Proceedings of Machine Learning Research vol 245:1–9, 2024 2024 International Conference on Machine Learning and Intelligent Computing

Thesis Reviewer Recommendation Based on Multi-Graph Neural Networks

Tao Wang
Nanjing University of Science and Technology, Nanjing, China

Peng Fu* FUPENG@NJUST.EDU.CN
Nanjing University of Science and Technology, Nanjing, China

Jing Zhang
Jiangsu Guidgine Educational Evaluation Inc, Nanjing, China

Kaixuan Chen
Jiangsu Guidgine Educational Evaluation Inc, Nanjing, China

Editors: Nianyin Zeng and Ram Bilas Pachori

Abstract

Thesis review is a crucial step in ensuring the quality of academic theses, and accurately recommending reviewers for theses is currently a problem that needs to be addressed. When reviewer information is incomplete, it is difficult to achieve good recommendation results. This paper proposes a Multi-Graph Neural Network algorithm for review reviewer recommendation in thesis review. Based on the keyword-reviewer bipartite graph, a graph neural network model is constructed. By utilizing graph neural networks, high-order relationships between reviews and keywords can be explored, enabling the discovery of reviews' implicit research interests and expanding their research interests to some extent. Additionally, incorporating keyword-keyword interaction graphs and review-review interaction graphs allows for information exchange operations in the two graphs separately, enhancing the representation of keywords and reviews. We conducted experiments on real thesis reviews and compared the proposed algorithm with other recommendation algorithms. The results show that the proposed algorithm achieves favorable results across various evaluation metrics, demonstrating the effectiveness of the algorithm presented in this paper.

Keywords: Thesis Reviewer Recommendation, Graph Neural Networks, Recommendation Algorithms, Bipartite Graph

1. Introduction

Thesis review is a crucial step in ensuring the quality of academic theses and plays an important role in enhancing the quality of graduate education. With the continuous increase in the number of graduate students, the automation of thesis review has become necessary and urgent. In the process of automating thesis review, there are challenges (Zhao and Zhang, 2022) to overcome, one of which is how to automatically assign reviewers.

In recent years, the quality of reviewer recommendations has attracted increasing attention (Shah, 2021) from researchers, mainly focusing on text-based methods. Protasiewicz (2014) extract keywords based on review knowledge and calculate matching degree using cosine similarity. Abduljaleel et al. (2021) use TF-IDF and cosine similarity to calculate matching degree. Tan et al. (2021) propose a WSIM model based on language models and LDA (Blei et al., 2003) to calculate the similarity between papers and reviews. Choi et al. (2023) used TextRank to extract feature sets from the txt. With the development of deep learning, there are various neural network models available for calculating similarity between texts, such as word2vec (Mikolov et al., 2013), RNN (Socher et al., 2014) , BERT (Devlin et al., 2018). Zhao et al. (2018) use word2vec to generate word embeddings for papers and reviews, and calculate the similarity between papers and reviews using Word Mover Distance.

The above-mentioned methods all calculate the similarity based on text information. When review information is incomplete, it is difficult to achieve good recommendation results. Therefore, this paper proposes a recommendation method based on graph models to address this issue. Recommendation systems based on graph models have been rapidly developing (Wang et al., 2021). Wang et al. (2019) proposed a method based on GNN (Kipf and Welling, 2016) to encode information in the embedding process, capturing the effects of collaborative filtering. He et al. (2020) proposed a lightweight GCN method that abandons feature transformations and non-linear activations. Mao et al. (2021) suggested an ultra-simplified graph neural network for recommendation, skipping the infinite-layer explicit message passing to achieve efficient recommendations.

In this paper, we propose a new method called Multi-Graph Neural Networks (MultiGNN) for reviewer recommendation. By utilizing graph neural networks, we can uncover high-order relationships between reviews and keywords, thus identifying reviews' implicit research interests. This method is a deep neural network model that integrates multiple graph. The main contributions of our work are as follows:

We propose MultiGNN, leveraging Graph Neural Networks to fully explore high-order relationships between reviews and keywords. By learning embedding of reviews and keywords through different propagation paths in various relational data.

We construct keyword-keyword and review-review interaction graphs to obtain associations between keywords and reviews. This can enrich the embedding of keywords and reviews, thereby enhancing the quality of recommendations.

2. Methods

In this section, we will discuss in detail the framework of MultiGNN, as shown in the Figure 1. The model proposed in this paper takes the keyword list ks and the reviewing review r as input, and outputs the predicted probability value ŷks, which indicates the probability that review r can review the paper. To accomplish this task, the model consists of three main parts: the embedding layer, the information interaction layer, and the prediction layer.

2.1. Embedding Layer

The embedding layer can obtain embedding vectors for keywords and reviews. The embedding vectors are represented by learnable parameter vectors, and their main function is to transform sparse feature vectors into dense embedding vectors, reducing the dimensionality of the model input.The keyword embedding vector and review embedding vector can be represented as:

$$e_{k_i} = W_k k_i; \;\; e_{r_i} = W_r r_i \tag{1}$$

where $\mathrm{W_k} \in R^{m_k \times n}$ and $\mathrm{W_r} \in R^{m_r \times n}$ is the learnable embedding matrices for encoding keywords and reviews, n represents the dimensionality of the embedding vectors.

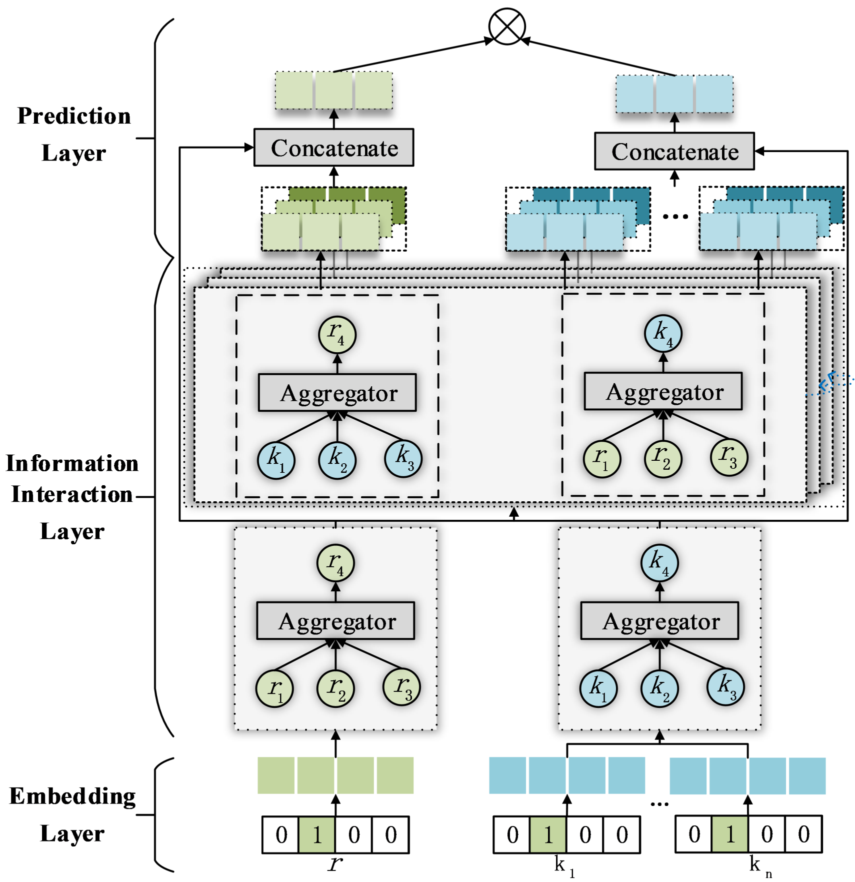

Figure 1: The architecture of Multi-Graph Neural Network model for reviewer recommendation.

2.2. Information Interaction Layer

The information interaction layer obtains the representation vectors of keywords and reviews. Considering the complex interaction between keywords and reviews, a bipartite graph neural network is constructed to handle the interaction data. Additionally, this paper proposes to construct keyword-keyword and review-review graphs. Finally, output the keyword and reviewer representation vectors.

2.2.1. BUILDING KEYWORD-KEYWORD AND REVIEW-REVIEW INTERACTION GRAPHS

The construction of interaction graphs is reflected through co-occurrence frequency. To build the interaction graph, it is necessary to calculate the co-occurrence frequency of keyword or review pairs. After obtaining the frequency matrix, edges can be determined by manually setting a threshold. The definition of the edge formula is as follows:

$$KK_{k_1,k_2}, RR_{r_1,r_2} = \begin{cases} 1, fre\left(k_1, k_2\right) \geq \alpha \\ 0, fre\left(k_1, k_2\right) < \alpha \end{cases} \tag{2}$$

where KK and RR is the keyword-keyword and review-review interaction graphs, α denotes the threshold of frequency.

2.2.2. KEYWORD-KEYWORD AND REVIEW-REVIEW INFORMATION INTERACTION

The information interaction between adjacent nodes can be achieved through node information aggregation operations. Specifically, for keyword nodes, information is passed from neighboring keyword nodes to the target keyword node. The review-review interaction follows a similar approach. The specific formula is defined as:

$$e_{kk}^{1} = \sum_{i \in N_{kk}} \frac{1}{\sqrt{|\ N_{kk}\ | N_{kk_i}|}} e_{kk_i}^{0};\ \ e_{rr}^{1} = \sum_{i \in N_{rr}} \frac{1}{\sqrt{|\ N_{rr}\ | N_{rr_i}|}} e_{rr_i}^{0} \tag{3}$$

where $\mathrm{N_{rr}}$ and $\mathrm{N_{kk}}$ is the sets of neighboring nodes for the review and keyword nodes. Then combined with the initial embedding to obtain the final node embedding vectors. The specific formula is defined as:

$$e_{kk}^{*} = e_{kk}^{0} + e_{kk}^{1};\ \ e_{rr}^{*} = e_{rr}^{0} + e_{rr}^{1} \tag{4}$$

where $\mathrm{e_{kk}^{*}}$ and e_{rr}^{*} is the embedding vectors outputted after the interaction operations.

2.2.3. BUILDING KEYWORD-REVIEW BIPARTITE GRAPH

If the list of keywords in papers published or reviewed by reviewer r, it is associated with these keywords. The Keyword-Review bipartite graph is treated as an undirected graph, and the definition of edges between nodes is as follows:

$$KR_{k,r} = \begin{cases} 1, k \in N\left(r\right) \\ \ 0,\ other \end{cases} \tag{5}$$

where KR is the keyword-review bipartite graph, and N(r) denotes the immediate neighboring set directly connected to review r.

2.2.4. KEYWORD-REVIEW INFORMATION INTERACTION

The information interaction between keyword nodes and review nodes is achieved through weighted sum operations. We explore higher-order connection information by setting up multiple layers of embedding propagation. Specifically, this information is passed from neighboring review nodes to the target keyword node; this information is passed from neighboring keyword nodes to the target reviewer node. The formula is defined as:

$$e_{k}^{l+1} = \sum_{r \in N_k} \frac{1}{\sqrt{|\ N_k\ | N_r|}} e_{r}^{l};\ \ e_{r}^{l+1} = \sum_{k \in N_r} \frac{1}{\sqrt{|\ N_r\ | N_k|}} e_{k}^{l} \tag{6}$$

where $\mathrm{N_k}$ and $\mathrm{N_r}$ is the sets of neighboring nodes for keyword k and review r.

2.3. Prediction Layer

The prediction layer is primarily responsible for calculating the interaction score between keywords and reviews.by interacting with the representation vectors of keywords and reviews, the final probability value is obtained.

After L layers of network propagation, multiple embedding vectors for keywords and reviews can be obtained. Since the embedding vectors in different layers emphasize messages from different connection paths. It is necessary to merge them together namely the vectors of keywords and reviewer. The formula is as follows:

$$e_r^* = concat\left(e_r^0, e_r^1, \ldots, e_r^L\right) \tag{7}$$

$$e_{ks}^* = \sum_{i}^{|ks|} concat\left(e_{ks_i}^0, e_{ks_i}^1, \ldots, e_{ks_i}^L\right) \tag{8}$$

where concat($\cdot$) is the concatenation operation. e_{ksi}^* is the representation vector of the i-th keyword in the keyword set.

Finally, we use the inner product operation between the representation vectors of the keyword set and the review's representation vector for model prediction $\hat{\mathrm{y}}_{\mathrm{ks}}$. The specific formula is as follows:

$$\hat{\mathrm{y}}_{ks} = e_r^{*T} e_{ks}^* \tag{9}$$

To learn the model parameters, this paper selects BPR loss as the training objective. BPR loss is defined as the negative log likelihood loss:

$$Loss = \sum_{(k,p,n)\in T} - In\ sigmoid\left(\hat{\mathrm{y}}_{kn} - \hat{\mathrm{y}}_{kp}\right) + \lambda \parallel \theta \parallel_2^2 \tag{10}$$

where $\mathrm{T} = \{(\mathrm{k},\mathrm{p},\mathrm{n}) \mid (\ k,p\ ) \in T^+, (\ k,n\ ) \in T^-\}$ is the training data, where T^+ denotes the real interaction data and T^- is the data obtained through negative sampling;

3. Experiments

3.1. Dataset and Metrics

The dataset in this study consists of real thesis review data from relevant universities between 2021 and 2023. The data includes information on theses, reviews, and interactions. After deduplication, filtering, merging, and other operations on the data, we obtained 23,427 interaction data entries, 46,465 reviewers, 13,044 theses, and 41,745 keywords. The dataset is randomly split into training set (90%), test set (10%).

According to the task of this study, in this experiment, we choose recall@k and ndcg@k as evaluation. recall@K is the recall rate of the true reviewing reviews in the recommendation list. ndcg@K is used to measure the accuracy of the position of true reviewing reviews in the recommendation list;

3.2. Experimental Results and Analysis

This section will demonstrate the experimental results and analyze them from multiple perspectives to prove the effectiveness of the proposed method in this paper. This includes comparing the proposed method with existing models(The keyword-based methods include BM25 and BERT. Based on graph-based methods include NGCF, LightGCN, and UltraGCN.), conducting ablation experiments to verify the effectiveness of the modules, and discussing the impact of parameters on the experiment.

This paper implements the proposed method and comparative methods using PyTorch. The parameters of the proposed model are as follows: batch size is 2048, dropout rate is 0.3, the dimensions of embedding vectors and embedding layers are both 128, Xavier initializer is used for embedding initialization, Adam optimizer is used for training the model with a learning rate of 0.001.

Table 1: The performance comparison of different methods.

Method	recall@5	recall@10	ndcg@5	ndcg@10
BM25	0.615	0.693	0.612	0.679
BERT	0.625	0.735	0.633	0.661
NGCF	0.619	0.741	0.620	0.663
LightGCN	0.672	0.781	0.680	0.726
UltraGCN	0.679	0.807	0.702	0.731
MultiGNN	**0.727**	**0.828**	**0.725**	**0.767**

From the comparison of experimental results in the Table 1, we can see that except for the NGCF model performing worse than the BERT based on word matching in the recall@5 and ndcg@5, all other graph-based methods show performance improvements over the word matching-based BERT. This proves that introducing relational models can enhance the recommendation effectiveness. Among the results, we observe that LightGCN and UltraGCN outperform NGCF, especially with an improvement of over 23% in the recall@5 metric, demonstrating the advantages of LightGCN and UltraGCN. The proposed MultiGNN method performs better than the other three graph-based methods in all metrics, thereby demonstrating the effectiveness of the proposed model in this paper.

Table 2: The effect of different modules on model performance.

Method	recall@5	recall@10	ndcg@5	ndcg@10
SingleGNN	0.672	0.781	0.680	0.726
$\text{DualGNN}_{\text{kk}}$	0.694	0.811	0.693	0.742
$\text{DualGNN}_{\text{ee}}$	0.693	0.802	0.681	0.726
MultiGNN	**0.727**	**0.828**	**0.725**	**0.767**

We validate the effectiveness of the proposed modules through ablation experiments. Specifically, the SingleGNN model removes the keyword-keyword interaction module and the review-review interaction module, the $\text{DualGNN}_{\text{kk}}$ model removes the review-review interaction module, and the $\text{DualGNN}_{\text{ee}}$ model removes the keyword-keyword interaction module. The ablation experiment results are presented in the Table 2, showing that the performance of both $\text{DualGNN}_{\text{ee}}$ and $\text{DualGNN}_{\text{kk}}$ is better than SingleGNN, thereby proving the effectiveness of the proposed review-review interaction module and keyword-keyword interaction module. Furthermore, from various evaluation metrics, MultiGNN outperforms both $\text{DualGNN}_{\text{kk}}$ and $\text{DualGNN}_{\text{ee}}$, indicating that integrating the review-review interaction module with the keyword-keyword interaction module leads to better performance.

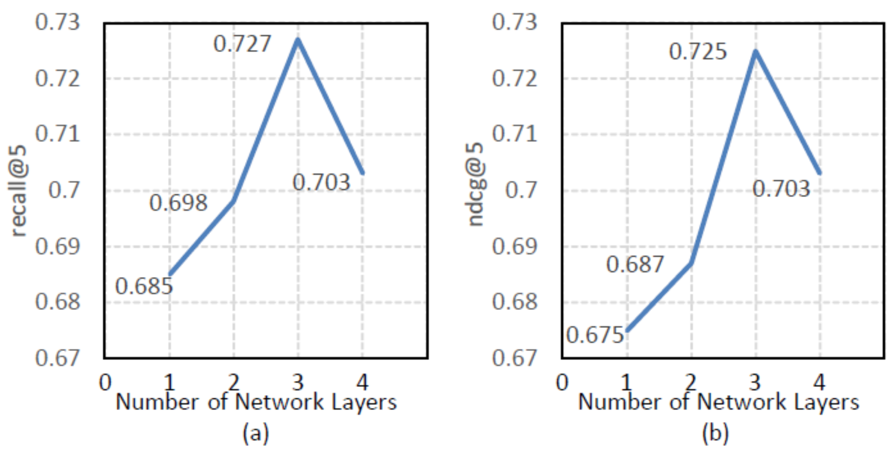

Figure 2: The effect of the number of network layers on model.

To study the impact of the number of network layers on the experiment, this paper conducted experiments with different numbers of layers and obtained the experimental results as shown in the Figure 2. From the figure, it can be observed that when the number of network layers is 3, the values are the highest. As the number of layers increases from less than 3, the model's performance improves. However, when the number of network layers reaches 4, the model's performance decreases. This indicates that more network layers do not necessarily lead to better performance. Having too many layers may result in over smoothing issues, where after multiple rounds of aggregating neighbor node features. Finally, it is concluded that selecting a network layer number of 3 yields the best performance in the model proposed in this paper.

4. Conclusion

This paper proposes a Multi-Graph Neural Network algorithm to address the issue of review reviewer recommendation in academic thesis evaluation. Based on the keyword-review bipartite graph, review-review interaction graph, and keyword-keyword interaction, a graph neural network is constructed to learn embedding of keywords and reviews. Experimental results on review record datasets demonstrate the effectiveness of the model, leading to the following conclusions: introducing relational connections in review reviewer recommendation can enhance the quality of recommendations, and integrating multiple graph structures in graph neural networks can enhance the model's representational power.

Acknowledgments

This research was supported by the Graduate Educational Teaching Reformation Research Project of Nanjing University of Science and Technology under Grant no. 8.

References

Ali Qays Abduljaleel et al. Reviewer assignment using weighted matching and hungarian algorithm. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 12(7):619–627, 2021.

David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

Dong-Hoon Choi, Jin Woo Hyun, and Young Rock Kim. An algorithm for peer reviewer recommendation based on scholarly activity assessment. *IEEE Access*, 11:39609–39620, 2023.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 639–648, 2020.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2016.

Kelong Mao, Jieming Zhu, Xi Xiao, Biao Lu, Zhaowei Wang, and Xiuqiang He. Ultragcn: ultra simplification of graph convolutional networks for recommendation. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pages 1253–1262, 2021.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013.

Jarosław Protasiewicz. A support system for selection of reviewers. In *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 3062–3065. IEEE, 2014.

Nihar B Shah. Wsdm 2021 tutorial on systematic challenges and computational solutions on bias and unfairness in peer review. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 1131–1133, 2021.

Richard Socher, Andrej Karpathy, Quoc V Le, Christopher D Manning, and Andrew Y Ng. Grounded compositional semantics for finding and describing images with sentences. *Transactions of the Association for Computational Linguistics*, 2:207–218, 2014.

Shicheng Tan, Zhen Duan, Shu Zhao, Jie Chen, and Yanping Zhang. Improved reviewer assignment based on both word and semantic features. *Information Retrieval Journal*, 24:175–204, 2021.

Shoujin Wang, Liang Hu, Yan Wang, Xiangnan He, Quan Z Sheng, Mehmet A Orgun, Longbing Cao, Francesco Ricci, and Philip S Yu. Graph learning based recommender systems: A review. In *30th International Joint Conference on Artificial Intelligence, IJCAI 2021. International Joint Conferences on Artificial Intelligence*, pages 4644–4652, 2021.

Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, pages 165–174, 2019.

Shu Zhao, Dong Zhang, Zhen Duan, Jie Chen, Yan-ping Zhang, and Jie Tang. A novel classification method for paper-reviewer recommendation. *Scientometrics*, 115:1293–1313, 2018.

Xiquan Zhao and Yangsen Zhang. Reviewer assignment algorithms for peer review automation: A survey. *Information Processing & Management*, 59(5):103028, 2022.