NeXLink: Node Embedding Framework for Cross-Network Linkages Across Social Networks

Rishabh Kaushal[1,2], Shubham Singh[2] and Ponnurangam Kumaraguru[2]

[1] Department of Information Technology,
Indira Gandhi Delhi Technical University for Women, Delhi, India
[2] Precog Research Lab,
Indraprastha Institute of Information Technology, Delhi, India
rishabhk,shubham12101,pk@iiitd.ac.in

Abstract. Users create accounts on multiple social networks to get connected to their friends across these networks. We refer to these user accounts as user identities. Since users join multiple social networks, therefore, there will be cases where a pair of user identities across two different social networks belong to the same individual. We refer to such pairs as Cross-Network Linkages (CNLs). In this work, we model the social network as a graph to explore the question, *whether we can obtain effective social network graph representation such that node embeddings of users belonging to CNLs are closer in embedding space than other nodes, using only the network information.* To this end, we propose a modular and flexible node embedding framework, referred to as *NeXLink*, which comprises of three steps. First, we obtain local node embeddings by preserving the local structure of nodes within the same social network. Second, we learn the global node embeddings by preserving the global structure, which is present in the form of common friendship exhibited by nodes involved in CNLs across social networks. Third, we combine the local and global node embeddings, which preserve local and global structures to facilitate the detection of CNLs across social networks. We evaluate our proposed framework on an augmented (synthetically generated) dataset of 63,713 nodes & 817,090 edges and real-world dataset of 3,338 Twitter-Foursquare node pairs. Our approach achieves an average Hit@1 rate of 98% for detecting CNLs across social networks and significantly outperforms previous state-of-the-art methods.

Keywords: Social Networks · Network Embedding · User Identity Linkage

1 Introduction

Online Social Networks (OSNs) are popular platforms on the Internet, helping users to connect with their friends, enabling them to view and share information. OSNs offer different types of content to its users. For instance, YouTube offers videos, Instagram offers images, while Facebook and Twitter offers a mix of text, images, and videos. OSNs also offer different types of friend network to its users.

LinkedIn provides access to the professional network while others like Facebook, help in connecting to personal friends. With the presence of these multiple social networks, it is evident that users join more than one social network to avail these several benefits offered by OSNs. In this scenario, it is of great interest to find user identities across multiple social networks belonging to the same individual, which we refer to as *cross-network linkage* and refer these identities as *linked identities*. User behaviors exhibited through these linked identities across multiple OSNs help in building a collective digital footprint [12]. Users' popularity and friendships trends [23] and influence [3] across OSNs can be better understood using such digit footprints. For an adversary to launch social engineering attacks [1], this helps in harvesting information about users based on their activities in multiple OSNs. In digital marketing, it helps to know and identify your customer [11] for the targeted advertisement.

Given the immense importance of finding linked identities, we propose a solution based on the construction of effective graph representations. The goal is to learn node embeddings in a social graph such that nodes with similar characteristics are represented by similar node embedding vectors. In the context of our problem, we ask the question *whether we can obtain effective social network graph representation such that node embeddings of users belonging to CNLs are closer in embedding space than other nodes*. In other words, as depicted in Figure 1, the goal is to propose an embedding framework that transforms nodes into embedding vectors such that nodes present in linked identities are closer in embedding space than other nodes. To this end, we propose a three-step *NeXLink* framework that learns node representations to detect CNLs across social networks. In the first step, the local structure of nodes within the same network is preserved. In social networks, these local structures would comprise of friendship relation or follow-followee relation maintained by user identities. In particular,

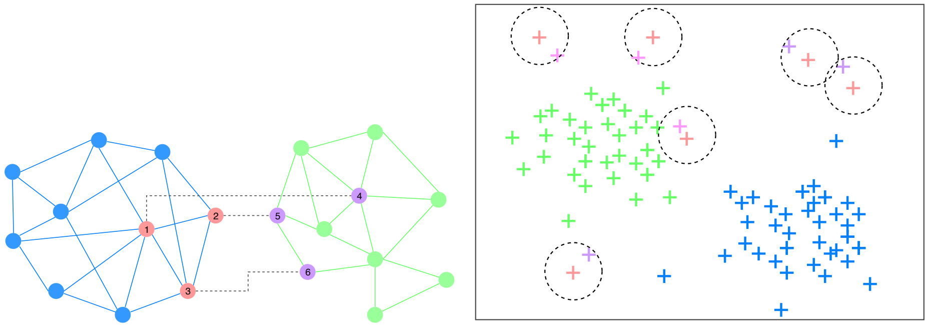


Fig. 1: Our proposed NeXLink framework learns node embeddings from two social networks (represented as graphs, on the left side) with few cross-network linkages. On the right side, we depict embedding space in which nodes corresponding to user identities belonging to same individual are closer than other nodes.

we learn node embeddings of nodes within the same network using the normalized edge weights so that nodes that are structurally near to each other, their corresponding embeddings are also close in embedding space. In the second step, the global structure of nodes connected across multiple networks is preserved. In social networks, these global structures would comprise of cross-platform linkages which represent user identities across social networks belonging to the same individual. These linked identities are expected to exhibit a number of common friends across social networks. In particular, we learn the node embeddings of nodes that are part of Cross-Network Linkages (CNLs) by biasing the random walk in proportion to the common friendship. As a result, node embeddings of nodes that are part of CNLs with more common friends are closer in embedding space. In the third step, we directly leverage the node embeddings to evaluate their efficacy in the detection of cross-network linkages across social networks. The code and data of our work are available at the GitHub repository.[3] We evaluate our proposed approach of the NeXLink framework on two datasets. The first dataset is an augmented dataset synthetically generated using the Facebook social network [18] comprising of 63,713 nodes (users) and 817,090 edges. Our approach works well in all possible augmentations of the Facebook dataset achieving an average Hit@1 rate of 98%, which means that the probability of hitting on the correct cross-network linkage across social networks is 98%. Further, our approach outperforms the state-of-the-art prior approaches of node representations namely LINE [17] and DeepWalk [15] on synthetically generated graphs, which we refer to as augmented dataset. The second dataset comprises of a real-world dataset of Twitter-Foursquare social networks [25] comprising of 3,338 nodes (user) pairs. We find that except for Hit@1 rate, our approach better than the state-of-the-art prior approaches of user identity linkages namely IONE [10] and REGAL [6] in Hit@5 rates and above. The key contributions of our work are as below.

- We propose a modular and flexible NeXLink framework as a two-step optimization process that preserves local structure within the same network and preserves global structure manifested in the form of cross-network linkages.
- We extensively evaluate our framework on two datasets, one augmented dataset obtained from Facebook and other real-world dataset comprising of Twitter-Foursquare node pairs. Our framework works well on the synthetically generated dataset and outperforms prior node representation approaches (LINE and DeepWalk) and identity linkage approaches (IONE and REGAL).

2 Related Work

Recently, there are a few prior works that have addressed the problem of user identity linkage using the network embedding approach whose aim is to learn a

[3] Code and dataset of our work can be found at: https://github.com/precog-iiitd/nexlink-netscix-2020

low dimensional representation for a given node in a graph. We categorize these prior methods in the field of network embedding into two main categories, as explained below.

Problem independent approaches: These works only aim to learn generic low-dimensional representations without focusing on user linkage problems. The objective is to learn effective node representations in low dimensions. Tang et al. [17] propose a framework for network embedding in large graphs to preserve node structures of nodes which are directly connected (first-order node proximity) and connected at a distance of two hops (second-order node proximity). Perozzi et al. [15] leverage the notion of the skip-gram model in language modeling to perform truncated random walks in order to learn latent representations of nodes in a graph. Wang et al. [19] preserve the first and second-order node proximity using a semi-supervised deep learning model. Grover et al. [2] extend the notion of a random walk by introducing biased walks in node neighborhood to learn feature representations of the node in a network. Xu et al. [22] propose two embeddings for each node that capture the structural proximity of nodes as well as the semantic similarity, which they express in terms of common interests. Liang et al. [8] propose a dynamic user and word embedding model (DUWE) that monitors over some time, the relationship between user and words to model their embeddings. Liu et al. [9] present a self-translation network embedding (STNE) framework that is a sequence-to-sequence framework taking into consideration both content and network features of the node.

Problem dependent approaches: These learn low-dimensional embedding focusing on specific problem, which in our case is to detect cross-network linkages representing user identities across social networks. Liu et al. [10] propose an input-output node embedding (IONE) framework to align user identities across social networks belonging to the same person by learning node representations that preserve follower-followee relationships. Man et al. [13] introduce a framework referred to as PALE, which predicts anchor links via embeddings. First, it converts a social network into a low dimensional node representation. They follow it up by learning a matching function that is supervised by known anchor links. Heimann et al. [6] explain the REGAL framework, which stands for representation learning-based graph alignment based on the cross-network matrix factorization method. Wang et al. [20] propose LHNE mode referred to as linked heterogeneous network embedding which creates an unified framework to leverage structure and content posted by users for learning node representations. Xie et al. [21] use the concept of factoid embedding, which is an unsupervised approach to perform user identity linkage. Our proposed approach outperforms some of these existing approaches, as explained later in this paper.

3 Proposed Approach

In this section, we discuss our proposed NeXLink framework for effective representation and detection of cross-network linkages across social networks. We consider two social networks X and Y as two undirected graphs $G_X(V_X, E_X)$

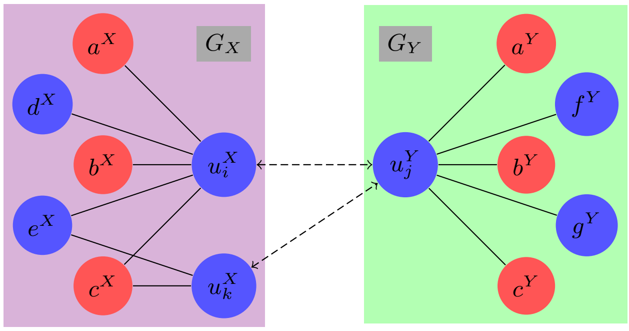


Fig. 2: Illustration of common neighbors of user identities u_i^X and u_j^Y belonging to networks G_X and G_Y. Since all neighbors are common, it is highly likely that u_i^X and u_j^Y belong to same individual than u_k^X and u_j^Y.

and $G_Y(V_Y, E_Y)$, where V_X & V_Y represent the nodes (users) of graphs and E_X & E_Y represent the edges. An edge between nodes u_i and u_j indicates friendship relation between users u_i and u_j. We divide the set of node pairs (u_i^X, u_j^Y) across social networks X and Y into two types, namely, cross-network linkages, denoted by $CNL(V_X, V_Y)$ and other pairs are denoted by $NCNL(V_X, V_Y)$. Nodes u_i^X and u_j^Y belonging to social networks X and Y are referred to as cross-network linkage if u_i^X and u_j^Y belong to the same individual and the pair $(u_i^X, u_j^Y) \in CNL(V_X, V_Y)$ else $(u_i^X, u_j^Y) \in NCNL(V_X, V_Y)$. Further, it may be observed in Figure 2, that the two users represented as two nodes u_i^X and u_j^Y have a^X, b^X and c^X as friends in social network X and same friends a^Y, b^Y and c^Y in social network Y. We refer to such familiar friends as common friendship and leverage this behavior in learning node representations in our NeXLink framework. Besides familiar friends, each node also has some friends which are specific to one social network only. In Figure 2, nodes d^X and e^X are friends of u_i^X in only social network X whereas nodes f^Y and g^Y are friends of u_j^Y in only social network Y. We note that above formulations for undirected graphs are also applicable in case of directed graphs, in which case the friendship relation would get replaced by follow-followee relation using directed edges.

3.1 Problem Statement

Given two graphs $G_X(V_X, E_X)$ and $G_Y(V_Y, E_Y)$ as input, we define cross-network linkage $CNL(G_X, G_Y)$ as the set of user identity pairs across these two networks X and Y, which belong to the same person. Similarly, we denote all other user pairs which do not belong to the same person by $NCNL(G_X, G_Y$. The goal of network embedding function (denoted by f_{emb}) is to transform each user identity $u_i^X \in V_X$ and $u_j^Y \in V_Y$ into low *d-dimensional* vectors z_i^X and z_j^Y such that if user identities u_i^X and u_j^Y belong to the same individual (i.e. they represent cross-network linkage), then their corresponding node embeddings z_i^X and z_j^Y are closer in embedding space else they are far apart.

$$\begin{aligned} z_i^X &= f_{emb}(u_i^X), \forall u_i^X \in V_X. \\ z_j^Y &= f_{emb}(u_j^Y), \forall u_j^Y \in V_Y. \\ such\ that& \\ sim(z_i^X, z_j^Y) &>> sim(z_k^X, z_j^Y)\ and \\ \exists\ (u_i^X, u_j^Y) \in CNL(V_X, V_Y) &\wedge (u_k^X, u_j^Y) \in NCNL(V_X, V_Y). \end{aligned} \tag{1}$$

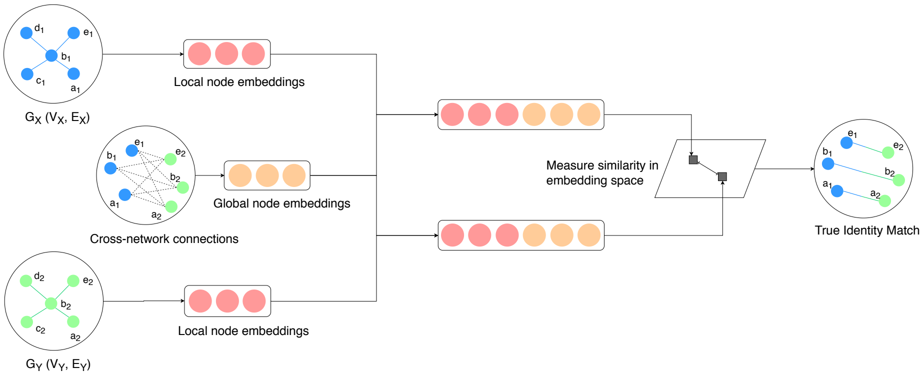


Fig. 3: NeXLink Framework. Architecture diagram of our proposed framework that learns node embeddings from two social networks (represented as graphs) to represent the cross-network linkages across social networks. Local node embeddings are concatenated with global node embeddings to generate final node embeddings.

3.2 NeXLink Framework

The goal of our proposed NeXLink node embedding framework is to obtain representations of nodes in two networks X and Y such that node pairs participating in cross-network linkages have similar node embeddings than other node pairs. To achieve this goal, we follow a two-step approach, as depicted in Figure 3. In the first step, structural similarities of nodes within or local in their respective networks are preserved independently of the other network. In the second step, similarities of nodes across (or global) the two networks are preserved using a common friendship relationship. Given the two-step process, the embedding function f_{emb} can be broken down into two embedding functions, as shown below.

$$\begin{aligned} z_i^X &= f_{global}(u_i^X) \oplus f_{local}(u_i^X), \forall u_i^X \in V_X. \\ z_j^Y &= f_{global}(u_j^Y) \oplus f_{local}(u_j^Y), \forall u_j^Y \in V_Y. \end{aligned} \tag{2}$$

There can be different ways of combining the global and local node embeddings, however, it turned out that concatenation is the best operation $\oplus$ to combine local and global node embeddings, which we finally used in our proposed NeXLink framework. Further, we note that our proposed approach makes use of only the network structure in the two social networks. However, it can be easily extended to include other sources of information from content and profile information of users, which we leave for future work.

Step 1 - Preserving Local Structure Within Social Networks We perform the first step on the intuition that directly connected user nodes within their respective social networks are likely to exhibit similar characteristics, based on the well established social behavioral principle of homophily [14]. Given two nodes u_i^X and u_k^X in same social network X, the goal is to define an encoding function f_{local} that takes these nodes as input and learns their *d-dimensional* embedding vectors $z_i^X \in R^d$ and $z_k^X \in R^d$. To learn z_i^X and z_k^X for all nodes in V_X, we rely upon the probabilistic approach. The empirical probability of the relationship between two nodes u_i^X and u_k^X within the same social network X can be defined as the normalized weight of edge $(w_{i,k}^X)$ between the nodes. Since we consider only the structural information of the network, therefore, for this work, we consider weights to have binary values 1 or 0, depending upon whether there is an edge or not, respectively. In general, the weight of the edge between nodes is intuitively proportional to the similarity between two nodes. Similarity, we can measure other criteria like content similarity. However, we consider only the network structure similarity in this work. We employ a well-established network embedding algorithm, LINE [17], to preserve the local structure.

Step 2 - Preserving Global Structure Across Social Networks We propose the second step based on the intuition that user nodes with common friends (CF) across the social networks are likely to belong to the same person. The degree to which two nodes (users) u_i^X and u_j^Y on two social networks X and Y, respectively, having *common friendship*, is expressed as below.

$$CF(u_i^X, u_j^Y) = \frac{N(u_i^X) \cap N(u_j^Y)}{N(u_i^X) \cup N(u_j^Y)} \tag{3}$$

where $N(u_i^X)$ and $N(u_j^Y)$ represent the set of friends of i^{th} user in network X and j^{th} user in network Y, respectively. Higher is the value of common friendship (CF), more likely the users u_i^X and u_j^Y would belong to the same person. Therefore, the goal of second encoding function f_{global} is to take u_i^X and u_j^Y as inputs and generate *d-dimensional* node embeddings vectors $z_{G,i}^X \in R^d$ and $z_{G,j}^Y \in R^d$, respectively by using supervisory information of common friendship between u_i^X and u_j^Y in networks X and Y, respectively, along with structural information. If u_i^X and u_j^Y have more common friends, their embedding vectors $z_{G,i}^X$ and $z_{G,j}^Y$ are expected to be closer in embedding space. We employ a well-established network embedding DeepWalk [15] algorithm to preserve the local structure.

4 Data

We evaluate our approach on two network datasets, one augmented, and another a real-world dataset.

4.1 Augmented Dataset

We use the Facebook friendship network dataset[4], provided by Viswanath et al. [18], comprising 63,713 users and 817,090 edges. We create an undirected graph from the dataset and filter out the nodes that have a degree less than 5, reducing the graph to 40,711 nodes and 766,579 edges. We use this graph to create two subgraphs using a sampling algorithm proposed by Man et al. [13]. Given a graph $G(V, E)$, the algorithm takes two parameters, α_s, α_c and produces two subgraphs $G_X(V_X, E_X), G_Y(V_Y, E_Y)$. The parameter α_s represents how likely are the two subgraphs to retain the edges from the original graph, or the sparsity level. Similarly, the parameter α_c indicates the expected fraction of edges shared among the two subgraphs, or the overlap level. Table 1 shows the number of edges and nodes in the generated subgraphs for different values of α_s and α_c. Once we have the subgraphs, we need to generate node pairs which represent CNLs and NCNLSs across the two subgraphs, which we call as X-node pairs. To do so, we consider all the common nodes in both the graphs, $V_{CNL} = V_X \cap V_Y$, and call them as our CNL nodes, while we term others as NCNL nodes. Now, we take a CNL node and initiate a random walk of a variable length t in G_X, and later in G_Y. The random walks generate $2 \times t$ nodes from G_X and G_Y collectively, and these nodes are then paired with the CNL node to form node pairs.

4.2 Real-World Dataset

Kong et al. [7] introduced a network dataset collected from Twitter and Foursquare social networks. The data collection process is described in [7, 24] and used in multiple social link prediction problems [10, 25, 26]. Since the dataset comprises two graphs on its own, we do not need to employ any sampling algorithm to generate subgraphs, and we present the statistical details about the dataset in Table 1. The cross-network linkages represent the users that have profiles on both the social networks. It is evident that such users are less in number in this real-world dataset, compared to the number of CNLs in our augmented dataset.

5 Experiments

We design our experiments to answer the following research questions:

RQ1 How do the values α_s and α_c affect the retrieval of a cross-network node match?

[4] http://socialnetworks.mpi-sws.org/data-wosn2009.html

Graph	**#Nodes**	**#Edges**	**#CNLs**
Augmented Dataset			
$G_X(\alpha_s = 0.5, \alpha_c = 0.5)$	40,558	383,463	39,061
$G_Y(\alpha_s = 0.5, \alpha_c = 0.5)$	40,563	382,380	
$G_X(\alpha_s = 0.5, \alpha_c = 0.9)$	40,562	383,360	40,458
$G_Y(\alpha_s = 0.5, \alpha_c = 0.9)$	40,547	383,528	
$G_X(\alpha_s = 0.9, \alpha_c = 0.5)$	40,602	422,295	40,418
$G_Y(\alpha_s = 0.9, \alpha_c = 0.5)$	40,708	689,481	
$G_X(\alpha_s = 0.9, \alpha_c = 0.9)$	40,709	689,856	40,705
$G_Y(\alpha_s = 0.9, \alpha_c = 0.9)$	40,709	690,103	
Real-World Dataset			
Twitter	5,120	130,575	1,288
Foursquare	5,313	54,233	

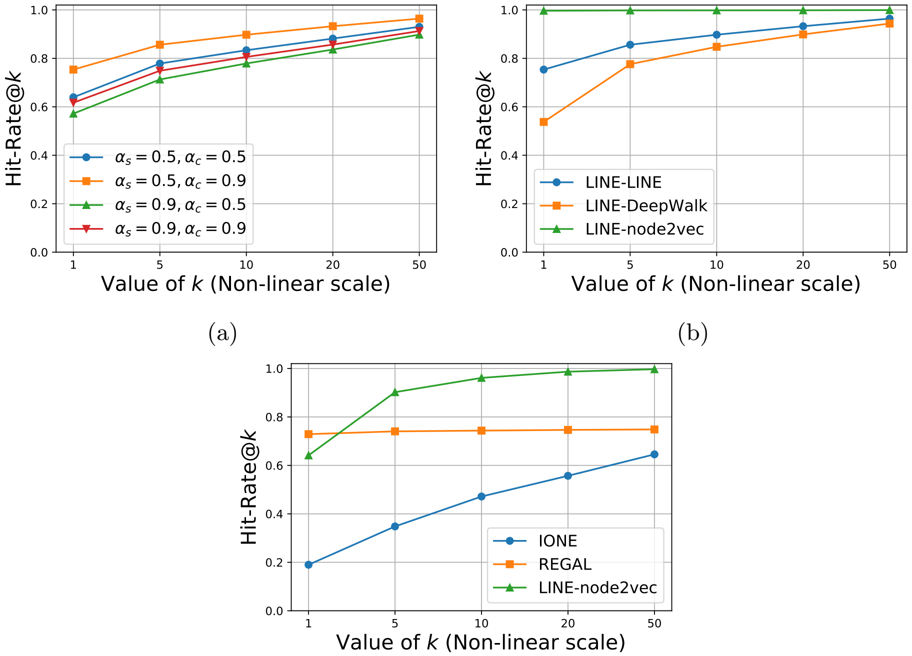
Table 1: Statistics for the two datasets used for the evaluation.

RQ2 How does the choice of second node embedding function f_{global} affect the cross-network node retrieval?

RQ3 How does our proposed NeXLink framework compare with other baselines on a real-world dataset?

We implement all our experiments using NetworkX [4] for graph functions and use OpenNE[5] to run network embedding implementations. To generate the $NCNL$ node pairs, we keep the depth of random walk, $t = 20$ throughout the experiments. When generating the embeddings for cross-network linkages, all embeddings functions treat node pairs as the edges of the cross-network graph, with CF values as the weights for cross-network edges. Given that our proposed NeXLink framework has two steps for the preservation of structure at the local and global level, we employ prior state-of-the-art node embedding methods at these steps. We typically employ LINE [17] to preserve local structure and consider only first-order proximity calculated over first-order nodes and run over 50 epochs, with early stopping. We do not use second-order proximity since that is taken care of in the second step of our NeXLink framework. We employ various node embedding methods (LINE [17] and DeepWalk[15]) to preserve the global structure in the second step of our NeXLink framework. However, as we explain in this section, it turns out that node2vec [2] when employing common friendship across social networks gives the best results. In node2vec, we set the parameters as $p = 1$ and $q = 2$ which, as mentioned by the authors, are more suited towards preserving structural equivalence. All embedding functions yield 128D embeddings. We evaluate our approach to measure how effectively can node embeddings preserve the CNLs in lower dimensional space, and how closely do

[5] https://github.com/thunlp/OpenNE



(a)

(b)

(c)

Fig. 4: Results of the three experiments for our research questions (RQ1-RQ3). (a) Comparison of Hit-Rate@k values for different sparsity (α_s) and overlap (α_c) levels. (b) Comparison of Hit-Rate@k values for different cross-network node embeddings. (c) Comparison of Hit-Rate@k values for the baselines and NeXLink (LINE-node2vec) over the real-world dataset.

network embeddings for CNL lie in that space. In order to compute closeness, we measure the cosine similarity over the node embeddings. When querying for a node u_i^X from the CNL pair (u_i^X, u_j^Y), we count a hit if the matching node embedding z_j^Y for node u_j^Y is present in a set of k node embeddings, ordered on their similarity. To measure accuracy, we calculate a ratio of hits over number of queries and term it as *Hit-Rate@k*. *Hit-Rate@k* is defined as:

$$Hit(u_i^X) = \begin{cases} 1, & \text{if } z_j^Y \in \{z_1^Y, z_1^Y, ..., z_k^Y\} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

$$Hit - Rate@k = \frac{\sum_{i=0}^{N_{CNL}} Hit(u_i^X)}{N_{CNL}} \tag{5}$$

We choose $k = [1, 5, 10, 20, 50]$ for all the experiments to evaluate our approach under different budget values.

5.1 Effect of Sparsity and Overlap levels

The α_s and α_c values affect the Common Friendship (CF) values for the CNL nodes, and since the second embedding function is trained to preserve the CF property across networks, we see significant differences in the performances with respect to the difference in α_s and α_c values. We start by employing LINE [17] to learn the local as well as cross-network similarity structure over the four subgraph configurations, as mentioned in Section 4, and present our results in Figure 4a. We observe that the X-node pairs with $\alpha_s = 0.5$ and $\alpha_c = 0.9$ values achieve the highest Hit-Rate@k for all values of k, starting from 0.75 at $k = 1$, and up to 0.96 at $k = 50$. The X-node pairs with $\alpha_s = 0.9$ and $\alpha_c = 0.5$ values achieve the lowest Hit-Rate@k values with 0.57 at $k = 1$ and 0.89 at $k = 50$. We attribute this behavior to the fact that less number of edges and more the overlap between the two subgraphs help the embeddings to capture structural similarities with less noise.

5.2 Effect of Cross-Network Node Embedding

We study the role of different network embedding techniques in our proposed NetXLink framework help to preserve CNLs and their impact on the performance of the detection of CNLs across social networks. LINE [17] is suitable for a majority of the number of graphs which preserve local network structure through first-order proximity, which makes it an ideal choice of node embedding method for our within-network embeddings. Along with LINE, we use DeepWalk [15] over *X-node pairs* to get cross-network embeddings, as it uses the structural information about inter-connected nodes by performing truncated random walks to learn latent representations of nodes in a graph, which in our case would be CNLs across networks. Similarly, we employ node2vec [2] which proposes a flexible notion of node neighborhood by designing a biased random walk to learn feature representations of graph nodes. Figure 4b shows the results of our experiments with different node embeddings. The LINE-DeepWalk performs relatively low at $k = 1$, but reaches closer to the Hit-Rate@k of LINE-LINE at higher values of k. It can be explained as the DeepWalk algorithm uses a random walk to sample neighbors of a node to gather the structural information, however, it doesn't take into the account the weights of the edges, because of which it can not leverage the CF values for cross-network links. LINE-LINE performs relatively well as it preserves the first-order proximity proportional to the CF values and achieves a Hit-Rate@1 of about 0.75. However, using the bias parameters from node2vec to better represent structural equivalence, we gain a significant advantage over LINE-LINE and LINE-DeepWalk to get a Hit-Rate@k of around 0.99 for most of the k values. By biasing the walk towards detecting cross-linkages and weighting the transition probabilities towards the CF values, LINE-node2vec gives an optimal representation of cross-linkages that are placed closer to each other in the embedding space.

5.3 Comparison with the Baselines

Finally, we evaluate our best performing combination of LINE-node2vec in the NeXLink framework with competing baselines. Along with the structural information, REGAL [6] allows using attribute information for node similarity. However, when comparing with our approach, we only use the structural information from the real-world dataset, described in 4.2. We also compare our approach with IONE [10] that takes two network graphs as input and produces node embeddings based on the follower and followee relationship among the nodes. We employ our best performing LINE-node2vec technique and elaborate on its performance on the real-world dataset. Figure 4c illustrates the performance of the baselines, as compared to our approach. Given the evaluation of IONE uses the same dataset, we were to successfully reproduce their results, as mentioned in their work [10]. However, it still underperforms when compared to the other approaches. REGAL achieves the highest Hit-Rate@1 as it uses node degrees to capture structural similarities, and node degrees partially contribute to the CF values. However, it still fails to leverage the essential CF values completely, as one of its limitations is not being able to take the edge weights into account. Therefore, its performance stagnates at higher k values. In contrast, LINE-node2vec starts below REGAL at $k = 1$, but achieves higher Hit-Rate@k values with the increase in k. LINE-node2vec learns both within-graph and cross-graph structural features from the real-world dataset and effectively represents the similarities in low-embedding space.

6 Limitations, Discussions and Future Work

While developing NeXLink, we identify some of the limitations of our approach. Firstly, we only include structural information indicating standard connections in the two networks, to learn node representations. We can utilize more rich features to gain more comprehensive node representations. Secondly, an essential step in our approach is to create cross-network pairs, which we accomplish using random walks. We can evaluate more efficient ways to sample the cross-network pairs. And last, the two significant limitations of node embeddings are (a) the need to define an objective function, based on which we learn the embeddings, and (b) node embedding models are transductive, which means that it is not possible to generate the embeddings for the nodes that we do not see during the training. To this end, we can consider the use of graph neural networks [16, 5].

In this work, we propose our *NeXLink* framework for effective representation of cross-network linkages across social networks. Our framework works by preserving the local structure of nodes within the same social network and global structure manifested in the form of common friends exhibited by nodes participating in cross-network linkages. We perform an extensive evaluation of our approach on two datasets, one of which we augment from Facebook social network, and the other comprises of Twitter-Foursquare node pairs. Given that NeXLink framework is flexible, we explored numerous state-of-the-art node embedding algorithms and found that LINE-node2vec performs the best when provided with supervisory information of common friendship. It performs with average Hit@1 rate of 98% across all configurations of the augmented dataset. Further our approach outperforms state-of-the-art node representation algorithms LINE and DeepWalk for representing cross-network linkages across the social networks. This can be primarily attributed to the fact that our approach preserves local and global cross-network links more effectively than these previous approaches which are specifically targeted to perform well on single networks. Our framework works better than other state-of-the-art node embedding approaches like IONE and REGAL for identity linkage on a real-world dataset. This is because our framework performs biased walks in accordance with the common friendship metric for cross-network links.

As future work, we can include node attributes derived from user profile configuration and user content in the NeXLink framework and their impact on performance measured. At the algorithmic level, deep learning-based approaches for node embedding would also be a right direction to explore.

References

1. Edwards, M., Larson, R., Green, B., Rashid, A.: Panning for gold: automatically analysing online social engineering. Computers & Security **39**, 396–405 (2013)
2. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 855–864. ACM (2016)
3. Guo, L., Zhang, D., Cong, G., Wu, W., Tan, K.L.: Influence maximization in trajectory databases. IEEE Transactions on Knowledge and Data Engineering **29**(3), 627–641 (2016)
4. Hagberg, A., Swart, P., S Chult, D.: Exploring network structure, dynamics, and function using networkx. Tech. rep., Los Alamos National Lab.(LANL), Los Alamos, NM (United States) (2008)
5. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: Advances in Neural Information Processing Systems. pp. 1024–1034 (2017)
6. Heimann, M., Shen, H., Safavi, T., Koutra, D.: Regal: Representation learning-based graph alignment. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management. pp. 117–126. ACM (2018)
7. Kong, X., Zhang, J., Yu, P.S.: Inferring anchor links across multiple heterogeneous social networks. In: Proceedings of the 22nd ACM international conference on Information & Knowledge Management. pp. 179–188. ACM (2013)
8. Liang, S., Zhang, X., Ren, Z., Kanoulas, E.: Dynamic embeddings for user profiling in twitter. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 1764–1773. ACM (2018)
9. Liu, J., He, Z., Wei, L., Huang, Y.: Content to node: Self-translation network embedding. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 1794–1802. ACM (2018)
10. Liu, L., Cheung, W.K., Li, X., Liao, L.: Aligning users across social networks using network embedding. In: IJCAI. pp. 1774–1780 (2016)
11. Lu, C.T., Shuai, H.H., Yu, P.S.: Identifying your customers in social networks. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management. pp. 391–400. ACM (2014)

12. Malhotra, A., Totti, L., Meira Jr, W., Kumaraguru, P., Almeida, V.: Studying user footprints in different online social networks. In: Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012). pp. 1065–1070. IEEE Computer Society (2012)
13. Man, T., Shen, H., Liu, S., Jin, X., Cheng, X.: Predict anchor links across social networks via an embedding approach. In: IJCAI. vol. 16, pp. 1823–1829 (2016)
14. McPherson, M., Smith-Lovin, L., Cook, J.M.: Birds of a feather: Homophily in social networks. Annual review of sociology **27**(1), 415–444 (2001)
15. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 701–710. ACM (2014)
16. Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G.: The graph neural network model. IEEE Transactions on Neural Networks **20**(1), 61–80 (2008)
17. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: Line: Large-scale information network embedding. In: Proceedings of the 24th International Conference on World Wide Web. pp. 1067–1077. International World Wide Web Conferences Steering Committee (2015)
18. Viswanath, B., Mislove, A., Cha, M., Gummadi, K.P.: On the evolution of user interaction in facebook. In: Proceedings of the 2nd ACM workshop on Online social networks. pp. 37–42. ACM (2009)
19. Wang, D., Cui, P., Zhu, W.: Structural deep network embedding. In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 1225–1234. ACM (2016)
20. Wang, Y., Feng, C., Chen, L., Yin, H., Guo, C., Chu, Y.: User identity linkage across social networks via linked heterogeneous network embedding. World Wide Web pp. 1–22 (2018)
21. Xie, W., Mu, X., Lee, R.K.W., Zhu, F., Lim, E.P.: Unsupervised user identity linkage via factoid embedding. In: 2018 IEEE International Conference on Data Mining (ICDM). pp. 1338–1343. IEEE (2018)
22. Xu, L., Wei, X., Cao, J., Yu, P.S.: On exploring semantic meanings of links for embedding social networks. In: Proceedings of the 2018 World Wide Web Conference on World Wide Web. pp. 479–488. International World Wide Web Conferences Steering Committee (2018)
23. Zafarani, R., Liu, H.: Users joining multiple sites: Friendship and popularity variations across sites. Information Fusion **28**, 83–89 (2016)
24. Zhang, J., Kong, X., Yu, P.S.: Transferring heterogeneous links across location-based social networks. In: Proceedings of the 7th ACM international conference on Web search and data mining. pp. 303–312. ACM (2014)
25. Zhang, J., Philip, S.Y.: Integrated anchor and social link predictions across social networks. In: IJCAI. pp. 2125–2132 (2015)
26. Zhang, J., Yu, P.S., Zhou, Z.H.: Meta-path based multi-network collective link prediction. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 1286–1295. ACM (2014)