Information Processing and Management 61 (2024) 103866

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

Multi-granularity attribute similarity model for user alignment across social platforms under pre-aligned data sparsity

Yongqiang Peng [a], Xiaoliang Chen [a,b,c,*], Duoqian Miao [b], Xiaolin Qin [d], Xu Gu [a,d], Peng Lu [c]

[a] School of Computer and Software Engineering, Xihua University, Chengdu 610039, PR China
[b] College of Electronic and Information Engineering, Tongji University, Shanghai, 201804, PR China
[c] Department of Computer Science and Operations Research, University of Montreal, Montreal, QC. H3C3J7, Canada
[d] Chengdu Institute of Computer Applications, Chinese Academy of Sciences, Chengdu 610041, PR China

ARTICLE INFO

Keywords:
Social networks
User alignment
Multi-granularity attribute embedding
Attribute reappearance

ABSTRACT

Cross-platform User Alignment (UA) aims to identify accounts belonging to the same individual across multiple social network platforms. This study seeks to enhance the performance of UA tasks while reducing the required sample data. Previous research has focused excessively on model design, lacking optimization throughout the entire process, making it challenging to achieve performance without heavy reliance on labeled data. This paper proposes a semi-supervised Multi-Granularity Attribute Similarity Model (MGASM). First, MGASM optimizes the embedding process through multi-granularity modeling at the levels of characters, words, articles, structures, and labels, and enhances missing data by leveraging adjacent text attributes. Next, MGASM quantifies the correlation between attributes of the same granularity by constructing Multi-Granularity Attribute Cosine Distance Distribution Vectors (MA-CDDVs). These vectors form the basis for a binary classification similarity model trained to calculate similarity scores for user pairs. Additionally, an attribute reappearance score correction (ARSC) mechanism is introduced to further refine the ranking of candidate users. Extensive experiments on the Weibo-Douban and DBLP17-DBLP19 datasets demonstrate that compared to state-of-the-art methods, The hit-precision of the MGASM series has significantly improved by 68.15% and 27.02%, almost reaching 100% precision. The F1 score has increased by 37.6% and 21.4%.

1. Introduction

Individuals frequently engage with multiple social networking platforms in the contemporary digital landscape, each offering a unique array of necessary services. For instance, LinkedIn caters specifically to professional networking and job seeking, while Twitter facilitates real-time news and discussions. In the Chinese context, Weibo has emerged as a microblogging platform similar to Twitter for rapidly sharing text and media. Douban meets essential needs for user reviews and recommendations about cultural products like books, music, and films. Different specialties around social connections, content sharing, messaging, and creative outlets compel users to diversify across networks to meet routine needs. The proliferation of social media usage has brought to the fore the significance of User Alignment (UA) tasks. The primary objective of UA is to predict individuals who may belong to the same natural person across different social networks, particularly in scenarios where explicit UA information is absent. UA is vital for enhancing scientific applications like link prediction (Zhang, Yu, & Zhou, 2014), cross-network user tracking (Oberle, Berendt,

* Corresponding author at: School of Computer and Software Engineering, Xihua University, Chengdu 610039, PR China.
E-mail address: chenxl@mail.xhu.edu.cn (X. Chen).

https://doi.org/10.1016/j.ipm.2024.103866
Received 24 February 2024; Received in revised form 8 August 2024; Accepted 10 August 2024
Available online 23 August 2024

Hotho, & Gonzalez, 2003; Ren et al., 2022), public opinion analysis (Liu & Wu, 2023), and cross-network recommendations (Singh, N., L., Sanghavi, Vaghela, Manoharan, Hamdi, & Tunze, 2022; Zhao, Zhao, He, Zhang, & Fan). However, UA is challenged by data heterogeneity, platform-specific user behaviors, and a lack of shared identifiers (email addresses, phone numbers, ID numbers, etc.). These factors complicate user profile linkage, behavioral modeling, and limit supervised learning due to scarce labeled examples. Specifically, these constraints pose two central obstacles that limit the performance of UA tasks:

- **Hurdles in Attribute Information Extraction.** For effective User Alignment (UA), a fundamental obstacle lies in extracting meaningful features from complex and sparse user data. The heterogeneity of user profile attributes further complicates the formulation of a standardized feature set that accurately represents users' digital identities. This challenge presents a dual aspect. Firstly, current methodologies encounter difficulties in adapting to various user profile attributes. For example, "car" and "engine" exhibit strong correlation in most scenarios. However, when these terms are used as user nicknames, their semantic relationship should not be considered valid evidence for user alignment, as nickname correlations in the real world are typically not manifested in word relationships. Therefore, different attributes may require handling with varying semantic granularity in feature extraction. Secondly, isolating and leveraging additional evidence from user-generated text is challenging. For example, user articles contain a wealth of information, but the current utilization of user articles remains coarse-grained. Addressing these issues necessitates sophisticated feature extraction techniques capable of identifying and utilizing subtle details in user data.
- **Challenge of High Precision.** Another major challenge in UA tasks is to improve matching accuracy under sparse data conditions. Given the specificity of UA, precision should be of particular concern because inaccurate judgments can directly and chainly impact downstream applications, leading to catastrophic consequences. Achieving high precision depends on advanced algorithms. These algorithms need a lot of training data to capture users' subtle behaviors across different platforms. However, user data obtained from real social networks is often sparse, and the acquisition of pre-aligned users relies on manual efforts, making it difficult to accumulate sufficient data. Additionally, social platforms typically have large user populations, including individuals with overlapping attributes, making it challenging to differentiate users with similar profiles in the absence of explicit distinguishing information.

In light of the aforementioned challenges, our research proposes a comprehensive solution called the Multi-Granularity Attribute Similarity Model (MGASM): Firstly, we segment user texts into characters, words, articles, and labels based on user profiles and obtain the structural features of the users' network through user relationship analysis. Optimal embedding methods are selected for each granularity to create vectors that accurately reflect language features. This feature extraction approach effectively addresses the obstacles in attribute information extraction. Secondly, we calculate the cosine distance between pre-aligned user pairs for each granularity feature vector, quantifying the correlation between attributes of the same granularity, known as the Multi-Granularity Attribute Cosine Distance Distribution Vector (MA-CDDV). Subsequently, we train a binary classifier using MA-CDDV. This enables the model to learn the distribution characteristics of pre-aligned user correlations across five granularities: characters, words, articles, labels, and structure. Similarity scores are generated for user vector pairs from different networks, where a score close to 1 indicates that the user accounts likely represent the same individual. Thirdly, we construct a BallTree (Dolatshah, Hadian, & Minaei-Bidgoli, 2015) using the feature vectors of all users from one social platform as the point set, with the exponential decay value of the similarity score as the distance metric. Then, we perform a nearest neighbor search using the feature vectors from another social platform to obtain a list of potential candidates. Lastly, through sampling surveys in real social network scenarios, we discovered that when a user's character-granularity attribute appears in another user's article-granularity attribute, it increases the likelihood that these two users represent the same natural person. We analyzed this correlation and identified two main reasons: (a) users promote themselves across platforms, (b) character-granularity attributes related to a user's interests or profession are likely to be mentioned again in article-granularity attributes on other social platforms. We refer to this phenomenon as Attribute Reappearance (AR) and have designed a correction factor to adjust the similarity scores of user pairs exhibiting AR, thereby optimizing the ranking of potential candidates and further improving the accuracy of UA. We call this process Attribute Reappearance Score Correction (ARSC).

The major contributions of this paper can be summarized as follows:

- **Comprehensive Granular Feature Extraction:** We introduce a feature extraction method that classifies user attributes into four levels of granularity, capturing a broad spectrum of user characteristics for enhanced user profile matching across social platforms.
- **Unsupervised Embedding Technique:** Our approach utilizes unsupervised embedding for user attribute modeling, reducing dependence on large-scale labeled data and enabling application in data sparse environment.
- **Innovative Multi-granularity Attribute Cosine Distance Distribution Vector:** This study introduces a novel vector (MA-CDDV) for quantifying the correlation of users on attributes of the same granularity. Training a binary classification similarity model with this vector can achieve outstanding performance while significantly reducing the required sample size.
- **Attribute Reappearance Score Correction Mechanism:** Introduced a score correction mechanism(ARSC), enhancing the model's ability to accurately identify in complex social network scenarios.

The remaining sections of this article are organized as follows. Section 2 reviews and summarizes relevant work. Section 3 formally defines the research problem. Section 4 elaborates on the research objectives. Section 5 provides a detailed description of the proposed model. Section 6 presents the experimental results. Section 7 describes the results of the experiments and their impact on the task domain. Finally, Section 8 concludes and provides future perspectives on this work.

2. Related work

UA is also known as user identity linkage, network alignment, or anchor link prediction. Its primary objective is to identify the same individual across different networks. This process is also referred to as "alignment". Aligned accounts are termed as aligned user pairs, serving as anchor points that connect various social networks. According to the different learning modes of UA methods, they can be categorized into three types: unsupervised learning, semi-supervised learning, and supervised learning.

Firstly, in the unsupervised learning mode, Zhou et al. (2020) proposed the NWUIL model, where the authors formulated the user identity linkage task as an optimal network transport problem. They introduced an unsupervised mapping process based on network Wasserstein distance to reduce the reliance on anchored nodes. In the research by Zhou, Lim, Lee, Zhu and Cao (2020), user discrimination features and restoration embeddings were emphasized. By designing user discrimination features, they obtained pairs of similar user identities across online social networks (OSN). These pairs were then utilized to adjust the underlying user embeddings, improving the basic user embeddings of existing UIL methods. To reduce the reliance on profile configurations, Liang et al. (2021) proposed an alignment framework called LSNA. LSNA guides the embedding process by integrating topological information and network relevance. Additionally, they addressed the scalability issue of large-scale network alignment problems through network decomposition strategies. Zhou et al. (2022) introduced the Unsupervised Adversarial Network Alignment (UANA) method. This framework combines Generative Adversarial Networks (GAN) and Reinforcement Learning (RL) techniques to address key challenges in network alignment. In the recent exploration of unsupervised methods, Lei, Feng, Jie, and Shu (2023) focused on achieving a balance between accuracy and efficiency. They performed targeted optimizations in both the model training phase and the network alignment phase, ensuring improved performance while reducing alignment time and memory requirements. Li et al. (2019) proposed the MC2 model to address the challenge of unsupervised alignment across multiple networks. The MC2 model first designed a matrix optimization to infer a common subspace from different social networks and developed an efficient alternating algorithm to solve the non-convex optimization problem.

Unfortunately, it is regrettable that the research in the domain of Unsupervised Alignment (UA) tasks is not abundant due to the significant challenges it faces. On the contrary, Supervised Learning and Semi-Supervised Learning have garnered widespread attention for their outstanding performance. In recent studies, Duan, Long, Xiao, Wang, and Li (2024) and Wei, Zhou, An, Yang, and Xiao (2023) integrated UA tasks with e-commerce, providing targeted optimizations and extending the scope of UA scenarios. Huang, Zhao, Zhang, Xing, Wu, and Ma (2023) proposed a Semantic-Enhanced Social Network User Alignment algorithm (SENUA). This algorithm aligns users using user attributes, User-Generated Contents (UGCs), and user check-ins. By leveraging semantic features from these three factors, it effectively reduces noise interference and further enhances the algorithm's adaptability to noise. Additionally, Shao, Wang, Gao, Shi, Shen, and Cheng (2023) introduced a model for user alignment by matching asymmetric information of geographic locations and text on two social platforms. They reduced the model's dependence on labeled data by externally introducing text-location pairs. Li et al. (2023) argued that embedding identity as a deterministic vector into a shared latent space cannot address the various uncertainties in real social networks. In their study, each social identity is represented as a Gaussian distribution in Wasserstein space to preserve the granularity of social profiles and the uncertainty of identity in the model. Similarly, Wang et al. (2022) also preserved structural information by embedding each node in the network as a Gaussian distribution. In earlier research, Chen et al. (2022) conducted in-depth studies on UA tasks. They innovatively introduced the concept of attribute hierarchy and improved UA performance through hierarchical embedding (Chen & Chen, 2022). In their subsequent research, they further optimized this hierarchy, proposing sub-word attributes to enhance UA tasks (Yang, Chen, & Chen, 2022). Sun et al. (2022) argue that most existing studies consider social networks as static, overlooking their inherent dynamics. Therefore, they first proposed a dynamic network alignment framework called DGA, which captures network evolution information and aligns embedding representations of the same individuals in a common subspace, thus addressing the dynamic network alignment problem.

With the development of graph-related research, many researchers have applied graph-related techniques to UA tasks (Li, Zhou, Chen & Zhao, 2023; Long, Chen, Du, & Wang, 2023; Park, Tran, Shin, & Cao, 2022b; Qi, Chen, Sun, Luan, & Tong, 2023). For example, Long et al. (2023) utilized a degree-aware graph neural network model to address the issue of long-tail user identity linkage (UIL). Similarly, Li, Zhou et al. (2023) applied Graph Convolutional Networks (GCN (Patnaik & Patgiri, 2023)) to UA problems for exploring spatial proximity between user actions and check records. Zhang and Tong (2018) found in earlier research that existing network alignment methods can utilize node attribute similarity as part of prior alignment information, yet most methods primarily explore topological consistency without considering consistency among the underlying network attributes. Therefore, they proposed a network alignment algorithm called FINAL, which leverages node/edge attribute information to guide the (topology-based) alignment process. Subsequently, the team continued to focus on research in the network alignment field and further discovered that assuming alignment consistency might lead to the problem of oversmoothing, making it difficult to distinguish between correct and misleading alignments, and existing methods lack a deep understanding and analysis of the trade-off between alignment consistency and diversity. To address this, they introduced the NeXtAlign method (2021) (Zhang, Tong, Jin, Xia, & Guo, 2021), which strives to maintain alignment consistency while reflecting alignment diversity, addressing the shortcomings of current network alignment methods in this trade-off. Additionally, Yan, Zhang, and Tong (2021) recognized that methods optimized based on attribute consistency are overly strict and unable to cope with the challenge of network heterogeneity, while methods based on network embeddings, although not assuming consistency, suffer from embedding space disparities. Hence, they proposed the BRIGHT method, which utilizes random walk restarts to construct a unified embedding space from anchor nodes, avoiding the limitations of consistency optimization methods and the issue of embedding space disparities. Similar findings were observed.

Furthermore, many methods for entity alignment tasks are also worth considering. For instance, Tang, Song, Huang, Gao, and Yu (2024) recently observed the suboptimal performance of previous methods in low-resource language knowledge graphs. In

response to this challenge, they generated pseudo-sentences based on relationship triplets, utilized pre-trained language models for representation generation, and explored semantic information from connected relationships through graph neural networks. Zhu, Bao, Liu, Han, Wang, and Peng (2023) focused on entity alignment in cross-lingual knowledge graphs. By integrating relationship awareness and attribute participation, they aimed to enhance alignment accuracy and robustness. This brings new insights to cross-lingual entity alignment. Li, Dong and Qin (2023) proposed a dual-view graph neural network model that encodes the graph from two perspectives to achieve better entity alignment. Munne and Ichise (2023) utilized embedding representations of entity abstract information and attribute information, combining them through weighted averaging to optimize the embedding process. Fanourakis, Efthymiou, Kotzinos, and Christophides (2023) analyzed the performance, advantages, and disadvantages of various embedding techniques in entity alignment tasks. Additionally, they discussed challenges in industrial datasets and proposed further research questions. The method proposed by Zhao et al. (2020) for handling missing data and sparsity in heterogeneous information networks has inspired us in addressing similar issues in UA tasks.

In addition to the aforementioned supervised and unsupervised methods, the recent work by Sun et al. (2023) is impressive. They proposed a deep reinforcement learning method called GroupAligner. Their work focuses on aligning social groups rather than individual profiles. By using a cyclic domain adaptation method based on Wasserstein distance to transfer knowledge from the source social network, they modeled group discovery as a sequential decision process and used reinforcement learning to handle it. This approach brings new insights into user alignment tasks.

These methods have significantly contributed to the development of UA but still exhibit shortcomings. The main issues include: (1) Feature embedding limitations, as they fail to fully capture the complex features of heterogeneous profiles. This leads to limited representations of user text attributes due to suboptimal extraction processes. (2) Challenges in the prediction phase, characterized by a lack of a preliminary quantification process for granularity attribute correlations and a failure to consider interactions between attributes across granularities. This results in difficulties balancing the contradiction between sample quantity and performance, making it challenging to achieve a leap in performance under sparse data conditions.

3. Problem definition

This section outlines the UA challenge in social networks and the notations used in our study, detailed in Table 1. A social networks is modeled as undirected graph $G = (V, E, A)$ with user accounts as vertices $V = \{v_1, v_2, \ldots, v_n\}$ and relational links (e.g., friendships, follower relationships) as edges $E = \{e_{ij} = (v_i, v_j) | v_i, v_j \in V\}$. Each user is associated with a set of attributes, including personal details (e.g., usernames, affiliations, interests) and user-generated content (e.g., posts, publications, comments). These attributes are represented as a set A that can be categorized into four granularities: character-granularity attribute subset A_{ch}, word-granularity attribute subset A_{wo}, article-granularity attribute subset A_{ar}, and label-granularity attribute subset A_{la}, allowing for multi-resolution analysis.

The goal of the UA task is to match user accounts across different network platforms in the absence of explicit unique identifiers. The formal description of this task is as follows:

Definition 1. Let $G^\alpha = (V^\alpha, E^\alpha, A^\alpha)$ and $G^\beta = (V^\beta, E^\beta, A^\beta)$ denote two disjoint social networks. Given a small seed set $S = \{(v_i, v_j) | v_i \in V^\alpha, v_j \in V^\beta\}$ of pre-aligned user account pairs from G^α and G^β corresponding to the same underlying individual. The *User Alignment (UA)* task involves extracting a set $L = \{(v_i, v_j) | v_i \in V^\alpha, v_j \in V^\beta, (v_i, v_j) \notin S\}$ of additional cross-network account pairs mapping to the same real-world identity.

An example of UA task between two social networks G^α and G^β is presented in Fig. 1, where the solid black lines such as (r_1, r'_1) within each network indicate existing user relationships, while dashed lines like (r_2, r'_2) suggest potential, yet unconfirmed, connections. The blue solid lines across networks represent the user alignment challenge. The objective is to identify new matching user pairs in L across G^α and G^β by exploiting attribute and connection similarities, using a limited set S of known seed alignments as supervision.

4. Research objectives

The primary objective of this study is to explore the issue of user alignment under sparse data conditions and propose the solution MGASM. Through theoretical analysis and empirical research, we aim to delve into the fundamental causes of this problem and strive to provide new insights for this field. The outcomes of this research are anticipated to contribute to the enhancement of existing user alignment (UA) technologies. They will also offer valuable references for future studies. Specifically, our research goals include:

- **Analyzing Inherent Challenges**: By thoroughly understanding the problem definition and application scenarios of User Alignment (UA), we seek to identify inherent obstacles that impede the performance of UA. This analysis will set the tone for our solution, laying the groundwork for addressing the challenges and providing a solid foundation for our proposed approach.
- **Analyzing Shortcomings of Existing Methods**: Through a comprehensive examination of current user alignment methods, we aim to gain a deeper understanding of the challenges they face in handling sparse data, heterogeneous information, and cross-platform differences. This process will assist us in clarifying the challenges within the User Alignment (UA) domain and provide guidance for our solution.

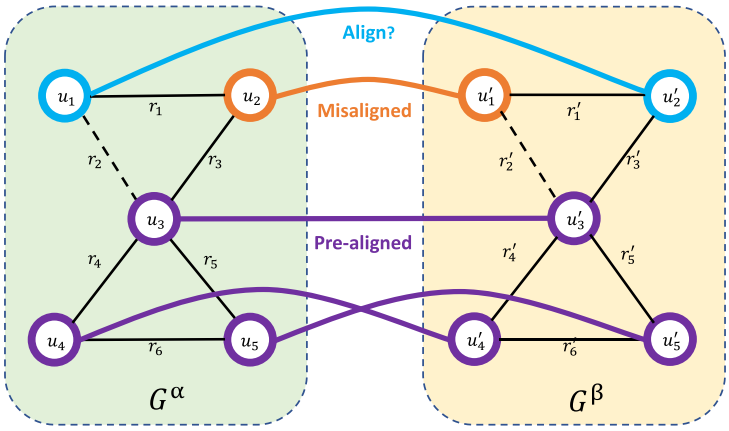


Fig. 1. Illustration of the user alignment task. Purple nodes represent known aligned user pairs, orange nodes represent non-aligned user pairs, black solid lines represent known user connections, black dashed lines indicate potential but unconfirmed connections, and blue solid lines represent the user alignment task. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1
Notation description.

Notation	Description
$G^\alpha = G^\beta$	Social networks α and β that participate in alignment.
V	Set of users in a social network.
E	Set of edges in a social network.
A	Set of user attributes in a social network.
$\mathbb{R}$	Set of real numbers.
$A_{ch}, Z_{ch}, \overrightarrow{\mathbf{z}^{ch}}$	Character-granularity attribute set A_{ch}, feature matrix Z_{ch} of A_{ch}, and feature vectors $\overrightarrow{\mathbf{z}^{ch}}$ composing Z_{ch}.
$\overrightarrow{\mathbf{x}^{ch}}$, X_{ch}	Character frequency vector and character frequency matrix.
$A_{wo}, Z_{wo}, \overrightarrow{\mathbf{z}^{wo}}$	Word-granularity attribute set A_{wo}, feature matrix Z_{wo} of A_{wo}, and feature vectors $\overrightarrow{\mathbf{z}^{wo}}$ composing Z_{wo}.
$A_{ar}, Z_{ar}, \overrightarrow{\mathbf{z}^{ar}}$	Article-granularity attribute set A_{ar}, feature matrix Z_{ar} of A_{ar}, and feature vectors $\overrightarrow{\mathbf{z}^{ar}}$ composing Z_{ar}.
$Z_{st}, \overrightarrow{\mathbf{z}^{st}}$	Structure feature matrix Z_{st} of a social network, and the feature vectors $\overrightarrow{\mathbf{z}^{st}}$ composing Z_{st}.
$A_{la}, Z_{la}, \overrightarrow{\mathbf{z}^{la}}$	Label-granularity attribute set A_{la}, feature matrix Z_{la} of A_{la}, and feature vectors $\overrightarrow{\mathbf{z}^{la}}$ composing Z_{la}.
$d_{ch}, d_{wo}, d_{ar}, d_{la}, d_{st}$	Dimensions of character-granularity, word-granularity, article-granularity, label-granularity, and structure embedding, respectively.
$Z, \vec{\mathbf{z}}$	Overall feature matrix Z for all users and feature vectors $\vec{\mathbf{z}}$ composing the feature matrix Z.
d	Total dimension $d = d_{ch} + d_{wo} + d_{ar} + d_{st} + d_{la}$
$E^p, \overrightarrow{e_p}$	Positive sample set E^p and vectors $\overrightarrow{e_p}$ composing E^p.
$E^n, \overrightarrow{e_n}$	Negative sample set E^n and vectors $\overrightarrow{e_n}$ composing E^n.
M_d, M_s	Distance matrix and similarity matrix

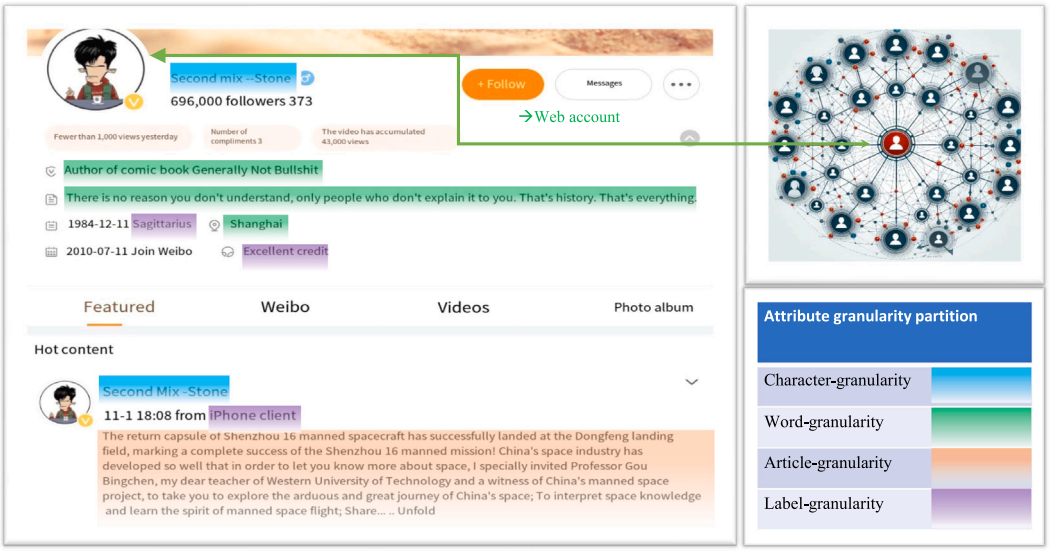
- **Designing a Comprehensive Solution**:Building upon a profound understanding of existing challenges, we will develop a comprehensive User Alignment (UA) solution aimed at overcoming challenges such as data sparsity, heterogeneity, and platform differences. Our objective is to deliver a high-performance, robust solution capable of adapting to diverse application scenarios.
- **Validating the Solution's Effectiveness**: Ultimately, our goal is to extensively test and validate the effectiveness and practicality of our proposed solution using real-world datasets. We will compare our approach with existing methods to demonstrate the significant performance advantages of our method across various application scenarios.

5. Multi-granularity attribute similarity model

5.1. Overview

This section delineates the architectural details underpinning the proposed Multi-Granularity Attribute Similarity Model (MGASM) for UA tasks. MGASM comprises three key components shown in Fig. 3:

- A multi-granularity attribute embedding modular that transforms user information of varying granularities, including character, word, article, label, and structure, into high-dimensional vector representations capturing semantic meanings.
- A MA-CDDVs construction module obtains distances by calculating the Cosine distance between the feature vectors of user pairs across two social networks with the same granularity attributes. The MA-CDDVs quantify the correlation between attributes of the same granularity for user pairs.



5/42

Fig. 2. An example of user attribute granularity division on the Sina Weibo website: the username is divided into character-granularity, the personal bio and location are divided into word-granularity, the phone model and reward information are divided into character-granularity, and the published articles are divided into article-granularity.

- A binary classification similarity model predicts the similarity score of user pairs based on the MA-CDDVs. This score is further adjusted using ARSC to indicate the likelihood that two users represent the same entity.

This approach architecture enables jointly modeling the intricate correlations across heterogeneous user attributes and distinct social networks. The main ideas of the three modules are described as follows.

Firstly, text-based user attributes from two social networks are classified into four granularities: characters (A_{ch}), words (A_{wo}), articles (A_{ar}), and labels (A_{la}). This classification principle is based on the different manifestations of the correlation of user attributes. Specifically, the correlation of A_{ch} for a pair of users is reflected by literal similarity, A_{wo} is reflected by word relationships, and A_{ar} is reflected by semantic similarity. A_{la} can limit the potential matching space. For example, taking the social network platform Sina Weibo (Fig. 2), usernames conform to the A_{ch} attribute classification principle. Personal introductions can be classified as A_{wo} attributes. The content posted by users can be considered as A_{ar} attributes. A_{la} attributes include device stars and other website labels. Then, different embedding functions are applied to each granularity to obtain vector representations, namely $\overrightarrow{\mathbf{z}^{ch}}$, $\overrightarrow{\mathbf{z}^{wo}}$, $\overrightarrow{\mathbf{z}^{ar}}$, and $\overrightarrow{\mathbf{z}^{la}}$. Additionally, structural feature vectors $\overrightarrow{\mathbf{z}^{st}}$ of users are learned through graph embedding methods.

Secondly, MA-CDDVs are obtained by calculating the Cosine distance between pre-aligned user pairs in the vectors of the same granularity attributes. This is done to quantify the correlation between attributes of the same granularity.

Thirdly, a lightweight binary classification similarity model is trained using MA-CDDVs. The model predicts similarity scores by analyzing the distribution of similarity between a pair of users across different granularity attributes. Due to the low-dimensional nature of MA-CDDVs (with the dimensionality equal to the number of attribute granularities), the model can quickly converge with very few samples.

5.2. Multi-granularity attribute embedding

This subsection delineates the embedding process for user attributes derived from the two social networks G^{α} and G^{β} under consideration for UA. We outline the attribute embedding process for a single social network, represented generically by the graph G without loss of generality. The techniques described are applicable to the user attribute sets from either of the social networks G^{α} or G^{β}.

5.2.1. Character-granularity attribute embedding

In the context of character-granularity attributes $A_{ch} = \{a_1^{ch}, a_2^{ch}, \ldots, a_i^{ch}, \ldots, a_n^{ch}\}$, which include elements like usernames or nicknames that lack rich semantic content, the focus is on capturing superficial string similarities rather than semantic meanings. For instance, the attributes "apple" and "pear" with respect to the nicknames of two users exhibit the semantic similarity of fruits yet convey limited meaning regarding user identity when used as semantics-devoid nicknames. Hence, for matching A_{ch}, it is critical to suppress semantic interpretations and instead prioritize superficial string similarities.

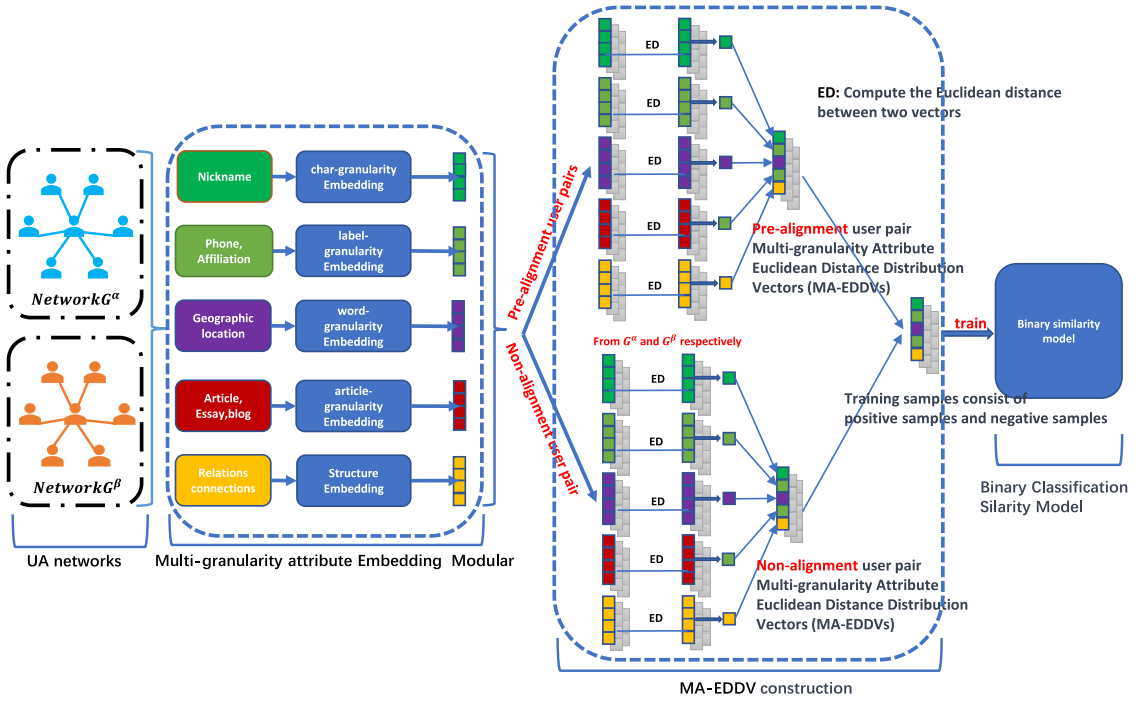


Fig. 3. The overall architecture of the MGASM model is presented. The first and second sections explain the origins of various granular attributes. The third section describes the construction process of MA-CDDVs, which are primarily used to quantify the correlation between attributes of the same granularity. These are then used as inputs for the binary classification similarity model in the fourth section.

A character frequency-based embedding approach is used to encode A_{ch}, transforming each attribute of a social network G into a frequency distribution vector that reflects stylistic patterns without considering explicit semantics. Formally, Let $A_{ch} = \{a_1^{ch}, a_2^{ch}, \ldots, a_i^{ch}, \ldots, a_n^{ch}\}$ represent the set of character-granularity attributes for all users in network $G = (V, E, A)$, with a_i^{ch} denoting the attribute of each user v_i. The attribute set A_{ch} can be decomposed into dictionary tokens $c = \{c_1, c_2, \ldots, c_k\}$ encompassing characters, numbers, and Q-grams (Ukkonen, 1992). Each token is assigned a unique index. Further refinement, concerning a particular attribute instance $a_i^{ch} \in A_{ch}$ belonging to user v_i, the attribute instance a_i^{ch} can be decomposed into $m \leq k$ constituent tokens $w = \{w_1, w_2, \ldots, w_m\}$ that map to entries in the dictionary. Each user's attribute a_i^{ch} can be represented as a k-dimensional character frequency vector denoted by $\overrightarrow{\mathbf{x}_i^{ch}} = \left[x_{c_1}, x_{c_2}, \ldots, x_{c_j}, \ldots, x_{c_k}\right]^T$, where element x_{c_j} corresponds to the occurrence count of token c_j in a_i^{ch}.

Taking the character attribute a^{ch} = "Alfred V. Aho" as an example, the character frequency vector $\overrightarrow{\mathbf{x}_i^{ch}}$ is constructed with elements representing the count of each character in the attribute, such as $[2, 0, 0, \ldots, 1, \ldots]$ for 'a', 'b', 'c', …, 'o', etc. Having established the representation for a single user's attributes, the character frequency matrix for the entire social network G is denoted as $X_{ch} = \left\{\overrightarrow{\mathbf{x}_1^{ch}}, \overrightarrow{\mathbf{x}_2^{ch}}, \ldots, \overrightarrow{\mathbf{x}_n^{ch}}\right\}^T \in \mathbb{R}^{n \times k}$, where k is the size of the dictionary. This matrix is high-dimensional and sparse due to the sparsity of individual character frequency vectors $\overrightarrow{\mathbf{x}_i^{ch}}$ and the nature of social networks.

To enhance computational efficiency, the sparse matrix X_{ch} is condensed into a lower-dimensional dense matrix Z_{ch} using an autoencoder (Zhai, Zhang, Chen, & He, 2018).

The encoder layer function is defined as:

$$Z_{ch} = f(X_{ch}) = X_{ch} W_1 + b_1 \tag{1}$$

with $W_1 \in \mathbb{R}^{(k \times d_{ch})}$ being the weight matrix reducing the dimensionality from k to d_{ch}, and $b_1 \in \mathbb{R}^{(1 \times d_{ch})}$ being the bias vector that aids in model fitting. This process yields a compressed hidden representation $Z_{ch} \in \mathbb{R}^{(n \times d_{ch})}$ from the original high-dimensional input $X_{ch} \in \mathbb{R}^{(n \times k)}$.

The decoder layer of the autoencoder is defined by the function:

$$X'_{ch} = g(Z_{ch}) = Z_{ch} W_2 + b_2 \tag{2}$$

where $W_2 \in \mathbb{R}^{(d_{ch} \times k)}$ is the weight matrix that expands the dimensionality of the hidden representation Z_{ch} back to the original size k, and $b_2 \in \mathbb{R}^{(1 \times d_{ch})}$ is a bias vector that contributes to the model's fitting capacity. This layer reconstructs the original high-dimensional input from the compressed hidden code $Z_{ch} \in \mathbb{R}^{(n \times d_{ch})}$, resulting in the matrix $X'_{ch} \in \mathbb{R}^{(n \times k)}$.

The mean squared error (MSE) is used as the loss function for the autoencoder due to the matching dimensions of the original matrix X_{ch} and the reconstructed matrix X'_{ch}. The MSE loss (omitting subscripts ch for simplicity) is calculated as

$$\text{MSELoss}(X, X') = \frac{1}{nk}\sum_{i=1}^{n}\sum_{j=1}^{k}(X'_{ij} - X_{ij})^2 \tag{3}$$

After training, when the input X_{ch} is provided, the autoencoder transforms X_{ch} into a low-dimensional dense representation Z_{ch} using Eq. (1). The embedding process of character-granularity attributes is summarized in Algorithm 1.

Algorithm 1 Character-granularity Attribute Embedding

Require: Character-granularity attribute set A_{ch} and embedding dimension d_{ch}
Ensure: Character-granularity feature matrix Z_{ch}

```
function CHARACTEREMBEDDING(A_ch, d_ch)
    Initialize X_ch = []
    tokens← charTokenize(A_ch)
    tokensDict← createDictionary(tokens)
    for a_i^ch in A_ch do
        chars← CharTokenize(a_i^ch)
        x_i^ch ← CountVector(chars, tokensDict)
        X_ch.append(x_i^ch)
    end for
    model← Autoencoder(hidden_dim = d_ch)
    repeat
        train(model, X_ch)
    until Convergence
    Z_ch ← model.encoder(X_ch)
    return Z_ch
end function
```

5.2.2. Word-granularity attribute embedding

Word-granularity attributes, such as locations or educational backgrounds, necessitate the modeling of complex word relationships. To learn semantic embeddings for these attributes, we utilize the word2vec (Guo, Huang, Dong, Zhang, & Xu, 2021) based on the Continuous Bag-of-Words (CBOW) architecture (Mikolov, Chen, Corrado, & Dean, 2013), which captures syntactic and semantic word contexts. Formally, let $A_{wo} = \{a_1^{wo}, a_2^{wo}, \ldots, a_i^{wo}, \ldots, a_n^{wo}\}$ represent the set of word-granularity attributes for users in network G, with a_i^{wo} being the word-granularity attribute for user v_i. These attributes are tokenized into unique words.

Building upon this, we combine the Wikipedia corpus with the word-granularity attributes of all users in G to construct a comprehensive corpus. Subsequently, we utilize this corpus to train a word2vec model. The vector representation for each sentence is obtained by averaging the Word2Vec vectors of each word in the word-granularity attributes. Taking the affiliation of Turing Award winner Alfred V. Aho in the DBLP dataset as an example, "Columbia University, New York City, USA", his word-granularity feature vector would be the mean of the Word2Vec word vectors corresponding to "University", "Columbia", "New York", and "USA". Assuming $\overrightarrow{\mathbf{z}_i^{wo}}$ represents the word-granularity attribute vector of user v_i, the word-granularity vectors for all users in G can be represented as the matrix $Z_{wo} = [\overrightarrow{\mathbf{z}_1^{wo}}, \overrightarrow{\mathbf{z}_2^{wo}}, \ldots, \overrightarrow{\mathbf{z}_i^{wo}}, \ldots, \overrightarrow{\mathbf{z}_n^{wo}}] \in \mathbb{R}^{n\times d_{wo}}$.

To address the issue of missing or indistinguishable word-granularity attributes in UA analysis, we enhance user embeddings by incorporating neighboring attribute information. The embedding $\overrightarrow{\mathbf{z}_i^{wo}}$ for each user v_i is updated using a blend of their own attributes and the average attributes of their neighbors, controlled by a tuning parameter $\lambda \in [0, 1]$:

$$\overrightarrow{\mathbf{z}_i^{wo}} = (1-\lambda)\overrightarrow{\mathbf{z}_i^{wo}} + \lambda\frac{1}{s_i}\sum_{j\in\mathcal{N}_i}\overrightarrow{\mathbf{z}_j^{wo}} \tag{4}$$

where, $\mathcal{N}_i = \{v_j \mid (v_i, v_j) \in E\}$ denotes the set of neighbors for user v_i, and $s_i = |N_i|$ is the number of neighbors. In cases where v_i has no word-granularity attributes, the embedding is solely based on the neighbors:

$$\overrightarrow{\mathbf{z}_i^{wo}} = \lambda\frac{1}{s_i}\sum_{j\in\mathcal{N}_i}\overrightarrow{\mathbf{z}_j^{wo}} \tag{5}$$

To mitigate the impact of missing attributes, we modify the approach to use the average of neighbors' embeddings without scaling by λ when word-granularity attributes are absent:

$$\overrightarrow{\mathbf{z}_i^{wo}} = \frac{1}{s_i}\sum_{j\in\mathcal{N}_i}\overrightarrow{\mathbf{z}_j^{wo}} \tag{6}$$

This adjustment ensures that the embeddings remain effective even when user attributes are missing. The process for embedding word-granularity attributes is detailed in Algorithm 2.

Algorithm 2 Words-granularity attribute embedding

Require: Words-granularity attribute set A_{wo} and embedding dimension d_{wo}
Ensure: Words-granularity feature matrix $\mathbf{Z}_{wo}$

function WORDSEMBEDDING(A_{wo}, d_{wo})
  $\mathbf{Z}'_{wo} \leftarrow []$
  corpus $\leftarrow$ buildCorpus(Wikipedia, A_{wo})
  model $\leftarrow$ Word2Vec(corpus, d_{wo})
  for a_i^{wo} **in** A_{wo} **do**
    words $\leftarrow$ wordsTokenize(a_i^{wo})
    $n \leftarrow$ len(**words**)
    $\overrightarrow{\mathbf{z}_i^{wo}} \leftarrow \frac{1}{n}\sum_{i=1}^{n}$ model.get(**words**[i])
    $\mathbf{Z}'_{wo} \leftarrow \mathbf{Z}'_{wo}$.append($\overrightarrow{\mathbf{z}_i^{wo}}$)
  end for
  $\mathbf{Z}_{wo} \leftarrow []$
  for $\overrightarrow{\mathbf{z}_i^{wo}}$ **in** $\mathbf{Z}'_{wo}$ **do**
    $\mathcal{N}_i \leftarrow$ getNeighborhood(v_i)
    $\overrightarrow{\mathbf{z}_i^{wo*}} \leftarrow \mathbf{0}$
    for j **in** $\mathcal{N}_i$ **do**
      $\overrightarrow{\mathbf{z}_i^{wo*}} \leftarrow \overrightarrow{\mathbf{z}_i^{wo*}} + \mathbf{Z}_{wo}[j]$
    end for
    if $\mathbf{z}_i^{wo}$ is not a zero-vector **then**
      $\overrightarrow{\mathbf{z}_i^{wo}} \leftarrow (1-\lambda)\overrightarrow{\mathbf{z}_i^{wo}} + \frac{\lambda}{s_i}\overrightarrow{\mathbf{z}_i^{wo*}}$
    else
      $\overrightarrow{\mathbf{z}_i^{wo}} \leftarrow \frac{1}{s_i}\sum_{j\in\mathcal{N}_i}\overrightarrow{\mathbf{z}_j^{wo}}$
    end if
    $\mathbf{Z}_{wo} \leftarrow \mathbf{Z}_{wo}$.append($\overrightarrow{\mathbf{z}_i^{wo}}$)
  end for
  return $\mathbf{Z}_{wo}$
end function

5.2.3. Article-granularity attribute embedding

Article-granularity attributes, such as user-generated blogs, reviews, and publications, are rich in individual perspectives and expertise. Traditional similarity matching and word-granularity analysis are inadequate for processing such complex freeform text data. Our approach focuses on extracting advanced semantic features to identify consistent patterns in users' content distribution across different networks. It is crucial to identify uniform patterns in how users' article-granularity attributes are semantically categorized in UA tasks. Users often share content related to their interests, like sports enthusiasts posting sports-related content on multiple platforms, or academics sharing publications in their research areas across academic networks. Analyzing these semantic category distributions can reveal meaningful correlations that transcend individual platforms.

Although unsupervised methods such as Probabilistic Latent Semantic Analysis (PLSA) (Hofmann, 2013), Correlated Topic Model (CTM) (Xun, Li, Zhao, Gao, & Zhang, 2017), and Latent Dirichlet Allocation (LDA) (Kim, Seo, Cho, & Kang, 2019) apply to text classification, contemporary deep pre-trained language models surpass these methods in capturing semantic nuances. Specifically, we use chinese-electra model for Chinese text and electra-base-discriminator model for English text, both sourced from the huggingface repository. These advanced models excel in semantic encoding and are benchmarked against LDA in our experiments to demonstrate their efficacy in mapping article-granularity attributes to category distributions.

To model article-granularity attributes in a social network G, we define $A_{ar} = \{a_1^{ar}, a_2^{ar}, \ldots, a_i^{ar}, \ldots, a_n^{ar}\}$ as the set of such attributes, with a_i^{ar} representing the article attribute for user v_i. The matrix $Z_{ar} = [\overrightarrow{\mathbf{z}_1^{ar}}, \overrightarrow{\mathbf{z}_2^{ar}}, \ldots, \overrightarrow{\mathbf{z}_i^{ar}}, \ldots, \overrightarrow{\mathbf{z}_n^{ar}}] \in \mathbb{R}^{n\times d_{ar}}$, consisting of the vectors $\overrightarrow{\mathbf{z}_i^{ar}}$, represents the article-granularity embeddings for all users in $\mathbb{R}^{n\times d_{ar}}$, where d_{ar} is the embedding dimension. Each $\overrightarrow{\mathbf{z}_i^{ar}}$ captures the semantic features of the corresponding article text, extracted using a pre-trained language model.

5.2.4. Label-granularity attribute embedding

Prior research (Chen & Chen, 2022) indicates that UA accuracy varies across platforms, with social networks like Facebook and Weibo showing lower accuracy compared to academic sites like DBLP, due to the less reliable nature of user-generated content. For example, user-generated content such as usernames and geolocations often lacks veracity. However, device-specific labels or identifiers appended to user posts, such as "Posted from iPhone" or "Posted from Web" are consistent and reliable. Users seldom conceal or alter their device identifiers and typically do not switch devices frequently within a short timeframe. These label-granularity attributes are useful for UA tasks and can be embedded using techniques similar to those for character-granularity attributes, resulting in a feature matrix $Z_{\text{la}} \in \mathbb{R}^{(d_{\text{la}}\times n)}$. As label-granularity attributes offer distinct and complementary alignment signals, isolating label embeddings enables the model to appropriately weigh their contributions, enhancing the alignment process without conflating different attribute signals.

5.2.5. Relationship network structure embedding

The relational network of users contains valuable information about social ties, interests, and geographic locations. For example, the same natural person usually has a similar circle of friends on different social media platforms. Structural similarities within these networks serve as robust indicators for the alignment of user accounts across different platforms.

The inherent graph-based structure of social networks necessitates formulating network embedding as a graph representation learning problem. We employ the Node2Vec algorithm (Grover & Leskovec, 2016) to learn low-dimensional node embeddings that reflect the network's structure. Node2Vec utilizes random walks to explore neighborhoods, then leverages a Word2Vec-style framework to embed nodes into a continuous vector space based on co-occurrence. A key benefit is the tunable walk randomness that trades off between breadth-first and depth-first searches, allowing customizable encapsulation of both structural equivalences and homophily. The obtained node embeddings have demonstrated effectiveness in node classification, link prediction, visualization, and other graph analytics tasks, by encoding useful semantic and contextual information. Node2Vec provides a flexible, scalable graph embedding framework suited to represent the rich connectivity patterns in social networks.

We construct a feature matrix $Z_{\text{st}} \in \mathbb{R}^{(d_{\text{st}} \times n)}$ from Node2Vec embeddings to represent the network structure, where n is the number of users and d_{st} is the embedding dimensionality. Each row in Z_{st} is a Node2Vec feature vector that encodes a user's network context, providing a structured representation to aid in modeling user relationships.

5.2.6. Feature integration

Having completed the embedding process for attributes across four specified granularities and the network structure, we have obtained a quintet of feature vectors for each user. These vectors, derived from four distinct embedding methodologies, correspond to the following granularities: character ($\overrightarrow{\mathbf{z}^{ch}}$), word ($\overrightarrow{\mathbf{z}^{wo}}$), article ($\overrightarrow{\mathbf{z}^{ar}}$), label ($\overrightarrow{\mathbf{z}^{la}}$), and structural ($\overrightarrow{\mathbf{z}^{st}}$). The aggregation of these vectors for all users within the network yields matrices Z_{ch}, Z_{wo}, Z_{ar}, Z_{la}, and Z_{st}, respectively. For analytical expediency, we denote the concatenated feature vectors of an individual user as $\overline{\mathbf{z}}$, and the comprehensive matrix representing the concatenated vectors of all users as $Z = \{\vec{x}_1, \ldots, \vec{x}_n\} \in \mathbb{R}^{(n \times d)}$, where d is the sum of the dimensions of the individual attribute and structural embeddings, expressed as $d = d_{ch} + d_{wo} + d_{ar} + d_{la} + d_{st}$.

5.3. Construction of multi-granularity attribute cosine distance distribution vector(MA-CDDV)

For any pair of users $(v_i^{\alpha}, v_j^{\beta})$ from social networks G^{α} and G^{β}, assuming $\overrightarrow{\mathbf{z}_{\alpha}}$ and $\overrightarrow{\mathbf{z}_{\beta}}$ respectively represent feature vectors of $(v_i^{\alpha}, v_j^{\beta})$ at a certain granularity (determined by the superscript), the function $CosDist(\overrightarrow{\mathbf{z}_{\alpha}}, \overrightarrow{\mathbf{z}_{\beta}})$ calculates the Cosine distance between the two vectors. The construction process of the MA-CDDV for the user pair $(v_i^{\alpha}, v_j^{\beta})$ is as follows:

$$\begin{aligned}\text{MA-CDDV} = \Big[& CosDist(\overrightarrow{\mathbf{z}_{\alpha}^{ch}}, \overrightarrow{\mathbf{z}_{\beta}^{ch}}) \\ , & CosDist(\overrightarrow{\mathbf{z}_{\alpha}^{wo}}, \overrightarrow{\mathbf{z}_{\beta}^{wo}}) \\ , & CosDist(\overrightarrow{\mathbf{z}_{\alpha}^{ar}}, \overrightarrow{\mathbf{z}_{\beta}^{ar}}) \\ , & CosDist(\overrightarrow{\mathbf{z}_{\alpha}^{la}}, \overrightarrow{\mathbf{z}_{\beta}^{la}}) \\ , & CosDist(\overrightarrow{\mathbf{z}_{\alpha}^{st}}, \overrightarrow{\mathbf{z}_{\beta}^{st}}) \Big]^{\mathsf{T}} \in \mathbb{R}^{n \times 1}\end{aligned} \tag{7}$$

The formula for calculating cosine distance is as follows:

$$CosDist = 1 - \frac{\sum_{i=1}^{d} \overrightarrow{\mathbf{z}_{\alpha_i}} \cdot \overrightarrow{\mathbf{z}_{\beta_i}}}{\sqrt{\sum_{i=1}^{d} \overrightarrow{\mathbf{z}_{\alpha_i}}^2} \cdot \sqrt{\sum_{i=1}^{d} \overrightarrow{\mathbf{z}_{\beta_i}}^2}} \tag{8}$$

The training dataset for the similarity model includes N_{tr} positive sample MA-CDDVs representing aligned pairs and $\frac{N_{tr}}{2}$ negative sample MA-CDDVs representing non-aligned pairs.

(1) Positive Samples:The MA-CDDV constructed from the feature vectors of pre-aligned user pairs, are denoted as E^p.

(2) Negative Samples: Negative samples consist of MA-CDDVs constructed from the feature vectors of non-aligned user pairs. To generate non-aligned user pairs and minimize the chance of mistakenly selecting true matches as negative samples, we employ the offset perturbation method. For each confirmed aligned user pair $(v_i^{\alpha}, v_j^{\beta})$ from networks G^{α} and G^{β}, where i and j represent their index positions in their respective social networks, we fix i and perturb j by an offset. This yields a new user pair $(v_i^{\alpha}, v_{j+\text{offset}}^{\beta})$. If the new index is within the valid user range of the network, $(v_i^{\alpha}, v_{j+\text{offset}}^{\beta})$ is considered a non-aligned pair. This process is repeated to generate N_{ne} negative sample MA-CDDVs, represented as the set E^n.

MA-CDDV quantifies the correlation between pre-aligned user pairs in the same granularity attributes and the distribution characteristics of correlation among different granularity attributes. It has extremely low dimensionality, equal to the number of divisions in granularity. Training a model using MA-CDDV can effectively reduce the learning burden of the model, achieving fast convergence. MA-CDDV benefits from its design of first quantifying and then concatenating, which provides excellent scalability. The dimensions of the vector can be dynamically adjusted based on the richness or reduction of known information. Additionally, MA-CDDV can freely choose quantification functions as needed, including but not limited to Euclidean distance, Manhattan distance, Hamming distance, and Wasserstein distance. For ease of subsequent description, we use the function $granCosdist(\overrightarrow{\mathbf{z}^{\alpha}}, \overrightarrow{\mathbf{z}^{\beta}})$ to represent the construction process of MA-CDDV.

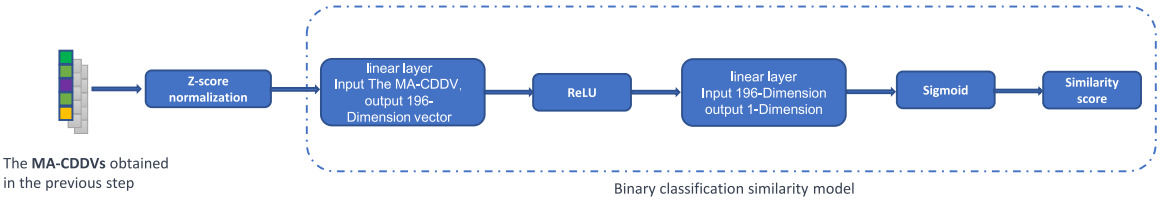


Fig. 4. The architecture of the binary classification similarity model begins by applying z-score normalization to the MA-CDDVs, which are then fed into a linear layer with an output dimension of 196. After passing through a ReLU activation function, the output is passed to another linear layer with a single output dimension. Finally, the similarity of the MA-CDDVs is obtained through a Sigmoid function.

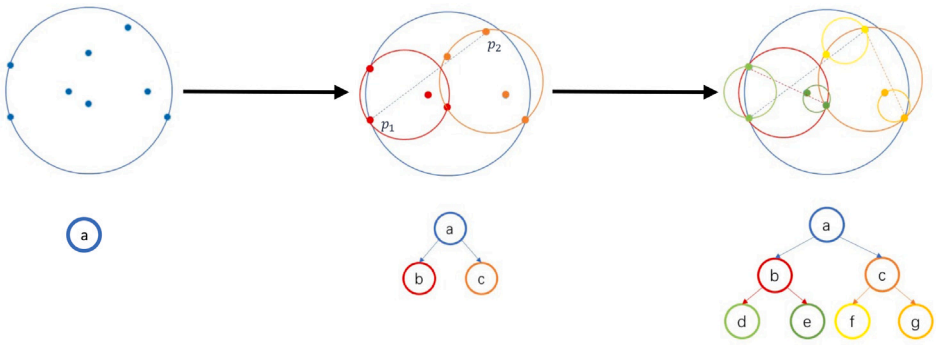


Fig. 5. An example of constructing a BallTree. First, create a minimal bounding hypersphere to contain all data points, serving as the root node a. Next, find the two farthest points p_1 and p_2 within the initial hypersphere, and divide the dataset into two new bounding hyperspheres based on the distances to p_1 and p_2, forming child nodes b and c. Finally, recursively partition each child node until the number of points within each hypersphere reaches or falls below the specified threshold.

5.4. Binary classification similarity model

5.4.1. Model design

The lack of extensive ground truth data is a significant challenge in UA research, making model performance in data sparse settings a key evaluation metric. Our previous work has successfully utilized unsupervised learning to derive user feature representations, using Cosine distance distribution vectors across multi-granularity attributes to measure inter-attribute relationships at different scales. These comprehensive and effective preliminary steps can significantly reduce the model's learning requirements. The model needs to discern patterns in the attribute associations to assess the likelihood that two user profiles correspond to the same individual. This method highlights the potential for an efficient predictive framework for UA in contexts with limited resources.

We have introduced a lightweight binary classification similarity model, consisting of a series of linear layers and activation functions, as illustrated in Fig. 4. The architecture aims to identify distribution patterns of correlations between same-granularity attributes in the MA-CDDV, without the need for complex feature engineering. Although it autonomously learns the importance of different user attributes from the data on each social platform, avoiding reliance on preset formulas. Our empirical results indicate that this approach effectively integrates cross-network alignment signals with adaptable similarity functions, achieving top-tier results in user alignment tasks.

5.4.2. Prediction

To improve UA prediction, we introduce a BallTree (Dolatshah et al., 2015) data structure, which is adept at managing distance-based queries within large datasets. BallTree organizes data by recursively dividing the metric space into nested hyper-spheres, enabling rapid nearest-neighbor searches and reducing query times significantly compared to linear scans, particularly in high-dimensional spaces.

Fig. 5 illustrates the BallTree construction process, which starts by enclosing all data points in a single hyper-sphere, then recursively partitions the space into smaller hyper-spheres until each contains a manageable number of points. The BallTree construction process of the example for organizing data points in a metric space involves three main steps:

- **Initialization**: A minimum bounding hypersphere is created to encompass all data points, establishing the root node of the BallTree (node a).
- **Partitioning**: The two farthest points within the initial hypersphere, p_1 and p_2, are identified. The dataset is then partitioned by calculating the distances from all other points to p_1 and p_2 using a distance metric (typically Cosine distance). This results in two new bounding hyperspheres, each containing points closer to either p_1 or p_2, forming the child nodes b and c of the root.

- **Recursive Division**: The partitioning step is recursively applied to each child node, continuing until the number of points in a hypersphere is at or below a set threshold (often 40 points) for computational efficiency.

Each node of the BallTree corresponds to a hypersphere defined by spatial parameters: center, radius, and the indices of the data points it contains. Nodes also have pointers to their left and right child subtrees, enabling efficient traversal and nearest neighbor searches within the tree's multidimensional space.

In this study, we address the challenge of efficiently organizing and indexing user data for UA across social networks by constructing a BallTree. We represent the attribute embeddings of users in networks G^{α} and G^{β} as $Z^{\alpha} = \{\overrightarrow{\mathbf{z}_1^{\alpha}}, \overrightarrow{\mathbf{z}_2^{\alpha}}, \ldots, \overrightarrow{\mathbf{z}_{n_\alpha}^{\alpha}}\}$ and $Z^{\beta} = \{\overrightarrow{\mathbf{z}_1^{\beta}}, \overrightarrow{\mathbf{z}_2^{\beta}}, \ldots, \overrightarrow{\mathbf{z}_{n_\beta}^{\beta}}\}$, respectively, where n_α and n_β denote the number of users in each network. Z^{α} is applied as the base for the BallTree construction. After building the BallTree, we create a matrix $M_d \in \mathbb{R}^{(n_\beta \times n_\alpha)}$ to store the distances between user pairs, initially filled with infinity. By iteratively taking $\mathbf{z}_i^{\beta}$ from Z^{β} and finding the top k nearest users from Z^{α} within the BallTree, we populate matrix M_d with the computed distances, forming the distance matrix. The distance metric function, which is used for both constructing and querying the BallTree, is formally defined:

$$distance = Exp(-model(granCosdist(\overrightarrow{\mathbf{z}_i}, \overrightarrow{\mathbf{z}_j}))) \tag{9}$$

In this formular, $\overrightarrow{\mathbf{z}_i}$ and $\overrightarrow{\mathbf{z}_j}$ denote the feature vectors of two users. $model$ is the model described in Section 5.4.1, and the $granCosdist()$ function is used to compute the MA-CDDV.

Analysis of user behaviors on two social platforms, G^{α} and G^{β}, revealed a notable pattern: the presence of a user's character-granularity attributes $A_c(v_i^{\alpha})$ in another user's article-granularity attributes $A_a(v_j^{\beta})$ increases the probability that both attributes pertain to the same individual. This correlation is likely due to users promoting their presence across platforms within their content and the persistence of character-granularity attributes, such as professional or personal interests, across related domains. For instance, a username referencing a specific cartoon character suggests a consistent interest that may be reflected in the user's content on other platforms, making such attributes valuable for user alignment detection. Such reappearance phenomenon of character-granularity attributes between two or more user accounts is termed **Attribute Reappearance (AR)** in our study.

To discern genuine connections, we set a threshold of 10 mentions to exclude the impact of high-frequency popular terms. We introduce a correction coefficient λ to adjust the similarity scores for these user pairs, with λ assigned a value greater than 1 (indicating a higher match likelihood). We use the following formula, which includes λ, to transform the distances in the distance matrix M_d into similarities, thereby converting M_d into a similarity matrix M_s.

$$M_s = \lambda \cdot Exp(-distance) \tag{10}$$

For clarity in further discussions, we refer to the process of adjusting similarity scores based on attribute reappearance as Attribute Reappearance Score Correction (ARSC), with λ representing the correction coefficient.

The effectiveness of the "ARSC" technique in enhancing UA is confirmed through empirical testing on real-world datasets, details of which are forthcoming in the evaluation section. The training of the similarity model and the development of the custom distance function, which incorporates corrections for AR, are outlined in Algorithm 3 and Algorithm 4, respectively. These algorithms detail the process of initially learning a classifier based on matched user embeddings and subsequently refining the similarity measurement to account for the unique characteristics of academic and social platforms by adjusting the λ coefficient.

The effectiveness of the "ARSC" technique in enhancing UA has been confirmed through empirical testing on real-world datasets, with specific details to be presented in the evaluation section. The training of the similarity model and the development of the custom distance function incorporating ARSC are outlined in Algorithm 3 and Algorithm 4.

Algorithm 3 Training of the Multi-granularity attribute Cosine distance similarity model

Require: The user feature matrix Z, The size of the training set N_{tr}.
Ensure: Similarity model based on the MA-CDDVs.

```
function getSimilarityModel(Z, N_tr)
    userIndexTupleArray ← getPositivSamples(N_tr)
    Initialize E^p ←[]
    Initialize E^n ←[]
    for each (i, j) in userIndexTupleArray do
        e^p ← granCosdist(Z[i], Z[j])
        E^p.append(e^p)
        if len(E^n) < N_tr/2 then
            j ← j + offset
            e^n ← granCosdist(Z[i], Z[j])
            E^n.append(e^n)
        end if
    end for
    model ← SimilarityModel()
    repeat
        train(model, E^p, E^n)
    until Convergence
    return model
end function
```

Algorithm 4 BallTree CustomMetric

Require: Two user feature vectors, $\overrightarrow{\mathbf{z}^{\alpha}}$ and $\overrightarrow{\mathbf{z}^{\beta}}$, each originating from two social networks G^{α} and G^{β}, The similarity model obtained by Algorithm 3.

Ensure: The distance between feature vectors based on similarity.

1: **function** CustomMetric($\overrightarrow{\mathbf{z}^{\alpha}}$, $\overrightarrow{\mathbf{z}^{\beta}}$, model)
2: $\vec{e} \leftarrow$ granCosdist($\overrightarrow{\mathbf{z}^{\alpha}}, \overrightarrow{\mathbf{z}^{\beta}}$)
3: distance $\leftarrow$ **model**($\vec{e}$)
4: **return** distance
5: **end function**

Table 2
The statistics of the datasets used in the experiments.

Datasets	Networks	#Users	#Relations	Min. degree	Ave. degree	Max. degree	Ave. coeff	#Matched pairs
Social networks	Weibo	9714	117,218	2	12.1	607	0.112	1397
	Douban	9526	120,245	2	12.6	608	0.101	
Co-authorship networks	DBLP17	9086	51,700	2	5.7	144	0.28	2832
	DBLP19	9325	47,775	2	5.1	138	0.322	

6. Experiments

This section systematically examines the datasets, baseline models, experimental configurations, and analysis pertinent to validating the proposed approach. It encompasses both comparative experiments and ablation studies.

6.1. Datasets

To validate the proposed Multi-Granularity Attribute Similarity Model and compare it with existing methods, we utilize two real-world datasets as detailed by Yang et al. (2022): one from social networks and another from co-authorship networks.

Social Networks: The Weibo-Douban (WD) dataset is derived from two popular Chinese platforms: Sina Weibo and Douban. Sina Weibo is a microblogging service akin to Twitter, where users engage in social interactions through multimedia content. Douban, on the other hand, focuses on cultural content sharing and discussions. The WD dataset is enriched by the diverse user interactions and content types on these platforms. A subset of Douban users who publicly link their Sina Weibo profiles provides a ground truth for pre-aligned users, aiding in the integration of the two networks for analysis.

Co-authorship Networks: The Digital Bibliography & Library Project (DBLP) dataset is a well-known co-authorship network within the computer science community, cataloging bibliographic information on academic publications. The DBLP network, with its unique author identifiers, facilitates the identification of pre-aligned users. For this study, we use snapshots of the DBLP network from December 1, 2017, and December 1, 2018, as target networks for alignment. Table 2 provides detailed statistics for both datasets.

6.2. Experimental settings

This section outlines the models included in the comparative experiments, the evaluation metrics used, the hardware specifications of the experimental setup, and the parameter configurations for each model.

6.2.1. Baseline and other methods

MGASM is compared against a range of established baseline methods and state-of-the-art models to ensure a thorough evaluation of its performance:

- **NSBVUIL** (Li et al., 2023): NSVUIL is a semi-supervised social network user matching framework that employs a hierarchical attention mechanism, simultaneously considering user attributes and structural information.
- **DeepDSA** (Matrouk, Srikanth, Kumar, Bhadla, Sabirov, & Saadh, 2023): DeepDSA primarily focuses on user structure. It transforms social network data into sequential input for a Transformer to address the oversmoothing problem in Graph Neural Networks (GNNs). Additionally, it enhances matching accuracy by assigning weights to different users and network structures.
- **GradAlign+** (Park et al., 2022b): The GradAlign+ method gradually discovers node pairs by computing similarities between nodes. It builds upon GradAlign (Park, Tran, Shin, & Cao, 2022a) and introduces node attribute augmentation, improving the model's robustness.
- **NeXtAlign** (Zhang et al., 2021): A superior semi-supervised network alignment method, which employs a novel sampling approach capable of handling alignment discrepancies during training, thereby effectively distinguishing between correct alignments and misleading ones. Due to the differences in dataset structures, NeXtAlign in this paper only considers the network structure of the users.

- **JARUA** (Yang et al., 2022): JARUA is a semi-supervised framework that introduces the concept of subwords while modeling multi-level attributes and uses a graph attention network for alignment. JARUA performs the best on the dataset used in this paper, and thus will serve as the primary benchmark for this study.

Two variants of MGASM are tested to evaluate the impact of different structures on performance, each variant utilizes ARSC:

- **MGASM_LDA**: This variant uses Latent Dirichlet Allocation (LDA) for article-granularity attribute embedding to compare the performance against pre-trained language models in article feature extraction.
- **MGASM_NL**: By excluding label-granularity attribute embedding, this variant assesses the impact of label-granularity attributes on the overall performance of MGASM.

6.2.2. Evaluation metric

The study utilizes precision, recall, and F1-score as standard metrics to evaluate the performance of the models. These metrics are widely accepted in the field for assessing classification models. **Precision** measures the accuracy of positive predictions, indicating the proportion of true positives among all predicted positives. **Recall** measures the model's ability to identify all relevant instances, indicating the proportion of true positives among actual positives. **F1** is the harmonic mean of precision and recall, providing a balance between the two metrics.

For the specific task of UA, the study also employs the hit-precision metric (Mu, Zhu, Lim, Xiao, Wang, & Zhou, 2016), which is suitable for ranking tasks where the goal is to find the correct match within the top k candidates. The formula for hit-precision is:

$$h(x) = \frac{k - (hit(x) - 1)}{k} \tag{11}$$

In this formula, $hit(x)$ denotes the rank position where the correct match is found within the top k candidates. The overall hit-precision is the average of individual scores for all successfully matched pairs:

$$\text{Hit-Precision} = \frac{1}{n}\sum_{i=1}^{n} h(x_i) \tag{12}$$

Typically, k is set to 3 for calculating hit-precision, unless otherwise specified. This metric provides insight into the model's effectiveness at ranking true matches highly, which is essential for the practical application of UA in real-world scenarios.

6.2.3. Hardware devices and environment

The computational experiments conducted in this study are implemented using the Python 3.7 programming language. The execution environment is based on a Windows 11 operating system. The hardware configuration includes a 12th Gen Intel(R) Core(TM) i9-12900H processor, which provides robust computational capabilities. Additionally, the system is equipped with an NVIDIA GeForce RTX 3080 Ti laptop GPU, featuring 8 GB of dedicated memory to facilitate efficient processing of machine learning tasks.

6.2.4. Parameter configuration

The parameter configuration for the MGASM model is carefully designed to ensure consistency and reliability in the experimental results. The key settings are represented as follows:

- **Embedding Dimensions:** The embedding dimensions for the various granularity attributes are uniformly set to $d_{ch} = d_{wo} = d_{ar} = d_{st} = d_{la} = 100$, ensuring consistency across different attribute types.
- **Training and Testing Sets:** The number of matched user pairs for training (N_{tr}) and testing (N_{te}) are determined through a random selection process. The range for N_{tr} is between 5% and 30% of the total training set, while N_{te} is set at 500 pairs for evaluating hit-precision. For precision, recall, and F1 score assessments, N_{te} consists of 250 positive samples and an additional 250 randomly generated non-matched user pairs (negative samples) for a balanced evaluation. To maintain consistency across all models, N_{tr} is standardized at 20% of the total training set. It is critical to keep the training and testing datasets strictly separate to avoid data leakage and ensure the validity of the evaluation.
- **Repetition of Experiments:** Each experimental run is conducted 10 times independently to account for variability in the results. The average of these runs is used for performance analysis, providing a more robust and reliable measure of the model's effectiveness.
- **Baseline Model Parameters:** The parameter configurations for the baseline models used in the comparative analysis are taken from the default settings reported in their original publications. This approach ensures that the comparison is fair and that the baseline models are evaluated under conditions recommended by their creators.

These parameter settings are essential for replicating the experiments and for understanding the context in which the MGASM model is evaluated. By adhering to these configurations, the study aims to provide a clear and accurate comparison of MGASM's performance against other models in the field.

6.3. Experimental analysis

This section provides a detailed comparison of the performance of the MGASM and its variants against established baseline models on two real-world datasets: WD and DBLPs.

Table 3
Precision of various UA methods on WD and DBLP datasets at different values of k.

Method	Weibo-Douban			DBLP17-DBLP19		
	k = 1	k = 3	k = 5	k = 1	k = 3	k = 5
NSVUIL	0.284	0.326	0.373	0.416 1	0.458	0.504
DeepDSA	0.271	0.299	0.332	0.541	0.601	0.667
Grad-Align+	0.299	0.302	0.305	0.288	0.304	0.316
JARUA	0.416	0.448	0.464	0.821	0.845	0.858
NeXtAlign	0.183	0.247	0.282	0.377	0.530	0.602
MGASM	**0.647**	**0.718**	**0.760**	0.683	0.718	0.733
MGASM_NL	0.475	0.513	0.538	0.681	0.707	0.719
MGASM_LDA	0.641	0.716	0.758	**0.882**	**0.912**	**0.928**

Table 4
Precision, Recall, and F1 scores of different UA methods on Weibo-Douban and DBLP17-DBLP19 datasets at various values of k.

Method	Weibo-Douban			DBLP17-DBLP19		
	Pre.	Rec.	F1	Pre.	Rec.	F1
NSBVUIL	31.14%	75.25%	44.05%	53.77%	78.01%	63.66%
DeepDSA	41.12%	67.47%	51.10%	63.63%	71.11%	67.16%
Grad-Align+	26.12%	82.47%	39.67%	43.36%	70.21%	53.61%
NeXtAlign	25.92%	81.14%	39.29%	42.10%	72.31%	53.22%
JARUA	41.23%	**87.94%**	56.14%	68.07%	**88.29%**	76.87%
MGASM	**100%**	62.93%	**77.25%**	99.61%	68.53%	81.20%
MGASM NL	99.39%	44.27%	61.26%	99.42%	67.53%	80.43%
MGASM LDA	**100%**	60.80%	75.62%	**99.85%**	87.60%	**93.32%**

6.3.1. Overall performance

The results summarized in Table 3 and Table 4 indicate that MGASM and its variants outperform the baseline models on the majority of metrics, showing significant improvement compared to the previously considered state-of-the-art JARUA method. The key observations are as follows:

- **Dataset Quality:** The quality of the datasets has a significant impact on the performance of the models. Earlier methods struggled with social network datasets, but MGASM has managed to reduce this performance gap.
- **Platform Characteristics:** Academic and social platforms have different features, and a single mechanism may not necessarily be applicable across all platforms.
- **Benchmarking:** JARUA serves as the primary benchmark due to its prior success, providing a reference point for evaluating the improvements made by MGASM.
- **Performance Gap:** While the JARUA method was the most effective among earlier approaches, MGASM has shown to be more effective, which is evident from the comparative results.

The tables referenced (Table 3 and Table 4) would typically contain detailed performance metrics such as hit-precision, precision, recall, and F1-score, allowing for a quantitative assessment of each model's ability to correctly align user identities across different platforms.

Table 3 highlights the hit-precision of MGASM on two datasets. On the WD dataset, MGASM achieves a hit-precision of 0.718 at the top-candidates parameter $k = 3$, surpassing JARUA's 0.427 by 68.15%. MGASM_LDA slightly underperforms compared to MGASM, indicating that pre-trained language models may better capture article topics in social media contexts compared to LDA. MGASM_NL's score decreases by at least 28.55% compared to MGASM, demonstrating the positive impact of introducing label granularity on the user alignment task. Despite this, MGASM_NL still outperforms JARUA by 20.14% in hit-precision, reaching 0.513, confirming the advanced nature of the MGASM architecture.

Contrasting its performance with that in the WD dataset, MGASM_LDA outperforms MGASM in the DBLP dataset. MGASM_LDA achieving a hit-precision of 0.912 at $k = 3$. This represents a 20.95% improvement over the JARUA benchmark's 0.754 and a 27.02% improvement over the base MGASM. This performance difference can be attributed to the unique composition of article-granularity attributes within the DBLP datasets, which primarily consist of article titles rather than comprehensive article content. Academic article titles are often laden with specialized terminology, including a plethora of proper nouns and abbreviations, making it challenging to accurately predict topic distributions unless pre-trained models are specifically fine-tuned for such tasks. Here, the statistical learning foundation of the LDA method proves more effective than pre-trained models not fine-tuned for academic terminology.

MGASM and its variants outperform existing methods in hit-precision under the majority of conditions. Notably, the improvements of MGASM are more significant on social media platforms than on academic platforms, demonstrating its potential in handling non-standardized data.

MGASM and its variants outperform existing methods in hit-precision in most cases. Specifically, compared to DeepDSA and NeXtAlign, which emphasize user network structural features, MGASM demonstrates remarkable unique advantages in handling

Table 5
The impact of ARSC on hit-precision for MGASM and its variants across different datasets. The table illustrates the hit-precision performance before using ARSC (BC), after using ARSC (AC), and the improvement (inc) achieved by incorporating ARSC.

Method	Weibo-Douban								
	$k = 1$			$k = 3$			$k = 5$		
	BC	AC	inc.(%)	BC	AC	inc.(%)	BC	AC	inc.(%)
MGASM	0.614	0.647	5.33%	0.685	0.718	4.81%	0.729	0.760	4.17%
MGASM_NL	0.441	0.475	7.70%	0.490	0.513	4.72%	0.519	0.538	3.54%
MGASM_LDA	0.595	0.641	7.73%	0.676	0.716	5.98%	0.725	0.758	4.48%
Method	DBLP17-DBLP19								
	$k = 1$			$k = 3$			$k = 5$		
	BC	AC	inc.(%)	BC	AC	inc.(%)	BC	AC	inc.(%)
MGASM	0.679	0.683	0.68%	0.699	0.718	2.70%	0.709	0.733	3.34%
MGASM_NL	0.667	0.681	2.05%	0.697	0.707	1.42%	0.711	0.719	1.17%
MGASM_LDA	0.859	0.882	2.64%	0.893	0.912	2.09%	0.914	0.928	1.62%

Table 6
The impact of ARSC on Precision, Recall, and F1 scores for MGASM and its variants across different datasets. Here, BC represents before using ARSC, AC represents after using ARSC, and inc indicates the improvement after using ARSC compared to before its use.

Method	Weibo-Douban								
	Pre.			Rec.			F1		
	BC	AC	inc.(%)	BC	AC	inc.(%)	BC	AC	inc.(%)
MGASM	0.998	1.000	0.22%	0.620	0.629	1.50%	0.765	0.772	1.01%
MGASM_NL	0.994	0.994	0.00%	0.437	0.443	1.23%	0.607	0.613	0.85%
MGASM_LDA	1.000	1.000	0.00%	0.601	0.608	1.16%	0.751	0.756	0.72%
Method	DBLP17-DBLP19								
	Pre.			Rec.			F1		
	BC	AC	inc.(%)	BC	AC	inc.(%)	BC	AC	inc.(%)
MGASM	0.992	0.996	0.41%	0.681	0.685	0.59%	0.808	0.812	0.52%
MGASM_NL	0.992	0.994	0.22%	0.672	0.675	0.49%	0.801	0.804	0.38%
MGASM_LDA	1.000	0.999	−0.15%	0.871	0.876	0.54%	0.931	0.933	0.22%

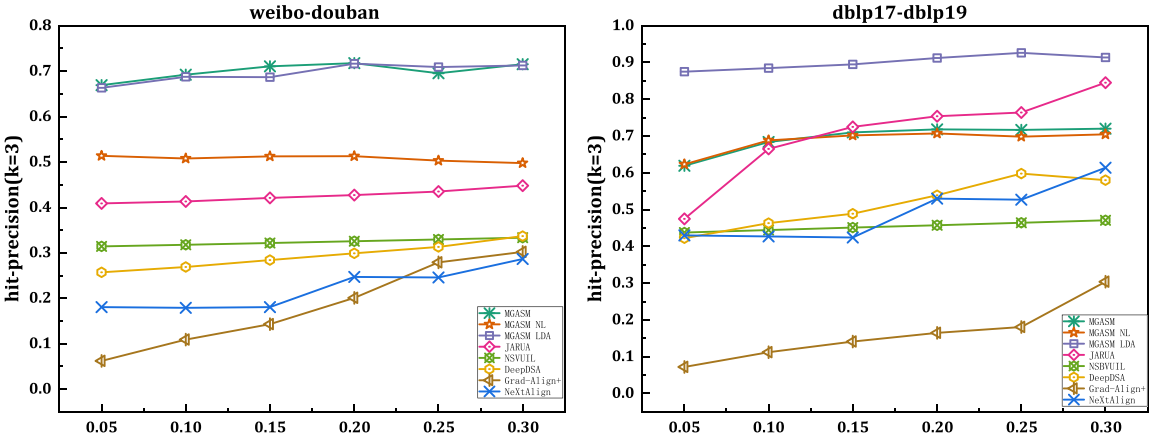
network topology. However, these methods lack targeted processing of user attributes and user-generated content on content-rich social platforms. Compared to the classic methods GradAlign+ and NSVUIL, which also consider user attributes and structural information, GradAlign+ requires a higher amount of training data, limiting its application in scenarios with scarce pre-aligned users. The NSVUIL algorithm is highly robust, with minimal impact from the quantity and quality of the training set, allowing it to perform steadily in complex scenarios. However, it falls short in capturing fine-grained user features. Notably, MGASM shows significantly greater improvements on social media platforms compared to academic platforms, demonstrating its potential in handling non-standardized data.

Table 4 provides a detailed overview of MGASM's performance across three evaluation metrics: precision, recall, and F1-score. Across all datasets, MGASM and its variants demonstrate exceptional precision, with the main model MGASM achieving precision levels of 100% and 99.96% on the WD and DBLPs datasets, respectively. Compared to JARUA's 41.23% and 68.07%, this represents a significant increase of 142.54% and 46.85%, respectively, indicating MGASM's extremely high accuracy in positive identifications. This is particularly promising for UA tasks that require high precision. Conversely, the recall of MGASM and its variants is slightly lower, possibly due to their methodology of using a single user as a reference and matching attribute distributions based on cosine distances. In complex social network environments, this may result in true positives being misclassified as negatives, where a pair of non-aligned users may exhibit a closer similarity to the feature of interest than the actual aligned users. JARUA and Grad-Align+ performed exceptionally well on this metric, making them highly promising for scenarios where sensitivity to user alignment miss rates is critical. MGASM demonstrated the best overall performance, as reflected in its F1 scores, which reached 77.25% on the WD dataset and 81.20% on the DBLPs dataset. Compared to JARUA's scores of 56.14% and 76.87%, these represent improvements of 37.6% and 5.63%, respectively.

Table 5 and Table 6 compare the performance metrics of MGASM and its variants before (BC) and after (AC) using ARSC across two datasets, Table 7 summarizes the average improvements on each metric after using ARSC. It can be observed that ARSC has varying degrees of promoting effect on hit-precision, precision, recall, and F1 scores. Specifically, in the WD dataset, using ARSC at $k = 3$ resulted in an average increase of 5.17% in hit-precision, and an average increase of 0.07%, 1.3%, and 0.86% in precision, recall, and F1 scores, respectively. The impact of ARSC on precision, recall, and F1 scores mirrors the trend in hit-precision but with slightly smaller magnitudes. The situation in the DBLP dataset is similar to that of the WD dataset, with an average increase of 2.07% in hit-precision, and an average increase of 0.16%, 0.54%, and 0.37% in precision, recall, and F1 scores, respectively. This underscores the importance of context-specific application of ARSC for optimizing user alignment methods.

Table 7
The average impact of using ARSC on hit-precision, precision, recall, and F1 scores for MGASM across different datasets.

Dataset	Hit-precision (k = 3)	Pre. (%)	Rec.	F1
WD	5.17%	0.07%	1.30%	0.86%
DBLPs	2.07%	0.16%	0.54%	0.37%



(a) In the WD dataset, the hit-precision performance of all models when N_{tr} varies from 5% to 30% (k=3).

(b) In the DBLPs dataset, the hit-precision performance of all models when N_{tr} varies from 5% to 30% (k=3).

Fig. 6. The hit-precision performance of the MGASM variant considering only a single attribute granularity as the training dataset size N_{tr} varies from 5% to 30%.

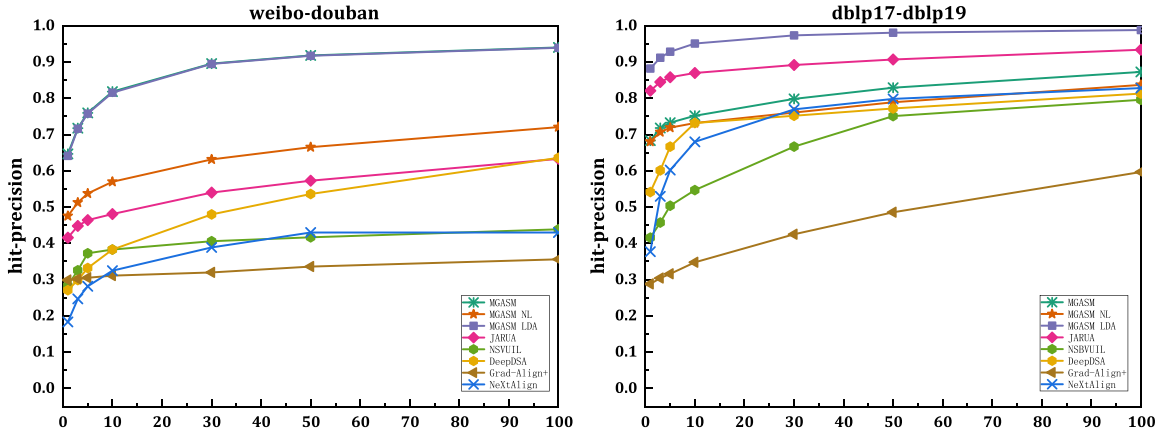
6.3.2. Influence of the training dataset size on the results

Fig. 6 presents the performance of all models as the training dataset size (N_{tr}) increases from 5% to 30%. In the WD dataset (see Fig. 10(a)), MGASM and its variants demonstrate rapid convergence, with MGASM achieving a hit-precision of 0.718 with only 5% of the training set. This represents a remarkable improvement of 68.15% compared to JARUA's hit-precision of 0.427. However, as N_{tr} increases, MGASM and its variants begin to oscillate around the maximum value after $N_{tr} > 20\%$, with no further performance improvement. In contrast, other models continue to show improvement, albeit at a slower rate. Even at $N_{tr} = 30\%$, MGASM maintains superior performance, with a hit-precision 59.6% higher than JARUA's.

In parallel experiments on the DBLP dataset (see Fig. 10(b)), all models perform better with relatively standardized data, but the overall trend is similar to that observed in the WD dataset. MGASM and its variants demonstrate significantly faster convergence. Among them, MGASM_LDA consistently maintains a leading advantage. It is worth noting that JARUA begins to surpass MGASM after $N_{tr} > 15\%$, indicating that JARUA can gain more benefits from the growth of the training set, while MGASM can exhibit performance surpassing or equal to mainstream advanced models with a smaller training set. NeXtAlign is also remarkable; although it only considers network structure in this study, it still demonstrates outstanding performance. At $N_{tr} = 30\%$, its performance is equivalent to 85.27% of MGASM's, indicating NeXtAlign's superior ability in capturing network features. Although NeXtAlign only considers network structure in this study, it still demonstrates outstanding performance. At $N_{tr} = 30\%$, its performance is equivalent to 85.27% of MGASM's, indicating NeXtAlign's superior ability in capturing network features.

Overall, as N_{tr} increases from 5% to 30%, the MGASM family and NSVUIL exhibit the least average gain from the training data, with increases of only 6.27% and 6.8%, respectively. This indicates that these two models are less dependent on the training dataset and are better suited for scenarios with limited pre-aligned users. In contrast, NeXtAlign and Grad-Align+ show significant improvements of 50.7% and 354.68%, respectively, suggesting they have greater potential when the training dataset is more substantial.

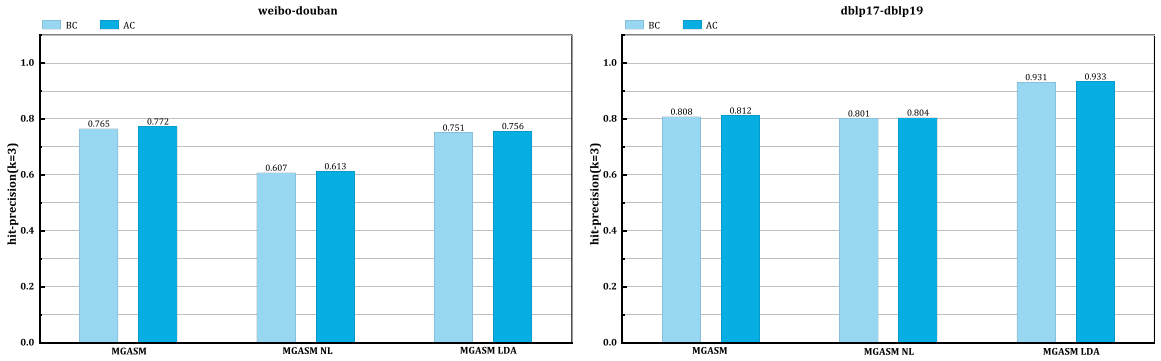
The notable performance of MGASM and MGASM_LDA in the WD and DBLP datasets, achieved with a modest N_{tr}, is impressive. It is important to recognize that increases in N_{tr} did not yield a linear rise in hit-precision, indicating a performance plateau. This may be due to the finite dimensionality of training samples, which is limited by the number of attribute granularities—five in the datasets used. The observed plateau in performance gains with increasing N_{tr} can be linked to the dimensionality of training samples, which is bound by the number of attribute granularities. The datasets were divided into five granularities, facilitating rapid model convergence and robust performance with fewer samples. However, this approach also presents a constraint: the five-dimensional vector has an inherent information ceiling, implying that the model's performance might have approached the maximum potential achievable with this data structure. To overcome this and enhance model performance, it is crucial to integrate more granular user information, such as publication times, content details, browsing histories, bookmarking records, and liking activities. This would provide a richer information base for the model to learn from.



(a) The hit-precision performance of each model at different top-k values in the WD dataset.

(b) The hit-precision performance of each model at different top-k values in the DBLPs dataset.

Fig. 7. The hit-precision performance of all models when top-k varies between 1 and 100.



(a) Performance comparison of hit-precision(k=3) of MGASM and its variants before and after using ARSC on the WD dataset.

(b) Performance comparison of hit-precision(k=3) of MGASM and its variants before and after using ARSC on the DBLPs dataset.

Fig. 8. Performance of MGASM and its variants on hit-precision (k = 3) before and after using ARSC on two datasets.

6.3.3. Performance of the model under different top-k settings

By varying top-k between 1 and 100, we tested the range search capabilities of all models across various datasets.

In the WD dataset, as shown in Fig. 7(a), In terms of hit-precision, MGASM still demonstrates the best performance, achieving a precision of 0.94 at top-k = 100, which is 47.8% higher than DeepDSA's 0.636, the second-best performer. Additionally, NeXtAlign shows impressive growth rates. As top-k increases from 1 to 100, NeXtAlign's hit-precision improves by 134.97%, ranking first. In contrast, due to its initially high base, MGASM exhibits a relatively lower growth rate at only 45.28%.

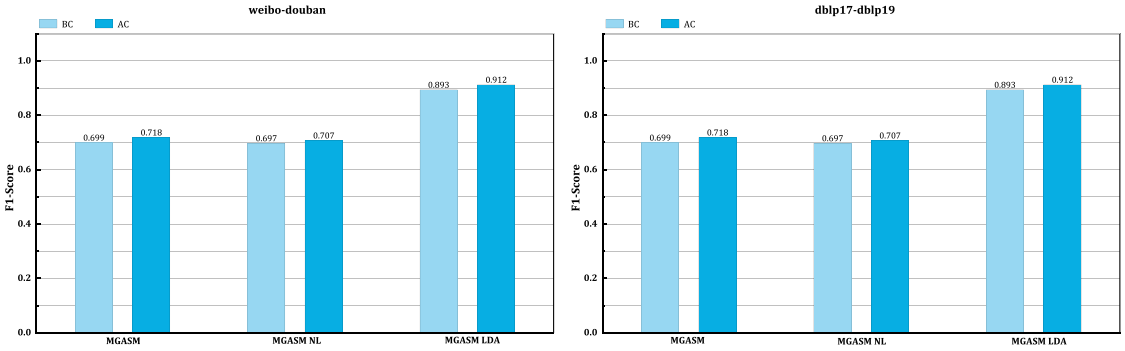
In the DBLP dataset, as shown in Fig. 7(b), MGASM_LDA achieved the highest hit-precision, surpassing the second-place JARUA by 5.78% at top-k = 100. NeXtAlign once again demonstrated the highest growth rate, reaching 119.63%. These results reveal the precision and range search capabilities of each model, providing insights for UA tasks in different scenarios. Specifically, among all the models compared, MGASM and its variant MGASM_LDA exhibit strong precision search capabilities, while NeXtAlign excels in range search capability.

Overall, one of MGASM's key strengths lies in its exceptional precision search capabilities. Although MGASM maintains an advantage at top-k = 100, if the top-k range continues to expand, its hit-precision might be surpassed by DeepDSA and NeXtAlign. This suggests that MGASM has relatively weaker fuzzy query capabilities, which aligns with the observation of its lower recall rate mentioned in Section 6.3.1.

6.3.4. Contribution of various components to the effectiveness of MGASM

By examining the contribution of different components and ARSC across various datasets, the effectiveness of MGASM and its variants is assessed.

In the WD dataset, as illustrated in Fig. 8(a), the hit-precision performance of MGASM significantly surpasses MGASM_NL, with an average improvement of 39.96%, emphasizing the importance of label granularity in enhancing model performance. MGASM slightly outperforms MGASM_LDA, highlighting the effectiveness of pre-trained language models in social network text classification.



(a) Performance comparison of F1 score of MGASM and its variants before and after using ARSC on the WD dataset.

(b) Performance comparison of F1 score of MGASM and its variants before and after using ARSC on the DBLPs dataset.

Fig. 9. Performance of MGASM and its variants F1 score before and after using ARSC on two datasets.

ARSC further enhances the performance of MGASM, with an average improvement of 5.17%. In the DBLP dataset, as shown in Fig. 8(b), MGASM_LDA emerges as the most effective, demonstrating an improvement in hit-precision of 27.02% compared to MGASM, indicating that LDA is more suitable for classifying academic titles with a certain readability threshold. The use of ARSC improves hit accuracy by an average of 2.07%.

Trends are confirmed through F1-score analysis in Fig. 9. In the WD dataset, MGASM's F1 score is 26.1% higher than MGASM_NL and 2.64% higher than MGASM_LDA, with ARSC improving performance by an average of 0.86%. In the DBLP dataset, MGASM_LDA remains superior to MGASM and MGASM_NL, outperforming them by 14.93% and 16.02%, respectively, with ARSC enhancing performance by an average of 0.37%. While the extent of performance improvement varies across different metrics, ARSC consistently proves beneficial.

6.3.5. Experiment on the importance of attribute granularity

To evaluate the significance of attribute embeddings at different granularity levels within the MGASM model, we conducted ablation experiments by testing the hit-precision performance when considering only a single attribute granularity. Specifically, we examined the following variants: MGASM_C (character granularity), MGASM_W (word granularity), MGASM_L (label granularity), MGASM_A (article granularity), and MGASM_S (structure granularity).

As illustrated in Fig. 10, the results on the WD dataset reveal that MGASM_C achieved the highest hit-precision of 0.415, followed by MGASM_L. This suggests that character and label granularity attributes are particularly influential in user alignment within the WD dataset. Conversely, in the DBLPs dataset, the article granularity attribute emerged as the most critical, with other granularities also contributing significantly. When MGASM only considers a single granularity, MA-CDDVs can only quantify the relevance of users with respect to attributes of that granularity. A binary similarity model also cannot effectively learn the correlation distribution between different granularities of attributes. As a result, the performance of models considering only a single granularity attribute in two datasets is significantly lower than that of the MGASM model, which considers multiple granularities. This advantage is attributed to multi-view learning theory. Each view provides information on different aspects of the same object, and by combining these views, a more comprehensive and accurate model can be obtained. This multi-view data learning approach offers stronger generalization ability and higher robustness compared to single-view learning. Specifically, the advantages of multi-granularity embedding arise from the following points:

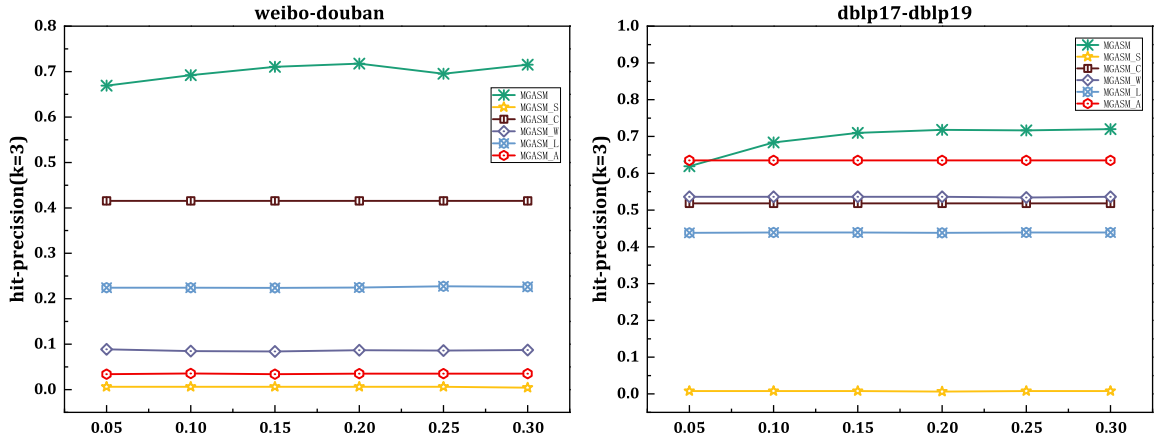
- **Complementarity Principle:** Embeddings of different granularities contain information at different levels, which is inherently complementary. By combining this complementary information, a richer representation of user features can be obtained, thereby enhancing the model's learning capability.
- **Information Correction:** Single-granularity embeddings might introduce certain inherent biases or shortcomings. For example, character-level embeddings might ignore semantics, while document-level embeddings might overlook subtle differences in vocabulary. Multi-view learning, by combining embeddings of multiple granularities, can effectively reduce these biases.

Therefore, through multi-granularity learning, the model can better capture the complex distribution of correlations between user attributes of different granularities across various network platforms. This makes the model more robust and generalizable when facing different types of data.

6.3.6. Dimension sensitivity analysis

Dimension sensitivity analysis was performed to determine the effect of embedding dimensionality on MGASM's performance, examining hit-precision, precision, recall, and F1-score against varying dimensions D.

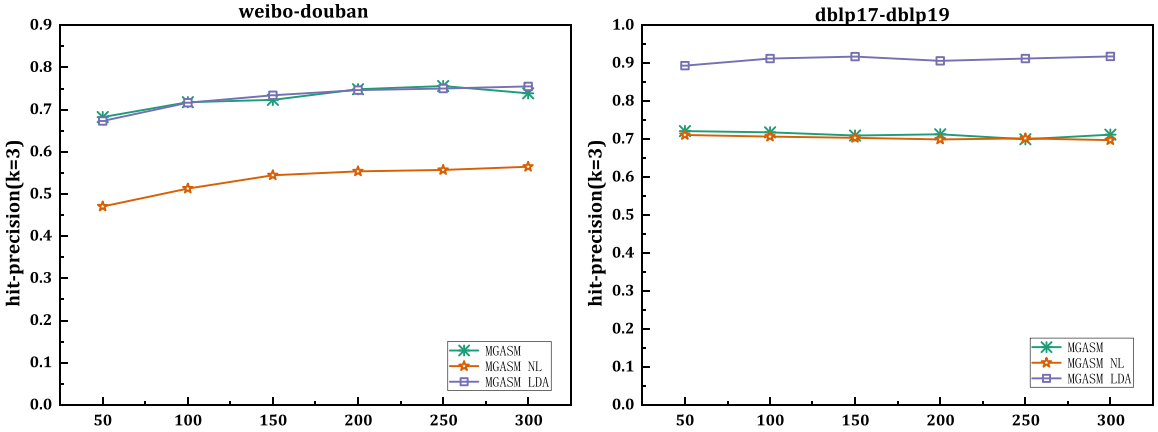
From Fig. 11, it can be observed that there is a significant difference in the dimension sensitivity of MGASM and its variants between the WD and DBLP datasets. In the WD dataset, performance metrics initially improve with increased dimensions, indicating



(a) In the WD dataset, the hit-precision performance of all models when N_{tr} varies from 5% to 30% (k=3).

(b) In the DBLPs dataset, the hit-precision performance of all models when N_{tr} varies from 5% to 30% (k=3).

Fig. 10. he hit-precision performance of all models on both datasets as the training dataset size N_{tr} varies from 5% to 30%.



(a) Performance of MGASM and its variants on hit-precision in the WD dataset when the dimension varies between 50 and 300.

(b) Performance of MGASM and its variants on hit-precision in the DBLPs dataset when the dimension varies between 50 and 300.

Fig. 11. The hit-precision performance of MGASM and its variants as the embedding dimension varies from 50 to 300.

that higher dimensionality captures more detailed information, thereby enhancing accuracy. However, after reaching the peak performance, further increases in dimensionality lead to a plateau in performance. In contrast, the hit-precision of MGASM and its variants remains almost stable in the DBLP dataset. We attribute these results to the differing complexities of the WD and DBLP datasets. When the embedding dimensions cannot fully accommodate the various granularity attributes of users, the model benefits from the increase in embedding dimensions. Conversely, when the embedding dimensions are sufficient to accommodate the granularity attributes of users, the model exhibits extremely low dimension sensitivity.

7. Results and impact

The empirical evaluations conducted on two real-world datasets clearly indicate that, even with a limited number of pre-aligned users available, MGASM outperforms existing baseline models and alternative methods. Specifically, on the Weibo-Douban dataset, we have achieved a hit-precision of over 60% and nearly 100% precision for the first time. This advantage is overwhelming. These breakthroughs enhance the reliability of various artificial intelligence applications, including but not limited to recommendation systems, search engines, information diffusion forecasting, network identity verification, and crime detection.

8. Conclusion

In this paper, we propose a novel end-to-end semi-supervised solution, named MGASM, to address the problem of user alignment across social networks under sparse data conditions. To tackle the challenges posed by attribute heterogeneity and sparsity, the

model integrates multi-granularity feature embeddings. Notably, it independently embeds label information such as user devices for the first time to enhance the embedding process, significantly improving performance. MGASM appropriately utilizes neighboring node information to smooth feature vectors. Additionally, we introduce a low-dimensional and efficient MA-CDDV vector, which, when combined with a binary classification model, significantly reduces the model's reliance on sample data. More innovatively, we introduce the ARSC (Attribute Reappearance Score Correction) mechanism, which considers the interactions between individual attributes across platforms, significantly enhancing various performance metrics and improving the model's judgment ability for similar users. However, this method still has limitations. While MA-CDDV provides the model with advantages such as rapid convergence and high scalability, its relatively simple structural design means that the model cannot continuously benefit from the expansion of the training set. This limitation may make it challenging to adapt to richer and more complex application scenarios in the future. Based on the recall rate metrics and generalization search capability experiments where MGASM did not achieve optimal performance, we recognize the potential value of exploring more complex combination strategies and will consider this as a direction for our future work.

Code availability

To support the reproducibility of our findings and encourage further research, this paper provides comprehensive access to the complete program code, datasets, and detailed instructions. This transparency ensures that interested researchers can replicate the experiments and potentially extend the work presented herein. Code and Data are available for download at the following web links https://github.com/pengyongqiang/MGASM.git.

CRediT authorship contribution statement

Yongqiang Peng: Writing – original draft, Visualization, Software, Formal analysis, Data curation, Conceptualization. **Xiaoliang Chen:** Writing – review & editing, Supervision, Project administration, Methodology, Investigation, Funding acquisition. **Duoqian Miao:** Funding acquisition. **Xiaolin Qin:** Resources, Funding acquisition. **Xu Gu:** Validation. **Peng Lu:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work is supported by the Science and Technology Program of Sichuan Province (Grant no. 2023YFS0424), the National Key R&D Plan "Key Special Project of Cyberspace Security Governance" (No. 2022YFB3104700), the National Natural Science Foundation (Grant nos. 61976158, 62376198), the Science and Technology Service Network Initiative (No. KFJ-STS-QYZD-2021-21-001), and the Talents by Sichuan provincial Party Committee Organization Department, and Chengdu - Chinese Academy of Sciences Science and Technology Cooperation Fund Project (Major Scientific and Technological Innovation Projects).

References

Chen, B., & Chen, X. (2022). MAUIL: multilevel attribute embedding for semisupervised user identity linkage. *Information Sciences*, *593*, 527–545. http://dx.doi.org/10.1016/J.INS.2022.02.023.

Dolatshah, M., Hadian, A., & Minaei-Bidgoli, B. (2015). Ball*-tree: Efficient spatial indexing for constrained nearest-neighbor search in metric spaces. *Computing Research Repository*, http://dx.doi.org/10.48550/arXiv.1511.00628, arXiv:1511.00628.

Duan, S., Long, Y., Xiao, Y., Wang, R., & Li, Q. (2024). E-commerce bookstore user alignment model based on multidimensional feature joint representation and implicit behavior compensation. *Expert Systems with Applications*, *238*(Part E), Article 122084. http://dx.doi.org/10.1016/J.ESWA.2023.122084.

Fanourakis, N., Efthymiou, V., Kotzinos, D., & Christophides, V. (2023). Knowledge graph embedding methods for entity alignment: experimental review. *Data Mining and Knowledge Discovery*, *37*(5), 2070–2137. http://dx.doi.org/10.1007/S10618-023-00941-9.

Grover, A., & Leskovec, J. (2016). node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 855–864). http://dx.doi.org/10.1145/2939672.2939754.

Guo, C., Huang, D., Dong, N., Zhang, J., & Xu, J. (2021). Callback2Vec: Callback-aware hierarchical embedding for mobile application. *Information Sciences*, *542*, 131–155. http://dx.doi.org/10.1016/J.INS.2020.06.058.

Hofmann, T. (2013). Probabilistic latent semantic analysis. *Computing Research Repository*, http://dx.doi.org/10.48550/arXiv.1301.6705, arXiv:1301.6705.

Huang, Y., Zhao, P., Zhang, Q., Xing, L., Wu, H., & Ma, H. (2023). A semantic-enhancement-based social network user-alignment algorithm. *Entropy*, *25*(1), 172. http://dx.doi.org/10.3390/E25010172.

Kim, D., Seo, D., Cho, S., & Kang, P. (2019). Multi-co-training for document classification using various document representations: TF-IDF, LDA, and Doc2Vec. *Information Sciences*, *477*, 15–29. http://dx.doi.org/10.1016/J.INS.2018.10.006.

Lei, Z., Feng, Q., Jie, C., & Shu, Z. (2023). An unsupervised rapid network alignment framework via network coarsening. *Mathematics*, *11*(3), 573. http://dx.doi.org/10.3390/math11030573.

Li, L., Dong, J., & Qin, X. (2023). Dual-view graph neural network with gating mechanism for entity alignment. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, *53*(15), 18189–18204. http://dx.doi.org/10.1007/S10489-022-04393-4.

Li, G., Sun, L., Zhang, Z., Ji, P., Su, S., & Yu, P. (2019). MC 2 :Unsupervised multiple social network alignment. (pp. 1151–1156). http://dx.doi.org/10.1109/BigData47090.2019.9005701.

Li, C., Wang, S., Xu, J., Liu, Z., Wang, H., Xie, X., et al. (2023). Semi-supervised variational user identity linkage via noise-aware self-learning. *IEEE Transactions on Knowledge and Data Engineering*, *35*(10), 10166–10180. http://dx.doi.org/10.1109/TKDE.2023.3250245.

Li, Q., Zhou, Q., Chen, W., & Zhao, L. (2023). User identity linkage via graph convolutional network across location-based social networks. In I. Garrigós, J. M. M. Rodríguez, & M. Wimmer (Eds.), *Lecture notes in computer science*: *Vol. 13893*, *Web engineering - 23rd international conference, ICWE 2023, alicante, Spain, June 6-9, 2023, proceedings* (pp. 158–173). Springer, http://dx.doi.org/10.1007/978-3-031-34444-2_12.

Liang, Z., Rong, Y., Li, C., Zhang, Y., Huang, Y., Xu, T., et al. (2021). Unsupervised large-scale social network alignment via cross network embedding. In *Proceddings of the 30th ACM international conference on information and knowledge management, virtual event* (pp. 1008–1017). http://dx.doi.org/10.1145/3459637.3482310.

Liu, Z., & Wu, X. (2023). Structural analysis of the evolution mechanism of online public opinion and its development stages based on machine learning and social network analysis. *International Journal of Computational Intelligence Systems*, *16*(1), 99. http://dx.doi.org/10.1007/S44196-023-00277-8.

Long, M., Chen, S., Du, X., & Wang, J. (2023). DegUIL: Degree-aware graph neural networks for long-tailed user identity linkage. http://dx.doi.org/10.48550/ARXIV.2308.05322, CoRR arXiv:2308.05322.

Matrouk, K., Srikanth, V., Kumar, S., Bhadla, M. K., Sabirov, M., & Saadh, M. J. (2023). Deep learning-based dynamic user alignment in social networks. *The ACM Journal of Data and Information Quality*, *15*(3), 33:1–33:26. http://dx.doi.org/10.1145/3603711.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. In *Proceddings of 1st international conference on learning representations*. http://dx.doi.org/10.48550/arXiv.1301.3781.

Mu, X., Zhu, F., Lim, E., Xiao, J., Wang, J., & Zhou, Z. (2016). User identity linkage by latent user space modelling. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1775–1784). http://dx.doi.org/10.1145/2939672.2939849.

Munne, R. F., & Ichise, R. (2023). Entity alignment via summary and attribute embeddings. *Logic Journal of the IGPL*, *31*(2), 314–324. http://dx.doi.org/10.1093/JIGPAL/JZAC021.

Oberle, D., Berendt, B., Hotho, A., & Gonzalez, J. (2003). Conceptual user tracking. In *Lecture notes in computer science*: *Vol. 2663*, *Web intelligence, first international atlantic web intelligence conference, AWIC 2003, madrid, Spain, May 5-6, 2003, proceedings* (pp. 155–164). Springer, http://dx.doi.org/10.1007/3-540-44831-4_17.

Park, J., Tran, C., Shin, W., & Cao, X. (2022a). Grad-align: Gradual network alignment via graph neural networks (student abstract). In *Thirty-sixth AAAI conference on artificial intelligence, AAAI 2022, thirty-fourth conference on innovative applications of artificial intelligence, IAAI 2022, the twelveth symposium on educational advances in artificial intelligence, EAAI 2022 virtual event, February 22 - March 1, 2022* (pp. 13027–13028). AAAI Press, http://dx.doi.org/10.1609/AAAI.V36I11.21650.

Park, J., Tran, C., Shin, W., & Cao, X. (2022b). GradAlign+: Empowering gradual network alignment using attribute augmentation. In *Proceedings of the 31st ACM international conference on information & knowledge management, atlanta* (pp. 4374–4378). http://dx.doi.org/10.1145/3511808.3557605.

Patnaik, U. K. C., & Patgiri, R. (2023). Chapter seven - MapReduce based convolutional graph neural networks: A comprehensive review. *Advanced Computing*, *128*, 213–231. http://dx.doi.org/10.1016/BS.ADCOM.2021.10.002.

Qi, D., Chen, S., Sun, X., Luan, R., & Tong, D. (2023). A multiscale convolutional gragh network using only structural information for entity alignment. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, *53*(7), 7455–7465. http://dx.doi.org/10.1007/S10489-022-03916-3.

Ren, J., Jiang, L., Peng, H., Lyu, L., Liu, Z., Chen, C., et al. (2022). Cross-network social user embedding with hybrid differential privacy guarantees. In *Proceedings of the 31st ACM international conference on information & knowledge management* (pp. 1685–1695). http://dx.doi.org/10.1145/3511808.3557278.

Shao, J., Wang, Y., Gao, H., Shi, B., Shen, H., & Cheng, X. (2023). AsyLink: user identity linkage from text to geo-location via sparse labeled data. *Neurocomputing*, *515*, 174–184. http://dx.doi.org/10.1016/J.NEUCOM.2022.10.027.

Singh, D. K. S., N., N., L., R., Sanghavi, P., Vaghela, R. S., Manoharan, P., et al. (2022). Social network analysis for precise friend suggestion for Twitter by associating multiple networks using ML. *International Journal of Information Technology and Web Engineering*, *17*(1), 1–11. http://dx.doi.org/10.4018/IJITWE.304050.

Sun, L., Du, Y., Gao, S., Ye, J., Wang, F., Ren, F., et al. (2023). GroupAligner: a deep reinforcement learning with domain adaptation for social group alignment. *ACM Transactions on the Web*, *17*(3), 1–30.

Sun, L., Zhang, Z., Wang, F., Ji, P., Wen, J., Su, S., et al. (2022). Aligning dynamic social networks: An optimization over dynamic graph autoencoder. *IEEE Transactions on Knowledge and Data Engineering*, *35*(6), 5597–5611.

Tang, J., Song, R., Huang, Y., Gao, S., & Yu, Z. (2024). Semantic-aware entity alignment for low resource language knowledge graph. *Frontiers of Computer Science*, *18*(4), Article 184319. http://dx.doi.org/10.1007/S11704-023-2542-X.

Ukkonen, E. (1992). Approximate string matching with q-grams and maximal matches. *Theoretical Computer Science*, *92*(1), 191–211. http://dx.doi.org/10.1016/0304-3975(92)90143-4.

Wang, Y., Peng, Q., Wang, W., Guo, X., Shao, M., Liu, H., et al. (2022). Network alignment enhanced via modeling heterogeneity of anchor nodes. *Knowledge-Based Systems*, *250*, Article 109116. http://dx.doi.org/10.1016/J.KNOSYS.2022.109116.

Wei, S., Zhou, X., An, X., Yang, X., & Xiao, Y. (2023). A heterogeneous E-commerce user alignment model based on data enhancement and data representation. *Expert Systems with Applications*, *228*, Article 120258. http://dx.doi.org/10.1016/J.ESWA.2023.120258.

Xun, G., Li, Y., Zhao, W. X., Gao, J., & Zhang, A. (2017). A correlated topic model using word embeddings. In *Proceedings of the twenty-sixth international joint conference on artificial intelligence* (pp. 4207–4213). http://dx.doi.org/10.24963/IJCAI.2017/588.

Yan, Y., Zhang, S., & Tong, H. (2021). Bright: A bridging algorithm for network alignment. In *Proceedings of the web conference 2021* (pp. 3907–3917).

Yang, M., Chen, B., & Chen, X. (2022). JARUA: Joint embedding of attributes and relations for user alignment across social networks. *Applied Sciences*, *12*(24), 12709.

Zhai, J., Zhang, S., Chen, J., & He, Q. (2018). Autoencoder and its various variants. In *IEEE international conference on systems, man, and cybernetics, SMC 2018, Miyazaki, Japan, October 7-10, 2018* (pp. 415–419). IEEE, http://dx.doi.org/10.1109/SMC.2018.00080.

Zhang, S., & Tong, H. (2018). Attributed network alignment: Problem definitions and fast solutions. *IEEE Transactions on Knowledge and Data Engineering*, *31*(9), 1680–1692.

Zhang, S., Tong, H., Jin, L., Xia, Y., & Guo, Y. (2021). Balancing consistency and disparity in network alignment. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining* (pp. 2212–2222).

Zhang, J., Yu, P. S., & Zhou, Z. (2014). Meta-path based multi-network collective link prediction. In *Proceddings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1286–1295). http://dx.doi.org/10.1145/2623330.2623645.

Zhao, K., Bai, T., Wu, B., Wang, B., Zhang, Y., Yang, Y., et al. (2020). Deep adversarial completion for sparse heterogeneous information network embedding. In *WWW '20: the web conference 2020, Taipei, Taiwan, April 20-24, 2020* (pp. 508–518). ACM / IW3C2, http://dx.doi.org/10.1145/3366423.3380134.

Zhao, C., Zhao, H., He, M., Zhang, J., & Fan, J. Cross-domain recommendation via user interest alignment. In *Proceedings of the ACM web conference* (pp. 887–896). http://dx.doi.org/10.1145/3543507.3583263.

Zhou, T., Lim, E., Lee, R. K., Zhu, F., & Cao, J. (2020). Retrofitting embeddings for unsupervised user identity linkage. In *Lecture notes in computer science*: *Vol. 2084*, *Advances in knowledge discovery and data mining - 24th Pacific-Asia conference, PAKDD 2020, Singapore, May 11-14, 2020, proceedings, part I* (pp. 385–397). Springer, http://dx.doi.org/10.1007/978-3-030-47426-3_30.

Zhou, Y., Ren, J., Jin, R., Zhang, Z., Zheng, J., Jiang, Z., et al. (2022). Unsupervised adversarial network alignment with reinforcement learning. *ACM Transactions on Knowledge Discovery from Data*, *16*(3), 50:1–50:29. http://dx.doi.org/10.1145/3477050.

Zhou, F., Wen, Z., Zhong, T., Trajcevski, G., Xu, X., & Liu, L. (2020). Unsupervised user identity linkage via graph neural networks. In *IEEE global communications conference, GLOBECOM 2020, virtual event, Taiwan, December 7-11, 2020* (pp. 1–6). IEEE, http://dx.doi.org/10.1109/GLOBECOM42002.2020.9322311.

Zhu, B., Bao, T., Liu, L., Han, J., Wang, J., & Peng, T. (2023). Cross-lingual knowledge graph entity alignment based on relation awareness and attribute involvement. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, *53*(6), 6159–6177. http://dx.doi.org/10.1007/S10489-022-03797-6.