rich semantic representation directly. From our insight view, the essential target of the clustering-based OpenRE algorithm is to construct a reasonable semantic space on the open-domain corpora, where all different relational facts can be distinguished clearly. Therefore, the learning of semantic representation is a fundamental part of the whole task. It can not only extend the functionality of the neural encoder (i.e., the semantic space construction ability of the neural model can be used in other scenes, such as classification, etc.) but also bring benefits to downstream clustering.

As a result, we pay attention to the unified semantic representation learning ability of neural encoders. Specifically, we employ deep metric learning to drive the neural encoder to build a distinguishable semantic space on open-domain datasets. However, most prevailing deep metric learning methods, such as triplet loss (Hoffer and Ailon, 2015), N-pair-mc (Sohn, 2016), or Proxy-NCA (Movshovitz-Attias et al., 2017), always bring low yield due to the poor supervision signals from the limited number of training data points. Inspired by (Wang et al., 2019), we chose Ranked List Loss (RLL) instead, which can capture set-based rich supervision signals. Meanwhile, RLL can preserve a better intraclass similarity structure within a hypersphere than other set-based schemes, hence constructing a more desirable semantic space.

Additionally, considering that the open scene corpora is usually full of noise, hence directly transferring knowledge may not be an ideal choice. To enhance the model's robustness, we also design virtual adversarial training for our semantic space construction algorithm. Experiments demonstrate that MORE-RLL can build more distinguishable semantic representations and obtain excellent performances on real-world datasets.

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

- We propose a novel clustering-based OpenRE framework, namely MORE-RLL. The MORE-RLL combines deep metric learning and neural encoder to build a unified relational semantic space to discriminate samples rather than utilize an additional classification layer. Thus, it can handle the enormous undefined relation types in the open-domain corpora and facilitate down-stream clustering to discover valuable novel types. Meanwhile, we adopt a Ranked List Loss to gain more prosperous supervision signals and construct a more desirable semantic space than other prevailing metric learning losses.
- Considering the noise and bias present in the text of open scenarios, we also design virtual adversarial training to enhance the robustness of MORE-RLL instead of directly transferring the knowledge that comes from clean RE datasets to the open-domain corpora.
- Experiments illustrate that the proposed MORE-RLL achieves state-of-the-art performance on real-world datasets, even if the imbalance distribution presents in the test set. Moreover, the visual analysis also demonstrates its excellent ability of relational representation learning and novel knowledge detecting.

2 Methodology

In this section, we will introduce our framework in detail. As shown in Figure 1, we exploit a neural encoder to extract relational representations from a batch of training samples. These sentence-level representations can be taken as relational semantic vectors, indicating the relative locations of facts in the semantic feature space. Then, we use them to calculate the Ranked List Loss (RLL) and gain rich

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1096-1108, Hohhot, China, Augest 13 - 15, 2021.

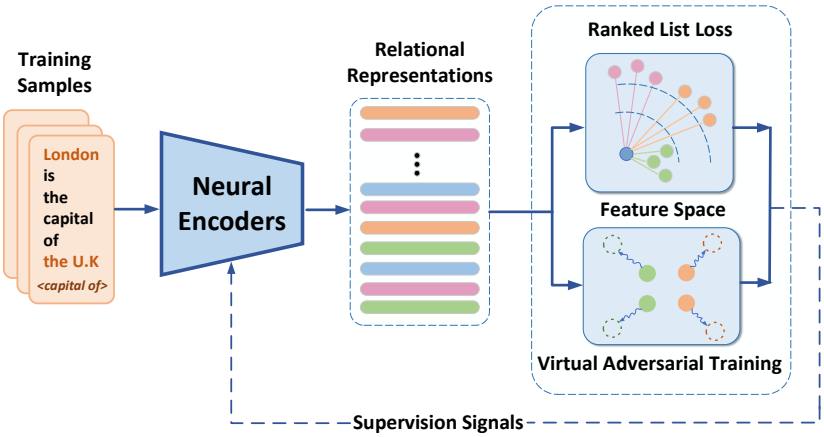


Figure 1: Overall architecture of MORE-RLL.

supervision signals to train the encoder. Besides, we set virtual adversarial training to smooth the feature space to overcome noise in the open scenes. We repeat these steps until the encoder is well-trained and then transfer the prior knowledge from the training samples to the open-domain corpora.

2.1 Neural Encoders

As a vital component of MORE-RLL, the neural encoder aims at extracting semantic representations of relation types from natural language sentences. In this paper, we mainly use CNN as the encoder. Meanwhile, we also experiment with the pre-trained language model to demonstrate the expansibility of our framework.

CNN+GloVe Following (Zeng et al., 2014; Wu et al., 2019), we take the CNN encoder as our primary choice and utilize the pre-trained GloVe embedding (Pennington et al., 2014). To be specific, we firstly use the pre-trained word embedding layer and a randomly initialized position embedding layer to transform the original text sequences. Both these embedding layers are trainable. Then, the outputs of these two embedding layers will be concatenated and passed to a one-dimensional CNN followed by a max-pooling layer. After that, we employ a linear layer to map these raw representations to a high-dimensional semantic space. So far, the structure of our model is the same as that of RSNs (Wu et al., 2019), so we don't detail it. However, unlike RSNs, which utilize an additional linear classifier to predict the similarity of the extracted representation pair, we simply construct such a feature space and prepare for the next step.

BERT Inspired by SelfORE (Hu et al., 2020), which exploited the pre-trained language model, we also choose BERT (Devlin et al., 2018) as our contextual neural encoder. Following the operation proposed by (Soares et al., 2019), we take the relational hidden states of BERT as representations rather than the output of $[CLS]$ token. More specifically, for a sentence $\mathcal{S} = \{s_1, ..., s_T\}$ (where s indicates the token and T is the length of $\mathcal{S}$), we insert four special tokens before and after each entity mentioned in a sentence and get a new sequence:

$$\begin{aligned}\mathcal{S} = [&s_1, .., [E1_{start}], s_p, .., s_q, [E1_{end}],\\ &.., [E2_{start}], s_k, .., s_l, [E2_{end}], .., s_T]\end{aligned} \tag{1}$$

We use this sequence as the input of BERT, and then we concatenate the last hidden states of BERT's outputs corresponding to $[E1_{start}]$, $[E2_{start}]$, take these relational hidden states as our raw representations. Same as what we have mentioned in the CNN encoder, we then use a linear mapping layer to process these representations.

After obtaining the representations extracted by the above neural encoders, we perform L_2 normalization on these high dimensional representations, thus construct a Euclidean semantic space where we can predict the similarity metrics of relations conveniently.

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1096-1108, Hohhot, China, August 13 - 15, 2021.

2.2 Ranked List Loss

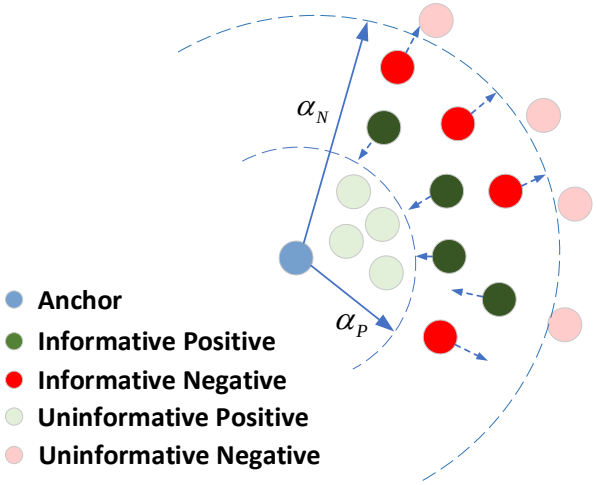


Figure 2: Illustration of the ranked list loss.

So far, we have introduced how to built a Euclidean semantic space. Therefore, the next problem is the optimization of this space, which we will detail in this section.

As we have introduced in the subsection 1, our essential objective is to make the neural encoder gain a unified representation learning ability on high-quality RE corpora. Thus, we adopt deep metric learning as the optimization algorithm on relational semantic space. What's more, considering the limited information from point-based or pair-based metric learning methods (e.g., triplet loss (Hoffer and Ailon, 2015), N-pair-mc (Sohn, 2016)), we try to use a set-based or group-based scheme instead. Inspired by (Wang et al., 2019), we finally choose Ranked List Loss (RLL) to explore set-based similarity structure from a training batch and gain richer supervision signals.

For an anchor selected from the training batch, RLL rank the similarity of all the same type (positive) points before the different categories (negative) points and preserve a preset margin between them. To be specific, given a batch of normalized relation representations $\mathcal{B} = \{r_1, ..., r_m\}$ generated from the neural encoder, and an instance (anchor) r_i in $\mathcal{B}$ (where m indicates the batch size), we expect that the positive points for the anchor in $\mathcal{B}$ can be gathered together while those negative points are the opposite. So we calculate the following formula:

$$\mathcal{L}(r_i, \mathcal{B}; f) = \sum_{r_j \in \mathcal{B}, j \neq i} [(1 - y_{ij})[\alpha_N - d_{ij}]_+ + y_{ij}[d_{ij} - \alpha_P]_+] \quad (2)$$

where f is model parameters; y indicates the relation type, $y_{ij} = 1$ if r_j is a positive point, $y_{ij} = 0$ otherwise; d_{ij} denotes the Euclidean distance between two points; α_P, α_N represent the positive and negative boundary respectively; $[.]_+$ denote the hinge function.

Intuitively, as shown in Figure 2, those positive instances outside the α_P will be pulled closer, while those negative points within the α_N will be pushed further. The remaining uninformative points which have already met our objective will not be taken into count because of the hinge function.

More concisely, for an anchor r_i in $\mathcal{B}$, let's define $\mathcal{L}_P(r_i, \mathcal{B}; f)$ as the total loss of all informative positive points, and $\mathcal{L}_N(r_i, \mathcal{B}; f)$ is the sum of all informative negative samples loss, thus the optimization objective function can be summarized as below:

$$\mathcal{L}_{RLL}(\mathcal{B}; f) = \sum_{r_i \in \mathcal{B}} [(1 - \lambda)\mathcal{L}_P(r_i, \mathcal{B}; f) + \lambda \mathcal{L}_N(r_i, \mathcal{B}; f)] \quad (3)$$

Here, the λ is the balance factor between $\mathcal{L}_P(r_i, \mathcal{B}; f)$ and $\mathcal{L}_N(r_i, \mathcal{B}; f)$. Usually, we set it as 0.5.

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1096-1108, Hohhot, China, Augest 13 - 15, 2021.

Additionally, given r_i as an anchor, there are always numerous informative negative points that can be found in $\mathcal{B}$. To deal with the magnitude difference laying in the negative loss, we follow (Wang et al., 2019), weight the negative examples according to the values of their loss:

$$w_{ij} = \exp[T_n * (\alpha_N - d_{ij})], r_j \in \mathcal{R}_i^{\mathcal{N}} \tag{4}$$

Where $\mathcal{R}_i^{\mathcal{N}}$ represents the set of informative negative samples of r_i, and T_n indicates the temperature factor which controls the degree of weighting these negative samples (because the temperature factor of positive samples, namely T_p, is always set to 0, we don't formalize it). When T_n is 0, every instance will be treated equally. But if set T_n to $+\infty$, the loss function will devote almost all attention to the hardest sample.

Consequently, the $\mathcal{L}_N(r_i, \mathcal{B}; f)$ in (3) can be updated as:

$$\mathcal{L}_N(r_i, \mathcal{B}; f) = \sum_{r_j \in \mathcal{R}_i^{\mathcal{N}}} [\frac{w_{ij}}{\sum_{r_j \in \mathcal{R}_i^{\mathcal{N}}} w_{ij}} (\alpha_N - d_{ij})] \tag{5}$$

After the neural encoder retrieves the supervision signals of $\mathcal{B}$, we sample next m instances sequentially from the training corpus and performed the above operation iteratively.

2.3 Virtual Adversarial Training

Different from many other tasks, there is always bias (e.g., the extreme imbalance of labels) and noise (e.g., spelling mistakes) present in the text of open scenarios. As a result, directly transfer the knowledge from the standardized corpus to an open-domain setting is not an ideal scheme. To address this issue, we design virtual adversarial training (VAT) (Miyato et al., 2018) to smooth the semantic space, hence enhance the model's robustness.

Specifically, for any given sentence S and its original representation r, we first generate a normalized perturbation ξ on the word embedding within S randomly, add it to the original word embedding, and then take this disturbed embedding as the input of encoder to build a new representation $\tilde{r}$. Next, we calculate the gradient g of the Euclidean distance between r and $\tilde{r}$ with respect to the ξ. Then, we regard ϵ times normalized g as the worst-case perturbation $\tilde{\xi}$, where ϵ is a small decimal number we set as 0.02 in all our experiments. Finally, we use $\tilde{\xi}$ to disturb the original embedding of S. In a word, given a batch of samples $\mathcal{B}$, we penalize neural encoder with the following VAT loss:

$$\mathcal{L}_{adv}(\mathcal{B}; f) = \frac{1}{m} \sum_{i=1}^{m} D(F(S_i; f), F(S_i + \tilde{\xi}_i; f)) \tag{6}$$

Where S_i indicates the i sequence, $F(S_i + \tilde{\xi}; f)$ denotes the distributed representation encoded by the neural model, while $F(S_i; f)$ is the original one (namely r_i) and D calculates the Euclidean distance between two representations. Intuitively, we expect that any representation r_i encoded by model in $\mathcal{B}$ is stable as possible under such worst-case perturbations.

Thence, the final objective loss function can be written as:

$$\mathcal{L}(\mathcal{B}; f) = \mathcal{L}_{RLL}(\mathcal{B}; f) + \beta \mathcal{L}_{adv}(\mathcal{B}; f) \tag{7}$$

Where β is a factor that indicates the weight of virtual adversarial training. Same as (Miyato et al., 2018), we set it as 1 practically.

3 Experiment

3.1 Datasets

FewRel is derived from Wikipedia and annotated by crowd workers (Han et al., 2018). Different from most other datasets, the entity pair of each instance in FewRel is unique, which makes the model unable to obtain shortcuts by memorizing the entities. Following the paper (Wu et al., 2019), we choose 64 relations as the train set and randomly select 16 relations with 1600 instances as the test set; the remaining sentences are validation set (as can be seen in Table 1).

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1096-1108, Hohhot, China, August 13 - 15, 2021.

NYT+FB-sup is generated from NYT+FB. The original NYT+FB is built by distant supervision, and its text sequences come from New York Times corpus (Sandhaus, 2008) while the relational types are extracted from Freebase (Bollacker et al., 2008). Following (Simon et al., 2019; Hu et al., 2020), we process the raw data and get the original NYT+FB. Since the whole dataset is built via distant supervision, its labels are full of noise and bias. In order to fit the supervision setting and to better simulate open scenes, we then divide the original dataset again, hence obtaining NYT+FB-sup. Usually, the relations which occur frequently are common categories, and those relations with rare instances are insufficient to be regarded as novel types. Therefore, we select relations with the number of instances between 20 and 2000 as novel relations and append them to the test set. As shown in Table 1, we finally obtained 72 novel relations equally divided between the test and validation set, leaving 190 relations as the train set, which contains both common and scarce types to simulate a real unbalanced environment.

	FewRel		**NYT+FB-sup**	
Partition	relation type	instance	relation type	instance
train	64	44800	190	25521
dev	16	9600	72	8100
test	16	1600	72	8063

Table 1: The statistical information on FewRel and NYT+FB-sup

3.2 Settings

In all our experiments, we use the NVIDIA RTX2080 graphics card. We choose Adam (Kingma and Ba, 2014) for our optimization and fix the learning rate with 3e-4 and 1e-5 on CNN and BERT, respectively. Since the batch size m is a significant factor in our metric learning-based framework, we use the conclusion that comes from 3.5.3. Expressly, we set m to 100, fix both the relation types C and the number of instances for each type K to 10. To solve out-of-memory problems when utilizing BERT, we use parallel training on 4 graphics cards. For the hyperparameter α_P, α_N, we follow the original paper (Wang et al., 2019) and set them as 0.8 and 1.2 separately, hence preserving a margin of 0.4 between these two boundaries. The temperature factor T_n in this work is 10, which is also the same as the original paper. We train our framework with 4 epochs on the training set and adopt early stopping.

The clustering algorithm is a general factor in the OpenRE task, so there can be multiple choices. Since the semantic space constructed by our framework is a normalized Euclidean space, the distance metric described by our model is linear, which is different from (Wu et al., 2019). Therefore, we utilize two commonly used algorithms: K-Means (Hartigan and Wong, 1979) and Mean-Shift (Cheng, 1995). On FewRel, we choose K-means as our downstream clustering algorithm and set the number of clusters as 24. And on NYT+FB-sup, we choose Mean-Shift instead of K-means to deal with the imbalance, which can automatically find clusters based on spatial density. We don't use Louvain (Blondel et al., 2008) or HAC (Ward Jr, 1963) in our framework. The reason is that the Louvain is a graph-based algorithm that is not very compatible with our normalized Euclidean semantic space, while the HAC is highly time-consuming.

What's more, we adopt B^3 F_1 score (Bagga and Baldwin, 1998) as our metrics, which is widely used in previous works (Marcheggiani and Titov, 2016; Elsahar et al., 2017; Wu et al., 2019; Hu et al., 2020). F_1 calculates the harmonic mean of precision and recall, and its value is more affected by the lower one, which can fairly demonstrate the performance of the model.

3.3 Main Results

We compare our MORE-RLL with four previous state-of-the-art baselines (Marcheggiani and Titov, 2016; Elsahar et al., 2017; Wu et al., 2019; Hu et al., 2020) on two datasets. All these models are evaluated on the test set to show their performance. The main results can be seen in Table 2, the scores of all algorithms are the highest among the statistical testings (some borrowed from the original paper). From Table 2, we can draw the following conclusions:

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1096-1108, Hohhot, China, August 13 - 15, 2021.

Method	FewRel			NYT+FB-sup		
	Prec.	Rec.	F1	Prec.	Rec.	F1
VAE (Marcheggiani and Titov, 2016)	17.9	69.7	28.5	20.3	40.7	27.1
RW-HAC (Elsahar et al., 2017)	31.8	46.0	37.6	25.2	33.9	28.9
SelfORE (Hu et al., 2020)	50.8	51.6	51.2	30.9	46.4	37.1
RSNs (Wu et al., 2019)	48.9	77.5	59.9	31.1	**52.0**	38.8
MORE-RLL(GloVe+CNN)	57.1	68.0	62.0	39.1	49.1	43.5
MORE-RLL(BERT)	**70.1**	**79.6**	**74.5**	**48.7**	50.8	**49.7**

Table 2: The results on FewRel and NYT+FB-sup.

- Benefiting by the rich supervision signals come from the labeled RE corpora (even the distant-supervised annotations), MORE-RLL(GloVe+CNN) outperforms all unsupervised or self-supervised methods on both datasets. The results indicate the effectiveness of prior knowledge transfer and our unified semantic representation learning strategy, which can be conducive to novel type-detection in open scenarios.
- MORE-RLL(GloVe+CNN) outperforms RSNs on precision and F_1. However, compared with RSNs, the superiority of MORE-RLL is not obvious on recall. Because the clustering method adopted by RSNs is Louvain (Blondel et al., 2008), which constantly produces coarse-grained clustering results, as mentioned in (Wu et al., 2019). Since the essential objective of OpenRE is to detect valuable novel relations, the quality of relation types detected by the model is more significant than the quantity. Therefore, the impressive precision of MORE-RLL also indicates its capability of high-quality knowledge discovery.
- The F_1 scores of all methods on NYT+FB-sup are lower than the results on FewRel. Even SelfORE (Hu et al., 2020), which has achieved admirable performance on NYT+FB, also has a poor performance. This phenomenon shows that NYT+FB-sup can simulate the real open scene and presents a challenging problem for all models. However, even in a hard setting, MORE-RLL can still maintain a better performance than others. This result proves that MORE-RLL is robust to noise in the dataset and can distinguish those ambiguous novels and rare classes in open-domain corpora.
- To demonstrate the expansibility of our framework, we also adopt BERT as our neural encoder. Owing to the powerful extracting ability of the pre-trained language model, the performance of our framework has been greatly improved. In fact, there are multiple encoders that can be adopted in our framework. Due to the limitation of the space, we do not extend it here.

3.4 Visualization Analysis

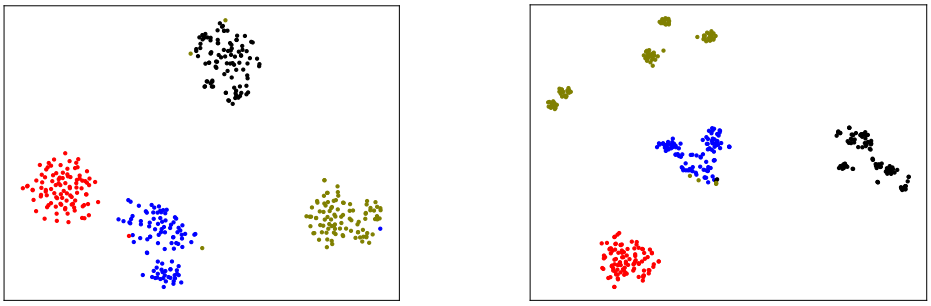
In order to intuitively demonstrate the capability of MORE-RLL on semantic representation learning, we visualize the semantic space of relational representations with t-SNE (Maaten and Hinton, 2008). Specifically, We randomly sample 4 relation types from the test set of FewRel and NYT+FB-sup, respectively, 100 instances per type, and construct representations from these instances on both MORE-RLL(GloVe+CNN) and RSNs, then color these representations according to their ground-truth types.

The Figure 3 illustrates the visualization on FewRel. The semantic space of MORE-RLL is more distinguishable that almost all four types can preserve the intraclass similarity within a hypersphere while leaving a distinct margin between any two categories. In contrast, RSNs attempt to shrink the representations of the same type into one point. Thus, the distribution of points for RSNs in each cluster is denser than MORE-RLL. However, excessive attention to the similarity between point pairs may drop the intraclass similarity structure, so it is easier for RSNs to divide samples of the same category into multiple subcategories.

Visualization on NYT+FB-sup can be seen in Figure 4. The semantic space on NYT+FB-sup is harder to construct compared with Figure 3. Because of the distant-supervision labels and the extreme imbalance in the training set, it is easy for both MORE-RLL and RSNs to be confused with some relation

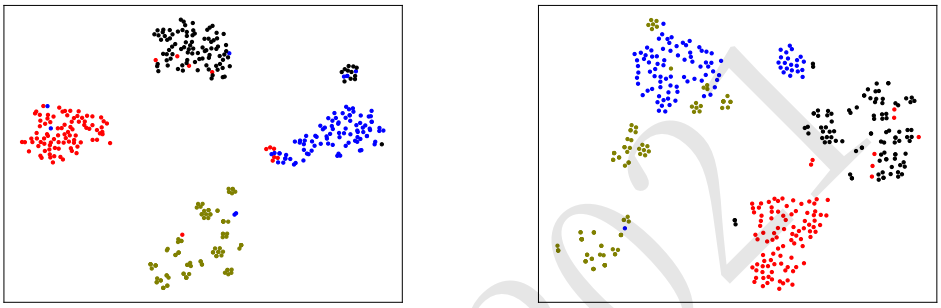
Proceedings of the 20th China National Conference on Computational Linguistics, pages 1096-1108, Hohhot, China, Augest 13 - 15, 2021.

types. For example, the olive points in Figure 4 cannot maintain the intraclass structures well. However, MORE-RLL can still preserve a relatively distinguishable semantic space even in such a challenging setting.



(a) MORE-RLL (b) RSNs

Figure 3: The t-SNE visualization on FewRel.



(a) MORE-RLL (b) RSNs

Figure 4: The t-SNE visualization on NYT+FB-sup.

3.5 Other Empirical Studies

In this subsection, we conduct several experiments: 1). Compare the VAT with other regularization strategies; 2). Compare the RLL with the other metric losses; 3). Explore the vital factor for the batch content. In all the following experiments, we use CNN as the neural encoder.

3.5.1 Comparing with Other Regularization Methods

In this paragraph, we compare VAT with other widely used regularization strategies to show the effectiveness of it, that is:

* $\boldsymbol{L_2}$ regularization. We adopt L_2 regularization on both the CNN and the linear mapping layer with the 2e-4 and 1e-3 weight decay ratio, respectively.
* **Dropout** (Srivastava et al., 2014). We apply Dropout on the word embedding, the drop rate we use here is 0.3.
* **R**andom **P**erturbation **T**raining (RPT) is a naive smoothing scheme that disturbs the original input with an isotropic distribution. Since the RPT can be regarded as a downgraded version of our VAT, we take the initial perturbation ξ (we have mentioned in 2.3) as a random perturbation and add it to the original word embedding within each sequence.

The hyparameters of each strategy above are chosen via greedy trials. For each of them, we report the best score among 10 experiments.

As can be seen in Table 3, with the help of VAT, MORE-RLL can achieve better performance. In contrast, some other widely used regularization methods (e.g., L_2, Dropout) can't bring phenomenal improvement due to the shallow structure of our framework. Meanwhile, suffering from the influence

of isotropic distribution, the RPT also achieve barely satisfactory compared with VAT. Besides the theoretical and performance promise, the VAT calculates "virtual adversarial direction" (as mentioned in the original paper (Miyato et al., 2018)), so there is no dependency on the ground-truth information for VAT. Consequently, it is more suitable to use VAT in the open domain than other label-dependency methods, e.g., adversarial training (Goodfellow et al., 2014).

	FewRel			NYT+FB-sup		
Method	Prec.	Rec.	F1	Prec.	Rec.	F1
MORE-RLL w/o VAT	56.6	60.3	58.4	36.4	48.9	41.8
MORE-RLL w/o VAT + L_2	52.9	66.4	58.9	39.0	45.5	42.0
MORE-RLL w/o VAT + Dropout	53.9	66.8	59.6	**39.5**	45.2	42.1
MORE-RLL w/o VAT + RPT	**57.7**	62.7	60.1	38.7	43.9	41.1
MORE-RLL	57.1	**68.0**	**62.0**	39.1	**49.1**	**43.5**

Table 3: The results of adopting different regularization methods.

3.5.2 Comparing with Other Metric learning Losses

In this paragraph, we try to demonstrate the effectiveness of the Ranked List Loss. We compare RLL with other prevailing metric losses:

* **T**riplet **L**oss (**TL**) (Hoffer and Ailon, 2015) is a pair-based metric loss, which aims to pull an anchor closer to a positive point while pushing further from a negative point.
* **N**-**P**air-**M**c **L**oss (**NPML**) (Sohn, 2016) is similar to triplet loss which increases the number of data points used to calculate. Further, it utilizes an efficient batch composition method.
* **P**roxy-**N**CA **L**oss (**PNL**) (Movshovitz-Attias et al., 2017) is a proxy-based loss, which aims at selecting proxies that represent the desirable cluster center of those positive or negative samples.
* **F**acility **L**ocation **L**oss (**FLL**) (Oh Song et al., 2017) is another outstanding set-based metric loss. The author takes the global embedding structure into account and proposes a better optimization strategy on FLL than the greedy algorithm.

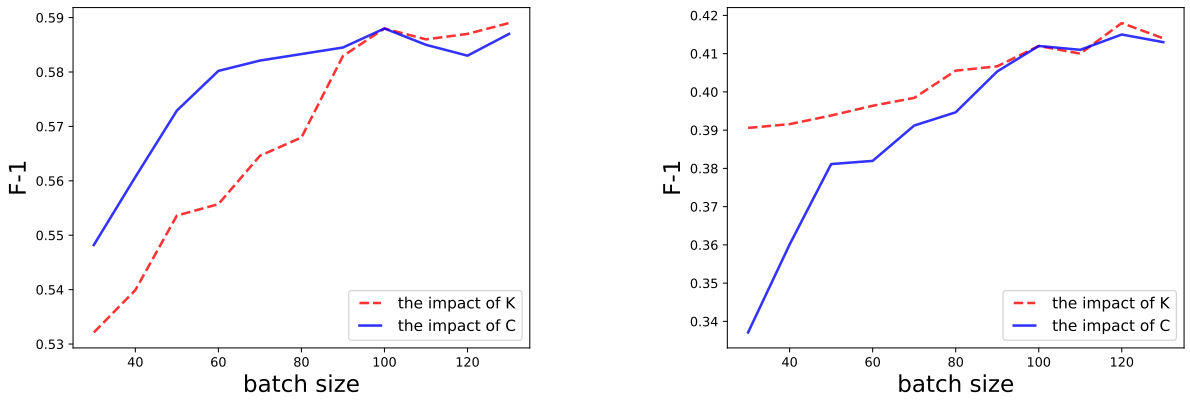
In this experiment, We fix the batch size m of all schemes to 100. All the results reported here are the highest one during 10 experiments. To avoid the influence of other factors, we don't use VAT on MORE-RLL here.

The result is presented in Table 4. It shows that these set-based metric losses do capture richer supervision signals than those pair-based methods. Moreover, RLL performs better than FLL, this mainly owing to the superiority of RLL on the intraclass similarity structure-preserving. At the same time, FLL also suffers from the excessive pursuit of the similarity metric. We also note that the FLL is much more time-consuming than RLL and is sensitive to its hyparameters, making it hard to optimize.

	FewRel			NYT+FB-sup		
Method	Prec.	Rec.	F1	Prec.	Rec.	F1
MORE-TL	37.2	44.2	40.4	32.3	30.2	31.2
MORE-NPML	40.6	48.6	44.2	32.5	33.6	33.1
MORE-PNL	46.3	55.4	50.4	31.7	33.1	32.4
MORE-FLL	50.0	57.0	53.3	32.7	41.3	36.5
MORE-RLL w/o VAT	**56.6**	**60.3**	**58.4**	**36.4**	**48.9**	**41.8**

Table 4: The results of adopting different metric losses.

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1096-1108, Hohhot, China, August 13 - 15, 2021.



(a) FewRel (b) NYT+FB-sup

Figure 5: The influence of training batch content on FewRel and NYT+FB-sup. The solid blue line represents the impact of changing in K while the dashed red line indicates the influence of changing in C

3.5.3 The Vital Factor for Batch Content

Unlike many other OpenRE methods, the training batch content is a significant part of our metric learning-based strategy. As mentioned in (Fehervari et al., 2019), almost all metric losses can benefit from a larger batch size due to the more prosperous signals. However, there are two critical factors for the batch content in RLL, that is, the number of relation types C and the number of instances K in each relation type (i.e., $m = C \times K$, m is the batch size). Though Wang et al. (Wang et al., 2019) has experimented with the impact of batch content on RLL, it is still ambiguous which factor for batch content is more contributing.

Accordingly, we change the training batch size in a larger range on both datasets to reveal the impact of C and K more clearly. To be specific, we fix one of these two factors to 10 and range the other from 3 to 13, then plot the results to show each factor's impact. All scores we report in this paragraph are averaged from 10 experiments.

As can be seen in Figure 5, we can draw the following conclusions:

- The batch size is a significant hyparameters in our framework. With a larger batch size, the neural encoder does capture more prosperous supervision signals. However, this improvement will not be noticeable when the batch size is sufficiently large. Hence, we recommend to take 100 ($C = K = 10$) as an ideal batch size on both FewRel and NYT+FB-sup.
- The result on FewRel illustrates that K seems to have a greater impact than C, i.e., the dashed red line rises more rapidly with batch size growth. Meanwhile, the batch content with larger K (the solid blue line) usually brings more performance improvement. As we have mentioned in 2.2, there is a vast gap between positive and negative loss present in RLL. Since the FewRel is a human-labeled corpus, the larger K is, the more high-quality positive supervision signals RLL can bring. Thus K is a more vital factor when training for these gold-label corpora.
- On the contrary, C plays a vital role when the training dataset is NYT+FB-sup. Unlike FewRel, the NYT+FB-sup corpus is full of noise, so it is difficult for RLL to generate beneficial positive signals. Taking another route, increasing the diversity of relationship types can dramatically increase the negative signals. Though there is still noise present, RLL is more likely to find those informative and beneficial negative points in it. Hence the neural encoder can have more opportunities to obtain instructive semantic signals. In this case, a larger C may be a desirable choice.

4 Conclusion

In this paper, we propose a novel supervised learning framework for open-domain relation extraction, namely MORE-RLL. It aims to make the neural network gain a unified relational representation encoding ability and handle the open-domain relational instances. We utilize deep metric learning to drive the

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1096-1108, Hohhot, China, August 13 - 15, 2021.

neural model to learn relational representations directly, thereby conducive to downstream clustering efficiency. Moreover, we set virtual adversarial training to enhance the robustness of the neural encoder. Our experiments show that MORE-RLL achieves state-of-the-art performance on real-world RE corpora comparing with previous methods and can build a more desirable semantic space. These all demonstrate the capability of our scheme on relational representation learning and novel relation detection.

Acknowledgements

This work was supported by the Key Research and Development Project of Zhejiang Province (No.2021C01164) and the National Innovation and Entrepreneurship Training Program for College Students (No.202113021002, No.202113021003).

References

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *The first international conference on language resources and evaluation workshop on linguistics coreference*, volume 1, pages 563–566. Citeseer.

Michele Banko and Oren Etzioni. 2008. The tradeoffs between open and traditional relation extraction. In *Proceedings of ACL-08: HLT*, pages 28–36.

M Banko, MJ Cafarella, S Soderland, M Broadhead, and O Etzioni. 2007. Open information extraction from the web in: Proceedings of the 20th international joint conference on artificial intelligence.

Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008.

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.

Yizong Cheng. 1995. Mean shift, mode seeking, and clustering. *IEEE transactions on pattern analysis and machine intelligence*, 17(8):790–799.

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, Elena Demidova, Simon Gottschalk, Christophe Gravier, and Frederique Laforest. 2017. Unsupervised open relation extraction. In *European Semantic Web Conference*, pages 12–16. Springer.

Istvan Fehervari, Avinash Ravichandran, and Srikar Appalaraju. 2019. Unbiased evaluation of deep metric learning algorithms. *arXiv preprint arXiv:1911.12528*.

Tianyu Gao, Xu Han, Ruobing Xie, Zhiyuan Liu, Fen Lin, Leyu Lin, and Maosong Sun. 2020. Neural snowball for few-shot relation learning. In *AAAI*, pages 7772–7779.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

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. *arXiv preprint arXiv:1810.10147*.

John A Hartigan and Manchek A Wong. 1979. Algorithm as 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)*, 28(1):100–108.

Elad Hoffer and Nir Ailon. 2015. Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition*, pages 84–92. Springer.

Xuming Hu, Lijie Wen, Yusong Xu, Chenwei Zhang, and Philip S Yu. 2020. Selfore: Self-supervised relational feature learning for open relation extraction. *arXiv preprint arXiv:2004.02438*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 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.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. 2018. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993.

Yair Movshovitz-Attias, Alexander Toshev, Thomas K Leung, Sergey Ioffe, and Saurabh Singh. 2017. No fuss distance metric learning using proxies. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 360–368.

Hyun Oh Song, Stefanie Jegelka, Vivek Rathod, and Kevin Murphy. 2017. Deep metric learning via facility location. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5382–5390.

Jeffrey Pennington, Richard Socher, and Christopher D 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.

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 2019 - 57th Annual Meeting of the Association for Computational Linguistics*, pages 1378–1387, Florence, Italy, July. Association for Computational Linguistics.

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

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.

Kihyuk Sohn. 2016. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in neural information processing systems*, pages 1857–1865.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706.

Xinshao Wang, Yang Hua, Elyor Kodirov, Guosheng Hu, Romain Garnier, and Neil M Robertson. 2019. Ranked list loss for deep metric learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5207–5216.

Joe H Ward Jr. 1963. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244.

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.

Chenyan Xiong, Russell Power, and Jamie Callan. 2017. Explicit semantic ranking for academic search via knowledge graph embedding. In *Proceedings of the 26th international conference on world wide web*, pages 1271–1279.

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 (Volume 1: Long Papers)*, pages 712–720.

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1096-1108, Hohhot, China, Augest 13 - 15, 2021.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2335–2344.

Kai Zhang, Yuan Yao, Ruobing Xie, Xu Han, Zhiyuan Liu, Fen Lin, Leyu Lin, and Maosong Sun. 2021. Open hierarchical relation extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5682–5693.

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1096-1108, Hohhot, China, Augest 13 - 15, 2021.