PROTOTYPICAL REPRESENTATION LEARNING FOR RELATION EXTRACTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Recognizing relations between entities is a pivotal task of relational learning. Learning relation representations from distantly-labeled dataset is difficult because of the abundant label noise and complicated expressions in human language. This paper aims to learn predictive, interpretable, and robust relation representations from distantly-labeled data that are effective in different settings, including supervised, distantly supervised, and few-shot learning. Instead of solely relying on the supervision from noisy labels, we propose to learn prototypes for each relation from contextual information to best explore the intrinsic semantics of relations. Prototypes are representations in the feature space abstracting the essential semantics of relations between entities in sentences. We learn prototypes based on objectives with clear geometric interpretation, where the prototypes are unit vectors uniformly dispersed in a unit ball, and statement embeddings are centered at the end of their corresponding prototype vectors on the surface of the ball. This approach allow us to learn meaningful, interpretable prototypes for the final classification. Results on several relation learning tasks show that our model significantly outperforms the previous state-of-the-art models. We further demonstrate the robustness of the encoder and the interpretability of prototypes with extensive experiments.

1 INTRODUCTION

Relation extraction aims to predict relations between entities in sentences, which is crucial for understanding the structure of human knowledge and automatically extending knowledge bases (Cohen & Hirsh, 1994; Bordes et al., 2013; Zeng et al., 2015; Schlichtkrull et al., 2018). Learning representations for relation extraction is challenging due to the rich forms of expressions in human language, which usually contains fine-grained, complicated correlations between marked entities. Although many works are proposed to learn representations for relations from well-structured knowledge (Bordes et al., 2013; Lin et al., 2015; Ji et al., 2015), when we extend the learning source to be unstructured distantly-labeled text (Mintz et al., 2009), this task becomes particularly challenging due to spurious correlations and label noise (Riedel et al., 2010).

This paper aims to learn predictive, interpretable, and robust relation representations from large-scale distantly labeled data. We propose a prototype learning approach, where we impose a prototype for each relation and learn the representations from the semantics of each statement, rather than solely from the noisy distant labels. Statements are defined as sentences expressing relations between two marked entities. As shown in Figure 1, a prototype is an embedding in the representation space capturing the most essential semantics of different statements for a given relation. These prototypes essentially serve as the center of data representation clusters for different relations and are surrounded by statements expressing the same relation.

We learn the relation and prototype representations based on objective functions with clear geometric interpretations. Specifically, our approach assumes prototypes are unit vectors uniformly dispersed in a unit ball, and statement embeddings are centered at the end of their corresponding prototype vectors on the surface of the ball. We propose statement-statement and prototype-statement objectives to ensure the *intra-class compactness* and *inter-class separability*. Unlike conventional cross-entropy loss that only uses *instance-level* supervision (which could be noisy), our objectives exploit the interactions among *all statements*, leading to a predictively powerful encoder with more interpretable

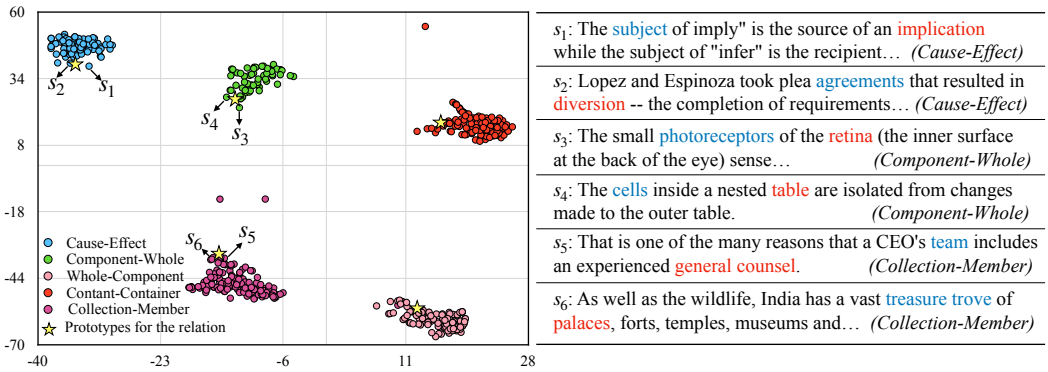


Figure 1: The t-SNE visualization of relation representations and corresponding prototypes learned by our model. In the right part, $s_{1:6}$ are examples of input statements, where red and blue represent the head and tail entities, and the *italics* in parenthesis represents the relation between them.

and robust representations. We future explore these properties of learned representations with extensive experiments.

We apply our approach using a pretraining fine-tuning paradigm. We first pretrain a relation encoder with prototypes from a large-scale distantly labeled dataset, then fine-tune it on the target dataset with different relational learning settings. We further propose a probing dataset, FuzzyRED dataset (Section 4.3), to verify if our method can capture the underlying semantics of statements. Experiments demonstrate the predictive performance, robustness, and interpretability of our method. For predictive performance, we show that our model outperforms existing state-of-the-art methods on supervised, few-shot, and zero-shot settings (Section 4.4 and 4.2). For robustness, we show how the model generalize to zero-shot setting (Section 4.2) and how the prototypes regularize the decision boundary (Section 4.5). For interpretability, we visualize the learned embeddings and their corresponding prototypes (Section 4.4) and show the cluster clearly follow the geometric structure that the objective functions impose. The source code and the FuzzyRED will be publicly available for reproducibility.

2 Related Work

Relation learning could be mainly divided into three categories. The logic-based methods reason relations via symbolic logic rules, with adopting probabilistic graphical models or inductive logic systems to learn and infer logic rules orienting to relations (Cohen & Hirsh, 1994; Wang et al., 2015; Yang et al., 2017; Chen et al., 2018; Kazemi & Poole, 2018; Qu & Tang, 2019). The graph-based methods encode entities and relations into low-dimensional continues spaces to capture structure features of KBs. (Nickel et al., 2011; Bordes et al., 2013; Yang et al., 2015; Wang et al., 2014; Nickel et al., 2016; Ji et al., 2015; Lin et al., 2015; Trouillon et al., 2016; Sun et al., 2019; Balažević et al., 2019). The text-based methods have been widely explored recently, focusing on extracting semantic features from text to learn relations.

The conventional methods to learn relations from the text are mainly supervised models, like statistical supervised models (Zelenko et al., 2003; GuoDong et al., 2005; Mooney & Bunescu, 2006). As deep neural networks have gained much attention then, a series of neural supervised models have been proposed (Liu et al., 2013; Zeng et al., 2014; Xu et al., 2015; Santos et al., 2015; Zhang & Wang, 2015; Verga et al., 2016; Verga & McCallum, 2016; Li et al., 2019). To address the issue of the insufficiency of annotated data, distant supervision has been applied to automatically generate a dataset with heuristic rules (Mintz et al., 2009). Accompanying with auto-labeled data, massive noise will be introduced into models. Accordingly, various denoising methods have been explored to reduce noise effects for distant supervision (Zeng et al., 2015; Lin et al., 2016; Jiang et al., 2016; Han et al., 2018a; Wu et al., 2017; Qin et al., 2018a;b; Feng et al., 2018). Further efforts pay attention to learning relations from the text in various specific scenarios, such as zero-few-shot scenarios (Levy et al., 2017; Han et al., 2018b; Gao et al., 2019; Soares et al., 2019; Ye & Ling, 2019) and open-domain scenarios (Banko et al., 2007; Fader et al., 2011; Mausam et al., 2012; Del Corro & Gemulla, 2013;

Angeli et al., 2015; Stanovsky & Dagan, 2016; Mausam, 2016; Cui et al., 2018; Shinyama & Sekine, 2006; Elsahar et al., 2017; Wu et al., 2019).

However, up to now, there are few strategies of relation learning that could be adapted to any relation learning tasks.Soares et al. (2019) introduces a preliminary method that learns relation representations from unstructured corpus. And the assumption that two same entities must express the same relation is imprecise, because it does not consider the semantics of contextual information. This paper abstracts the task from the text as a metric and prototype learning problem and proposes a interpretable method for relation representation learning. Apart from the differences in methods and task scenarios, row-less universal schema (Verga & McCallum, 2016) has a similar spirit with our method, where relation type embeddings guide clustering of statements for entity pair representations and embeddings of entity pairs are regarded as an aggregation of relations.

Although the intuition that assigns a representative prototype for each class is similar to some previous studies like Prototypical Networks (Snell et al., 2017) (ProtoNet), there exist some essential distinctions between our approach and the ProtoNet. The ProtoNet computes the prototype of each class as the average of the embeddings of all the instance embeddings, that could be regarded as a non-linear version of the nearest class mean approach (Mensink et al., 2013). The idea of mean-of-class prototypes could also be traced to earlier studies in machine learning (Graf et al., 2009) and cognitive modeling and psychology (Reed, 1972; Rosch et al., 1976). In our method, prototypes and relation encoder are collaboratively and dynamically trained by three objectives, which divides the high-dimensional feature space into disjoint manifolds. The ProtoNet performs instance-level classification to update the parameters, which is not robust for noisy labels. Our method carries out a novel prototype-level classification to effectively regularize the semantic information. The prototype-level classification reduces the distortion caused by noisy labels of the decision boundary fitted by NNs. Moreover, the ProtoNet is designed for few-zero-shot learning and our methods are more like a semi-supervised pre-training approach that could be applied to supervised, few-shot, and transfer learning scenarios. Furthermore, the ProtoNet does not contain the metric learning among instances, and our method simultaneously optimizes the prototype-statement and statement-statement metrics. We also make a geometry explanation of our method (Section 3.1).

3 METHOD

Our method follows a pretraining-finetuning paradigm. We first pretrain an encoder with distantly labeled data, then fine-tune it to downstream learning settings. We start with the pretraining phase. Given a large-scale, distantly labeled dataset containing training pairs (w, r) where $w = [w_1, ..., w_n]$ is a relation statement, i.e., a sequence of words with a pair of entities $[h, t]$ marked, h being the head and t being the tail, $\exists i_1, i_2, h = w_{i_1:i_2}, \exists j_1, j_2, t = w_{j_1:j_2}$, r is the relation between h and t. r is a discrete label and is probably noisy. We aim to learn an encoder $\text{Enc}_\phi(\cdot)$ parameterized by ϕ that encodes w into a representation $s \in \mathbb{R}^m$, m being the dimension:

$$s = \text{Enc}_\phi(w). \tag{1}$$

A prototype z for relation r is an embedding in the same metric space with s that abstracts the essential semantics of r. We use $[(z^1, r^1), ..., (z^K, r^K)], z^k \in \mathbb{R}^m$ to denote the set of prototype-relation pairs. K is the number of different relations. z^k is the prototype for relation r^k and superscripts denote the index of relation type. Given a batch $\mathcal{B} = [(s_1, r_1), ..., (s_N, r_N)]$, N is the batch size and subscripts denote batch index, the similarity metric between two statement embeddings $d(s_i, s_j)$, and between a statement embedding and a prototype $d(s, z)$ are defined as:

$$d(s_i, s_j) = 1/(1 + \exp(\frac{s_i}{||s_i||} \cdot \frac{s_j}{||s_j||})), \quad d(s, z) = 1/(1 + \exp(\frac{s}{||s||} \cdot \frac{z}{||z||})). \tag{2}$$

Geometrically, this metric is based on the *angles of the normalized embeddings restricted in a unit ball*. We will explain the geometric implication of this metric in the next section. Inspired by Soares et al. (2019), we define a contrastive objective function between statements:

$$\mathcal{L}_{\text{S2S}} = -\frac{1}{N^2} \sum_{i,j} \frac{\exp(\delta(s_i, s_j) d(s_i, s_j))}{\sum_{j'} \exp((1 - \delta(s_i, s_{j'})) d(s_i, s_{j'}))}, \tag{3}$$

where $\delta(s_i, s_j)$ denotes if s_i and s_j corresponds to the same relations, i.e., given $(s_i, r_i), (s_j, r_j)$, $\delta(s_i, s_j) = 1$ if $r_i = r_j$ else 0. The computation of the numerator term forces that statements with

same relations to be close, and the denominator forces that those with different relations are dispersed. When constructing the batch $\mathcal{B}$, we equally sample all relation types to make sure the summation in the denominator contains all relations. Essentially, equation 3 ensures *intra-class compactness* and *inter-class separability* in the representation space. Additionally, equation 3 contrast one positive sample to all the negative samples in the batch, which puts more weights on the negative pairs. We find it empirically more effective than previous objectives like Soares et al. (2019).

3.1 Learning Prototypes

Now we discuss the objective functions for learning prototypes. Denote $\mathcal{S} = [s_1, ..., s_N]$ as the set of all embeddings in the batch $\mathcal{B}$, given a fixed prototype z^r for relation r, we denote $\mathcal{S}^r$ the subset of all statements s_i in $\mathcal{S}$ with relation r, and $\mathcal{S}^{-r}$ the set of the rest statements. $\mathcal{Z}^{-r}$ as the set of prototypes z' for all other relations except r. We impose two key inductive biases between prototypes and statements: (a) for a specific relation r, the "distance" between z^r and any statements with the same relation r should be less than the "distance" between z^r and any statements with relations $r' \neq r$. (b) the "distance" between z^r and any statements with relation r should be less than the "distance" between any prototypes $z' \in \mathcal{Z}^{-r}$ and statements with relation r. To realize these two properties, we define two objective functions:

$$\mathcal{L}_{\text{S2Z}} = -\frac{1}{N^2} \sum_{s_i \in \mathcal{S}^r, s_j \in \mathcal{S}^{-r}} \big[\log d(z^r, s_i) + \log(1 - d(z^r, s_j))\big], \tag{4}$$

$$\mathcal{L}_{\text{S2Z'}} = -\frac{1}{N^2} \sum_{s_i \in \mathcal{S}^r, z' \in \mathcal{Z}^{-r}} \big[\log d(z^r, s_i) + \log(1 - d(z', s_i))\big], \tag{5}$$

where equation 4 corresponds to (a) and equation 5 corresponds to (b). These objectives effectively *splits the data representations into K disjoint manifolds centering at different prototypes*. We further highlight the differences between equation 4 and 5 and a conventional cross-entropy loss: there is no interactions between different statements in the cross-entropy loss that solely relies on the *instance level* supervision, which is particularly noisy under a noisy-label setting. On the other hand, our loss functions consider distances between different statements and prototypes, which exploits the interactions between statements. This type of interaction would effectively serve as a regularization to the decision boundary, as we will empirically verify in section 4.5.

Combining equations 2, 3, 4 and 5, we further give a geometric explanation of the representation space: a prototype is a unit vector (because we normalize the vector length in equation 2) starting from the origin and ending at the surface of a unit ball, and statements for that prototypes are unit vectors with approximately same directions centering at the prototype (because our objective functions 3, 4, 5 push them to be close to each other). Under the optimal condition, different prototype vectors would be uniformly dispersed with the angles between them as large as possible (because the distance metric is based on the angle in equation 2 where the minimum is at π, and we maximize distances between prototypes in equation 4 and 5). Consequently, the dataset is clustered with *each cluster centered around the end of one prototype vector* on the surface of the unit ball. The illustration of the intuition is shown in Figure 2.

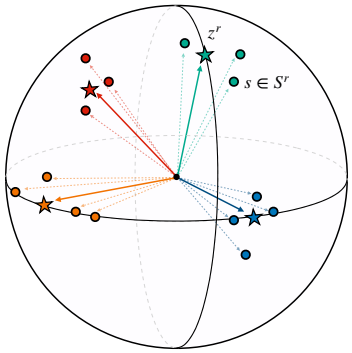


Figure 2: An illustration of the geometric explanation. Stars represent prototypes and circles represent statements.

To further regularize the semantics of the prototypes, we use a prototype-level classification objective:

$$\mathcal{L}_{\text{CLS}} = \frac{1}{K} \sum_k \log p_\gamma(r^k | z^k), \tag{6}$$

where γ denotes the parameters of an auxiliary classifier. Our prototype essentially serves as a regularization averaging the influence of label noises. We further validate this regularization in Section 4.5 by showing that it reduces the distortion of the decision boundary caused by noisy labels. Finally, with hyper-parameters λ_1, λ_2 and λ_3, the full loss is:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{S2S}} + \lambda_2(\mathcal{L}_{\text{S2Z}} + \mathcal{L}_{\text{S2Z}}) + \lambda_3 \mathcal{L}_{\text{CLS}}. \tag{7}$$

3.2 FINE-TUNING ON DIFFERENT LEARNING SETTINGS

We apply our learned encoder and prototypes to two typical relational learning settings: (a) supervised; (b) few-shot. Under the supervised learning setting, having the pretrained encoder at hand, we fine-tune the model on a dataset of a target domain (which could again, be distantly labeled) containing new training pairs (w, r), we encode w to the representation and feed it to a feed-forward classifier:

$$p(r|w) = \text{FF}(\text{Enc}_\phi(w)), \tag{8}$$

where $\text{FF}(\cdot)$ denotes a feed-forward layer, and we train it with the conventional cross-entropy loss.

Under the few-shot learning setting, we assume a slightly different input data format. Specifically, the training set consists of the relations $[r^1, ..., r^K]$ and a list of supporting statements: $s_1^k, ..., s_L^k$ for all r^k. Denote $\mathcal{S}^\star = \{s_l^k\}_{l=1,...,L}^{k=1,...,K}$ all supporting statements for all relations. Given a query statement q, the task is to classify the relation r^* that q expresses. During traing, we use the average distance to different prototypes as the logits of a feed-forward classifier and train the model with cross-entropy loss. During testing, we classify q according to the minimum similarity metrics:

$$s_{l^*}^{k^*} = \arg\min_{s_l^k \in \mathcal{S}^\star} d(s_l^k, \text{Enc}_\phi(q)), \quad r^* = k^*. \tag{9}$$

Note this is equivalent to choosing the argmax of the logits because the logits are formed by the similarity scores. We note that model pre-trained by our approach performs surprisingly well even with no training data for fine-tuning, which is reported in Figure 3 and 4 in Section 4.2.

4 EXPERIMENTS

In order to evaluate the performance of prototypical metrics learning, we conduct extensive experiments on three tasks: supervised relation learning, few-shot learning and our proposed fuzzy relation learning evaluation. We make a comprehensive evaluation and analysis of our method, as well as full-scale comparisons between our work with previous state-of-the-art methods.

4.1 EXPRIMENTAL DETAILS

Pretraining dataset. We prepare the weakly-supervised data by aligning relation tuples from the Wikidata database to Wikipedia articles. Specifically, all entities in wikipedia sentences are identified by a named entity recognition system. For an entity pair (h, t) in a sentence w, if it also exists in the knowledge base with a relation r, then w will be distantly annotated as r. After processing, we collect more than 0.86 million relation statements covering over 700 relations.

Baseline Models. We primarily compare our model with MTB because it is a previous SOTA model. For fair comparison, we re-implement MTB with $\text{BERT}_{\text{base}}$ (Devlin et al., 2018) and pretrain it on our collected distant data as same as our methods. We note under our setting, the reported number of MTB is smaller than its original paper because: (a) the original paper uses much larger dataset than ours (600 million vs 0.86 million); (b) they use $\text{BERT}_{\text{large}}$ encoder. In addition, multiple previous state-of-the-art approaches are picked up for comparison in all the three relation learning tasks (detailed later). For baseline models (PCNN, Meta Network, GNN, Prototypical Network, MLMAN) with publicly available source code, we run the source code manually, and if the produced results are close to those in the original paper, we would select and report the results in the original paper. For ones without open source code (BERT_{EM}, MTB), we re-implement the methods with $\text{BERT}_{\text{base}}$ pre-trained on our distant data and report the results. We also implement a baseline that directly trains $\text{BERT}_{\text{base}}$ on our distant data with cross-entropy loss, namely BERT_{CE} in the section.

The IND baseline. We additionally design a baseline, where the prototypes are pre-computed by vectorizing the extracted relation patterns. In this way, for each statement, we pre-compute a fixed metric as its weight demonstrating the similarity with the corresponding prototype. The IND baseline is primarily for validating the effectiveness of trained prototypes over the rule-based prototypes. Given the distantly-labeled training set, we use the Snowball algorithm from Agichtein & Gravano (2000) to extract relation patterns. This algorithm takes a corpus of relational instances as inputs and outputs a list of patterns and their embeddings for each relation. After getting these patterns, we use patterns to match instances and calculate how many instances can each pattern match. For

each relation label, we select the top-k patterns that match the most instances and use the average of their embeddings as the prototype embedding. Note that these embeddings are not comparable to the sentence embeddings generated by our encoders, so the loss functions in our equation 4 and 5 are not usable for the prototypes generated here. To incorporate the information of the extracted prototype embeddings, we slightly modify equation 3 and use the similarity of instances to the embeddings to weight the instance embeddings. The modified loss function is:

$$\mathcal{L}'_{\text{IND}} = -\frac{1}{N^2}\sum_{i,j}\frac{\exp(\delta(s_i, s_j)d(s'_i, s'_j))}{\sum_{j'}\exp((1-\delta(s_i, s_{j'}))d(s'_i, s'_{j'}))}, \tag{10}$$

$$s'_i = \text{sim}(w_i, z'(w_i)) \cdot s_i, \quad s'_j = \text{sim}(w_j, z'(w_j)) \cdot s_j, \quad s'_j = \text{sim}(w'_j, z'(w'_j)) \cdot s'_j, \tag{11}$$

where w_i denote the corresponding sentence to s_i, $z'(w_i)$ denote the correspond prototype.

Implementation Details For encoding relation statements into a feature space, we adopt deep Transformers (Vaswani et al., 2017), specifically BERT, as E_θ. We take the input and the output of the BERT encoder as follows: given a statement s, we add special entity markers to mark M_h and M_t before the entities, then input the statement and special markers to Enc_ϕ to compute final representations. Note that our framework is designed independently to the encoder choice, and other neural architectures like CNN (Zeng et al., 2014) and RNN (Zhang & Wang, 2015) can be easily adopted as the encoder. We use PyTorch (Paszke et al., 2019) framework to implement our model. All the experiments run on NVIDIA Tesla V100 GPUs. The encoder is optimized with the combination of the prototype-statement and statement-statement metrics as well as the masked language model loss with the settings as follows: For the sake of saving resources, we choose $\mathrm{BERT}_{\mathrm{base}}$ as the backbone instead of $\mathrm{BERT}_{\mathrm{large}}$, although our method sacrifices a considerable part of the performance. We select AdamW (Loshchilov & Hutter, 2018) with the learning rate of $1e-5$ for optimization. Meanwhile, Warmup (Popel & Bojar, 2018) mechanism is used during training. The number of layers is 12, and the number of heads is 12. The hidden size is set to 768 and the batch size is set to 60. We train the model for 5 epochs in each experiment.

4.2 Few-Shot Relation Learning

Dataset We use a large few-shot relation learning dataset FewRel (Han et al., 2018b) in this task. FewRel consists of 70000 sentences (about 25 tokens in each sentence) of 100 relations (700 sentences for each relation), on the basis of Wikipedia. We utilize the official evaluation setting which splits the 100 relations into 64, 16, 20 relations for training, validation, and testing respectively.

Experimental settings We use accuracy as the evaluation metric in this task. The batch size for few-shot training is 4, the training step is 20000 and the learning rate is 2e-5 with AdamW. We follow the original meaning of N-way-K-shot in this paper, please refer to (Han et al., 2018b) for details.

Results and analysis The results for few-shot relation learning are illustrated in Table 1. It is worth noting that we report the average score rather than the best score of each method for the sake of fairness. One observation is that the task is challenging without the effectiveness of pre-trained language models. Prototypical network (CNN) only yields 69.20 and 56.44 accuracies (%) for 5 way 1 shot and 10 way 1 shot respectively. Methods that have achieved significant performances on supervised RE (such as PCNN, GNN, and Prototypical Network) suffer grave declines. MTB learns the metrics between statements based on large pre-trained language models and yield surprising results on this task. Better performance gains by considering the semantic information by applying fixed-prototypes to MTB (IND strategy). The collaborative prototype-based methods are proven to be effective as shown in the last four rows. The results of COL ($\mathcal{L}_{\mathrm{S2S'}} + \mathcal{L}_{\mathrm{Z2S}} + \mathcal{L}_{\mathrm{Z2S'}}$) indicate that the loss of MTB have little influence for the performance. Finally, our final collaborative model achieve the best performance on all the N-way K-shot settings. Our method also outperforms human in 5 way 1 shot setting (92.41 vs 92.22) and 10 way 1 shot setting (86.39 vs 85.88).

Effect of the amount of training data In real-world scenarios, it is costly and labor-consuming to construct a dataset like FewRel containing 70,000 positive instances. Some relations rarely appear in public articles and obtain candidate instances via distance supervision introduces much noise. Therefore, the amount of instances to annotate may much larger than 70, 000. To investigate the ability of few-shot classification when lack of high-quality task-specific training data, we intentionally control the amount of instances in training data and run evaluations with the new training set. Figure 3 and Figure 4 respectively display the accuracy of the 5-way-1-shot task given a limited number

Table 1: Few-shot classification accuracies (%) on FewRel dataset. The last block is the results of the series of our models with prototypes. We use $\mathcal{L}_{\text{Z}}$ to briefly indicate $\mathcal{L}_{\text{Z2S}} + \mathcal{L}_{\text{Z2S}'}$.

Method	**5 way 1 shot**	**5 way 5 shot**	**10 way 1 shot**	**10 way 5 shot**
Finetune (PCNN) (Han et al., 2018b)	45.64	57.86	29.65	37.43
Meta Network (CNN) (Han et al., 2018b)	64.46	80.57	53.96	69.23
GNN (CNN) (Han et al., 2018b)	66.23	81.28	46.27	64.02
Prototypical Network (Han et al., 2018b)	69.20	84.79	56.44	75.55
MLMAN (Ye & Ling, 2019)	82.98	92.66	73.59	87.29
BERT_{EM} (Soares et al., 2019)	88.70	95.01	81.93	90.05
MTB (only $\mathcal{L}_{\text{S2S}'}$) (Soares et al., 2019)	89.09	95.32	82.17	91.73
BERT_{CE}	91.02	95.40	84.95	91.43
IND ($\mathcal{L}_{\text{S2S}'}$ + fixed prototypes)	89.90	95.42	82.47	91.55
COL (Only $\mathcal{L}_{\text{Z}}$)	90.40	94.73	84.27	91.58
COL ($\mathcal{L}_{\text{Z}} + \mathcal{L}_{\text{CLS}}$)	91.12	95.45	85.10	91.75
COL ($\mathcal{L}_{\text{S2S}'} + \mathcal{L}_{\text{Z}} + \mathcal{L}_{\text{CLS}}$)	91.08	95.52	85.83	92.18
COL Final ($\mathcal{L}_{\text{S2S}} + \mathcal{L}_{\text{Z}} + \mathcal{L}_{\text{CLS}}$)	**92.51**	**95.88**	**86.39**	**92.76**

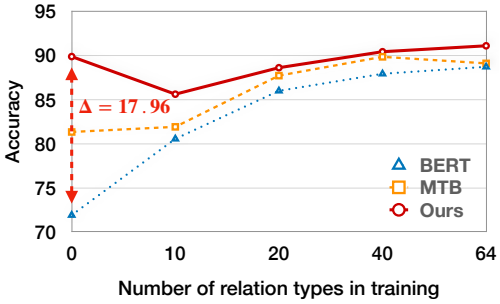


Figure 3: Impact of # of relation types

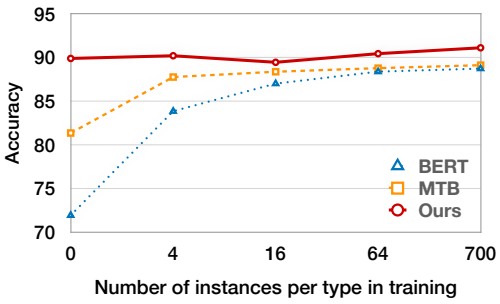


Figure 4: Impact of # of instances per type

of relation types or amount of instances of each relation in training data. As shown in the figures, pre-trained relation encoders outperform the basic BERT_{EM} model when short of training data. Encoder learned with our proposed method shows large superiority over the others, when there is no training data (0), the absolute improvement of our framework is **17.96**. Besides, the accuracy increase with the increment of training data regardless of providing more relations or more instances of each relation. However, the diversity of relations is more important, since the performance in Figure 4 outperforms those in Figure 3 while the former is trained on a much smaller training data. Lack of diversity even hurts the performance when applying relation encoder which is learned with prototype-statement metric. Similar conclusions have been mentioned in Soares et al. (2019).

4.3 Fuzzy Relation Evaluation

Dataset Aimed at evaluating relation representation learning on the noisy weakly supervised corpus, we propose a new evaluation dataset named FuzzyRED to quantitatively evaluate the capture of relational information. FuzzyRED is constructed aimed at the shortcoming of distant supervision, that is not all the statements contain the same entity pairs express the same relation. All the instances are from the weakly-supervised data and manually annotated binary labels to judge if the sentence semantically expresses the relation. For example, for "Jack was born in *London*, *UK*", even if the relation *Capital* is correct between the entities in knowledge base, the sentence does not express the *capital* relationship and will be labeled as False in FuzzyRED. The dataset contains 1000 manually annotated instances covering 20 relations. For detailed information please refer to Appendix A

Experimental settings The pre-trained relation encoders are directly utilized on FuzzyRED and expected to separate the false positive instances from true positives. For each relation, we calculate the minimum of statement-statement distances among all true positives as the threshold T. Then for each instance s to be classified, we sample [1] K (k-shot) true positive instances $s' \in \{s_1, s_2, ..., s_K\}$ and compare the average score of $a = p(l = 1|s, s')$ with T. If $a > T$, the instance is classified

[1] For each instance, we sample 100 times and report the average accuracy.

as positive. FuzzyRED is considerably challenging since it is designed for the issue of distant supervision and directly reflects the gap between semantic relations and annotation information.

Results The accuracies of FuzzyRED are reported in Table 2. The accuracy of labels annotated according to the DS assumption is denoted by DS, which indicates that the majority of the dataset are noise. From the experimental results, we could firstly observe that all the relation encoders suffer from the noise issue and get lower results. The impact of noise increases with the k increasing. But Table 2 still shows that pre-trained relation encoders do detect a few noise data thus they all yield better performance than DS. Taking consideration of noise by prototypes leads to better performances, and the fixed prototypes play a small role to detect the complex semantics that express relations. Even though the COL model achieves remarkable improvement, the evaluation task is still very challenging because of the fuzziness of relation semantics.

Table 2: Accuracies (%) on FuzzyRED.

Method	1-shot	3-shot	5-shot
DS	46.3	46.3	46.3
MTB	51.3	50.7	50.7
IND	51.7	50.8	50.7
COL Final	**53.2**	**52.1**	**52.1**

4.4 Supervised Relation Learning

Dataset We use classic benchmark dataset SemEval 2010 Task 8 (Hendrickx et al., 2010) as the dataset for supervised relation learning. SemEval 2010 Task 8 contains nine different pre-defined bidirectional relations and a `None` relation, thus the task could id a 19-way classification. For model selection, we randomly sample 1500 instances from the official training data as the validation set.

Experimental settings For evaluation, we use standard precision, recall, and F-measure for supervised RE task. Since the annotation schema of SemEval dataset differs from the data for the encoder training, we fine-tune the relation encoder in SemEval training data. The batch size is set to 20, the training epoch is 30, the learning rate is 1e-5 with Adam (Kingma & Ba, 2014). We note that the numbers for MTB are different from their paper because their original setting uses BERTLarge with 600 million instances and is trained on TPUs. Given the restricted resources at hand, to make fair comparison, we re-implemented their model with BERTBase and pretrain it in our setting (0.86 million instances).

Table 3: Accuracies (%) on SemEval 2010 Task 8.

Method	P	R	F
Bi-RNN (Zhang & Wang, 2015)	-	-	79.6
Self-attention (Bilan & Roth, 2018)	-	-	84.8
BERT (Soares et al., 2019)	86.0	89.1	87.4
MTB (Soares et al., 2019)	86.5	89.0	87.7
IND	86.5	**89.7**	**88.0**
COL Final	**87.9**	88.2	**88.0**

Results The results for supervised RE are reported in Table 3. As a classic benchmark task, results on SemEval 2010 Task 8 are already relatively high compared with other supervised RE tasks. The BERT based models outperform the traditional model significantly. By fine-tuning the relation encoder in Semeval training data, we gain 0.6 percent improvement over the original BERT model. We also calculate the Silhouette Coefficient (Rousseeuw, 1987) to evaluate the denseness of the learned representations for one class. The computation is $\frac{\beta-\alpha}{\max(\alpha,\beta)}(-1,1)$, where α is the mean distance between a sample and all other points in the same class and β is the mean distance between a sample and all other points in the next nearest class. The BERT model yields -0.2438 and our model yields 0.3307. Since the SemEval data contains rich semantical information, it is meaningful to visualize the relation representations produced by the trained relation encoders. In this part, we use t-SNE (Maaten & Hinton, 2008) method to project the 768-dimensional relation representations of the test set of SemEval data to 2-dimensional points. The visualization is illustrated in Figure 5, where each round marker represents a statement and a star marker represents a prototype. First, the visualization shows

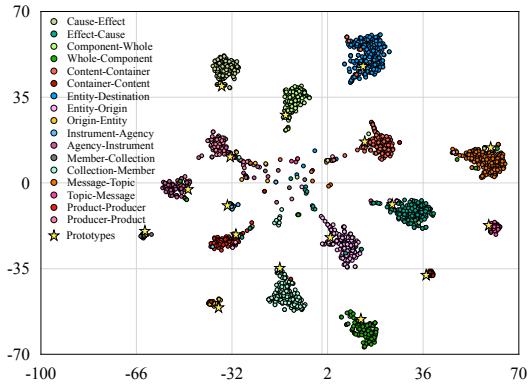


Figure 5: A t-SNE visualization of the relation representations and prototypes of SemEval data.

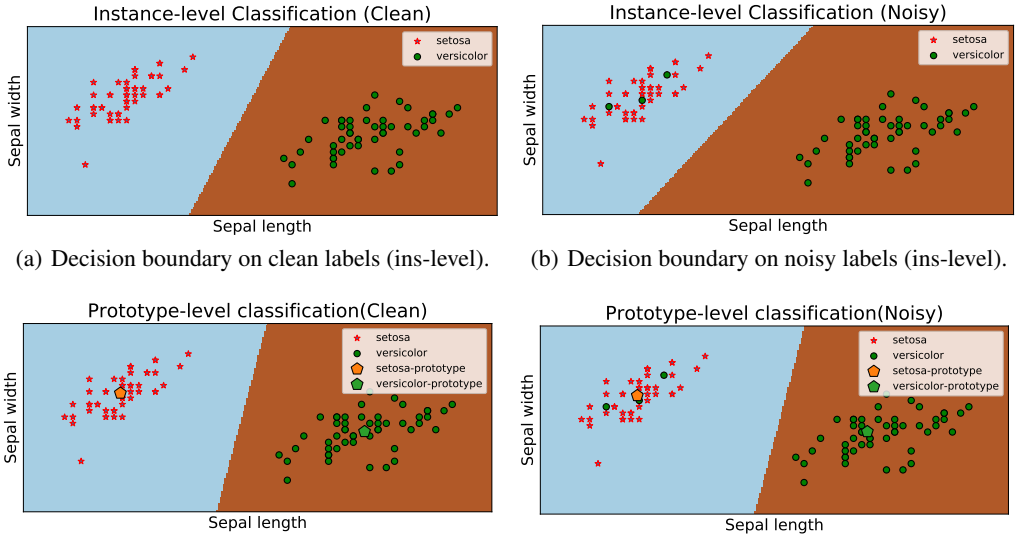
that statements are well clustered and the prototypes are effectively learned for all the relations. Another interesting observation is that the visualization shows visible directions and it is identical to our geometric explanation Section 3.1. We can imagine that project Figure 2 to the 2-dimensional space will result in Figure 5.

4.5 A TOY EXPERIMENT OF DECISION BOUNDARIES FOR PROTOTYPICAL LEARNING

In section 3.1, we mention that the prototype-level classification (equation 6) reduces the disortion of the decision boundary caused by noisy labels. In this section, we carry out a toy experiment to explore the point. We perform a simple binary classification on the iris dataset (Blake, 1998). Here, we use the instance-level linear classifier and the prototype-level classifier. For the prototype-level classifier, the decision function could be written as

$$g(x) = \text{sign}\, \|(x - z_-)\|^2 - \|(x - z_+)\|^2 \,, \tag{12}$$

where x is a data point and z_- and z_+ represent the prototypes for two classes. Equation 12 is also equivalent to a linear classifier $g(x) = \text{sign}(w^\top x + b)$ with $w = z_+ - z_-$ and $b = \frac{\|z_-\|^2 - \|z_+\|^2}{2}$. Prototypes z are computed as the classical mean-of-class (Reed, 1972) method, thus the decision boundary could be easily depicted. We focus on the distortion of the decision boudaries, as shown in Figure 6, when the instances are wrongly labeled, the instance-level boundary is easily impacted and overfitted by noisy labels (Figure 6(b)). And the boundary of prototype-level classifier is hardly affected because of the regularization, it is an interesting phenomenon and we believe it worth further theoretical and empirical study. The results also demonstrate the potential of our model for the generalization of other standard/noisy/few-shot classification tasks.



(a) Decision boundary on clean labels (ins-level). (b) Decision boundary on noisy labels (ins-level).

(c) Decision boundary on clean labels (proto-level). (d) Decision boundary on noisy labels (proto-level).

Figure 6: A set of experiments on iris dataset to illustrate the distortion of decision boundaries for instance-level and prototype-level learning.

5 CONCLUSION

In this paper, we re-consider the essence of relation representation learning and propose an effective method for relation learning directly from the unstructured text with the perspective of prototypical metrics. We contrastively optimize the metrics between statements and infer a prototype to abstract the core features of a relation class. With our method, the learned relation encoder could produce predictive, interpretable, and robust representations over relations. Extensive experiments are conducted to support our claim, and the method also shows the good potential of generalization. In future work, we plan to carry out experiments on other classification tasks (e.g., Computer Vision) across different settings (standard, few-shot, noisy and uneven classes).

REFERENCES

Eugene Agichtein and Luis Gravano. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the ACM DL*, pp. 85–94, 2000.

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. Leveraging linguistic structure for open domain information extraction. In *Proceedings of ACL-IJCNLP*, pp. 344–354, 2015.

Ivana Balažević, Carl Allen, and Timothy M Hospedales. Hypernetwork knowledge graph embeddings. In *Proceedings of NIPS*, pp. 553–565, 2019.

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

Ivan Bilan and Benjamin Roth. Position-aware self-attention with relative positional encodings for slot filling. In *Proceedings of CoRR*, 2018.

Catherine Blake. Uci repository of machine learning databases. *http://www. ics. uci. edu/˜ mlearn/MLRepository. html*, 1998.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Proceedings of NIPS*, pp. 2787–2795, 2013.

Wenhu Chen, Wenhan Xiong, Xifeng Yan, and William Yang Wang. Variational knowledge graph reasoning. In *Proceedings of NAACL-HLT*, pp. 1823–1832, 2018.

William W Cohen and Haym Hirsh. Learning the classic description logic: Theoretical and experimental results. In *Principles of KR*, pp. 121–133, 1994.

Lei Cui, Furu Wei, and Ming Zhou. Neural open information extraction. In *Proceedings of ACL*, pp. 407–413, 2018.

Luciano Del Corro and Rainer Gemulla. Clausie: clause-based open information extraction. In *Proceedings of WWW*, pp. 355–366, 2013.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL*, pp. 4171–4186, 2018.

Hady Elsahar, Elena Demidova, Simon Gottschalk, Christophe Gravier, and Frederique Laforest. Unsupervised open relation extraction. In *Proceedings of ESWC*, pp. 12–16, 2017.

Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. In *Proceedings of EMNLP*, pp. 1535–1545, 2011.

Jun Feng, Minlie Huang, Li Zhao, Yang Yang, and Xiaoyan Zhu. Reinforcement learning for relation classification from noisy data. In *Proceedings of AAAI*, pp. 5779–5786, 2018.

Tianyu Gao, Xu Han, Hao Zhu, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. Fewrel 2.0: Towards more challenging few-shot relation classification. In *Proceedings of EMNLP*, pp. 6249–6254, 2019.

Arnulf BA Graf, Olivier Bousquet, Gunnar Rätsch, and Bernhard Schölkopf. Prototype classification: Insights from machine learning. *Neural computation*, 21(1):272–300, 2009.

Zhou GuoDong, Su Jian, Zhang Jie, and Zhang Min. Exploring various knowledge in relation extraction. In *Proceedings of ACL*, pp. 427–434, 2005.

Xu Han, Zhiyuan Liu, and Maosong Sun. Neural knowledge acquisition via mutual attention between knowledge graph and text. In *Proceedings of AAAI*, pp. 4832–4839, 2018a.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. Fewrel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of EMNLP*, pp. 4803–4809, 2018b.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of SemEval*, pp. 33–38, 2010.

Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of ACL*, pp. 687–696, 2015.

Xiaotian Jiang, Quan Wang, Peng Li, and Bin Wang. Relation extraction with multi-instance multi-label convolutional neural networks. In *Proceedings of COLING*, pp. 1471–1480, 2016.

Seyed Mehran Kazemi and David Poole. Relnn: A deep neural model for relational learning. In *Proceedings of AAAI*, pp. 6367–6375, 2018.

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

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. Zero-shot relation extraction via reading comprehension. In *Proceedings of CoNLL*, pp. 333–342, 2017.

Ziran Li, Ning Ding, Zhiyuan Liu, Haitao Zheng, and Ying Shen. Chinese relation extraction with multi-grained information and external linguistic knowledge. In *Proceedings of ACL*, pp. 4377–4386, 2019.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of AAAI*, pp. 2181–2187, 2015.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. Neural relation extraction with selective attention over instances. In *Proceedings of ACL*, volume 1, pp. 2124–2133, 2016.

ChunYang Liu, WenBo Sun, WenHan Chao, and Wanxiang Che. Convolution neural network for relation extraction. In *Proceedings of ADMA*, pp. 231–242, 2013.

I Loshchilov and F Hutter. Fixing weight decay regularization in adam. In *Proceedings of ICLR*, 2018.

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

Mausam, Michael Schmitz, Stephen Soderland, Robert Bart, and Oren Etzioni. Open language learning for information extraction. In *Proceedings of EMNLP-CoNLL*, pp. 523–534, 2012.

Mausam Mausam. Open information extraction systems and downstream applications. In *Proceedings of IJCAI*, pp. 4074–4077, 2016.

Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriela Csurka. Distance-based image classification: Generalizing to new classes at near-zero cost. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2624–2637, 2013.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of ACL*, pp. 1003–1011, 2009.

Raymond J Mooney and Razvan C Bunescu. Subsequence kernels for relation extraction. In *Proceedings of NIPS*, pp. 171–178, 2006.

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

Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. Holographic embeddings of knowledge graphs. In *Proceedings of AAAI*, pp. 1955–1961, 2016.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Proceedings of NeuRIPS*, pp. 8024–8035, 2019.

Martin Popel and Ondřej Bojar. Training tips for the transformer model. *The Prague Bulletin of Mathematical Linguistics*, 110(1):43–70, 2018.

Pengda Qin, Weiran Xu, and William Yang Wang. Dsgan: generative adversarial training for distant supervision relation extraction. In *Proceedings of ACL*, pp. 496–505, 2018a.

Pengda Qin, Weiran Xu, and William Yang Wang. Robust distant supervision relation extraction via deep reinforcement learning. In *Proceedings of ACL*, pp. 2137–2147, 2018b.

Meng Qu and Jian Tang. Probabilistic logic neural networks for reasoning. In *Proceedings of NIPS*, pp. 7710–7720, 2019.

Stephen K Reed. Pattern recognition and categorization. *Cognitive psychology*, 3(3):382–407, 1972.

Sebastian Riedel, Limin Yao, and Andrew McCallum. Modeling relations and their mentions without labeled text. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 148–163. Springer, 2010.

Eleanor Rosch, Carolyn B Mervis, Wayne D Gray, David M Johnson, and Penny Boyes-Braem. Basic objects in natural categories. *Cognitive psychology*, 8(3):382–439, 1976.

Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

Cicero Nogueira dos Santos, Bing Xiang, and Bowen Zhou. Classifying relations by ranking with convolutional neural networks. In *Proceedings of ACL-IJCNLP*, pp. 626–634, 2015.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *Proceedings of ESWC*, pp. 593–607, 2018.

Yusuke Shinyama and Satoshi Sekine. Preemptive information extraction using unrestricted relation discovery. In *Proceedings of NAACL-HLT*, pp. 304–311, 2006.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Proceedings of NIPS*, pp. 4077–4087, 2017.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of ACL*, pp. 2895–2905, 2019.

Gabriel Stanovsky and Ido Dagan. Creating a large benchmark for open information extraction. In *Proceedings of EMNLP*, pp. 2300–2305, 2016.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. In *Proceedings of ICLR*, 2019.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *Proceedings of ICML*, pp. 2071–2080, 2016.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of NIPS*, pp. 5998–6008, 2017.

Patrick Verga and Andrew McCallum. Row-less universal schema. In *Proceedings of ACL*, pp. 63–68, 2016.

Patrick Verga, David Belanger, Emma Strubell, Benjamin Roth, and Andrew McCallum. Multilingual relation extraction using compositional universal schema. In *Proceedings of NAACL*, pp. 886–896, 2016.

William Yang Wang, Kathryn Mazaitis, Ni Lao, and William W Cohen. Efficient inference and learning in a large knowledge base. *Machine Learning*, 100:101–126, 2015.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of AAAI*, pp. 1112–1119, 2014.

Ruidong Wu, Yuan Yao, Xu Han, Ruobing Xie, Zhiyuan Liu, Fen Lin, Leyu Lin, and Maosong Sun. Open relation extraction: Relational knowledge transfer from supervised data to unsupervised data. In *Proceedings of EMNLP-IJCNLP*, pp. 219–228, 2019.

Yi Wu, David Bamman, and Stuart Russell. Adversarial training for relation extraction. In *Proceedings of EMNLP*, pp. 1778–1783, 2017.

Yan Xu, Lili Mou, Ge Lizhang2015relation, Yunchuan Chen, Hao Peng, and Zhi Jin. Classifying relations via long short term memory networks along shortest dependency paths. In *Proceedings of EMNLP*, pp. 1785–1794, 2015.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. In *Proceedings of ICLR*, 2015.

Fan Yang, Zhilin Yang, and William W Cohen. Differentiable learning of logical rules for knowledge base reasoning. In *Proceedings of NIPS*, pp. 2319–2328, 2017.

Zhi-Xiu Ye and Zhen-Hua Ling. Multi-level matching and aggregation network for few-shot relation classification. In *Proceedings of ACL*, pp. 2872–2881, 2019.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. Kernel methods for relation extraction. *Proceedings of JMLR*, 3(Feb):1083–1106, 2003.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, Jun Zhao, et al. Relation classification via convolutional deep neural network. In *Proceedings of COLING*, pp. 2335–2344, 2014.

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

Dongxu Zhang and Dong Wang. Relation classification via recurrent neural network. In *Proceedings of CoRR*, 2015.

A FUZZY RELATION EVALUATION DATASET

A.1 DISTANT SUPERVISION

Even though our framework is adaptive for a variety of data types, weakly (distantly) supervised data is the most suitable in terms of quantity and quality. Generally, weakly supervised data is constructed on the basis of a large unstructured textual data (e.g. Wikipedia) and a large structured knowledge base (e.g. Wikidata) via the method of distant supervision. First, all the entities are identified by a named entity recognition system which could tag persons, organizations and locations, etc. Those entities form an entity set $\mathcal{E}$ that is mentioned in Section 2. Then if for a sentence I, the entity pair (h, t) also exists in the knowledge base with a relation r, then I will be distantly annotated as r. After the distant supervision, all the relations form a relation set $\mathcal{R}$.

A.2 MOTIVATION OF FUZZYRED

The distant supervision assumption demonstrates that if two entities participate in a relation, then all the sentences that mention these two entities express that relation. While obtaining plenty of weakly (distant) supervised data, this assumption has also become the primary drawback of distant supervision. Generally, one entity pair in the distant dataset may correspond to multiple relations in different textual environments. For example, the entity pair *(Steve Jobs, California)* has three relations, which are *Place of Birth*, *Place of Work*, and *Place of Residence*. Thus, instances of these corresponding relations prone to introducing false-positive noise. We named such relations as fuzzy relations. Intuitively, conducting extraction on fuzzy relations is more challenging. To empirically evaluate if encoder trained on distant supervised could really learn the correct semantic from the noisy distant supervised corpus and perform well on Fuzzy relations, we present a novel dataset, namely, Fuzzy Relation Extraction Dataset (FuzzyRED). In this section, We will firstly introduce the construction process for FuzzyRED and report some statistics.

A.3 CONSTRUCTION PROCESS OF FUZZYRED

Broadly speaking, the dataset is selected from a distant supervised corpus and judged by humans to determine whether each instance really expresses the same relation as the distant supervision labels. For instance, sentence "*Jobs* was born in *California*" really expresses the supervision label "Place of Birth" while sentence "*Jobs* lived in *California*" does not. The detailed process is as follows:

1. First, following the distant supervision assumption, we align the relation triples in Wikidata with Wikipedia articles to generate massive relation instances.
2. We count the number of corresponding relations for each entity pair, denote as $\mathrm{Count_R(pair)}$. For example, the entity pair (Steve Jobs, California) has three relations, which are *Place of Birth*, *Place of Work*, and *Place of Residence*. Larger $\mathrm{Count_R(pair)}$ implies a higher risk of introducing noise. Thus, we consider a pair whose $\mathrm{Count_R}(pair)$ larger than two as an ambiguous seed of the relations that it is involved in.
3. Then, for each relation, we count the number of ambiguous seeds, denote as $\mathrm{Count_S(rel)}$. The bigger $\mathrm{Count_S(rel)}$ is, the more likely the relation will get the wrong data, which will lead to a worse learning effect of the relation.
4. We sort the relations according to $\mathrm{Count_S(rel)}$ and select the top 20 relations as fuzzy relations of FuzzyRED. For each relation, 50 instances are randomly selected to annotate. For the specific annotation process, each annotator strictly judges whether the sentence could express the given relation according to the definition of the relation in Wikidata.

A.4 STATISTICS OF FUZZYRED

In FuzzyRED, there are 20 different kinds of relations, and 50 instances for each relation. As mentioned above, every instance is manually classified as true positive (TP) or false positive (FP). Table 4 shows the statistics of FP rates, which are calculated as $\frac{\mathrm{count(FP)}}{\mathrm{count(TP)+count(FP)}}$. As shown in Table 4, the average FP rate is more than 50 percent, in other words, the majority of the data are FPs. Hence, it is challenging to learn a relation extraction model from FuzzyRED if it doesn't take the

Table 4: Statistics of false positive (FP) rate of the raw data of FuzzyRED. An FP statement means that the statement does not express the relation but is distantly annotated the relation.

	Median	Average	Maximum	Minimum
False Positive Rate (%)	49.0	54.1	94.0	19.0

noise problem into consideration. In this paper, we empirically evaluate different relation encoders by conducting a binary classification task. A better encoder is expected to distinguish TPs from FPs. Specifically, a model based on such relation encoder should predict a higher probability for TPs.

B ALGORITHMS

In Section 3.2, we introduce the adaptation of different downstream scenarios for our framework, which trains an encoder Enc_ϕ. In this section, we provide the algorithms for the three relation learning scenarios.

Algorithm 1 reports the training of supervied relation learning.

Algorithm 1 Training for Supervised Relation Extraction

Input: Supervised dataset $S_r = \{s_{1:n}\}$, statement encoder Enc_ϕ
while not converge **do**
  Sample mini-batches $S_{batches}$ from S_r.
  for S_{batch} **in** $S_{batches}$ **do**
    for s **in** S_{batch} **do**
      $\boldsymbol{s} = \mathrm{Enc}_\phi(s)$
      $p(y|s,\theta) = \mathrm{Softmax}(\boldsymbol{W}\boldsymbol{s} + \boldsymbol{b})$
    end for
    Update $\boldsymbol{W}, \boldsymbol{b}$ and Enc_ϕ w.r.t. $l(\theta) = \sum_{s \in S_{batch}} \log p(y|s,\theta)$
  end for
end while

Similarly, Algorithm 2 illustrates the training algorithm for few-shot relation learning.

Algorithm 2 Training for Few-shot Relation Extraction

Input: Few-shot dataset S_f, statement encoder Enc_ϕ
repeat
  $\mathcal{R}' = \mathrm{SampleRelation}(N, \mathcal{R})$.
  $S = \varnothing, Q = \varnothing$
  for r **in** $\mathcal{R}'$ **do**
    $\{s_i^r\} = \mathrm{SampleInstance}(k, r), i \in \{1, 2...k\}$
    $q^r = \mathrm{SampleInstanc}(1, r)$
    $S \bigcup \{s_i^r\}, Q \bigcup \{q^r\}$
  end for
  for q_r **in** Q **do**
    for $s_i^{r'}$ **in** S **do**
      $\mathrm{sim}_i^{r'} = \mathrm{Enc}_\phi(q_r) \cdot \mathrm{Enc}_\phi(s_i^{r'})$
    end for
    $\hat{r} = \mathrm{argmax}_{r', r' \in \mathcal{R}'}(\mathrm{sim}_i^{r'})$
    Update Enc_ϕ w.r.t. $\mathrm{CrossEntropy}(r, \hat{r})$
  end for
until enough steps

Algorithm 3 reports the training of fuzzy relation learning. Note that FuzzyRED is designed for evaluation, so the algorithm shows the inference phase.

Algorithm 3 Training for Fuzzy Relation Evaluation

Input: FuzzyRED $S_z = \{s_{1:n}\}$, statement encoder Enc_ϕ
while not converge **do**
  for s **in** S_z **do**
    $\{s_i\} = \mathrm{SampleInstance}(k), i \in \{1, 2, ..., k\}$
    Compute a w.r.t $a = \frac{1}{k} \sum p(l = 1|s, s_i)$
    if $a > T$ **then**
      s is true positive (TP)
    else
      s is false positive (FP)
    end if
  end for
end while