Classifying Relation via Piecewise Convolutional Neural Networks with Transfer Learning

Yuting Han[1], Zheng Zhou[1], Haonan Li[2], Guoyin Wang[1*], Wei Deng[1], and Zhixing Li[1]

[1] Chongqing Key Laboratory of Computational Intelligence, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
[2] The University of Melbourne, Victoria, Australia
{wanggy}@cqupt.edu.cn

Abstract. Relation classification is an important semantic processing task in natural language processing (NLP). Traditional works on relation classification are primarily based on supervised methods and distant supervision which rely on the large number of labels. However, these existing methods inevitably suffer from wrong labeling problem and may not perform well in resource-poor domains. We thus utilize transfer learning methods on relation classification to enable relation classification system to adapt resource-poor domains along with different relation type. In this paper, we exploit a convolutional neural network to extract lexical and syntactic features and apply transfer learning approaches for transferring the parameters of convolutional layer pre-training on general-domain corpus. The experimental results on real-world datasets demonstrate that our approach is effective and outperforms several competitive baseline methods.

Keywords: Relation classification · Piecewise convolution neural networks · Transfer learning.

1 Introduction

Relation classification is a crucial NLP task and plays a key role in various domains, e.g., information extraction, question answering etc. The task of relation classification is to classify semantic relations between pairs of marked entities in given texts and can be defined as follows: given a sentence S with the annotated pairs of entities e_1 and e_2, we aim to identify the relations between e_1 and e_2. For instance, in the sentence *"Mental* $[illness]e_1$ *is one of the biggest causes of personal* $[unhappiness]e_2$ *in our society"*, the entities *illness* and *unhappiness* are of relation *Cause-Effect*(e_1, e_2).

Most existing supervised relation classification approaches rely on large amount of labeled relation-specific training data, requiring employing human to annotate the free text with information, which is time consuming and labor intensive. The erroneous annotated relation type may hurt the accuracy of results. Furthermore, the trained classifier needs readjustment when adapting

the new relationship mention. Different from supervised approaches, methods based on distant supervision [1] have been developed to alleviate the costly human annotation. They assume that any sentence that contains a pair of entities expresses a relation in knowledge bases (KBs), then all sentences which contain these entity-pairs will express corresponding relation. For example, $(Donald\_Trump, own, TrumpPlaza)$ is the relation fact in KB. Distant supervision will regard all sentences containing this entity-pair as the relation of own. This is much easier to automatically label training data and mainly relies on multi-instance to solve the noisy data problem. However, when people intend to obtain some new relation classification in source-poor domain, distantly supervised methods are insufficient to handle the situation.

Transfer learning is an appropriate idea to utilize the knowledge learned from resource-rich domains to transfer to resource-poor target domains with new relation type. Traditional transfer learning methods [2] can be classified into four types: instance-transfer, feature representation-transfer, parameter-transfer relational and knowledge-transfer. Parameter-transfer is adopted in this work to initialize the model parameters in the target domain. We assume that different relation types in the source and target domains share same features spaces, some words and common syntactic structures. Table 1 shows these examples. Actually, the target domain learned some distributions from source domain by transferring the parameters. The invariance of CNN structures are these distributions determined by parameter values.

Table 1. Examples of similar syntactic structures across different domain and relation types. The first and second entities are shown in italic and bold, respectively.

Syntactic Pattern	Domains	Relation Instance	Relation Type
arg-1 of arg-2	Source domain	A *herd* is a large group of **animals**	**Memer-Collection**
		The *chapter* of the **book**	**Component-Whole**
	Target domain	*Leader* of a minority **government**	**Employ-Executive**
		the youngest *son* of **Suharto**	**Person-Family**

In this paper, we propose a syntax transferring piecewise convolutional neural networks (ST-PCNN) for resource-poor domains relation classification. The architecture is illustrated in Fig.1. The main contributions of our work can be summarized as follows: 1) To address the relation classification in resource-poor domains, we apply transfer learning approaches for transferring the parameters of model learned in one domain to new domains. 2) We explore the convolutional neural network to extract lexical and syntactic features which make full use of context information compared with the existing neural relation classification model.The experimental results show that our model achieves significant improvements in relation classification.

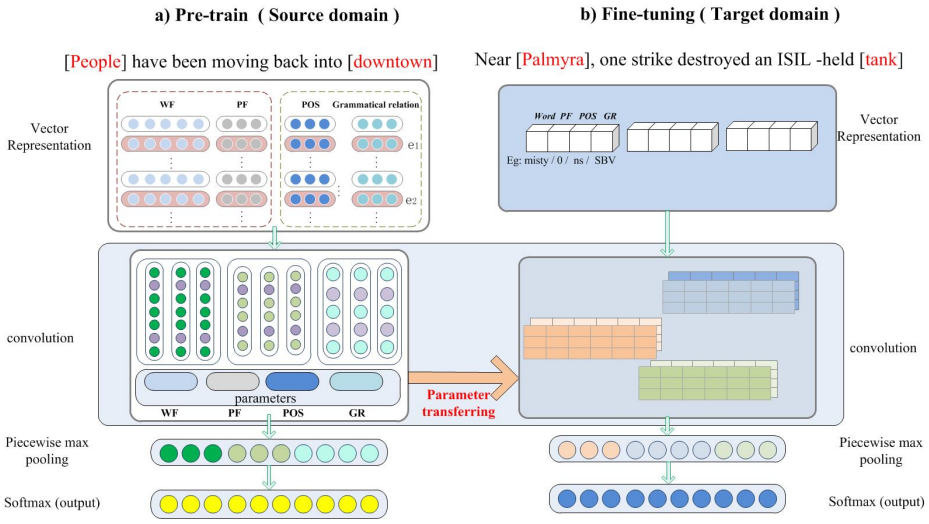

Fig. 1. The architecture of ST-PCNN

2 Related Work

Relation classification is a crucial task in the NLP community. Different existing approaches have been widely utilized for relation classification, such as unsupervised relation discovery and supervised classification. In the unsupervised paradigms, contextual information between entities is used as semantic relation of all entity pairs to make clustering effectively [3]. In the supervised paradigm, relation classification is considered as a multi-classification problem and feature-based, kernel-based methods are proposed to deal with the issue [4, 5]. However, a comprehensible drawback of supervised method is the performance particularly depends on the quality of acquire features which are generated by NLP tools or human designed. Recent years, many researchers use deep neural networks to learn the features automatically [6, 7]. These methods still suffer from a lack of sufficient labeled-data for training. Thus the distant supervision (DS) method was presented. Contrarily, the DS method is confronted with the problem of wrong labeling and noisy data, which substantially hurt the results of classification. Attention mechanism and adversarial training are proposed to alleviate the noisy problem. Furthermore, some researchers applied the transfer learning approaches for the resource-poor domains relation classification [8].

3 Methodology

Given a static source task T_S related to relation extraction, target task T_T for relation classification with $T_S \neq T_T$, and a set of sentences $\{s_1, s_2, , s_n\}$ on T_T with annotated head and tail entities e_1, e_2, we would classify each relation r

between two entities. Moreover, we improve the performance by transferring the parameters of model. Fig.1 shows the ST-PCNN architecture and the processing steps are as follows. First, pre-train the ST-PCNN model and retain the convolution layer parameters. Then fine-tune the model using target data and classify the relation between entities.

3.1 The Neural Network Architecture

The neural network architecture component for relation classification is described in the Fig.1. It illustrates the procedure in four main parts, including *Vector Representation, Convolution, Piecewise Max Pooling* and *Softmax Output*. We make full use of four types of information to map each word into a low dimension feature vector for relation classification.

1) **Word Embeddings.** Each word in a given sentence is mapped to a k dimensional real-valued vector by looking up the embedding matrix. Given a sentence consisting of N words $S = \{w_1, w_2, , w_n\}$, every word w_i is represented by a real-valued vector v_i. The dimension d^a of word embedding is x, we represent the word features (WFs) as WFs=$[v^a]$.

2) **Position Embeddings.** Similar to [6], we use position features (PFs) to specify entity pairs. In this paper, the PFs are defined to the combination of relative distances from the current word to head and tail entities. We assume that the size of position embedding dimension d^b is y and obtain the distance vectors $d^b{}_1$ and $d^b{}_2$ by looking up the position embedding matrixes, transforming the relative distances of current word to e_1 and e_2 into real valued vectors, and PFs = $[v^b{}_1, v^b{}_2]$.

3) **Part-of-Speech tags (POS).** To make use of the specific information from words themselves, we label each input word with its POS tag, e.g., noun, verb, etc. In our experiment, we only make use of a coarse-grained POS category, containing 36 different tags. We represent the POS feature vector as d^c. The dimension dc of POS tags is x.

4) **Grammatical relations.** Dependency paths (DP) are most informative to determine the two entities relation. We utilize dependency paths to obtain the grammatical relations between words. As Fig.2 shows, in the dependency path, each two neighbor words like *Financial* and *stress* are linked by a dependency relation *amod*. In our experiment, grammatical relations are grouped into 53 classes, mainly based on a coarse-grained classification.

Finally, we concatenate the word embeddings, position embeddings, POS and Grammatical relations embeddings of all words and transform an instance into a matrix $S \in \mathbb{R}^{s*d}$, where s is the sentence length and $d = d^a + d^b + d^c + d^e$. The matrix S is subsequently fed into the convolution layer.

Convolution is defined as an operation between a vector of convolution weights w and a vector of inputs treating as a sequence q. The weights matrix w is considered as the filter of convolution. In the example shown in Fig.1, we assume the length of the sliding window, filter size is $w(w = 3)$; thus , $w \in \mathbb{R}^n (n = w*d)$. Let us consider S as a sequence $\{q_1, q_2, ., q_n\}$, and let $q_{i:j}$ refer to the concatenation of q_i to q_j. We use n filters ($W = \{w_1, w_2, ., w_n\}$). Thus the convolution

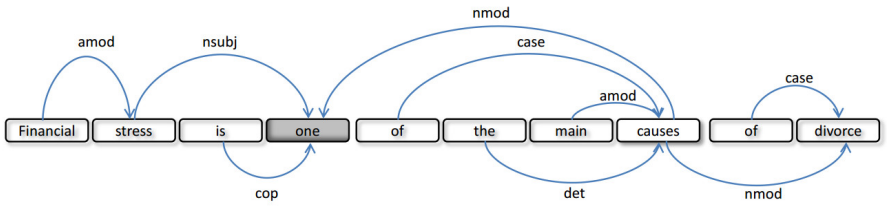

Fig. 2. Dependency parse example between stress and divorce entity pairs.

operation obtain another sequence $p \in \mathbb{R}^(n + w - 1)$ expressed as follows:

$$p_{ij} = w_i q_{j-w+1:j} \ (1 \leq i \leq n \qquad 1 \leq j \leq n + w - 1) \tag{1}$$

out-of-range input values q_i, where $i < 1$ or $i > n$, are taken to be zero. The convolution result is a matrix $C = \{c_1, c_2, ., c_s\} \in \mathbb{R}^{s*(n+w-1)}$. Fig.1 shows an example of 4 different filters in the convolution procedure.

In traditional Convolution Neural Networks (CNNs), single max pooling is not sufficient to capture the fine-grained feature and structural information between two entities. In relation extraction, an input sentence can be divided into three segments based on the two selected entities. Therefore we use the piecewise max pooling[9]. As shown in Fig.1, the output of each convolutional filter c_i is divided into three segments $\{c_{i1}, c_{i2}, c_{i3}\}$ by head and tail entities. The piecewise max pooling procedure can be expressed as follows:

$$p_{ij} = max(c_{ij}) \ (1 \leq i \leq n \qquad 1 \leq j \leq 3) \tag{2}$$

We obtain a 3-dimensional vector $p_i = \{p_{i1}, p_{i2}, p_{i3}\}$, which concatenates as $p_{1:n}$. Finally the piecewise max pooling outputs a vector:

$$g = \tanh(p_1 : n) \tag{3}$$

To compute the confidence of each relation, the feature vector g is fed into a soft max classifier.

$$o = w_1 g + b \tag{4}$$

$W_1 \in \mathbb{R}^{m*d}$ is the transformation matrix, and $o \in \mathbb{R}^m$ is the final output of the network, where m is equal to the number of possible relation types for the relation extraction system.

The ST-PCNNs based relation classification method proposed here could be stated as a quintuple $\theta = (D, W, W_1)$. (D present the concatenation of the WFs, PFs, POS and Grammatical relations embeddings.) Given an input example s, the network with parameter θ outputs the vector o, where the i-th component o_i contains the score for relation i. To obtain the conditional probability $p(i|x, \theta)$, we apply a softmax operation over all relation types:

$$p(i|x, \theta) = \frac{e^{o_i}}{\sum_{k=1}^{m} e^{o_k}} \tag{5}$$

Given all our (suppose T) training examples $(x^{(i)}; y^{(i)})$, we can then write down the log likelihood of the parameters as follows:

$$J(\theta) = \sum_{i=1}^{T} \log p(y^{(i)}|x^{(i)}, \theta) \tag{6}$$

We maximize $J(\theta)$ through stochastic gradient descent over shuffled mini-batches with the Adadelta [9] update rule.

3.2 Transferring knowledge of parameters

Parameter-transfer approaches assume that individual models for related tasks share some parameters or prior distributions of hyperparameters. Our parameter transferring procedure consists of the following steps, which we shows in the Fig.1: a) General-domain ST-PNCC pre-training; and b) target task ST-PCNN fine-tuning. Pre-training can accelerate the convergence rate of the model and can be beneficial for tasks with small datasets even with hundreds of labeled examples. We pre-train the ST-PCNN model on these datasets and retain the convolution layer parameters separately for use on the target relation classification task. The parameters transferred from the source domain will make the target domain data operate faster as it only needs to adapt to the idiosyncrasies of the target data. We utilize the *discriminative fine-tuning* and *slanted triangular learning rates*[11] for fine-tuning the model.

4 Experiments

Our experiments are designed to demonstrate that parameter transferring methods can increase the performance of specific domain relation classification tasks and take full advantage of automatically learned features using ST-PCNNs. We evaluate the performance of our method on several widely used datasets. We use resource-rich datasets, SemEval, NYT and FewRel Train as Source domains and three resource-poor datasets as target domains. The relations in the source and target data are basically different. Table 2 describes the division of groups The numbers of sentences and relation types describe the source domains data are rich and target domains are pool. We use AUC (Area Under Curve) as the evaluation metric in our experiments and the official macro-averaged F1-score to evaluate the model with other state-of-art models performance.

4.1 Experimental Settings

In this paper, we use the word2vec tool to train the word embeddings on the source and target domain datasets. Our experiments directly utilize 50-dimensional vectors by comparing the word embeddings beyond the scope of this paper. We experimentally study the effects of the parameters on the model. Following previous research work, we tune our models using three-fold validation on the training set. In Table 3 we show all parameters used in the experiments.

Table 2. Summary of Datasets

Datasets domains	Group A		Group B		Group C	
Source Target	SemEval	Dstl	New York Times	SemEval	FewRel Train	FewRel Test
#Relation Types	19	11	18,252	9	90	10
#Sentences	10,710	1200	522,611	10,710	63,000	7,000

Table 3. Summary of Datasets

Window size l	3
Word dimension d^a	50
Position dimension d^b	2*5
POS dimension d^c	10
Grammatical dimension d^e	10
Batch size B	160
Learning rate λ	0.5
Dropout probability p	0
Sentence_max_len	120

4.2 Comparison with other relation classification methods

To evaluate the proposed method, we select the following four relation classification methods for comparison through F1-Score evaluation: **SVM**[12] is a handcrafted features method and uses SVM for classification. **PCNN**[13] proposes a novel model dubbed the Piecewise Convolutional Neural Networks (PCNNs) with multi-instance learning to address the relation classification problem. **SDP-LSTM**[14] presents a neural network to classify the relation of two entities in a sentence. The architecture leverages the short dependency path (SDP). **HATT**[15] proposes hybrid attention-based prototypical networks for the problem of noisy few-shot relation classification.We implement them with the source codes released by the authors. Table 4 shows the F1 score for each method. We can observe that: (1) feature-based methods use the handcraft rules so it did not fit with other datasets. Thus we use the results presented by the author. (2) On the FewRel dataset, ST-PCNN significantly outperforms all other methods. It demonstrates that the source domain dataset, FewRel training data, is more similar to the target FewRel test data thought they have different relation type. (3) On the SemEval 2010 datasets, our method's mediocre performance may be because the source data NYT noise influence the transferring result. Thus we can see the source data noise and the similarity between source and target domains data would influence the experiment results to some extent.

4.3 Effect of Transfer learning

To evaluate the effects of the transfer learning method for relation classification, we empirically show the target domains datasets AUC scores with different e-

Table 4. Comparison of relation classification systems.

Classifier	Dstl F1	SemEval 2010 F1	FewRel F1
SVM	-	0.822*	-
PCNN	0.794	0.840	0.854
SDP-LSTM	0.792	0.836	0.844
HATT	0.804	0.810	0.849
ST-PCNN	0.803	0.838	0.872

pochs. We compare the S-PCNN and transferring results (ST-PCNN). We see the performance of transfer learning method on three target datasets. On D-stl, we observe a similarly dramatic improvement by 0.85 compared to the Dstl dataset directly running in the model with the AUC value 0.82. On FewRel, we improved by 0.94, rather than the baseline of 0.925. And on the dataset of SemEval 2010, we did not get a desired result. We see the transferring result has a slight decline compared with the baseline may because the source domain NYT corpus itself include noise data with automatic annotation. While these noise data with wrong label would also transfer to the target domain. Thus we see the source domain datasets noise would influence the final result of target data. We summarise the results in Figure 3.

4.4 Effect of different Features

Considering transferring different parameters corresponding to the features extracted from neural network, we performed transferring these features on the target domain datasets from the lexical and syntactic features of Table 5 to determine which type of features contributed the most. The results are presented in Table 5, from which we can observe that grammatical relations are effective for target data relation classification. The AUC score is improved remarkably when new features are added. When all of the lexical and syntactic level features are combined, we achieve relatively better results.

5 Conclusion and Future Works

In this paper, we exploit Syntactic Transferring Piecewise Convolutional Neural Networks (ST-PCNNs) for source-poor domain relation classification. In our method, source domain features are extracted and transferred to pre-train the neural networks for target domain relation classification tasks. Experimental results show that the proposed approach offers significant improvements over comparable methods. In the future, we will explore the following direction: 1) We will explore an evaluation model to measure the correlation between existing source domain data and target text corpus. 2) We will use multi-granularity convolution filter to select different granularity features.

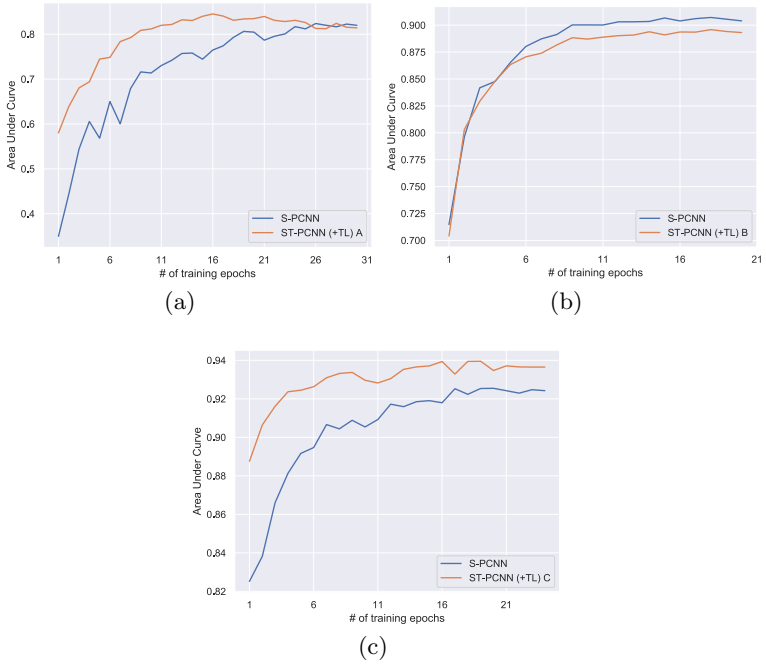

Fig. 3. AUC score for S-PCNN and ST-PCNN vs. training with different epochs on Dstl, SemEval 2010, FewRel datasets (from left to right)

Table 5. Effect of different features

Channels (Dstl)	**AUC**
Baseline	0.820
Word Embeddings + Position Embeddings-TL	0.834
+ POS embeddings (only)-TL	0.840
+ GR embeddings (only) -TL	0.842
+ POS + GR embeddings-TL	0.847

Channels(SemEval2010)	**AUC**
Baseline	0.907
Word Embeddings + Position Embeddings-TL	0.894
+ POS embeddings (only)-TL	0.894
+ GR embeddings (only) -TL	0.895
+ POS + GR embeddings-TL	0.896

Channels (FewRel)	**AUC**
Baseline	0.925
Word Embeddings + Position Embeddings-TL	0.938
+ POS embeddings (only)-TL	0.940
+ GR embeddings (only) -TL	0.942
+ POS + GR embeddings-TL	0.942

6 Acknowledgements

This work was supported by the National Key Research and Development Program of China (Grant no. 2016YFB1000905), the National Natural Science Foundation of China (Grant nos. 61572091, 61772096).

References

1. Mintz, Mike, et al. "Distant supervision for relation extraction without labeled data." 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. Association for Computational Linguistics, 2009.
2. Pan, Sinno Jialin, and Qiang Yang. "A survey on transfer learning." IEEE Transactions on knowledge and data engineering 22.10 (2010): 1345-1359.
3. Min, Bonan, et al. "Ensemble semantics for large-scale unsupervised relation extraction." Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. Association for Computational Linguistics, 2012.
4. Culotta, Aron, and Jeffrey Sorensen. "Dependency tree kernels for relation extraction." Proceedings of the 42nd annual meeting on association for computational linguistics. Association for Computational Linguistics, 2004.
5. GuoDong, Zhou, et al. "Exploring various knowledge in relation extraction." Proceedings of the 43rd annual meeting on association for computational linguistics. Association for Computational Linguistics, 2005.
6. Zeng, Daojian, et al. "Relation classification via convolutional deep neural network." (2014).
7. Nguyen, Thien Huu, and Ralph Grishman. "Relation extraction: Perspective from convolutional neural networks." Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing. 2015.
8. Liu, Tianyi, et al. "Neural relation extraction via inner-sentence noise reduction and transfer learning." arXiv preprint arXiv:1808.06738 (2018).
9. Zeng, Daojian, et al. "Distant supervision for relation extraction via piecewise convolutional neural networks." Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. 2015.
10. Zeiler, Matthew D. "ADADELTA: an adaptive learning rate method." arXiv preprint arXiv:1212.5701 (2012).
11. Howard, Jeremy, and Sebastian Ruder. "Universal language model fine-tuning for text classification." arXiv preprint arXiv:1801.06146 (2018).
12. Hendrickx, Iris, et al. "Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals." Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions. Association for Computational Linguistics, 2009.
13. Zeng, Daojian, et al. "Distant supervision for relation extraction via piecewise convolutional neural networks." Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. 2015.
14. Xu, Yan, et al. "Classifying relations via long short term memory networks along shortest dependency paths." proceedings of the 2015 conference on empirical methods in natural language processing. 2015.
15. Gao, Tianyu, et al. "Hybrid Attention-Based Prototypical Networks for Noisy Few-Shot Relation Classification." (2019).