MRG-SER: Self-supervised Spatial Entity Resolution Based on Multi-Relational Graph

Hanchen Qiu
Southeast University
Nanjing, China
hc_qiu2024@163.com

Haojia Zhu
Southeast University
Nanjing, China
zhuhaojia@seu.edu.cn

Zhicheng Li
Southeast University
Nanjing, China
lizhicheng@seu.edu.cn

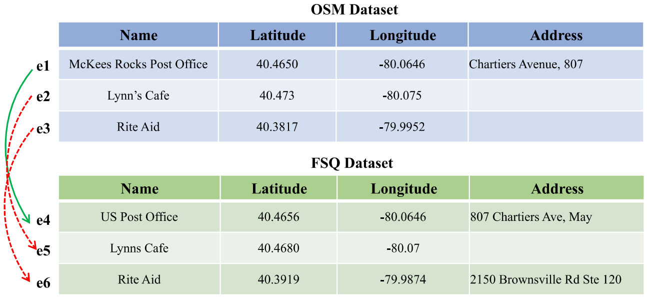
Jiahui Jin
Southeast University
Nanjing, China
jjin@seu.edu.cn

ABSTRACT

Spatial Entity Resolution (SER), which aims to determine whether different points of interest in the real world point to the same spatial entity, is known to be a labour-intensive task. It contributes to the quality of building high quality geospatial databases, which in turn improves the quality of navigation, social networking and logistics services. Existing SER methods are unable to achieve high performance due to insufficient feature discovery or scarcity of labelled data. We introduces MRG-SER, a self-supervised spatial entity resolution framework based on multi-relational graphs. It has the following advantages:(1) Automatic generation of high quality positive and negative labels without manual labelling. (2) Effective discovery of spatial entity neighbourhood features. MRG-SER consists of two modules, i.e. Spatial Entity Automatic label Generation (SEAG) and Multi-Relational Graph based Spatial Entity Matching (MRG-SEM). SEAG is used to generate a set of high quality positive and negative labels for training. In MRG-SEM, we first construct spatial entity multi-relational graphs and extract graph features by GNNs. Second, we combine the graph features, sentence features and distance features of entities to collaboratively predict whether entity pairs match or not. Experiments have demonstrated the accuracy and validity of MRG-SER, which is even superior to the most advanced supervised SER methods.

VLDB Workshop Reference Format:
Hanchen Qiu, Haojia Zhu, Zhicheng Li, and Jiahui Jin. MRG-SER: Self-supervised Spatial Entity Resolution Based on Multi-Relational Graph. VLDB 2024 Workshop: 3rd International Workshop on Large-Scale Graph Data Analytics (LSGDA 2024).

1 INTRODUCTION


e1
e2
e3
e4
e5
e6

OSM Dataset

Name	Latitude	Longitude	Address
McKees Rocks Post Office	40.4650	-80.0646	Chartiers Avenue, 807
Lynn’s Cafe	40.473	-80.075	
Rite Aid	40.3817	-79.9952	

FSQ Dataset

Name	Latitude	Longitude	Address
US Post Office	40.4656	-80.0646	807 Chartiers Ave, May
Lynns Cafe	40.4680	-80.07	
Rite Aid	40.3919	-79.9874	2150 Brownsville Rd Ste 120

Figure 1: Example of spatial entity resolution. (Green solid and red dashed lines represent matched and unmatched spatial entity pairs, respectively.)

High-quality geospatial databases are crucial for enhancing the effectiveness of Location-Based Services (LBS), such as navigation, logistics, and targeted advertising. Spatial data, primarily composed of spatial entities known as Points of Interest (POIs). These spatial entities typically include spatial attributes (e.g., longitude and latitude) and textual attributes (e.g., name and address). However, the representation of the same spatial entity can vary across different data sources, resulting in duplication and inconsistency during data integration. Consequently, spatial entity resolution (SER) has garnered significant attention. SER aims to identify and match records from different data sources that refer to the same real-world spatial entity [2], thereby creating comprehensive, high-quality geospatial databases.

Example 1.1. As shown in Fig. 1 ,e_1 and e_4 are matches, but there are synonyms (e.g., Avenue = Ave), address reversals, and low name similarity; e_2 and e_5 are flagged as mismatches due to subtle name differences (e.g., Lynn’s vs. Lynns); Rite Aid is a well-known drugstore chain, and e_3 and e_6 are 1.3 kilometers apart, so they do not match.

Existing research on spatial entity resolution is primarily categorized into rule-based [1, 5, 11, 22, 25, 27, 31] and learning-based approaches [2, 3, 24]. Rule-based methods depend on expert domain knowledge, which is often inflexible and challenging to apply across different datasets. And learning-based methods require large-scale, high-quality labeled data for training, which involves substantial manpower and time costs. For instance [6], achieving F-measures of approximately 99% with random forests may necessitate up to 1.5 million labels, which could require approximately 40 workers around 2 months to complete. Therefore, we propose for the first time to apply self-supervised learning to SER.

Proceedings of the VLDB Endowment. ISSN 2150-8097.

Existing unsupervised entity resolution methods [4, 8, 30, 36] are not suitable for SER. Ge et al. [8] introduced a self-supervised entity resolution framework called CollaborEM. ZeroER [30] is an innovative unsupervised entity resolution method that employs a Gaussian mixture model to learn the distributions of matches and mismatches. In conclusion, none of these methods consider the spatial features of the entities themselves or the spatial neighbourhood features between entities, rendering them unsuitable for SER. The current challenges are therefore twofold, as set out below.

Challenge I: ***Spatial features affect the generation of labeled data in self-supervised learning.*** Learning-based methods rely on high-quality labelled data and have high time and labour costs. Some self-supervised methods[8] are capable of automatically generating labelled data. However, longer attributes in spatial entities (e.g., addresses) affect the quality of the marker generation and reduce the accuracy of the model.

Challenge II: ***Spatial neighbourhood features between entities are challenging to learn.*** Existing methods tend to evaluate each pair of spatial entities in isolation, ignoring the complex neighbourhood features between entities, and therefore fail to identify pairs of entities with significantly different names but belonging to the same region. Some entity resolution methods construct attribute-relationship graphs [4, 33] to explore the implicit relationships between entities. However, these methods do not consider spatial attributes, resulting in poor performance in spatial entity resolution tasks.

Contributions. We propose MRG-SER, a self-supervised spatial entity resolution framework based on multi-relational graphs. MRG-SER consists of two modules, Spatial Entity Automatic label Generation (SEAG) and MRG-based Spatial Entity Matching (MRG-SEM). The contributions of this paper are summarised as follows:

- For **Challenge I**, we propose SEAG. This module investigates reliable positive and negative label generation strategies to enlarge the training samples and significantly reduce the model's dependence on labelled data.
- For **Challenge II**, we firstly constructed a Spatial Entity Multi-relational Graph (SEMRG), which fully captures the spatial neighbourhood graph features of the entities. Secondly, we propose the MRG-SEM, which combines graph features, sentence features and distance features of spatial entities to jointly predict the matching relationships between entity pairs.
- Extensive experiments based on real urban spatial entity data are performed and compared with existing state-of-the-art algorithms to verify the effectiveness and feasibility of MRG-SER.

2 RELATED WORK

2.1 Entity resolution

Existing entity resolution efforts are mainly based on rule-based [7, 13, 29], crowdsourcing [9, 28]. Rule-based approaches offer high interpretability but need to involve domain experts and suffer from limited flexibility. Crowdsourcing methods rely on people's judgement, which can reduce accuracy and efficiency if the number or quality of workers is low. In recent years, pre-trained large language models (LLMs) have been rapidly developed and applied to the field of entity resolution [8, 12, 14, 16, 18, 19, 37], achieving high accuracy rates. Peeters et al. [18] proposed a dual-objective training method called JointBERT for entity matching. DITTO [16] identifies matched pairs of entities by fine-tuning LLMs, treating the entity matching task as a sequential pair classification problem. All of the above methods rely on a large amount of expensive labelled data, and how to reduce the reliance on labelled data has become a hot topic of discussion in the current academic community. Zhang et al. [36] proposed an unsupervised entity parsing graph-theoretic fusion framework that evaluates the matching probability of two records based on the TF-IDF and the record graph. Ge et al. proposed a self-supervised entity matching framework called CollaborEM [8], which trains the two records through automatic label generation and collaborative entity matching training for matching entities in two stages. All of the above works do not fully consider the spatial characteristics of entities themselves and the spatial neighbourhood features between entities, so they cannot be directly applied to spatial entity resolution tasks.

Table 1: Notations and Descriptions in MRG-SER

Notation	Description
G	Multi-relational graph, $G = \{E, R, A\}$
E	Nodes in G, $E = \{E_A, E_P, E_{Attr}\}$
E_A	Set of AOIs in G
E_P	Set of POIs in G
E_{Attr}	Set of attributes in G
$e \in E_P$	A POI(spatial entity) belonging to E_P
R	Set of node relationships in G
A	Set of edge types in G
$PSet$	Positive label set
$NSet$	Negative label set
h_e	Graph embedding of e in MRG
$G_{abs}(h_a, h_b)$	Differencies between h_a and h_b
$G_{dot}(h_a, h_b)$	Similarities between h_a and h_b
$E_b(e_i, e_j)$	Distance feature between e_i and e_j
M	Attribute similarity matrix

2.2 Spatial entity resolution

A part of the research focuses solely on spatial objects, which are entities that possess only spatial attributes. Relevant studies include road network data matching [22, 27, 31, 32], location point matching [21, 23, 26, 34], polygon area matching [10, 20], etc. Another part of the research focuses on spatial entities that possess both spatial and textual attributes. Existing spatial entity resolution work is mainly based on rule-based [1, 5, 11, 22, 25, 27, 31] and learning-based approaches [2, 3, 24]. Rule-based methods give matching decisions through predefined rules and logic. Isaj et al. [11] proposed a heuristic framework for geographic entity matching based on the idea of Pareto optimality. Although the above methods are interpretable, they cannot capture phenomena such as multiple meanings of words (e.g., Fig. 1 e_1 and e_4). In recent years, deep learning have developed rapidly. Balsebre et al. [2] proposed a joint spatial entity resolution framework based on supervised learning

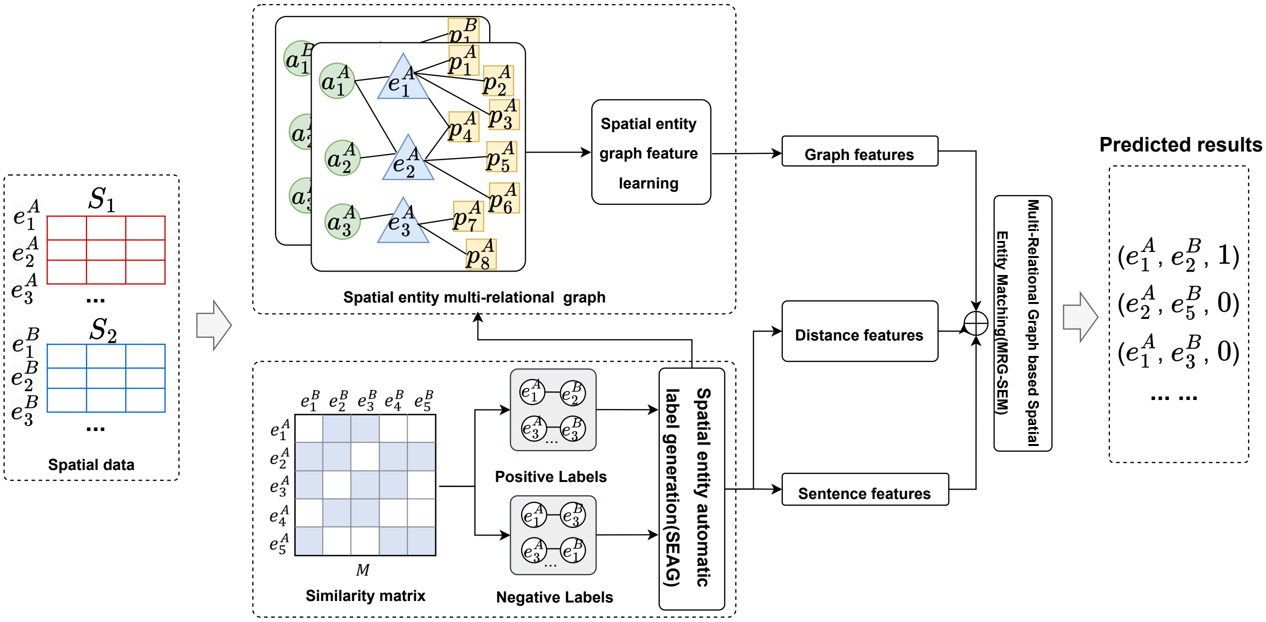


Figure 2: MRG-SER framework.

called Geo-ER. Although the accuracy is high, Geo-ER relies on a large amount of pre-labeled training data, resulting in high labor costs.

3 MRG-SER

3.1 Problem formulation

Spatial entity resolution aims to find matching tuple pairs between two relational datasets that refer to the same real-world entity. Let S be a relational dataset containing $|S|$ tuples and m attributes $A = \{A[1], A[2], \cdots, A[m]\}$. Each tuple $e \in S$ consists of a set of attribute values, represented as $V = \{e.A[1], e.A[2], \cdots, e.A[m]\}$, where $e.A[m]$ is the m-th attribute value of the tuple e, corresponding to the attribute $A[m] \in A$. The ER task can be represented as $T = \{(e^A, e^B) \in S_1 \times S_2 | e^A \equiv e^B\}$, where $e^A \in S_1$, $e^B \in S_2$, and $\equiv$ denotes the matching relationship between the tuples e^A and e^B. Table 1 summarises the symbols frequently used in this paper.

3.2 Framework overview

3.2.1 Self-supervised learning strategy. MRG-SER takes as input the spatial relationship datasets S_1 and S_2 and outputs a set of prediction results in the form of $(e_1^A, e_2^B, label)$. The structure of MRG-SER is illustrated in Fig. 2. It consists of two main components: Spatial Entity Automatic Label Generation (SEAG) and Multi-Relational Graph based Spatial Entity Matching (MRG-SEM). In MRG-SER, we implement self-supervised learning through the following steps:

(1) **Automatic label generation**: Initially, SEAG construct the similarity matrix M and generates high-quality sets of positive labels $PSet$ and negative labels $NSet$ based on name, text, and spatial similarity between entities.
(2) **Spatial entity matching**: Next, we construct spatial entity multi-relational graphs, extract features such as the spatial neighborhood of entities using the AttrGNN[17] model, and use MRG-SEM to integrate graph features, sentence features, and distance features of spatial entities to jointly predict matching results.
(3) **Self-supervised learning**: Finally, we use the $PSet$ and $NSet$ generated by SEAG to train AttrGNN and MRG-SEM, enabling self-supervised learning without manual intervention.

3.2.2 AOI-based spatial entity multi-relational graph construction. We design a spatial entity multi-relational graph (SEMRG) which consists of three types of nodes: area of interest nodes (AOIs), point of interest nodes (POIs) and attribute nodes. AOIs and POIs are connected by a *belongTo* relationship, as shown in Fig. 3. By using AOIs to establish proximity relationships between spatial entities, we address the shortcomings of other methods that cannot be applied to spatial data. Furthermore, we connect different spatial entities that share the same attribute value nodes, thereby preserving semantic relationships between different spatial entities.

SEMRG is represented as $G = \{E, R, A\}$, where E is a set of nodes, including E_A, E_P and E_{Attr}. The relationship set R is used to connect nodes and mainly contains three types of edges: POI-Attribute Value, AOI-Attribute Value, and POI-AOI. Each edge is stored in the form of a triple $R = \{(e, a, v) | e, v \in E, a \in A\}$. The attribute set $A = \{name, address, category, phone, belongTo\}$ represents the types of edges that connect nodes.

3.3 Spatial entity automatic label generation

The structure of Spatial Entity Automatic label Generation (SEAG) is shown in Fig. 4. First construct the similarity matrix M. Input

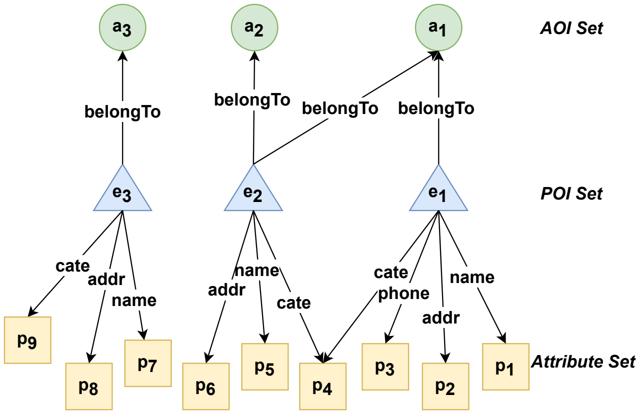


Figure 3: Structure of the spatial entity multi-relationship graph.

a set of spatial entities S_1, S_2 and construct a two-dimensional matrix $M \in [0,1]^{|S_1|\times|S_2|}$, where $|S_1|$ and $|S_2|$ denote the number of entities in the set of spatial entities, respectively. The element $M[i][j]$ in the ith row and jth column of the matrix represents the combined semantic and spatial distance similarity between entities $e_i^A \in S_1$ and $e_j^B \in S_2$. Then, it is computed for each row and column element in the similarity matrix M. In this paper, the attributes of a spatial entity are categorized into three parts: name, other textual information, and coordinates. For name and other textual information, the representation vectors are processed and output using BERT, and then the cosine similarity function (Cos) is used to measure the name semantic similarity $Sim_N(e_i^A, e_j^B)$ and other textual semantic similarity $Sim_{Info}(e_i^A, e_j^B)$ of the two elements.

$$Sim_N(e_i^A, e_j^B) = \mathrm{Cos}(BERT(e_i^A[name], e_j^B[name])) \tag{1}$$

$$Sim_I(e_i^A, e_j^B) = \mathrm{Cos}(BERT(e_i^A[info], e_j^B[info])) \tag{2}$$

For coordinate information, we calculate Haversine distance using Eq. 10, which models the Earth as a sphere model and calculates the latitude and longitude distance between two spatial entities on the sphere based on the equatorial radius. The Haversine distance is then regularized to obtain the distance similarity:

$$Dist(e_i^A, e_j^B) = Norm(dis(e_i^A, e_j^B)) \tag{3}$$

Based on three key dimensions, we calculate the values of the combined attribute similarity matrix. The computation process follows Eq. 4, and the hyperparameters α, β, and γ are introduced to regulate the weights of name, other text, and distance in the final similarity score. $M[i][j]$ denotes the combined attribute similarity between entities e_i^A and e_j^B.

$$\begin{aligned} M[i][j] = \alpha \cdot Sim_N(e_i^A, e_j^B) \quad &+ \beta \cdot Sim_I(e_i^A, e_j^B) \\ - \gamma \cdot Dist(e_i^A, e_j^B) & \end{aligned} \tag{4}$$

Finally, the automatic label generation strategy is designed based on the spatial attribute similarity matrix M. Sorting the combined attribute similarity fetches for each row in M, the Top-K similar most neighboring matrices $Nearest_A = [e_{ij}]_{1\le i\le|S_1|,1\le j\le K}$ of entities in S_1 within S_2 are similar to the most neighboring matrices $Nearest_A = [e_{ij}]_{1\le i\le|S_1|,1\le j\le K}$, where $K \in \mathbb{Z}^+$, and generating the $similarlyNearest_B = [e_{ij}]_{1\le i\le|S_2|,1\le j\le K}$, as illustrated in Fig. 4 for the case when $k = 3$.

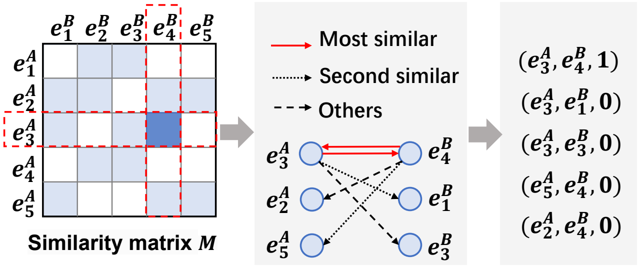


Figure 4: Spatial entity automatic label generation strategy example.

The positive label set $PSet$ and the negative label set $NSet$ are determined based on the nearest neighbour matrices $Nearest_A$ and $Nearest_B$. We use the IKGC algorithm [35] to generate positive labels. Specifically, under the premise that two entities are determined to be the most similar entities to each other, it is required that the difference between the similarity value of the most similar entity pair and that of the second most similar entity pair in the Top-K similarity matrix is greater than a predefined threshold t before the entity pair is included in the positive sample set $PSet$.

Based on the set of positive labeled tuple pairs $PSet$, given a positive labeled entity pair (e_i, e_j), a set of negative labeled entities is obtained by using a number of proximity entities that are not the most similar to each other in $Nearest_A$ and $Nearest_B$ to replace e_i or e_j, respectively, in the positive labeled entity pair.

3.4 MRG-based Spatial Entity Matching

3.4.1 Spatial entity graph feature learning. This section aims to embed spatial entities from different datasets into the same vector space, ensuring that the matching entities are similar in the vector space. For this purpose, we use the advanced entity alignment model AttrGNN [17] to generate graph embeddings for each spatial entity, as shown in equations 5 and 6.

$$o_{n_i}^{l+1} = \mathrm{AGGREGATION}\left(\{h_{n_j}^l, r_{ij} \mid \forall n_j \in N(n_i)\}\right) \tag{5}$$

$$h_{n_i}^{l+1} = \mathrm{UPDATE}\left(h_{n_i}^l, o_{n_i}^{l+1}\right) \tag{6}$$

where $h_{n_i}^{l+1}$ denotes the embedding vector of the node n_i at the $l+1$ level, which is obtained by aggregating the information of the neighbouring nodes. $N(n_i)$ denotes the set of neighbouring nodes of node n_i in the graph. r_{ij} denotes the embedding vector of the edges connecting n_i and n_j. The process of graph feature learning usually consists of two steps: firstly, aggregating the information of the neighbouring nodes of the current node through the aggregation function (AGGREGATION); and then integrating the aggregated information to the current node using the update function (UPDATE). Specifically, aggregating information about neighbours is implemented using the attention mechanism, while updating nodes is implemented using mean aggregation.

Table 2: Experimental Data

City	Source (S_1 - S_2)	$\lvert S_1\rvert$	$\lvert S_2\rvert$	$\lvert C\rvert$	Number of Matches	Positive Label Proportion
Nanjing	Dianping - Meituan	12356	828	31437	415	1.32%
Pittsburgh	OSM-FSQ	2564	2474	71141	1247	1.75%

The GNN-based approach requires equivalent sets of entity pairs to train the model, and we use the positive and negative sample sets $PSet$ and $NSet$ obtained from the automatic label generation module(Section 3.3) to train the dataset, with the goal of minimizing the cosine similarity between the vectors corresponding to the two matching entities, and maximizing the cosine similarity between the vectors of the two mismatching entities. Therefore, the loss function is set as follows:

$$L_k = \sum_{PSet}\left(\sum_{NSet}\left[\cos(\mathbf{e}_i^k, \mathbf{e}_j^k) - \cos(\mathbf{e}_i^k, \mathbf{e}_j'^k) + \eta\right]_+ + \sum_{NSet}\left[\cos(\mathbf{e}_i^k, \mathbf{e}_j^k) - \cos(\mathbf{e}_i'^k, \mathbf{e}_j^k) + \eta\right]_+\right) \tag{7}$$

where, $(e_i, e_j) \in PSet$ and $(e_i', e_j) \in NSet$, $PSet$ and $NSet$ are the positive and negative samples obtained from the automatic label generation module. k denotes different GNN channel GC_k. Cos denotes the cosine similarity function. η denotes a hyperparameter that increases the model's ability to discriminate between pairs of widely differing entities, with a default of $\eta = 1.0$. The conditional expression $[x]_+ = \max(x, 0)$ indicates that the value x is taken if x is positive, otherwise the expression returns 0.

3.4.2 *Spatial entity feature extraction.*

MRG-based Spatial Entity Matching (MRG-SEM) predicts the final matching results by using three features of spatial entities: graph features, sentence features and distance embedding vectors. Implementation details are given below.

Graph feature extraction: Suppose that in the multi-relational graphs G_1 and G_2, the graph embeddings of entities e_1 and e_2 are h_{e_1} and h_{e_2}, the element-by-element differences $G_{abs}(h_{e_1}, h_{e_2})$ and element-by-element similarities $G_{dot}(h_{e_1}, h_{e_2})$ of the two embeddings are computed respectively.

$$G_{abs}(h_{e_1}, h_{e_2}) = |h_{e_1} - h_{e_2}| \tag{8}$$

$$G_{dot}(h_{e_1}, h_{e_2}) = h_{e_1} \odot h_{e_2} \tag{9}$$

where $G_{abs}(h_{e_1}, h_{e_2})$ denotes the element-by-element difference of the two graph features, and $|\cdot|$ denotes the absolute value of the element-by-element difference of the element calculations of h_{e_1} and h_{e_2} in each dimension. $G_{dot}(h_{e_1}, h_{e_2})$ denotes the element-by-element similarity of the features of the two graphs, and $\odot$ denotes the Hadamard product, where each element is the product of the corresponding elements in e_1 and e_2. The results of $G_{abs}(h_{e_1}, h_{e_2})$ and $G_{dot}(h_{e_1}, h_{e_2})$ are vectors of the same dimension $\mathbb{R}^g$ as h_{e_1} and h_{e_2}.

The graph features capture and compare the similarity of the two entity vectors in the model, if the value of G_{abs} is smaller, their embedding vectors will be more similar, which in turn suggests that the two entities may be matched. If G_{dot} obtains a larger value, it means that the two entities have similar embedding features in multiple dimensions, i.e., the two vectors are consistent in more than one way, which in turn suggests that the two entities may have a matching relationship.

Sentence feature extraction: The main idea is to use the Pre-trained language model to extract the textual attributes of two entities to form a sentence, for which the classification task is fine-tuned. In the input phase of the model, this module inputs entity pairs (e_1, e_2) and constructs the entity pairs as sequence pairs. Each spatial entity e contains attributes $e[a], a \in Attr$, which divides the set of attributes $Attr$ into a set of textual attributes $Attr_t = \{name, address, cate, phone\}$ and a set of spatial attributes $Attr_s = \{lat, lon\}$. Individual entities are serialised by adding the tokens [COL] and [VAL] to the attribute name and attribute value, where [COL] precedes the attribute name and [VAL] precedes the attribute value. Construct textual attribute sequences $Seq_t(e_1)$ and $Seq_t(e_2)$ for entities e_1 and e_2, respectively.

Distance embedding vector: We use Haversine formula to calculate the distance between two spatial entities e_1 and e_2:

$$dis(e_1, e_2) = Haversine(\varphi_1, \varphi_2, \lambda_1, \lambda_2, r) \tag{10}$$

where (φ_1, λ_1) are the latitude and longitude of entity e_1 and r is the radius of Earth. The spatial distance embedding vector generation module uses the Eq. 10 to calculate the spatial distance $dis(e_1, e_2)$ of a spatial entity pair (e_1, e_2), obtains the maximum spatial distance $maxDist$ of the entity pair in the dataset, and based on the maximum distance, normalises $dis(e_1, e_2)$ into a [-1,1] interval of values and embedded into a vector space of dimension d_{dist} as shown in Eq. 11.

$$E_b(e_1, e_2) = \theta_{dist}^T \cdot \left(2 \cdot \frac{dis(e_1, e_2)}{maxDist} - 1\right) + v_{dist} \tag{11}$$

where $\theta_{dist}, v_{dist} \in \mathbb{R}^{d_{dist}}$ are the parameters that can be learnt during the training process of the model, and the parameters are adjusted by minimising the loss function so as to better embed the information of the distances between spatial entities.

3.5 Loss function and self-supervised training

In the training process, in addition to the input of a sequence $Seq_t'(e_1, e_2)$ consisting of a set of entities, the positive and negative labels $y \in \{0, 1\}$ generated by Section 3.3 are also input. We concatenates the above three vectors to form a new multidimensional vector and feed it into a fully connected layer, which outputs the raw prediction scores of match or mismatch, and finally determines the categories predicted. We use a variant of cross-entropy loss as the objective training function, as shown in Eq. 12.

$$L(y = k|(e_1, e_2)) = -\log \frac{\exp(v_k)}{\sum_{j\in\{0,1\}} \exp(v_j)} \tag{12}$$

$$v_k = W(E_{[CLS]}; E_b; G_{abs}; G_{dot}) \tag{13}$$

where $\forall k \in \{0, 1\}$, L is the loss function; y is the true label of the entity pair (e_1, e_2), which is 1 for matched entity pairs and 0 for unmatched ones; and v_k is the log odds (logits) computed by the

Table 3: Overall SER results (The best and second scores are in bold and italic, respectively.)

Models	NanJing			Pittsburgh		
	precision	recall	F1 Score	precision	recall	F1 Score
CollaborEM(TKDE 2021)	**0.9604**	0.5253	0.6791	**0.9053**	0.3660	0.5212
GraphER(AAAI 2020)	0.5698	0.5698	0.5698	0.5563	0.6146	0.6865
GTMiner(SIGMOD 2023)	0.9333	0.8077	0.8660	0.8831	*0.8281*	0.8533
GeoER(WWW 2022)	0.8146	**0.9389**	*0.8723*	0.8740	0.8127	*0.8437*
MRG-SER(Ours)	*0.8933*	*0.9054*	**0.8993**	*0.8982*	**0.8506**	**0.8738**

model, which represents the probability that a tuple pair belongs to category k, with k standing for either 0 or 1. v is a function of the sentence feature $E_{[CLS]}$, the distance feature $E_b(e_1, e_2)$, graph features $G_{abs}(h_{e_1}, h_{e_2})$ and $G_{dot}(h_{e_1}, h_{e_2})$ are jointly generated, and $W \in \mathbb{R}^{(n+d_{dist}+2c)\times|k|}$ is a trainable weight matrix that maps the input features to the final logits space, converting the input high-dimensional features into probabilities of predicted categories. The use of $(;;;)$ in Eq. 13 denotes the vertical stacking of vectors.

4 EXPERIMENTS

4.1 Dataset and experimental settings

We collect a total of 18,222 spatial entities in Nanjing and Pittsburgh from four real-world LBSs: Dianping, Meituan, OpenStreetMap (OSM), and Foursquare (FSQ), and manually label the datasets as experimental benchmarks, as shown in Table 2. We randomly divide the dataset into training, validation and test sets with a ratio of 5:2:3 and use the AdamW optimiser. In particular, the Pittsburgh dataset used in this paper differs from GeoER[2] in terms of numbers and positive label proportion. The AttrGNN [17] is used for graph feature extraction. In all experiments, the max sequence length is set to 256; the learning rate is set to $2e-5$; the batch size is set to 32; the epochs is set to 10. In addition, we empirically set the hyper-parameters α, β, and γ to 0.595, 0.105, and 0.3, and experimentally tuned the parameter b to 0.03. All programs are run on an NVIDIA GeForce RTX 3090.

4.2 Comparison methods

We compare MRG-SER with current state-of-the-art solutions in the field of (spatial) entity resolution. The results are reported in terms of the precision, recall, and F1 score on the test set.

- **GTMiner**[3](SIGMOD 2023) proposes an entity relationship prediction model that predicts three kinds of relationships (same-as, serves, part-of) in spatial entities to construct a knowledge graph oriented to spatial entity relationships. We use the same-as relationship as a comparative result for entity resolution.
- **GeoER**[2](WWW 2022) proposes a deep learning framework for spatial entity resolution. The method applies BERT to extract semantic features of textual attributes and measures spatial distance features, and combines the graph attention mechanism to integrate the information of neighboring entities, thus obtaining higher entity resolution results.
- **CollaborEM[8]**(TKDE 2021) is a self-supervised entity resolution method for traditional relational data, which constructs small-scale attribute graphs and jointly identifies the matching probabilities of candidate entity pairs based on GCN graph features and text features.
- **GraphER[15]**(AAAI 2020) is a GCN-based implementation of an entity resolution model. The method constructs relational data as an entity record graph and uses GCN to capture and integrate multiple types of relationships to improve the accuracy of entity matching.

4.3 MRG-SER overall performance

Table 3 summarises the performance of MRG-SER and its competing methods. The results show that MRG-SER performs best on the Nanjing and Pittsburgh datasets with F1 scores of 0.8993 and 0.8738 respectively, outperforming the self-supervised methods GeoER and GTMiner. This is attributed to the fact that SEAG generates reliable labelled data to train the model, and that MRG-SER adequately takes into account spatial neighbourhoods between entities. And we observe that CollaborEM and GraphER, which are oriented towards relational data, treat spatial attributes as ordinary textual attributes and therefore lose the key feature of identifying whether two entities match, with F1 scores below 0.7.

Compared to MRG-SER, GTMiner and GeoER perform slightly less well in the matching task, probably because they are better suited to different entity resolution scenarios. GeoER's neighbourhood features are more affected by the sparsity of the spatial entity distribution, while the Pittsburgh dataset has a sparser entity distribution and thus cannot effectively exploit the similarity of neighbourhood features for matching. GTMiner's performance in identifying matching relationships relies in part on inferring other spatial entity relationships. However, in the NanJing dataset, there are fewer "serves" and "part-of" relationships between entities, which may adversely affect the matching performance. The above results show that MRG-SER outperforms spatial data-oriented supervised learning methods, which demonstrates the effectiveness of our method in the spatial entity resolution task.

4.4 Analysis of label generating quality

We validate the quality of the labels generated by the SEAG module through the use of TP, FN, TN, FP, TPR, TNR. The True Positive Rate (TPR) represents the proportion of matched entities that are correctly labeled, denoted as $\frac{TP}{TP+FN}$. The True Negative Rate (TNR) represents the proportion of mismatched entities that are correctly labeled, denoted as $\frac{TN}{TN+FP}$.

We first evaluated the quality and quantity of labels generated by SEAG, as shown in Fig. 5. Where SEAG is our proposed automatic label generation strategy, and SEAG(-dist) is that the spatial distance

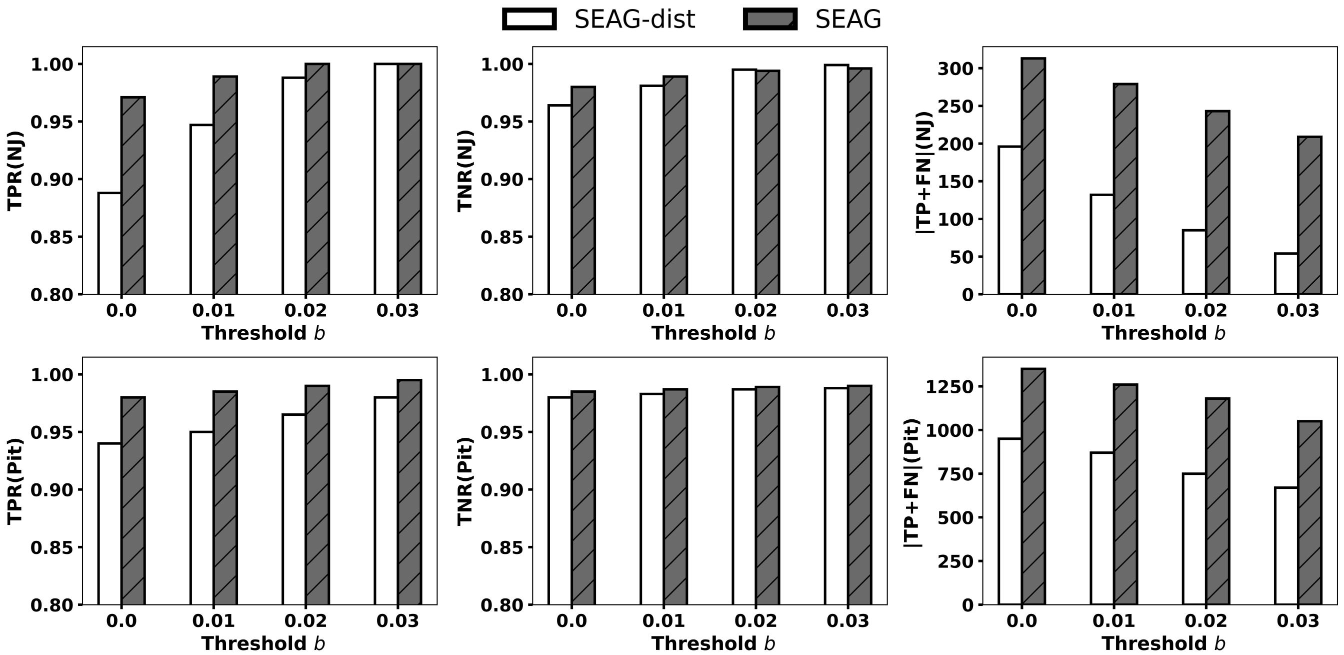


Figure 5: The effect of similarity threshold b on the quality and quantity of automatic label generation.

Table 4: Positive and negative label generation results.

Datasets	Models	Positive label generated			Negative label generated			$\|PSet\|$	$\|NSet\|$
		TP	FN	TPR	TN	FP	TNR		
NanJing	SEAG (-dist)	54	0	100%	872	1	99.8855%	55	872
	SEAG	214	0	100%	3650	18	99.5093%	232	3650
Pittsburgh	SEAG (-dist)	600	11	98.1997%	10683	106	99.0175%	706	10694
	SEAG	976	2	99.7955%	16132	102	99.3717%	1078	16134

is not considered. The results show that as b increases from 0 to 0.03, the TPR and TNR of both strategies gradually increase; the number of labels generated by SEAG gradually decreases, but still meets the training requirements. When $b = 0.03$, the TNR of the Nanjing and Pittsburgh datasets reach 99.51% and 99.37%, and the TPR reaches 100% and 99.80%, respectively, proving the effectiveness of SEAG. Meanwhile, the differences in the histograms indicate that considering spatial distance helps to improve the performance of SEAG, e.g. two entities with lower name similarity due to abbreviation have a higher probability of matching due to their closer spatial distance. Second, we calculated the effect of label generation for the Nanjing dataset and the Pittsburgh dataset when the threshold $b = 0.03$ using the SEAG and SEAG(-dist) methods, respectively, as shown in Table 4. Among them, the Pittsburgh dataset performs poorly, which is due to the fact that this dataset has a large number of missing "address" attributes, which limits the assessment of semantic similarity by SEAG.

4.5 Ablation study

We performed an ablation study of MRG-SER to evaluate the effectiveness of the different components. The results are shown Fig. 6, where the labels listed have the following meanings: MRG-SER (complete framework), w/o Dist Emb (MRG-SER without considering distance features) and w/o Graph Emb (MRG-SER without considering graphical features). The results show that the F1 scores of MRG-SER are significantly higher than those of w/o Dist Emb and w/o Graph Emb. And w/o Dist Emb has the lowest F1 scores for both. This is due to the presence of a large number of entities in the dataset with similar distances but inconsistent textual information. On all metrics, w/o Graph Emb performs slightly worse than MRG-SER, indicating that considering spatial neighbourhood features helps to improve the accuracy of SER.

5 CONCLUSION

We propose MRG-SER, a framework based on multi-relation graphs for self-supervised spatial entity resolution, which can efficiently perform spatial entity resolution tasks with zero-labelled data. Experimental results show that MRG-SER outperforms existing methods in terms of accuracy and effectiveness. These results can be attributed to the high quality of SEAG, the full capture of spatial neighbourhood relationships between entities, and the sufficient extraction of entity features. In the future, we plan to design a graph

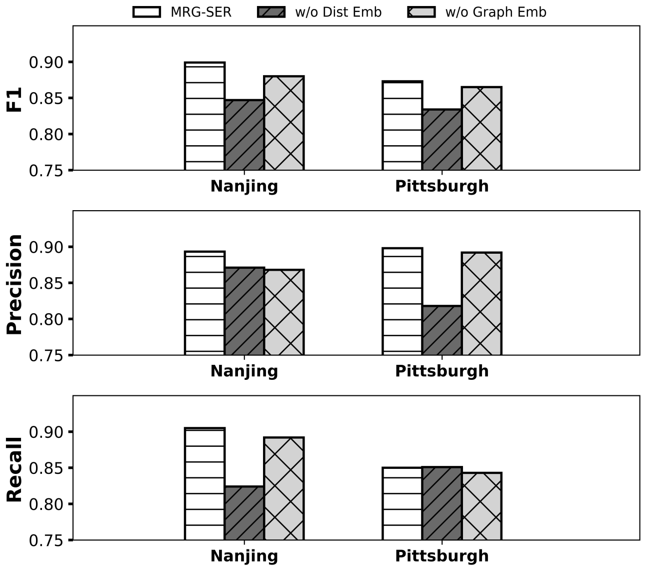


Figure 6: Effectiveness of different components in MRG-SER.

feature engineering solution for spatial entity resolution instead of AttrGNN.

REFERENCES

[1] Spiros Athanasiou, Giorgos Giannopoulos, Damien Graux, Nikos Karagiannakis, Jens Lehmann, Axel-Cyrille Ngonga Ngomo, Kostas Patroumpas, Mohamed Ahmed Sherif, and Dimitrios Skoutas. 2019. Big POI data integration with Linked Data technologies. In *In Proceedings of the 22nd International Conference on Extending Database Technology (EDBT)*. 477–488.

[2] Pasquale Balsebre, Dezhong Yao, Gao Cong, and Zhen Hai. 2022. Geospatial entity resolution. In *In Proceedings of the ACM Web Conference (WWW)*. 3061–3070.

[3] Pasquale Balsebre, Dezhong Yao, Gao Cong, Weiming Huang, and Zhen Hai. 2023. Mining Geospatial Relationships from Text. *The ACM on Management of Data* 1, 1 (2023), 1–26.

[4] Riccardo Cappuzzo, Paolo Papotti, and Saravanan Thirumuruganathan. 2020. Creating embeddings of heterogeneous relational datasets for data integration tasks. In *In Proceedings of the ACM International Conference on Management of Data (SIGMOD)*. 1335–1349.

[5] Yue Deng, An Luo, Jiping Liu, and Yong Wang. 2019. Point of interest matching between different geospatial datasets. *ISPRS International Journal of Geo-Information* 8, 10 (2019), 435.

[6] Xin Luna Dong and Theodoros Rekatsinas. 2018. Data integration and machine learning: A natural synergy. In *Proceedings of the 2018 international conference on management of data*. 1645–1650.

[7] Wenfei Fan, Xibei Jia, Jianzhong Li, and Shuai Ma. 2009. Reasoning about record matching rules. *The VLDB Endowment* 2, 1 (2009), 407–418.

[8] Congcong Ge, Pengfei Wang, Lu Chen, Xiaoze Liu, Baihua Zheng, and Yunjun Gao. 2021. CollaborEM: a self-supervised entity matching framework using multi-features collaboration. *IEEE Transactions on Knowledge and Data Engineering)* (2021), 12139–12152.

[9] Jiacheng Huang, Wei Hu, Zhifeng Bao, and Yuzhong Qu. 2020. Crowdsourced collective entity resolution with relational match propagation. In *In Proceedings of the 36th IEEE International Conference on Data Engineering (ICDE)*. 37–48.

[10] Yong Huh, Kiyun Yu, and Joon Heo. 2011. Detecting conjugate-point pairs for map alignment between two polygon datasets. *Computers, Environment and Urban Systems* 35, 3 (2011), 250–262.

[11] Suela Isaj, Torben Bach Pedersen, and Esteban Zimányi. 2022. Multi-Source Spatial Entity Linkage. *IEEE Transactions on Knowledge and Data Engineering* 34, 3 (2022), 1344–1358.

[12] Jungo Kasai, Kun Qian, Sairam Gurajada, Yunyao Li, and Lucian Popa. 2019. Low-resource Deep Entity Resolution with Transfer and Active Learning. In *In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*. 5851–5861.

[13] Ioannis Koumarelas, Thorsten Papenbrock, and Felix Naumann. 2020. MDedup: Duplicate detection with matching dependencies. *The VLDB Endowment* 13, 5 (2020), 712–725.

[14] Bing Li, Yukai Miao, Yaoshu Wang, Yifang Sun, and Wei Wang. 2021. Improving the efficiency and effectiveness for bert-based entity resolution. In *In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. 13226–13233.

[15] Bing Li, Wei Wang, Yifang Sun, Linhan Zhang, Muhammad Asif Ali, and Yi Wang. 2020. Grapher: Token-centric entity resolution with graph convolutional neural networks. In *In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. 8172–8179.

[16] Yuliang Li, Jinfeng Li, Yoshihiko Suhara, AnHai Doan, and Wang-Chiew Tan. 2020. Deep entity matching with pre-trained language models. *The VLDB Endowment* (2020), 50–60.

[17] Zhiyuan Liu, Yixin Cao, Liangming Pan, Juanzi Li, Zhiyuan Liu, and Tat-Seng Chua. 2020. Exploring and Evaluating Attributes, Values, and Structures for Entity Alignment. In *In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 6355–6364.

[18] Ralph Peeters and Christian Bizer. 2021. Dual-objective fine-tuning of BERT for entity matching. *The VLDB Endowment* 14 (2021), 1913–1921.

[19] Anna Primpeli and Christian Bizer. 2021. Graph-boosted active learning for multi-source entity resolution. In *In Proceedings of the 20th International Semantic Web Conference (ISWC)*. 182–199.

[20] Juan J Ruiz-Lendínez, Manuel A Ureña-Cámara, and Francisco J Ariza-López. 2017. A Polygon and Point-Based Approach to Matching Geospatial Features. *ISPRS International Journal of Geo-Information* 6, 12 (2017), 399.

[21] Eliyahu Safra, Yaron Kanza, Yehoshua Sagiv, Catriel Beeri, and Yerach Doytsher. 2010. Location-based algorithms for finding sets of corresponding objects over several geo-spatial data sets. *International Journal of Geographical Information Science* 24, 1 (2010), 69–106.

[22] Michael Schäfers and Udo W Lipeck. 2014. SimMatching: adaptable road network matching for efficient and scalable spatial data integration. In *In Proceedings of the 1st International Conference on Advances in Geographic Information Systems PhD Workshop (SIGSPATIAL)*. 1–5.

[23] Vivek Sehgal, Lise Getoor, and Peter D Viechnicki. 2006. Entity resolution in geospatial data integration. In *In Proceedings of the 14th annual ACM international symposium on Advances in geographic information systems (GIS)*. 83–90.

[24] Setu Shah, Vamsi Meduri, and Mohamed Sarwat. 2021. GEM: An efficient entity matching framework for geospatial data. In *In Proceedings of the 29th International Conference on Advances in Geographic Information Systems (SIGSPATIAL)*. 346–349.

[25] Vivek R Shivaprabhu, Booma Sowkarthiga Balasubramani, and Isabel F Cruz. 2017. Ontology-based instance matching for geospatial urban data integration. In *In Proceedings of the 3rd ACM SIGSPATIAL Workshop on Smart Cities and Urban Analytics (UrbanGIS)*. 1–8.

[26] Claus Stadler, Jens Lehmann, Konrad Höffner, and Sören Auer. 2012. Linkedgeodata: A core for a web of spatial open data. *Semantic Web journal* 3, 4 (2012), 333–354.

[27] Xiaohua Tong, Dan Liang, and Yanmin Jin. 2014. A linear road object matching method for conflation based on optimization and logistic regression. *International Journal of Geographical Information Science* 28, 4 (2014), 824–846.

[28] Norases Vesdapunt, Kedar Bellare, and Nilesh Dalvi. 2014. Crowdsourcing algorithms for entity resolution. *The VLDB Endowment* 7, 12 (2014), 1071–1082.

[29] Jiannan Wang, Guoliang Li, Jeffrey Xu Yu, and Jianhua Feng. 2011. Entity matching: How similar is similar. *The VLDB Endowment* 4, 10 (2011), 622–633.

[30] Renzhi Wu, Sanya Chaba, Saurabh Sawlani, Xu Chu, and Saravanan Thirumuruganathan. 2020. Zeroer: Entity resolution using zero labeled examples. In *In Proceedings of the ACM International Conference on Management of Data (SIGMOD)*. 1149–1164.

[31] Bisheng Yang, Yunfei Zhang, and Feng Lu. 2014. Geometric-based approach for integrating VGI POIs and road networks. *International Journal of Geographical Information Science* 28, 1 (2014), 126–147.

[32] Bisheng Yang, Yunfei Zhang, and Xuechen Luan. 2013. A probabilistic relaxation approach for matching road networks. *International Journal of Geographical Information Science* 27, 2 (2013), 319–338.

[33] Dezhong Yao, Yuhong Gu, Gao Cong, Hai Jin, and Xinqiao Lv. 2022. Entity resolution with hierarchical graph attention networks. In *In Proceedings of the ACM International Conference on Management of Data (SIGMOD)*. 429–442.

[34] Li Yu, Peiyuan Qiu, Xiliang Liu, Feng Lu, and Bo Wan. 2018. A holistic approach to aligning geospatial data with multidimensional similarity measuring. *International Journal of Digital Earth* 11, 8 (2018), 845–862.

[35] Weixin Zeng, Xiang Zhao, Wei Wang, Jiuyang Tang, and Zhen Tan. 2020. Degree-aware alignment for entities in tail. In *In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 811–820.

[36] Dongxiang Zhang, Dongsheng Li, Long Guo, and Kian-Lee Tan. 2020. Unsupervised entity resolution with blocking and graph algorithms. *IEEE Transactions on Knowledge and Data Engineering* 34, 3 (2020), 1501–1515.

[37] Chen Zhao and Yeye He. 2019. Auto-em: End-to-end fuzzy entity-matching using pre-trained deep models and transfer learning. In *In Proceedings of the World Wide Web Conference (WWW)*. 2413–2424.