ECAI 2024
U. Endriss et al. (Eds.)

doi:10.3233/FAIA241054

Entity Matching Across Small Networks Using Node Attributes

Zahra Ahmadi[a], Zijian Zhang[a], Hoang H. Nguyen[a], Sergio Burdisso[b], Srikanth Madikeri[b], Petr Motlicek[b], Erinc Dikici[c], Gerhard Backfried[c], Marek Kovac[d], Květoslav Maly[d] and Daniel Kudenko[a]

[a]L3S Research Center, Leibniz Universität Hannover, Germany
[b]Idiap Research Institute, Martigny, Switzerland
[c]HENSOLDT Analytics, Vienna, Austria
[d]Phonexia, Brno, Czech Republic

Abstract. Entity matching, also known as user identity linkage, is a critical task in data integration. While established techniques primarily focus on large-scale networks, there are several applications where small networks pose challenges due to limited training data and sparsity. This study addresses entity matching in the field of criminology, where small networks are common and the number of known matching nodes is restricted. To support this research, we exploit a multimodal dataset, collected as part of a security-related project, consisting of an intercepted telephone calls network (i.e., ROXSD data) and a network of social forum interactions (i.e., ROXHOOD data) collected in a simulated environment, although following real investigation scenario. To improve accuracy and efficiency, we propose a novel approach for entity matching across these two small networks using node attributes. Existing techniques often merely focus on topology consistency between two networks and overlook valuable information, such as network node attributes, making them vulnerable to structural changes. Inspired by the remarkable success of deep learning, we present UGC-DeepLink, an end-to-end semi-supervised learning framework that leverages user-generated content. UGC-DeepLink encodes network nodes into vector representations, capturing both local and global network structures to align anchor nodes using deep neural networks. A dual learning paradigm and the policy gradient method transfer knowledge and update the linkage. Additionally, node attributes, such as call contents and forum exchanged texts, enhance the ranking of matching nodes. Experimental results on ROXSD and ROXHOOD demonstrate that UGC-DeepLink surpasses baselines and state-of-the-art methods in terms of identity-match ranking. The code and dataset are available at https://github.com/erichoang/UGC-DeepLink.

1 Introduction

Capturing the dynamics of criminal groups is crucial for generating actionable insights in intelligence monitoring. However, despite its theoretical and practical significance, there is a notable research gap in addressing this problem. This gap is particularly evident when dealing with data originating from multiple sources, where leveraging such data can yield augmented information. These diverse data sources encompass a wealth of contextual information, including voice, text, and images, as well as valuable network-related (i.e., traffic) data. By representing the dynamics of criminal interactions as graphs, we can identify cross-platform account matching to analyze the networks. This essential step, known as user identity linkage, aims to enhance identity verification and privacy protection. However, conducting research in the field of criminology poses challenges due to the high security and privacy concerns associated with criminal data. As a result, there is a scarcity of publicly available data for the criminology research community. To address this limitation, this work is built on a multimodal dataset specifically designed to facilitate research and exploration in the realm of criminal networks[1]. This dataset comprises two interconnected networks: an intercepted telephone calls network (ROXSD) and a network of social forum interactions (ROXHOOD), both collected in a simulated environment while following real scenarios of criminal activities [19]. ROXSD provides insights into communication patterns and relationships among individuals involved in criminal activities through intercepted telephone calls, whereas ROXHOOD captures users' interactions within social forums, allowing for the analysis of online discussions and connections within criminal groups. The dataset includes some overlaps in both target and non-target individuals, enhancing its applicability and realism. Together, both datasets offer a valuable resource for future studies in the field of criminology, enabling researchers to investigate and analyze criminal networks within a controlled yet realistic environment.

Existing approaches to address the problem of user identity linkage primarily fall into two main categories: feature-based approaches that involve manual feature engineering based on domain knowledge and have a more historical nature [37] and network embedding techniques that leverage recent advancements in graph neural networks to preserve the proximity of users with similar relationships [41, 12]. However, these methods have certain limitations when applied to intelligent criminal monitoring. They either rely heavily on high-quality user-generated content, including user profiles, or face challenges related to insufficient training data when focusing solely on network topology. Moreover, these methods predominantly focus on large-scale online social networks like Facebook, Twitter, and LinkedIn. However, in the context of intelligent criminal monitoring, the target groups often represent a minority, and data collection needs to be limited to ensure that privacy concerns are respected. Our initial study related to ROXANNE project was done through applying speaker identification step while leveraging the frequency

[1] https://www.roxanne-euproject.org/data

of previous interactions extracted from a graph [8]. To address the existing issues, this paper introduces a comprehensive framework, UGC-DeepLink, which takes into account the heterogeneity of users' activities and behaviors across different sources. It aims to capture latent semantic relationships among users based on network structures and leverages user-generated content to enhance the ranking of matching nodes. The framework consists of three main components: (1) a network sampling component that generates training sequences while preserving the maximum network structure and creates a node embedding, representing each node in the network as a low-dimensional vector, (2) a deep neural network that learns a non-linear transformation for aligning users across networks using anchor nodes, with a dual-learning process that improves the performance of user identity linkage and enhances supervised training, and (3) a re-ranking strategy based on user-generated content to stabilize the outcomes, particularly when labeled samples are limited. The proposed UGC-DeepLink framework builds upon the earlier method called DeepLink [44]. While DeepLink enhances node-matching results by including mapping functions and receiving anchor nodes, it alone does not yield stable results in small networks with limited training pairs. In real-world criminal investigation scenarios, investigators often possess knowledge of only a limited number of matched nodes and require stable results from the machine learning method on the remaining target nodes. Our experimental analysis demonstrates that the UGC-DeepLink framework successfully meets this goal.

2 Multi-Modal Data Collection

Our dataset comprises two parts that we explain in the following:

Intercepted Calls Network (ROXSD): The ROXSD story is built on a drug dealing case in which a group of criminals communicates with each other over the telephone [22]. Their calls are intercepted (wire-tapped) by several (fictional) police organizations. The wire-tapped data also includes a number of "innocent" people communicating with the criminals and with each other in several languages. For instance, some offenders speak Czech internally while planning local criminal activities but use (possibly heavily accented) English when planning transnational activities with others. The topic of each call was manually scripted, and the characters were role-played by voice actors. The scenario consists of some fictional interconnected cases:

- Case 1. Two university students (C01_M, G01_M) in Prague are suspected of selling drugs. G01_M speaks both German and English in his calls. The police wire-tap two of C01_M's mobile telephones and find out that he is in contact with other individuals who are either users or distributors of drugs, and the communication is mainly in Czech or Slovak. He also communicates in English with one of the main dealers, R01_M. Most of the communication occurs at the point where the drugs change hands.
- Case 2. The police suspect two Vietnamese guys, V01_M and V02_M, dealing in large quantities of drugs, and V02_M may have a production site. They mostly speak Vietnamese and frequently call each other. They are also often in touch with two other Vietnamese contacts. V01_M starts communicating with an unknown person in English, planning a large delivery of drugs.
- Case 3. The links between cases 1 and 2 are uncovered during the individual investigations. Then, the police realized that the criminal activities spread across borders to Austria and Germany. C07_F replaces C01_M and starts close interaction with C06_M, who often travels in Czechia and abroad. The two extend their activities in Germany and connect with G01_M and with G03_M, respectively. C06_M has multiple intercepted calls with G03_M. The police decide to intercept the phone number of the hotel where they are staying and the phone number of a bar where they are attending, even though most of the calls from and to these numbers are non-target communication. C06_M transfers further drugs from the Vietnamese lab to Germany for G03_M. Later, the dealers register themselves in the darknet "drug" forum called "ROXHOOD" and start messaging other people. For example, they share information about a party in a bar in Munich where potential drug users and distributors meet. As an outcome, the narcotics business unfolds to other countries such as France and Greece.

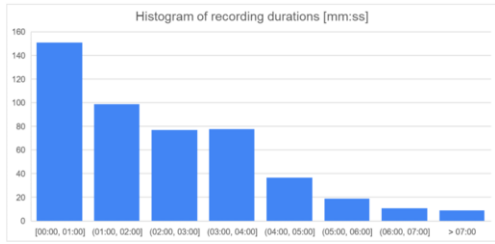

Figure 1: Histogram of recording durations.

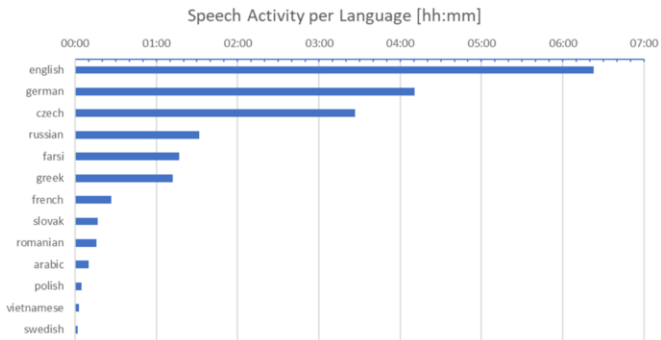

Figure 2: Speech activity per language.

The complete scenario of the ROXSD calls contains 432 intercepted telephone conversations of over 18 hours of speech, saved in 8kHz, 16-bit, stereo wave files. Figure 1 shows the histogram of recordings with respect to their durations.

ROXSD is multilingual not only on the call level but also within the calls. A conversation may start in one language and continue in a different language. Moreover, the language of conversation may change when the phone is handed over to another person. There are also cases where the speakers switch the language precipitously briefly and then switch back. A total of 15 languages are spoken in the calls. Thirteen of them are real conversations, whereas two are only single phrases or brief sentences. Figure 2 shows the distribution of speech activity across languages in the dataset.

The ROXSD calls are also complemented with a set of ground truth information, which we call the "metadata". The metadata set consists of several categories such as *speaker data*, *call data*, *call transcripts*, *NLP annotations*, and other information. The information in the metadata set is anonymized/pseudonymized wherever possible.

The ROXSD network, as illustrated in Figure 3 (a), has been constructed using phone call records. The network primarily consists of two types of nodes representing target and non-target speakers. Furthermore, certain speakers with undefined behaviors or relationships in the network, which are common during investigations, are

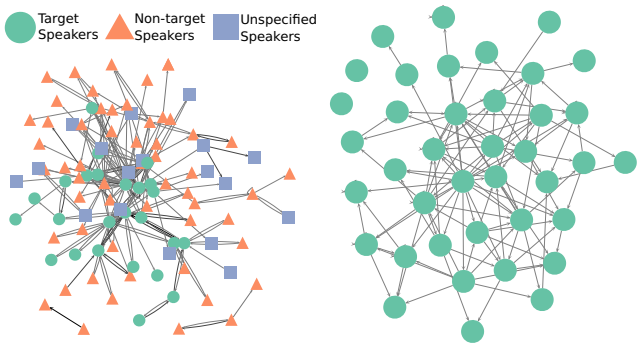

Figure 3: (a) The ROXSD network structure. The thickness of the edges indicates the frequency of calls between speakers. (b) The ROXHOOD network structure.

marked as unspecified nodes. After data cleaning and preprocessing, the ROXSD network comprises a total of 104 nodes and 208 edges. Twenty-five of these nodes are target persons, 60 are non-target persons, and 19 are unspecified characters. Note that the speakers are not necessarily the owners of the respective telephone numbers. The weight of an edge between two speakers increases with the frequency of their calls. We use node2vec for the node embedding based on the topological structure and the *word-based* embedding without fixing out-of-vocabulary (OOV) terms.

Drug Forum Network (ROXHOOD): ROXHOOD is an extension of ROXSD that mimics social media communications. It was designed as a forum website where the users exchange messages about several topics with each other and in which legal and illegal content coexist. In order to avoid any legal and ethical issues arising from accessing and working on publicly available social media platforms, ROXHOOD was specifically built as a simulation environment for data collection over a fully featured mock-up website developed in Misago[2]. It was accompanied by Rocraw, a tool to collect the available data from the ROXHOOD threads and to store them in an Elasticsearch instance.

Participants could create an account with a username, password, and a fake email account. Registered users were allowed to start a new thread, comment, or search and also to set up a private thread with their invitees. They could write messages in text format, upload files of various formats (e.g., docx, jpeg, mpeg), share links, comment on threads below the initial post, and mention each other (via @usernameX). Moderators could also create categories together with an unlimited number and depth of subcategories.

Data could be downloaded directly in a text file, including metadata information such as *thread url*, *thread title*, *number of posts* (per thread), *posts*, *thread creator*, *mentions-replies* for each post, *message*, *user*, *date*, *url*, *has image* (true/false), *has attachment* (true/false), *contains url* (true/false), *image urls*, *attach urls*, and *message urls*. In the ROXHOOD set, there are 32 public threads and 333 public posts from 48 registered users. The content was analyzed by named entity recognition, and 110 entities were extracted. The entities were about persons (21), locations (49), and time (40). In addition to the text messages, ROXHOOD also contains video messages with spoken content in English. Across the 23 videos, 107 utterances and 91 entities were identified. Here, the number of persons, locations, and time entities are 44, 26, and 21, respectively.

The ROXHOOD network is constructed based on the intercommunication patterns amongst users on the drug forum. Connections are established when a user replies to, quotes, or mentions another user or a specific location within a thread, post, or comment. Additionally, users are considered connected to thread creators when they start following a thread on the forum. The resulting ROXHOOD network from crawled data consists of 40 nodes and 121 edges, as depicted in Figure 3 (b). It is worth noting that 16 users within the ROXHOOD network have also been identified as speakers in the ROXSD network. These overlapping nodes serve as ground-truth data for training and evaluation in our experiments. Similar embedding algorithms, as employed in the ROXSD network, are used in this context as well.

[2] https://misago-project.org/

3 Entity Matching Framework

In this paper, we focus on the problem of entity matching between ROXSD and ROXHOOD. Formally speaking, we consider two graphs, each of them represented by $G = (V; E)$, where V is the set of vertices representing a user, and E is the set of edges connecting users. Each network is represented with a unique latent user space according to the probabilistic distributions of its nodes, and our goal is to link and match entities based on that:

Entity Matching Definition: Given any two networks G_s and G_t, the goal is to predict all the pairs of entities u_s and u_t, chosen from $\mathcal{U}_s$ and $\mathcal{U}_t$ respectively, belong to the same person (i.e., $u_s = u_t$). That is, learning a binary function $\Phi_U : \mathcal{U}_s \times \mathcal{U}_t \rightarrow \{0; 1\}$ such that

$$\Phi_U(\mathcal{U}_s, \mathcal{U}_t) = \begin{cases} 1 & u_s = u_t \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

where $\Phi_U(\mathcal{U}_s, \mathcal{U}_t) = 1$ means a correct linkage. However, in practice, finding a perfect Φ_U function is difficult to obtain. Hence, we try to find an approximate graph mapping function. Generally, the mapping function Φ is unknown for a given G, and our objective is to learn a bilateral mapping (Φ and Φ^{-1}) such that the two networks are aligned by maximizing the similarity of all aligned pairs.

We first extract samples from a network, then embed nodes into latent space, and finally learn entity linkage via supervised dual learning. Although this general approach could achieve good results in large networks, the model's performance is not always stable in small networks with a limited number of training samples due to the random nature of the method and specific graph structures. Therefore, we propose a modified solution, UGC-DeepLink, for small multimodal networks (Figure 4). The detailed description of each step is as follows:

3.1 Node Embedding

Nodes are embedded into a latent space by generating multiple sequences using several rounds of random walks for each node, $u_i \in V$. Random walks can implicitly capture and encode the hidden basic structural network information and relationships among the nodes and reduce the sensitivity of calculations to slight changes in a graph. At each step, a node u_i is selected, and we proceed along a randomly selected edge until length L is reached. Generally, this step could be time-consuming, especially in biased random walks like LINE [34] and node2vec [9]. However, criminal networks are relatively small, and using traditional random walks and the sampling step do not add a significant burden.

Once node sequences are sampled for node u_i in iteration r, we use the Skip-gram model [16] to update its representation and predict the context of a node. In graph representation, the skip-gram model

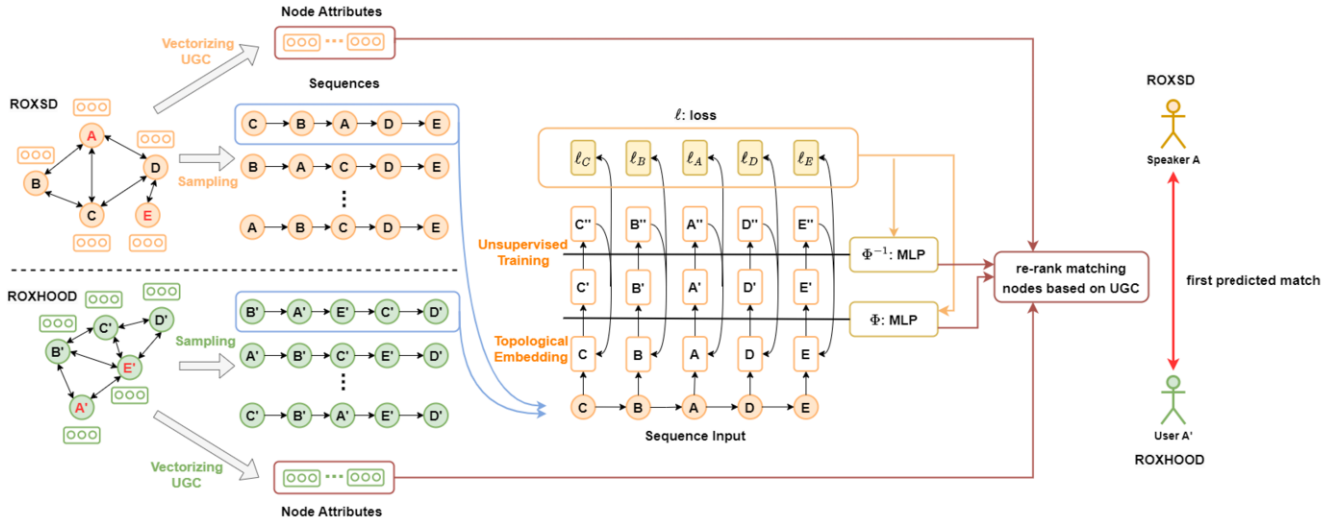

Figure 4: UGC-DeepLink Architecture: it learns a dual-mapping deep neural network on the node embedding of a training sequence and returns the final entity matching ranking while considering the user-generated content embeddings.

maximizes the average log probability in a given sequence of node $u_1, u_2, ..., u_m \in \mathcal{G}$:

$$\frac{1}{m}\sum_{t=1}^{m}\sum_{j=-w}^{w}\log p(u_{t+j}|u_t), j \neq 0, \tag{2}$$

where w is the sliding window size, longer windows on training nodes achieve higher accuracy at the cost of longer training time. The conditional probability is defined as the occurrence of the j-hop neighbor, u_{t+j}, given node u_t:

$$p(u_{t+j}|u_t) = \frac{\exp\left(\mathbf{v}_{u_{t+j}}^T \mathbf{v}'_{u_t}\right)}{\sum_{i=1}^{m}\exp\left(\mathbf{v}_{u_i}^T \mathbf{v}'_{u_t}\right)}. \tag{3}$$

v_{u_i} and v'_{u_i} are the input and output vector representations of node u_i, and m is the network's node size. In order to improve training efficiency, a negative sampling strategy (as proposed in [17]) is employed. Each node is sampled with probability $p_n(u)\ d_{u_i}^{3/4}$, where d_{u_i} is the degree of node u_i and a stochastic gradient descent algorithm is used to solve Equation 2.

3.2 Learning Mapping Function

Once node embeddings are calculated for each network, the mapping functions between two networks should be learned using Multi-Layer Perceptron (MLP) and based on the labeled anchor nodes. The mapping function $\Phi(\mathbf{v}(u_i))$ is learned by minimizing the loss function that is based on the cosine similarity of the mapped vector and the embedding representation ($\mathbf{v}(u_j)$):

$$l(\mathbf{v}(u_i), \mathbf{v}(u_j)) = \min(1 - \cos(\Phi(\mathbf{v}(u_i)), \mathbf{v}(u_j))). \tag{4}$$

The loss ranges from zero, meaning exactly the same, to two, meaning exactly opposite. In practice, a batch size of $h (h << n)$ vectors is fed to MLP at each step until the n anchor nodes are visited.

The MLP-based anchor mapping is a one-way process since a reverse mapping, Φ^{-1}, does not take into account the training anchor nodes. Therefore, we use dual learning to exploit these anchor nodes fully and improve two mapping functions by leveraging the duality of Φ and Φ^{-1}. Hence, we use two steps:

1. *Unsupervised Pretraining*: For each anchor node u_b (labeled and unlabeled), we first obtain $v'(u_b)$ via the mapping, $\mathbf{v}'(u_b) = \Phi(\mathbf{v}(u_b))$, and then map back via $\Phi^{-1}(\mathbf{v}'(u_b))$ to get a vector $\mathbf{v}''(u_b)$. Similar to autoencoders, the loss of this auto-mapping is calculated based on the difference between $\mathbf{v}(u_b)$ and $\mathbf{v}''(u_b)$. Moreover, the anchor nodes in one network are blinded with respect to the other.
2. *Supervised Learning*: The labeled anchor nodes are used to improve the mapping function Φ and Φ^{-1} by playing a dual learning game. It starts from u_a of the source network. We use Φ to map its vector on the target space and search its k-nearest vectors $S(\mathbf{v}'(u_a)) = Top(\Phi(\mathbf{v}(u_a)))$, indicating the most similar k embedding vectors of the target anchor nodes. The target agent then computes a reward $r_{s,t}^a = \frac{1}{k}\sum_{i=1}^{k}\log(\cos(\mathbf{v}(u_i), \mathbf{v}'(u_a)) + 1)$ based on the similarity of two vectors, which ranges between zero to two. Mapping the exact identity of u_a is difficult; therefore, we search and average over top-k vectors.
   Intuitively, the reward function of the dual mapping is calculated by:

$$r_{t,s}^a = \frac{1}{k}\sum_{i=1}^{k}\log(\cos(\Phi^{-1}\mathbf{v}'(u_i), \mathbf{v}(u_a)) + 1), \tag{5}$$

   where it measures the average similarity between $\Phi^{-1}\mathbf{v}'(u_i)$ and $\mathbf{v}(u_a)$. Thus, the action value of selecting user (state) u_a is a linear combination of $r_{s,t}^a$ and $r_{t,s}^a$, which indicates the estimated probability of correct real identity linkage by the mapping functions. The expected reward for the h^{th} batch is calculated as:

$$\mathbb{E}[r_h] = \sum_{a=1}^{[n/h]}(\alpha r_{s,t}^a + (1-\alpha) r_{t,s}^a), \tag{6}$$

   where α is learned and tuned in the training.

3.3 Re-rank Matched Nodes Using Node Attributes

Leveraging features of the structure-based user identity linkage approach can improve the linkage efficiency. In the existing network structure embedding, some embedding vectors for neighboring nodes may be too close to distinguish from one another. Accordingly, considering node attributes, such as usernames, and embedding the features irrelevant to network structures into the network embedding

vector obtained from Equation 2 may be useful for discriminating among the top-k candidates.

In our proposed methodology, we leverage vectorized individual user-generated content (UGC) as their node attributes. We employ four distinct methods for generating these attributes:

- **sparse-word:** We compile a single document for each user by concatenating all of their generated content. Subsequently, we normalize the text by removing Unicode accents and vectorizing it into a word-level sparse vector. Each position in the vector corresponds to a word, with its value representing the *tf-idf* value [28][3].
- **sparse-char:** Similar to the previous method, we create a single document for each user, but instead of a word-level vector, we generate a char-level *tf-idf* sparse vector. Each position in the vector corresponds to char 4-grams and 5-grams found in the user content, weighted by *tf-idf*. Our experimental results indicate that these two configurations yielded the best performance.
- **dense-avg:** We convert the individual content of each user into a dense vector using a multilingual BERT model [6]. Specifically, we employ the *paraphrase-multilingual-mpnet-base-v2* model from *Sentence-BERT* [26] to obtain a sentence embedding for each user-generated content. The user's overall dense vector is then obtained by averaging all content embeddings, effectively representing the centroid of the user's content embeddings.
- **dense-weighted:** Similar to the previous method, we use a weighted average to obtain the final user-level embedding. The weight assigned to each document is inversely proportional to its cosine similarity to all other users' content embeddings. In other words, content embeddings that are more unique or distant from other users' content carry more weight on average, exerting a greater influence on the final vector.

Algorithm 1 presents the pseudo-code of UGC-DeepLink. $(G_1(V_1, E_1), G_2(V_2, E_2))$ represent the pair of graphs, $\mathcal{C}_1$ and $\mathcal{C}_2$ indicate the user-generated content of the nodes in V_1 and V_2 respectively, and $L_{\text{train}} = \{0,1\}^{|V_1||V_2|}$ denotes the training data for the known node linkages. We calculate the prior similarities, denoted as S^{prior}, between every pair of nodes by computing the pairwise cosine similarity of their node attributes (line 3). For a given node u_i that should be matched, we initially predict all its potential candidate matches using the MLP model (lines 7-10). Subsequently, we re-rank its top-k candidates based on their prior similarities (lines 11-13). The algorithm returns the affinities between the two graph nodes. Our approach is based on the assumption that both the local topology of the networks and the behavioral patterns of real individuals contribute to the identification of their connections.

Algorithm 1 UGC-DeepLink Algorithm

1: **Input**: $G_1(V_1, E_1), G_2(V_2, E_2), \mathcal{C}_1, \mathcal{C}_2, L_{\text{train}}$
2: $E_1^{\text{prior}}, E_2^{\text{prior}} = vectorize(\mathcal{C}_1), vectorize(\mathcal{C}_1)$
3: $S^{\text{prior}} = pairwise\_cos\_similarity(E_1^{\text{prior}}, E_2^{\text{prior}})$
4: $\Phi = \arg\min_\Phi E_{(u_i,u_j)\in L_{\text{train}}} l(\Phi(\mathbf{v}(u_i)), \mathbf{v}(u_j))$
5: affs := []
6: **for** $u_i \in V_1$ **do**
7: S^{dl} := []
8: **for** $u_j \in V_2$ **do**
9: S^{dl}.push($\cos(\Phi(\mathbf{v}(u_i)), \mathbf{v}(u_j))$)
10: **end for**
11: **for** $u_j \in \arg sort_{\text{desc}}(S^{\text{dl}})[1:k]$ **do**
12: $S^{\text{dl}}[u_j] = S^{\text{prior}}[u_j] + 1$
13: **end for**
14: affs.push(S^{dl})
15: **end for**
16: **Return** affs

4 Experiments

We conducted a comparative analysis of our approach against four other comparison methods: TF-IDF, IsoRank, NetAlign, and vanilla DeepLink. Below is a brief description of these methods:

TF-IDF: This baseline relies only on the vectorized UGC node attribute, here, the content of calls or forum texts. Initially, a linear model is trained on the training set to establish a transformation between the node embeddings from two graphs. This is done under the assumption that the UGC vector space exhibits linear correlation. During the inference, the source node's embedding is transformed to the target node's embedding space, and potential candidates are ranked using cosine similarity.

IsoRank [31]: This algorithm approximates the objective of the network alignment problem by formulating it as an integer quadratic program without direct consideration for the matching constraints. It aims to find a matrix Z that satisfies the following equation:

$$\gamma A^T D_A Z D_B B + (1-\gamma)W = Z,$$

where A and B represent the adjacency matrices of the two graphs, and D_A and D_B are diagonal matrices of their degrees. Here, $W_{i,j} = w_{i,j}$ is binary, taking the value one if there is a linkage between u_i from the first graph and u_j from the second graph. The resulting values, $Z_{i,j}$, provide a heuristic likelihood of node matching between u_i and u_j.

NetAlign [17, 2]: This approach employs a message-passing algorithm based on the influence of a node's closest neighbors rather than distant ones. It predicts an affinity matrix for nodes in both graphs, allowing for ranking based on the affinity scores.

Vanilla DeepLink [44] uses an MLP to generate a predicted vector for each node in the source graph only based on network structure. Candidates are ranked based on cosine similarity.

4.1 Experimental Setting and Results

We fixed the parameters of each method as follows: In the UGC-DeepLink framework, the node2vec embedding dimension, the MLP hidden dimension, and the initial learning rate are set to 500, 800, and 0.05, respectively. The number of walks, walk length, and total training steps in node2vec are 80, 20, and 1000, respectively. The value of k in the top-k re-ranking step is set to 10. The vanilla Deeplink uses the same parameters as UGC-DeepLink. We used the best parameters of IsoRank and NetAlign, as reported in their papers, and scikit-learn to train a linear regression model in the TF-IDF baseline. We used 80% of the 16 pairs of matching nodes for training and 20% for testing.

When evaluating entity matching methods, various prediction and ranking metrics are commonly used. Ranking metrics are suitable for assessing approaches that provide a top-k ranking list of potential matching user identities, as opposed to selecting only one candidate. In our experiments, we evaluate methods based on some commonly used ranking metrics such as AUC, Mean Reciprocal Rank (MRR), and Success@k and report the results in Table 1. We should note that

[3] Implementation is carried out using the *TfidfVectorizer* class from *Scikit-learn* [24].

Method	AUC	Mean Reciprocal Rank (MRR)	Success@k (k=2)
TF-IDF	0.42	0.40	0.50
IsoRank	0.54	0.50	0.50
NetAlign	0.50	0.52	0.50
Vanilla DeepLink	**0.75**	0.56	**0.75**
UGC-DeepLink (sparse-word)	N.A.	0.69	**0.75**
UGC-DeepLink (sparse-char)	N.A.	0.65	0.50
UGC-DeepLink (dense-avg)	N.A.	0.52	0.50
UGC-DeepLink (dense-weighted)	N.A.	**0.81**	**0.75**

Table 1: Performance comparison of node matching methods on ROXSD and ROXHOOD.

Method	train%	MRR	Success@k (k=2)
UGC-DeepLink (sparse-word)	4/12	0.50	0.50
	5/12	**0.69**	**0.75**
	6/12	**0.69**	**0.75**
	7/12	0.65	0.50
	8/12	0.52	0.50
	9/12	**0.69**	**0.75**
UGC-DeepLink (sparse-char)	4/12	0.52	0.50
	5/12	0.48	0.25
	6/12	**0.69**	**0.75**
	7/12	0.52	0.50
	8/12	0.52	0.50
	9/12	0.65	0.50
UGC-DeepLink (dense-avg)	4/12	0.50	0.50
	5/12	**0.83**	**0.75**
	6/12	0.69	**0.75**
	7/12	0.56	**0.75**
	8/12	0.52	0.50
	9/12	0.52	0.50
UGC-DeepLink (dense-weighted)	4/12	0.50	0.50
	5/12	0.69	**0.75**
	6/12	0.69	**0.75**
	7/12	0.58	**0.75**
	8/12	0.65	0.50
	9/12	**0.81**	**0.75**

Table 2: Impact of the number of train/val pairs on the performance of different versions of UGC-DeepLink. The best results for each method are in bold.

AUC is not applicable for UGC-DeepLink since the top-k results of the Vanilla Deeplink are re-ranked by their UGC embeddings. Therefore, there are no homogenous classification probabilities to calculate the AUC.

We observe that the UGC-DeepLink framework performs best compared to all the baselines and state-of-the-art methods. This is expected since it is the only approach that considers both the network structure and the shared text contents as node attributes. As expected, TF-IDF performs worst since it uses a simple linear regression and only considers the UGC content. Vanilla DeepLink performs better than IsoRank and NetAlign, indicating the rich capacity of deep neural networks in learning the matching network structure. Among the four types of user-generated content vectors, dense-weighted embeddings demonstrate a notable enhancement in terms of reciprocal ranks (MRR). We further extended our experiments to evaluate how the number of training samples impacts the performance of the UGC-DeepLink framework. As shown in Table 2, BERT-based embeddings (dense-avg and dense-weighted) consistently achieve reciprocal ranks exceeding 0.8, underscoring their effectiveness and superior performance in retrieving the correct match as the first item (on average) compared to tf-idf-based vectors. However, across all variations, the ability to retrieve the correct match within the top two items of all queries reaches up to 0.75. Moreover, apart from dense-weighted user-generated content vectors, the other embeddings do not necessarily require a larger set of training data to perform optimally.

Table 3 presents the results of the UCG-DeepLink for a sample case study involving the matching of ROXSD and ROXHOOD users. The train and validation splits are provided as-is, while the *Test* rows display the prediction output of the UGC-DeepLink. Rows such as `ro05M` - **Tiby** indicate that for the input `ro05M`, the UGC-DeepLink correctly predicts its matching node (Tiby) in the first position. Conversely, the row `en07M_NT` – Makis, Tiby, Sheldon, **scooby** (fail) signifies that for the input `en06M_NT`, the prediction fails because the ground-truth matching (scooby) is ranked at position 4, which exceeds $k = 2$.

Split	roxsd-node	roxhood-node
Train	`de03M_T`	Horus
	`cs06M_T`	Okram
	`cs07F_T`	Kiki
	`fa03F_NT`	jasmine
	`el10F_T`	apo
	`de01M_T`	Elysium
	`de02F_T`	Prinzessin
	`cs17M_T`	Rumreich
	`cs05M_NT`	Pablo
Val	`en05M_NT`	Square
	`el01M_NT`	beavis
	`pl01M_NT`	Driver
Test	`ro05M`	**Tiby**
	`el11M`	**Makis**
	`en07M_NT`	Makis, Tiby, Sheldon, **scooby** (fail)
	`ro03M`	**Sheldon**

Table 3: A sample case study: The full train/val/test data for the ROXSD-ROXHOOD matching is shown.

4.2 A Note on Ethical Concerns

UGC-DeepLink offers improved privacy protection compared to alternative classical methods in criminology by reducing reliance on sensitive personal data for identity verification. It avoids using sensitive data during model training and focuses on node attributes like call contents and forum exchanges, represented as embedded vectors rather than detailed content. This approach strikes a balance between effective identity matching and privacy preservation. However, there is an inherent trade-off between matching accuracy and privacy protection - more accurate matching typically requires more detailed personal data analysis. While UGC-DeepLink minimizes sensitive data use by primarily leveraging network connections, this may result in some compromise in matching accuracy. However, our experiments show promising results in this regard. The exact extent of this trade-off and how it compares to other methods would require further analysis and could be the subject of another study.

5 Related Work

Most existing solutions to the user identity linkage problem are typically divided into three main categories:

1. Attribute-based approaches: These approaches calculate distances among attributes (e.g., usernames, locations, avatars) to infer latent anchor links between user accounts [37, 11]. Similarity measures like Jaro-Winker distance, Jaccard similarity, or Levenshtein distance [30] are used for text-based data, and mean square error, peak signal-to-noise ratio, and Levenshtein distance [14] for graphical data. On the other hand, frequency-based methods leverage statistical patterns to compare distances like the bag-of-words or TF-IDF [41] models. Some methods employ unsupervised approaches to determine username similarity [11], while others use supervised approaches to learn behavioral patterns in

username selection [37]. However, relying solely on usernames for user linkage may lack sufficient precision, and methods like Mu et al.'s [20] delve into the latent user space based on user attributes to determine the intrinsic structure of users. Adequate precision typically demands complete user attribute information. Matrix factorization-based methods and algorithms like IsoRank [31] and NetAlign [1] have also been used but may encounter challenges with sparse and large-scale networks.

2. User Generated Content (UGC) based approaches: These approaches extract user-generated content such as interests, writing style [13],and temporal/spatial trajectories [27, 29] to capture user identity characteristics. Writing style-based approaches work well for text-dominated social networks, while other methods extract core interests through temporal topic extraction [23] or unique activity patterns from location data [27]. However, these frameworks may not be suitable for linking users across different types of social networks.
3. Network-based approaches: These methods leverage network connectivity and topology using network representation learning techniques to calculate similarities between nodes and their features [41, 12, 44]. They analyze either the global network structure or focus on local interaction features between users and their neighbors. Network-feature-based user identity distance can be modeled using *neighborhood-based models* [41, 42] or *embedding-based models* [12, 44, 38]. Neighborhood-based models explicitly represent graphs using shared identified friends [45], in/out neighbors and in/out-degree [21], while embedding-based techniques learn latent representations of networks [34, 15, 43]. Due to the success of network representation learning in many applications, several recent studies have used them to extract latent network features. Network alignment algorithms can be categorized into pairwise network alignment [39, 10, 44, 25], collective network alignment [5, 7], high-order network alignment [18, 32], and hierarchical network alignment [40], depending on the alignment scenario and objectives.

In recent years, embedding-based approaches have gained popularity for learning network structures and latent properties as feature vectors, providing more effective cross-network entity matching than traditional feature engineering. These approaches offer advantages such as requiring minimal or no supervision and reducing the need for expensive and time-consuming feature engineering that relies on domain expertise. They can be categorized into *proximity-based* and *feature-based* methods. Proximity-based approaches extract similarity among individuals within networks and identify similar individuals across networks using first-order [15] and second-order proximity [12]. Feature-based approaches combine social networks as hypergraphs and rank objects based on their likelihood of corresponding to objects in another network [33, 4]. These methods are effective when input networks have rich attributes. Further related topics to cross-network entity matching include *knowledge graph alignment* [46, 35] that aligns entities across different knowledge graphs and *cross-layer dependency inference* that infers the node dependencies in the multi-layered networks [3, 36].

6 Conclusion

This paper introduces a comprehensive multimodal dataset consisting of an intercepted telephone calls network (ROXSD) and a network of social forum interactions (ROXHOOD) in a simulated environment to enhance research in the field of criminology. We then propose UGC-DeepLink, a new framework designed for entity matching in small networks. By incorporating node attributes and deep learning techniques, UGC-DeepLink achieves accurate node matching in small criminal networks. Our experimental results on ROXSD and ROXHOOD demonstrate the effectiveness and stability of UGC-DeepLink in identity-match ranking, even with limited training data in the context of intelligent criminal monitoring. Future work in this area could explore the integration of additional data sources, other forms of embeddings, and the development of more sophisticated deep learning models to improve accuracy and scalability further.

References

[1] M. Bayati, M. Gerritsen, D. F. Gleich, A. Saberi, and Y. Wang. Algorithms for large, sparse network alignment problems. In *Proceedings of the Ninth IEEE International Conference on Data Mining (ICDM)*, pages 705–710, 2009.

[2] M. Bayati, D. F. Gleich, A. Saberi, and Y. Wang. Message-passing algorithms for sparse network alignment. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 7(1):1–31, 2013.

[3] C. Chen, H. Tong, L. Xie, L. Ying, and Q. He. Fascinate: fast cross-layer dependency inference on multi-layered networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 765–774, 2016.

[4] H. Chen, H. Yin, X. Sun, T. Chen, B. Gabrys, and K. Musial. Multi-level graph convolutional networks for cross-platform anchor link prediction. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, pages 1503–1511, 2020.

[5] X. Chu, X. Fan, D. Yao, Z. Zhu, J. Huang, and J. Bi. Cross-network embedding for multi-network alignment. In *Proceedings of the World Wide Web Conference (WWW)*, pages 273–284, 2019.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, and T. Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

[7] B. Du and H. Tong. Mrmine: Multi-resolution multi-network embedding. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 479–488, 2019.

[8] M. Fabien, S. S. Sarfjoo, S. Madikeri, and P. Motlicek. Graph2speak: Improving speaker identification using network knowledge in criminal conversational data. In *1st ISCA Symposium on Security and Privacy in Speech Communication*, pages 10–13, 2021. doi: 10.21437/SPSC.2021-3.

[9] 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*, pages 855–864, 2016.

[10] M. Heimann, H. Shen, T. Safavi, and D. Koutra. Regal: Representation learning-based graph alignment. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 117–126, 2018.

[11] J. Liu, F. Zhang, X. Song, Y.-I. Song, C.-Y. Lin, and H.-W. Hon. What's in a name? an unsupervised approach to link users across communities. In *Proceedings of the sixth ACM International Conference on Web Search and Data Mining (WSDM)*, pages 495–504, 2013.

[12] L. Liu, W. K. Cheung, X. Li, and L. Liao. Aligning users across social networks using network embedding. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, volume 16, pages 1774–80, 2016.

[13] S. Liu, S. Wang, F. Zhu, J. Zhang, and R. Krishnan. Hydra: Large-scale social identity linkage via heterogeneous behavior modeling. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, pages 51–62, 2014.

[14] A. Malhotra, L. Totti, W. Meira Jr, P. Kumaraguru, and V. Almeida. Studying user footprints in different online social networks. In *Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 1065–1070, 2012.

[15] T. Man, H. Shen, S. Liu, X. Jin, and X. Cheng. Predict anchor links across social networks via an embedding approach. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1823–1829, 2016.

[16] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems (NeurIPS)*, 26, 2013.

[17] A. Mnih and Y. W. Teh. A fast and simple algorithm for training neural probabilistic language models. In *Proceedings of the 29th International Conference on International Conference on Machine Learning*, pages 419–426, 2012.

[18] S. Mohammadi, D. F. Gleich, T. G. Kolda, and A. Grama. Triangular alignment (tame): A tensor-based approach for higher-order network alignment. *IEEE/ACM transactions on Computational Biology and Bioinformatics*, 14(6):1446–1458, 2016.

[19] P. Motlicek, E. Dikici, S. Madikeri, P. Rangappa, M. Janosik, G. Backfried, D. Thomas-Aniola, M. Schurz, J. Rohdin, P. Schwarz, M. Kova´c, K. M. y, D. Bobos˘, M. Leibiger, C. Kalogiros, A. Alexopoulos, D. Kudenko, Z. Ahmadi, H. H. Nguyen, A. Krishnan, D. Zhu, D. Klakow, M. Jofre, F. Calderoni, D. Marraud, N. Koutras, N. Nikolau, C. Apostiki, P. Douris, K. Gkountas, E. Sergidou, W. Bosma, J. Hughues, and H. P. Team. Roxsd: The roxanne multimodal and simulated dataset for advancing criminal investigations. In *Odyssey 2024: The Speaker and Language Recognition Workshop*, June 2024.

[20] 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*, pages 1775–1784, 2016.

[21] A. Narayanan and V. Shmatikov. De-anonymizing social networks. In *Proceedings of the 30th IEEE Symposium on Security and Privacy*, pages 173–187, 2009.

[22] H. H. Nguyen, M. Fabien, P. Motlicek, S. Parida, and K. Maly. Roxsd: a simulated dataset of communication in organized crime. In *1st ISCA Symposium on Security and Privacy in Speech Communication*, 2021.

[23] Y. Nie, Y. Jia, S. Li, X. Zhu, A. Li, and B. Zhou. Identifying users across social networks based on dynamic core interests. *Neurocomputing*, 210: 107–115, 2016.

[24] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[25] H. Petric Maretic, M. El Gheche, G. Chierchia, and P. Frossard. Got: an optimal transport framework for graph comparison. *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 32, 2019.

[26] N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

[27] 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 (WWW)*, pages 707–719, 2016.

[28] F. Sebastiani. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47, 2002.

[29] J. Shao, Y. Wang, H. Gao, B. Shi, H. Shen, and X. Cheng. Asylink: user identity linkage from text to geo-location via sparse labeled data. *Neurocomputing*, 515:174–184, 2023.

[30] Y. Shen and H. Jin. Controllable information sharing for user accounts linkage across multiple online social networks. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management (CIKM)*, pages 381–390, 2014.

[31] R. Singh, J. Xu, and B. Berger. Global alignment of multiple protein interaction networks with application to functional orthology detection. *National Academy of Sciences*, 105(35):12763–12768, 2008.

[32] Q. Sun, X. Lin, Y. Zhang, W. Zhang, and C. Chen. Towards higher-order topological consistency for unsupervised network alignment. *arXiv preprint arXiv:2208.12463*, 2022.

[33] S. Tan, Z. Guan, D. Cai, X. Qin, J. Bu, and C. Chen. Mapping users across networks by manifold alignment on hypergraph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28, 2014.

[34] 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 (WWW)*, pages 1067–1077, 2015.

[35] Z. Wang, Q. Lv, X. Lan, and Y. Zhang. Cross-lingual knowledge graph alignment via graph convolutional networks. In *Proceedings of the Conference on Empirical methods in Natural Language Processing (EMNLP)*, pages 349–357, 2018.

[36] Y. Yan, Q. Zhou, J. Li, T. Abdelzaher, and H. Tong. Dissecting cross-layer dependency inference on multi-layered inter-dependent networks. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM)*, pages 2341–2351, 2022.

[37] R. Zafarani and H. Liu. Connecting users across social media sites: a behavioral-modeling approach. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 41–49, 2013.

[38] J. Zhang and P. S. Yu. Pct: partial co-alignment of social networks. In *Proceedings of the 25th International Conference on World Wide Web (WWW)*, pages 749–759, 2016.

[39] S. Zhang and H. Tong. Final: Fast attributed network alignment. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1345–1354, 2016.

[40] S. Zhang, H. Tong, R. Maciejewski, and T. Eliassi-Rad. Multilevel network alignment. In *Proceedings of the World Wide Web Conference (WWW)*, pages 2344–2354, 2019.

[41] Y. Zhang, J. Tang, Z. Yang, J. Pei, and P. S. Yu. Cosnet: Connecting heterogeneous social networks with local and global consistency. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1485–1494, 2015.

[42] Y. Zhang, L. Wang, X. Li, and C. Xiao. Social identity link across incomplete social information sources using anchor link expansion. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, pages 395–408, 2016.

[43] C. Zheng, L. Pan, and P. Wu. Jora: Weakly supervised user identity linkage via jointly learning to represent and align. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[44] F. Zhou, L. Liu, K. Zhang, G. Trajcevski, J. Wu, and T. Zhong. Deeplink: A deep learning approach for user identity linkage. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, pages 1313–1321. IEEE, 2018.

[45] X. Zhou, X. Liang, H. Zhang, and Y. Ma. Cross-platform identification of anonymous identical users in multiple social media networks. *IEEE Transactions on Knowledge and Data Engineering*, 28(2):411–424, 2015.

[46] H. Zhu, R. Xie, Z. Liu, and M. Sun. Iterative entity alignment via knowledge embeddings. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2017.