Few-shot Relation Extraction via Bayesian Meta-learning on Relation Graphs

Meng Qu 1 2  Tianyu Gao 3 Louis-Pascal A. C. Xhonneux 1 2  Jian Tang 1 4 5

Abstract

This paper studies few-shot relation extraction, which aims at predicting the relation for a pair of entities in a sentence by training with a few labeled examples in each relation. To more effectively generalize to new relations, in this paper we study the relationships between different relations and propose to leverage a global relation graph. We propose a novel Bayesian meta-learning approach to effectively learn the posterior distribution of the prototype vectors of relations, where the initial prior of the prototype vectors is parameterized with a graph neural network on the global relation graph. Moreover, to effectively optimize the posterior distribution of the prototype vectors, we propose to use the stochastic gradient Langevin dynamics, which is related to the MAML algorithm but is able to handle the uncertainty of the prototype vectors. The whole framework can be effectively and efficiently optimized in an end-to-end fashion. Experiments on two benchmark datasets prove the effectiveness of our proposed approach against competitive baselines in both the few-shot and zero-shot settings.

1. Introduction

A fundamental problem in natural language processing is relation extraction, which aims to identify the relations between entities in sentences. The problem is usually studied as a supervised classification task by training with labeled sentences. However, annotating a large set of sentences is time-consuming and expensive. As a result, the number of labeled data is very limited for this task. In practice, a common solution to this challenge is distant supervision (Mintz et al., 2009), where a knowledge graph is utilized to automatically generate training data. For example, given a triplet (Washington, Capital, US) in a knowledge graph, all the sentences containing the two entities Washington and US will be labeled with the relation Capital. In this way, a large quantity of training data can be generated for each relation, and such an approach has been extensively studied and has been proven very effective. However, a limitation of distant supervision is that the generated training data can be very noisy. This is because there could be multiple relations between two entities, and it is hard to determine which relation the entity pair belongs to in a particular context, or whether there is any relation expressed by the sentence.

An alternative approach for relation extraction, which is attracting growing interest, is meta-learning for relation extraction (Han et al., 2018; Gao et al., 2019). The idea of meta-learning is to train models with a large number of diverse tasks, each of which has a few examples for demonstration, so that the learned model can quickly generalize to new tasks with only a few examples. For example, the Model-Agnostic Meta-Learning (MAML) algorithm (Finn et al., 2017) tries to find a good initialization for the parameters of a neural model, based on which the model can be quickly adapted to a new task with several steps of gradient descent. Another example is the prototypical network (Snell et al., 2017), which learns a prototype vector from a few examples for each class, and further uses the prototype vectors for prediction. Based on these techniques, handful recent studies (Han et al., 2018; Gao et al., 2019) are able to train relation extraction models with only a few examples for each relation. Although these methods achieve encouraging improvements, the performance remains unsatisfactory as the amount of information in training data is still limited.

To more effectively generalize to new relations and tasks, in this paper we study modeling the relationships between different relations, and propose to leverage a global graph between different relations. In practice, such a global graph can be obtained in different ways. For example, we can use the knowledge graph embedding algorithms (Bordes et al., 2013; Sun et al., 2019) to infer the relation embeddings and then construct a K-nearest neighbor graph based on the relation embeddings. The global relation graph provides the prior knowledge on the relationships between different relations, which allows us to transfer supervision between these relations and even generalize to these relations without...
leveraging any labeled sentences (i.e. zero-shot learning).

Moreover, we propose a novel Bayesian meta-learning approach for few-shot relation extraction, which solves the problem by learning the prototype vectors of relations based on the labeled sentences (a.k.a. support set) and the global relation graph. Instead of learning a point estimation as in MAML (Finn et al., 2017) or prototypical networks (Snell et al., 2017), we follow existing work on Bayesian meta-learning (Gordon et al., 2019; Kim et al., 2018) and aim to infer the posterior distribution of the prototype vectors, which is able to effectively handle the uncertainty of the vectors. The posterior can be naturally factorized into a likelihood function on the support set, and a prior of prototype vectors obtained from the global graph. We parameterize the prior distribution of prototype vectors of relations by applying a graph neural network (Kipf & Welling, 2017) to the global graph, allowing us to effectively utilize the relationships between different relations encoded in the graph.

For the posterior distribution of prototype vectors, existing studies (Gordon et al., 2019; Ravi & Beatson, 2019) usually parameterize it as a Gaussian distribution, and amortized variational inference is generally used to learn the posterior distribution. However, the posterior distribution of prototype vectors can be much more complicated than the Gaussian distribution. In this paper, instead of assuming a specific distribution for the posterior distribution of prototype vectors, we propose to directly sample from the posterior distribution with the stochastic gradient Langevin dynamics technique (Welling & Teh, 2011), which is very general and can be applied to different distributions. Our approach can be viewed as a stochastic version of MAML (Finn et al., 2017), where random noises are added at each step of the gradient descent to model the uncertainty of prototype vectors. The sampled prototype vectors are further used to make predictions for query sentences in test sets, and the whole framework can be optimized in an end-to-end fashion.

We conduct extensive experiments to evaluate the proposed approach on two benchmark datasets of few-shot relation extraction. Empirical results prove the effectiveness of our proposed approach over many competitive baselines in both the settings of few-shot and zero-shot relation extraction.

2. Related Work

2.1. Few-shot Learning and Meta-learning

Our work is related to few-shot learning and meta-learning. The goal is to train deep learning models with diverse tasks, where each task is specified by a few examples for demonstration, so that the model can be quickly adapted to new tasks. One type of representative methods is the metric-based method (Vinyals et al., 2016; Snell et al., 2017; García & Bruna, 2018; Sung et al., 2018). The basic idea is to learn a prototype vector for each class based on the few examples, and use the prototype vectors for prediction. Another type of representative methods is the optimization-based method (Finn et al., 2017; Ravi & Larochelle, 2017). Typically, these methods formalize the problem as a bi-level optimization problem (Franceschi et al., 2018). The outer loop learns global parameters shared across different tasks, such as the initialization of model parameters. The inner loop adapts the shared parameters to each task by performing several steps of gradient descent according to the few examples. Compared with these methods, which aim to learn a point estimation of prototype vectors or model parameters, our approach treats them as random variables and models their posterior distributions, which can thus handle the uncertainty of these prototype vectors or parameters.

In addition, there are several recent studies (Kim et al., 2018; Gordon et al., 2019; Ravi & Beatson, 2019) also using Bayesian learning techniques for meta-learning, where the posterior distributions of prototype vectors or model parameters are inferred. However, these methods ignore the relationships of different classes, while we model their relationships by applying a graph neural network (Kipf & Welling, 2017; Gilmer et al., 2017; Veličković et al., 2018) to a global graph of classes, allowing our approach to better generalize to all different classes. Furthermore, we model the posterior distribution in a more effective way. For Gordon et al. (2019) and Ravi & Beatson (2019), they use a simple Gaussian distribution parameterized by an amortization network to approximate the true posterior distribution. However, the true posterior distribution can be more complicated than the Gaussian distribution, and hence these methods are less precise. Another method from Kim et al. (2018) uses Stein Variational Gradient Descent (SVGD) (Liu & Wang, 2016) to draw samples from the posterior distribution for optimization, but SVGD relies on a properly-designed kernel function for different samples, which can be hard to choose. In contrast, our approach uses the stochastic gradient Langevin dynamics (Welling & Teh, 2011) to perform Monte Carlo sampling for optimization, which is more flexible and effective as we will show in the experiment.

2.2. Relation Extraction

Relation extraction is a fundamental task in natural language processing. Given two entities in a sentence, the goal is to predict the relation expressed in the sentence. Most existing studies (Zeng et al., 2014; 2015; Zhang et al., 2017) focus on the supervised or semi-supervised settings of relation extraction, where they assume massive labeled sentences are available. However, the number of labeled sentences can be very limited in practice. Some studies try to solve the challenge of insufficient labeled sentences by resorting to knowledge graphs, where existing facts are used to annotate unlabeled sentences through distant supervision (Mintz
et al., 2009) or provide additional training signals (Schwartz et al., 2016; Qu et al., 2018; Zhu et al., 2019a). Nevertheless, the data or signals obtained in this way can be very noisy. Some more recent studies (Han et al., 2018; Gao et al., 2019; Soares et al., 2019) instead focus on few-shot relation extraction, and the goal is to develop models which can be trained with only a few labeled sentences. By combining meta-learning methods with BERT encoders (Devlin et al., 2019), these methods achieve impressive results. However, they ignore the connections of different relations, which naturally exist in practice. In contrast, we treat a global graph of relations as prior knowledge, and propose a principled Bayesian meta-learning approach to leverage the global graph, which is able to better generalize across different relations.

3. Problem Definition

Relation extraction is an important task in many research areas, which aims at predicting the relation of two entities given a sentence. Most existing methods for relation extraction require a large number of labeled sentences as training data, which are expensive to obtain. Therefore, more recent studies focus on few-shot relation extraction, where only a few examples for each relation are given as training data. However, the results are still far from satisfactory due to the limited information in these examples. To further improve the results, another data source should be considered.

In this paper, we propose to study few-shot relation extraction with a global graph of relations, where a global graph describing the relationships of all possible relations is assumed to be an additional data source. More formally, we denote the global relation graph as \( G = (R, L) \), where \( R \) is the set of all the possible relations, and \( L \) is a collection of links between different relations. The linked relations are likely to have more similar semantic meanings.

In few-shot relation extraction, each time we only consider a subset of relations from the whole relation set, i.e., \( T \subseteq R \). Given a few support sentences \( S \) of these relations, where \( x_S = \{x_s\}_{s \in S} \) represents the text of these sentences, and \( y_S = \{y_s\}_{s \in S} \) represents the corresponding labels with each \( y_s \in T \), our goal is to learn a neural classifier for these relations by using the global graph and support sentences. Then given some unlabeled sentences as queries \( x_Q = \{x_q\}_{q \in Q} \), we will apply the classifier to predict their labels, i.e., \( y_Q = \{y_q\}_{q \in Q} \) with each \( y_q \in T \).

4. Model

In this section, we introduce our approach for few-shot relation extraction with global relation graphs. Our approach represents each relation with a prototype vector, which can be used for classifying the query sentences. In contrast to most existing meta-learning methods, which learn a point estimation of the prototype vector, we treat the prototype vector as a random variable to model its posterior distribution. The posterior distribution is naturally composed of two terms, i.e., a prior of the prototype vector obtained from the global relation graph, and a likelihood function on the support sentences. Our approach parameterizes the prior distribution by applying a graph neural network to the global relation graph. By using such a graph-based prior, our approach can effectively generalize to different relations.

However, optimization remains challenging in our approach, as the posterior distribution of prototype vectors has a complicated form. To address that, we approximate the posterior distribution through Monte Carlo sampling, where multiple samples of prototype vectors are drawn by using the stochastic gradient Langevin dynamics. By doing so, our approach can be effectively optimized in an end-to-end fashion.

4.1. A Probabilistic Formalization

Our approach uses Bayesian learning techniques for few-shot relation extraction, where we formalize the problem in a probabilistic way. More specifically, recall that given a subset of relations \( T \subseteq R \), the goal is to predict the labels \( y_Q \) of some query texts \( x_Q \) based on a global relation graph \( G \) and a few support sentences \( (x_S, y_S) \). Formally, our goal can be stated as computing the following log-probability:

\[
\log p(y_Q | x_Q, x_S, y_S, G).
\]  

(1)

We compute the probability by representing each relation \( r \in T \) with a prototype vector \( v_r \), which summarizes the semantic meaning of that relation. By introducing such prototype vectors, the log-probability can be factorized as:

\[
\log p(y_Q | x_Q, x_S, y_S, G) = \log \int p(y_Q | x_Q, v_T)p(v_T | x_S, y_S, G)dv_T
\]  

(2)

where \( v_T = \{v_r\}_{r \in T} \) is a collection of prototype vectors for all the target relations in \( T \). These prototype vectors are characterized by both the labeled sentences in the support set and global relation graph through the distribution \( p(v_T | x_S, y_S, G) \). With such prototype vectors to represent target relations, the distribution of query sentence labels can then be defined through a softmax function as follows:

\[
p(y_Q | x_Q, v_T) = \prod_{q \in Q} p(y_q | x_q, v_T), \quad \text{with each}
\]

\[
p(y_q = r | x_q, v_T) = \frac{\exp(E(x_q) \cdot v_r)}{\sum_{r' \in T} \exp(E(x_q) \cdot v_{r'})}.
\]  

(3)

where \( E \) is a sentence encoder, which encodes a query sentence \( x_q \) into an encoding \( E(x_q) \). Intuitively, we compute the dot product of the encoding and the prototype vector \( v_r \) to estimate how likely the sentence expresses the relation.
Under such a formalization, the key is how to parameterize 
\( p(v_T|x_S, y_S, G) \), which is the posterior distribution of proto-
type vectors conditioned on the support sentences and the 
global relation graph. Next, we introduce how we parameter-
ize the posterior distribution in our proposed approach.

4.2. Parameterization of the Posterior Distribution

To model the posterior distribution of prototype vectors, we 
notice that the posterior can be naturally factorized into a 
prior distribution conditioned on the support sentences and a 
likelihood function on the few support sentences. Therefore, 
we can formally represent the posterior as follows:

\[
p(v_T|x_S, y_S, G) \propto p(y_S|x_S, v_T)p(v_T|G), \tag{4}
\]

where \( p(v_T|G) \) is the prior for the prototype vectors and 
\( p(y_S|x_S, v_T) \) is the likelihood on support sentences.

To extract the knowledge from the global relation graph 
to characterize the prior distribution, we introduce a graph 
normalized function (Kipf & Welling, 2017; Gilmer et al., 2017; 
Veličković et al., 2018; Qu et al., 2019) in our approach. 
The graph neural network \( \mathcal{F} \), which can learn a 
latent representation \( h_r \) for each relation \( r \), i.e., \( h_r = \mathcal{F}(G)_r \). 
More specifically, the graph neural network \( \mathcal{F} \) initializes 
the latent embedding \( h_r \) of each relation as its initial feature 
vector. Then \( \mathcal{F} \) iteratively updates the latent embedding of 
each relation \( r \) according to the embeddings of \( r \) and \( r \)'s 
neighbors. Formally, \( \mathcal{F} \) updates the embeddings as follows:

\[
h_r \leftarrow \mathcal{U} \left\{ \sum_{r' \in \text{NB}(r)} \mathcal{M}(h_{r'}, h_r) \right\}, \tag{5}
\]

where \( \text{NB}(r) \) is the neighbor of \( r \) in the global graph, and 
\( \mathcal{M} \) is a transformation function. Basically, for each relation 
\( r \), we apply \( \mathcal{M} \) to the latent embeddings of \( r \)'s 
neighbors and then aggregate the transformed embeddings together. 
Finally, the latent embedding of \( r \) is updated based on its 
previous value and the aggregated embeddings through an 
update function \( \mathcal{U} \). After several rounds of such update, 
the relationships between different relations encoded in 
the global graph can be effectively preserved into the final relation 
embeddings, which can serve as regularization for the 
prototype vectors. Motivated by that, we parameterize the 
prior distribution of prototype vectors \( p(v_T|G) \) as follows:

\[
p(v_T|G) = \prod_{r \in T} p(v_r|h_r) = \prod_{r \in T} \mathcal{N}(v_r|h_r, I), \tag{6}
\]

where we model the prior distribution of each relation \( r \in T \) 
independently. For each relation, we define its prior as a 
Gaussian distribution, where the mean is set as the latent 
representation \( h_r \) given by the graph neural network \( \mathcal{F} \). In 
this way, the knowledge from the relation graph can be 
effectively distilled into the prior distribution, allowing our 
approach to better generalize to a wide range of relations.

Besides the graph-based prior, we also consider the likelihood 
on support sentences when parameterizing the posterior 
distribution of prototype vectors. Similar to the likelihood 
on the query sentences in Eq. (3), the likelihood on support sentences can be characterized as below:

\[
p(y_s|x_s, v_T) = \prod_{s \in S} p(y_s|x_s, v_T), \text{ with each} \tag{7}
\]

\[
p(y_s = r|x_s, v_T) = \frac{\exp(\mathcal{E}(x_s) \cdot v_r)}{\sum_{r' \in T} \exp(\mathcal{E}(x_s) \cdot v_{r'})},
\]

where \( \mathcal{E} \) is the sentence encoder. By applying the likelihood 
on support sets to the prior distribution of prototype vectors, 
we can effectively adapt the prior distribution to the target 
relations with the few support sentences. In this way, the 
posterior distribution combines the knowledge from both 
the global relation graph and the support sentences, which 
can thus be used to effectively classify query sentences.

4.3. Optimization and Prediction

In the above section, we have introduced how we parameterize 
the posterior distribution of prototype vectors. Next, we 
explain the model optimization and prediction.
During both model optimization and prediction, we have to deal with the log-probability $\log p(y_Q|x_Q,x_S,y_S,G)$, where we either maximize or compute this value. However, according to Eq. (2), the log-probability relies on the integration over prototype vectors, which is intractable. Therefore, we estimate the log-probability with Monte Carlo sampling, where several samples of prototype vectors are drawn from the posterior distribution for approximation:

$$
\log p(y_Q|x_Q,x_S,y_S,G) = \log \mathbb{E}_{p(y_T|x_S,y_S,G)}[p(y_Q|x_Q,v_T)] \\
\approx \log \frac{1}{L} \sum_{l=1}^{L} p(y_Q|x_Q,v_T^{(l)}).
$$

(8)

Here, $v_T^{(l)}$ is a sample drawn from the posterior distribution, i.e., $v_T^{(l)} \sim p(v_T|x_S,y_S,G)$. However, as shown in Eq. (4), the posterior distribution combines both a graph-based prior and a likelihood function on support sentences. The graph-based prior is a Gaussian distribution while the likelihood function is specified by a softmax function. Therefore, the posterior distribution has a very complicated form, and sampling from the posterior is nontrivial. To address the problem, in this paper we use the stochastic gradient Langevin dynamics (Welling & Teh, 2011), which performs sampling via multiple stochastic updates. Formally, to draw a sample $\hat{v}_T$ of the prototype vector, we can randomly initialize the sample $v_T$, and iteratively update the sample as follows:

$$
\hat{v}_T \leftarrow \hat{v}_T + \frac{1}{2} \nabla_{\hat{v}_T} \log p(y_S|x_S,\hat{v}_T)p(\hat{v}_T|G) \\
+ \sqrt{\epsilon} \tilde{z},
$$

(9)

where $\tilde{z} \sim \mathcal{N}(0, I)$ is a sample from the standard Gaussian distribution. After a burn-in period, $\hat{v}_T$ can then be treated as a true sample from the posterior distribution of prototype vectors. In the above update rule, the term $\nabla_{\hat{v}_T} \log p(y_S|x_S,\hat{v}_T)p(\hat{v}_T|G)$ is highly related to the MAML algorithm (Finn et al., 2017), as they both aim at moving the sample towards maximizing the likelihood $\log p(y_S|x_S,\hat{v}_T)$ on support sentences, and thereby adapt to the target relations in $T$. The difference is that we also leverage the graph-based prior $p(\hat{v}_T|G)$ to guide this process. Moreover, at each step, a random noise $\tilde{z}$ is added, allowing us to get different samples from the posterior distribution $p(v_T|x_S,y_S,G)$, rather than a single sample with maximum posterior probability. In other words, our approach is able to model the uncertainty of prototype vectors.

However, the Langevin dynamics requires a burn-in period, which can take a long time. To accelerate this process, it is helpful to let the samples stay in the high-density regions of the posterior distribution, such that we can better explore around those regions (Gong et al., 2019). Therefore, we try to initialize the sample $\hat{v}_T$ at a point with high posterior probability. Towards this goal, we theoretically justify in appendix that a proper initialization can be given as below:

$$
\hat{v}_T \leftarrow \{\hat{v}_r\}_{r \in T}, \text{ with each} \\
\hat{v}_r \leftarrow m_r + h_r - m,
$$

(10)

where $h_r$ is the latent embedding of relation $r$ given by the graph neural network on the global relation graph, $m_r$ is the mean encoding of all the sentences under relation $r$ in the support set, and $m$ is the mean encoding of all the sentences in the support set. Intuitively, for each relation $r$, we add the latent embedding $h_r$ from the global relation graph and the mean encoding $m_r$ from the given examples of that relation. Also, we subtract the mean encoding $m$ of all sentences in the support set, so that we can better distinguish sentences from different relations. In practice, we introduce two hyperparameters for $h_r$ and $m$ to control their relative weights. With such an initialization, we can empirically guarantee that the Langevin dynamics will quickly converge.

Once we obtain the samples of prototype vectors from their posterior distribution, we can then use the samples to optimize and compute $\log p(y_Q|x_Q,x_S,y_S,G)$ according to Eq. (8). The whole optimization process is end-to-end, and we summarize the optimization algorithm in Alg. 1.

### 5. Experiment

In this section, we empirically evaluate our proposed approach on two benchmark datasets, and we consider both the few-shot and the zero-shot learning settings.

#### 5.1. Datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>FewRel</th>
<th>NYT-25</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Possible Relations</td>
<td>828</td>
<td>828</td>
</tr>
<tr>
<td>Training Relations</td>
<td>64</td>
<td>10</td>
</tr>
<tr>
<td>Validation Relations</td>
<td>16</td>
<td>5</td>
</tr>
<tr>
<td>Test Relations</td>
<td>20</td>
<td>10</td>
</tr>
<tr>
<td>Sentences per Relation</td>
<td>700</td>
<td>100</td>
</tr>
</tbody>
</table>
We use two benchmark datasets for evaluation. One dataset is the FewRel dataset (Han et al., 2018; Gao et al., 2019), which has been recently proposed for few-shot relation extraction. Note that only the training set and validation set of FewRel are released, and the test set is not public, so researchers have to evaluate models on the remote cluster provided by the FewRel team. Because of that, we conduct most of the performance analysis on the validation set of FewRel, and for the test set we only report a final number.

The other dataset is NYT-25. The raw data of NYT-25 is from the official website of FewRel \(^1\) (Han et al., 2018; Gao et al., 2019), where labeled sentences under 25 relations are provided by annotating the New York Times data. However, the dataset has not been split into training, validation and test sets. Therefore, we randomly sample 10 relations for training, 5 for validation and the remaining 10 for test.

For both datasets, the relations are from a knowledge graph named Wikidata \(^2\), which has 828 unique relations in total. To construct the global graph of all the relations, we first employ GraphVite (Zhu et al., 2019b) to run the TransE algorithm (Bordes et al., 2013) on Wikidata to learn a 512-dimensional embedding vector for each relation. Then we use the relation embeddings to construct a 10-nearest neighbor graph as the global relation graph, and the learned relation embeddings are treated as the initial relation features in GNN \( \mathcal{F} \). The detailed statistics are summarized in Table 1.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>5-Way 1-Shot</th>
<th>5-Way 5-Shot</th>
<th>10-Way 1-Shot</th>
<th>10-Way 5-Shot</th>
</tr>
</thead>
<tbody>
<tr>
<td>GNN</td>
<td>75.66</td>
<td>89.06</td>
<td>70.08</td>
<td>76.93</td>
</tr>
<tr>
<td>SNAIL</td>
<td>67.75</td>
<td>86.58</td>
<td>54.29</td>
<td>77.54</td>
</tr>
<tr>
<td>Proto</td>
<td>80.68</td>
<td>89.60</td>
<td>71.48</td>
<td>82.89</td>
</tr>
<tr>
<td>Pair</td>
<td>88.32</td>
<td>93.22</td>
<td>80.63</td>
<td>87.02</td>
</tr>
<tr>
<td>MTB</td>
<td>89.80</td>
<td>93.59</td>
<td>83.37</td>
<td>88.64</td>
</tr>
<tr>
<td>MAML</td>
<td>89.70</td>
<td>93.55</td>
<td>83.17</td>
<td>88.51</td>
</tr>
<tr>
<td>Versa</td>
<td>88.52</td>
<td>93.15</td>
<td>81.62</td>
<td>87.73</td>
</tr>
<tr>
<td>BMAML</td>
<td>89.65</td>
<td>93.40</td>
<td>82.94</td>
<td>88.20</td>
</tr>
<tr>
<td>REGRAB</td>
<td>90.30</td>
<td>94.25</td>
<td>84.09</td>
<td>89.93</td>
</tr>
</tbody>
</table>

\(^1\) https://github.com/thunlp/FewRel

\(^2\) https://www.wikidata.org/wiki/Wikidata:Main_Page

Table 2. Results of few-shot classification on the FewRel test set (%). We rerun all the algorithms with the same sentence encoder BERT\(_{\text{BASE}}\). Our approach outperforms all the baseline methods.

In our approach, we use BERT\(_{\text{BASE}}\) (Devlin et al., 2019) as encoder to encode all the tokens in a sentence. Then we follow Soares et al. (2019) and combine the token encodings of entity mentions in a sentence as the sentence encoding.

We do the same thing for the advanced meta-learning algorithms, i.e., MTB, Proto, MAML, Versa, BMAML, in order to make fair comparison. The details of how we compute sentence encodings are explained in the appendix. For the softmax function of the likelihood on support and query sentences, we apply an annealing temperature of 10. For the Gaussian prior of prototype vectors, we apply a one-layer convolutional network \( \mathcal{F} \) to the global relation graph to compute the mean. We have also tried more layers, but only obtain very marginal improvement. For the stochastic gradient Langevin dynamics, the number of samples to draw is set as 10 by default, which is the same as used by other Bayesian meta-learning methods, and we perform 5 steps of update for these samples with the initial step size \((\epsilon, \kappa, \tau)\) as 0.1 by default. The graph encoder and sentence encoder are tuned by SGD with learning rate as 0.1. For other hyperparameters, they are selected on the FewRel validation set through grid search.

5.3. Parameter Settings

5.4. Results

5.4.1. Comparison with Baseline Methods

The main results on FewRel test set, FewRel validation set and NYT-25 test set are presented in Tab. 2, Tab. 3 and
Table 3. Results of few-shot classification on FewRel validation set (%). All the methods use the same encoder for fair comparison.

<table>
<thead>
<tr>
<th>Category</th>
<th>Algorithm</th>
<th>5-Way</th>
<th>5-Way</th>
<th>10-Way</th>
<th>10-Way</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1-Shot</td>
<td>5-Shot</td>
<td>1-Shot</td>
<td>5-Shot</td>
</tr>
<tr>
<td>Meta-learning</td>
<td>Pair</td>
<td>85.66</td>
<td>89.48</td>
<td>76.84</td>
<td>81.76</td>
</tr>
<tr>
<td></td>
<td>MTB</td>
<td>84.61</td>
<td>88.76</td>
<td>75.22</td>
<td>80.15</td>
</tr>
<tr>
<td></td>
<td>Proto</td>
<td>82.92</td>
<td>91.32</td>
<td>73.24</td>
<td>83.68</td>
</tr>
<tr>
<td></td>
<td>MAML</td>
<td>82.93</td>
<td>86.21</td>
<td>73.20</td>
<td>76.06</td>
</tr>
<tr>
<td>Bayesian Meta-learning</td>
<td>Versa</td>
<td>84.47</td>
<td>88.44</td>
<td>74.70</td>
<td>79.20</td>
</tr>
<tr>
<td></td>
<td>BMAML</td>
<td>85.80</td>
<td>89.71</td>
<td>76.66</td>
<td>81.34</td>
</tr>
<tr>
<td></td>
<td>REGRAB</td>
<td>87.95</td>
<td>92.54</td>
<td>80.26</td>
<td>86.72</td>
</tr>
</tbody>
</table>

Table 4. Results of few-shot classification on NYT-25 test set (%). All the methods use the same sentence encoder for fair comparison.

<table>
<thead>
<tr>
<th>Category</th>
<th>Algorithm</th>
<th>5-Way</th>
<th>5-Way</th>
<th>10-Way</th>
<th>10-Way</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1-Shot</td>
<td>5-Shot</td>
<td>1-Shot</td>
<td>5-Shot</td>
</tr>
<tr>
<td>Meta-learning</td>
<td>Pair</td>
<td>80.78</td>
<td>88.13</td>
<td>72.65</td>
<td>79.68</td>
</tr>
<tr>
<td></td>
<td>MTB</td>
<td>88.90</td>
<td>95.53</td>
<td>83.08</td>
<td>92.23</td>
</tr>
<tr>
<td></td>
<td>Proto</td>
<td>77.63</td>
<td>87.25</td>
<td>66.49</td>
<td>79.51</td>
</tr>
<tr>
<td></td>
<td>MAML</td>
<td>86.96</td>
<td>93.36</td>
<td>79.62</td>
<td>88.32</td>
</tr>
<tr>
<td>Bayesian Meta-learning</td>
<td>Versa</td>
<td>87.70</td>
<td>93.77</td>
<td>81.27</td>
<td>89.35</td>
</tr>
<tr>
<td></td>
<td>BMAML</td>
<td>87.03</td>
<td>93.90</td>
<td>79.74</td>
<td>88.72</td>
</tr>
<tr>
<td></td>
<td>REGRAB</td>
<td>89.76</td>
<td>95.66</td>
<td>84.11</td>
<td>92.48</td>
</tr>
</tbody>
</table>

Tab. 4 respectively. For fair comparison, the same sentence encoder BERT<sub>BASE</sub> is used for all the compared approaches. From Tab. 2, we can see that the results of GNN and SNAIL are less competitive, showing that they are less effective to model textual data. Compared with Pair and MTB, which are specifically designed for few-shot relation extraction, our approach achieves relatively better results in all the tables, showing that our approach can better generalize to a variety of relations given a few examples. Besides, our approach also outperforms other meta-learning methods. Comparing our approach with MAML and the prototypical network (Proto), the performance gain mainly comes from two aspects. On the one hand, our approach considers a global graph of different relations, which provides prior knowledge about the relationships of all the relations, allowing our approach to better adapt to different relations. On the other hand, our approach uses a Bayesian learning framework, which effectively deals with the uncertainty of the prototype vectors for different relations. Moreover, our approach is also superior to other Bayesian meta-learning methods, i.e., Versa and BMAML. The reason is that we consider a graph-based prior in the posterior distribution, making our approach more powerful. Also, our approach performs optimization through Monte Carlo sampling with stochastic gradient Langevin dynamics, which models and optimizes the posterior distribution in a more effective way.

5.4.2. Analysis of the Graph-based Prior

Compared with existing studies on few-shot relation extraction, our approach uses a global relation graph. The relation graph provides knowledge of the relationship between different relations, allowing our approach to better generalize to various relations. To leverage such knowledge, our approach parameterizes the prior distribution of prototype vectors for different relations by applying a graph neural network to the relation graph. In this section, we conduct experiments to analyze the effect of such a graph-based prior distribution.

Table 5. Analysis of the graph-based prior on FewRel validation set (%). Removing the graph-based prior reduces the accuracy.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>5-Way</th>
<th>10-Way</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1-Shot</td>
<td>1-Shot</td>
</tr>
<tr>
<td>REGRAB with graph prior</td>
<td>87.95</td>
<td>80.26</td>
</tr>
<tr>
<td>REGRAB w/o graph prior</td>
<td>85.82</td>
<td>77.70</td>
</tr>
</tbody>
</table>

We first conduct some ablation study on the FewRel validation set, where we compare two variants of our approach, i.e., with or without using the graph-based prior. The results are presented in Tab. 5. We can see that removing the graph-based prior results in significantly worse results, showing the effect of the graph-based prior for improving the relation extraction performance in the few-shot learning setting.

Moreover, with such a graph-based prior, our approach is able to handle relation extraction in the zero-shot learning setting, where no labeled sentences of each relation are given. Next, we present results on the FewRel validation set and NYT-25 test set to justify that. Remember our approach constructs the relation graph based on the pre-trained relation embeddings, and then applies a graph neural network for parameterizing the prior of prototype vectors. Compared with using graph neural networks, a more straightforward way is to directly apply a feedforward neural network to the pre-trained relation embeddings to derive the prior of prototype vectors. To demonstrate the advantage of graph neural networks, we also compare with the aforementioned variant.
Figure 2. Results in zero-shot learning settings. Our approach can still achieve quite good results even without any labeled sentences.

We present the results in Fig. 2. From the figure, we can see that even without any labeled sentences as demonstration, our approach is still able to achieve quite impressive results, which proves its effectiveness. Also, comparing with the variant without using graph neural networks, our approach achieves significantly better results in both datasets. The observation shows that the graph neural network can help our approach better leverage the relationships of relations.

5.4.3. Analysis of the Optimization Algorithm

Table 6. Ablation study of the optimization algorithm on FewRel validation set (%). The Langevin dynamics leads to better results.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>5-Way 1-Shot</th>
<th>16-Way 1-Shot</th>
</tr>
</thead>
<tbody>
<tr>
<td>REGRAB with Langevin</td>
<td>87.95</td>
<td>60.26</td>
</tr>
<tr>
<td>REGRAB with Amortized VI</td>
<td>85.74</td>
<td>77.41</td>
</tr>
</tbody>
</table>

Compared with existing methods, the other difference of our approach is that we use the stochastic gradient Langevin dynamics during training, where several samples of prototype vectors are drawn for optimization. More specifically, we initialize a set of samples according to Eq. (10), and then perform multiple steps of update based on Eq. (9). In this part, we thoroughly analyze the optimization algorithm.

We start with the ablation study on the FewRel validation set, where we compare with a variant which parameterizes the posterior of prototype vectors as a Gaussian distribution. The mean of the Gaussian distribution is set as the value given by Eq. (10), which is the same as the initialization of samples we use in the Langevin dynamics. Such a variant is similar to the amortized variational inference method used in Versa (Gordon et al., 2019), an existing Bayesian meta-learning algorithm. We present the results in Tab. 6. We see that our approach with Langevin dynamics achieves relatively better results than the variant with amortized variational inference, which proves the effectiveness of approximating the posterior distribution of prototype vectors by drawing samples with the Langevin dynamics.

In addition, when drawing samples from the posterior distribution, the Langevin dynamics performs multiple steps of update on a set of samples. Therefore two important hyperparameters of this process are the number of samples and the number of update steps. Next, we analyze these hyperparameters by doing sensitivity analysis. We take the FewRel validation set as an example, and report the accuracy of 5-way 1-shot classification. To better understand the result, we introduce a variant, where we only initialize the samples of prototype vectors through Eq. (10), without further updating them according to Eq. (9). Fig. 3(a) shows the results under different numbers of samples. We can see if only one or two samples are used, the results are quite poor, which are even worse than the variant without updating the samples. The reason is that if we only use few samples, the estimation of the log-probability in Eq. (8) can have high variance, leading to poor results. As we use more samples, the results are quickly improved, and the results converge when around 8-10 samples are used, which is quite efficient. Besides, Fig. 3(b) presents the results under different numbers of update steps. As the number of steps is increased, the accuracy also rises, since the samples are moving towards high-density regions of the posterior to perform exploration. After only 4-5 steps, the accuracy quickly converges, which is very efficient. This observation proves the effectiveness of our initialization strategy presented in the Eq. (10).

6. Conclusion

This paper studies relation extraction in the few-shot learning setting, and the key idea is to consider a global relation graph, which captures the global relationship between relations. We propose a novel Bayesian meta-learning approach, which aims to model the posterior distribution of prototype vectors for different relations. The prior distribution in the posterior is parameterized by applying a graph neural network to the global relation graph. The stochastic gradient Langevin dynamics is used to optimize the posterior distribution. Experiments on two datasets prove the effectiveness of our approach. In the future, we plan to study learning the structure of the relation graph automatically by following existing studies (Franceschi et al., 2019). Besides, we also plan to apply our approach to other applications, such as few-shot image classification (Triantafillou et al., 2020).
Acknowledgements

This project is supported by the Natural Sciences and Engineering Research Council (NSERC) Discovery Grant, the Canada CIFAR AI Chair Program, collaboration grants between Microsoft Research and Mila, Amazon Faculty Research Award, Tencent Lab Rhino-Bird Gift Fund and a NRC Collaborative R&D Project (AI4D-CORE-06). We would like to thank Andreea Deac and Zhaocheng Zhu for reviewing the paper before submission.

References


Mintz, M., Bills, S., Snow, R., and Jurafsky, D. Distant supervision for relation extraction without labeled data. In ACL, 2009.


