Retrofitting Embeddings for Unsupervised User Identity Linkage

Tao Zhou[1](✉), Ee-Peng Lim[2], Roy Ka-Wei Lee[3], Feida Zhu[2], and Jiuxin Cao[4]

[1] School of Computer Science and Engineering, Southeast University, Nanjing, China
zhoutao@seu.edu.cn

[2] School of Information Systems, Singapore Management University,
Singapore, Singapore
{eplim,fdzhu}@smu.edu.sg

[3] Department of Computer Science, University of Saskatchewan,
Saskatchewan, Canada
roylee@cs.usask.ca

[4] Jiangsu Provincial Key Laboratory of Computer Networking Technology,
School of Cyber Science and Engineering, Southeast University, Nanjing, China
jx.cao@seu.edu.cn

Abstract. User Identity Linkage (UIL) is the problem of matching user identities across multiple online social networks (OSNs) which belong to the same person. The solutions to UIL problem facilitate cross-platform research on OSN users and enable many useful applications such as user profiling and recommendation. As the UIL labeled data are often lacking and costly to obtain, learning user embeddings for matching user identities using an unsupervised approach is therefore highly desired. In this paper, we propose a novel unsupervised UIL framework for enhancing existing user embedding-based UIL methods. Our proposed framework incorporates two key ideas, *user-discriminative features* and *retrofitting embedding*. The user-discriminative features enable us to differentiate a specific user identity from other users in its OSN. From the user-discriminative features, we derive pairs of similar user identities across OSNs for retrofitting the base user embeddings of existing UIL methods. Through extensive experiments on three real-world OSN datasets, we show that our framework can leverage user-discriminative features to improve the accuracy of different user embedding-based UIL methods significantly. The quantum of improvement can also be surprisingly good even for existing UIL methods with very poor matching accuracy.

Keywords: User identity linkage · Retrofitting embedding · Discriminative features

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-47426-3_30) contains supplementary material, which is available to authorized users.

H. W. Lauw et al. (Eds.): PAKDD 2020, LNAI 12084, pp. 385–397, 2020.
https://doi.org/10.1007/978-3-030-47426-3_30

1 Introduction

With rapid development of online social networks (OSNs), the number of OSN platforms increases quickly serving different user needs. The multiplicity of OSNs motivates the problem of User Identity Linkage (UIL), which aims to match user accounts from different OSN platforms belonging to the same persons. UIL addresses the issue of fragmented user information across platforms, which is important to cross-platform user profiling, recommendation applications and research on social networks including information diffusion, community analysis, and influential user modeling.

1.1 Unsupervised User Identity Linkage

We denote an OSN as $G = (U, E)$, where U represents the set of user identities and $E \subseteq U \times U$ represents the set of links between user identities. Each user identity $u_i \in U$ is associated with some attributes, e.g., *name*, *content*, etc.

Given two OSNs G_s and G_t as the source and target platforms respectively, the UIL task is to find for each user identity u_s from U_s a user identity u_t from U_t such that u_s and u_t belong to the same real person. While the problem is defined for a two-platform setting, it can be easily extended to more platforms.

UIL methods can be classified into *supervised*, *semi-supervised* and *unsupervised* approaches. Most of the existing UIL methods adopt the supervised and semi-supervised approaches [19]. These approaches require ground truth labeled user identity pairs for training while the collecting of labeled data suffers from many problems. The unsupervised approach to UIL can avoid the issues from labeled data. It, however, has another set of challenges: 1) Unsupervised UIL method has to cope with multiple attributes with heterogeneity domains, preferably in a unified manner; 2) Discriminative cross-platform attribute similarities are needed to compare the attribute values; 3) The attribute importance need to be incorporated without pre-labeling.

1.2 Objectives and Proposed Framework

Our main research objective is to create unsupervised UIL methods that can cope with the above challenges. With recent advances in embedding techniques, user embeddings techniques have been shown to be effective in solving the UIL problem in the unsupervised approach [21]. The user embedding techniques essentially map every user identity (from any OSN) to a common embedding space. User identities with similar attribute values are expected to be mapped to similar locations. Hence, user embeddings can effectively address the first challenge.

To address the remaining two challenges, we propose a general framework for unsupervised UIL using two main ideas, namely: (a) **user-discriminative features**, and (b) **retrofitting embeddings**. User-discriminative features are ones that are indicative of specific user identities in an OSN. Retrofitting embeddings is a technique, used largely in word embeddings, to modify an existing *base embeddings* of words to incorporate some synonym word-pair knowledge [3].

To the best of our knowledge, this paper is the first that introduce retrofitting to improve user embedding-based UIL techniques.

Our framework is novel in that it can accommodate any unsupervised UIL method as user embeddings. The framework then improves the user embeddings of the existing method using user identity pairs obtained by pairing user identities that are similar based on user-discriminative features.

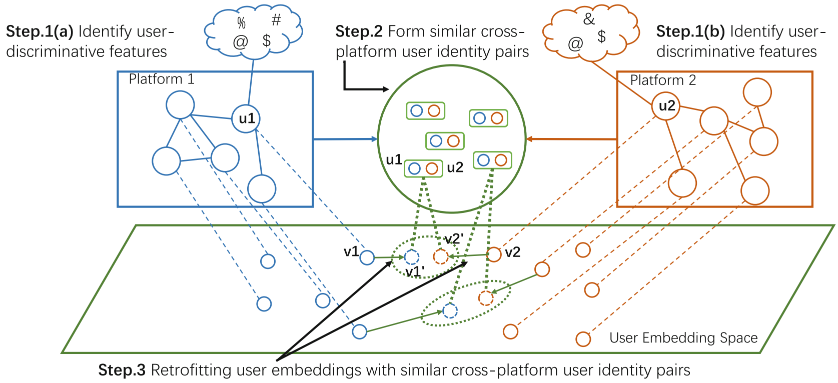
Step.1(a) Identify user-discriminative features
%
#
@
$
Step.2 Form similar cross-platform user identity pairs
&
@
$
Step.1(b) Identify user-discriminative features
Platform 1
u1
Platform 2
u2
u1
u2
v1
v2'
v2
v1'
User Embedding Space
Step.3 Retrofitting user embeddings with similar cross-platform user identity pairs

Fig. 1. Proposed User Identity Linkage Framework.

Overview of Proposed Framework. Fig. 1 depicts our proposed UIL framework for two OSN platforms, Platforms 1 and 2. The choice of two platforms is to keep the description simple, as the framework can be easily generalized to handle more OSN platforms. The framework takes an existing user embedding as input, which we call the **base embeddings**. In the figure, user identities $u1$ and $u2$ are from different OSNs, and they are assigned with embeddings vectors $v1$ and $v2$ respectively in the base embedding space. The other user identities are similarly mapped into the same user embeddings space.

In Step 1, the framework identifies user-discriminative features for each user identity in Platform 1, and does the same for Platform 2.

In Step 2, we form a set of cross-platform similar user identity pairs by pairing user identities with some overlapping user-discriminative features.

In Step 3, a set of similar cross-platform user identity pairs are used to retrofit the base user embeddings for final user identity linkage. As only qualified part of user pairs are considered for retrofitting, this process would be highly efficient.

Research Contribution. We summarize our key contributions as follows:

- We propose user-discriminative features to overcome the issues of multiple attributes of heterogeneous domains in different OSN platforms. We introduce a parameter to incorporate the importance of attribute in UIL.
- We propose an unsupervised UIL algorithm based on retrofitting embeddings which take advantage of both base user embeddings and similar user-identity

pairs. To the best of our knowledge, this is the first time retrofitting is used to achieve higher UIL accuracy for different base user embeddings. Moreover, retrofitting is highly efficient compared to the base user embedding learning.
- We conduct extensive experiments on three real-world OSN datasets with many different settings. The results show the effectiveness of our methods.

2 Related Work

2.1 Supervised and Semi-supervised Approaches

There are many UIL methods adopting the supervised approach [5,6,10–12,14–16,22]. They can be broadly classified into those using *classification techniques* and others using *embedding techniques*. The former typically extract features from user attributes (e.g., user name, profile description, content, and network) to train a classifier for predicting pairs of user identities to belong to the same users or not. For example, Zafarani et al. [22] proposed MOBIUS, a UIL method which utilizes username features and a Naive Bayes classifier.

There are few recent works using the user embedding techniques for supervised UIL. Man et al. [10] proposed PALE, a supervised embedding based UIL method that utilizes network features. PALE employs network embeddings incorporating known pairs of matching user identities from different OSNs as anchor links. Mu et al. proposed another supervised method called ULink [12] to map known pairs of matching user identities to a common latent space.

Semi-supervised approach considers both labeled and unlabeled pairs of matching user identities in model learning [1,2,8,13,18,20,23,24]. HYDRA is a semi-supervised framework which models user behaviors and structure consistency [8]. COSNET is a semi-supervised method which utilizes local and global consistency across multiple networks [23]. Unlike the above methods, our proposed model adopts an unsupervised embedding approach to address the UIL problem.

2.2 Unsupervised Approach

There are relatively fewer works on unsupervised UIL [4,7,17,21]. Gao et al. [4] proposed CNL, an unsupervised method, to utilize multiple attributes to perform UIL in an incremental manner. Factoid Embedding [21] is the state-of-art unsupervised approach which utilizes multiple user attributes to learn a user embeddings for UIL. The method first constructs factoids from user attributes and learns user identity latent representation by embedding the factoids. However, the user embedding is learned based on only local information of OSNs. Cross platforms features have been ignored. The UIL results, therefore, could be poor if the local user attribute information is noisy. Our proposed framework addresses this limitation by introducing user-discriminative features. In this paper, we will demonstrate how Factoid Embeddings can be used as input base embeddings so as to achieve improved UIL results.

3 Base User Embeddings and User-Discriminative Features

3.1 Base User Embeddings

Our proposed framework supports different types of base user embeddings. While these embeddings are input to our framework, we would like to introduce two important embeddings techniques: one for matching user identities based on a single user attribute, and another for matching based on multiple user attributes.

Single Attribute-Based User Embeddings. Here we use *username* attribute to illustrate the single attribute-based user embeddings, also known as **name embeddings**. Note that the process could be applied to any other attribute which has similarity measure for two users. We define username similarity using cosine similarity on TF-IDF vectors of n-gram representation of usernames. We first collect all n-grams ($n \in [2, 5]$ in this work) from username of user identities from **all** the OSN platforms. Next, we construct the n-gram TF-IDF vector of every user in $U_s \cup U_t$. Then we define $sim_{username}(u_i, u_j)$ as the cosine similarity between the TF-IDF vectors of users u_i and u_j denoted by w_i and w_j respectively. The username embeddings $\mathbf{v}$ are learnt by minimizing:

$$O_{username} = \sum_{u_i, u_j \in U_s \cup U_t} \left(\mathbf{v}_i^\top \mathbf{v}_j - cosine(w_i, w_j)\right)^2 \tag{1}$$

Factoid Embeddings. It is an embedding UIL method making use of multiple user attributes [21]. Details are in Section A.2 of Supplementary Material.

3.2 User-Discriminative Features

User-discriminative features are ones that help distinguishing a user identity from others in the same OSN. From the discriminative features of user identities, we derive the **cross-platform similar user identity pairs**. Each cross-platform similar user identity pair (u_i, u_j) is assigned a similarity score s_{ij}. The larger the s_{ij}, the higher the likelihood that the u_i and u_j from different OSNs belong to the same user.

As a user identity can have multiple attributes, we derive different types of user-discriminative features and the associated cross-platform similarity score s_{ij}'s as follows. One can derive for other user attributes in a similar manner.

User-Discriminative Name Features. We use n-grams in username to generate discriminative name feature. For each user identity u_i in OSN platform G_s, we collect n-grams ($n \in [2, 5]$) in its username as NG_i^s. Let NG_s be the set of all n-grams of platform $G_s = (U_s, E_s)$, i.e., $\cup_{u_i \in U_s} NG_i$. The set of user-discriminative n-grams is then defined by $DN_s = \{ng | ng \in NG_s, |\{u_i | ng \in NG_i^s\}| < t_n\}$ where t_n is a pre-defined threshold to keep only the ngrams that are not popular among user identities. In a similar way, we define the user-discriminative n-grams for the target OSN as DN_t.

Given user u_i from G_s and user u_j from G_t, we finally define the similarity score s_{ij} by Jaccard Similarity, i.e.:

$$s_{ij} = \frac{DN_i^s \cap DN_j^t}{DN_i^s \cup DN_j^t} \tag{2}$$

where $DN_i^s = NG_i^s \cap DN_s$ and $DN_j^t = NG_j^t \cap DN_t$.

Note that s_{ij} is normalized to $[0, 1]$ to avoid the effect of scale difference.

User-Discriminative Content Features. Users may generate different types of content such as text and images. In this paper, we consider textual content attribute only. The calculation is similar with user-discriminative name features except we exchange n-grams with words in user content and use t_c as the threshold for selecting content features not popular among users rather than t_n.

User-Discriminative Network Features. We denote the neighbor set of a user identity u in OSN platform G_s as $N^s(u)$. If the degree of u', a neighbor of u, is large, u' would be less important for identifying u because many user identities have u' as their neighbor. We thus use the degree of user identity to determine user-discriminative neighbors. For $u_i \in U_s$, we define the user-discriminative neighbors of u_i as $DB_i^s = \{u' | u' \in N^s(u_i), degree^s(u') < t_d\}$. t_d is a threshold to determine neighbors who do not have many social connections. Similarly, the user-discriminative neighbors of u_j in OSN G_s, DB_j^t is defined.

Unlike the earlier attributes, it is not possible to expect overlapping neighbors between two OSN platforms. We adopt some base user embeddings to determine pairs of similar discriminative neighbors across platforms. The *similar discriminative neighbor pairs*, denoted as DP_{ij}, according to the base embeddings.

$DP_{ij} = \{(u_{i'}, u_{j'}) | u_i' \in DB_i^s, u_j' \in DB_j^t, (1 - cos(\hat{\mathbf{v}}_{i'}, \hat{\mathbf{v}_{j'}})) < t_s\}$ where t_s is the dissimilarity threshold and $\hat{\mathbf{v}_\mathbf{k}}$ is u_k's base embedding,

Intuitively, the number of unique user identity pairs in DP_{ij} reflects the cross-platform identity similarity between u_i and u_j. Hence, s_{ij} is defined as:

$$s_{ij} = |\{(u_{i'}, u_{j'}) \in DP_{ij}\}| \tag{3}$$

4 Retrofitting Embedding for UIL

4.1 Retrofitting Embedding

The intuition of our method is to retrofit by pushing cross-platform similar user identity pairs closer in embedding space while keeping the other base user embedding vectors unchanged as much as possible. For each cross-platform user identity pair (u_i, u_j), we would retrofit the affected user embedding vectors according to the cross-platform similarity score s_{ij}. The larger s_{ij}, the closer the retrofitted embedding vectors of u_i and u_j should be.

For a user identity u_i, let $\hat{\mathbf{v}}_i$ denote the base embedding vector of u_i generated using base user embeddings. We use $\mathbf{v}_i$ to denote the retrofitted embedding vector of u_i, which needs to be learned. Let P be the set of cross-platform similar user identity pairs with s_{ij} scores, i.e. $P = \{(u_i, u_j)\} | u_i \in U_s, u_j \in U_t, s_{ij} > 0\}$.

We learn the retrofitted embedding vector $\mathbf{v}$ for all the $u \in U_s \cup U_t$ by minimizing the following objective function:

$$O = \sum_{u_i \in U_s \cup U_t} \left(\varphi\left(\mathbf{v}_i, \hat{\mathbf{v}}_i\right) + \alpha \sum_{(u_i,u_j) \in P} s_{ij} * \varphi\left(\mathbf{v}_i, \mathbf{v}_j\right) \right) \tag{4}$$

where $\varphi\left(\mathbf{a}, \mathbf{b}\right)$ is the cosine distance between vectors $\mathbf{a}$ and $\mathbf{b}$, i.e. $\varphi\left(\mathbf{a}, \mathbf{b}\right) = 1 - cos\left(\mathbf{a}, \mathbf{b}\right)$. α $(0 \leq \alpha \leq 1)$ is the weight to adjust the degree of retrofitting.

4.2 Variants of Retrofitting Embeddings

Stepwise Approach. As we want to use multiple discriminative features to retrofit the base user embedding, we adopt a stepwise approach to retrofit the user embedding iteratively, which is to regard the retrofitted embeddings as new base embeddings when another discriminative feature is applied.

Hierarchical Approach to Generate User-Discriminative Features. To comprehensively capture the similar user identities, we introduce a hierarchical approach to generate the user-discriminative features. We basically select a few thresholds t's, and derive different sets of user-discriminative features based on the thresholds. Each cross-platform user identity pair will then be assigned a set of scores s_{ij}'s, one for each set of user-discriminative features (e.g., name).

4.3 Optimization

With cosine distance, we cannot apply the traditional approach in retrofitting to minimize the objective function. Instead, we use Stochastic Gradient Descent (SGD) for optimization. To apply SGD, we rewrite the objective function in Eq. 4 as following by moving the position of summation:

$$O = \sum_{\{u_i,u_j\} \in P} \left(\varphi\left(\mathbf{v}_i, \hat{\mathbf{v}}_i\right) + \varphi\left(\mathbf{v}_j, \hat{\mathbf{v}}_j\right) + \beta * s_{ij} * \varphi\left(\mathbf{v}_i, \mathbf{v}_j\right) \right) \tag{5}$$

where β is the weight to adjust the degree of retrofitting.

In each iteration, we update the embedding $\mathbf{v}_i$ and $\mathbf{v}_j$ by the following rule:

$$\mathbf{v}_i \leftarrow \mathbf{v}_i - \gamma \frac{\partial O}{\partial \mathbf{v}_i} \qquad \mathbf{v}_j \leftarrow \mathbf{v}_j - \gamma \frac{\partial O}{\partial \mathbf{v}_j} \tag{6}$$

where γ is the learning rate. The detailed optimization is available in Section C of Supplementary Material.

We summarize our retrofitting embedding in Algorithm 1 which excludes hierarchical user-discriminative feature for clarity. Each of the hierarchical features would need one step of retrofitting similar as lines 2–17. The retrofitting algorithm is efficient as only the user pairs with positive score will be used.

Finally, once we have learned the retrofitted embedding vector $\mathbf{v}$ for all the $u \in U_s \cup U_t$, we will compute the cosine similarity between two user's embedding vectors as linkage score for each user pair across platforms. The user pairs with larger scores are more likely to belong to the same underlying natural person.

4.4 Selecting Parameter $\boldsymbol{\beta}$

In the optimization objective function, weight β is an important parameter to control the learning of retrofitting embedding. Smaller β indicates the retrofitted embedding preserves more of the base user embedding, while a larger β will give more weight to the user-discriminative features to change the base embedding. Therefore, β should be set larger when the base user embedding has not shown strong ability in performing UIL, and the user-discriminative features should be given more weight to improve UIL. The choice of β for different user-discriminative features would also balance the importance of multiple attributes.

5 Experiments and Results

5.1 Dataset Preparation

We evaluate our proposed framework using three real-world datasets, namely Instagram-Twitter (*IG-TW*), Foursquare-Twitter (*FQ-TW*), and *IG-TW content* datasets. These are social network datasets that are significantly larger than other social networks for UIL research involving. We start from a set of Singapore-based Twitter users and retrieve users who declared their Instagram or Foursquare accounts to construct the multi-platform datasets. From the *IG-TW* dataset, we extracted user identities with post content only into the *IG-TW content* dataset. The statistics of all the datasets are summarized in Table 1.

5.2 Baselines and Retrofitting Embeddings

We compare methods based on the proposed framework with several other unsupervised baseline methods. The baseline methods include:

- **Name embeddings (NE)**: This represents a single attribute-based user embeddings UIL method (see Sect. 3.1).
- **TF-IDF**: User identity is represented by a TF-IDF vector of content words.
- **Content embedding (CE)**: This represents a user identity by content embedding defined as the average of word vectors (obtained through the NLP tool Spacy) of words found in the content attribute [9].

Algorithm 1. Retrofitting Embedding For UIL

```
Input:
    Source platform user set $U_s$, target platform user set $U_t$;
    Base user embedding vectors $\hat{\mathbf{v}}_i$ for each $u_i \in U_s \cup U_t$;
    Attribute set A and user-dicriminative features for each attribute $a_k, k \in [1, |A|]$;
Output:
    Retrofitted embedding vector $\mathbf{v}_i$ for each $u_i \in U_s \cup U_t$;
for $k \in [1, |A|]$ do
    # prepare scores of shared discriminative features #
    calculate cross-platform similarity scores of $s_{ij}^k$ for each user pair $(u_i, u_j)$ $(u_i \in U_s,\ u_i \in U_t)$;
    create pair set $P^k = \{(u_i, u_j) | u_i \in U_s, u_i \in U_t, s_{ij}^k > 0\}$;
    # retrofitting embedding #
    for each $u_i \in U_s \cup U_t$ do
        initialize retrofitted embedding vectors $\mathbf{v}_i$ as $\hat{\mathbf{v}}_i$;
    end for
    repeat
        for each $\{u_i, u_j\} \in P^k$ do
            update $\mathbf{v}_i$ as $\mathbf{v}_i \leftarrow \mathbf{v}_i - \gamma \frac{\partial O}{\partial \mathbf{v}_i}$ ;
            update $\mathbf{v}_j$ as $\mathbf{v}_j \leftarrow \mathbf{v}_j - \gamma \frac{\partial O}{\partial \mathbf{v}_j}$ ;
        end for
    until convergence or reach maximum number of iterations;
    for each $u_i \in U_s \cup U_t$ do
        assign $\hat{\mathbf{v}}_i = \mathbf{v}_i$ unless it is the last attribute;
    end for
end for
return $\{\mathbf{v}_i | u_i \in U_s \cup U_t\}$
```

Table 1. Dataset description (uname: username, sname: screen name, Twitter as target)

Dataset	IG-TW		FQ-TW		IG-TW (content)	
Platform	Instagram	Twitter	Foursquare	Twitter	Instgram	Twitter
#Users	12,109	21,034	17,294	19,796	800	800
#Links	163,403	170,675	262,330	319,635	4,189	3,155
Avail Info	uname, sname, network		sname, network		user post, network	
# GT pairs	1,228		3,482		800	

- **Weighted content embedding (CE_weighted)**: This is similar to content embedding except that the user identity is represented by a weighted average of word vectors of words found in the content attribute. The weight of a word is defined by its TF-IDF value in the content.
- **Factoid Embedding (FE)**: This is the state-of-art unsupervised method [21]. In our experiment, FE makes use of all attributes in each dataset (FE for

IG-TW content dataset is denoted as $\mathbf{FE}_c$ for differentiation). Name embedding and content embedding are used as the attribute embeddings in FE when applying the name and content attributes respectively.

We evaluate different RE's using our proposed framework with different baselines as their base user embeddings. We use RE_p^q to denote our proposed method with q as base user embeddings and p as the user-discriminative feature(s) used for retrofitting. These proposed methods are $\mathbf{RE}_n^{NE}$, $\mathbf{RE}_{nb}^{NE}$, $\mathbf{RE}_c^{CE}$, $\mathbf{RE}_n^{FE}$, $\mathbf{RE}_{nb}^{FE}$, $\mathbf{RE}_c^{FE_c}$ and $\mathbf{RE}_{cb}^{FE_c}$, where n, b and c denote user-discriminative features for name, network and content respectively. Note that the order of applying user-discriminative features depends on the base embedding. For the sake of showing the usefulness of user-discriminative features, we start retrofitting the more discriminative features before the less discriminative ones.

5.3 Experiment Configuration

When generating user-discriminative features, we need to configure the threshold for each attribute as defined in Sect. 3. In the following experiments, t_n for name attribute is set to 5. For the content attribute, we adopt a hierarchical approach to generate user-discriminative features using multiple thresholds. The details will be elaborated in subsequent sections. Threshold t_d is set to 20 for all the datasets, and t_s in network attribute is set to be 0.4, 0.2 and 0.15 when FE, NE, and CE are used as the base user embedding respectively. The choice of t_s is dependent on the similarity distribution of user pairs cross platforms.

In retrofitting embedding, the parameter β needs to be configured. We set β to 1 when name attribute is used for retrofitting (a special case will be mentioned in later experiment sections). When the content attribute is employed, β is varied for the different settings in a hierarchical approach used to generate the user-discriminative features. For network attribute, β is set to be 15, 10 and 4 respectively for *IG-TW*, *FQ-TW* and *IG-TW content* datasets. The maximum number of iterations for the optimization is set to be 30,000. It is also interesting to note that our optimization process is fast as we only consider pairs of users who have non-zero similarity scores based on their user-discriminative features.

5.4 Experiment Results and Analysis

All the methods are required to rank the ground truth matching identity as high as possible based on the linkage score (cosine similarity of embeddings). We use **HitRate@K** and **Mean Reciprocal Rank(MRR)** to evaluate the ranking.

Experiments with Name and Network Attributes. *IG-TW* and *FQ-TW* datasets both offer username and network attributes. The baselines of **NE**, **FE**, and **RE**s which utilized both username and network attributes are included for comparison. The results of experiments are shown in Tables 2 and 3 respectively.

For *IG-TW* dataset, $\mathbf{RE}_n^{NE}$ slightly outperforms **NE**, indicating that the user-discriminative name features can improve the UIL accuracy even when

Table 2. Results in IG-TW dataset

	H@1	H@3	H@5	H@10	MRR
NE	0.8314	0.8648	0.8787	0.8974	0.8538
RE$_n^{NE}$	0.8404	0.8689	0.8811	0.8982	0.8603
RE$_{nb}^{NE}$	0.8893	0.9088	0.9178	0.9283	**0.9023**
FE	0.8265	0.8697	0.8844	0.9088	0.8539
RE$_n^{FE}$	0.8436	0.8762	0.8909	0.9080	0.8646
RE$_{nb}^{FE}$	0.9153	0.9349	0.9430	0.9495	**0.9277**

Table 3. Results in FQ-TW dataset

	H@1	H@3	H@5	H@10	MRR
NE	0.5827	0.6789	0.7128	0.7550	0.6430
RE$_n^{NE}$	0.5827	0.6781	0.7128	0.7550	0.6430
RE$_{nb}^{NE}$	0.6163	0.7074	0.7361	0.7725	**0.6716**
FE	0.5761	0.6786	0.7134	0.7588	0.6402
RE$_n^{FE}$	0.5796	0.6789	0.7128	0.7599	0.6419
RE$_{nb}^{FE}$	0.6551	0.7453	0.7769	0.8139	**0.7108**

NE has already made good use of username attribute in base user embeddings. When the user-discriminative network features are applied, **RE**$_{nb}^{NE}$ achieves even more improvement over **RE**$_n^{NE}$ and **NE**. Both **RE**$_{nb}^{NE}$ and **FE** utilize name and network attributes. **RE**$_{nb}^{NE}$ significantly outperforms **FE**, demonstrating that retrofitting embedding can effectively use cross-platform similar user identities based on different user-discriminative features. The user-discriminative network features can improve the results more significantly because they are under-explored in the base user embedding. More results are in Supplementary D.4.

For *FQ-TW* dataset, the performances of **RE**$_n^{NE}$ and **NE** are similar. A possible reason is that *FQ-TW* dataset only contains screen name, and has less useful information that can be used for retrofitting. β has been set to a relatively lower value (0.08) to avoid introducing noise in this specific case. Even though it is difficult to improve using user-discriminative name feature, the retrofitting could still be controlled to retain the base user embedding performance.

On the whole, our proposed methods using the retrofitted embeddings have outperformed the state-of-art embedding based methods. **RE**$_{nb}^{FE}$, which combines **FE** with user-discriminative name and network features, obtains the best performance in linking user identities across multiple platforms.

Experiments with Content and Network Attributes. *IG-TW content* dataset contains both content and network attribute information. Thus, the experiment on this dataset involves the baselines **CE**, **FE**$_c$, and **RE**s using user-discriminative content and network features. The results of the experiment on *IG-TW content* dataset are shown in Table 4. The first four methods make use of content-only information. Our method **RE**$_c^{CE}$, with **CE** as the base user embedding, has significantly outperformed the baselines.

The **FE**$_c$ in Table 4 uses **CE** as its attribute embedding. We observe that **FE**$_c$ has obtained a significant improvement in performance over **CE** by introducing the network information. However, it is interesting to note that **RE**$_c^{CE}$ outperforms **FE**$_c$, and **RE**$_c^{FE_c}$ has even further improved the performance with the use of better base embedding (i.e., **FE**$_c$).

RE$_c^{FE_c}$**_h1** and **RE**$_c^{FE_c}$**_h2** are retrofitting embedding methods that utilize hierarchical approach introduced in Sect. 4.2 to generate the user-discriminative content features. **RE**$_c^{FE_c}$ uses user-discriminative content features with $t_c = 2$,

Table 4. Results in Instagram-Twitter content dataset

Method	H@1	H@2	H@3	H@4	H@5	H@10	H@30	MRR
TF-IDF	0.1488	0.1675	0.1775	0.1863	0.2000	0.2375	0.3175	0.1805
CE	0.0563	0.0675	0.0825	0.0963	0.1013	0.1425	0.2375	0.0875
CE_weighted	0.0463	0.0625	0.0725	0.0813	0.0825	0.1125	0.1863	0.0732
RE_c^{CE}	0.6238	0.6438	0.6525	0.6550	0.6563	0.6725	0.7000	**0.6428**
FE_c	0.1788	0.2125	0.2275	0.2425	0.2613	0.3313	0.4625	0.2297
$\mathrm{RE}_c^{FE_c}$	0.7113	0.7263	0.7350	0.7388	0.7413	0.7488	0.7638	0.7261
$\mathrm{RE}_c^{FE_c}$_h1	0.7175	0.7463	0.7588	0.7650	0.7675	0.7750	0.7950	0.7413
$\mathrm{RE}_c^{FE_c}$_h2	0.7238	0.7500	0.7638	0.7688	0.7738	0.7813	0.7913	0.7465
$\mathrm{RE}_{cb}^{FE_c}$	0.7413	0.7713	0.7800	0.7850	0.7888	0.7938	0.8113	**0.7638**

while $\mathbf{RE}_c^{FE_c}$bf _h1 uses features with $t_c = 2, 5$ and $\mathbf{RE}_c^{FE_c}$**_h2** uses features with $t_c = 2, 5, 10$. From the results, we can see the employment of the hierarchical approach to generate user-discriminative content features improve performance. Finally, $\mathbf{RE}_{cb}^{FE_c}$ outperforms the remaining baselines by incorporating the user-discriminative content and network features.

6 Conclusion

In this paper, we propose a novel unsupervised user identity linkage (UIL) framework for enhancing existing UIL methods based on user embeddings techniques. Our proposed framework incorporates two key ideas, *user-discriminative features* and *retrofitting embeddings*. Our framework applies the user-discriminative features to derive pairs of cross-platform similar user identities for retrofitting the base user embeddings. Through extensive experiments on three real-world OSN datasets, we show that our proposed framework can leverage user-discriminative features to effectively improve the accuracy of different base user embeddings. For future work, we will conduct a more in-depth study on the parameters used in our framework, and will design methods to optimize them automatically.

Acknowledgements. This research was supported by the National Research Foundation, Prime Minister's Office, Singapore under its International Research Centres in Singapore Funding Initiative. This work is also supported by National Natural Science Foundation of China under Grants No.61772133, No.61972087, National Social Science Foundation of China under Grants No. 19@ZH014, Jiangsu Provincial Key Project under Grants No.BE2018706, Natural Science Foundation of Jiangsu province under Grants No.SBK2019022870, Jiangsu Provincial Key Laboratory of Computer Networking Technology, Jiangsu Provincial Key Laboratory of Network and Information Security under Grants No. BM2003201, Key Laboratory of Computer Network and Information Integration of Ministry of Education of China under Grants No. 93K-9, and China Scholarship Council.

References

1. Bennacer, N., Jipmo, C.N., Penta, A., Quercini, G.: Matching user profiles across social networks. In: CAiSE (2014)
2. Buccafurri, F., Lax, G., Nocera, A., Ursino, D.: Discovering links among social networks. In: ECML/PKDD (2012)
3. Faruqui, M., Dodge, J., Jauhar, S.K., Dyer, C., Hovy, E.H., Smith, N.A.: Retrofitting word vectors to semantic lexicons. In: NAACL (2015)
4. Gao, M., Lim, E.P., Lo, D., Zhu, F., Prasetyo, P.K., Zhou, A.: CNL: collective network linkage across heterogeneous social platforms. In: ICDM (2015)
5. Goga, O., Lei, H., Parthasarathi, S.H.K., Friedland, G., Sommer, R., Teixeira, R.: Exploiting innocuous activity for correlating users across sites. In: WWW (2013)
6. Iofciu, T., Fankhauser, P., Abel, F., Bischoff, K.: Identifying users across social tagging systems. In: ICWSM (2011)
7. Liu, J., Zhang, F., Song, X., Song, Y.I., Lin, C.Y., Hon, H.W.: What's in a name?: an unsupervised approach to link users across communities. In: WSDM (2013)
8. Liu, S., Wang, S., Zhu, F., Zhang, J., Krishnan, R.: Hydra: Large-scale social identity linkage via heterogeneous behavior modeling. In: SIGMOD (2014)
9. Liu, Y., Liu, Z., Chua, T.S., Sun, M.: Topical word embeddings. In: AAAI. pp. 2418–2424 (2015)
10. Man, T., Shen, H., Liu, S., Jin, X., Cheng, X.: Predict anchor links across social networks via an embedding approach. In: IJCAI (2016)
11. Mu, X., Xie, W., Lee, R.K., Zhu, F., Lim, E.: Ad-link: an adaptive approach for user identity linkage. In: ICBK, pp. 183–190 (2019)
12. Mu, X., Zhu, F., Lim, E.P., Xiao, J., Wang, J., Zhou, Z.H.: User identity linkage by latent user space modelling. In: KDD (2016)
13. Narayanan, A., Shmatikov, V.: De-anonymizing social networks. In: IEEE Symposium on Security and Privacy (2009)
14. Nie, Y., Jia, Y., Li, S., Zhu, X., Li, A., Zhou, B.: Identifying users across social networks based on dynamic core interests. Neurocomputing **210**, 107–115 (2016)
15. Peled, O., Fire, M., Rokach, L., Elovici, Y.: Entity matching in online social networks. In: Socialcom (2013)
16. Perito, D., Castelluccia, C., Kaafar, M.A., Manils, P.: How unique and traceable are usernames? In: PETS (2011)
17. Riederer, C., Kim, Y., Chaintreau, A., Korula, N., Lattanzi, S.: Linking users across domains with location data: Theory and validation. In: WWW (2016)
18. Shen, Y., Jin, H.: Controllable information sharing for user accounts linkage across multiple online social networks. In: CIKM (2014)
19. Shu Kai, E.A.: User identity linkage across online social networks: a review. SIGKDD Explorations Newsletter **18**(2), 5–17 (2017)
20. Tan, S., Guan, Z., Cai, D., Qin, X., Bu, J., Chen, C.: Mapping users across networks by manifold alignment on hypergraph. AAAI **14**, 159–165 (2014)
21. Xie, W., Mu, X., Lee, R.K.W., Zhu, F., Lim, E.P.: Unsupervised user identity linkage via factoid embedding. In: ICDM (2018)
22. Zafarani, R., Liu, H.: Connecting users across social media sites: a behavioral-modeling approach. In: KDD (2013)
23. Zhang, Y., Tang, J., Yang, Z., Pei, J., Yu, P.S.: Cosnet: Connecting heterogeneous social networks with local and global consistency. In: KDD (2015)
24. Zhou, X., Liang, X., Zhang, H., Ma, Y.: Cross-platform identification of anonymous identical users in multiple social media networks. TKDE **28**(2), 1 (2016)