

An approach of anchor link prediction using graph attention mechanism

Van-Vang Le^{1,4}, Phuong Nguyen Huy Pham^{2,4}, Tran Kim Toai³, Vaclav Snasel⁴

¹Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh, Vietnam

²Faculty of Information Technology, Ho Chi Minh City University of Food Industry, Ho Chi Minh, Vietnam

³Ho Chi Minh University of Technology Education, Ho Chi Minh, Vietnam

⁴Faculty of Electrical Engineering and Computer Science, VSB-Technical University of Ostrava, Ostrava-Poruba, Czech Republic

Article Info

Article history:

Received Jun 17, 2022

Revised Jul 31, 2022

Accepted Aug 12, 2022

Keywords:

Anchor link prediction

Network alignment

Graph attention network

ABSTRACT

Nowadays social networks such as Twitter, LinkedIn, and Facebook are a popular and necessary platform. It is considered a miniature of an actual social network because of its advantages in connecting and sharing information between users. The analysis of data on online social networks has become a field that has attracted a lot of attention from the research community and anchor link prediction is one of the main research directions in this field. Depending on demand, a user can simultaneously participate in many different online social networks, anchor link prediction is a kind of task that determines the identity of a user on many different social networks. In this article, we proposed an algorithm that determines missing/future anchor links between users from two different online social networks. Our algorithm utilizes the graph attention technique to represent the source and target network into the low-dimension embedding spaces, we then apply the canonical correlation analysis to recline their embeddings into same latent spaces for final prediction.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Phuong Nguyen Huy Pham

Faculty of Information Technology, Ho Chi Minh City University of Food Industry (HUFI)

140 Le Trong Tan, Tay Thanh ward, Tan Phu district, Ho Chi Minh City, Vietnam

Email: phuongpnh@hufi.edu.vn

1. INTRODUCTION

Recently, with the diversity of different online social networks, anyone in real-life social network can take part in several online social networks for many different purposes. Most of them participate in these online social networks with the same or similar user properties such as full name, username, and gender. However, in some cases, users may not disclose personal information consistently across different social networks. This leads to using this information to predict anchor links will not be accurate because of the noisy information. Thus, anchor link prediction, which is a task of matching users over social networks, is a major challenge and attracts a lot of attention from the scientific community up to present.

There are many different methods for anchor link prediction problems. The initial studies handled the anchor link prediction problem by exploiting self-defined user profile and user generated contents to measure the similarity to get the prediction result. The traditional methods attempt to align users across online social networks using self-defined user personal profile such as name, gender, age, location [1]-[3], and user's generated content such as tweets, posts, publications [4], [5]. Usually, the methods that follow this approach often use heuristics to process text data and compare similarity. These methods are sensitive to the similarity metrics or self-defined user information, and thus come with limitations due to the imbalance of users' demographic data in different information networks and privacy issues in retrieving user profile information.

Recently, the rapid development of network embedding techniques [6]-[8] has opened a new research trend in the field of anchor link prediction. Network embedding is a task of learning the representation of a node in the network in which it has low dimension than the original one but still preserving the network properties and structure. Following this strategy, some methods [9]-[13] attempt to find the low-dimensional embedding of every node in the source and target online social networks using a graph neural network [14] which is a generalization of convolution neural networks to process some kind of data represented by the graph structure. Few years recently, graph neural networks (GNN) have been considered as a powerful and pragmatic technique for any problem that can be represented by graphs. Therefore, there are many variant models developed based on GNN, such as recurrent GNN [15], convolutional GNN [16], [17], graph auto-encoders [18], [19], and spatial-temporal GNN [20]. Graph attention network [21] is one of the modern models widely applied in fields such as link prediction [22]-[24], node classification [25], node clustering [26], recommendation system [27], [28], information diffusion [29], and in this paper, we apply this technique to resolve the anchor link prediction problem.

MAUIL [30] is an anchor link prediction method which combines multiple embedding techniques to increase the accuracy of the anchor link prediction model. It uses three levels of attribute embedding techniques to preserve the node attributes of the network and use the Line method [31] to embed network information in terms of network structure. Although, this method has obtained great results, which has been demonstrated experimentally to give better results compared to other solutions. However, it also exposes some limitations in terms of performance as well as complexity. Inspired by MAUIL, we propose an anchor link prediction method based on the idea of graph attention mechanism to increase the accuracy and performance of this model. Here the main contributions of our paper:

- We propose a combination method of anchor link prediction to improve the accuracy of the MAUIL method by substituting the network embedding method that is used in MAUIL by a graph attention mechanism to find the embedding of the source and target network.
- We apply canonical correlation analysis to project their representation onto same latent embedding spaces and compute the alignment matrix of nodes between source and target network. The experiment on real life datasets shows that our proposed method is outperforming than the original one.

2. PROPOSED METHOD

Our proposed model is a combination of three modules: multilevel attribute embedding, graph attention network, and regularized canonical correlation analysis (RCCA)-based correlation analysis. Consequently, a total of four embedding matrices (three for attribute-based embedding and one for graph attention network (GAT)-based embedding) are integrated to establish the final embedding of each social networks. As described in the Figure 1, our model can be divided into three steps as shown in:

- At first, we feed user's attributes of source and target network into three-level embedding techniques to compute node embedding which preserves the attribute information of each user in network.
- Then, we use attribute embedding as initial feature vector along with network structure as the input of graph attention mechanism. This process utilizes the contribution of neighbor nodes to the aggregation step to construct the final embedding.
- Finally, we apply canonical correlation analysis to project the source and target embedding into the same latent representation. Then, we compute final network alignment matrix based on the similarity score of embedding vectors between source and target network.

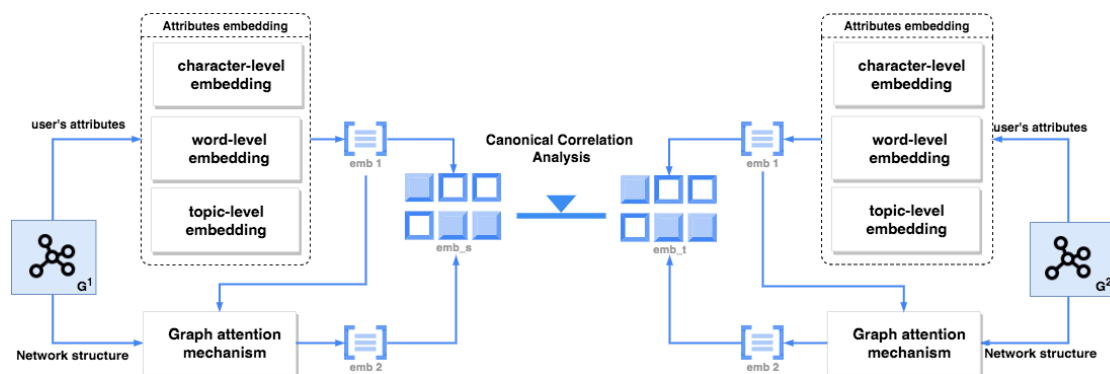


Figure 1. Framework overview for anchor link prediction

2.1. Multi-level attribute embedding techniques

2.1.1. Character-level attribute embedding

This embedding technique aims to learn the representation of node using the text similarity of similar usernames. Text content in the user's name may contain multiple type of characters such as alphabets, numeric, spaces, and special symbols, and generally we see them as tokens. For each username, we count the frequency of each unique token to get a list of tuples $t = [(t_1: f_1), (t_2: f_2), (t_3: f_3), \dots, (t_m: f_m)]$, where t_i is the i^{th} token which appears in username, f_i is the number of occurrences of token t_i in username. During the counting process, we also build a token dictionary which contains all unique tokens that appear in the usernames of all users. Finally, the corresponding count-weighted vector for user v_i is $\overline{X}_i^c = [f_{i1}, f_{i2}, \dots, f_{ip}] \in R^p$ where f_{ij} is the frequency of j^{th} token in the token dictionary which occurs in the username of user v_i , p is the total number of unique tokens in dictionary.

The embedding matrix for all node in each network $X^c = [\overline{X}_1^c, \overline{X}_2^c, \dots, \overline{X}_n^c] \in R^{n \times p}$ is reduced by apply the auto-encoder method which is an essential and powerful technique to reduce the dimensional of data. This work implements a one-layer auto-encoder to integrate the token frequency vectors into distributed embedding space. After this step, we acquire the feature matrix of each network at character-level $P_c = [\overline{P}_1^c, \overline{P}_2^c, \dots, \overline{P}_n^c] \in R^{n \times d}$ where $R^{n \times p}$ is the original high-dimensional and $R^{n \times d}$ is reduced-dimensional character-level representation of character level embedding.

2.1.2. Word-level attribute embedding

This embedding technique aims to learn the representation of node using the similarity of text of similar word group or short sentences such as affiliations, social relationship, and working experiences. We apply the Word2vec model [32], which is one of the most popular techniques for converting texts into feature vectors to embed the characteristics of user at word-level. The attribute embedding at word-level of the user v_i in a network G with n users is denoted by a_i^w and may contain part of the word. We can represent the words in these short sentences using a sequence of m unique words $w = w_1, w_2, w_3, \dots, w_m$. We use them as input corpus to build the vocabulary dictionary, target words, and contextual words list for each target word using a window size. Then, they are fed into the continuous bag of words (CBOW) model [32], a neural network model that predicts the target word by trying to understand the context of the surrounding words.

$$P_i^w = (1 - \lambda)Z_i + \lambda \frac{1}{|N_i|} \sum_{j \in N_i} Z_j \quad (1)$$

To clearly distinguish between word-level and character-level embedding, we enhance the word-level embedding of a node by adding the neighboring node's embedding information. Thus, we use the (1) to regularize the embedding of each user by a real number parameter $\lambda \in [0,1]$ along with the contribution of their neighbor embeddings. Where, Z_i is the word-level embedding vector of i^{th} node of Word2vec model, N_i are the neighbor nodes of i^{th} node.

2.1.3. Topic-level attribute embedding

In this topic-level embedding, we use latent Dirichlet allocation (LDA) [33] which is a popular topic modeling technique to extract topics from a given corpus. This embedding technique aims to learn the representation of node using the text similarity of attribute texts of user in terms of long sentences or paragraphs such as description of books, projects, and published articles. All of this information is merged in order to create the corpus data as input for this embedding technique. We treat user's attributes at topic-level as a document which may contain many words. Firstly, we clean, preprocess and tokenize the text corpus data to words. Then, we build a document-word matrix $\in R^{|\mathcal{D}| \times |\mathcal{W}|}$ where $|\mathcal{D}|$ is the number of documents/users, $|\mathcal{W}|$ is the number of distinct words in the word-level dictionary. LDA converts this document-word matrix into two other matrices: document-topic matrix and topic-word matrix.

The goal of this process is to find the most optimal representation of the document-topic matrix $\in R^{|\mathcal{D}| \times |\mathcal{T}|}$ and the topic-word matrix $\in R^{|\mathcal{T}| \times |\mathcal{W}|}$ through an iterative process, where $|\mathcal{T}|$ is the number of topics. At the first iteration, it randomly assigns a list of topics to each word in a document to generate the initial document-topic and topic-word matrices. Then, LDA will iterate over each document D_i and each word W_j in document in order to update the correct topic for a specific word W_j with an assumption that all the topics that have been assigned are correct except the current word.

2.2. Graph attention network

We apply graph attention mechanism [21], an efficient graph neural network to find the low dimensional embedding which maximizes the preservation of network attributes and local structure. The

GAT model has a mechanism to assign the different attention coefficients to all neighborhoods of a node at the aggregation step to increase the performance of prediction tasks. We also employ a multi-head attention mechanism to prevent noisyness and make the prediction model more stable.

$$l_{ij} = \vec{\sigma}^T [W \vec{x}_i || W \vec{x}_j] \quad (2)$$

First, we use the embedding vector generated from the multilevel attribute embedding technique (as proposed in subsection 2.1.) as the initial feature vector for the GAT model. We denote \vec{x}_i as the feature vector of a user node v_i in network G , and \vec{x}_j is the initial feature vector of the user node $v_j \in N_i$ are its neighbors. We use (2) to compute the important score l_{ij} between user node v_i and all its neighbors $v_j, 1 \leq j \leq |N_i|$. Where, $W \in R^{D \times D}$ is the weight matrix and $\vec{\sigma} \in R^{2D}$ is the weight vector that is the model parameters, D is the original dimension of the initial feature vector, D' is the dimension of the hidden layer.

$$\alpha_{i,j} = \frac{\exp(\text{LeakyReLU}(l_{i,j}))}{\sum_{t=1}^{|N_i|} \exp(\text{LeakyReLU}(l_{i,t}))} \quad (3)$$

These important scores are then typically normalized using the softmax function, in order to be comparable across different neighborhoods. We use the (3) to compute the normalized attention coefficient $\alpha_{i,j}$ between node v_i and all its neighbors v_i . Where, *LeakyReLU* is a type of activation function based on a ReLU.

$$\vec{u}_i = \sigma \left(\frac{1}{K} \sum_{k=1}^K \sum_{j=1}^{|N_i|} \alpha_{i,j}^k W^k \vec{x}_j \right) \quad (4)$$

Finally, we compute the embedding vector of user node v_i in graph G using (4) which implements a multi-head attention to stabilize the self-attention learning process. In which, K is the number of attention mechanisms and k is the k^{th} attention mechanism.

2.3. Canonical correlation analysis

After representing the attribute level and the structure level of each network in the above steps, we will have the embedding matrices $X \in R^{d \times n}$ and $Y \in R^{d \times m}$ representing the information of the source network and the target network. Where m and n are the number of users in the source and target networks, d is the final dimensional concatenate from the four embedding techniques mentioned (5).

$$p = \max \text{corr}(h_i^T X, m_j^T Y) = \max \frac{h_i^T C_{XY} m_j}{\sqrt{(h_i^T C_{XX} h_i)(m_j^T C_{YY} m_j)}} \quad (5)$$

We use canonical correlation analysis (CCA) technique to represent these two distinct spaces X, Y on the same common semantic space. CCA is a technique for learning the linear correlational relations among multiple multidimensional datasets. CCA finds a canonical latent space that maximizes association between projections of these datasets onto that common space. RCCA technique mostly define the canonical matrices as $A = [a_1; a_2; \dots; a_k] \in R^{d \times k}$ and $B = [b_1; b_2; \dots; b_k] \in R^{d \times k}$, it includes k pairs of linear projections. We use (5) to find the canonical matrices A and B which maximize the correlation source and target network. The canonical matrices of the anchor link problem are resolved by projecting the embedding of the associated social networks G^X/G^Y into the canonical matrices A and B to get the common correlated space $Z^X = A^T X \in R^{k \times n}$ and $Z^Y = B^T Y \in R^{k \times m}$

3. EXPERIMENTAL

3.1. Datasets

We experiment our proposed model in two real-life alignment datasets [30]. Table 1 illustrates the statistics of these data sets.

- Weibo vs Douban. This dataset was gathered from two well-known Chinese social networks, Weibo and Douban. It contains 1,397 anchor links.
- Data base and logic programming17 (DBLP17) vs DBLP19. This dataset was collected from computer science bibliography website. In this research, two snapshots of the DBLP network were collected in two different periods of time and we treat them as one pair of alignment networks. The anchor links were constructed from the authors who have the same unique key in two different snapshots of network, and it contains 2,832 anchor links.

Table 1. Statistics of real-life data sets

	#Nodes	#Edges	#Anchors
Weibo	9,714	117,218	1,397
Douban	9,526	120,245	
DBLP17	9,086	51,700	2,832
DBLP19	9,325	47,775	

3.2. Evaluation metrics

In our study, we used the Hit-precision metric [1] to assess the performance of the proposed method. The Hit-precision can be computed using (7). This metric measures the number of true anchor link appears in the top-k candidates.

$$h(x) = \frac{k - (\text{hit}(x) - 1)}{k} \quad (6)$$

In (6), $\text{hit}(x)$ is the position of a truly predicted user in the top-k candidates of the output collection. Suppose that n is the number of assessed user pairs, we can compute the hit-precision using the average score of the truly predicted user pairs.

$$\text{Hit-precision} = \sum_{i=1}^n h(x_i) \quad (7)$$

3.3. Baselines

We compare our proposed method with the following state-of-the-art anchor link prediction methods:

- MAUIL contains three modules: attribute-based embedding module, network-based embedding module, and RCCA-based module.
- Our method is an improvement of MAUIL by adding the graph attention mechanism. This method includes four modules: attribute-based embedding, network structure -based embedding module, graph attention embedding module, and RCCA-based module.

3.4. Performance comparison

We compare the performance of our proposed method with the baselines. In the experiment, all the hyperparameters of both compared methods and our method are tuned to perform the best on the test dataset. For our method, the output dimensional for each embedding technique is set to the same value $D = 100$, the number of canonical components is empirically set to $k = 80$ for both the Weibo-Douban and DBLP datasets. Correspondingly, the regulation parameters $R = 1000$ are considered for the Weibo-Douban and DBLP dataset, respectively.

Table 2 is convincing results on the prediction of the anchor link for the DBLP17-DBLP19 and Weibo-Douban dataset. From this table, we can discover that our model consistently outperforms all baselines in two pairs of datasets. We also tested our model on many different training ratios and evaluate the performance on difference hit-precision. Figure 2(a) shown the performance on hit-precision@5, Figure 2(b) shown the performance on hit-precision@10, Figure 2(c) shown the performance on hit-precision@20, Figure 2(d) shown the performance on hit-precision@30 for DBLP dataset. Similarly, Figure 3(a) shown the performance on hit-precision@5, Figure 3(b) shown the performance on hit-precision@10, Figure 3(c) shown the performance on hit-precision@20, Figure 3(d) shown the performance on hit-precision@30 for DBLP dataset. The experimental results show that, our model gives consistently better results for all training ratios. The accuracy of the model is proportional to the training data, when the training data reaches about 50%-60%, the accuracy almost converges.

Table 2. Comparison of Hit-precision with training ratio 30%

Metric	Weibo vs Douban		DBLP-17 vs DBLP-19	
	Our method	MAUIL	Our method	MAUIL
Hit-precision@1	0.2380	0.2310	0.7720	0.7510
Hit-precision@3	0.2913	0.2797	0.8060	0.7810
Hit-precision@5	0.3236	0.3099	0.8224	0.7970
Hit-precision@10	0.3774	0.3597	0.8412	0.8213
Hit-precision@15	0.4104	0.3920	0.8543	0.8324
Hit-precision@20	0.4326	0.4163	0.8636	0.8379
Hit-precision@25	0.4488	0.4361	0.8695	0.8413
Hit-precision@30	0.4615	0.4527	0.8747	0.8435

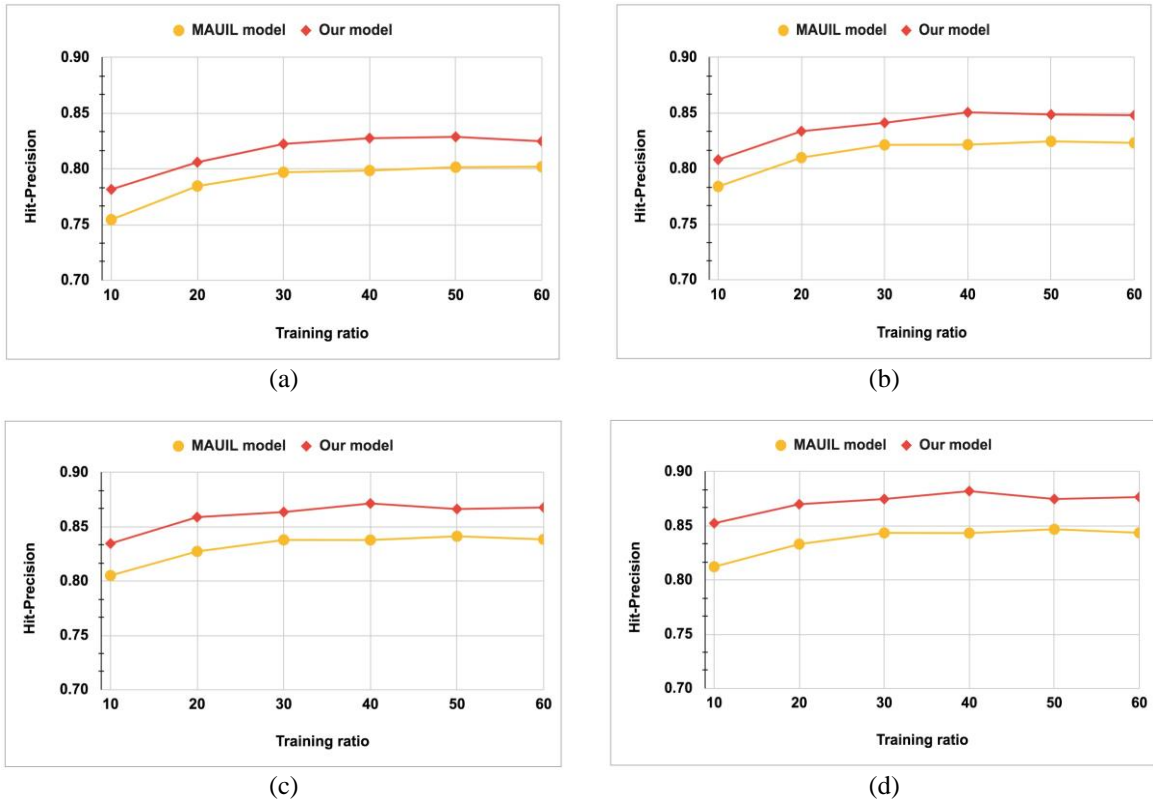


Figure 2. Precision@K comparison on DBLP dataset (a) precision@5, (b) precision@10, (c) precision@20, and (d) precision@30

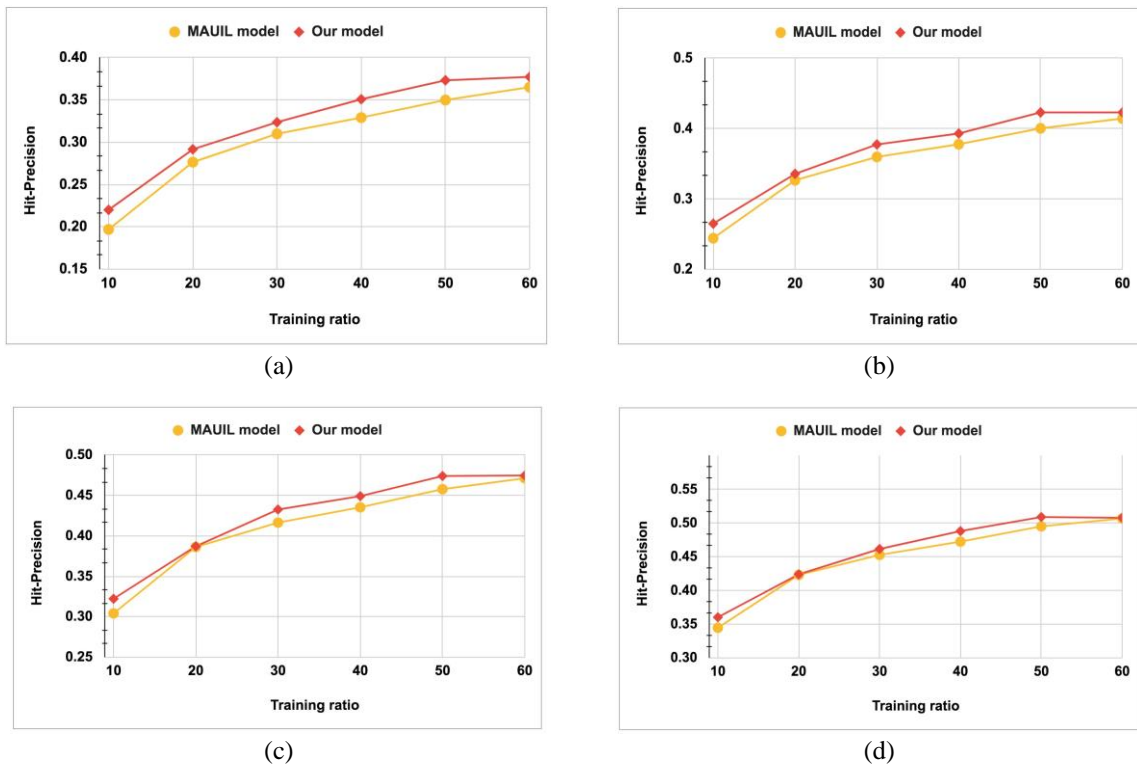


Figure 3. Precision@K comparison on Weibo-Douban dataset (a) precision@5, (b) precision@10, (c) precision@20, and (d) precision@30

4. CONCLUSION

In this article, we study and apply multilevel embedding techniques to learn the representation of nodes in online social networks. We project learned embedding onto same latent space using canonical correlation analysis and apply the models in representation learning along with other techniques to predict the formulation of the anchor link across information networks, specific tasks in information network analysis. The experiments on the real-life dataset indicate that our method can substantially enhance the precision compare to the traditional methods. The following is a summary of our contributions in this article: i) we have learned theoretical knowledge related to network representation learning, graph attention network, and anchor link prediction; ii) we combined the multilevel embedding techniques for text-based attributes, graph attention mechanism, and canonical correlation analysis into the anchor link prediction; and iii) we have experimented with two real-life data sets. We also have evaluated and compared experimental results with related algorithms and our model consistently outperforms all baselines.




REFERENCES

- [1] X. Mu, F. Zhu, E.-P. Lim, J. Xiao, J. Wang, and Z.-H. Zhou, "User identity linkage by latent user space modelling," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2016, pp. 1775–1784, doi: 10.1145/2939672.2939849.
- [2] A. T. Hadgu and J. K. R. Gundam, "Learn2link: Linking the social and academic profiles of researchers," in *Proceedings of the International AAAI Conference on Web and social media*, vol. 14, pp. 240–249, May 2020.
- [3] C. Riederer, Y. Kim, A. Chaintreau, N. Korula, and S. Lattanzi, "Linking users across domains with location data: theory and validation," in *Proceedings of the 25th international conference on world wide web*, 2016, pp. 707–719, doi: 10.1145/2872427.2883002.
- [4] J. Feng *et al.*, "Dplink: user identity linkage via deep neural network from heterogeneous mobility data," in *The World Wide Web Conference*, May 2019, pp. 459–469, doi: 10.1145/3308558.3313424.
- [5] X. Kong, J. Zhang, and P. S. Yu, "Inferring anchor links across multiple heterogeneous social networks," in *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, Oct. 2013, pp. 179–188, doi: 10.1145/2505515.2505531.
- [6] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, Aug. 2016, pp. 855–864, doi: 10.1145/2939672.2939754.
- [7] L. F. Ribeiro, P. H. Saverese, and D. R. Figueiredo, "struc2vec: Learning node representations from structural identity," in *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, 2017, pp. 385–394, doi: 10.1145/3097983.3098061.
- [8] M. Grohe, "word2vec, node2vec, graph2vec, x2vec: Towards a theory of vector embeddings of structured data," in *Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, Jun. 2020, pp. 1–16, doi: 10.1145/3375395.3387641.
- [9] L. Liu, W. K. Cheung, X. Li, and L. Liao, "Aligning users across social networks using network embedding," *Ijcai*, vol. 16, pp. 1774–80, Jul. 2016.
- [10] X. Li, Y. Shang, Y. Cao, Y. Li, J. Tan, and Y. Liu, "Type-aware anchor link prediction across heterogeneous networks based on graph attention network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, pp. 147–155, Apr. 2020, doi: 10.1609/aaai.v34i01.5345.
- [11] H. T. Trung, T. V. Vinh, N. T. Tam, H. Yin, M. Weidlich, and N. Q. V. Hung, "Adaptive network alignment with unsupervised and multi-order convolutional networks," *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, 2020, pp. 85–96, doi: 10.1109/ICDE48307.2020.00015.
- [12] V. V. Le, T. T. Tran, P. N. Pham, and V. Snasel, "Anchor link prediction in online social network using graph embedding and binary classification," *International Conference on Computational Collective Intelligence*, pp. 229–240, 2020, doi: 10.1007/978-3-030-63007-2_18.
- [13] L. Lan, H. Peng, C. Tong, X. Bai, and Q. Dai, "Cross-network community sensing for anchor link prediction," *2021 International Joint Conference on Neural Networks (IJCNN)*, 2021, pp. 1–8, doi: 10.1109/IJCNN52387.2021.9534462.
- [14] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," in *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, Jan. 2009, doi: 10.1109/TNN.2008.2005605.
- [15] H. Dai, Z. Kozareva, B. Dai, A. Smola, and L. Song, "Learning steady-states of iterative algorithms over graphs," in *International conference on machine learning PMLR*, Jul. 2018, pp. 1106–1114.
- [16] R. Levie, F. Monti, X. Bresson, and M. M. Bronstein, "CayleyNets: graph convolutional neural networks with complex rational spectral filters," in *IEEE Transactions on Signal Processing*, vol. 67, no. 1, pp. 97–109, Jan. 2019, doi: 10.1109/TSP.2018.2879624.
- [17] W.-L. Chiang, X. Liu, S. Si, Y. Li, S. Bengio, and C.-J. Hsieh, "Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, Jul. 2019, pp. 257–266, doi: 10.1145/3292500.3330925.
- [18] S. Cao, W. Lu, and Q. Xu, "Deep neural networks for learning graph representations," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, Feb. 2016, doi: 10.1609/aaai.v30i1.10179.
- [19] J. You, R. Ying, X. Ren, W. Hamilton, and J. Leskovec, "Graphrnn: Generating realistic graphs with deep auto-regressive models," in *International conference on machine learning. PMLR*, Jul. 2018, pp. 5708–5717.
- [20] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 922–929, doi: 10.1609/aaai.v33i01.3301922.
- [21] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, vol. 1050, p. 20, 2017, doi: 10.48550/arXiv.1710.10903.
- [22] V. Carchiolo, C. Cavallo, M. Grassia, M. Malgeri, and G. Mangioni, "Link prediction in time varying social networks," *Information*, vol. 13, no. 3, p. 123, Mar. 2022, doi: 10.3390/info13030123.
- [23] R. Giubilei and P. Brutti, "Supervised classification for link prediction in facebook ego networks with anonymized profile information," *Journal of Classification*, pp. 1–24, 2022, doi: 10.1007/s00357-021-09408-2.
- [24] S. Li *et al.*, "Heterogeneous attention concentration link prediction algorithm for attracting customer flow in online brand community," in *IEEE Access*, vol. 10, pp. 20898–20912, 2022, doi: 10.1109/ACCESS.2022.3151112.
- [25] B. Li, D. Pi, and Y. Lin, "Learning ladder neural networks for semi-supervised node classification in social network," *Expert Systems with Applications*, vol. 165, p. 113957, Mar. 2021, doi: 10.1016/j.eswa.2020.113957.




- [26] A. Q. Ohi, M. F. Mridha, F. B. Safir, M. A. Hamid, and M. M. Monowar, "Autoembedder: a semisupervised dnn embedding system for clustering," *Knowledge-Based Systems*, vol. 204, p. 106190, Sep. 2020, doi: 10.1016/j.knsys.2020.106190.
- [27] C. Shi, B. Hu, W. X. Zhao, and P. S. Yu, "Heterogeneous information network embedding for recommendation," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 2, pp. 357–370, 1 Feb. 2019, doi: 10.1109/TKDE.2018.2833443.
- [28] Y. He, Y. Zhang, L. Qi, D. Yan, and Q. He, "Outer product enhanced heterogeneous information network embedding for recommendation," *Expert Systems with Applications*, vol. 169, p. 114359, May 2021, doi: 10.1016/j.eswa.2020.114359.
- [29] P. N. Pham, B.-N. T. Nguyen, Q. T. Co, and V. Sna'sel, "Multiple benefit thresholds problem in online social networks: An algorithmic approach," *Mathematics*, vol. 10, no. 6, p. 876, Mar. 2022, doi: 10.3390/math10060876.
- [30] B. Chen and X. Chen, "Mauil: Multilevel attribute embedding for semisupervised user identity linkage," *Information Sciences*, vol. 593, pp. 527–545, May 2022, doi: 10.1016/j.ins.2022.02.023.
- [31] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *Proceedings of the 24th international conference on world wide web*, May 2015, pp. 1067–1077, doi: 10.1145/2736277.2741093.
- [32] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, Jan. 2013, doi: 10.48550/arXiv.1301.3781.
- [33] D. Kim, D. Seo, S. Cho, and P. Kang, "Multi-co-training for document classification using various document representations: Tf-idf, lda, and doc2vec," *Information Sciences*, vol. 477, pp. 15–29, Mar. 2019, doi: 10.1016/j.ins.2018.10.006.

BIOGRAPHIES OF AUTHORS






Van-Vang Le    is Ph.D. student in Computer Science at Faculty of Electrical Engineering and Computer Science, VSB-Technical University of Ostrava, Ostrava-Poruba, Czech Republic. He received the M.S. degree in Information System at University of Science, Vietnam National University, Ho Chi Minh City in 2013. Currently, he is lecturer at Faculty of Information Technology, Ton Duc Thang University (TDTU), Vietnam. His research interests include information network analysis, network alignment, graph neural network, and network representation learning. He can be contacted at email: levanvang@tdtu.edu.vn.






Phuong Nguyen Huy Pham    received the M.S. degree in computer science from the Pierre and Marie Curie University, also known as Paris 6, France, in 2010. He is currently a Lecturer at the Ho Chi Minh City University of Food Industry (HUFI), Vietnam. He is currently working the Ph.D. degree in computer science at VSB-Technical University of Ostrava, Ostrava, Czech Republic. His research interests include complex networks, approximation algorithm, influence maximization, and combinatorial optimization in social networks. He can be contacted by email: phuongpnh@hufi.edu.vn.



Tran Kim Toai    is currently a Ph.D student in the Department of Computer Science, VSB-Technical of Ostrava, Czech Republic. He works as a researcher and lecturer in Ho Chi Minh University Technical Education. His research and development experience include over 10 years in industry and academia. He has worked in a multinational company and a university environment involving networking, cyber security, artificial intelligence, data mining, neural network, data analysis, finance, and applied to various real problems. He can be contacted by email: toaitk@hcmute.edu.vn.



Vaclav Snašel    is currently a Professor with the Department of Computer Science, VSB-Technical University of Ostrava, Czech Republic. He works as a Researcher and a University Teacher. He is also the Dean of the Faculty of Electrical Engineering and the Computer Science Department. He is also the Head of the Research Program IT4 Knowledge Management, European Center of Excellence IT4 Innovations. His research and development experience include over 30 years in industry and academia. He works in a multi-disciplinary environment involving artificial intelligence, social networks, conceptual lattice, information retrieval, semantic web, knowledge management, data compression, machine intelligence, neural networks, web intelligence, nature and bio-inspired computing, data mining, and applied to various real-world problems. He can be contacted by email: vaclav.snasel@vsb.cz.