Relation Classification via Target-Concentrated Attention CNNs

Jizhao Zhu[1], Jianzhong Qiao[1(✉)], Xinxiao Dai[2], and Xueqi Cheng[3]

[1] College of Computer Science and Engineering, Northeastern University, Shenyang 110169, China
zhujz.neu@gmail.com, qiaojianzhong@mail.neu.edu.cn
[2] Shenyang Open University, Shenyang 110003, China
daixx.syou@foxmail.com
[3] CAS Key Laboratory of Network Data Science and Technology, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 110190, China
cxq@ict.ac.cn

Abstract. Relation classification is a key natural language processing task that receives much attentions these years. The goal is to assign pre-defined relation labels to the nominal pairs marked in given sentences. It is obvious that different words in a sentence are differentially informative. Moreover, the importance of words is highly relation-dependent, i.e., the same word may be differentially important for different relations. To include sensitivity to this fact, we present a novel model, referred to as TCA-CNN, which takes the attention mechanism at the word level to pay different attention to individual words according to the semantic relation concentrated when constructing the representation of a sentence. Experimental results show that TCA-CNN achieves a comparable performance compared with the state-of-the-art models on the SemEval 2010 relation classification task.

Keywords: Relation classification · Convolutional Neural Networks · Attention mechanism

1 Introduction

Relation classification is one of the fundamental tasks in natural language processing (NLP). It plays an important role in various scenarios, e.g., information extraction [1], question answering [2], knowledge base construction [3,4], etc. The goal of relation classification is to assign pre-defined relation labels to the nominal pairs marked in given sentences. For instance, given the sentence "Givers gain moral strength and [happiness]$_{e_1}$ from [giving]$_{e_2}$." with the annotated nominal pair, namely e_1 and e_2, the goal would be to automatically

J. Zhu–The work was conducted when Jizhao Zhu visited CAS Key Lab of Network Data Science and Technology.

D. Liu et al. (Eds.): ICONIP 2017, Part II, LNCS 10635, pp. 137–146, 2017.
https://doi.org/10.1007/978-3-319-70096-0_15

recognize that this sentence expresses *Cause-Effect* relation between e_1 and e_2, denoted as *Cause-Effect*(e_1, e_2).

Traditional relation classification methods mainly fall into feature- and kernel-based categories. Feature-based methods use a large number of lexical, syntactic or semantic features and feed them into a chosen classifier such as support vector machines (SVM) to classify relations. Conversely, kernel-based do not need much effort on feature engineering, but well-designed kernel functions, which are usually based on syntactic or dependency structures. All these methods have been shown to be effective and yield relatively high performance. However, they strongly depend on extracted features or designed kernels derived from the output of pre-existing NLP tools, which unavoidably lead to the propagation of the errors and hurt the performances of these models. Recently, the methods based on deep neural networks with highly automatic feature learning have made remarkable progress. A large number of works on relation classification use convolutional neural networks (CNN) [5], recursive neural networks (RecursiveNN) [6] and recurrent neural networks (RNN) [7] to reduce the extensive manual feature engineering or other external resources, and have already achieved impressive results.

Although these existing deep neural networks based models have been quite effective, they treat all words equally when composing the representation of the sentence meaning. Obviously, different words in a sentence are differentially informative. For this reason, the attention mechanism was adopted by [8] and the state-of-the-art performance was achieved on the benchmark SemEval 2010 Task 8. However, a word may express kinds of semantic relations with different probabilities. To illustrate, we take the aforementioned sentence as an example. It is intuitive that the importance of word "from" is higher when the semantic relation *Cause-Effect* is concentrated than *Message-Topic* to construct the sentence representation. Therefore, the importance of a word is related to the concentrated semantic relation when constructing the sentence representation, but the existing models have not noticed this yet.

In this paper, we present a novel model, Target-Concentrated Attention Convolutional Neural Networks (TCA-CNN), which takes the attention mechanism at the word level to pay more or less attention to individual words when different semantic relations are concentrated to compose the representation of a sentence. We evaluate our model for relation classification on standard benchmark dataset of SemEval 2010 Task 8. Experimental results show that our proposed method achieves an excellent result compared with existing baselines. The main contributions of our work can be summarized as follows:

- We propose an end-to-end learning model, named TCA-CNN, without extensive feature engineering and external knowledge, and it could capture the key parts of a sentence when different relations are concentrated to compose the sentence representation.
- We present a new pairwise margin-based loss function which is superior to the typical cross-entropy loss functions.

– Experiments conducted on the benchmark dataset of SemEval 2010 Task 8 demonstrate that TCA-CNN achieves a comparable performance with the state-of-the-art models.

2 Related Work

We briefly review the existing studies on relation classification. Traditional methods strongly depend on the extracted features which are often derived from the output of pre-existing NLP tools. So it is unavoidable to the propagation of the errors in the existing tools and the performance of these methods are limited. Recently, deep neural networks have shown promising results and they learn underlying features automatically. Socher et al. [6] proposed MVRNN by using a recursive neural network to tackle relation classification. They managed to capture the compositional aspects of the sentence semantics by exploiting syntactic trees. Zeng et al. [11] exploited DNN to classify relations with lexical, sentence level features and word position features, and they took all of the words as input without complicated pre-precessing. Nevertheless, these methods still depend on additional features from lexical resources and NLP tools. Based on CNNs, Santos et al. [15] proposed the CR-CNN model with special treatment for the *Other* label. Xu et al. [7] leveraged CNNs to learn representation from shortest dependency paths, and address the relation directionality by special treatment on sampling. Additionally, some other deep learning models have been proposed such as [7,16,18]. Since different words in a sentence are differentially informative, Wang et al. [8] introduced the attention mechanism into relation classification task and proposed a novel convolutional neural network architecture relying on two levels of attention in order to better discern patterns in heterogeneous contexts. In this paper, we also adopt attention mechanism to attend to the key parts of a sentence when constructing the representation of then sentence and experimental results show that our model achieves a comparable performance with the state-of-the-art models.

3 Our Proposed Model

In this section, we describe the proposed model for relation classification with target-concentrated attention mechanism. An overview of our architecture is illustrated in Fig. 1. The reason for choosing a CNN rather than other deep neural networks like RNN with long short-term memory unit (LSTM) [9] or gated recurrent unit (GRU) [10], etc., is we argue that CNN is more suitable to detect the key part of sentence relevant to the concentrated relation. The only input for the network is the tokenized text string of the sentence and a semantic relation. First, the input sentence is encoded with the concatenation of word vector and position vectors, where word order is captured by exploiting the positional encoding. Next, the attention mechanism is used to capture the relevance of words with respect to the concentrated relation. After that, a convolutional layer followed by max-pooling is applied to construct a representation of the

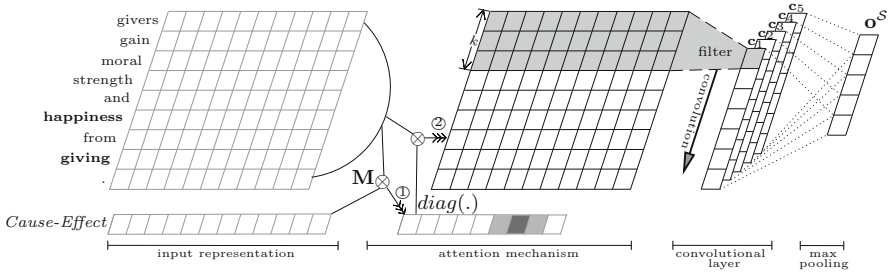

Fig. 1. The architecture of TCA-CNN.

sentence. Finally, by using a scoring function to measure the proximity between the sentence representation and the given relation. In this paper, we use capitalized letter with boldface to denote matrix, and the corresponding lowercase letter with boldface to represent column vector.

3.1 Input Representation

Given a sentence $\mathcal{S}$ with words w_i for $i = 1, 2, \ldots, n$, where n is the sentence length, and two marked nominals e_1 and e_2, we first convert each word into a real-valued vector. Let $\mathbf{E}^w \in \mathbb{R}^{d_w \times |V|}$ denote the word embeddings matrix, where V is the input vocabulary and d_w is the word vector dimension. The i-th word in $\mathcal{S}$ is transformed into the vector $\mathbf{e}_i^w$ by looking up the word embeddings matrix.

It is obvious that contexts surrounding the nominal pair are critical to determine the semantic relation between marked nominals. Therefore, we also incorporate the word position embedding (position features) proposed by [11] to reflect the relative distances of the current word to the marked nominals e_1 and e_2. Take the sentence shown in Fig. 1 as an example, the relative distances of "from" to "happiness" and "giving" are -1 and 1, respectively. Then, the relative distance is mapped to a vector with size d_p, and d_p is a hyper-parameter to be chosen by the user. Let $\mathbf{e}_{i,1}^p, \mathbf{e}_{i,2}^p \in \mathbb{R}^{d_p}$ denote the position vectors corresponding to the i-th word in a sentence. The overall word embedding $\mathbf{w}_i$ for the i-th word can be obtained by concatenating the word embedding with these two position vectors, namely $\mathbf{w}_i = [\mathbf{e}_i^w; \mathbf{e}_{i,1}^p; \mathbf{e}_{i,2}^p]$ ($[\mathbf{x}_1; \mathbf{x}_2]$ denotes the vertical concatenation of $\mathbf{x}_1$ and $\mathbf{x}_2$). Based on these $\mathbf{w}_i$, the input representation for the sentence $\mathcal{S}$ can be represented as a matrix $\mathbf{S} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n]$.

3.2 Input Attention

Since not all words contribute equally to the representation of the sentence meaning. Moreover, the importance of words are highly relation-dependent, i.e., the same word may be differentially important for different relations. In this paper, we introduce attention mechanism to automatically capture the relevance of

words with respect to the concentrated relation when constructing the sentence representation.

On the basis of input representation, we can measure the importance of words in a sentence concentrated on the given semantic relation $r \in R$, where R is the relation set. In this paper, we choose a bilinear function to characterize the importance ξ_i of the i-th word in a sentence with the semantic relation r, given by:

$$\xi_i = \mathbf{w}_i^\top \mathbf{M}\mathbf{r} + b, \tag{1}$$

thereof, $\mathbf{M}$ is a weighting matrix to be learned during the training process, $\mathbf{r} \in \mathbb{R}^{d_r}$ is the embedding of relation r, and $b \in \mathbb{R}$ is the bias term. Then, the normalized importance weight α_i can be obtained through a softmax function, namely:

$$\alpha_i = \frac{\exp(\xi_i)}{\sum_{k=1}^{n} \exp(\xi_k)}. \tag{2}$$

After that, the diagonal attention matrix $\mathbf{A}$ can be obtained, as follows:

$$\mathbf{A} = diag(\alpha_1, \alpha_2, \ldots, \alpha_n). \tag{3}$$

Finally, the input for the convolutional layer can be get by multiplying $\mathbf{S}$ with $\mathbf{A}$, in the form:

$$\mathbf{Q} = \mathbf{S}\mathbf{A}. \tag{4}$$

3.3 Sentence Representation

The next phrase of our proposed model is to construct the distributed representation $\mathbf{o}^S$ for the input sentence $\mathcal{S}$. The convolutional layer first captures local contextual information with a sliding window of size k over the sentence and the k will be chosen by the user. Afterwards, it combines all local contextual information via a map-pooling operation to obtain a fixed-sized vector for the input sentence. Let $\mathbf{z}_i \in \mathbb{R}^d$ refer to the concatenation of the k successive words embeddings centered around the i-th word:

$$\mathbf{z}_i = [\mathbf{q}_{i-(k-1)/2}; \ldots; \mathbf{q}_{i+(k-1)/2}], \tag{5}$$

where $d = k \times (d_w + 2 \times d_p)$. Since the window may be outside of the sentence boundaries when it slides near the boundary, an extra padding token is repeated multiple times at the beginning and the end of the input. The convolution operation is defined as the dot product of a weight matrix $\mathbf{W}^c \in \mathbb{R}^{d \times l}$ with the matrix $\mathbf{Z} \in \mathbb{R}^{d \times (n-k+1)}$ and then adding a bias vector $\mathbf{b}^c \in \mathbb{R}^l$, where l is the number of filters. We apply a non-linear activation function at the output of the convolutional operation, such as the hyperbolic tangent. For the i-th filter, the convolutional operation can be expressed by:

$$\mathbf{c}_{ij} = \tanh(< \mathbf{w}_i^c, \mathbf{z}_j > + \mathbf{b}_i^c), \tag{6}$$

where $\mathbf{b}_i^c$ is the bias term. Afterwards, the representation vector $\mathbf{o}^S$ for the input sentence can be obtained through the max-pooling operation on each $\mathbf{c}_i = \{\mathbf{c}_{ij}\}$ for $j = 1, 2, \ldots, (n - k + 1)$, so that:

$$\mathbf{o}^S = [\max(\mathbf{c}_1), \max(\mathbf{c}_2), \ldots, \max(\mathbf{c}_l)]^\top. \tag{7}$$

3.4 Scoring

In this work, we propose a new scoring function $\zeta_\theta(\mathcal{S}, r)$ to measure the proximity between the sentence representation $\mathbf{o}^S$ and the given relation r, as follows:

$$\zeta(\mathcal{S}, r) = (\mathbf{o}^S)^\top \mathbf{U}\mathbf{r}, \tag{8}$$

where $\mathbf{U}$ is a weighting matrix to be learned during training.

3.5 Model Training

The model could be trained in and end-to-end way with standard back propagation. We define a margin-based pairwise loss function $\mathcal{L}$ based on Eq. (8), in the form:

$$\mathcal{L} = \phi(\zeta(\mathcal{S}, r^-) + \gamma - \zeta(\mathcal{S}, r^+)) + \beta\|\theta\|^2, \tag{9}$$

where $\phi = \log(\exp(x) + 1)$, γ is the margin separating the positive pair from the negative one, $\zeta(\mathcal{S}, r^+)$ is the matching score between the sentence representation $\mathbf{o}^S$ and the ground-truth relation r^+, $\zeta(\mathcal{S}, r^-)$ denote the matching score between $\mathbf{o}^S$ and the incorrect relation r^-, β is the L_2-regularization term and θ is the parameter set consisting of $\mathbf{M}, \mathbf{U}, \mathbf{W}^c, \mathbf{b}^c$.

Table 1. Hyper-parameters used in our experiments.

Parameter	Parameter Name	Value
d_w	word embedding size	100
d_p	word position embedding size	80
d_r	relation embedding size	80
k	filter size	4
l	filter number	1000
γ	margin	1.0
λ	initial learning rate	0.002

In our experiments, we use the publicly available word2vec[1] skip-gram architecture [12] to learn the initial word embeddings on Wikipedia[2]. The embeddings of out-of-vocabulary words and all relations are randomly initialized with

[1] https://code.google.com/p/word2vec/.
[2] https://dumps.wikimedia.org/enwiki/.

uniform samples from *U(-1.0,1.0)*. All hyper-parameters are jointly learned via minimizing the loss function of Eq. (9). Additionally, AdaGrad [13] is used as our optimization method during the training process. Table 1 reports all the hyper-parameters used in the following experiments.

4 Experiments

4.1 Dataset and Evaluation Metrics

To evaluate the performance of TCA-CNN, we conduct experiments on SemEval-2010 Task 8 dataset [14] which is a widely used benchmark for relation classification and freely available[3] on the internet. The dataset contains 10,717 examples, including 8,000 training instances and 2,717 test instances, annotated with 9 directed relation labels and 1 undirected *Other* label. Taking the directionality of the relation labels into account, e.g., *Cause-Effect*(e_1, e_2) and *Cause-Effect*(e_2, e_1) are different relation labels, we treat each directed relation labels as two in our model. We evaluate the model performance by using the SemEval-2010 Task 8 official scorer in terms of the macro-average F1-scores for the 9 directed relations (excluding *Other*).

Table 2. Comparison with other published results of Neural Network models.

Classifier	F1
MVRNN [6]	82.4
CNN+Softmax [11]	82.7
CR-CNN [15]	84.1
DepNN [16]	83.6
depLCNN [17]	83.7
depLCNN+NS [17]	85.6
SDP-LSTM [7]	83.7
DRNNs [18]	85.8
Att-Input-CNN [8]	87.5
TCA-CNN	87.3

4.2 Results and Analysis

The experimental results on the test set are reported in Table 2. MVRNN and DepNN are based on RecursiveNN, whereas DepNN achieves an F1-score of 83.6% exceeding MVRNN with a relative improvement of 1.7% by capturing the features of shortest dependency paths via CNN. Both CNN-based model depLCNN and RNN-based SDP-LSTM leverage the shortest dependency paths

[3] http://docs.google.com/View?id=dfvxd49s_36c28v9pmw.

between the marked nominal pair and obtain the identical results. By considering the relation directionality with a negative sampling strategy, depLCNN further improves the result to 85.6%. From the results, we can see that our novel target-concentrated attention based architecture achieves the F1-score of 87.3%, outperforming the well known CR-CNN model by 3.2% and DRNNs by 1.5%, but the accuracy is a slightly lower than the state-of-the-art model Att-Input-CNN. The results indicate that TCA-CNN effectively captures the key part of sentence for constructing the representation of a sentence.

4.3 Visualization of Attention

In order to validate that our model is able to select informative words in a sentence with the semantic relation concentrated, we can obtain the attention weight α in Eq. (2) and visualize the word level attention weights in Fig. 2 for the sentence mentioned in Introduction.

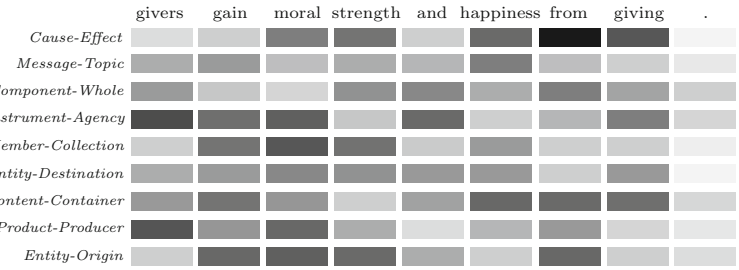

Fig. 2. Visualization of Attention.

Each line in Fig. 2 shows the representation of how attention focuses on words with the interaction of the concentrated semantic relation. The color depth indicates the degree of importance, namely the darker the more important. From the Fig. 2, we can observe that when concentrating on: (1) the ground-truth relation of *Cause-Effect*, the word "from" was assigned the highest attention weight, and the words such as "happiness" and "giving" also are important. However, it is surprised to find that the non-entity tokens "moral" and "strength" are assigned the same level importance as "happiness". After detailed analyzing the sentence, we can know that the "moral strength" also is a result caused by giving; (2) other relations, we first take the relation *Message-Topic* as an example, the key word "from" is assigned a lower attention value, which means "from" is irrelevant with respect to the semantic relation *Message-Topic*. As a result, the output sentence representation would have low matching score with *Message-Topic*. Besides, the similar phenomena can be found when the rest of the other relations are concentrated to compose the sentence representations.

5 Conclusion

In this work we propose an end-to-end learning model, referred to as TCA-CNN, with target-concentrated attention mechanism for relation classification. Our motivation is that different words in a sentence are differentially informative and the importance of words are highly relation-dependent. The experimental results based on the SemEval-2010 Task 8 dataset show that TCA-CNN achieves a comparable performance compared with the state-of-the-art models. In the future, it might be interesting to jointly model the entity pair and relation with attention mechanism, since it is intuitive that the relation interact closely with the entity pair in a sentence.

Acknowledgments. This work is supported by the 973 Program of China under Grant Nos. 2013CB329606 and 2014CB340405, the National Key Research and Development Program of China under Grant No. 2016YFB1000902, the National Natural Science Foundation of China (NSFC) under Grant Nos. 61272177, 61402442, 61572469, 91646120 and 61572473.

References

1. Wu, F., Weld, D.S.: Open information extraction using Wikipedia. In: 48th Annual Meeting of the Association for Computational Linguistics, pp. 118–127. ACL Press, Stroudsburg (2010)
2. Golub, D., He, X.: Character-level question answering with attention. arXiv preprint arXiv:1604.00727 (2016)
3. Shin, J., Wu, S., Wang, F., De Sa, C., Zhang, C., Ré, C.: Incremental knowledge base construction using deepdive. Proc. VLDB Endowment **8**, 1310–1321 (2015)
4. Jia, Y., Wang, Y., Lin, H., Jin, X., Cheng, X.: Locally adaptive translation for knowledge graph embedding. In: 30th AAAI Conference on Artificial Intelligence, pp. 992–998. AAAI Press, Menlo Park (2016)
5. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**, 2278–2324 (1998)
6. Socher, R., Huval, B., Manning, C.D., Ng, A.Y.: Semantic compositionality through recursive matrix-vector spaces. In: 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 1201–1211. ACL Press, Stroudsburg (2012)
7. Xu, Y., Mou, L., Li, G., Chen, Y., Peng, H., Jin, Z.: Classifying relations via long short term memory networks along shortest dependency paths. In: 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1785–1794. ACL Press, Stroudsburg (2015)
8. Wang, L., Cao, Z., de Melo, G., Liu, Z.: Relation classification via multi-level attention cnns. In: 54th Annual Meeting of the Association for Computational Linguistics, pp. 1398–1307. ACL Press, Stroudsburg (2016)
9. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**, 1735–1780 (1997)
10. Cho, K., Van Merriënboer, B., Bahdanau, D., Bengio, Y.: On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259 (2014)

11. Zeng, D., Liu, K., Lai, S., Zhou, G., Zhao, J.: Relation classification via convolutional deep neural network. In: 25th International Conference on Computatinal Linguistics: Technical Papers, pp. 2335–2344. ACM, New York (2014)
12. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
13. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res. **12**, 2121–2159 (2011)
14. Hendrickx, I., Kim, S. N., Kozareva, Z., Nakov, P., Ó Séaghdha, D., Padó, S., Pennacchiotti, M., Romano, L., Szpakowicz, S.: 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, pp. 94–99. ACL Press, Stroudsburg (2009)
15. Santos, C.N.D., Xiang, B., Zhou, B.: Classifying relations by ranking with convolutional neural networks. arXiv preprint arXiv:1504.06580 (2015)
16. Liu, Y., Wei, F., Li, S., Ji, H., Zhou, M., Wang, H.: A dependency-based neural network for relation classification. In: 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing(Short Papers), pp. 285–290. ACL Press, Stroudsburg (2015)
17. Xu, K., Feng, Y., Huang, S., Zhao, D.: Semantic relation classification via convolutional neural networks with simple negative sampling. In: 2015 Conference on Empirical Methods in Natural Language Processing, pp. 536–540. ACL Press, Stroudsburg (2015)
18. Xu, Y., Jia, R., Mou, L., Li, G., Chen, Y., Lu, Y., Jin, Z.: Improved relation classification by deep recurrent neural networks with data augmentation. arXiv preprint arXiv:1601.03651 (2016)