Hybrid Attention-Based Prototypical Networks for Noisy Few-Shot Relation Classification

Tianyu Gao*, Xu Han*, Zhiyuan Liu†, Maosong Sun
Department of Computer Science and Technology, Tsinghua University, Beijing, China
Institute for Artificial Intelligence, Tsinghua University, Beijing, China
State Key Lab on Intelligent Technology and Systems, Tsinghua University, Beijing, China
{gty16,hanxu17}@mails.tsinghua.edu.cn,
{liuzy,sms}@tsinghua.edu.cn

Abstract

The existing methods for relation classification (RC) primarily rely on distant supervision (DS) because large-scale supervised training datasets are not readily available. Although DS automatically annotates adequate amounts of data for model training, the coverage of this data is still quite limited, and meanwhile many long-tail relations still suffer from data sparsity. Intuitively, people can grasp new knowledge by learning few instances. We thus provide a different view on RC by formalizing RC as a few-shot learning (FSL) problem. However, the current FSL models mainly focus on low-noise vision tasks, which makes them hard to directly deal with the diversity and noise of text. In this paper, we propose hybrid attention-based prototypical networks for the problem of noisy few-shot RC. We design instance-level and feature-level attention schemes based on prototypical networks to highlight the crucial instances and features respectively, which significantly enhances the performance and robustness of RC models in a noisy FSL scenario. Besides, our attention schemes accelerate the convergence speed of RC models. Experimental results demonstrate that our hybrid attention-based models require fewer training iterations and outperform the state-of-the-art baseline models. The code and datasets are released on https://github.com/thunlp/HATT-Proto.

Introduction

Relation classification (RC) is an import task in information extraction, aiming to classify the relation between two given entities based on their related context. Due to the capability of extracting textual information and benefiting many NLP applications (e.g., information retrieval, dialog generation, and question answering), RC appeals to many researchers. Conventional supervised models have been widely explored in this task (Zelenko, Aone, and Richardella 2003; Zeng et al. 2014; Gormley, Yu, and Dredze 2015), however, their performance heavily depends on the scale and quality of training data. In practice, manual labeling of high-quality data is time-consuming and human-intensive, which means these supervised models usually suffer from scarce data and are thus hard to generalize well.

To construct large-scale data, Mintz et al. (2009) propose a novel distant supervision (DS) mechanism to automatically label training instances by aligning existing knowledge graphs (KGs) with text. DS is a heuristic rule: for an entity pair in KGs, those sentences mentioning both the entities will be labeled with their relations in KGs. DS enables RC models to work on large-scale training corpora and thus becomes a primary approach for RC recently. Because DS brings inevitable noise in itself, many efforts are devoted to further reducing noise (Wu, Bamman, and Russell 2017; Feng et al. 2018). Although these DS models achieve promising results on common relations, their classification performance still drops dramatically when there are only few training instances for some relations. Empirically, adopting DS can automatically annotate adequate amounts of training data. However, this data usually just covers a limited part of relations. Many relations are long-tail and still suffer from data deficiency. The current DS models ignore the problem of long-tail relations, which makes these models hard to extract comprehensive information from plain text.

It is intuitive that people can learn new knowledge after being taught just few instances. Hence, we provide a different view to formulate RC in a few-shot learning (FSL) scenario, which requires models to handle classification tasks with a handful of training instances. Note that there are also some works attempting to handle RC in a zero-shot scenario (Levy et al. 2017), incorporating extra information to classify relations never appearing in training sets. Although zero-shot RC is meaningful and has a strong academic exploring value, it is a little far away from real-world scenes. In fact, even for people, it is hard to grasp new knowledge without any examples but limited extra information.

Some efforts have also been devoted to FSL. The early works (Caruana 1995; Bengio 2012) mainly focus on applying transfer learning methods to fine-tune pre-trained models, which adopt latent information from the common classes containing adequate instances. Then, metric learning methods (Koch, Zemel, and Salakhutdinov 2015) have been proposed to learn the distance distributions among classes. Recently, meta learning is proposed, which encourages models to learn fast-learning abilities from previous experience and rapidly generalize to new concepts (Ravi and Larochelle 2017; Munkhdalai and Yu 2017). Among these models, prototypical networks (Snell, Swersky, and Zemel

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

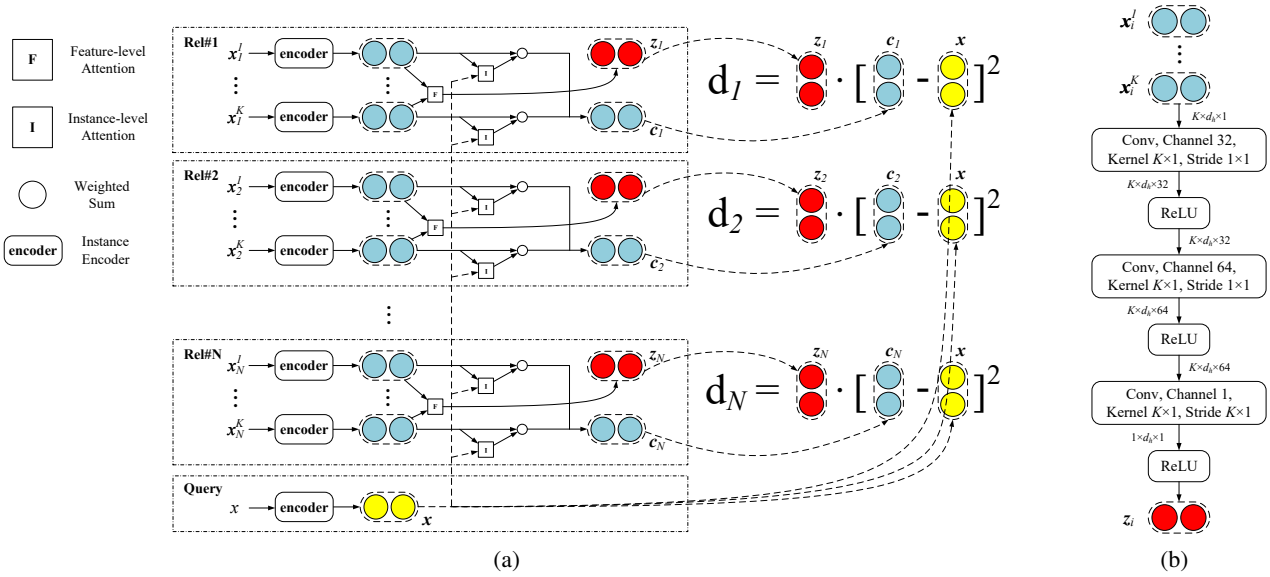


Figure 1: Architectures of our proposed models. Figure 1(a) shows the framework of hybrid attention-based Prototypical Networks. Figure 1(b) demonstrates the architectures of feature-level attention.

2017) achieve the state-of-the-art results on several FSL benchmarks, meanwhile are simple and effective. Though FSL methods develop fast, most of these works concentrate on image classification. There lacks systematic researches about adopting FSL for NLP tasks. Different from images, text is more diverse and noisy, which means these current FSL models are hard to directly generalize to NLP applications, including the task of RC with noisy data.

To address these issues, we propose hybrid attention-based prototypical networks for noisy few-shot RC. Similar to the vanilla prototypical networks, our methods also adopt neural networks to embed all instances in a support set and compute a feature vector (prototype) for each relation via these instance embeddings. Then, we classify the relation between the entity pair mentioned in a query instance by measuring the distance between the query instance embedding and relation prototypes. For noisy few-shot RC, both data and features are sparse. Little noise in the support set may cause a huge deviation of relation features, and not all dimensions of relation features in the space are discriminative enough to support final classification. Our hybrid attentions are specially designed to alleviate the influence of noisy data and sparse features.

As illustrated in Figure 1(a), our models employ a hybrid attention consisting of an instance-level attention and a feature-level attention. The instance-level attention module is able to select more informative instances in the support set and denoise those noisy instances during training. The feature-level attention module can highlight important dimensions in the feature space and formulate specific distance functions for different relations, which enables our model to alleviate the problem of feature sparsity. With the hybrid attentions making the model focus more on those important instances and features, FSL models not only become more effective and robust, but also have fewer training constraints and converge more rapidly.

We conduct experiments on a real-world dataset whose KGs are extracted from Wikidata and text is derived from Wikipedia. Experimental results demonstrate that our hybrid attention-based prototypical networks significantly outperform other baseline methods. By adding different levels of data noise, we validate the robustness of our model, and prove our model is more suitable for handling the diversity and noise of text as compared to the current FSL models. Additionally, experiments also show that our hybrid attentions accelerate the convergence speed during training.

Related Works

RC is an important task in NLP and many models have been proposed for it. Conventional algorithms like kernel methods (Zelenko, Aone, and Richardella 2003; Zhou et al. 2005) and embedding methods (Gormley, Yu, and Dredze 2015) are used. However, supervised learning requires large amounts of data which are hard to acquire. To alleviate this problem, DS mechanism (Mintz et al. 2009) has been proposed for RC, which generates data automatically by aligning KGs with text. Though DS makes large-scale training data possible, it also brings the wrong labeling problem. To solve this, Riedel, Yao, and McCallum (2010) treat DS as a multi-instance single-label task. Then, multi-instance multi-label setting is also proposed and has become a common practice in this field (Hoffmann et al. 2011; Surdeanu et al. 2012).

Neural networks have shown great power in supervised tasks and been widely used in several NLP tasks. Many works have explored the approaches to use neural networks in RC. Zeng et al. (2014) and Santos, Xiang, and Zhou (2015) use convolutional neural networks for RC.

Those models are all trained on the sentence level and suffer from insufficient data. Zeng et al. (2015) utilize DS and neural networks to perform at-least-one multi-instance learning, choosing only one instance of each entity pair for classification. Then, Lin et al. (2016) enhance it by adopting selective attention over instances, which benefits from all instances and meanwhile restrains the influence of wrong-labeled ones. Many attention-based RC models have been proposed to reduce noise caused by DS (Verga et al. 2016; Verga and McCallum 2016; Liu et al. 2017; Huang and Wang 2017) and introduce more extra information (Ji et al. 2017; Han, Liu, and Sun 2018), including some sophisticated mechanisms, such as reinforcement learning (Feng et al. 2018; Zeng et al. 2018) and adversarial training (Wu, Bamman, and Russell 2017; Wang et al. 2018). These works mainly adopt DS to make large-scale datasets and reduce the noise caused by DS, regardless of the effect of long-tail relations.

FSL also enables models to learn high-quality features with insufficient data, without adding large-scale data like DS. Many works apply transfer learning methods to fine-tune pre-trained models for FSL, which transfer latent information from the common classes containing adequate instances to the uncommon classes with only few instances (Caruana 1995; Bengio 2012; Donahue et al. 2014). Then metric learning methods (Koch, Zemel, and Salakhutdinov 2015; Vinyals et al. 2016) have also been proposed to learn the distance distributions among classes, and similar classes are adjacent in the distance space. Recently, the idea of meta-learning is proposed, which encourages models to learn fast-learning abilities from previous experience and rapidly generalize to new concepts (Ravi and Larochelle 2017; Santoro et al. 2016; Finn, Abbeel, and Levine 2017; Munkhdalai and Yu 2017). Among these models, prototypical networks (Snell, Swersky, and Zemel 2017) is simple to implement, fast to train and it achieves the state-of-the-art results on several FSL tasks. It calculates the prototype for each class and classifies query instances by calculating their Euclidean distances. Our proposed method is based on prototypical networks. Though few-shot methods develop fast in the recent years, most of these works concentrate on CV applications. Both of the popular FSL datasets Omniglot (Lake, Salakhutdinov, and Tenenbaum 2015) and mini-ImageNet (Vinyals et al. 2016) are designed for CV applications. Yu et al. (2018) attempt to adopt FSL for text classification and achieve promising results. However, there are still few systematic researches about adopting FSL for NLP tasks.

In this paper, we provide a different view to formulate RC in a noisy FSL scenario. Due to the small amounts of samples, FSL are more easily to be affected by data noise, especially considering that human annotators are more likely to make mistakes in language tasks than visual tasks. It is necessary to consider the diversity and complicacy of semantic information when designing FSL models for NLP applications, especially the task of RC. Thus FSL models with not only high performance but also resistance to noisy data is necessary, which are exactly the qualities that our hybrid attention-based methods have.

Methodology

In this section, we introduce the overall framework of our hybrid attention-based prototypical networks, starting with notations and definitions.

Notations and Definitions

Few-shot RC is defined as a task to predict the relation r between the entity pair (h, t) mentioned in a query instance x, given a relation set $\mathcal{R}$ and a support set $\mathcal{S}$. $\mathcal{S}$ is defined as follows,

$$\begin{aligned}\mathcal{S} = \{&(x_1^1, h_1^1, t_1^1, r_1), \ldots, (x_1^{n_1}, h_1^{n_1}, t_1^{n_1}, r_1), \\ &\ldots \\ &(x_m^1, h_m^1, t_m^1, r_m), \ldots, (x_m^{n_m}, h_m^{n_m}, t_m^{n_m}, r_m)\}, \\ &r_1, r_2, \ldots, r_m \in \mathcal{R},\end{aligned} \tag{1}$$

where $(x_i^j, h_i^j, t_i^j, r_i)$ means that the semantics of the instance x_i^j indicates there is a relation r_i between the entity pair (h_i^j, t_i^j). The entities h_i^j and t_i^j are all mentioned in the instance x_i^j. Each instance x_i^j is denoted as a word sequence $\{w_1, w_2, \ldots\}$.

In a FSL scenario, the instance number n_i of the relation r_i is usually quite small. Few-shot RC models have to learn features from the few instances in the support set $\mathcal{S}$ and predict the relation r for any given query instance x. Following the recent FSL setting, we adopt N way K shot for few-shot RC as follows,

$$N = m = |\mathcal{R}|, K = n_1 = \ldots = n_m. \tag{2}$$

Framework

We introduce the overall framework of our proposed hybrid attention-based prototypical networks in detail. As shown in Figure 1(a), our model consists of three parts:

Instance Encoder. Given an instance and its mentioned entity pair, we employ neural networks to encode the instance semantics into an embedding. In this paper, we implement the instance encoder with convolutional neural networks (CNN) in view of both model performance and time efficiency. In fact, other neural architectures such as recurrent neural networks (RNN) can also be used as sentence encoders.

Prototypical Networks. After computing instance embeddings, we adopt prototypical networks to compute a prototype for each relation via instance embeddings in the support set. By comparing the distance between a query instance embedding and each relation prototype, we can finally classify the relation between the entity pair mentioned in the query instance.

Hybrid Attention. Based on prototypical networks, we further propose a hybrid attention mechanism to enhance the classification performance and convergence speed. Our hybrid attention mechanism includes two parts, the instance-level attention module to help determine more query-related instances to compute a better prototype for each relation, and the feature-level attention module to alleviate the problem of feature sparsity and measure the space distance in a more suitable way. Both parts cooperate with each other during training.

Instance Encoder

Given an instance $x = \{w_1, \ldots, w_n\}$ mentioning two entities, we apply convolutional neural networks to encode the raw instance into a continuous low-dimensional embedding $\mathbf{x}$, aiming to capture the instance semantics. The instance encoder consists of an embedding layer and an encoding layer.

Embedding Layer The embedding layer is used to map discrete words in the instance into continuous input embeddings. Given an instance x, we map each word w_i in the instance to a real-valued embedding $\mathbf{w}_i \in \mathbb{R}^{d_w}$ to express semantic and syntactic meanings of the word. These embeddings are pre-trained via GloVe (Pennington, Socher, and Manning 2014).

Because words closer to the entities show more impact on the determination of relation, we adopt position embeddings following Zeng et al. (2014). For each word w_i, we embed its relative distances to the two entities into two d_p-dimensional vectors, and then concatenate them as a unified position embedding $\mathbf{p}_i \in \mathbb{R}^{2\times d_p}$.

We could achieve a final input embedding for each word by concatenating its word embedding and position embedding. By gathering all the input embeddings in the instance, we have an embedding sequence ready for the encoding layer as follows,

$$\begin{aligned}\{\mathbf{e}_1 \ldots, \mathbf{e}_n\} = \{[\mathbf{w}_1; \mathbf{p}_1], \ldots, [\mathbf{w}_n; \mathbf{p}_n]\},\\ \mathbf{e}_i \in \mathbb{R}^{d_i}, d_i = d_w + d_p \times 2.\end{aligned} \tag{3}$$

Encoding Layer In the encoding layer, we select CNN to encode the input embeddings $\{\mathbf{e}_1 \ldots, \mathbf{e}_n\}$ into the final instance embedding $\mathbf{x}$. CNN slides a convolution kernel with the window size m over the input embeddings to get the d_h-dimensional hidden embeddings,

$$\mathbf{h}_i = \text{CNN}\big(\mathbf{e}_{i-\frac{m-1}{2}}, \ldots, \mathbf{e}_{i+\frac{m-1}{2}}\big), \tag{4}$$

where $\text{CNN}(\cdot)$ is a convolution operation (Zeng et al. 2014).

A pooling operation is then applied over these hidden embeddings to output the final instance embedding $\mathbf{x}$ as follows,

$$[\mathbf{x}]_j = \max\Big\{[\mathbf{h}_1]_j, \ldots, [\mathbf{h}_n]_j\Big\}, \tag{5}$$

where $[\cdot]_j$ is the j-th value of a vector.

For simplicity, we denote such an instance encoding operation, including both embedding and encoding layers, as the following equation,

$$\mathbf{x} = f_\phi(x), \tag{6}$$

where ϕ is the learnable parameters of the instance encoder.

Prototypical Networks

The main idea of prototypical networks is to use one vector, also named prototype, to represent each relation. The vanilla approach to compute the prototype is to average all the instance embeddings in the support set for each relation,

$$\mathbf{c}_i = \frac{1}{n_i}\sum_{j=1}^{n_i}\mathbf{x}_i^j, \tag{7}$$

where $\mathbf{c}_i$ is the prototype computed for the relation r_i and $\mathbf{x}_i^j$ is an instance embedding of the relation r_i in the support set $\mathcal{S}$. There are n_i instances of the relation r_i in the support set $\mathcal{S}$.

In vanilla prototypical networks, all instances are taken into consideration equally. However, in our hybrid attention-based prototypical networks, the simple average mechanism is replaced with our instance-level attention to highlight those more related instances in the support set, which we will discuss in the next part.

We can then compute the probabilities of the relations in $\mathcal{R}$ for the query instance x as follows,

$$p_\phi(y = r_i|x) = \frac{\exp\big(-d(f_\phi(x), \mathbf{c}_i)\big)}{\sum_{j=1}^{|\mathcal{R}|}\exp\big(-d(f_\phi(x), \mathbf{c}_j)\big)}, \tag{8}$$

where $d(\cdot,\cdot)$ is the distance function for two given vectors.

There are multiple choices for the distance function. According to Snell, Swersky, and Zemel (2017), Euclidean distance outperforms other distance functions. Hence, we adopt Euclidean distance with feature-level attention in our networks, which can achieve better results than vanilla Euclidean distance. We will also detailedly present our feature-level attention in the next part.

Hybrid Attention

Our hybrid attention consists of two modules, the instance-level attention module to select more informative instances in the support set, and the feature-level attention module to highlight important dimensions in the distance function.

Instance-level Attention For each relation, the original prototypical networks adopt the average vector of the instances as the relation prototype. Due to the lack of support data in a FSL scenario, one instance with a representation far from other instances will cause a huge deviation of the corresponding prototype. This often happens when data is noisy or relations cover diverse semantics. Meanwhile, the vanilla FSL model has never seen concrete query instances before extracting features from support sets. Hence, the vanilla model may extract some features that are not helpful for query classification.

These phenomenons bring the problem of unsuitable prototypes for final query classification. To enhance the ability of prototypical networks, we propose an instance-level attention module to focus more attention on those query-related instances and reduce the effect of noise. We argue that not all instances are equal when given a query, each instance representation is given a weight α_j, and Eq (7) is replaced by the following formula,

$$\mathbf{c}_i = \sum_{j=1}^{n_i}\alpha_j\mathbf{x}_i^j. \tag{9}$$

α_j is defined as follows,

$$\begin{aligned}\alpha_j &= \frac{\exp(e_j)}{\sum_{k=1}^{n_i}\exp(e_k)},\\ e_j &= \text{sum}\Big\{\sigma\big(g(\mathbf{x}_i^j)\odot g(\mathbf{x})\big)\Big\},\end{aligned} \tag{10}$$

where $g(\cdot)$ is a linear layer, $\odot$ is element-wise production, $\sigma(\cdot)$ is an activation function and sum$\{\cdot\}$ means the sum of all elements of the vector. In this paper, we choose $\tanh$ for $\sigma(\cdot)$ to produce results among $[-1, 1]$.

Via instance-level attention, instances with features more similar to queries gain higher weights, and final prototypes are closer to those instances. Intuitively, the instances of a given relation may be quite different, or even some of them have been wrongly labeled. Query instances may be close to only some of the instances of the relation. By giving different weights for different instances, prototypes can be more "typical" when compared to the original average vectors.

Feature-level Attention Snell, Swersky, and Zemel (2017) demonstrate that selection of distance functions will significantly affect the capacity of prototypical networks. The original model uses simple Euclidean distance as the distance function. Because there are only few instances in the support set, the features extracted from the support set suffer from the problem of data sparsity. Hence, some dimensions are more discriminative for classifying special relations in the feature space. We propose a feature-level attention to alleviate the problem of feature sparsity and measure the space distance in a more suitable way.

The feature-level attention will pay more attention to those more discriminative feature dimensions when computing space distance. Especially, we adopt a new distance function instead of plain Euclidean distance,

$$d(\mathbf{s}_1, \mathbf{s}_2) = \mathbf{z}_i \cdot (\mathbf{s}_1 - \mathbf{s}_2)^2 \tag{11}$$

where $\mathbf{z}_i$ is the score vector for relation r_i computed via our feature-level attention extractor. The structure of the feature-level attention extractor is shown in Figure 1(b).

The extractor calculates how linear separable each dimension of the feature is, based on the distribution of the sentence representations of each relation. The more useful one feature dimension is, the higher corresponding score it will get. By multiplying the attention scores to the squared differences, we change the distance metrics to better fit the given relations and support instances.

Experiments

In experiments, we will show that our hybrid attention-based prototypical networks achieve better results on few-shot RC tasks with both noisy data and clean data. We further demonstrate that hybrid attention brings convergence speeding up, and detailedly show how instance-level attention and feature-level attention work in the feature space respectively.

Dataset and Evaluation Metrics

We evaluate our models on FewRel, a few-shot RC dataset (Han et al. 2018) [1]. It has 64 relations for training, 16 relations for validation and 20 relations for test. There are no overlapping relations between training and test set. Each relation has 700 instances in FewRel.

[1] https://github.com/thunlp/FewRel

To demonstrate the robustness of our hybrid attention in noisy data, we adopt four levels of random noisy settings: no noisy data, 10% noisy data, 30% noisy data and 50% noisy data. We assume that all the data in FewRel dataset is correct, and during training and test, each instance in the support set has a probability of $rate$ to be corrupted with a sentence whose relation is not the same as the original one, where $rate = 0\%, 10\%, 30\%, 50\%$.

Parameter Settings

All the hyperparameters are shown in table 1. For CNN parameters, we follow the settings used in Zeng et al. (2014). According to Snell, Swersky, and Zemel (2017), feeding more classes than test settings during training leads to better results, we thus randomly choose 20 classes for each batch when training.

We tune all other hyperparameters of all models by grid search using the validation set, especially for determining the best initial learning rate and weight decay. For prototypical networks and our proposed model, we use step policy to decay the learning rate. That is to say, the learning rate will be multiplied by γ every s steps. Due to the different convergence speed, we adopt different s and total training steps for different models. To be more specifically, we train prototypical networks 30000 iterations with $s = 20000$, while training hybrid attention-based model 15000 iterations with $s = 5000$. All models are trained on the training set, then the best epochs on the validation set are picked to be tested on the test set.

We use the word embeddings pre-trained by GloVe (Pennington, Socher, and Manning 2014) as our initial embeddings. In practice, we choose the embedding set (Wikipedia 2014 + Gigaword 5) which contains $6B$ tokens, $400K$ vocabulary and word embeddings are of 50 dimensions.

Convolutional Window Size m	3
Word Embedding Dimension d_w	50
Position Embedding Dimension d_p	5
Hidden Layer Dimension d_h	230
Batch Size	4
Training Classes for One Batch	20
Initial Learning Rate	0.1
Weight Decay	10^{-5}
Learning Rate Decay γ	0.1

Table 1: Parameter settings.

Overall Evaluation Results

Table 2 reports the accuracy of prototypical networks with and without hybrid attentions on the test set under different experiment settings. We name the vanilla prototypical networks "proto". "proto-IATT","proto-FATT",and "proto-HATT" are models with instance-level, feature-level and hybrid attentions respectively. From the table, we can find that our hybrid attention-based prototypical networks are more robust when facing noisy data. As the noise rate rising, the advantages of our proposed models become more obvious. By using the hybrid attentions and giving different scores

Noise Rate	Model	5 Way 5 Shot	5 Way 10 Shot	10 Way 5 Shot	10 Way 10 Shot
0%	Proto	89.05 ± 0.09	90.79 ± 0.08	81.46 ± 0.13	84.01 ± 0.13
	Proto-HATT	$\mathbf{90.12 \pm 0.04}$	$\mathbf{92.06 \pm 0.06}$	$\mathbf{83.05 \pm 0.05}$	$\mathbf{85.97 \pm 0.08}$
10%	Proto	87.63 ± 0.10	90.15 ± 0.08	79.39 ± 0.14	83.05 ± 0.12
	Proto-HATT	$\mathbf{88.74 \pm 0.06}$	$\mathbf{91.45 \pm 0.05}$	$\mathbf{81.09 \pm 0.08}$	$\mathbf{85.08 \pm 0.07}$
30%	Proto	82.45 ± 0.09	87.64 ± 0.07	72.43 ± 0.12	79.31 ± 0.11
	Proto-HATT	$\mathbf{84.71 \pm 0.07}$	$\mathbf{89.59 \pm 0.05}$	$\mathbf{75.68 \pm 0.11}$	$\mathbf{82.43 \pm 0.07}$
50%	Proto	72.91 ± 0.15	81.71 ± 0.10	61.11 ± 0.17	71.29 ± 0.14
	Proto-HATT	$\mathbf{76.57 \pm 0.07}$	$\mathbf{85.17 \pm 0.09}$	$\mathbf{65.97 \pm 0.11}$	$\mathbf{76.42 \pm 0.13}$

Table 2: Accuracy comparison between prototypical networks with or without the hybrid attentions (%). *Noise rate* indicates the probability of an instance in the support set to be wrong-labeled. As shown in the table, our attention methods largely improve the performance under both clean and noisy data.

Model	5 Way 5 Shot	10 Way 5 Shot
Finetune*	68.66 ± 0.41	55.04 ± 0.31
kNN*	68.77 ± 0.41	55.87 ± 0.31
Meta Network*	80.57 ± 0.48	69.23 ± 0.52
GNN*	81.28 ± 0.62	64.02 ± 0.77
SNAIL*	79.40 ± 0.22	68.33 ± 0.25
Proto*	84.79 ± 0.16	75.55 ± 0.19
Proto	89.05 ± 0.09	81.46 ± 0.13
Proto-IATT	89.63 ± 0.08	82.16 ± 0.13
Proto-FATT	89.70 ± 0.03	82.45 ± 0.05
Proto-HATT	$\mathbf{90.12 \pm 0.04}$	$\mathbf{83.05 \pm 0.05}$

Table 3: Accuracy comparison between different models (%). Results with * are reported in Han et al. (2018).

to instances and features, our models know which instances and which parts of features to focus on when training, and meanwhile capture the correct paths for backpropagation. This helps models to resist the adverse effects of data noise. Our models even outperform the baselines a lot on clean data, which proves that hybrid attentions are also useful in few-shot tasks with clean data. We also compare our methods with other FSL and RC models. For RC models, we conduct comprehensive evaluations of RC models with simple few-shot strategies such as finetune or kNN. For FSL models, we compare with Meta Network (Munkhdalai and Yu 2017), GNN (Garcia and Bruna 2018), and SNAIL (Mishra et al. 2018), which are all current state-of-the-art FSL models. The evaluation results are shown in Table 3. As shown in the table, our two attention modules both make contributions to improve the performance, and our proposed hybrid attention-based methods achieve the best results.

Convergence Speed

We compare our hybrid attention-based models with the original one on model convergence speed. Hybrid attention-based prototypical models converge faster than the original one and this phenomenon is more obvious with data noise.

Figure 2: Loss of different models on the training set.

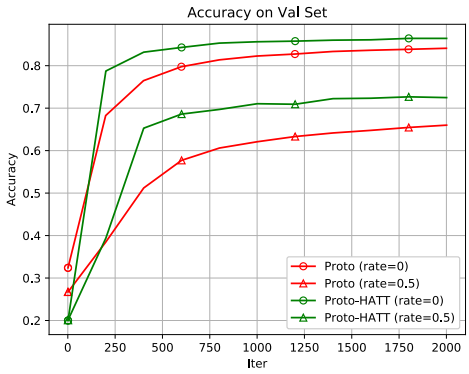
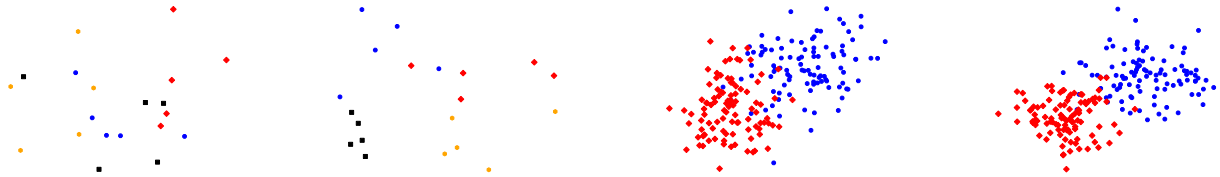


Figure 3: Accuracy of different models on the validation set.

As shown in Figure 2 and Figure 3, the speed of both loss decrease and accuracy increase, becomes lower when adding data noise. Vanilla prototypical networks must spend almost twice the time of the hybrid attention models to reduce the loss by 40%, while under 50% noise rate, this gap is even bigger. In the figure, convergence speed decreases when training with noisy data, and this problem is alleviated by using hybrid attention.



(a) Features with lower scores. (b) Features with higher scores. (c) Emb trained without HATT. (d) Emb trained with HATT.

Figure 4: Comparison between features with different feature-level attention scores (Figure 4(a) and Figure 4(b)). Points with different colors (indicating different classes) in Figure 4(b) are more separable than in Figure 4(a); Comparison between instance embeddings trained with or without hybrid attentions (Figure 4(c) and Figure 4(d)). Points in Figure 4(d) are easier to classify while those in Figure 4(c) just lump together.

Support Set	
Relation	Sentences
(A) facet of	(1) In 2001, he also published the "Khaki Shadows" that recounted the ***military history*** of ***Pakistan*** during the cold war. (2) However, critics have questioned the universal applicability of this model outside ***Singapore***'s communitarian political system and coordinated ***urban planning program***.
(B) series	(1) **(Got the highest attention score)** "***Crying Out Loud***" is the twenty-third episode of the sixth season of the American sitcom "***Modern Family***", and the series' 143rd episode overall. (2) The novel is the fourth in Moorcock's four book ***The History of the Runestaff*** series, and the narrative follows on immediately from the preceding novel "***the Sword of the Dawn***".
Query Sentence	
(A) or (B)	The song appeared on the first episode of the ***fourth season*** of the American adult animated sitcom "***American Dad!***".

Table 4: Case study of instance-level attention. Words with color blue are head entities and those with red are tail entities. Sentence (1) of relation (B) got the highest instance-level attention score for its close connection with the query instance.

Effect of Instance-Level Attention

By studying the cases that prototypical networks fail but our model predicts correctly, we show that our instance-level attention is able to locate the instances that have most similarity to the query ones. As shown in Table 4, models need to predict whether the query instance is an instance of the relation "facet of" or "series". It is quite challenging since those two relations both express the meaning of subordination. The query one is an instance of "series", but prototypical networks predict it wrongly into "facet of". By using instance-level attention, sentence (1) of the relation "series" was given the highest attention score, for it is semantically and syntactically closer to the query instance, and thus our model is able to classify the query instance into the correct relation.

Effect of Feature-Level Attention

To show the effect of feature-level attention, we sort all 230 features of hidden embeddings by their feature-level attention scores and select the highest 20 features and the lowest 20 ones, and map them to 2D points by using Principal Component Analysis (PCA)[2]. Comparing those two plots in Figure 4(a) and Figure 4(b), it is quite obvious that those features with higher attention scores are easier to be classified.

Comparing Encoder Capacity

We find out that our hybrid attention mechanism not only works out fine by focusing on relative instances and capturing effective features, but it also helps encoders to learn better embeddings due to it gives different weights to instances and features during backpropagation. We draw the results of the encoders with and without hybrid attention by PCA. In Figure 4(c) and Figure 4(d), it is easy to find that embeddings trained with attention are better since those points are more linear separable. Actually, if we train the model with hybrid attention and test without attention, it can still achieve better results than the baselines.

Conclusion and Future Work

In this paper, we propose hybrid attention-based prototypical networks for noisy few-shot relation classification task. Our hybrid attentions consist of two modules, an instance-level attention which highlights those query-related instances, and a feature-level attention which alleviates the problem of feature sparsity. In our experiments, we evaluate our models with several random noise settings and few-shot settings, which demonstrate that our hybrid attentions significantly improve the robustness and efficiency of the FSL models. Our models not only achieve the state-of-the-art results and perform better in noisy data, but also converge a lot faster when training. In the future, we will explore to incorporate our hybrid attention schemes with other FSL models and adopt more neural encoders to make our model more general.

[2]There are different feature-level attention scores for different relations. For plotting points of different relations in the one figure, we simply average the scores over relations here.

Acknowledgement

This work is supported by the National Key Research and Development Program of China (No. 2018YFB1004503) and the National Natural Science Foundation of China (NSFC No. 61572273, 61661146007). This work is also part of the NExT++ project, supported by the National Research Foundation, Prime Ministers Office, Singapore under its IRC@Singapore Funding Initiative.

References

Bengio, Y. 2012. Deep learning of representations for unsupervised and transfer learning. In *Proceedings of ICML*, 17–36.

Caruana, R. 1995. Learning many related tasks at the same time with backpropagation. In *Proceedings of NIPS*, 657–664.

Donahue, J.; Jia, Y.; Vinyals, O.; Hoffman, J.; Zhang, N.; Tzeng, E.; and Darrell, T. 2014. Decaf: A deep convolutional activation feature for generic visual recognition. In *Proceedings of ICML*, 647–655.

Feng, J.; Huang, M.; Zhao, L.; Yang, Y.; and Zhu, X. 2018. Reinforcement learning for relation classification from noisy data. In *Proceedings of AAAI*, 5779–5786.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of ICML*, 1126–1135.

Garcia, V., and Bruna, J. 2018. Few-shot learning with graph neural networks. In *Proceedings of ICLR*.

Gormley, M. R.; Yu, M.; and Dredze, M. 2015. Improved relation extraction with feature-rich compositional embedding models. In *Proceedings of EMNLP*, 1774–1784.

Han, X.; Zhu, H.; Yu, P.; Wang, Z.; Yao, Y.; Liu, Z.; and Sun, M. 2018. Fewrel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of EMNLP*, 4803–4809.

Han, X.; Liu, Z.; and Sun, M. 2018. Neural knowledge acquisition via mutual attention between knowledge graph and text. In *Proceedings of AAAI*, 4832–4839.

Hoffmann, R.; Zhang, C.; Ling, X.; Zettlemoyer, L.; and Weld, D. S. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of ACL*, 541–550.

Huang, Y. Y., and Wang, W. Y. 2017. Deep residual learning for weakly-supervised relation extraction. In *Proceedings of EMNLP*, 1803–1807.

Ji, G.; Liu, K.; He, S.; Zhao, J.; et al. 2017. Distant supervision for relation extraction with sentence-level attention and entity descriptions. In *Proceedings of AAAI*, 3060–3066.

Koch, G.; Zemel, R.; and Salakhutdinov, R. 2015. Siamese neural networks for one-shot image recognition. In *Proceedings of ICML Workshop*.

Lake, B. M.; Salakhutdinov, R.; and Tenenbaum, J. B. 2015. Human-level concept learning through probabilistic program induction. *Proceedings of Science* 350(6266):1332–1338.

Levy, O.; Seo, M.; Choi, E.; and Zettlemoyer, L. 2017. Zero-shot relation extraction via reading comprehension. In *Proceedings of CoNLL*, 333–342.

Lin, Y.; Shen, S.; Liu, Z.; Luan, H.; and Sun, M. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of ACL*, 2124–2133.

Liu, T.; Wang, K.; Chang, B.; and Sui, Z. 2017. A soft-label method for noise-tolerant distantly supervised relation extraction. In *Proceedings of EMNLP*, 1790–1795.

Mintz, M.; Bills, S.; Snow, R.; and Jurafsky, D. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of ACL-IJCNLP*, 1003–1011.

Mishra, N.; Rohaninejad, M.; Chen, X.; and Abbeel, P. 2018. A simple neural attentive meta-learner. In *Proceedings of ICLR*.

Munkhdalai, T., and Yu, H. 2017. Meta networks. In *Proceedings of ICML*, 2554–2563.

Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global vectors for word representation. In *Proceedings of EMNLP*, 1532–1543.

Ravi, S., and Larochelle, H. 2017. Optimization as a model for few-shot learning. In *Proceedings of ICLR*.

Riedel, S.; Yao, L.; and McCallum, A. 2010. Modeling relations and their mentions without labeled text. In *Proceedings of ECML-PKDD*, 148–163.

Santoro, A.; Bartunov, S.; Botvinick, M.; Wierstra, D.; and Lillicrap, T. 2016. Meta-learning with memory-augmented neural networks. In *Proceedings of ICML*, 1842–1850.

Santos, C. N. d.; Xiang, B.; and Zhou, B. 2015. Classifying relations by ranking with convolutional neural networks. In *Proceedings of ACL-IJCNLP*, 626–634.

Snell, J.; Swersky, K.; and Zemel, R. 2017. Prototypical networks for few-shot learning. In *Proceedings of NIPS*, 4077–4087.

Surdeanu, M.; Tibshirani, J.; Nallapati, R.; and Manning, C. D. 2012. Multi-instance multi-label learning for relation extraction. In *Proceedings of EMNLP*, 455–465.

Verga, P., and McCallum, A. 2016. Row-less universal schema. In *Proceedings of ACL*, 63–68.

Verga, P.; Belanger, D.; Strubell, E.; Roth, B.; and McCallum, A. 2016. Multilingual relation extraction using compositional universal schema. In *Proceedings of NAACL*, 886–896.

Vinyals, O.; Blundell, C.; Lillicrap, T.; Wierstra, D.; et al. 2016. Matching networks for one shot learning. In *Proceedings of NIPS*, 3630–3638.

Wang, X.; Han, X.; Lin, Y.; Liu, Z.; and Sun, M. 2018. Adversarial multi-lingual neural relation extraction. In *Proceedings of COLING*, 1156–1166.

Wu, Y.; Bamman, D.; and Russell, S. 2017. Adversarial training for relation extraction. In *Proceedings of EMNLP*, 1778–1783.

Yu, M.; Guo, X.; Yi, J.; Chang, S.; Potdar, S.; Cheng, Y.; Tesauro, G.; Wang, H.; and Zhou, B. 2018. Diverse few-shot text classification with multiple metrics. In *Proceedings of NAACL*, 1206–1215.

Zelenko, D.; Aone, C.; and Richardella, A. 2003. Kernel methods for relation extraction. *Proceedings of JMLR* 3(Feb):1083–1106.

Zeng, D.; Liu, K.; Lai, S.; Zhou, G.; and Zhao, J. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING*, 2335–2344.

Zeng, D.; Liu, K.; Chen, Y.; and Zhao, J. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of EMNLP*, 1753–1762.

Zeng, X.; He, S.; Liu, K.; and Zhao, J. 2018. Large scaled relation extraction with reinforcement learning. In *Proceedings of AAAI*, 5658–5665.

Zhou, G.; Su, J.; Zhang, J.; and Zhang, M. 2005. Exploring various knowledge in relation extraction. In *Proceedings of ACL*, 427–434.