Hui et al. EURASIP Journal on Wireless Communications and Networking (2020) 2020:118
https://doi.org/10.1186/s13638-020-01720-6

EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

Few-shot relation classification by context attention-based prototypical networks with BERT

Bei Hui, Liang Liu, Jia Chen*, Xue Zhou and Yuhui Nian

* Correspondence: jchen@uestc.edu.cn
School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China

Abstract

Human-computer interaction under the cloud computing platform is very important, but the semantic gap will limit the performance of interaction. It is necessary to understand the semantic information in various scenarios. Relation classification (RC) is an import method to implement the description of semantic formalization. It aims at classifying a relation between two specified entities in a sentence. Existing RC models typically rely on supervised learning and distant supervision. Supervised learning requires large-scale supervised training datasets, which are not readily available. Distant supervision introduces noise, and many long-tail relations still suffer from data sparsity. Few-shot learning, which is widely used in image classification, is an effective method for overcoming data sparsity. In this paper, we apply few-shot learning to a relation classification task. However, not all instances contribute equally to the relation prototype in a text-based few-shot learning scenario, which can cause the prototype deviation problem. To address this problem, we propose context attention-based prototypical networks. We design context attention to highlight the crucial instances in the support set to generate a satisfactory prototype. Besides, we also explore the application of a recently popular pre-trained language model to few-shot relation classification tasks. The experimental results demonstrate that our model outperforms the state-of-the-art models and converges faster.

Keywords: Attention mechanism, Few-shot learning, Language model, Relation classification

0.0.0.1 Introduction In a cloud-computing scenario, human-computer interaction operations occur frequently [1–4]. However, there usually exists a semantic gap between human and computer's understanding of mutual behaviors. The contradiction of semantic understanding greatly limits human-computer interaction. It is a feasible method to solve the semantic gap of human-computer interaction by using artificial intelligence technology [5–10]. Among all of the AI technologies, semantic understanding is an effective way to solve the semantic gap [11, 12]. The semantic cognitive formal description is the main method, and it is the core of data collection, analysis and processing in human-computer interaction. Named entity recognition (NER) and relation classification (RC) are used to capture semantic information and implement the description of semantic formalization.

NER has made great progress in knowledge acquisition, but RC is still difficult to solve when data is sparse. Our research focuses on the classification of relations in few-shot scenarios.

RC is an important task in knowledge acquisition, which aims at identifying a type of relation between two specified entities based on their related context. Because it benefits many natural language processing (NLP) applications (e.g., question answering [13] and knowledge base completion [14]), many approaches have been proposed for this task. Of these approaches, supervised models have been widely used in this task [15–19]. However, these models are typically limited by the quantity and quality of the training data because manual labeling of high-quality training data is time-consuming and labor-intensive. Besides, in computing paradigms, the model should be fast and take up less space.

To overcome the problem of insufficient data, distant supervision (DS) was proposed by Mintz [20]. DS is a heuristic rule: for an entity pair in knowledge graphs (KGs), the sentences that mention both entities will be labeled with their relations in KGs. A large-scale training dataset can be obtained via DS. However, DS inevitably introduces noise. Many efforts have been devoted to reducing this noise [21–26]. Although DS realizes satisfactory results on common relations, its performance will degrade for long-tail relations [27]. Hence, it is necessary to study the RC model when the data is insufficient.

Intuitively, people can learn new knowledge after being taught just a few instances. Therefore, Han et al. (2018) [27] formulated RC as a few-shot learning (FSL) task, which required models that can handle a classification task with a handful of training instances. They adopted the most recent state-of-the-art few-shot learning methods for RC. Gao et al. (2019) [28] proposed hybrid attention-based prototypical networks for noisy few-shot RC. Many additional efforts have also been devoted to FSL. Caruana (1995) [29], Bengio (2012) [30], and Donahue et al. (2014) [31] used transfer learning methods to fine-tune the pre-trained model. Metric learning methods [32–34] have been proposed for learning the distance distributions among classes. Recently, meta-learning is proposed and encourages models to quickly learn from previous experience and to rapidly generalize to new concepts [35, 36]. However, most of these FSL methods are concentrated on image classification. In contrast to images, the text is diverse and not directly computable; hence, current FSL models cannot be used directly for NLP tasks. In these methods, the prototypical networks [34] are simple and effective. However, we find that not all instances are equal in support set when the prototypical networks are used for relation classification tasks. So, it brings the prototype deviation problem. One of the main tasks of this paper is to generate a satisfactory prototype for a few-shot relation classification task in a text-based support set.

To solve the problem, we propose context attention-based prototypical networks for few-shot RC. The prototypical networks [34] must identify a feature vector from support set as the prototype for each relation and classify the relation between the entity pair in a query instance by measuring the distances between the query instance embedding and the relation prototype. For the prototype representation of each relation, the contribution of each support instance is not equal. Therefore, directly adopting the average vector of all instances in the support set as the relation prototype is not a satisfactory approach. As listed in Table 1, the current relation prototype is the “subsidiary”

Table 1 Main strategy of the context attention mechanism, which is to score instances in the support set

The current prototype in the support set is the subsidiary	
Score	Instance
0.35	Toyota Australia is a subsidiary of Toyota Motor Corporation, which is based in Japan.
0.35	Beijing Enlight Pictures was a subsidiary of Beijing Enlight Media for 100% stake.
0.25	In 2006, Rykodisc was bought by the Warner Music Group.
0.05	The building houses the astrophysics and particle physics sub - departments of the Department of Physics at Oxford University.

The colors indicate the entity types: blue for head entity and red for tail entity

in the support set, which represents the affiliation between companies. In instances 1 and 2, the relation between two entities is an affiliate relation between companies; hence, the score is the highest. In instance 3, the relation between two entities is also an affiliate relation between companies; however, it is not as clear as in instances 1 and 2 and the score is lower. In instance 4, the relation between two entities is an affiliate between schools; hence, the score is the lowest. According to the above description, for the instances in the support set, the diversity of the text will cause prototype deviation. To generate a satisfactory prototype in practice, we propose a method, namely, the context attention mechanism, for determining the prototype of a relation class. The main strategy of the context attention mechanism is to score each instance in the support set according to the importance of the instance to the prototype.

In addition, we also explore the utilization of a pre-trained language model to further improve the performance of the few-shot RC task. In previous works, word embedding tools (e.g., Word2Vec [37] and Glove [38]) have been used to obtain word vectors directly, whereas language models transform words into distributed representations according to context information. Recently, pre-trained language models have performed well in common language representations by using large amounts of unlabelled data (e.g., ELMo [39], OpenAI GPT [40], and BERT [41]). Of these models, bidirectional encoder representations from transformers (BERT) [41] are the most representative. Although BERT has yielded amazing results on eleven natural language processing tasks, it has not yet been explored for the few-shot relation classification task. Thus, we have conducted relevant investigations in this paper. To the best of our knowledge, we are the first to apply the BERT model to the few-shot RC task.

Our main contributions in the paper are as follows:

1) Context attention (CATT) mechanism is proposed, which can effectively alleviate the prototype deviation problem by scoring different instances in support set to indicate the importance of the instance to the prototype. It doesn't take any extra parameters.
2) The application of pre-trained language model BERT in the few-shot RC task is explored. Combining the context attention and the pre-trained language model not only makes our model more efficient but also converges faster.
3) We conduct experiments on a real-world dataset for a few-shot RC task by using our proposed model. The experimental results demonstrate that our model outperforms state-of-the-art models and meets the requirements of the computing paradigms.

The remainder of the paper is arranged as follows. Section 2 introduces the related works of relation classification, few-shot learning and language model. We detail our methodology in Section 3. The experimental results are shown in Section 4. Conclusion and future work are given in Section 5.

1 Related works

Except for a few unsupervised clustering methods [42, 43], most methods [44] on relation classification are based on supervised learning, which is typically cast as a multi-class classification task. Traditional methods often rely on handcrafted features and NLP upstream tasks [44–46]. These methods were limited to specified domains and do not exhibit satisfactory generalization performance.

In recent years, many works have utilized deep learning. Deep neural networks (DNN) have performed well on supervised tasks and been widely used in NLP domains. RC has also benefited from DNN. Zeng et al. (2014) [18] used a convolutional neural networks (CNN) to extract lexical and sentence-level features without complicated pre-processing. To model a sentence with the complete and sequential information of all words, Zhang et al. (2015) [47] combined bidirectional long short-term memory networks (BLSTM) and features that are derived from the lexical resources. Zhou et al. (2016) [48] proposed an attention-based BLSTM for capturing the most important semantic information in a sentence. Wang et al. (2016) [49] proposed a CNN with two levels of attention for this task to better discern patterns in heterogeneous contexts. When the data is insufficient, Mintz et al. (2009) [20] proposed the DS method for constructing large-scale datasets. To alleviate the wrong label problem and capturing structural and other latent information in DS, Zeng et al. (2015) [23] designed piecewise convolutional neural networks (PCNNs) with multi-instance learning. Lin et al. (2016) [24] built sentence-level attention over multiple instances to dynamically reduce the weights of noisy instances. To enhance the robustness of neural networks and improve their generalizability, Wu et al. (2017) [25] applied adversarial training in RC within the multi-instance multi-label learning framework. Feng et al. (2018) [26] utilized reinforcement learning techniques to select high-quality sentences from a sentence bag. These approaches reduce noise in DS by using various techniques; however, they cannot handle long-tail relations in practice.

FSL can generalize to new classes that are not seen during training given only a few instances of each new class. Hence, FSL can also learn high-quality features with insufficient data of a relation class. Many works use transfer learning methods to fine-tune pre-trained models for FSL, which transfer latent information from the common classes with sufficient instances to the uncommon classes with only a few instances [29–31]. Metric learning methods are popular in FSL [50]. For example, Koch et al. (2015) [32] presented a strategy for performing one-shot classification via learning deep convolutional siamese neural networks on the Omniglot dataset [51]. Vinyals et al. (2016) [33] built matching networks for one-shot learning by combining metric learning that is based on deep neural features and the augmentation of neural networks with external memories. Snell et al. (2017) [34] proposed a simple method, namely, prototypical networks, for few-shot learning. Prototypical networks represent each class in terms of examples of the class in a representation space that is learned by a neural network. The meta-learning approach is another relevant FSL method. Ravi et al. (2016) [34] proposed an LSTM-based meta-learner model that learns an

exact optimization algorithm, which is used to train another learner neural network classifier in the FSL. Munkhdalai et al. (2017) [35] proposed a novel meta-learning method, namely, meta-networks, that learns meta-level knowledge across tasks and shifts its inductive biases via fast parameterization for rapid generalization.

Currently, the major FSL methods are focused on image domains, only a few works are devoted to NLP applications. Han et al. (2018) [27] introduced FSL into the RC task and systematically adopt the most recent state-of-the-art FSL methods for RC. To deal with the diversity and noise of few-shot relation classification tasks, Gao et al. (2019) [28] designed instance-level and feature-level attention schemes that are based on prototypical networks for highlighting the crucial instances and features, respectively, thereby significantly improving the performance and robustness of RC models in a noisy FSL scenario. In previous FSL approaches, the prototypical networks [34] are considered effective. The prototype is calculated for each class and query instances are classified by calculating the Euclidean Distance between the prototype and query instances. Therefore, the prototype is highly important in prototypical networks.

In the application of deep neural networks in NLP, word embedding is essential. Word2Vec and Glove have long been popular. Word2Vec is introduced by [37], which is an efficient method for learning high-quality vector representations of words from large amounts of unstructured text data. Pennington et al. (2014) [38] proposed Glove for word representation. Glove is a weighted least-squares model that trains on global word-word co-occurrence counts. However, polysemy cannot be represented in these models. Until recently, the language model is pre-trained on a large network with a large amount of unlabeled data. Many downstream tasks of NLP have been realized by fine-tuning on a pre-trained language model. Peters et al. (2018) [39] proposed the ELMo model, which is a new type of deep contextualized word representation that attempts to address the polysemy and the complex characteristics of word use. ELMo uses a vector that is derived from a bidirectional LSTM that is trained with a coupled language model objective on a large text corpus to represent a word. OpenAI GPT was proposed by Radford(2018) [40], and it combines unsupervised pre-training and supervised fine-tuning methods to understand language. Devlin et al. (2018) [41] proposed a BERT model that is pre-trained on a masked language model task and a next sentence prediction task via a large cross-domain corpus. BERT yields state-of-the-art results for a range of NLP tasks, thereby demonstrating the enormous potential of pre-trained language models.

In this paper, to generate a satisfactory prototype in prototypical networks, we propose the context attention-based prototypical networks. Our solution is to score the instances in the support set via a context attention mechanism to highlight the importance of the instances. Another objective of this paper is to explore the pre-training language model BERT that is used for the few-shot RC task.

2 Methodology

This section introduces the context attention-based prototypical networks in detail. In addition, we also demonstrate the combination of pre-trained language models in our model.

Before we start, we give the notation and the definition. Formally, the few-shot relation classification is designed to obtain a function $\mathcal{F} : (\mathcal{R}, x) \rightarrow y$. This function represents a mapping relation: given a set of relation labels $\mathcal{R}$ and a text instance x, the

predicted relation label$y$is output. Here, $\mathcal{R} = \{r_1, r_2, ..., r_m\}$, (m∈N denotes the number of relations.) defines the relation set into which all instances are classified. In this paper, S is used to represent the support set in few-shot learning:

$$\mathrm{S} = \left\{ \begin{array}{c} \left(x_1^1, r_1\right), \left(x_1^2, r_1\right), ..., \left(x_1^{n_1}, r_1\right), \\ \left(x_2^1, r_2\right), \left(x_2^2, r_2\right), ..., \left(x_2^{n_2}, r_2\right), \\ ..., \\ \left(x_m^1, r_m\right), \left(x_m^2, r_m\right), ..., \left(x_m^{n_m}, r_m\right) \end{array} \right\} \tag{1}$$

which includes n_i instances for each relation $r_i \in \mathcal{R}$, where x_i^j is a sentence instance with a pair of entities, i represents a relation, and j represents an instance in relation i. The query data x is an unlabelled instance to classify. $y \in \mathcal{R}$ is the prediction of x that is given by $\mathcal{F}$.

The N-way K-shot setting is widely adopted to FSL. We also use this setting for the few-shot RC problem, where N is the size of the relation set, and K is the number of instances in each relation set.

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

2.1 Framework

Here, we introduce the main modules of our model. As illustrated in Fig. 1, the model consists of three parts:

(1) Sentence encoder: given a sentence that mentions two entities, we must extract features from the sentence and represent the sentence with a low-dimensional real-valued vector. The sentence encoder consists of an embedding layer and an encoding layer. In this paper, we use a pre-trained language model as the embedding layer and implement the encoding layer with convolutional neural networks.

(2) Prototypical networks: we use prototypical networks to compute a prototype for each relation in the support set. To classify a query instance, we compute the Euclidean Distance between the query instance and each relation prototype and the relation prototype that corresponds to the smallest distance is selected as the predicted relation of the query instance.

(3) Context attention: to further enhance the RC performance and the convergence speed, we propose the context attention-based prototypical networks. The main strategy of the context attention mechanism is to score instances in a support set.

First, the sentence encoder is used to obtain the vectorized representation of each sentence. Then, the relation prototype is generated by the context attention. Finally, the prototypical networks are used to classify the relation between entities.

2.2 Sentence encoder

For a sentence $\mathrm{x} = \{w_1, w_2, ..., w_n\}$ that mentions two entities, we use a pre-trained language model, namely, BERT, to embed each word. Then, CNN is used to encode these embedded word vectors into a continuous low-dimensional vector as the sentence vector.

2.2.1 Embedding layer

The main function of the embedding layer is to map words in the instance to continuous input embeddings. In general, we use a trained tool directly as word embeddings,

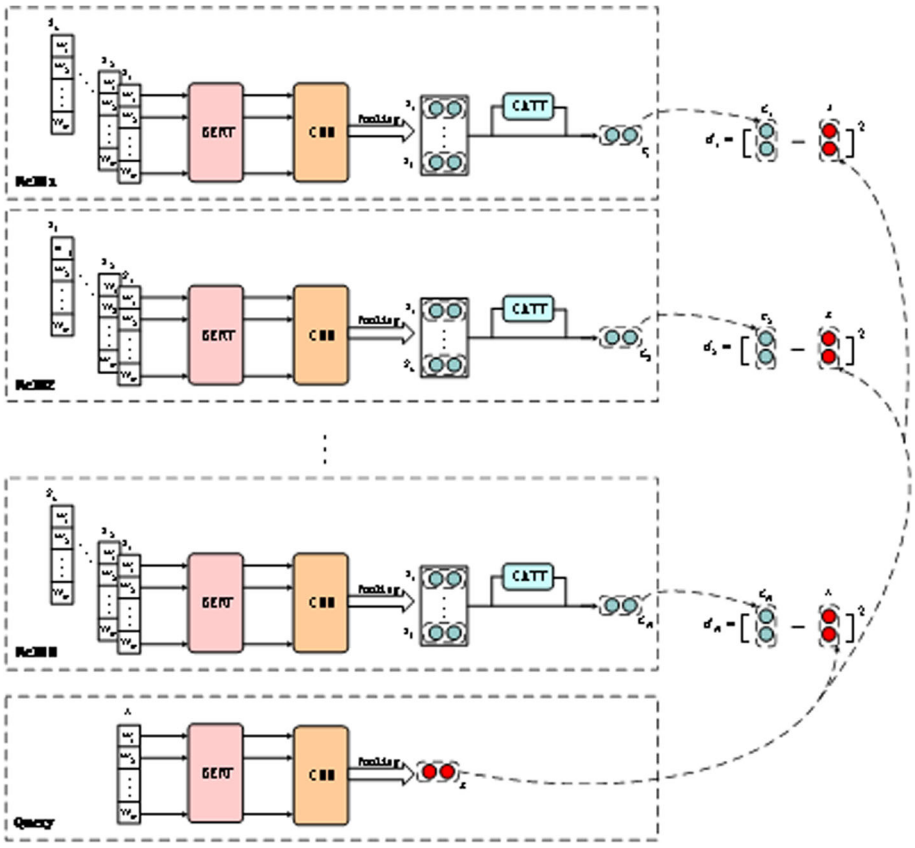

Fig. 1 Architecture of our proposed model. Shows the model structure proposed in the paper, which is mainly composed of three parts, namely, the sentence encoder, prototypical networks, and context attention

such as Word2Vec [37] and Glove [38]. However, polysemy cannot be represented using these static models. In our model, we use $BERT_{BASE}$ [39] as the embedding layer. In BERT, to more effectively represent the semantic information of a word, its context is combined. Therefore, the distributed representations of a word can differ among sentences.

To highlight the entities in a sentence, we use entity indicators [52]. Given a sentence $x = \{w_1, w_2, \ldots, w_n\}$ with four marked indicators of entity position, we encode each word w_i in the sentence to a real-valued embedding $\boldsymbol{e}_i \in \mathbb{R}^{d_w}$ to express semantic and syntactic meanings of the word via $\mathrm{BERT}_{\mathrm{BASE}}$.

2.2.2 Encoding layer

The encoding layer extracts features from the word vector $\boldsymbol{e}_i \in \mathbb{R}^{d_w}$, which are used to construct a sentence feature vector. Recurrent neural networks (RNN) and the convolutional neural networks (CNN) are both widely used in deep neural networks (DNN). In this paper, to be consistent with the previous methods and to facilitate the comparison of the following experiments, we use a CNN to extract sentence features.

A CNN slides a convolution kernel with the window size of m over the word vector $\{\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_n\}$ to obtain the d_h-dimensional hidden embeddings,

$$\boldsymbol{h}_i = \mathrm{CNN}\left(\boldsymbol{e}_{i-\frac{m-1}{2}}, \ldots, \boldsymbol{e}_{i+\frac{m-1}{2}}\right) \tag{3}$$

where CNN(.) is a convolution operation.

To output the final instance embeddings, a max-pooling operation is applied over these hidden embeddings,

$$[\boldsymbol{s}]_j = \max\left([\boldsymbol{h}_1]_j, [\boldsymbol{h}_2]_j, \ldots, [\boldsymbol{h}_n]_j\right) \tag{4}$$

where $[.]_j$ is the jth value of the specified vector.

We express an instance encoding operation, which includes both the embedding and encoding layers, as the following equation:

$$\boldsymbol{s} = f_\phi(x) \tag{5}$$

where ϕ denotes the learnable parameters of the instance encoding. f is a function, it is a scalar. x is an instance of a sentence, also a scalar. $\boldsymbol{s}$ is the embedded vector of the output.

2.3 Prototypical networks

The prototypical networks [32] are few-shot classification models that assume that for each class there exists a prototype that represents a relation. The prototype is computed by averaging all the instance embeddings $\boldsymbol{S}$ in the support set for each relation

$$c_i = \frac{1}{n_i}\sum\nolimits_{j=1}^{n_i} s_i^j \left(s_i^j \in S\right) \tag{6}$$

where $\boldsymbol{c}_i$ is the prototype that is computed for a relation r_i; $\boldsymbol{s}_i^j$ is the embedded vector of instance j in the support set relation r_i, it is a low-dimensional real-value vector that represents the vectorized form of each text sentence; and n_i denotes the number of instances in a relation r_i in the support set.

Then, we can compute the probabilities of the relations in $\mathcal{R}$ for a query instance x as follows:

$$p_\phi(y = r_i|x) = \frac{\exp\left(-d\left(f_\phi(x), \boldsymbol{c}_i\right)\right)}{\sum_{j=1}^{|\mathcal{R}|}\exp\left(-d\left(f_\phi(x), \boldsymbol{c}_j\right)\right)} \tag{7}$$

where $d(.,.)$ is the distance function for two specified vectors, the prototypical networks [34] adopt the Euclidean distance.

2.4 Context attention

In the prototypical networks [34], each relation prototype is determined by the average vector of all instances. However, in practice, the meaning of a relation is rich, namely, a relation can express multiple meanings. In a support set, not all instances express the same relational meaning. Therefore, the prototype that is produced via the vector averaging approach is not a satisfactory prototype. Vector averaging of all instances in the support set results in the prototype deviation problem.

We argue that not all instances are of equal importance in a support set. To determine a satisfactory prototype, we propose a context attention approach that focuses more attention on prototype-related instances. To represent the correlation between

instances $\boldsymbol{S}$ in a support set, we calculate a matrix product between instances, divide each by $\sqrt{d_w}$, and apply a softmax function to obtain the weights between instances. The final instance $\boldsymbol{S}_{\text{new}}$ is obtained via another matrix multiplication between the weights and the instances. The equation is as follows:

$$\boldsymbol{S}_{\text{new}} = \text{CATT}(\boldsymbol{S}) = \text{softmax}\left(\frac{ss^T}{\sqrt{d_w}}\right)\boldsymbol{S} \tag{8}$$

The meaning of equation (8) is the new embedded vector $\boldsymbol{S}_{\text{new}}$ obtained by using context attention($CATT$) on embedded instance $\boldsymbol{S}$. The exact calculation of CATT is determined by the softmax function that follows. Now, the prototype is obtained by the following equation:

$$\boldsymbol{c}_i = \frac{1}{n_i}\sum\nolimits_{j=1}^{n_i} \boldsymbol{s}_i^j \left(\boldsymbol{s}_i^j \in \boldsymbol{S}_{\text{new}}\right) \tag{9}$$

To make better use of the features in instances, we use multi-head attention [53] in our model. The equation is as follows:

$$\text{MultiHead}(\boldsymbol{S}, \boldsymbol{S}, \boldsymbol{S}) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h) \tag{10}$$

where head_i=Attention($\boldsymbol{S}_{d_m}, \boldsymbol{S}_{d_m}, \boldsymbol{S}_{d_m}$)

In this work we employ h = 12 parallel attention heads, the dimension of each head is $d_m = \frac{d_w}{h}$.

The proposed context attention mechanism can assign to each instance a weight corresponding to their contribution for the current relation prototype. Therefore, our framework can avoid the prototype deviation caused by the average instance embeddings.

3 Experiments

This section evaluates the performance of our model on a real dataset in terms of the accuracy rate and the convergence speed. We will also analyze the roles of the context attention mechanism and the pre-trained language model in several cases.

3.1 Datasets and parameter settings

We evaluate our models on the FewRel dataset in this paper, which is developed by Han [27]. The FewRel dataset consists of 100 relations, each of which has 700 instances. It has 64 relations for training, 16 relations for validation and 20 relations for testing. There are no overlapping relations among the training, validation and test sets. Since the test set is not available directly, we evaluate our models on the training and validation sets. To evaluate the performance of our model, we conduct two sets of control experiments: a comparison between our model and previous models and an analysis of the influences of the modules in our model.

All the hyperparameters are listed in Table 2. For the input, we set the maximum length of a sentence to 64. Limited by the performance of our machine, the batch size is set to 1 and the number of training classes for each batch is set to 8. The learning rate is set to 2E−5. We set the number of training iterations to 10000 to yield the optimal result. The convolution window size is set to 3. In the CNN operation, the dimension of the hidden layer is consistent with the dimension of the word embeddings,

Table 2 Parameter settings

Max length of a sentence	64
Batch size	1
Training classes for one batch	8
Learning rate	2e-5
Train iterations	10000
Convolutional window size	3
Hidden layer dimension d_h	768
Number of multihead	12

which is set to 768. In the multi-head, the number of heads is set to 12. All models are trained on the training set and compared in terms of accuracy on the validation set; instances in the validation set are not used in the training process.

3.2 Overall evaluation results

Before we discuss the results, it should be noted that the metric adopted in this paper is accuracy. Accuracy is one metric for evaluating classification models. Informally, accuracy is the fraction of predictions our model got right. Formally, accuracy has the following definition:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \times 100\% \tag{11}$$

We compare the models in terms of accuracy in Table 3. CNN in the model name indicates that the convolutional neural networks are adapted for feature extraction in the encoding layers of these models. In this paper, the proposed model is denoted as Proto_CATT_BERT(CNN), which indicates that our model is composed of context attention-based prototypical networks and that the BERT is used as a pre-trained language model in the embedding layer of the model. Model Proto_HATT(CNN) is proposed by [28] and uses hybrid attention-based methods to solve noisy few-shot RC tasks. The other models (Meta Network (CNN), GNN(CNN), SNAIL(CNN), and Prototypical Networks(CNN)) are provided by Han [15], which are all current state-of-the-art FSL models. According to the table, our model, namely, Proto_CATT_BERT, outperforms the others on several N-way K-shot tasks. The values of other models in the table above are the results that are obtained by retraining on the training set and testing in the validation set according to the source codes that are provided in the related papers. In the 5-way 5-shot task, five relations need to be distinguished, and each relation type has only five instances, which is in line with the application scenario with the few-shot learning. The accuracy of our proposed model is 94.86%, which is 7.6% higher

Table 3 Accuracy comparison among models (%)

Model	5 way 5 shot	5 way 10 shot	10 way 5 shot	10 way 10 shot
Meta network(CNN)	80.03 ±0.52	82.96 ± 0.50	70.31 ± 0.48	73.03 ± 0.44
GNN(CNN)	77.75 ± 0.44	80.56 ± 0.38	66.02 ± 0.40	69.30 ± 0.42
SNAIL(CNN)	80.57 ± 0.24	81.62 ± 0.21	68.03 ± 0.22	71.32± 0.20
Prototypical networks(CNN)	85.57 ± 0.14	88.17 ± 0.10	75.01 ± 0.16	78.50 ± 0.11
Proto_HATT(CNN)	87.23 ± 0.08	89.53 ± 0.06	77.45 ± 0.06	80.98 ± 0.08
Proto_CATT_BERT(CNN)	**94.86** ± 0.04	**95.74** ± 0.05	**90.01** ±0.04	**91.60** ± 0.03

than the model Proto_HATT(CNN). In other N-way K-shot tasks, our model is far superior to other models.

To evaluate the effects of the modules in our model, we report the results in Table 4. According to Table 4, adding the context attention (CATT) mechanism directly to the prototypical networks can improve the accuracy of the model, namely, the Proto_CATT(CNN) model outperforms the prototypical networks(CNN) model. This demonstrates that the CATT mechanism can improve the performance of the few-shot RC model by scoring instances to generate a satisfactory prototype for each relation. According to the first and third rows of Table 4, the accuracy of the Proto_BERT(fine-tuning) model is 91.86%, and that of the Prototypical Networks(CNN) model is 85.57%, more than 6.3%. This indicates that BERT can further improve the accuracy of the task. In addition, the accuracy of the Proto_BERT(CNN) model exceeds that of the Proto_BERT(fine-tuning) model. We conclude that the model that is built by adding a layer of CNN after BERT outperforms the result of fine-tuning on BERT. Therefore, the pre-trained language model is also effective on few-shot RC tasks. In other N-way K-shot tasks, BERT, and CATT modules also outperformed other modules.

3.3 Convergence speed

We compare the convergence speeds of the models to explore the efficiency of these models in terms of time, as shown in Figs. 2 and 3. According to these figures, the Proto_CATT model that uses CATT outperforms the baseline model proto in terms of the speeds of both loss decrease and accuracy increase. By adding the pre-trained language model, namely, BERT, the model converges faster. By adding CATT to the original prototypical networks, the prototype deviation can be alleviated in the support set. When classifying query instances, the accuracy is higher and the loss is lower; hence, the convergence is faster. The pre-trained language model is obtained after training on a large corpus. It can directly represent the vector distribution of words or sentences. Therefore, initially, the accuracy will be very high, thereby rendering the convergence faster after iterations. Finally, CATT and HATT [28] converge at the same rate. However, according to Eqs. 7 and 8, it can be concluded that CATT does not need additional parameters compared with HATT [28].

3.4 Result analysis

To further evaluate the roles of the modules, this section analyses the impacts of the context attention mechanism and the pre-trained language model on the network in special cases.

3.4.1 Effect of context attention

Via examples, we find that our model can produce a satisfactory prototype, whereas the original prototypical networks produce a poor prototype. In Fig. 4, marker "x" corresponds to

Table 4 Accuracy comparison among modules (%)

Model	5 way 5 shot	5 way 10 shot	10 way 5 shot	10 way 10 shot
Prototypical networks(CNN)	85.57 ± 0.14	88.17 ± 0.10	75.01 ± 0.16	78.50 ±0.11
Proto_CATT(CNN)	87.48 ± 0.12	89.28 ± 0.08	77.46 ±0.13	80.39 ± 0.14
Proto_BERT(fine-tuning)	91.86 ± 0.02	93.34 ± 0.04	85.44 ± 0.03	87.51 ± 0.04
Proto_BERT(CNN)	93.54 ± 0.05	94.68 ± 0.04	88.85 ± 0.06	90.18 ± 0.05
Proto_CATT_BERT(CNN)	**94.86** ± 0.04	**95.74** ± 0.05	**90.01** ± 0.04	**91.60** ± 0.03

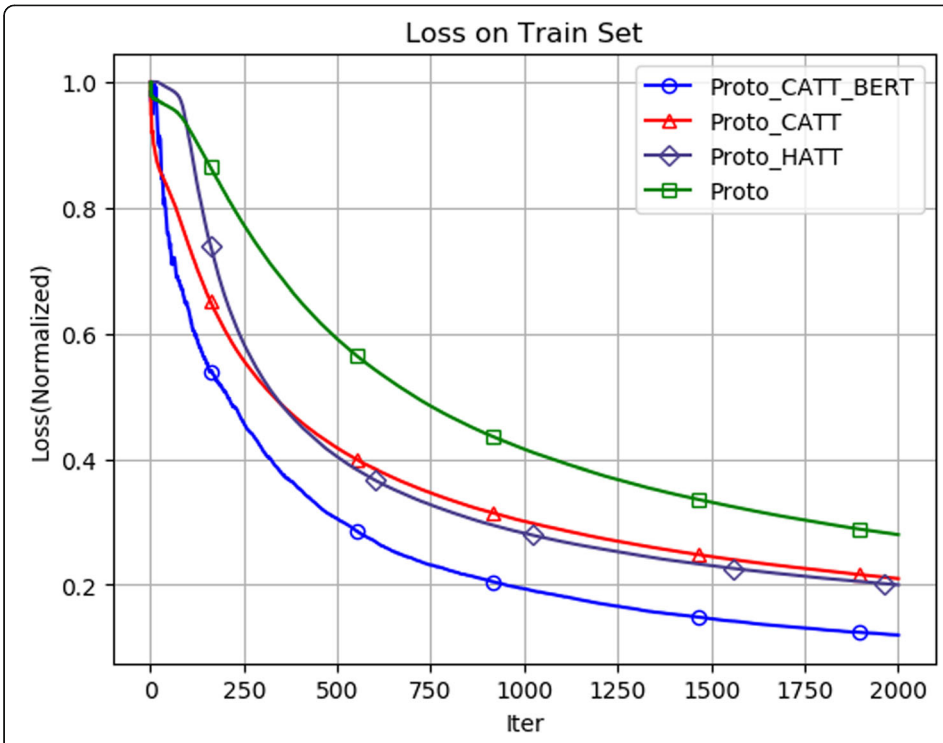

Fig. 2 Losses of models on the training set.. Shows the change curves of different models during the training phase

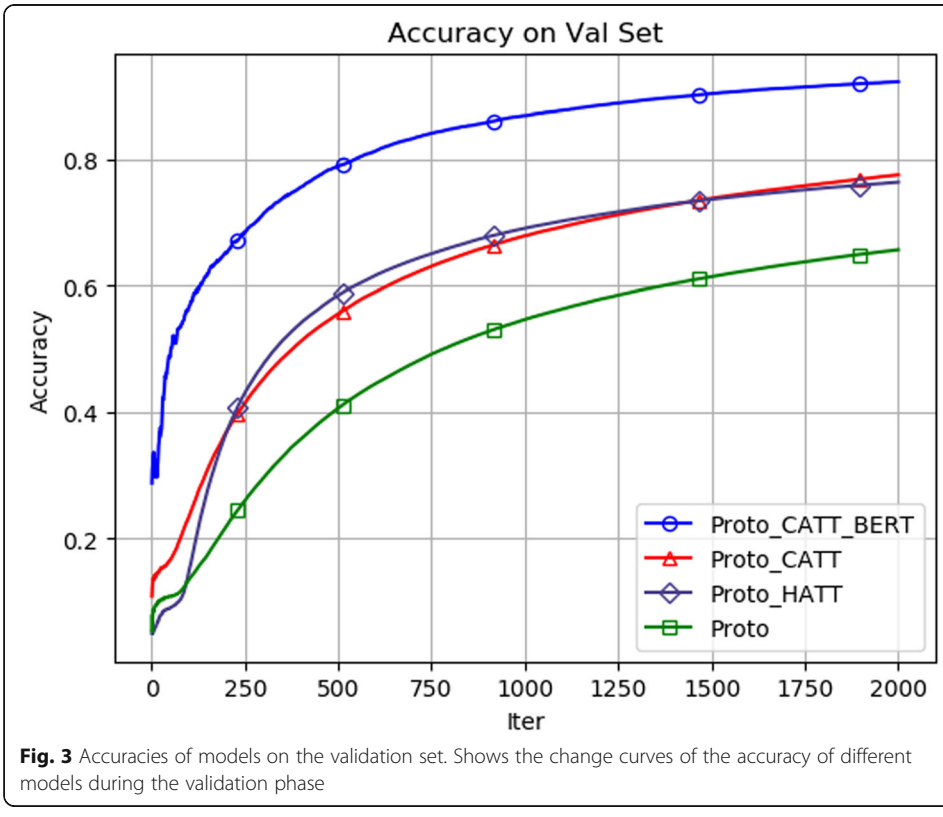

Fig. 3 Accuracies of models on the validation set. Shows the change curves of the accuracy of different models during the validation phase

the "part of" relation prototype and solid circle corresponds to 40 query instances. Because the prototypical networks are kind of metric models, the results of model depend on the distance between the query instance and the prototype. Therefore, the smaller the distance, the better the model performance. According to Fig. 4a, b, the prototype that is generated with CATT is more accurate than the prototype that is generated without CATT, which has deviated. The CATT can select instances with high correlation with the relation prototype and reduce the influence of those with low correlation. Hence, The CATT can facilitate the identification of a satisfactory prototype by networks and improve the performance of the model.

3.4.2 Effect of pre-trained language model

To evaluate the effect of the pre-trained language model, we select two relations from the validation set, namely, constellation and sport, which have 60 instances per relation. Our model encodes all instances to obtain instance feature vectors of dimension d_w. Then, we

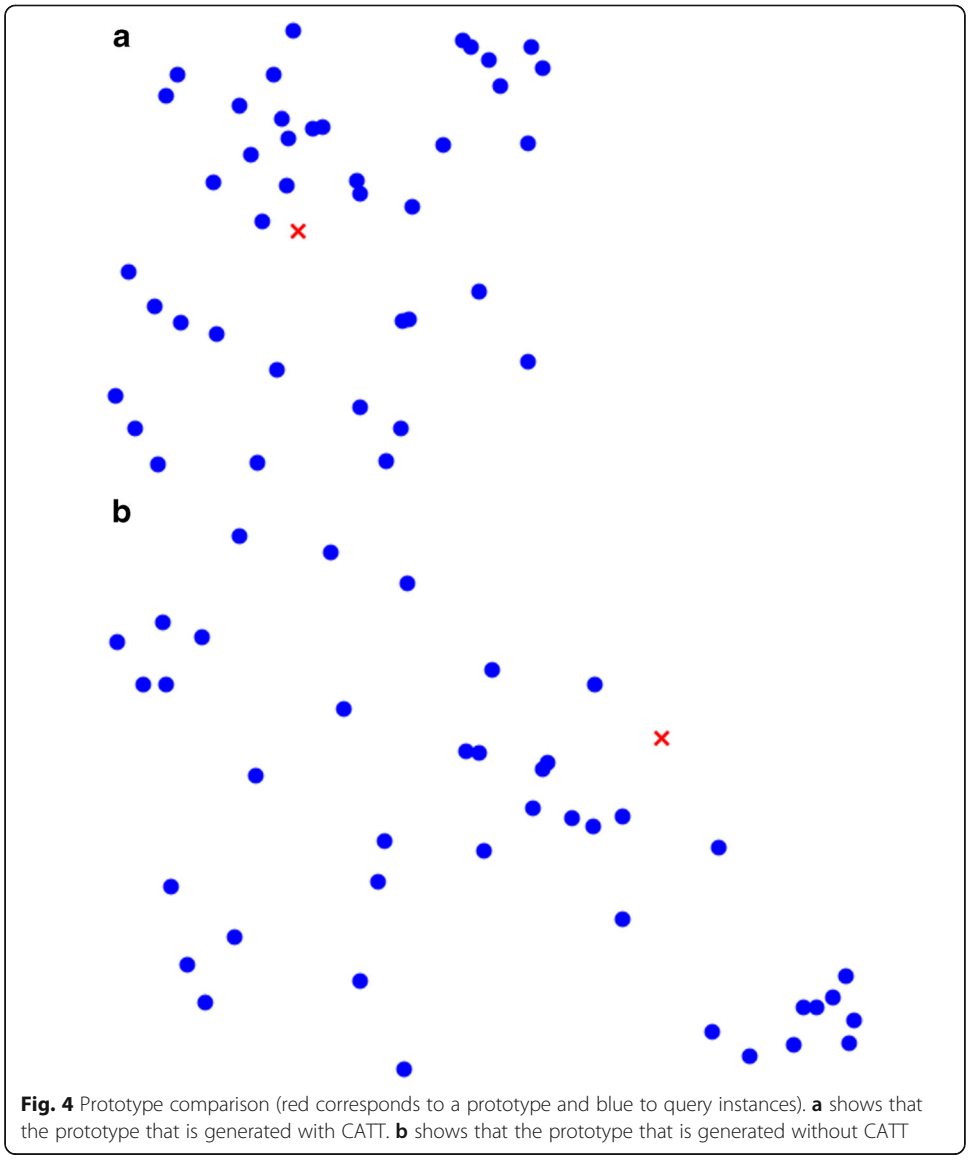

Fig. 4 Prototype comparison (red corresponds to a prototype and blue to query instances). **a** shows that the prototype that is generated with CATT. **b** shows that the prototype that is generated without CATT

map them to 2D points by using principal component analysis (PCA). Comparing the two plots in Fig. 5a, b, the solid box and marker "+" indicate two relations, respectively. Instances that are embedded with BERT are easier to classify. Since RC is a kind of classification tasks, the model whose results are more easily linearly separable performs better. Hence, BERT can help encoders learn embeddings that improve the performance of the model.

4 Conclusions and future work

In this paper, we propose context attention-based prototypical networks for few-shot relation classification tasks. The main strategy of the context attention mechanism is to assign weights to instances to highlight the importance of instances under relation prototypes, which can generate a satisfactory prototype to alleviate the prototype deviation problem. In addition, we explore how the pre-trained language model can be used in the few-shot RC task. We evaluate our model on a real dataset. The experimental results demonstrate that our model can increase the accuracy and the convergence speed on the RC task. In the future, we will explore whether it is possible to map a relation prototype to another vector

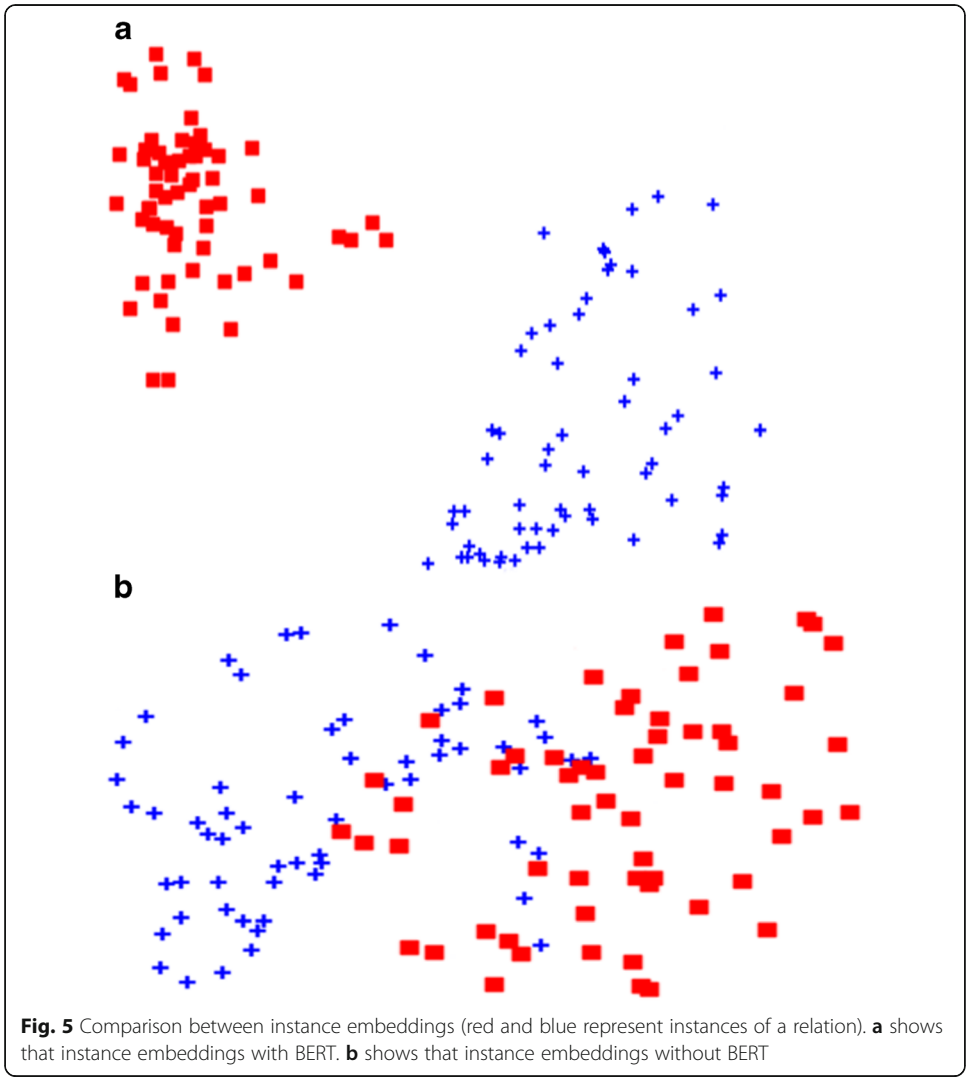

Fig. 5 Comparison between instance embeddings (red and blue represent instances of a relation). **a** shows that instance embeddings with BERT. **b** shows that instance embeddings without BERT

space by using only one projection of that vector as the relation prototype, to solve problems in which relation has multiple meanings.

Abbreviations

AI: Artificial intelligence; NER: Named entity recognition; RC: Relation classification; NLP: Natural language processing; DS: Distant supervision; KGs: Knowledge graphs; FSL: Few-shot learning; BERT: Bidirectional encoder representations from transformers; CATT: Context attention; DNN: Deep neural networks; CNN: Convolutional neural networks; BLSTM: Bidirectional long short-term memory networks; PCNNs: Piecewise convolutional neural networks

Author details

School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China.

Authors' contributions

Hui is the main author of this paper and has done most of the research. Liu was responsible for the experiment design and implementation. Chen participated in the experimental design and wrote the result analysis part . Zhou and Nian collated and analyzed experimental data. All authors read and approved the final manuscript.

Funding

This work is supported by National Key R&D Program of China (No. 2018YFC0807500).

Competing interests

The authors declare that they have no competing interests.

Received: 27 January 2020 Accepted: 23 April 2020
Published online: 08 June 2020

References

1. C. Kong, G. Luo, L. Tian, X. Cao, Disseminating authorized content via data analysis in opportunistic social networks. Big Data Mining and Analytics **2**(1), 12–24 (2019)
2. Xu Zheng, and Zhipeng Cai. Privacy-preserved data sharing towards multiple parties in industrial IoTs. IEEE Journal on Selected Areas in Communications (JSAC). Accepted.
3. S. Kumar, M. Singh, Big data analytics for healthcare industry: impact, applications, and tools. Big Data Mining and Analytics **2**(1), 48–C57 (2019)
4. Lei Yu, Liuhua Chen, Zhipeng Cai, Haiying Shen, Yi Liang and Yi Pan. Stochastic load balancing for virtual resource management in datacenters. IEEE Transactions on Cloud Computing.
5. L. Qi, X. Zhang, W. Dou, C. Hu, C. Yang, J. Chen, A two-stage locality-sensitive hashing based approach for privacy-preserving mobile service recommendation in cross-platform edge environment. Futur. Gener. Comput. Syst. **88**, 636–643 (2018)
6. Lianyong Qi, Yi Chen, Yuan Yuan, Shucun Fu, Xuyun Zhang, Xiaolong Xu. A QoS-aware virtual machine scheduling method for energy conservation in cloud-based cyber-physical systems. World Wide Web Journal, 2019. DOI: https://doi.org/10.1007/s11280-019-00684-y.
7. Wenwen Gong, Lianyong Qi, Yanwei Xu. Privacy-aware multidimensional mobile service quality prediction and recommendation in distributed fog environment. Wireless Communications and Mobile Computing, vol. 2018, Article ID 3075849, 8 pages, 2018.
8. Lei Yu and Zhipeng Cai*. Dynamic scaling of virtualized networks with bandwidth guarantees in cloud datacenters. The 35th Annual IEEE International Conference on Computer Communications (INFOCOM 2016).
9. X. Chen, C. Li, D. Wang, S. Wen, J. Zhang, S. Nepal, X. Yang, K. Ren, Android HIV: a study of repackaging malware for evading machine-learning detection. IEEE Transactions on Information Forensics and Security **15**(1), 987–1001 (2020)
10. Zhipeng Cai and Xu Zheng. A Private and Efficient Mechanism for Data Uploading in Smart Cyber-Physical Systems. IEEE Transactions on Network Science and Engineering (TNSE) . Accepted.
11. I. Stojmenovic, S. Wen, X. Huang, et al., An overview of fog computing and its security issues[J]. Concurrency and Computation: Practice and Experience **28**(10), 2991–3005 (2016)
12. J. Jiang, S. Wen, S. Yu, X. Yang, W. Zhou, Identifying Propagation Sources in Networks: State-of-the-Art and Comparative Studies. IEEE Communications Surveys and Tutorials **19**(1), 465–481 (2017)
13. X. Yao and B. Van Durme. Information extraction over structured data: Question answering with freebase. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), volume 1, pages 956–966, 2014.
14. J. Chen, N. Tandon, and G. de Melo. Neural word representations from large-scale commonsense knowledge. In 2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), volume 1, pages 225–228. IEEE, 2015.
15. D. Zelenko, C. Aone, A. Richardella, Kernel methods for relation extraction. J. Mach. Learn. Res. **3**(Feb), 1083–1106 (2003)
16. R. J. Mooney and R. C. Bunescu. Subsequence kernels for relation extraction. In Advances in neural information processing systems, pages 171–178, 2006.
17. T. Wu, S. Wen, Y. Xiang, et al., Twitter spam detection: Survey of new approaches and comparative study[J]. Computers & Security **76**, 265–284 (2018)
18. D. Zeng, K. Liu, S. Lai, G. Zhou, and J. Zhao. Relation classification via convolutional deep neural network. In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pages 2335–2344, Dublin, Ireland, Aug. 2014. Dublin City University and Association for Computational Linguistics.

19. Hanwen Liu, Huaizhen Kou, Chao Yan, Lianyong Qi. Link prediction in paper citation network to construct paper correlated graph. EURASIP Journal on Wireless Communications and Networking, 2019.DOI: https://doi.org/10.1186/s13638-019-1561-7.
20. M. Mintz, S. Bills, R. Snow, and D. Jurafsky. Distant supervision for relation extraction without labeled data. In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2, pages 1003–1011. Association for Computational Linguistics, 2009.
21. S. Riedel, L. Yao, and A. McCallum. Modeling relations and their mentions without labeled text. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pages 148–163. Springer, 2010.
22. M. Surdeanu, J. Tibshirani, R. Nallapati, and C. D. Manning. Multiinstance multi-label learning for relation extraction. In Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning, pages 455–465. Association for Computational Linguistics, 2012.
23. D. Zeng, K. Liu, Y. Chen, and J. Zhao. Distant supervision for relation extraction via piecewise convolutional neural networks. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 1753–1762, 2015.
24. Y. Lin, S. Shen, Z. Liu, H. Luan, and M. Sun. Neural relation extraction with selective attention over instances. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), volume 1, pages 2124–2133, 2016.
25. Y. Wu, D. Bamman, and S. Russell. Adversarial training for relation extraction. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 1778–1783, 2017.
26. J. Feng, M. Huang, L. Zhao, Y. Yang, and X. Zhu. Reinforcement learning for relation classification from noisy data. In Proceedings of the ThirtySecond AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018, pages 5779–5786, 2018.
27. X. Han, H. Zhu, P. Yu, Z. Wang, Y. Yao, Z. Liu, and M. Sun. FewRel: A large-scale supervised few-shot relation classification dataset with stateof-the-art evaluation. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 4803–4809, Brussels, Belgium, Oct.-Nov. 2018. Association for Computational Linguistics.
28. T. Gao, X. Han, Z. Liu, and M. Sun. Hybrid attention-based prototypical networks for noisy few-shot relation classification. In The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The ThirtyFirst Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019., pages 6407–6414, 2019.
29. R. Caruana. Learning many related tasks at the same time with backpropagation. In Advances in neural information processing systems, pages 657– 664, 1995.
30. Y. Bengio. Deep learning of representations for unsupervised and transfer learning. In Proceedings of ICML Workshop on Unsupervised and Transfer Learning, pages 17–36, 2012.
31. J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In International conference on machine learning, pages 647–655, 2014.
32. G. Koch, R. Zemel, and R. Salakhutdinov. Siamese neural networks for one-shot image recognition. In ICML deep learning workshop, volume 2, 2015.
33. O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al. Matching networks for one shot learning. In Advances in neural information processing systems, pages 3630–3638, 2016.
34. J. Snell, K. Swersky, and R. Zemel. Prototypical networks for few-shot learning. In Advances in Neural Information Processing Systems, pages 4077–4087, 2017.
35. S. Ravi and H. Larochelle. Optimization as a model for few-shot learning. In 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, 2017.
36. T. Munkhdalai and H. Yu. Meta networks. In Proceedings of the 34th International Conference on Machine Learning-Volume 70, pages 2554– 2563. JMLR. org, 2017.
37. T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781, 2013.
38. J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pages 1532–1543, 2014.
39. M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. arXiv preprint arXiv:1802.05365, 2018.
40. A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving language understanding by generative pre-training. URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf, 2018.
41. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.
42. T. Hasegawa, S. Sekine, and R. Grishman. Discovering relations among named entities from large corpora. In Proceedings of the 42nd annual meeting on association for computational linguistics, page 415. Association for Computational Linguistics, 2004.
43. J. Chen, D. Ji, C. L. Tan, and Z. Niu. Unsupervised feature selection for relation extraction. In Companion Volume to the Proceedings of Conference including Posters/Demos and tutorial abstracts, 2005.
44. N. Kambhatla. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In Proceedings of the ACL 2004 on Interactive poster and demonstration sessions, page 22. Association for Computational Linguistics, 2004.
45. Z. GuoDong, S. Jian, Z. Jie, and Z. Min. Exploring various knowledge in relation extraction. In Proceedings of the 43rd annual meeting on association for computational linguistics, pages 427–434. Association for Computational Linguistics, 2005.
46. F. M. Suchanek, G. Ifrim, and G. Weikum. Combining linguistic and statistical analysis to extract relations from web documents. In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 712–717. ACM, 2006.

47. S. Zhang, D. Zheng, X. Hu, and M. Yang. Bidirectional long short-term memory networks for relation classification. In Proceedings of the 29th Pacific Asia conference on language, information and computation, pages 73–78, 2015.
48. P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao, and B. Xu. Attention-based bidirectional long short-term memory networks for relation classification. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), volume 2, pages 207–212, 2016.
49. L. Wang, Z. Cao, G. de Melo, and Z. Liu. Relation classification via multilevel attention cnns. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers, 2016.
50. A. Bellet, A. Habrard, and M. Sebban. A survey on metric learning for feature vectors and structured data. arXiv preprint arXiv:1306.6709, 2013.
51. B.M. Lake, R. Salakhutdinov, J.B. Tenenbaum, Human-level concept learning through probabilistic program induction. Science **350**(6266), 1332–1338 (2015)
52. I. Hendrickx, S. N. Kim, Z. Kozareva, P. Nakov, D. O Séaghdha, S. Padó, M. Pennacchiotti, L. Romano, and S. Szpakowicz. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions, pages 94–99. Association for Computational Linguistics, 2009.
53. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In Advances in neural information processing systems, pages 5998–6008, 2017.

Publisher's Note