applied sciences

MDPI

Article

User Identity Linkage across Social Networks by Heterogeneous Graph Attention Network Modeling

Ruiheng Wang [1], Hongliang Zhu [1], Lu Wang [1], Zhaoyun Chen [1], Mingcheng Gao [1] and Yang Xin [1,2,*]

[1] National Engineering Laboratory for Disaster Backup and Recovery, Information Security Center, School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China; ruiheng@bupt.edu.cn (R.W.); zhuhongliang@bupt.edu.cn (H.Z.); wltongxue@bupt.edu.cn (L.W.); chenzhaoyun@bupt.edu.cn (Z.C.); gmc@bupt.edu.cn (M.G.)

[2] Guizhou Provincial Key Laboratory of Public Big Data, Guizhou University, Guizhou 550025, China

* Correspondence: yangxin@bupt.edu.cn

Received: 4 July 2020; Accepted: 5 August 2020; Published: 7 August 2020

Abstract: Today, social networks are becoming increasingly popular and indispensable, where users usually have multiple accounts. It is of considerable significance to conduct user identity linkage across social networks. We can comprehensively depict diversified characteristics of user behaviors, accurately model user profiles, conduct recommendations across social networks, and track cross social network user behaviors by user identity linkage. Existing works mainly focus on a specific type of user profile, user-generated content, and structural information. They have problems of weak data expression ability and ignored potential relationships, resulting in unsatisfactory performances of user identity linkage. Recently, graph neural networks have achieved excellent results in graph embedding, graph representation, and graph classification. As a graph has strong relationship expression ability, we propose a user identity linkage method based on a heterogeneous graph attention network mechanism (UIL-HGAN). Firstly, we represent user profiles, user-generated content, structural information, and their features in a heterogeneous graph. Secondly, we use multiple attention layers to aggregate user information. Finally, we use a multi-layer perceptron to predict user identity linkage. We conduct experiments on two real-world datasets: OSCHINA-Gitee and Facebook-Twitter. The results validate the effectiveness and advancement of UIL-HGAN by comparing different feature combinations and methods.

Keywords: user identity linkage; social network; heterogeneous graph; neural network

1. Introduction

With the rapid development of the Internet and information technology, social networks are becoming more and more indispensable and various. Due to different social networks embody different functions, the purpose and behavior characteristics of users are also different [1]. It is of considerable significance to conduct user identity linkage across social networks. We can comprehensively depict diversified characteristics of user behaviors, accurately model user profiles [2], conduct recommendations across social networks [3], and track user behaviors across social networks by user identity linkage.

The existing methods of user identity linkage can be divided into three categories [4]: User profile-based, user generated content-based, and structural information-based. User profiles mainly include a username, nickname, avatar, etc. Zafarani et al. [5] extracted features of patterns to human limitations, exogenous factors, and endogenous factors from username for user identity linkage. User-generated content includes time information, spatial information, and published content.

Liu et al. [6] extracted time information, spatial information, and others from user-generated content to analyze interest and writing style for user identity linkage. Structural information can be expressed as two forms: Directed link and undirected link. Man et al. [7] used the network embedding method to extract friend relationship information for user identity linkage. Some studies also use multiple kinds of data, such as Wang et al. [8], extracted friend relationships and represented user interests in the latent space for user identity linkage.

However, the existing user identity linkage methods have the following problems: (1) In some cross social network scenarios, user profiles, user-generated content, and structural information usually have problems of week unilateral expression ability, resulting in unsatisfactory user identity linkage performances. (2) There are potential relationships between user profiles, user-generated content, and structural information, which is often ignored when using a particular type of information alone. Therefore, we need an effective way to aggregate user profiles, user-generated content, and structural information to solve the above problems.

Lately, graph neural networks (GNN) has achieved better performances on node representation learning [9]. GAT [10] uses node self-attention mechanisms to learn graph structure representation. GEMSEC [11] is a representation learning method that combines random walk, skip-gram, and clustering. GCN [12] is a semi-supervised graph structure representation method with graph convolutional networks. Most of the existing GNN methods are applied to the same data source for data mining, and are seldom applied to user identity linkage. As a graph has a strong relationship expression ability, we can express user profiles, user-generated content, and structural information in a heterogeneous graph. We can use GNN to learn feature representation and aggregate user profiles, user-generated content, and structural information in the latent space.

In the paper, we propose a user identity linkage method based on a heterogeneous graph attention network mechanism (UIL-HGAN). To solve problems of weak data expression ability and ignored potential relationships, firstly, user profiles, user-generated content, structural information, and their features are represented as nodes in a heterogeneous graph from different social networks, respectively. We build edges between these nodes and users. Secondly, we use multiple attention layers to represent users by aggregating user profiles, user-generated content, and structural information in the latent space. Finally, we use a multi-layer perceptron to predict user identity linkage from different social networks. The main contributions of this paper are summarized as follows:

1. We propose a novel method to represent user profiles, user-generated content, structural information, and their features in a heterogeneous graph;
2. We propose a novel user identity linkage method based on a heterogeneous graph attention network mechanism called UIL-HGAN;
3. We conduct experiments on two real-world datasets to test and validate the effectiveness and advancement of UIL-HGAN.

2. Related Work

User identity linkage across social networks has excellent research significance in data mining, network security, etc. Existing methods can be divided into three categories [4]: User profile-based, user generated content-based, and structural information-based. In this section, we briefly summarize existing methods.

2.1. User Profile-Based User Identity Linkage

User profiles mainly include information entered by users in the registration process such as username, nickname, region, self-introduction, education experience, and avatar. Zafarani et al. [5] proposed MOBIUS, to extract features of patterns to human limitations, exogenous factors, and endogenous factors from a username for user identity linkage. Zhang et al. [13] calculated user profiles' similarity, such as username, language, URL, location, and avatar for the linkage of user identities. Mu et al. [14] proposed a method, ULink, based on a projection algorithm, to model user

profiles in the latent user space. Li et al. [15] proposed a machine learning method to analyze the user name pattern for user identity linkage.

2.2. User Generated Content-Based User Identity Linkage

User-generated content includes time information, spatial information, and published content. Goga et al. [16] proposed a supervised learning method, LU-Link, to extract geo-locations, timestamps, and writing styles from user-generated content for user identity linkage. Liu et al. [6] proposed Hydra to extract time information, spatial information, and others to analyze interest and writing style for the linkage of user identities. Li et al. [17] proposed U-UIM to calculate spatial similarity, time similarity, and content similarity. Riederer et al. [18] proposed a method, POIS, based on user trajectory for user identity linkage.

2.3. Structural Information-Based User Identity Linkage

Structural information can be expressed as two forms: Directed link and undirected link. Man et al. [7] proposed PALE, using the network embedding method to extract friend relationship information. Miao et al. [19] proposed a method, EUIA, for user identity linkage by comparing the embedding of nodes in low dimensional space. Wang et al. [20] proposed a semi-supervised method, APAN, which extends the idea of DeepWalk [21] to embed nodes for user identity linkage. Zhou et al. [1] proposed a semi-supervised method, FRUI, to extract features from the neighborhood-based network. Li et al. [22] analyzed the similarities of k-hop neighbors and compared the effects of several friendship-based classifiers for user identity linkage.

2.4. Multiple Type-Based User Identity Linkage

Some methods use multiple kinds of features for user identity linkage at the same time. Park et al. [23] proposed a conditional random field-based method called JLA to use user profiles and social relations for user identity linkage. Kong et al. [24] proposed a supervised learning method, MNA, to extract style features from user-generated content, combined with structural information. Nie et al. [25] proposed a method called DCIM to calculate the similarity of social relations and articles for the linkage of user identities. Wang et al. [8] proposed a method called LHNE to extract friend relationships, and user interests to represent in the latent space for user identity linkage.

2.5. Distinction with Current Works

The existing methods for user identity linkage pay attention on methods of extracting features from user profiles, user-generated content, and structural information. They mainly have the following problems: (1) User profiles, user-generated content, and structural information usually have problems of week unilateral expression ability, resulting in unsatisfactory performances of user identity linkage. (2) There are some potential relationships between user profiles, user-generated content, and structural information, ignoring these potential relationships leads to little improvement of user identity linkage performance. The distinction between this paper and the existing methods is that we focus on aggregating different types of features to solve problems of weak data expression ability and ignored potential relationships.

3. Preliminary

In this section, we define the preliminary concepts used in this paper.

A heterogeneous graph is a particular type of graph. Compared with a traditional graph, a heterogeneous graph contains many types of nodes and edges. In this paper, user profiles, user-generated content, structural information, and their features represent nodes in a heterogeneous graph from different social networks, respectively, and we build edges between these nodes and users. These node types include the user node, the user profile node, the user-generated content node,

the structural information node, the category node, the feature node, and the structural embedding node. The category nodes are subdivisions of user profiles and user-generated content, such as the username in profile. The feature nodes are unique and universal, extracted from user profiles and user-generated content. The structural embedding nodes dynamically generate after the embedding of every user-to-user edge.

Definition 1 (Heterogeneous Social Network Graph). *A heterogeneous social network graph denotes as* $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, *where* $\mathcal{V} = \{v_i | i = 1, \ldots, N\}$ *is a node set, and* $\mathcal{E} = \{e_{ij} | (i, j) = 1, \ldots, N\}$ *is an edge set. A node set* $\mathcal{V} = \{\mathcal{V}^U, \mathcal{V}^P, \mathcal{V}^G, \mathcal{V}^S, \mathcal{V}^C, \mathcal{V}^F, \mathcal{V}^E\}$ *consists of several types of nodes: The user node* $\mathcal{V}^U$, *the user profile node* $\mathcal{V}^P$, *the user-generated content node* $\mathcal{V}^G$, *the structural information node* $\mathcal{V}^S$, *the category node* $\mathcal{V}^C$, *the feature node* $\mathcal{V}^F$, *and the structural embedding node* $\mathcal{V}^E$.

The edge pattern between the user node v_i and the user node v_j is $v_i \overset{e_{ij}}{\longleftrightarrow} v_j$, where $v_i, v_j \in \mathcal{V}^U$. The edge pattern among the user node v_m and the feature node v_i is $v_i \overset{e_{ij}}{\longleftrightarrow} v_j \overset{e_{jk}}{\longleftrightarrow} v_k \overset{e_{km}}{\longleftrightarrow} v_m$, where $v_i \in \mathcal{V}^F$, $v_j \in \mathcal{V}^C$, $v_k \in \mathcal{V}^P \cup \mathcal{V}^G$, and $v_m \in \mathcal{V}^U$. The edge pattern among the user node v_k and the structural embedding node v_i is $v_i \overset{e_{ij}}{\longleftrightarrow} v_j \overset{e_{jk}}{\longleftrightarrow} v_k$, where $v_i \in \mathcal{V}^E$, $v_j \in \mathcal{V}^S$, and $v_k \in \mathcal{V}^U$. An example of a heterogeneous social network graph is shown in Figure 1, which includes three user nodes, two user profile nodes, one user-generated content node, three structural information nodes, four category nodes, five feature nodes, and six structural embedding nodes. The user-to-user edges include two types, the edge between user A and user B is the "friend" edge, and the edge between user A and user C is the "blacklist" edge. For user profiles, the "gender" category nodes linked with user A and user B connect to the "male" feature node, and the "name" category node linked with user A connects to three feature nodes. For user-generated content, the "interest" category node linked with user C connects to a feature node. For structural information, every structural embedding node dynamically generates after the embedding of each user-to-user edge. The number of generated nodes is related to the type of user-to-user edges.

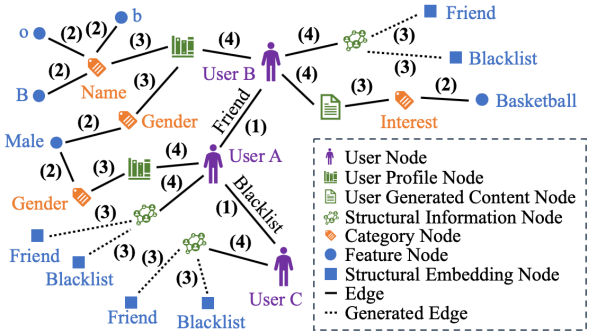

Figure 1. Example of a heterogeneous social network graph.

Without loss of generality, we focus on user identity linkage in two social networks, denoted as $\mathcal{G}_s$ and $\mathcal{G}_t$ respectively. As shown in Figure 2, for users in the two social networks, we intend to predict linkage pairs where these users belong to the same identity.

Definition 2 (User Identity Linkage). *Given two social networks* $\mathcal{G}_s$ *and* $\mathcal{G}_t$, *user identity linkage aims to predict whether a pair of entity* $v_i \in G_s$ *and* $v_j \in G_t$ *belong to the same user identity, i.e.,*

$$F(v_i, v_j) = \begin{cases} 1 & \text{if } v_i \text{ and } v_j \text{ belong to} \\ & \text{a same user identity,} \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

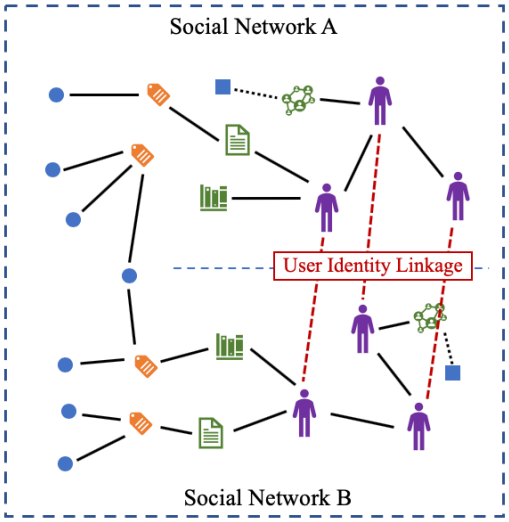

Figure 2. User identity linkage.

4. Methods

In this section, we introduce the proposed user identity linkage method based on a heterogeneous graph attention network mechanism UIL-HGAN. The overall process is shown in Figure 3. Firstly, we embed the nodes and different types of user-to-user edges to the latent space and generate the structural information nodes, such as (1) in Figures 1 and 3. Secondly, we introduce the structure of the attention layer. We use multiple attention layers to represent the user nodes by aggregating the feature nodes, the category nodes, the structural embedding nodes, the user profile nodes, the user-generated content nodes, and the structural information nodes, such as (2), (3), (4) in Figures 1 and 3. Finally, we perform a multi-layer perceptron to train the model and predict user identity linkage.

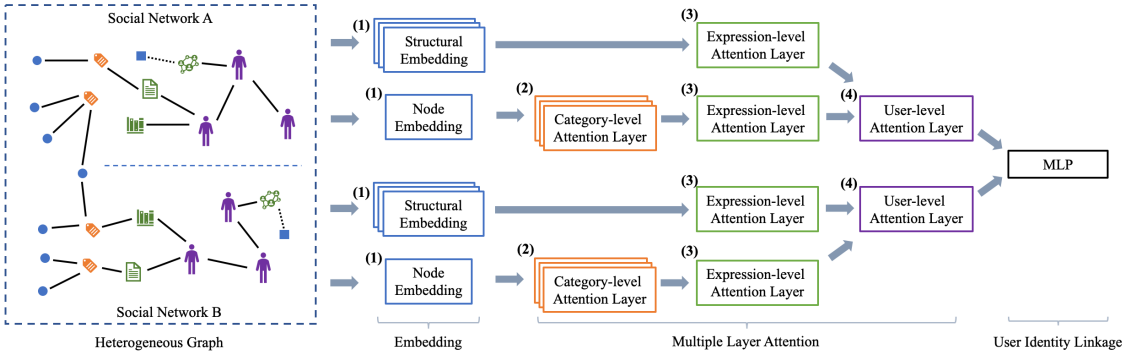

Figure 3. User identity linkage method based on a heterogeneous graph attention network mechanism (UIL-HGAN).

4.1. Embedding

The embedding process can be divided into two categories: The embedding of every user-to-user edge and the embedding of every node in the heterogeneous social network graph.

4.1.1. Embedding of Every User-to-User Edge

In the heterogeneous social network graph, the edges between users have several types. For example, in Figure 1, the edge between user A and user B is the "friend" edge, and the edge between user A and user C is the "blacklist" edge. We use the first-order proximity [26] to embed

every user-to-user edge. Given two users $v_i, v_j \in \mathcal{V}^U$ in the heterogeneous social network graph, the probability that the edge whose type is t, existing between v_i and v_j can be calculated as follows:

$$p(v_i, v_j) = \sigma(z_i^T \cdot z_j) = \frac{1}{1 + e^{-z_i^T \cdot z_j}}, \tag{2}$$

where $z_i, z_j \in \mathbb{R}^{d'}$ are d'-dimensional vectors of v_i, v_j in the latent space, and σ is the sigmoid function.

To obtain the vectors z_i, z_j in the latent space, combining with (2) and the log-likelihood function, we need to minimize the following:

$$-\sum_{(i,j)\in\mathcal{E}} \log p(v_i, v_j). \tag{3}$$

To avoid the existence of trivial solutions, the loss function of the embedding can be written as follow:

$$L_t = -\sum_{(i,j)\in\mathcal{E}} \log \sigma(z_i^T \cdot z_j) - \sum_{(i,k)\notin\mathcal{E}}^{K} \log(1 - \sigma(z_i^T \cdot z_k)), \tag{4}$$

where t is the type of edge, and K is the total number of negative sampling edges. For all edge types $t \in T$, the total loss function is written as:

$$L_{\text{SE}} = \sum_{t\in T} L_t. \tag{5}$$

Then, to facilitate the subsequent feature aggregation, we transform the embedding vector into d-dimension in the latent space, and the transformation process is as follow:

$$h = \tanh(W \cdot z + b), \tag{6}$$

where $W \in \mathbb{R}^{d' \times d}$ is the transformation matrix, $b \in \mathbb{R}^d$ is the bias vector, $h \in \mathbb{R}^d$ is the transformed vector as input to subsequent attention layers, and tanh is the hyperbolic tangent function.

Finally, according to the type of edges, the structural embedding nodes are created in the heterogeneous social network graph, whose embedding vector is h' in the latent space.

4.1.2. Embedding of Every Node

In a heterogeneous social network graph, given any node $v_i \in V$, we randomly initialize the node embedding vector $h_i \in \mathbb{R}^d$ in the latent space by the Glorot uniform initializer. In the subsequent user identity linkage tasks, the embedding vectors of nodes will be updated by every training epoch.

4.2. Multiple Attention Layers

In this subsection, we firstly introduce the structure of the attention layer. As shown in Figure 3, we use multiple attention layers that includes the category-level attention layers, the expression-level attention layers, and the user-level attention layers to represent the user nodes by aggregating the feature nodes, the category nodes, the structural embedding nodes, the user profile nodes, the user-generated content nodes, and the structural information nodes in the latent space.

4.2.1. Definition of the Attention Layer

To aggregate multiple types of nodes in the latent space, we need to obtain the neighbor information between nodes. To achieve this goal, we introduce the attention layer, which works in a graph. In the attention layer, we aggregate the 1-hop neighbor nodes by the attention mechanism. Figure 4 is an example of an attention layer, v_a, v_b, v_c, v_d are 1-hop neighborhoods of node v_i, h_a, h_b, h_c, h_d, h_i are the embedding vectors or the attention layer aggregated vectors of these nodes in the latent space, respectively. The goal of the attention layer is to aggregate h_a, h_b, h_c, h_d and h_i. Then, we mathematically explain how the attention layer works.

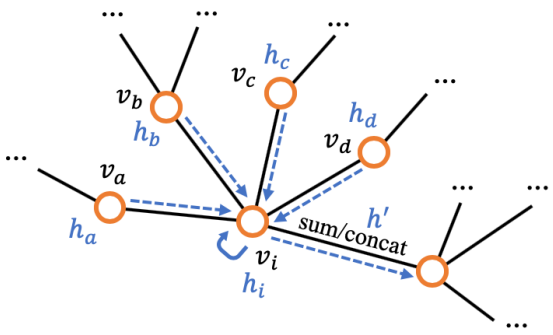

Figure 4. Attention layer.

The heterogeneous social network graph has a node v_i and its neighbor node $v_j \in N(v_i, t)$, where $N(v_i, t)$ is a set of neighbor nodes whose type is t of node v_i. h_i, h_j are the embedding vectors or the attention layer aggregated vectors of v_i, v_j in the latent space. The calculation of the attention coefficient $\phi : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ by self-attention [27] is shown as follow:

$$\phi_{v_i v_j} = \phi(h_i, h_j) = \mathrm{a}^T \cdot [h_i \| h_j], \tag{7}$$

where $\mathrm{a} \in \mathbb{R}^{2d}$ is the learnable weight vector, and $\|$ is the concatenate function. The attention coefficient $\phi(v_i v_j)$ indicates the importance between v_i and v_j. To make the different attention coefficients easier to compare, we calculate all the attention coefficients of neighbor nodes whose type is t and normalized by a softmax function as follow:

$$\alpha_{v_i v_j} = \frac{\exp\left(\phi_{v_i v_j}\right)}{\sum_{v_k \in N(v_i, t)} \exp\left(\phi_{v_i v_k}\right)}. \tag{8}$$

Finally, we aggregate the embedding vectors, or the attention layer aggregated vectors of all neighbor nodes whose type is t of node v_i:

$$h' = \tanh\Big(\sum_{v_j \in N(v_i, t)} \alpha_{v_i v_j} h_j\Big) \text{ or} \tag{9}$$

$$h' = \tanh\Big(\Big\|_{v_j \in N(v_i, t)} \alpha_{v_i v_j} h_j\Big), \tag{10}$$

where tanh is the hyperbolic tangent function, $\|$ is the concatenate function, and h' is the current attention layer aggregated vector of node v_i, which is used to input the next attention layer or a multi-layer perceptron. The calculation of h' will not update the embedding vector h_i of node v_i in the latent space. Therefore, an attention layer can be expressed as:

$$h' = \mathrm{AttLayer}(v_i, t, \Sigma) \text{ or} \tag{11}$$

$$h' = \mathrm{AttLayer}(v_i, t, \|). \tag{12}$$

4.2.2. Category-Level Attention Layer

The category-level attention layer aggregates the node embedding vectors from the feature nodes to the category node in the latent space, as shown in Figure 1 (2). The heterogeneous social network graph has a category node $v_i \in \mathcal{V}^C$ which connects the feature node $v_j \in N(v_i, \mathcal{V}^F)$. Let h_i, h_j denote

the embedding vector of node v_i, v_j in the latent space. So the category-level attention layer aggregated vector h' of node v_i is:

$$h' = \text{AttLayer}(v_i, \mathcal{V}^F, \Sigma) = \tanh\Big(\sum_{v_j \in N(v_i, \mathcal{V}^F)} \alpha_{v_i v_j} h_j\Big), \tag{13}$$

where h' is used to input the next expression-level attention layer.

4.2.3. Expression-Level Attention Layer

The expression-level attention layer aggregates the vectors from the category nodes or the structural embedding nodes to the user profile node or the user-generated content node or the structural information node in the latent space, as shown in Figure 1 (3). The heterogeneous social network graph has a node $v_i \in \mathcal{V}^P \cup \mathcal{V}^G \cup \mathcal{V}^S$ connecting the category node $v_j \in N(v_i, \mathcal{V}^C)$ or the structural embedding node $v_j \in N(v_i, \mathcal{V}^E)$. Let h_i denote the embedding vector of node v_i in the latent space. In the expression-level attention layer, vector h_j is different from the category-level attention layer. If the type of v_j is the category node, let h_j denote the attention layer aggregated vector of v_j. If the type of v_j is the structural embedding node, let h_j denote the embedding vector of v_j from (6). Therefore, the expression-level attention layer aggregated vector h' of node v_i is:

$$h' = \text{AttLayer}(v_i, \mathcal{V}^C, \Sigma) = \tanh\Big(\sum_{v_j \in N(v_i, \mathcal{V}^C)} \alpha_{v_i v_j} h_j\Big) \text{ or} \tag{14}$$

$$h' = \text{AttLayer}(v_i, \mathcal{V}^E, \Sigma) = \tanh\Big(\sum_{v_j \in N(v_i, \mathcal{V}^E)} \alpha_{v_i v_j} h_j\Big), \tag{15}$$

where h' is used to input the next user-level attention layer.

4.2.4. User-Level Attention Layer

The user-level attention layer aggregates the vectors from the user profile nodes or the user-generated content node or the structural information nodes to the user node in the latent space, as shown in (4) in Figure 1. The heterogeneous social network graph has a node $v_i \in \mathcal{V}^U$ connecting the user profile nodes or the user-generated content node or the structural information node $v_j \in N(v_i, \mathcal{V}^P \cup \mathcal{V}^G \cup \mathcal{V}^S)$. Let h_i denote the embedding vector of node v_i in the latent space. Let h_j denote the attention layer aggregated vector of node v_j in the latent space. So, the user-level attention layer aggregated vector h' of node v_i is:

$$h' = \text{AttLayer}(v_i, \mathcal{V}^P \cup \mathcal{V}^G \cup \mathcal{V}^S, \|) = \tanh\Big(\Big\|_{v_j \in N(v_i, \mathcal{V}^P \cup \mathcal{V}^G \cup \mathcal{V}^S)} \alpha_{v_i v_j} h_j\Big), \tag{16}$$

where h' is used to input the next multi-layer perceptron to predict user identity linkage, and $\|$ is the concatenate function.

In summary, we use multiple attention layers that includes the category-level attention layers, the expression-level attention layers, and the user-level attention layers to represent the user nodes by aggregating the feature nodes, the category nodes, the structural embedding nodes, the user profile nodes, the user-generated content nodes, and the structural information nodes in latent space. In the heterogeneous social network graph, only the embedding vector of the feature node and the structural embedding node is used to input the attention layer. The embedding vector of other types of nodes is used to calculate the attention coefficient.

4.3. User Identity Linkage

In this subsection, we use a multi-layer perceptron to predict user identity linkage. Give two social networks $\mathcal{G}_s, \mathcal{G}_t$ and two user nodes $v_i \in \mathcal{G}_s, v_j \in \mathcal{G}_t$. Let h_i, h_j denote the user-level attention

layer aggregated vectors of v_i, v_j, respectively. First of all, to improve the training speed and avoid the gradient disappearing, the batch normalization [28] is applied to normalize h_i, h_j respectively:

$$h'_i = \mathrm{BN}(h_i) \text{ and} \tag{17}$$

$$h'_j = \mathrm{BN}(h_j), \tag{18}$$

where BN is the batch normalization layer. The input of the multi-layer perceptron is $x = [h'_i \| h'_j]$ and $\|$ is the concatenate function. The l-th layer of the neural network is defined as:

$$y^l(x) = \mathrm{ReLU}(W^l \cdot y^{l-1}(x) + b^l), \tag{19}$$

where W^l, b^l are the l-th layer parameters, $\mathrm{ReLU}(a) = \max(0, a)$ is the linear rectification function, and $y^0(x) = x$. Finally, the 1-dimension vector $\hat{y} \in [0, 1]$ is the output by the neural network as the predicted value of user identity linkage:

$$\hat{y}_{ij} = \sigma(W \cdot y^l(x) + b), \tag{20}$$

where σ is the sigmoid function, $W \in \mathbb{R}^{d \times 1}$ is the weight matrix, d is the dimension of $y^l(x)$, and $b \in \mathbb{R}^1$ is the bias vector. Then the loss of the user identity linkage is:

$$L_{\mathrm{UIL}} = -\sum_{v_i \in \mathcal{G}_s, v_j \in \mathcal{G}_t} y_{ij} \log(\hat{y}_{ij}) + (1 - y_{ij}) \log(1 - \hat{y}_{ij}), \tag{21}$$

where y_{ij} is the ground truth of user identity linkage pair (v_i, v_j).

4.4. Overview of UIL-HGAN

In UIL-HGAN, we first embed the nodes and different types of user-to-user edges to the latent space and generate the structural information nodes. Secondly, we use multiple attention layers to represent the user nodes by aggregating the feature nodes, the category nodes, the structural embedding nodes, the user profile nodes, the user-generated content nodes, and the structural information nodes. Finally, we perform a multi-layer perceptron to train the model and predict user identity linkage. The total loss function denotes as follow:

$$L = L_{\mathrm{SE}} + L_{\mathrm{UIL}}. \tag{22}$$

The model training process is shown in Algorithm 1, and the model parameters update by using Adam optimizer [29] to minimize the loss function. The model predicting process is shown in Algorithm 2.

Algorithm 1 Model training

Input: Two heterogeneous social network graph $\mathcal{G}_s, \mathcal{G}_t$, user identity linkage pair set $S = \{(v_i, v_j, y_{ij}) | v_i \in \mathcal{G}_s, v_j \in \mathcal{G}_t\}$

Perform the embedding of every user-to-user edge in $\mathcal{G}_s \cup \mathcal{G}_t$;
Perform the embedding of every node in $\mathcal{G}_s \cup \mathcal{G}_t$;
for (v_i, v_j, y_{ij}) in S **do**
    Calculate the aggregated vector h_i of v_i by multiple attention layers in $\mathcal{G}_s$;
    Calculate the aggregated vector h_j of v_j by multiple attention layers in $\mathcal{G}_t$;
    Calculate the predicted value $\hat{y}_{ij}$ of user identity linkage by the multi-layer perceptron;
    Calculate the total loss L;
    Update all parameters by Adam optimizer;
end for

Algorithm 2 Model predicting

Input: Two heterogeneous social network graph $\mathcal{G}_s, \mathcal{G}_t$, user pair v_i, v_j where $v_i \in \mathcal{G}_s, v_j \in \mathcal{G}_t$
Output: The predicted value $\hat{y}_{ij}$ of user identity linkage.

Calculate the aggregated vector h_i of v_i by multiple attention layers in $\mathcal{G}_s$;
Calculate the aggregated vector h_j of v_j by multiple attention layers in $\mathcal{G}_t$;
Calculate the predicted value $\hat{y}_{ij}$ of user identity linkage by the multi-layer perceptron;
return $\hat{y}_{ij}$

5. Experiments

In this section, we conduct experiments on two real-world datasets to test and validate the effectiveness and advancement of UIL-HGAN. The first dataset consists of two kinds of social networks: OSCHINA and Gitee, and the second dataset consists of Facebook and Twitter.

5.1. Datasets

5.1.1. OSCHINA-Gitee

OSCHINA is the largest open-source technology community with 4 million members in China, providing a platform for developers to discover and exchange technologies. Gitee is a code hosting platform with 5 million members, providing a free private warehouse for hosting service. We used the breadth-first search algorithm to obtain 161,428 users from OSCHINA and 126,308 users from Gitee. The user information pages of OSCHINA provide Gitee links, which can be used as the ground truth for user identity linkage. Finally, we had a total of 5649 active users. We used this dataset to validate the effectiveness of UIL-HGAN.

5.1.2. Facebook-Twitter

Many methods evaluate the performances of user identity linkage on Facebook and Twitter. Facebook is a social platform where users can share pictures, links, and videos. Twitter is a microblog liked social network. We got 102,893 users from Facebook and 80,378 users from twitter. About.me is a third-party user information integration platform where users can add Facebook and Twitter links on the home page, which can be used as the ground truth for user identity linkage. Finally, we had a total of 8251 active users. We used this dataset to validate the advancement of UIL-HGAN.

5.2. Metrics for Comparison

In this paper, we use the popular evaluation metric *hit-precision* [14] to evaluate user identity linkage performance by comparing the top-k candidates of the predicted values. The *hit-precision* can be calculated as follow:

$$h(x) = \frac{k - (hit(x) - 1)}{k} \tag{23}$$

where $hit(x)$ is the position of the correct value in the top-k candidates. However, if the correct value is not in the top-k candidates, $hit(x) = k + 1$. For example, let $k = 5$, given the top-k candidate sets $\hat{Y} = \{\hat{y}_1, \hat{y}_2, \hat{y}_3, \hat{y}_4, \hat{y}_5\}$, if the correct value $\hat{y} = \hat{y}_2$, then $hit(x) = 2$ and $h(x) = \frac{k-1}{k} = 0.8$. If the correct value $\hat{y} \notin \hat{Y}$, then $hit(x) = 6$ and $h(x) = 0$. For N test users, we average the *hit-precisions* by $\frac{\sum_i^N h(x_i)}{N}$. In experiments, the *hit-precision* results express as *hit@k*.

5.3. Validation of Effectiveness

In this subsection, we use the OSCHINA-Gitee dataset to validate the effectiveness of UIL-HGAN. In UIL-HGAN, the multiple attention layers consist of the category-level attention layer, the expression-level attention layer, and the user-level attention layer. We used (Username), (URL-ID), (Username, URL-ID), and (Username, URL-ID, Follow) to carry out four groups of comparative experiments to validate the effectiveness of each layer in the multiple attention layers. Username and URL-ID are user profiles, and Follow is structural information. In the experiment, we set the dimension of features in the latent space as 32, set the ratio of positive and negative samples as 0.2, and set the training ratio to 20%, 40%, 60%, and 80%. The multi-layer perceptron has two hidden layers with the same dimension 32. The results of the experiment are shown in Table 1 and Figure 5.

Table 1. Performance of different feature groups on the OSCHINA-Gitee dataset.

Group	Training Ratio	*hit@1*	*hit@50*	*hit@100*	*hit@500*	*hit@1000*
(Username)	20%	0.5070	0.5108	0.5184	0.5623	0.6032
	40%	0.5230	0.5258	0.5328	0.5773	0.6077
	60%	0.5500	0.5520	0.5567	0.5902	0.6175
	80%	0.5620	0.5654	0.5700	0.5941	0.6220
(URL-ID)	20%	0.5480	0.5515	0.5580	0.6025	0.6361
	40%	0.5660	0.5683	0.5726	0.6095	0.6375
	60%	0.5870	0.5882	0.5911	0.6217	0.6515
	80%	0.5900	0.5910	0.5924	0.6147	0.6408
(Username, URL-ID)	20%	0.5560	0.5653	0.5795	0.6664	0.7214
	40%	0.5950	0.6054	0.6185	0.6854	0.7263
	60%	0.6330	0.6355	0.6425	0.7032	0.7483
	80%	0.6510	0.6524	0.6603	0.7212	0.7607
(Username, URL-ID, Follow)	20%	0.6230	0.6355	0.6564	0.7492	0.8106
	40%	0.6400	0.6474	0.6673	0.7658	0.8276
	60%	0.6720	0.6889	0.7087	0.8045	0.8552
	80%	0.6910	0.6999	0.7160	0.8046	0.8520

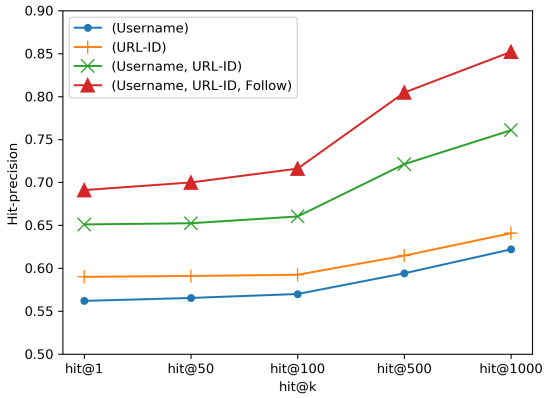

Figure 5. Performance of different feature combinations with an 80% training ratio on the OSCHINA-Gitee dataset.

In Table 1 and Figure 5, the results of (Username) and (URL-ID) validate the effectiveness of the category-level attention layer, where features of usernames and URL-IDs are aggregated. The *hit@1* precisions with an 80% training ratio of (Username) and (URL-ID) are 0.5620 and 0.5900, showing that the URL-ID feature has a better performance than the username feature in user identity linkage.

Username and URL-ID are aggregated in the expression-level attention layer. Compared with the results of (Username), (URL-ID), and (Username, URL-ID) in Figure 5, the curve of (Username, URL-ID) is above the curves of (Username) and (URL-ID). It shows that (Username, URL-ID) has a better user identity linkage performance and validates the expression-level attention layer's effectiveness.

User profiles and structural information are aggregated in the user-level attention layer. Compared with the results of (Username, URL-ID) and (Username, URL-ID, Follow) in Table 1, the *hit@k* precisions of (Username, URL-ID, Follow) have larger values than the *hit@k* precisions of (Username, URL-ID). It shows that (Username, URL-ID, Follow) has a better user identity linkage performance and validates the user-level attention layer's effectiveness.

In this subsection, the OSCHINA-Gitee dataset's results validate each layer's effectiveness in the multiple attention layer and validate the effectiveness of UIL-HGAN. To a certain extent, UIL-HGAN solves the problems of weak data expression ability and ignored potential relationships.

5.4. Validation of Advancement

In this subsection, we use the Facebook-Twitter dataset to validate the advancement of UIL-HGAN. To facilitate the comparison with the existing user identity linkage methods, we use usernames and friend relationships in the dataset. In the experiment, we compared the methods as follows:

- SVR [15]: A supervised learning method is proposed to extract the longest common string, the longest common subsequence, the Jenson Shannon distance, the editing distance, and others as features of username for user identity linkage. We use support vector regression (SVR) to calculate the *hit-precision*;
- PALE [7]: A supervised user identity linkage method of structural information, that uses a network embedding method to represent nodes in a low dimension;
- UIL-HGAN(P): An UIL-HGAN method using only usernames;
- UIL-HGAN(S): An UIL-HGAN method using only friend relationships.

For different user identity linkage methods, we set the training ratio to 20%, 40%, 60%, and 80%, respectively, and calculate *hit@1*, *hit@50*, *hit@100*, *hit@500*, and *hit@1000* separately. The experimental results are shown in Table 2. The *hit-precision* jitter is shown of different training ratios in the same method. As the *hit@1* precision directly reflects the performance of user identity linkage, to intuitively

compare different methods, *hit@1* precisions under different training ratios of different methods are shown in Figure 6.

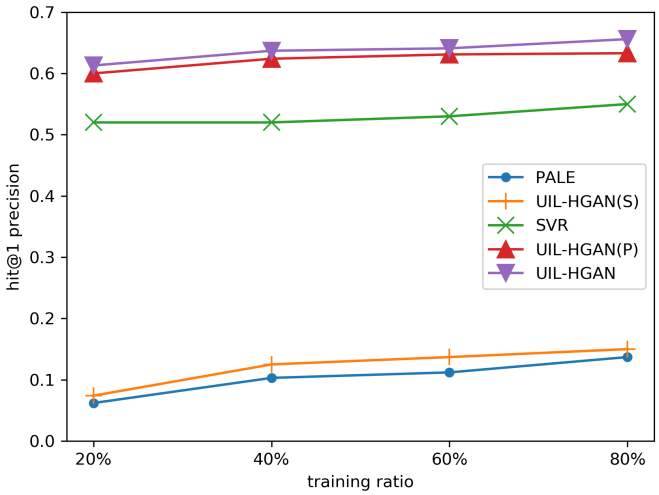

Figure 6. *hit@1* precisions of different methods on the Facebook-Twitter dataset.

Table 2. Performance of different methods on the Facebook-Twitter dataset.

Method	Training Ratio	*hit@1*	*hit@50*	*hit@100*	*hit@500*	*hit@1000*
PALE	20%	0.0620	0.0623	0.0629	0.0808	0.1188
	40%	0.1030	0.1032	0.1037	0.1157	0.1489
	60%	0.1120	0.1120	0.1120	0.1179	0.1467
	80%	0.1370	0.1370	0.1370	0.1389	0.1496
UIL-HGAN(S)	20%	0.0740	0.0740	0.0740	0.0849	0.1318
	40%	0.1250	0.1250	0.1250	0.1253	0.1473
	60%	0.1370	0.1370	0.1370	0.1370	0.1545
	80%	0.1500	0.1500	0.1500	0.1501	0.1642
SVR	20%	0.5200	0.6836	0.7108	0.7613	0.7803
	40%	0.5200	0.6872	0.7128	0.7644	0.7822
	60%	0.5300	0.6850	0.7123	0.7596	0.7726
	80%	0.5500	0.6906	0.7167	0.7607	0.7754
UIL-HGAN(P)	20%	0.6000	0.6079	0.6177	0.6708	0.7100
	40%	0.6240	0.6285	0.6388	0.6840	0.7177
	60%	0.6310	0.6363	0.6473	0.6944	0.7253
	80%	0.6330	0.6391	0.6470	0.6842	0.7155
UIL-HGAN	20%	0.6130	0.6150	0.6254	0.6933	0.7363
	40%	0.6370	0.6379	0.6463	0.6864	0.7174
	60%	0.6410	0.6420	0.6473	0.6851	0.7159
	80%	0.6560	0.6564	0.6592	0.6847	0.7105

UIL-HGAN(S) and PALE only use friendly relationships for user identity linkage. When the training ratio is 20%, the *hit@1* precision of PALE is 0.0620, while the *hit@1* precision of UIL-HGAN(S)

is 0.0740. When the training ratio is 80%, the *hit@1* precision of PALE is 0.1370, while the *hit@1* precision of UIL-HGAN(S) reached 0.1500. It shows that in the aspect of friendly relationships, our proposed method UIL-HGAN(S) has a better performance of user identity linkage than PALE.

Both UIL-HGAN(P) and SVR use only usernames for user identity linkage. In Figure 6, no matter what the training ratio was set to, the *hit@1* precision curves are all above SVR, which is more evident than UIL-HGAN(S) and PALE. It shows that in regards to usernames, our proposed method UIL-HGAN(P) had a better performance of user identity linkage than SVR.

The comparisons among the above methods have proved the advancement of our proposed method unilaterally. UIL-HGAN uses both usernames and friendly relationships in the experiments. In Figure 6, UIL-HGAN had a better performances of user identity linkage than UIL-HGAN(P) and UIL-HGAN(S), which validates the advancement of our proposed method.

In this subsection, we use the Facebook-Twitter dataset to validate the advancement of UIL-HGAN. In the experiment, we found that different features have different convergence rates. It usually takes a longer training time to achieve better performances to aggregate multiple types of features for user identity linkage.

6. Conclusions

To solve problems of weak data expression ability and ignored potential relationships, we propose a novel method to represent user profiles, user-generated content, structural information, and their features in a heterogeneous graph. We propose a novel user identity linkage method based on a heterogeneous graph attention network mechanism called UIL-HGAN. We conduct experiments on two real-world datasets to test and validate the effectiveness and advancement of UIL-HGAN.

UIL-HGAN has two limitations. First of all, the input dimension of the attention layer must be the same. To some extent, it causes inconvenience to the aggregation of different features. Secondly, we focus on the aggregations between different types of features and only use a simple method to extract and create feature nodes in the experiment. In the future, sophisticated features can be extracted as feature nodes to improve user identity linkage performance.

Author Contributions: Conceptualization, R.W.; methodology, R.W.; software, R.W.; validation, R.W., H.Z., L.W., and Z.C.; formal analysis, R.W., H.Z., L.W., Z.C., and M.G.; investigation, R.W., L.W., and Z.C.; resources, H.Z., M.G., and Y.X.; data curation, H.Z. and M.G.; writing–original draft preparation, R.W.; writing–review and editing, R.W. and H.Z.; visualization, R.W., H.Z., and Y.X.; supervision, H.Z., M.G., and Y.X.; project administration, Y.X.; funding acquisition, H.Z. and Y.X. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Key R&D Program of China under Grant 2017YFB0802300, in part by the Major Scientific and Technological Special Project of Guizhou Province under Grant 20183001, in part by the Foundation of Guizhou Provincial Key Laboratory of Public Big Data under Grant 2018BDKFJJ008 and Grant 2018BDKFJJ020, and in part by the National Statistical Scientific Research Project of China under Grant 2018LY61 and Grant 2019LY82.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhou, X.; Liang, X.; Zhang, H.; Ma, Y. Cross-platform identification of anonymous identical users in multiple social media networks. *IEEE Trans. Knowl. Data Eng.* **2015**, *28*, 411–424. [CrossRef]
2. Sang, J.; Deng, Z.; Lu, D.; Xu, C. Cross-OSN user modeling by homogeneous behavior quantification and local social regularization. *IEEE Trans. Multimed.* **2015**, *17*, 2259–2270. [CrossRef]
3. Huang, S.; Zhang, J.; Wang, L.; Hua, X.S. Social friend recommendation based on multiple network correlation. *IEEE Trans. Multimed.* **2015**, *18*, 287–299. [CrossRef]
4. Shu, K.; Wang, S.; Tang, J.; Zafarani, R.; Liu, H. User identity linkage across online social networks: A review. *ACM SIGKDD Explor. Newsl.* **2017**, *18*, 5–17. [CrossRef]
5. Zafarani, R.; Liu, H. 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*; ACM: New York, NY, USA, 2013.

6. Liu, S.; Wang, S.; Zhu, F.; Zhang, J.; Krishnan, R. Hydra: Large-scale social identity linkage via heterogeneous behavior modeling. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*; ACM: New York, NY, USA, 2014.
7. Man, T.; Shen, H.; Liu, S.; Jin, X.; Cheng, X. Predict anchor links across social networks via an embedding approach. *IJCAI* **2016**, *16*, 1823–1829.
8. 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* **2019**, *22*, 2611–2632. [CrossRef]
9. Zhang, Z.; Cui, P.; Zhu, W. Deep learning on graphs: A survey. *IEEE Trans. Knowl. Data Eng.* **2020**. [CrossRef]
10. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y. Graph attention networks. *arXiv* **2017**, arXiv:1710.10903.
11. Rozemberczki, B.; Davies, R.; Sarkar, R.; Sutton, C. Gemsec: Graph embedding with self clustering. In *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*; ACM: New York, NY, USA, 2019.
12. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
13. Zhang, H.; Kan, M.Y.; Liu, Y.; Ma, S. Online social network profile linkage. In *Asia Information Retrieval Symposium*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 197–208.
14. Mu, X.; Zhu, F.; Lim, E.P.; Xiao, J.; Wang, J.; Zhou, Z.H. User identity linkage by latent user space modelling. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*; ACM: New York, NY, USA, 2016.
15. Li, Y.; Peng, Y.; Ji, W.; Zhang, Z.; Xu, Q. User identification based on display names across online social networks. *IEEE Access* **2017**, *5*, 17342–17353. [CrossRef]
16. Goga, O.; Lei, H.; Parthasarathi, S.H.K.; Friedland, G.; Sommer, R.; Teixeira, R. Exploiting innocuous activity for correlating users across sites. In *Proceedings of the 22nd international conference on World Wide Web*; ACM: New York, NY, USA, 2013.
17. Li, Y.; Zhang, Z.; Peng, Y.; Yin, H.; Xu, Q. Matching user accounts based on user generated content across social networks. *Future Gener. Comput. Syst.* **2018**, *83*, 104–115. [CrossRef]
18. Riederer, C.; Kim, Y.; Chaintreau, A.; Korula, N.; Lattanzi, S. Linking users across domains with location data: Theory and validation. In *Proceedings of the 25th International Conference on World Wide Web*; IW3C2: Geneva, Switzerland, 2016.
19. Miao, Q.; Wang, L.; Duan, D.; Guo, X.; Li, X. Embedding Based Cross-network User Identity Association Technology. In *Proceedings of the 2019 3rd International Conference on Digital Signal Processing*; ACM: New York, NY, USA, 2019.
20. Wang, S.; Li, X.; Ye, Y.; Feng, S.; Lau, R.Y.; Huang, X.; Du, X. Anchor link prediction across attributed networks via network embedding. *Entropy* **2019**, *21*, 254. [CrossRef]
21. 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*; ACM: New York, NY, USA, 2014.
22. Li, Y.; Su, Z.; Yang, J.; Gao, C. Exploiting similarities of user friendship networks across social networks for user identification. *Inf. Sci.* **2020**, *506*, 78–98. [CrossRef]
23. Bartunov, S.; Korshunov, A.; Park, S.T.; Ryu, W.; Lee, H. Joint link-attribute user identity resolution in online social networks. In P*roceedings of the 6th International Conference on Knowledge Discovery and Data Mining, Workshop on Social Network Mining and Analysis*; ACM: New York, NY, USA, 2012.
24. 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*; ACM: New York, NY, USA, 2013.
25. Nie, Y.; Jia, Y.; Li, S.; Zhu, X.; Li, A.; Zhou, B. Identifying users across social networks based on dynamic core interests. *Neurocomputing* **2016**, *210*, 107–115. [CrossRef]
26. 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*; ACM: New York, NY, USA, 2015.

27. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*; Curran Associates: New York, NY, USA, 2017.
28. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv* **2015**, arXiv:1502.03167.
29. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.