IIAI Open Conference Publication Series
IIAI Letters on Informatics and Interdisciplinary Research
Vol. 003, LIIR080
DOI: https://doi.org/10.52731/liir.v003.080

Graph Convolutional Networks for Link Prediction in Argument Structure Extraction

Shiyao Ding *, Takayuki Ito *

Abstract

Argument structure extraction is a fundamental problem in argumentation mining which aims at automatically constructing arguments from unstructured textual documents. Re-garding each argument as a node, then predicting relationships of those nodes is a key task for argument structure extraction, which is called link prediction. Graph neural networks (GNN), as an efficient approach to cope with graphical data, have shown good performances on link prediction tasks. However, they usually focus on predicting whether parts of links really exist in one argument graph, rather than predicting existence of the links between any two nodes, though it is more practical in real applications of argument mining. Thus, in this paper our goal is to predict links between any two nodes given only nodes information. This task is more difficult than the traditional link prediction tasks since the prediction scale exponentially increases with node number. We propose a GNN based model for link predic-tion which outputs the embedding of all the nodes. Then, the probability of link existence between any two nodes is calculated by their embedding results. Finally, we use a dataset of essay comments to perform evaluations and the results confirm the effectiveness of our proposed method.

Keywords: Argumentation mining, Argument structure extraction, Link prediction, Graph neural networks.

1 Introduction

Argument structure extraction is a fundamental problem in argumentation mining which aims to automatically structure arguments from unstructured textual documents [1]. Argument structure extraction has a broad application in many fields [2] . F or in stance, large-scale online discussions on the Web such as D-Agree [3][4] and Slack usually require to process argument structure extraction for efficient d i s cussion. To c onstruct a t ree-like discussion structure, argument structure extraction majorly has two types of tasks : node classification and link prediction [5] [6]. In node classification tasks, each argument is as a node and the goal is to accurately classify a label such as topic, idea and issue for each node. In link prediction tasks, the goal is to accurately predict whether links exist among the nodes.

* Kyoto University, Kyoto, Japan

As stated above, argument structure naturally corresponds to graphical data. In recent years, graph neural networks (GNN), as an effective approach to process graphical data [Wu 2020], have been studied well. Various types of GNN such as graph convolutional networks (GCN) [7] and graph Attention Networks(GAT) [8] are proposed and have achieved good performances on its fundamental tasks of node classification and link prediction. Correspondingly, GNN can also be applied to the tasks of argument structure extraction and have achieved good performances. In the existing research, only a part of links are known and are used to predict the other part of links. While it requires to predict any possible link with only node information in some real-world link prediction tasks of argumentation structure extraction.

Thus, the goal of this paper is to predict the link possibilities of any two nodes under only node information. This task is difficult due to the large prediction space where the number of link predictions exponentially increases with the number of nodes. We propose a GCN-based any link prediction model that uses GCN to obtain each node (argument) embedding, then the link of any two nodes is predicted based on those nodes' embedding results. Finally, we use a dataset of essay comments from essayforum.com which includes 90 essays with 2,071 sentences and 25,584 words to evaluate our proposed method. The results show that our proposed method can achieve a good accuracy on the link prediction tasks.

2 Related Work

Argument structure extraction, as a classic task in argumentation mining which has attracted much attention [1]. For instance, Suzuki et al. propose an approach that includes two steps which are node extraction and link extraction [5]. Specifically, they employ bidirectional long short-term memory (Bi-LSTM) for both of node extraction and link extraction tasks. As for the link prediction task, they only consider to predict part of links given some other link information. They do not consider to perform the tasks with only node information and to predict all possible link existences.

In another hand, GNN is a classical deep neural network for coping with graphical data, which is also called DNN running on graph. Many classical GNNs have been proposed to cope with different features of tasks. One classic GNN is GCN where convolution process of graph data is used by adding the weighted feature values of adjacent nodes to the feature value of one node in the graph. Through this process, the feature values of each node include the node itself and its adjacent nodes. Then various variants of GCN have been proposed such as relational-GCN where multiple relationships among nodes are considered. Link prediction, as a classical task in GNN research, has been studied in many works such as social networks [9], biomedical networks [10] and recommendation systems [11]. However, they usually train GNN on one single graph while there are multiple argument graphs required to be constructed in our goal, which is more difficult.

3 Problem

In this section, we state the problem of argument structure extraction on various graphs. An argument structure can be described as a graph

$$G = < N, E >, \tag{1}$$

where

$$N = \{n^1, n^2, ..., n^{|N|}\} \tag{2}$$

is the set of arguments with each argument n^i that can be a sentence of natural language, and each argument belongs to one of labels, i.e., $lab^j \in Lab$ such as issues, opinions, pros and cons.

$$E = \{(n^i, n^j) | n^i \prec n^j, n^i \in N, n^j \in N\} \tag{3}$$

is the set of edges with the edge (n^i, n^j) exists if argument n^i has a relationship with argument n^j, i.e., $n^i \prec n^j$. For instance, if n^i is "Are there some advice to improve city" and n^j is "Construct more public infrastructures", then edge (n^i, n^j) will exist since n^j supports n^i.

However, the information of graph G usually cannot be completely known, which requires to complement the unknown information. It is majorly classified into two types of tasks: node classification task which complements the information of node labels and link prediction task which complements the information of link. Specifically, in the node classification task, given the graph $G =< N, E >$ without label information, the goal is to predict each node's label. It means to find a function f that outputs each node's label accurately, i.e., $f(n^i) = lab^j$. Moreover, in some cases the information of E is also unknown, which makes the classification task more difficult. The complexity of node classification task linearly increases with the number of nodes i.e, $O(|N|)$.

In the link prediction task, only the information of node N is given, the goal is to predict the link existence of any two nodes. It means to find a function g that outputs the probability of link existence of any two nodes. i.e., $g(n^i, n^j) = p^{ij}$. As shown in the left part of Figure 1, 16 nodes with three labels are given and are inputted to the link prediction model. Then, the output of the link prediction model is the probabilities p^{ij} of the link existences between any two nodes, as shown in the right part of Figure 1. Since it requires to predict the links of any two nodes, whose complexity exponentially increases with the number of nodes $O(|N|^2)$.

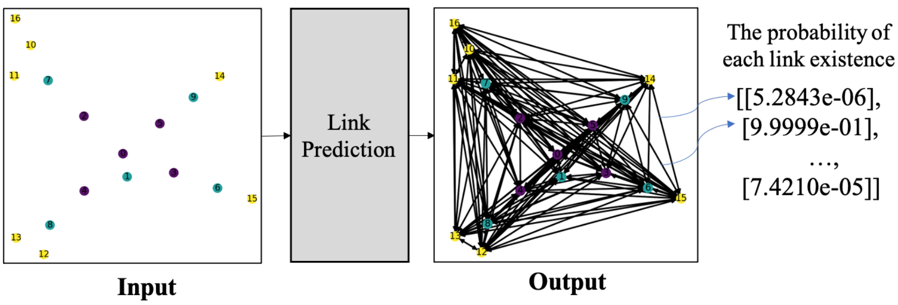

Figure 1: An example of prediction between any two nodes.

4 Algorithm

In this section, we introduce our GCN model for link prediction given only node information N without label information which includes two parts: encoder and decoder. In order to make node information can be tackled by GCN, we first embed each node n^i as a vector v^i with numerical values. There are a lot of methods for embedding natural language like Word2Vec. In this paper, we use bidirectional encoder representations from transformers (BERT) to embed each node as a vector with 1024 dimensions.

Each directed edge represents the relationship from source node to destination node. Then, we construct feature matrix $X \in \mathbb{R}^{|N|\times 1024}$ where each row i corresponds one vector v^i and adjacency matrix A describes a link whose element a^i_j corresponds to 1 if link (n^i, n^j) exists, 0 otherwise. Let us consider the argument graph shown in Figure 1 which includes 16 nodes where feature matrix X, adjacency matrix A and degree matrix D are given by

$$X = \begin{bmatrix} v^1_1 & v^1_2 & \dots & v^1_{1024} \\ v^2_1 & v^2_2 & \dots & v^2_{1024} \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ v^{16}_1 & v^{16}_2 & \dots & v^{16}_{1024} \end{bmatrix} A = \begin{bmatrix} a^1_1 & . & . & . & a^1_{16} \\ . & . & & & . \\ . & & . & & . \\ . & & & . & . \\ a^{16}_1 & . & . & . & a^{16}_{16} \end{bmatrix} D = \begin{bmatrix} d_{11} & 0 & 0 & 0 & 0 \\ 0 & d_{22} & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & d_{1515} & 0 \\ 0 & 0 & 0 & 0 & d_{1616} \end{bmatrix}.$$

Further, based on the above matrices we can construct a matrix $\widetilde{A}$ which includes relationship information between nodes with different importance.

$$\widetilde{A} = D^{\frac{1}{2}}(D-A)D^{\frac{1}{2}}. \tag{4}$$

The goal is to obtain the embedding result of each node by GCN. Then we can use the embedding results of any two nodes to predict the probability of link existence between them. The specific calculation process of embedding in one convolutional layer is stated as

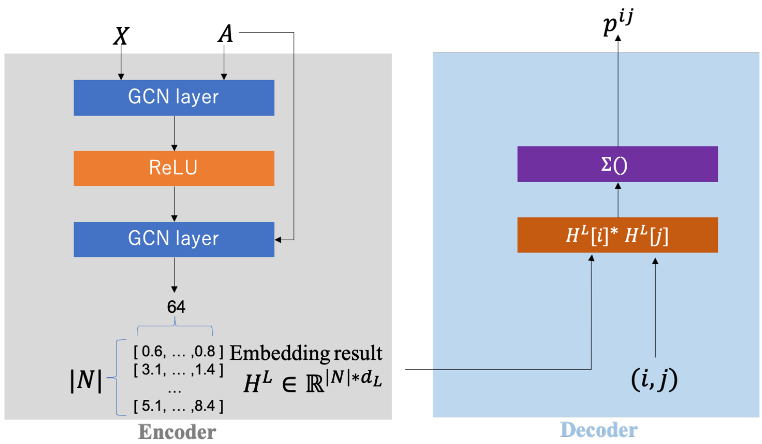

Figure 2: The framework of our proposed method.

follows. We take H^0=X as the input of the first GCN layer and obtain the latent variable H^1 of the next layer and then repeat to this process. In each layer, we use adjacency matrix to update each node's latent features by multiply $\widetilde{A}H^l$. Then, we let $\widetilde{A}H^l$ multiply a parameter matrix W^l which can be updated in the direction that can achieve the learning goal. Moreover, we throw $\widetilde{A}H^lW^l$ into an activation layer δ; where the ReLU function is used in this paper. Such kind of encoding process, shown in the left part of Figure 2 (encoder), can be represented as follows.

$$H^{l+1} = \delta(\widetilde{A}H^lW^l), \tag{5}$$

where $H^l \in \mathbb{R}^{|\mathbf{N}|\times d_l}$ is the latent variable in the l-th layer with d_l dimensions. Finally we can obtain the latent variable H^L of the final layer L as the embedding result for each node, which is called process of encoder.

As for the process of decoder, the probability of link existence depends on the inner product of the two node embedding results, which is calculated by the following equation, shown in the right part of Figure 2,

$$p^{ij} = \sum(H^L[i] * H^L[j]), \tag{6}$$

where $H^L[i]$ and $H^L[j]$ represent the embedding results of node v^i and v^j. Finally, the goal is to minimize the following loss function.

$$\min \sum_{ij} \log(bin(e^{ij}) - p^{ij}), \tag{7}$$

where $bin(e^{ij})$ is 1 if link e^{ij} exists, 0 otherwise (in this paper, we do not distinguish the type of links). As for the specific training process, we usually only have accurate graphs which are called positive samples. Thus, we need also to generate some negative samples where some links that do not really exist will be generated. Then, we randomly chose positive samples and negative samples to train our model.

5 Evaluation

In this section, we evaluate our proposed method on a dataset of essay comments from essayforum.com [12]. It includes 90 essays with 2,071 sentences 25,584 words. First, we show the result of training on one single graph by GCN as shown in Figure 3. It shows

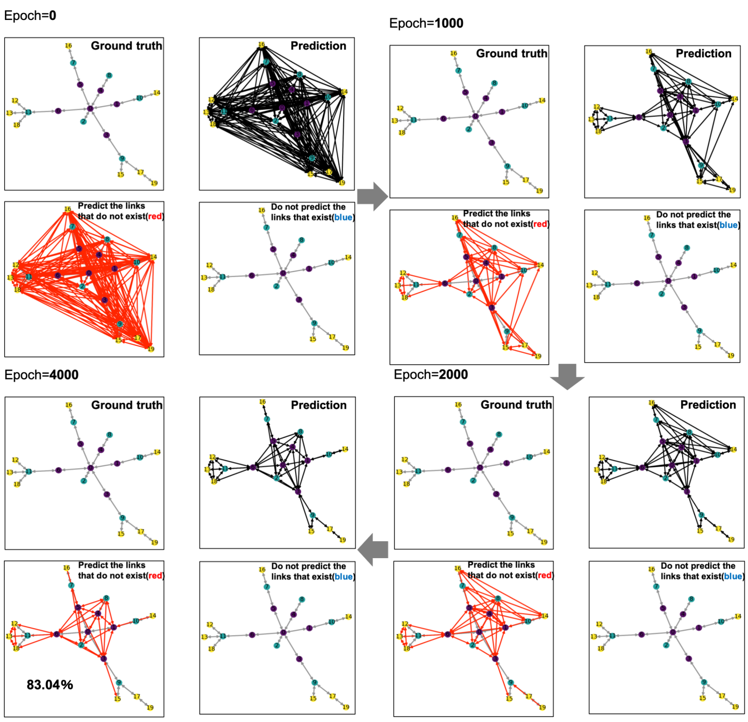

Figure 3: The training process on one single graph.

the training results of epochs of 0, 1000, 2000, 6000. As for each epoch result, it consists of four parts. The upper-left part shows the ground truth of the argument graph where different colors correspond to different labels (e.g., topics and opinions). The upper-right part shows the link prediction results. The lower-left part shows the links do not really exist but are wrongly predicted which are colored in red. The lower-right part shows the links do really exist but have not been predicted which are colored in blue. We can see that prediction accuracy is improved along with the increase of learning epoch. Finally, our proposed method achieves an accuracy of 83.04%. Then, we train on multiple graphs and one of the results is shown in Figure 4. Since forcing one single GCN model to cope with multiple graphs is more difficult where the final accuracy decreases to 62.09%.

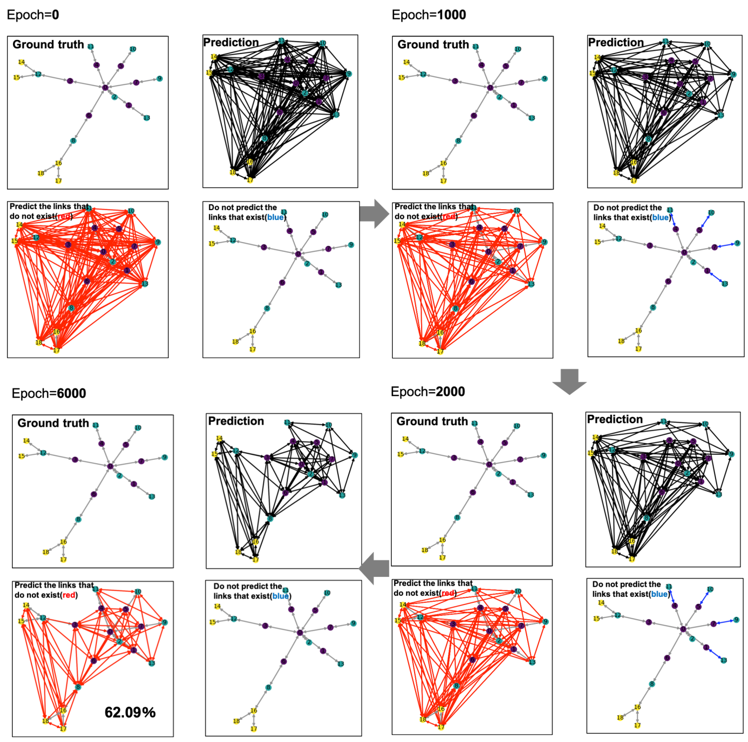

Figure 4: The training process on multiple graphs.

6 Conclusion

This paper studied the problem of link prediction between any two nodes in argument structure abstraction given only node information. Our solution was to propose a GCN based model that can obtain the embedding results of all the nodes, then the probability of link existence between any two nodes can be calculated by the embedding results. We ran several experiments and the results confirm that our proposed method can achieve an accuracy around 80% on a single graph and an accuracy around 60% on multiple graphs. We plan to improve the prediction accuracy on multiple graphs in the future.

Acknowledgement

This work was supported by JST CREST Grant Number JPMJCR20D1, Japan.

References

[1] M. Lippi and P. Torroni, "Argumentation mining: State of the art and emerging trends," *ACM Transactions on Internet Technology (TOIT)*, vol. 16, no. 2, pp. 1–25, 2016.

[2] T. Ito, Y. Imi, T. Ito, and E. Hideshima, "Collagree: A faciliator-mediated large-scale consensus support system," *Collective Intelligence*, vol. 2014, pp. 10–12, 2014.

[3] T. Nishdia, T. Ito, T. Ito, E. Hideshima, S. Fukamachi, A. Sengoku, and Y. Sugiyama, "Core time mechanism for managing large-scale internet-based discussions on collagree," in *2017 IEEE International Conference on Agents (ICA)*. IEEE, 2017, pp. 46–49.

[4] T. Ito, R. Hadfi, and S. Suzuki, "An agent that facilitates crowd discussion," *Group Decision and Negotiation*, vol. 31, no. 3, pp. 621–647, 2022.

[5] S. Suzuki, N. Yamaguchi, T. Nishida, A. Moustafa, D. Shibata, K. Yoshino, K. Hiraishi, and T. Ito, "Extraction of online discussion structures for automated facilitation agent," in *Annual Conference of the Japanese Society for Artificial Intelligence*. Springer, 2019, pp. 150–161.

[6] T. Ito, R. Hadfi, J. Haqbeen, S. Suzuki, A. Sakai, N. Kawamura, and N. Yamaguchi, "Agent-based crowd discussion support system and its societal experiments," in *International Conference on Practical Applications of Agents and Multi-Agent Systems*. Springer, 2020, pp. 430–433.

[7] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[8] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *stat*, vol. 1050, p. 20, 2017.

[9] D. Liben-Nowell and J. Kleinberg, "The link prediction problem for social networks," in *Proceedings of the twelfth international conference on Information and knowledge management*, 2003, pp. 556–559.

[10] Y. Long, M. Wu, Y. Liu, Y. Fang, C. K. Kwoh, J. Chen, J. Luo, and X. Li, "Pre-training graph neural networks for link prediction in biomedical networks," *Bioinformatics*, vol. 38, no. 8, pp. 2254–2262, 2022.

[11] D. Zheng, M. Wang, Q. Gan, X. Song, Z. Zhang, and G. Karypis, "Scalable graph neural networks with deep graph library," in *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, 2021, pp. 1141–1142.

[12] C. Stab and I. Gurevych, "Identifying argumentative discourse structures in persuasive essays," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 46–56.