東南大學
SOUTHEAST UNIVERSITY

SMU
SINGAPORE MANAGEMENT UNIVERSITY

Living Analytics Research Centre

Paper158: Retrofitting Embeddings for Unsupervised User Identity Linkage

Tao Zhou, Ee-Peng Lim, Roy Ka-Wei Lee, Feida Zhu, and Jiuxin Cao

Outline

I. Introduction

II. Related work

III. Approach – Retrofitting Embedding for UIL

IV. Experiments and Results

V. Conclusion

Outline

01 Introduction

- Multiplicity of online social networks (OSNs)
- Same users operating multiple user identities in different OSNs

User Identity Linkage Problem (UIL)

- Online social network (OSN) $G = (U,E)$
- Each user identity $u_i \in U$ is associated with some attributes, e.g., name, content, etc.
- Given two OSNs G_s 、 G_t as the source and target platforms

User Identity Linkage Problem： For each user identity u_s from U_s, find a user identity u_t from U_t， such that they belong to the same real person.

Outline

02 Related Work

- Supervised Approaches
  - Classification techniques
    - extract features from user attributes
    - train a classifier for predicting pairs of user identities
  - Embedding techniques
    - PALE: Structure features -- network embeddings
    - ULink: Profile features -- latent space
- Semi-supervised Approaches
  - Considers both labeled and unlabeled pairs of matching user identities in model learning
  - HYDRA, COSNET

Require ground truth label

Issues in collecting labeled data

- **Unsupervised Approaches**
  - Relatively fewer works
  - CNL: perform UIL in an incremental manner
  - Factoid Embedding: learn user embeddings with multiple attributes

02 Related Work

- **Challenges in unsupervised approach to UIL**
  - Multiplicity of user attributes and heterogeneous attribute domains
    - User name — String
    - User generated text — Text
    - User relationship — Network
    - User generated image — Visual imformation
    - ...
  - Similarity of cross-platform user attributes
    - need similarity measure to compare the attribute values
    - necessary to identify discriminative cross-platform attribute similarities
      - incorporate them into unsupervised UIL methods
  - Importance of user attributes

Outline

03 Approach

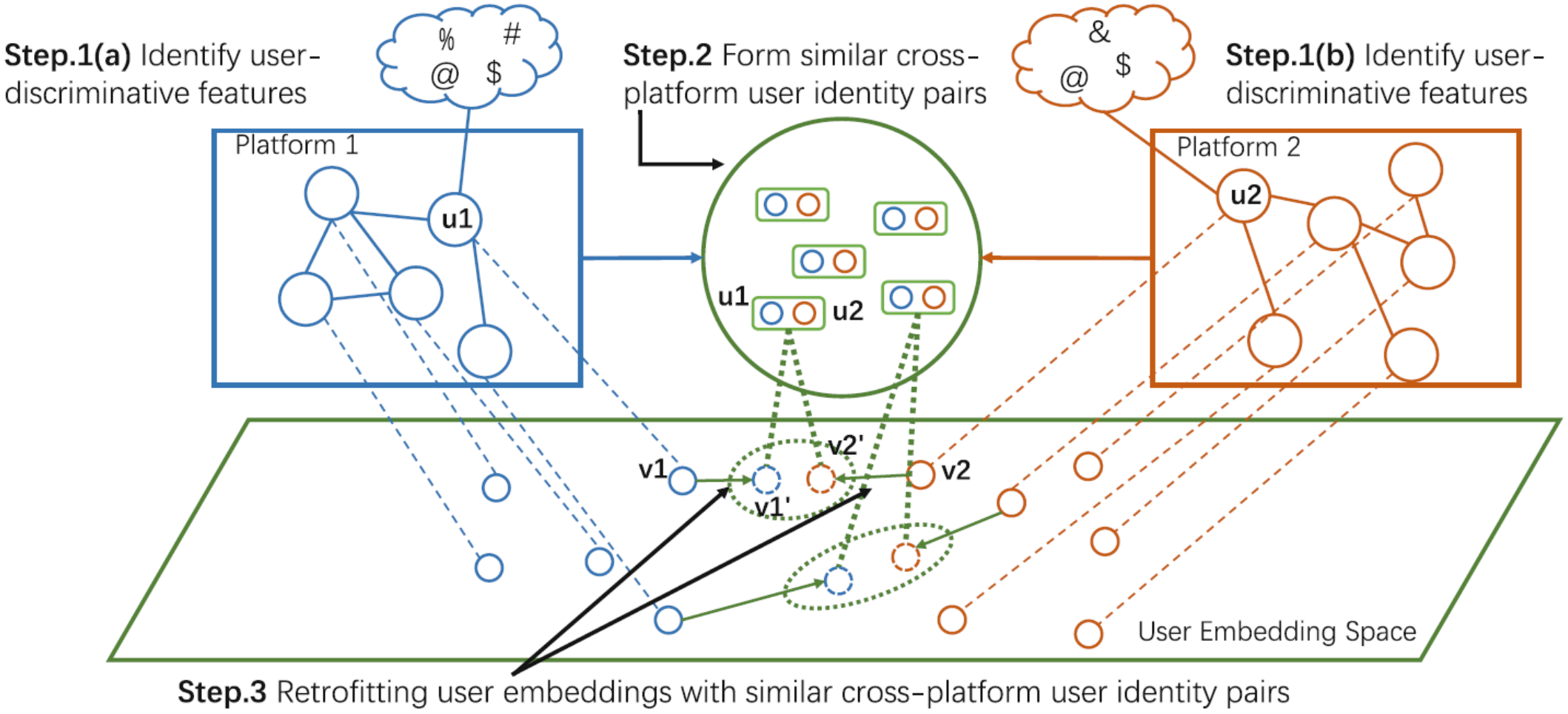
Retrofitting Embeddings + user-discriminative features

- Multiplicity of user attributes and heterogeneous attribute domains
  - **Retrofitting Embeddings**
  - map every user identity to a common embedding space with user attributes
  - fuse heterogeneous attributes by integrating base and retrofitting information

- Similarity of cross-platform user attributes
  - **user-discriminative features**
    - ones that are indicative of specific user identities in an OSN
    - used to distinguish useful information from noise for the purpose of UIL

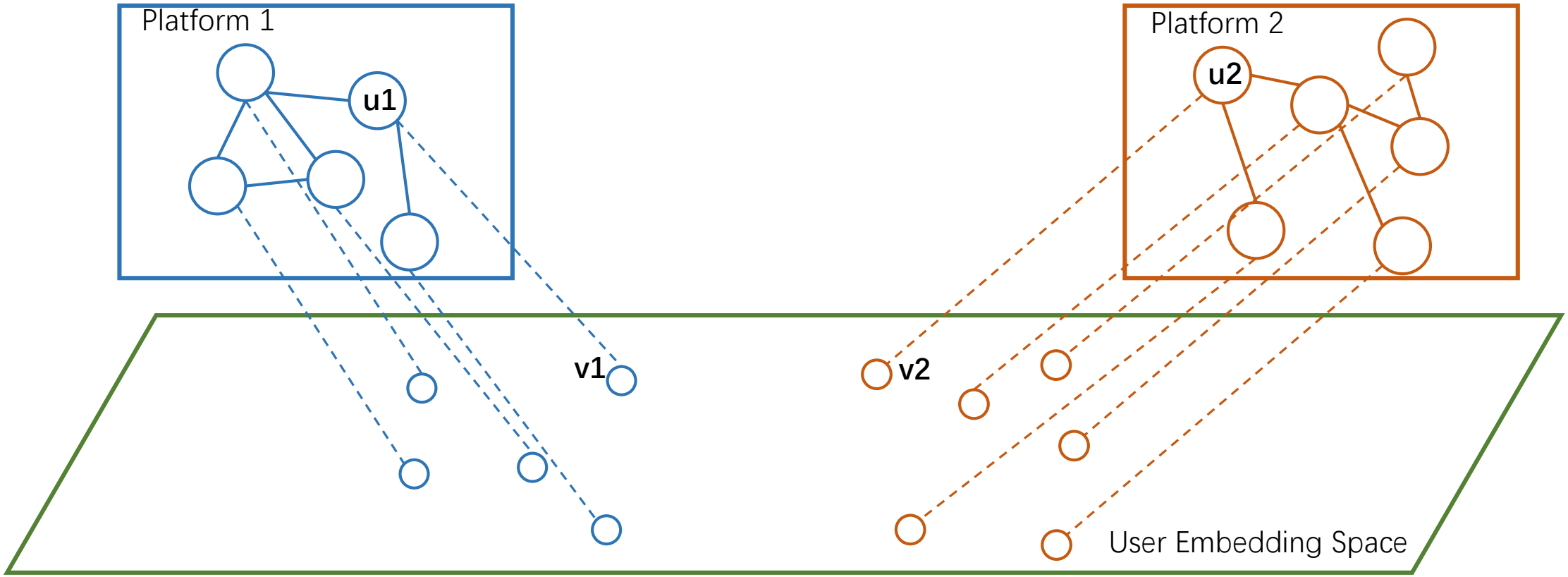
- Importance of user attributes
  - **Retrofitting Embeddings**
  - Weight control for different attributes in retrofitting process

03 Approach – Framework



03 Approach – Base Embedding

- Framework takes an existing user embedding as input
- User identities in each platform are mapped into the same user embeddings space



03 Approach – Base Embedding

1. Single Attribute-Based User Embeddings

$$O_a = \sum_{u_i, u_j \in U_s \cup U_t} \left(\mathbf{v}_i^\top \mathbf{v}_j - sim_a(u_i, u_j)\right)^2$$

Embeddings to be learnt

Similarity measure determining how two user names are similar to each other

- Name Embeddings

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

n-gram TF-IDF vector

03 Approach – Base Embedding

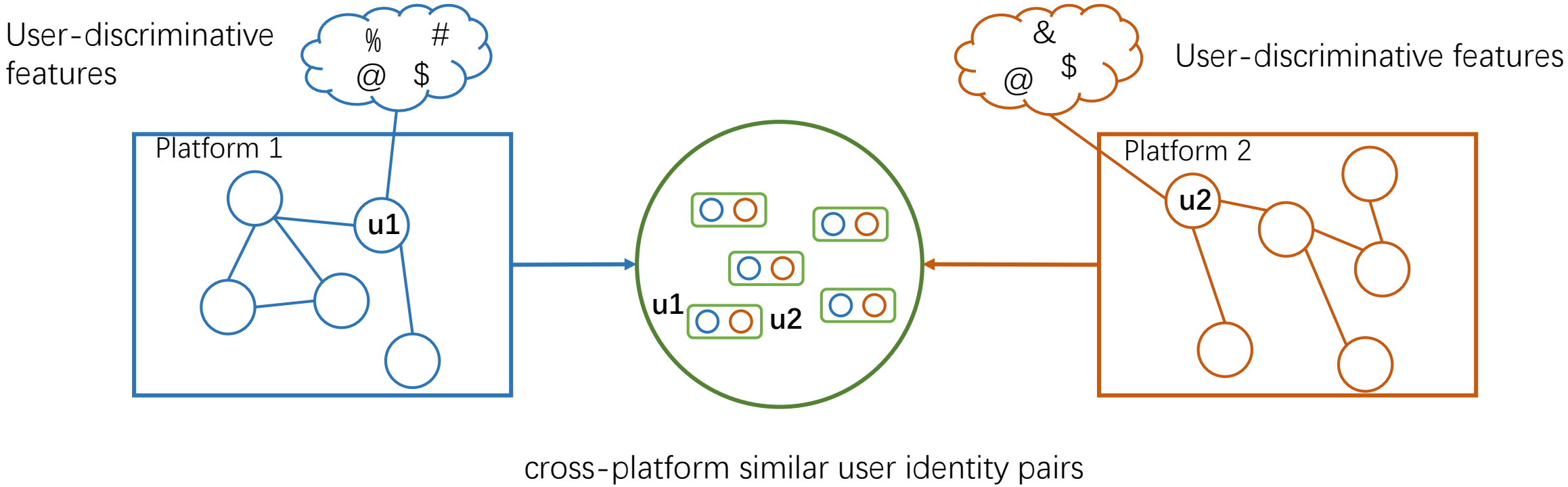
2. Factoid embedding*

- Embed all the objects
  - Name
  - User
  - …
- Factoid:
  - <u1, has_name, alice>
  - <u2, has_name, sam>
  - <u1, follows, u2>
  - …
- Key idea:
  - User embeddings that share many similar factoids will be closer to one another in the user embedding space

*Unsupervised User Identity Linkage via Factoid Embedding. ICDM. 2018

03 Approach – User-discriminative features

- User-discriminative features
- Cross-platform similar user identity pairs



03 Approach – User-discriminative features

- **User-discriminative features**
  - Help distinguishing a user identity from others in the same OSN
  - Derive different types of user-discriminative features from different user attributes like user name, user content, user network, etc..
    - E.g.: User-discriminative name features (discriminative n-grams)
  - Focus on real useful user information for UIL from the noise
- **Cross-platform similar user identity pairs**
  - Each cross-platform similar user identity pair (u_i , u_j) is assigned a similarity score s_{ij} according to user-discriminative features
    - E.g.: Jaccard Similarity of the set of user-discriminative n-grams
  - The larger the s_{ij}, the higher the likelihood that the u_i and u_j belong to the same user

03 Approach – Retrofitting Embedding

- $\hat{\mathbf{v}}_i$: base representation of u_i
- $\mathbf{v}_i$: retrofitted representation of u_i (to be learnt)
- $P = \{(u_i, u_j) | u_i \in U_s, u_j \in U_t, s_{ij} > 0\}$

➢minimize the objective function:

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

keeping the base user embedding and retrofitted embedding close

pushing cross-platform similar user identity pairs closer

$\varphi(a, b)$: cosine distance between vectors a and b

α $(0 \leq \alpha \leq 1)$: weight to adjust the degree of retrofitting

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

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

03 Approach – Retrofitting Embedding

Algorithm 1 Retrofitting Embedding For UIL

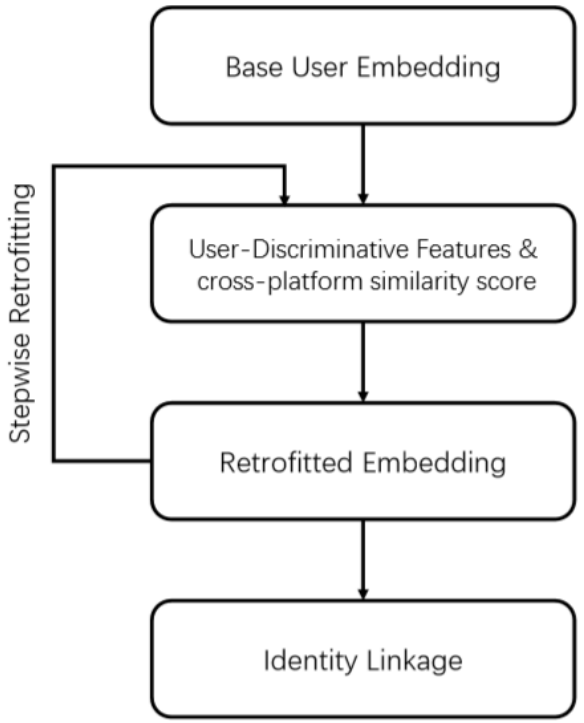
Input:
Source platform user set U_s, target platform user set U_t;
Base user embedding vectors $\hat{\mathbf{v}}_i$ for each $u_i \in U_s \cup U_t$;
Attribute set A and user-dicriminative features for each attribute $a_k, k \in [1, |A|]$;

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 $P^k = \left\{\{u_i, u_j\} \,|u_i \in U_s, u_i \in U_t, s_{ij}^k > 0\right\}$;
  # 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 $\{\mathbf{v}_i | u_i \in U_s \cup U_t\}$

Variants of Retrofitting Embeddings

1. **Stepwise Approach**



2. **Hierarchical Approach** to Generate User-Discriminative Features

- stricter threshold -> more user-discriminative features
- select a few thresholds
- derive different sets of user-discriminative features based on the thresholds
- a set of scores s_{ij} 's

Outline

04 Experiments and Results

➢ Dataset

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	

04 Experiments and Results

➢Experiment Metrics

HitRate@K: for each source user u_s^*, if u_t^* is within the top K candidates, i.e. $rank(u_t^*) \leqslant K$, then this matching is considered as correct.

$$HitRate@K = \frac{\#\ of\ correct\ predicted\ pairs}{\#\ of\ all\ matching\ pairs} \tag{11}$$

Mean Reciprocal Rank(MRR) MRR is a widely used measure for evaluating ranking algorithm. For the linkage task, it is defined as:

$$MRR = \frac{1}{n} \sum_{(u_s^*, u_t^*)} \frac{1}{rank(u_t^*)} \tag{12}$$

➢Baselines and Retrofitting Embeddings

- Name embeddings (NE)
- TF-IDF
- Content embedding (CE)
- Weighted content embedding (CE_weighted)
- Factoid Embedding (FE)

- Retrofitting Embedding RE_p^q
  - q as base user embeddings
  - p as the user-discriminative feature(s) used in RE

04 Experiments and Results

➢Experiments with Name and Network Attributes

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**

04 Experiments and Results

➢Experiments with Content and Network Attributes

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
$RE_c^{FE_c}$	0.7113	0.7263	0.7350	0.7388	0.7413	0.7488	0.7638	0.7261
$RE_c^{FE_c}$_h1	0.7175	0.7463	0.7588	0.7650	0.7675	0.7750	0.7950	0.7413
$RE_c^{FE_c}$_h2	0.7238	0.7500	0.7638	0.7688	0.7738	0.7813	0.7913	0.7465
$RE_{cb}^{FE_c}$	0.7413	0.7713	0.7800	0.7850	0.7888	0.7938	0.8113	**0.7638**

Outline

05 Conclusion

- A novel unsupervised user identity linkage (UIL) framework
- Enhancing existing UIL methods based on user embeddings techniques
- User-discriminative features and retrofitting embeddings
- Effectively improve the accuracy of different base user embeddings
- The quantum of improvement can also be surprisingly good even for original UIL methods with very poor matching accuracy

Thank You!