Proceedings of the 3rd International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/5/20230712

Graph convolutional network with position and global information improves relation extraction

Jingxuan Chen

Jinan University-University of Birmingham Joint Institute, Jinan University, Guangzhou, Guangdong, 511436, China

jxc1424@student.bham.ac.uk

Abstract. With the continuous improvement of the related relation extraction model, researchers focused on optimizations of the input of the sentence expression structures, such as dependency trees, and slight improvements and integrations of basic models applied to this task. This paper utilizes position-aware attention mechanisms to highlight critical information and makes good use of global information generated by the bidirectional LSTM, absorbing characteristics and advantages of those related models to propose a new Position-Aware Graph Convolutional Network (PA-GCN) model and describe it in three steps. Moreover, this paper trials the model on the RACRED dataset to evaluate its performance and suitability. The experiment results demonstrate that the PA-GCN model outperformed others listed in this paper, achieving an F_1 score of 66.6%, which verifies an improvement of the model and the effectiveness of adding an attention-aware mechanism.

Keywords: graph convolutional network, bidirectional long short-term memory network, position-aware attention mechanism, relation extraction.

1. Introduction

The idea of using graph structure to resolve practical issues has a significant influence on the development of relation classification and relation extraction. With the attention of the non-Euclidean data increasing, this kind of application has become more extensive. Many neural network scholars began to utilize tree structures together with neural networks related to graphs to handle these tasks.

However, the relation extraction process is not a trivial task, and the relation classification is also complicated, which requires the machine to recognize a piece of text with a semantic property of interest to make a correct annotation [1]. Early scientists created methods mainly using sentence analysis tools to progress in this field to realize the recognition, extraction, and construction. In 2015, Zeng et al. did not rely on the tool kits of natural language processing. They realized the application of Convolutional Deep Neural Networks in relation to extraction has an outstanding performance with the position features [1]. From then on, neural models have been frequently adopted for relation extraction. Not only the convolutional neural network (CNN) but an entity relation extraction model with respect to Long Short-Term Memory network (LSTM) have been widely studied recently [2,3]. Zhang et al. (two groups) significantly contributed to this field. Zhang, Qi, and Manning proposed a position-aware attention mechanism over the LSTM network (PA-LSTM). Zhang et al. focused on processing dependency trees

DOI: 10.54254/2755-2721/5/20230712

and applied a Graph Convolution network (GCN) to extracting relation. Then, they verified the feasibility and efficiency of the PA-LSTM and the Contextualized GCN (C-GCN) [4,5].

This paper's inspiration comes from utilizing the position-aware attention mechanism and the model that takes advantage of both GCN and LSTM. The inspires contain: 1) Gu et al. created a neural network model with the Position-aware Bidirectional Attention, and a new neural sequence model, over an LSTM network with a position-aware attention mechanism, is proposed by Zhang, Qi, and Manning. They demonstrated the importance of position information [5,6]. 2) Zhang et al. and Marcheggiani et al. revealed that the dependency-based and sequence models are complementary and used good experiments and complementary performance to prove that the strengths of GCN and LSTM can be complemented well [4,7].

Therefore, following these ideas, this paper will first introduce and analyse several related models concerning GCN and LSTM along with a dataset, then compare and obtain the more suitable model for solving the relation extraction task. According to the observation that the C-GCN and the PA-LSTM model showed different accuracy rates on different TACRED dev examples and their innovative way of splicing in proportion, this paper will try to integrate these advantages later [4,5]. A new model called Position-Aware Graph Convolutional Network will be proposed to pursue a higher efficiency and accuracy in the relation extraction task.

As for the experiments, this paper will apply the Position-Aware Graph Convolutional Network to the TAC Relation Extraction Dataset and demonstrate the suitability of the different models in tackling the relation extraction problems via the comparison of F_1 scores. Simultaneously, the significance of the summary vector, one of the outputs of Bi-LSTM, and the position vector generated from the Position-aware attention mechanism are proved.

2. Model

2.1. Graph convolutional networks

Given a graph, let it be expressed as G = {V, E, A}. In this expression, V is a node set including n words vectors with d dimension, and A, a weighted adjacency matrix, represents the status of the note connecting. Based on Graph Laplacian matrix, Graph Fourier transform and the goal of simplifying and improving feasibility, Kipf et al. created Graph Convolutional Network with an efficient layer-wise propagation rule to aggregate the node information and provided a method for a classification of related nodes [8].

$$h_i^{(l)} = \sigma(\sum_{j=1}^{n} (\tilde{A}_{ij} / \sum_{k=1}^{n} \tilde{A}_{ik}) W^{(l)} h_j^{(l-1)} + b^{(l)}) \quad (1)$$

Using GCN as a feature extractor and using the adjacency matrix to represent the tree structure to consider relationship between nodes can combine nodalizated data with other adjacency feature to output more useful vectors containing their information by the method of weighted summation. According to the layer-wise propagation rule (1), the word vectors h_i are input into the model to compose the matrix that can be updated through the GCN layers. W is a weight matrix work on h to realize linear transformation and weight assignment. A is the adjacency matrix where $A_{ij} = 1$ means node adjacency. Adding a self-loop to change A into $\tilde{A}$ can transmit its own information to the next layer of GCN, and regularizing can reduce the serious numerical difference from the degree of each vertex. After the matrix product of above items, it just needs to add bias and use activation function σ to complete the calculations of each layer.

Following the layer-wise propagation rule (1) and the above characteristics, it can be noticed that GCN may be appropriate for a variety of NLP applications, including semantic role labelling and neural machine translation, and can be used to extract relations between words to a certain degree [7]. GCN will be involved and widely used in the remainder of this paper.

2.2. Graph convolutional networks

Zhang et al., who drew on the ideas of GCN, Bidirectional LSTM and feed-forward neural network and so on, presented the Contextualized Graph Convolutional Networks model (C-GCN) [4]. At the

DOI: 10.54254/2755-2721/5/20230712

beginning of the model, the words can be encoded by GLoVe to change to word vectors x_k which means the word in the k-th position and the sentence will be formalized as X=$[x_1, x_2, \dots, x_n]$, where $[x_{s_1}, \dots, x_{s_2}]$ and $[x_{o_1}, \dots, x_{o_2}]$ mean every subjects and every objects [9]. And then, a more effective and accurate sequence structures model Bi-LSTM are chosen to process the word vectors because it can capture the context information of each word better and allow both past and future features can be extracted and made full use of by the hidden state [10]. After operated from the different two direction, they will be fed into l-layers GCN for a further transformation. During the process of GCN, the adjacency matrix is constructed according to dependency tree that uses human language technical tools (Stanford CoreNLP) and path-centric pruning technique to build. And the model assigns the weights to the input values [11].

$$\begin{cases} h_s = M\left(h_{s_1:s_2}^{(l)}\right) = M(GCN\left(BiLSTM\left(x_{s_1:s_2}\right)\right) \\ h_t = M\left(h^{(l)}\right) = M(GCN\left(BiLSTM(x)\right)) \\ h_o = M\left(h_{o_1:o_2}^{(l)}\right) = M(GCN\left(BiLSTM\left(x_{o_1:o_2}\right)\right) \end{cases} \tag{2}$$

At the end of the C-GCN model, the network combines the output of the key information h_o and h_s with the vector h_t formed after the max pooling function M between all the vectors, putting the fixed vector to obtain the h_{final} vector in the feed-forward neural network, and finally calculating the probability by a softmax operation.

2.3. Position-aware graph convolutional network

In the recent study, most current studies typically ignore that the position data is also a crucial step toward some aspects during the relation extraction processing [6]. Consequently, this paper makes good use of the position information and proposes a new model: Position-Aware Graph Convolutional Network (PA-GCN). This model's details can be divided into three stages: the Bi-LSTM step, position information integration, and the GCN step.

As illustrated in the C-GCN model, the sentence X can be initialized as a form of $[x_1, x_2, \dots, x_n]$ and like the subject and the object. Then, in view of the idea of bi-direction and the thinking from multiple perspective, this paper stacks both GCN and Bi-LSTM to be easier to handle the long-distance entities as well as the local features. Two vectors Bi-LSTM processed will be spliced with each other as a whole vector. Similarly, the summery vector q comes from q_L and q_R, $q_L(or\ q_R) = h_{L_n}\left(or\ h_{R_n}\right)$, and its function is to provide a global information.

Since the excellent complement of dependency structures and sequence structure and the critical role of the position information, this paper will set a position-aware attention mechanism for the next step and convert the position information into the digital form. Aim to calculate the distance between the subject (or object) with other words, and this paper uses the following rule to represent the position [1,5].

$$lp_i^s = \begin{cases} i - j_{s_1}, & i < j_{s_1} \\ 0, & j_{s_1} \le i \le j_{s_2} \\ i - j_{s_2}, & i < j_{s_2} \end{cases} \tag{3}$$

$$p_i^o = \begin{cases} i - j_{o_1}, & i < j_{o_1} \\ 0, & j_{o_1} \le i \le j_{o_2} \\ i - j_{o_2}, & i < j_{o_2} \end{cases} \tag{4}$$

Obtained position embedding vectors p_i^s and p_i^o, which means the position information is successfully extracted and the ability of word expression may be enhanced. The key parts can be also highlighted and the information impact from the other words will not completely be ignored simultaneously. Moreover, the embedding vectors will be combined to generate the weights a_i together with the output of Bi-LSTM following Eq. (5). They will be combined to each other and adding an activation function σ in order to make a transformation and a position information integration.

$$a_i = \lambda\, \sigma(W_h\,[h_{L_i}; h_{R_i}] + W_q\,[q_L; q_R] + W_s p_i^s + W_o p_i^o) \tag{5}$$

DOI: 10.54254/2755-2721/5/20230712

To get a numerical result, the function is multiplied by a learnable parameter λ whose dimension is same as the dimension of the column of these weight matrices W. After normalized, the term of a_i will be multiplied by the output vector from Bi-LSTM to get the input of GCN and move on to the last crucial step.

$$h_i^{(0)} = \exp(a_i)\left[h_{L_i}; h_{R_i}\right]/(\sum_{j=1}^{n} \exp(a_j)) \quad (6)$$

The size of the vectors that input GCN are affected by these formulas, and the $h_i^{(0)}$ can be considered as inputs of GCN accompanied with the adjacency matrix generated according to the independency tree. The feature vector h_i can be aggregated with nearby word vectors h_j to achieve the influence like the convolution. Eventually, the object and subject vectors will be concatenated with the other vectors after max pooling operation and they will be converted into a probability by feed-forward neural network together with the softmax layer, which resembles the C-GCN model. Through this model, position vector p and summery vectors q can be used effectively to realize relation extraction.

3. Experimental results and analysis

3.1. TACRED dataset

Since the annotated data cannot satisfy the request of the models, a large supervised dataset called TAC Relation Extraction Dataset (TACRED) that is targeted on the basis of TAC KBP relations was collected by Zhang, Qi, and Manning [5]. Through the extensive research of sentence-level relation extraction datasets, they gathered human annotations from the TAC KBP challenges and crowdsourcing to construct a massive dataset with 41 relation types and 1194740 examples [12]. This article evaluates the model on the TACRED because it contains more relation instances and is more suitable for its complexity and authenticity.

3.2. Implementation details

After the description of PA-GCN, the substantial improvement of it should be verified. During the experiment, this paper adheres to randomization, contrast, repetition, and balance principles. To ensure the control of variables and accurately reflect the impacts of the proposed models compared with other models to select the most appropriate model, a large number of similar or identical parameters are set in this experiment. Meanwhile, this paper modelled his experiment to compare the PA-GCN model with the data of some traditional or improved methods referred to in the experiment from Zhang et al., such as logistic regression, SDP-LSTM, and Tree-LSTM [4]. For training the model, in order to construct and initialize the input word vectors and the input adjacency matrix, 300-dimensional GloVe vectors are utilized, and the Stanford CoreNLP is used to obtain the part-of-speech, named entity recognition annotations, and to generate the dependency parse trees [11]. The pruning technique plays an essential role in information filtering. The cross-entropy loss function with l_2 regularization, Stochastic Gradient Descent (SGD) method with learning rate 1.0, and dropout operation through randomly omitting half of the neurons are applied in this comparative experiment. When moving to the GCN Step, a 2-layer GCN model with ReLU function is selected, and the output comes from two feedforward layers at the final of this experiment.

In addition to the above experiment, an ablation experiment is made to reveal the effectiveness of the corn of change: adding position embedding vector p and summery vector q, one of the outputs of Bi-LSTM. They are removed successively to explore the contribution degree of them. When changing, this paper keeps other parameters and stages unchanged to increase the experiment's credibility. Since the effect and speed of the neural network model training depends on selecting appropriate optimizers, this article also analyses the different outcomes between optimizers, performing a comparative experiment on SGD and Adam optimizers [13]. The test will run ten times to avoid the contingency and

DOI: 10.54254/2755-2721/5/20230712

consider an average value. Lastly, all experiments will leverage the F_1 score as an index to judge the model's accuracy, and the data will be presented in table form.

Table 1. The performance with the different models on the same dataset using the F_1 score as an accuracy indicator.

Model	F_1
LR	59.4
SDP-LSTM	58.7
Tree-LSTM	62.4
PA-LSTM	65.1
GCN	64.0
C-GCN	66.4
PA-GCN	**66.6**

Table 2. The results evaluated on TACRED dev set after an ablation test.

Model	F_1
w/o summary vector q	66.5
w/o position vector p	66.3
PA-GCN	**66.6**

Table 3. A comparison under two different optimizers and the result is also expressed by F_1 scores.

optimizer	F_1
Adam	66.6
SGD	66.6

3.3. Result and analysis

Model comparison. Table 1 demonstrates that the models based on GCN are superior to those mainly using LSTM, maybe due to the later advent of GCN compared with LSTM on the relation extraction task. As time goes by, this series of models has also been improved recently. It is notifiable that when the model combines both of them, the F_1 scores also increase. Concretely, PA-LSTM obtains superior performance over the other models based on LSTM. The addition of a position-aware mechanism shows a remarkable improvement of 6.8% and 2.7% F_1 scores over the SDP-LSTM and the Tree-LSTM model, respectively. As expected, the PA-GCN, the proposed model also containing position attention mechanism, outperforms the other models in the experiment by at least 0.2% F_1 score, achieving absolute promotion of 66.6%, and there is a 2.6% increase on the traditional GCN model. All in all, this experiment demonstrates its ability on relation extraction tasks and improve C-GCN. It can be hypothesized that the improvement from C-GCN to PA-GCN can be attributed to the addition of position vector p and summery vector q because it is the most apparent difference between the two models. The next step of this paper, therefore, will make an ablation experiment to explore their influence of them.

Model ablation. It is worth noting that there is a limitation of the effect of summary vector q, and a huge of time is wasted for adjusting 0.1% deviation. In most cases, the impact of the summary vector q is hard to measure. On the contrary, the position embedding vectors p_i^s and p_i^o have an apparent influence but do not reach a 1.1% F_1 score, as shown by Zhang et al. [6]. In other words, they did not significantly contribute to this experiment. For an intuitive understanding, the other operation mentioned in section 2.3 can also make the contribution from different aspects and remedy their lacks to a certain extent. For instance, LSTM can also capture the position features only in a single direction. Consequently, the appearance of the position vector is just the icing on the cake. Last but not least, maybe for the difference in training hardware between this paper and Zhang's, after removing both summary vector q and position vector p, the model cannot reach the F_1 score of 66.4%. In contrast, it can be considered a reasonable error.

DOI: 10.54254/2755-2721/5/20230712

Optimizers analysis. For selecting a good optimizer, table 3 compares the effect of applying two different optimizers; it can be found that the results of them are similar to the PA-GCN model, which means the outcomes of them are both approximately 66.6% F_1 score. The result proves that selecting one of these optimizers has little bearing on the experimental results. Empirically, it can be reasonably speculated that exploring a better approach to improve model performance through optimizer selection has a common necessity.

4. Conclusion

This paper gave a comprehensive and detailed review of the GCN and C-GCN models and introduced the Position-Aware Graph Convolutional Network model in three steps. The core thought of it is to use the mechanism of position attention and the structure of GCN. Moreover, it adopts bidirectional LSTM that provides a summary vector to adjust the mixed weight. After finishing the experiments on RACTAD Datasets, the results show that verifies the proposed model having a higher efficiency than others mentioned in this paper in handling relation extraction tasks and draws a conclusion that it is an effective way to improve the C-GCN model after comparing its performance with others and analysis. Unfortunately, although PA-GCN has higher accuracy, the model is more time-consuming, and the complexity is not optimistic. In future work, it can be believed that a more convenient and superior model can remedy those shortcomings, making good use of Position, Global Information, and other mechanisms sufficiently, created for a better approach to deal with relation extraction task.

References

[1] Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, Jun Zhao, et al. 2014. Relation classification via convolutional deep neural network. In Proceedings of the 24th International Conference on Computational Linguistics (COLING 2014), pages 2335– 2344.

[2] Xu, K., Feng, Y., Huang, S., & Zhao, D. (2015). Semantic relation classification via convolutional neural networks with simple negative sampling. arXiv preprint arXiv:1506.07650.

[3] Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., & Xu, B. (2016, August). 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) (pp. 207-212).

[4] Zhang, Y., Qi, P., & Manning, C. D. (2018). Graph convolution over pruned dependency trees improves relation extraction. arXiv preprint arXiv:1809.10185

[5] Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Positionaware attention and supervised data improve slot filling. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017).

[6] Gu, S., Zhang, L., Hou, Y., & Song, Y. (2018, August). A position-aware bidirectional attention network for aspect-level sentiment analysis. In Proceedings of the 27th international conference on computational linguistics (pp. 774-784).

[7] Marcheggiani D, Titov I. Encoding sentences with graph convolutional networks for semantic role labeling. 2017.arXiv preprint arXiv:1703.04826

[8] Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.

[9] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. GloVe: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014), volume 14, pages 1532– 1543.

[10] Graves, Alex, Navdeep Jaitly, and A-R Mohamed. 2013. Hybrid speech recognition with deep bidirectional LSTM. In IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU). pages 273–278.

[11] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In Association for Computational Linguistics (ACL) System Demonstrations, pages 55–60.

DOI: 10.54254/2755-2721/5/20230712

[12] Stoica, G., Platanios, E. A., & Póczos, B. (2021, May). Re-tacred: Addressing shortcomings of the tacred dataset. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 35, No. 15, pp. 13843-13850).

[13] Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In Proceedings of ICLR.